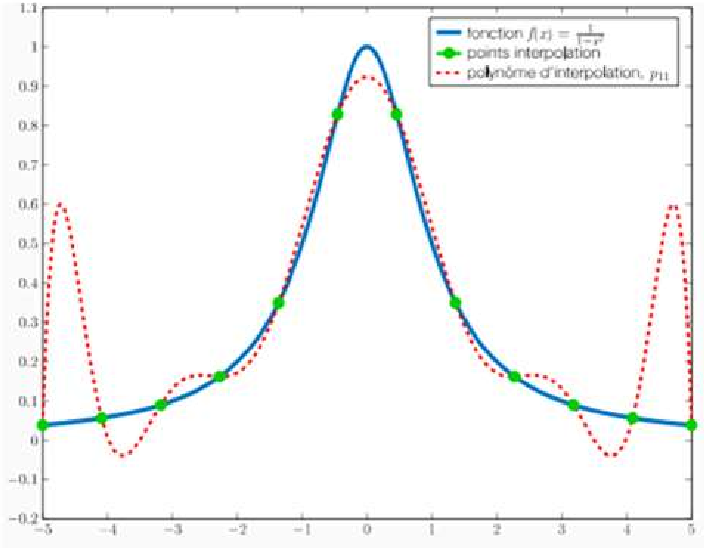
Remarque : i le nombre de points est élevé

– **Attention !**

On n'obtient pas de meilleurs résultats en augmentant le degré du polynôme d'interpolation. Supposons que nous cherchions à approcher la fonction $f(x) = 1/(1 + 25x^2)$ sur l'intervalle [−1, 1]. Un polynôme d'interpolation de degré trop élevé sur des points équidistants conduit à de fortes oscillations aux bords de l'intervalle (cf. Figure 1.1). Ce phénomène est connu sous le nom de phénomène de Runge. En pratique, on effectue des interpolations avec des polynômes de degrés faibles sur des petits intervalles plutôt que des polynômes de degrés élevés sur de grands intervalles.

Si le nombre de points est élevé

- entre les points, le polynôme fait ce qu'il veut ; on ne peut pas l'empêcher d'osciller !!!
  et plus son degré est élevé plus il est susceptible d'osciller !
- Le phénomène de Runge Kutta



Phénomène Runge-Kutta : fonction f(x)= $\dfrac{1}{1+x^2}$

Interpolation 12 points (noeuds) polynôme de degré 11

Trouver d'autres alternatives !

Interpolation par splines cubiques

Principe :

- on approche la courbe par morceaux (localement)
- on prend des polynômes de degré faible (3) pour éviter les oscillations

Comment

- on décompose l’espace de définition (des points) en un ensemble contigu d’intervalles sur lesquels on applique des interpolations polynômiales de degré 3

➔ Résultat un ensemble de polynômes définis de façon continue et « lisse »

Splines cubiques : définition

- Définition :
  - On appelle spline cubique (d'interpolation) une fonction notée *g*, qui vérifie les propriétés suivantes :
    - $g \in C^2[a,b]$ (*g* est deux fois continument dérivable),
    - *g* coïncide sur chaque intervalle $[x_i, x_{i+i}]$ avec un polynôme de degré inférieur ou égal à 3,
    - $g(x_i) = y_i \quad pour\ i = 0 \ldots n$
    - $g' \quad est\ continue\ sur\ [x_0, x_n]$
    - $g'' \quad est\ continue\ sur\ [x_0, x_n]$

Splines cubiques : définition

- Remarque (voir plus loin):
  - Il faut des conditions supplémentaires pour définir la spline d'interpolation **de façon unique**
  - Ex. de conditions supplémentaires : conditions aux limites
    - $g''(a) = g''(b) = 0$ spline naturelle.

- Remarques :
  - Sur le plan pratique, ces conditions permettent d'avoir une courbe continue et d'aspect lisse ;
  - Forme ≡ forme d'une barre souple soumise à des contraintes physiques
  - Sur le plan technique, cela permet de poser les équations qui permettent d'obtenir l'expression mathématique de la fonction.

