Agrégation de mathématiques
Modélisation et méthodes numériques

Année 2021 – 22

Laurent DUMAS

Infos sur
http://dumas.perso.math.cnrs.fr/Agreg-UMGP.html

1ère séance : Interpolation de Lagrange

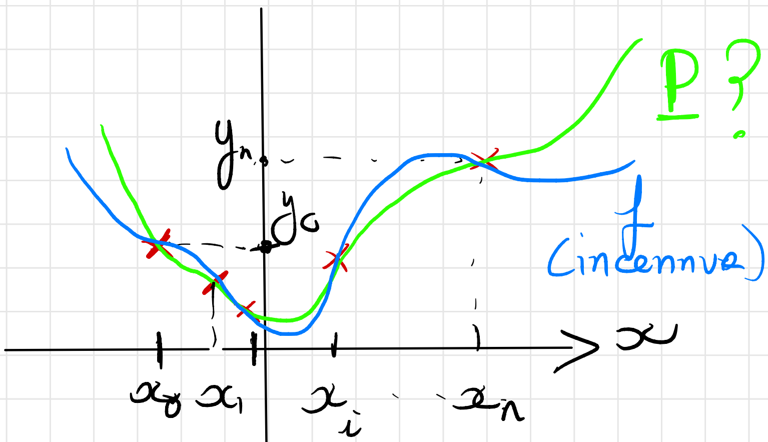
1

→ Problème posé : on dispose de $(n+1)$ données discrètes $(x_i, y_i)_{0 \leq i \leq n}$ telles que :

$$\begin{cases} * \; x_0 < x_1 < \dots < x_n \; (\in \mathbb{R}) \\ * \; \{y_0, \dots, y_n\} \in \mathbb{R}^{n+1} \end{cases}$$

On cherche une fonction polynôme $P : \mathbb{R} \to \mathbb{R}$ telle que

$\forall i \in \{0, \dots, n\}, \; P(x_i) = y_i$



(en modélisation : interpolation polynomiale de données expérimentales exemple : texte krigeage).

Questions :

* Existence et unicité de P ?
* Algorithme de construction ? (rapide, robuste, précis)
* Erreur commise par rapport à la fonction f ?

---

1) Polynôme d'interpolation de Lagrange

On a le résultat suivant d'existence et d'unicité :

Théorème : sous les notations précédentes, il existe un unique polynôme $P \in \mathbb{R}[X]$ de degré inférieur ou égal à n, tel que

$\forall i \in \{0, \dots, n\}, \; P(x_i) = y_i$

preuve

→ On définit l'application

$$\varphi : \begin{pmatrix} \mathbb{R}_n[X] \longrightarrow \mathbb{R}^{n+1} \\ P \longmapsto (P(x_0), P(x_1) \dots P(x_n)) \end{pmatrix}$$

* φ est linéaire
* $P \in \operatorname{Ker} \varphi$ ssi $P(x_0) = P(x_1) = \dots = P(x_n) = 0$

$P = 0$ (car P est de degré $\leq n$ et a $(n+1)$ racines distinctes) $\Rightarrow \varphi$ est injective

* $\dim \mathbb{R}_n[X] = \dim(\mathbb{R}^{n+1}) = n+1$

φ est donc bijective. Il existe donc un unique $P \in \mathbb{R}_n[X]$ tel que

$\forall i \in \{0, \dots n\} \; P(x_i) = y_i$

$(P = \varphi^{-1}(y_0, \dots y_n))$ //

Remarques : 1) il existe une démonstration constructive : on note

$$\boxed{l_i(x) = \frac{\prod\limits_{\substack{j=0 \\ j \neq i}}^{n} (x - x_j)}{\prod\limits_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j)}} \quad (\neq 0)$$

On a $l_i \in \mathbb{R}_n[X]$ et $l_i(x_j) = \delta_{i,j}$

En considérant P tel que

$$\boxed{P(x) = \sum_{i=0}^{n} y_i\, l_i(x)}$$

on a $P \in \mathbb{R}_n[X]$ et

$$P(x_i) = \sum_{j=0}^{n} y_j\, \delta_{ij} = y_i$$

(unicité par l'absurde)

