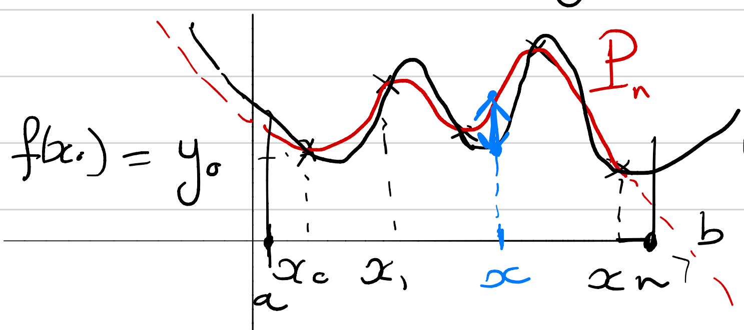
Séance 1 : Modélisation et méthodes numériques

Interpolation polynomiale

Objectif : comment reconstruire une fonction polynomiale à partir de données discrètes $(x_i, y_i)_{0 \le i \le n}$ où $x_0 < x_1 < \dots < x_n$ et $y_i \in \mathbb{R}$.



Qns : * existence et unicité de P_n ?
* Algorithme efficace de construction ?
(* rapide, précis, robuste)
* estimation d'erreur $(\|f - P_n\|)$

1) Interpolation polynomiale de Lagrange. Définition

Théorème : il existe un unique polynôme P_n de degré $\le n$ $(P_n \in \mathbb{R}_n[x])$ tq $\forall i \in \{0, \dots, n\},\ P_n(x_i) = y_i$

preuve : 2 types de preuve (constructive ou abstraite)

* Existence : on définit la famille $(l_i)_{0 \leq i \leq n}$ de polynômes :

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

On remarque que $l_i(x_j) = \delta_{ij}$

On note alors

$$P_n(x) = \sum_{i=0}^{n} y_i l_i(x) \quad (1)$$

On a
$$\begin{cases} * \ P_n \in \mathbb{R}_n[X] \\ P_n(x_j) = \sum_{i=0}^{n} y_i \delta_{ij} = y_j \end{cases}$$

* Unicité

si R et P conviennent, on a
$$\begin{cases} * \ (R-P)(x_i) = 0, \quad 0 \leq i \leq n \\ * \ R - P \in \mathbb{R}_n[X]. \end{cases}$$

$\Rightarrow$ R - P a (n+1) racines et est donc nul.

Remarques

* On parle de la famille des polynômes de base de Lagrange associée aux points $(x_i)_{0 \leq i \leq n}$ (il s'agit d'une base de $\mathbb{R}_n[X]$)

* L'algorithme de construction (1) n'est pas satisfaisant d'un point de vue numérique (précision et coût)

2) Algorithme de construction

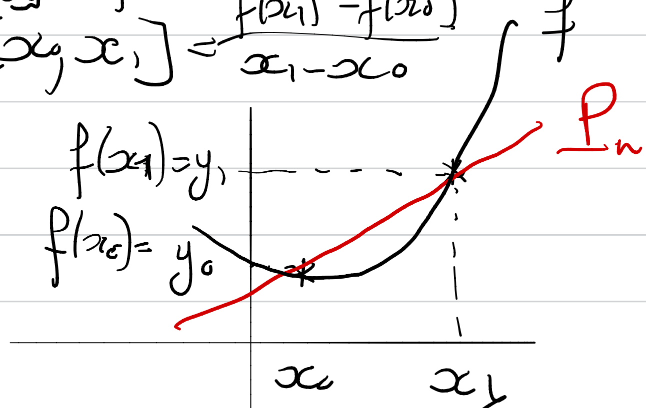
* **Def** : on note $f[x_0, \ldots, x_n]$ le coefficient de P_n (de degré n) associé à $y_i = f(x_i)$.