Splines : illustration

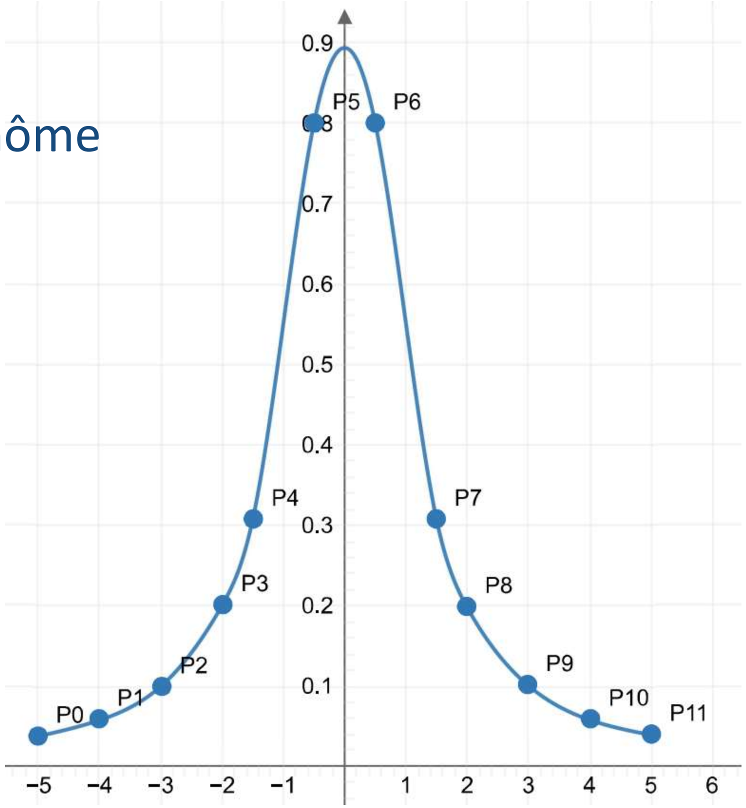
Fonction formée par plusieurs morceaux de polynôme

$$P_1(x) = a_1 x^3 + b_1 x^2 + c_1 x + d_1$$

$$\vdots$$

$$\vdots$$

$$P_{11}(x) = a_{11} x^3 + b_{11} x^2 + c_2 x + d_{11}$$



Splines cubiques : équations

- Déterminer la spline d'interpolation
  - $\mathcal{S}$ coïncide sur chaque intervalle $[x_i; x_{i+1}]$ avec un polynôme de degré inférieur ou égal à 3
  - $\mathcal{S}''$ *est de degré 1 et est déterminé par 2 valeurs:*
    - *mi = S''(xi) et   mi+1 = S''(xi+1)*
  - *Notations :*
    - $h_i = x_{i+1} - x_i$ pour $i = 0 \ldots n-1$
    - $\delta_i = [x_i; x_{i+1}]$
    - $S_i(x)$ le polynôme de degré 3 qui coïncide avec $\mathcal{S}$ sur l'intervalle δ_i

Splines cubiques : équations

$S''_i(x)$ est linéaire : on peut l'estimer par la méthode de Lagrange

- $\forall\, x \in \delta_i$

$$\boxed{S_i''(x) = m_{i+1}\frac{x - x_i}{h_i} + m_i\frac{x_{i+1} - x}{h_i}}$$

- Pour obtenir $\boldsymbol{S_i'}(x)$ on intègre : $\boldsymbol{S_i''}(x)$

$$\boxed{S_i'(x) = m_{i+1}\frac{(x - x_i)^2}{2h_i} - m_i\frac{(x_{i+1} - x)^2}{2h_i} + a_i}$$

(a_i constante d'intégration)

Splines cubiques : équations

- Pour calculer $\boldsymbol{S}_i(x)$, on intègre $\boldsymbol{S'}_i(x)$

$$\boxed{S_i(x) = m_{i+1}\frac{(x-x_i)^3}{6h_i} + m_i\frac{(x_{i+1}-x)^3}{6h_i} + a_i(x-x_i) + b_i}$$

(b_i constante d'intégration)