2) On appelle P le polynôme d'interpolation de Lagrange aux points $(x_i)_{0 \leq i \leq n}$ associé à la famille $(y_i)_{0 \leq i \leq n}$

On appelle $(l_i)_{0 \leq i \leq n}$ la famille des polynômes de base de Lagrange aux points $(x_i)_{0 \leq i \leq n}$ (c'est bien une base de $\mathbb{R}_n[X]$).

2) Algorithme de construction

L'expression de P précédente ne convient pas d'un point de vue numérique pratique car :

→ elle est assez coûteuse ($\sim 4n^2$ opérations)

→ elle génère de fortes erreurs numériques lorsque les (x_i) sont proches (en raison de divisions par des valeurs proches de 0). On dit que la méthode est mal conditionnée.

→ Tout le calcul doit être repris si on rajoute un point (x_{n+1}, y_{n+1}).

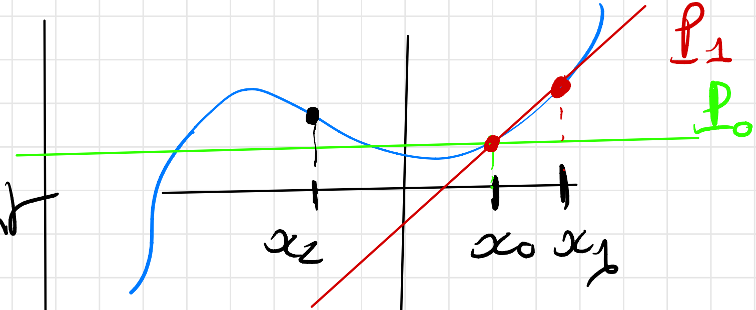
Il existe une méthode évitant ces difficultés, appelée méthode des différences divisées.

* **définition** : on note $f[x_0,\dots,x_n]$ le coefficient de P de degré n (polynôme de Lagrange aux points $(x_i, f(x_i))$ où $f : \mathbb{R} \to \mathbb{R}$).

Par exemple :

$$f[x_0] = f(x_0)$$

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



on peut montrer que

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

Plus généralement :

Proposition 1 On a

$$f[x_0,\dots,x_j] = \frac{f[x_1,\dots,x_j]-f[x_0,\dots,x_{j-1}]}{x_j-x_0}$$

Schématiquement, on peut construire la suite de ces valeurs :

$$\begin{array}{ccccccc}
f[x_0] & & f[x_0,x_1] & & \cdots & & f[x_0,\dots,x_n] \\
\vdots & & & & & & \\
 & & f[x_{n-1},x_n] & & & & \\
f[x_n] & & & & & &
\end{array}$$

(à implémenter en Python/Matlab/Scilab, ...)

(preuve : voir poly de cours ou biblio)

Les coefficients de la première ligne du tableau sont les coordonnées de P dans une base particulière

Proposition 2

Soit P le polynôme d'interpolation aux points $(x_i, f(x_i))_{0 \le i \le n}$. On a

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

(preuve : poly ou biblio)

Pour compléter cet algorithme de construction de P, on peut utiliser la méthode d'estimation de Hörner :