  (f fonction réelle quelconque)

* Exemple

→ $f[x_0] = f(x_0)$

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



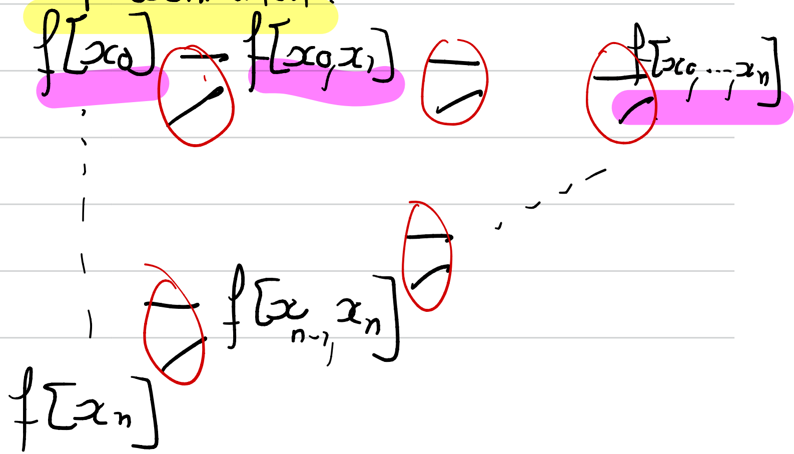
Proposition : on a la relation :

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

$(1 \leq j \leq n)$

(preuve : voir poly)

Représentation :



On a le résultat suivant :

Th : soit P_n le polynôme de Lagrange de f associé aux points $(x_i)_{0 \leq i \leq n}$.

On a :

$$P_n(x) = f[x_0] + \sum_{i=1}^{n} f[x_0, \dots x_i](x - x_0) \dots (x - x_i)$$

Preuve : récurrence sur n (poly)

Remarque :

1.) Il s'agit de l'expression de P_n dans la base $\{1, (x-x_0), (x-x_0)(x-x_1), \dots, (x-x_0)(\dots) \dots (x-x_{n-1})\}$

(base de Newton)

2) Cette expression est plus intéressante d'un pt de vue numérique :

* coût : $O(n^2)$
* précision et robustesse : OK

3) Une fois calculées les différences divisées, on utilise l'algorithme de Horner pour évaluer $P_n(x)$:

$$u_n \to u_{n-1} \dots \to u_0 = P_n(x)$$

$$\begin{cases} u_n = f[x_0, \dots x_n] \\ u_k = f[x_0, \dots x_k] + (x - x_k) u_{k+1} \quad 0 \leq k \leq n-1 \end{cases}$$

(Réf : Demailly)

3.) Estimation d'erreur

On cherche à estimer l'erreur commise

$$E_n(f)(x) = f(x) - P_n(x)$$

en un point $x \in \mathbb{R}$ quelconque.

3.1) Estimation ponctuelle

Th soit $f \in C^{n+1}([a,b], \mathbb{R})$ (avec $x_i \in [a,b]$). Soit $x \in [a,b]$

On a

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

$\xi \in]a,b[$ et $\Pi_n(x) = \prod_{i=0}^{n} (x - x_i)$

En particulier)

$$|f(x) - P_n(x)| \leq \frac{|\Pi_{n+1}(x)|}{(n+1)!} \|f^{(n+1)}\|_\infty$$

preuve : * si $x = x_i$; ok.

* Soit Q_{n+1} le polynôme :

$$Q_{n+1}(t) = \underbrace{P_n(t)}_{d^\circ n} + \frac{E_n(f)(x)}{\Pi_n(x)} \underbrace{\Pi_n(t)}_{d^\circ n+1}$$

On a

* $Q_{n+1} \in \mathbb{R}_{n+1}[X]$
* $Q_{n+1}(x_i) = f(x_i)$ $0 \leq i \leq n$
* $Q_{n+1}(x) = f(x)$

$\Rightarrow Q_{n+1}$: P.I.L. de f aux points $(x_0, \dots, x_n, x)$.