- $S_i(xi) = yi \longrightarrow y_i = \frac{m_i {h_i}^2}{6} + b_i$ 1

- $S_i(xi+1) = yi+1 \longrightarrow y_{i+1} = \frac{m_{i+1} {h_i}^2}{6} + a_i h_i + b_i$ 2

Splines cubiques : équations

- S'(x) est continue : $S_i'(x_i) = -m_i \frac{h_i}{2} + a_i = m_i \frac{h_{i-1}}{2} + a_{i-1} = S_{i-1}'(x_i)$ 3

- 1 *et* 2 $a_i = \frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{6}(m_{i+1} - m_i)$

- *on remplace les a_i dans :* 3

$$\boxed{h_{i-1}m_{i-1} + 2(h_i + h_{i-1})m_i + h_i m_{i+1} = 6\left(\frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{h_{i-1}}(y_i - y_{i-1})\right)}$$ 4

Splines cubiques : équations

- Rappel : on cherche les m_i (n+1 inconnues)
  - on a seulement (n-1) équations grâce aux données

  - Pour obtenir une solution unique il manques 2 équations :
  - il faut rajouter 2 conditions → par exemple condition aux limites
    - $m_0 = m_n = 0$ *(appelée : spline naturelle)*

Splines cubiques : calcul des coefficients

4 $$\boxed{h_{i-1}m_{i-1} + 2(h_i + h_{i-1})m_i + h_i m_{i+1} = 6\left(\frac{1}{h_i}(y_{i+1} - y_i) - \frac{1}{h_{i-1}}(y_i - y_{i-1})\right)}$$

- Ex de résolution avec $h_i = x_{i+1} - x_i$ (h_i constant) :
  - $$m_{i-1} + 4m_i + m_{i+1} = \frac{1}{h^2}(y_{i-1} - 2y_i + y_{i+1}) = f_i$$
  - Forme matricielle

$$S * M = f = \begin{pmatrix} 4 & 1 & & & 0 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ \cdots \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ \cdots \\ f_{n-1} \end{pmatrix}$$

  - S matrice inversible, tridiagonale, à diagonale strictement dominante, système facile à résoudre.