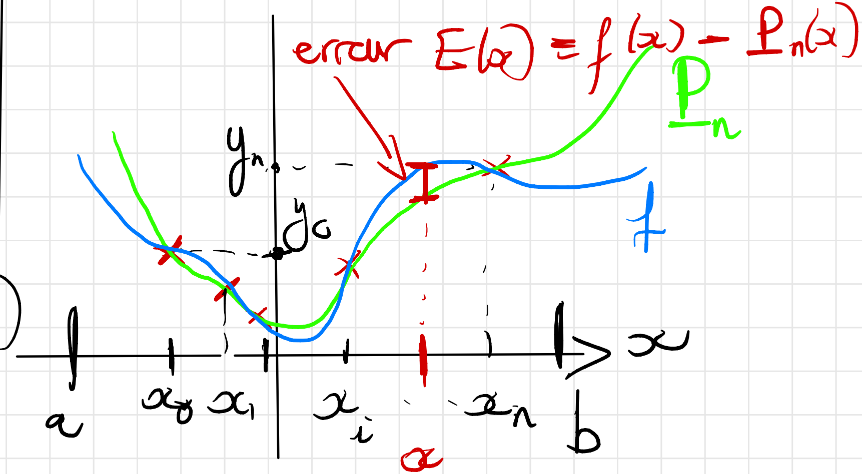
$$\boxed{\begin{cases} u_n = f[x_0,\dots,x_n] \\ u_k = f[x_0,\dots,x_k] + (x-x_k)\,u_{k+1} \end{cases}}$$

conduisant à $u_0 = P(x)$

Exercice : implémenter avec le langage de votre choix la méthode des différences divisées (Tps ≤ 1h puis 30')
(avec validation et tracé si possible)
(à envoyer à laurent.dumas@uvsq.fr)

3) Erreur d'interpolation

On dispose d'une fonction $f : \mathbb{R} \to \mathbb{R}$ continue (voire C^∞). On cherche à estimer (majorer) l'erreur commise entre $f(x)$ et $P_n(x)$ où P : P.I.L. aux points $(x_i, f(x_i))_{0 \le i \le n}$



On a le résultat suivant :

<u>Théorème</u> : On suppose $f \in C^{n+1}(\mathbb{R}, \mathbb{R})$. Sous les notations précédentes, on a

$$E_n(x) = f(x) - P_n(x) = \frac{\Pi_n(x)}{(n+1)!} f^{(n+1)}(\xi)$$

où $\begin{cases} \xi \in [x_0, x_n] \text{ et} \\ \Pi_n(x) = (x-x_0)\dots(x-x_n) \end{cases}$

preuve: classique avec l'introduction d'une fonction auxiliaire et l'application itérée du théorème de Rolle.

(à regarder et savoir reproduire)

En pratique, lorsque $n \to +\infty$, peut-on en déduire que

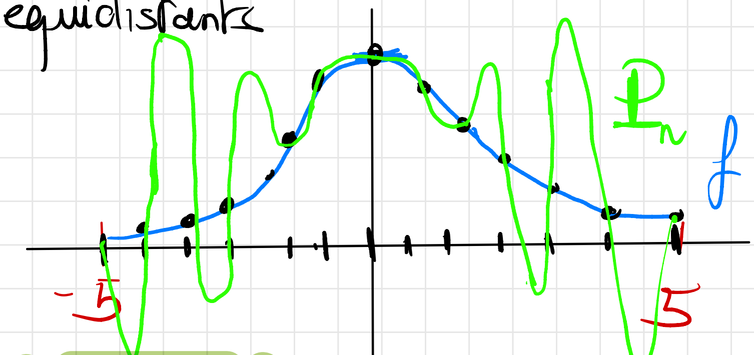
$$P_n(x) \to f(x) \quad ?$$

* Cela n'est pas vrai en général

car $\left|\frac{\Pi_{n+1}(x)}{(n+1)!}\right| \leq \frac{(b-a)^{n+1}}{(n+1)!} \to 0$

mais $|f^{(n+1)}(\xi)|$ peut tendre vers $+\infty$

Exemple : $f(x) = \frac{1}{1+x^2}$ à interpoler sur $[-5, 5]$ avec des points équidistants



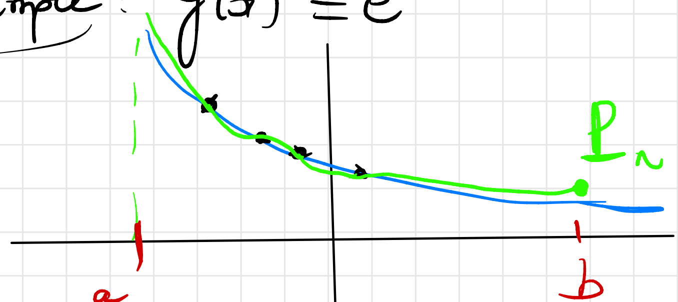
(exercice : test sur code précédent)

On parle du phénomène de non convergence de Runge (voir [Demailly])

A noter que pour un intervalle plus réduit, par exemple $[-1,1]$, il y a convergence de P_n vers f dans ce cas.