En particulier

$F(t) = f(t) - Q_{n+1}(t)$ est tel que

$F(x_0) = F(x_1) = \dots = F(x_n) = F(x) = 0$

Par Rolle successifs :

* F' s'annule $(n+1)$ fois
* F'' ---- n fois

⋮

$F^{(n+1)}$ ------ 1 fois. Ainsi

$\exists \xi \in]a,b[$ tq

$$f^{(n+1)}(\xi) = Q^{(n+1)}(\xi)$$

soit : $f^{(n+1)}(\xi) = (n+1)! \dfrac{E_n(f)(x)}{\Pi_n(x)}$ //

Remarques :

1). Lorsque $n \to +\infty$

$$\frac{|\Pi_n(x)|}{(n+1)!} \leq \frac{(b-a)^{n+1}}{(n+1)!} \xrightarrow[n \to +\infty]{} 0$$

mais on peut avoir

$$\|f^{(n+1)}\|_\infty \to +\infty$$

2) On n'a pas de résultat général donnant $\|f - P_n\|_\infty \to 0$

(cas particuliers : poly, TD)

Il existe des cas particuliers

de nonconvergence "célèbres" :

$$f(x) = \frac{1}{1+x^2} \quad \text{sur } [a,b] = [-5,5]$$

(x_i équirépartis sur $[-5,5]$)

(phénomène de Runge)

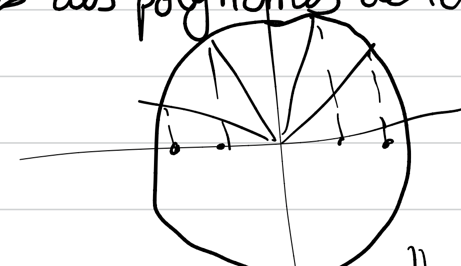
(voir TP)

3) La qualité de l'approximation dépend de la répartition des points sur $[a,b]$. Il existe une répartition "optimale" de points qui correspond aux points de Tchebychev :

$$x_i = \cos\left(\frac{2k+1}{2n+2}\pi\right)$$

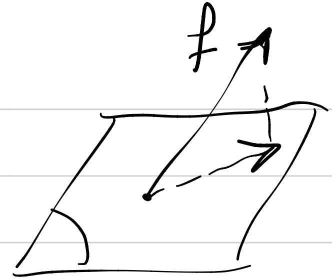
$$0 \le k \le n$$

(racines des polynômes de Tchebychev)



(voir TD et Demailly)

3.2) Estimation globale



Def : Λ_n (constante de Lebesgue associée aux (x_i)) est définie par :

$$\Lambda_n = \|\lambda_n\|_\infty \quad \text{où} \quad \lambda_n(x) = \sum_{i=0}^{n} |l_i(x)|$$

(sur $[a,b]$)

Th :

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

preuve : (voir poly)

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

Remarque :

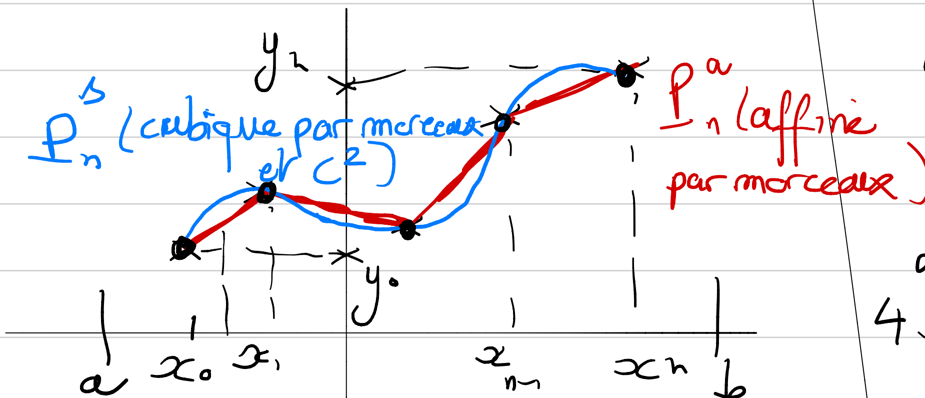
1) si $f \in C^0([a,b], \mathbb{R})$ on a

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

(Stone-Weierstrass) mais on peut montrer que

$$\Lambda_n \xrightarrow[n \to +\infty]{} +\infty$$

4) Interpolation polynomiale par morceaux



4.1) Interpolation affine par morceaux

* Construction + algorithme : OK
* Estimation d'erreur : si $f \in C^0([a,b],\mathbb{R})$