Splines cubiques : algorithme

pour $i = 2; n-1$

$S(i,i) \leftarrow 2(h_i + h_{i-1})$

$S(i,i-1) \leftarrow h_{i-1}$

$S(i,i+1) \leftarrow 2h_i$

$f(i-1) \leftarrow 6\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)$

fin pour

$T \leftarrow S(2:n-1, 2:n-1)$

$m \leftarrow S/f$

$m \leftarrow [0, m, 0]$

pour $i = 1; n-1$

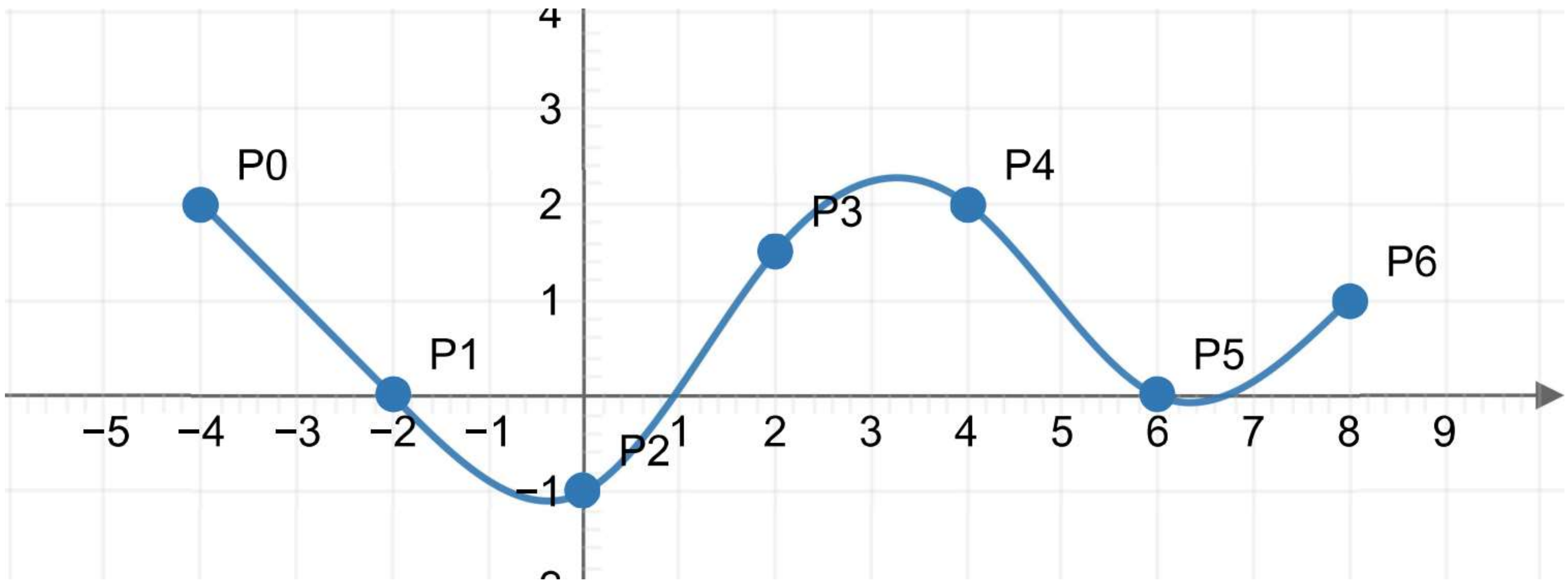
$a(i) \leftarrow \frac{1}{h_i}(y_{i+1} - y_i) - \frac{h_i}{6}(m_{i+1} - m_i)$