* Il y a convergence dans certains cas en particulier si $\|f^{(n)}\|_\infty$ est bornée.

Exemple : $f(x) = e^{-x}$



On a $\|f - P_n\|_{L^\infty([a,b])} \xrightarrow[n\to+\infty]{} 0$

Il existe un second résultat d'erreur, non pas ponctuelle, mais globale :

Théorème : Soit $(x_i)_{0\leq i\leq n} \in C[a,b]^{n+1}$ et $f \in \mathcal{C}([a,b],\mathbb{R})$. On a

$$\boxed{\|f - P_n\|_{L^\infty([a,b])} \leq (1+\Lambda_n)\, d(f, \mathbb{R}_n[X])}$$

où $d(f,\mathbb{R}_n[X]) = \inf_{Q\in\mathbb{R}_n[X]} \|f - Q\|_{L^\infty}$

et $\Lambda_n = \max_{x\in[a,b]} \left(\sum_{i=0}^{n} |l_i(x)| \right)$

preuve : voir poly ou biblio

En pratique, on ne peut conclure à la convergence vers 0 de

$\|f - P_n\|_{L^\infty([a,b])}$ car :

$$d(f, \mathbb{R}_n[X]) \xrightarrow[n \to \infty]{} 0$$

(Stone Weierstrass)

mais

$\Lambda_n \to +\infty$ qd $n \to +\infty$

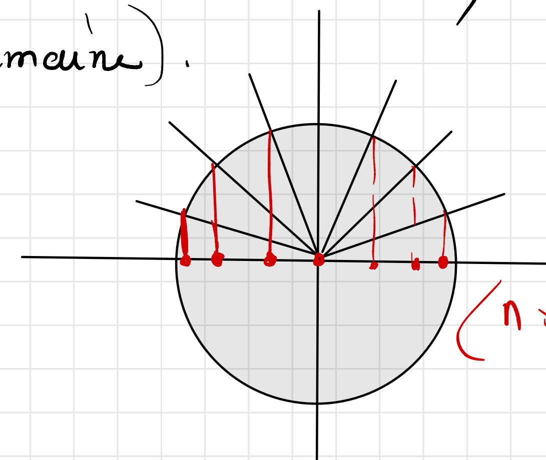
quel que soit le choix de points.

À noter qu'il existe une répartition optimale des points d'interpolation par lesquels Λ_n tend "le moins vite" vers $+\infty$. Il s'agit de la répartition de Tchebychev :

$$x_i = \cos\left(\frac{2k+1}{2n+2}\pi\right) \quad (\text{sur } [-1,1])$$

$$0 \le i \le n$$

(ils sont plutôt concentrés au bord du domaine).



(n = 6)

En conclusion, l'interpolation de Lagrange, dans un cadre appliqué, est dangereuse si n est grand.
Cependant, elle est utilisable par des valeurs de n faibles ($\leq 5-6$).
Elle présente par ailleurs une importance très forte pour construire des méthodes de quadrature (méthodes composées).

Il existe d'autres types d'interpolation mieux adaptés dans la pratique :

→ interpolation affine par morceaux
→ interpolation par splines cubiques
→ krigeage (voir texte modélisation)

* On peut aussi recourir à des méthodes d'approximation polynomiale (en norme L^∞, L^2, ...)

Exercices :
→ TD1 (en ligne ce soir)
(3 ou 4 exercices à préparer pour la prochaine séance)