$\|P_n^a - f\|_\infty \longrightarrow 0$

si le pas de la subdivision

$\max\limits_{0\leq i\leq n-1} |x_{i+1} - x_i| \xrightarrow[n\to+\infty]{} 0$

(voir poly)

(Defaut : manque de régularité et aspect purement local)

4.2) Interpolation par splines cubiques

Soit E_s, l'espace des fonctions $s : [a,b] \to \mathbb{R}$ tq

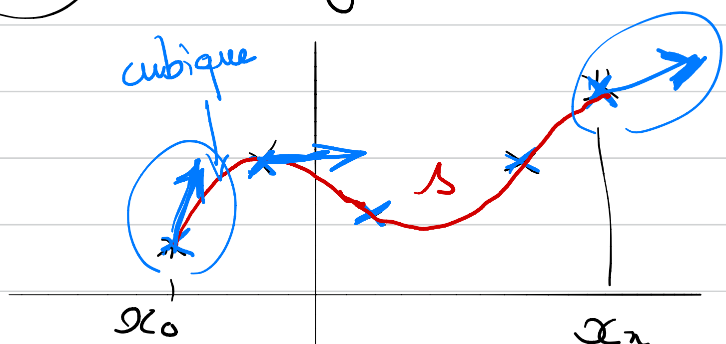
* $s_{/[x_i, x_{i+1}]} \in \mathbb{R}_3[X]$
* $s \in C^2([a,b],\mathbb{R})$

Th : l'application

$$\sigma \left(\begin{array}{l} E_s \longrightarrow \mathbb{R}^{n+3} \\ s \longmapsto (s(x_0), \dots, s(x_n), s'(x_0), s'(x_n)) \end{array} \right.$$

est un isomorphisme d'espaces vectoriels.

Preuve : voir poly.



Remarque : 1) on peut aussi imposer les 2 conditions :

$$s''(x_0) = s''(x_n) = 0$$

(spline naturelle)

2) Il existe un algorithme rapide (en $O(n)$) basé sur l'inversion d'une matrice tridiagonale définie positive. (voir cours algèbre numérique).

3) Il existe une estimation d'erreur de $\|f - P_n^\Delta\|_\infty$ en $O\left(\frac{1}{n^4}\right)$ si f est C^4.

4) Il s'agit de la méthode la plus utilisée en pratique (LaTeX, jeux vidéos).

Semaine prochaine (à préparer)

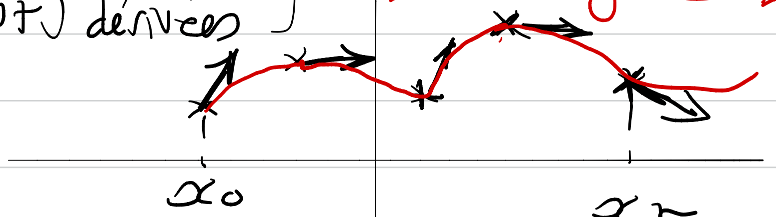
* TP 1, Ex 1 (avec Python)
  (voir eventuellement Ex 3)

* TD 1, Ex 1 (à faire) (voir Ex 3, 4)

* Exercice 2 : interpolation de Hermite

(n+1) pts
(n+1) dérivées

P : de degré 2n+1

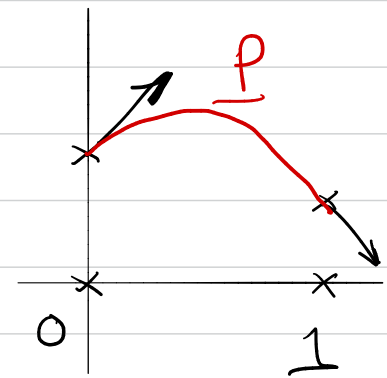


(i)

$\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}$

$\exists! \, P \in \mathbb{R}_3[x]$ tq

$$\left.\begin{array}{l} P(0) = \alpha_1 \\ P'(0) = \alpha_2 \\ P(1) = \alpha_3 \\ P'(1) = \alpha_4 \end{array}\right\}$$



* $P(x) = a + bx + cx^2 + dx^3$
  puis on exprime a, b, c, d en fonction de $\alpha_1, \alpha_2, \alpha_3$ et α_4.

Aussi :

$$\Phi : \left(\begin{array}{l} \mathbb{R}_3[x] \longrightarrow \mathbb{R}^4 \\ P \longmapsto (P(0), P'(0), P(1), P'(1)) \end{array} \right)$$

* Φ est injective car si

$P(0) = P'(0) = P(1) = P'(1) = 0$, 0 et 1 sont racines doubles

$\Rightarrow P = 0$ ($P \in \mathbb{R}_3[x]$)

* $\dim \mathbb{R}_3[x] = \dim(\mathbb{R}^4)$

Φ est bijective//

$$P = \Phi^{-1}((\alpha_1, \alpha_2, \alpha_3, \alpha_4))$$

ii) $\left.\begin{array}{l} P_1(0) = 1 \\ P_1'(0) = 0 \\ P_1(1) = 0 \\ P_1'(1) = 0 \end{array}\right\}$

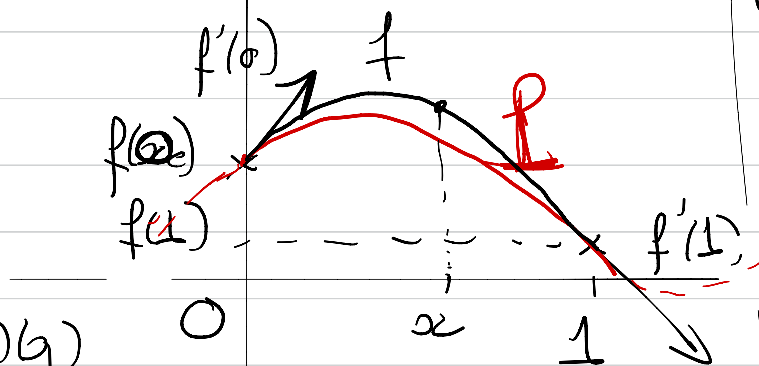
le plus simple est de revenir à l'expression générale de P en fonction des (α_i) :

$$\left\{\begin{array}{l} a = \alpha_1 \\ b = \alpha_2 \\ a + b + c + d = \alpha_3 \\ b + 2c + 3d = \alpha_4 \end{array}\right.$$

et en déduire P_1, P_2, P_3, P_4.

On a $P(x) = \sum_{i=1}^{4} \alpha_i P_i(x)$ par unicité//

(iii)



$$\underbrace{f(x)-P(x)}_{E(f)(x)} = \frac{\pi(x)}{4!} f^{(4)}(\xi) \ ?$$

$$\pi(x) = \underbrace{x^2}\underbrace{(x-1)^2}$$

(0 et 1 racines doubles)

On note (si $x \in]0,1[$)

$$Q(t) = \underbrace{P(t)}_{d^\circ 3} + \frac{E(f)(x)}{\pi(x)} \underbrace{\pi(t)}_{d^\circ 4}$$

On a :

* $Q \in \mathbb{R}_4[X]$
* $\begin{cases} Q(0) = P(0) = f(0) \\ Q'(0) = f'(0) \\ Q(1) = f(1) \\ Q'(1) = f'(1) \\ Q(x) = f(x) \end{cases}$

$R = Q - f$

$R \to \mid 0 \ * \quad * \ x \quad * \ 1$

$R' \to \mid \ * \quad \times \quad \times \quad *$

* R' s'annule 4 fois (Rolle)
* $R^{(4)}$ - - - 1 fois (Rolle)

soit $\underbrace{Q^{(4)}(\xi)}_{\frac{E(f)(x)}{\Pi(x)}4!} = f^{(4)}(\xi)$

$$\boxed{E(f)(x) = f^{(4)}(\xi)\frac{\Pi(x)}{4!}}$$

(Polynômes de Hermite = extension des polynômes de Lagrange)

Mêmes restrictions en termes de convergence ...