$b(i) \leftarrow y(i) - \frac{m_i h_i}{6}$

fin pour

Splines cubiques : exemple

- Ex : avec 7 points → spline cubique d'interpolation



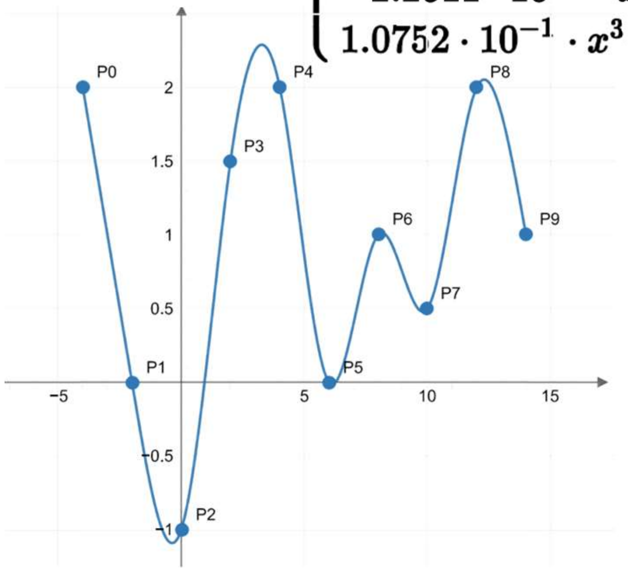
Splines cubiques : exemple

- Ex : avec 9 points → spline cubique d'interpolation

$$P_0(-4|2); P_1(-2|0); P_2(0|-1); P_3(2|1.5); P_4(4|2); P_5(6|0); P_6(8|1); P_7(10|0.5); P_8(12|2); P_9(14|1)$$

Equation

$$f(x) = \begin{cases} -4.9636 \cdot 10^{-4} \cdot x^3 + -5.9563 \cdot 10^{-3} \cdot x^2 + -1.0218 \cdot x + -2.0238, & \text{if } x \in [-4, -2], \\ 1.2748 \cdot 10^{-1} \cdot x^3 + 7.6191 \cdot 10^{-1} \cdot x^2 + 5.1390 \cdot 10^{-1} \cdot x + -1.0000, & \text{if } x \in (-2, 0], \\ -1.9693 \cdot 10^{-1} \cdot x^3 + 7.6191 \cdot 10^{-1} \cdot x^2 + 5.1390 \cdot 10^{-1} \cdot x + -1.0000, & \text{if } x \in (0, 2], \\ -2.7258 \cdot 10^{-2} \cdot x^3 + -2.5612 \cdot 10^{-1} \cdot x^2 + 2.5500 \cdot x + -2.3574, & \text{if } x \in (2, 4], \\ 2.4346 \cdot 10^{-1} \cdot x^3 + -3.5048 \cdot x^2 + 1.5545 \cdot 10^{1} \cdot x + -1.9684 \cdot 10^{1}, & \text{if } x \in (4, 6], \\ -2.5910 \cdot 10^{-1} \cdot x^3 + 5.5413 \cdot x^2 + -3.8732 \cdot 10^{1} \cdot x + 8.8870 \cdot 10^{1}, & \text{if } x \in (6, 8], \\ 2.3043 \cdot 10^{-1} \cdot x^3 + -6.2073 \cdot x^2 + 5.5257 \cdot 10^{1} \cdot x + -1.6177 \cdot 10^{2}, & \text{if } x \in (8, 10], \\ -2.2511 \cdot 10^{-1} \cdot x^3 + 7.4589 \cdot x^2 + -8.1405 \cdot 10^{1} \cdot x + 2.9377 \cdot 10^{2}, & \text{if } x \in (10, 12], \\ 1.0752 \cdot 10^{-1} \cdot x^3 + -4.5159 \cdot x^2 + 6.2293 \cdot 10^{1} \cdot x + -2.8102 \cdot 10^{2}, & \text{if } x \in (12, 14]. \end{cases}$$



Conclusion

- Interpolation polynomiale
  - évaluer la fonction en un point : Polynôme de Lagrange -> méthode de Neville
  - *compiler* la fonction : Polynôme de Newton

- Interpolation polynomiale par morceau : splines
  - spline cubique d'interpolation : passage par les nœuds (points d'interpolation), mais on limite les oscillations.
  - spline cubique d'approximation : on régule mieux la fonction, mais minimise la distance aux nœuds (les points de passage)

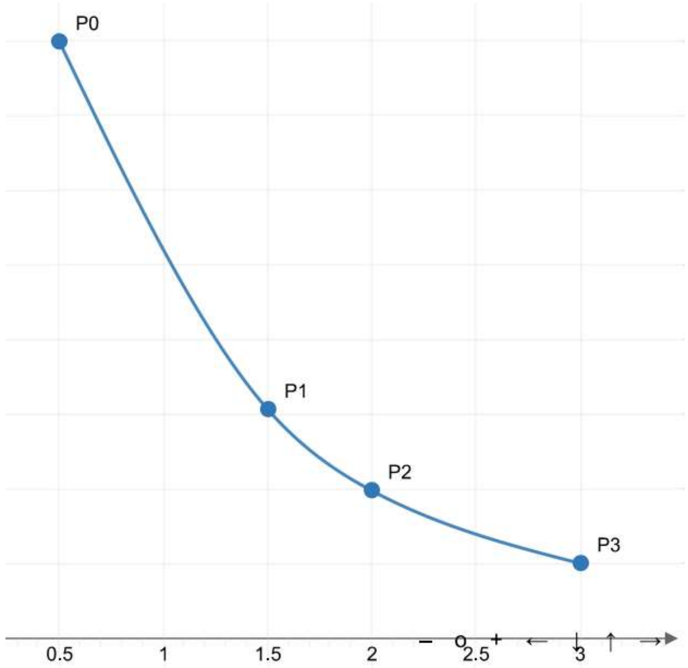
Exemple

X	0.5	1.5	2.0	3.0
Y	0.7999	0.3107	0.1981	0.1008

- Après résolution du système :

Equation

$$f(x) = \begin{cases} 8.7320 \cdot 10^{-2} \cdot x^3 + -1.3098 \cdot 10^{-1} \cdot x^2 + -5.1463 \cdot 10^{-1} \cdot x + 1.0790, & \text{si } x \in [0.5, 1.5], \\ -1.2328 \cdot 10^{-1} \cdot x^3 + 8.1672 \cdot 10^{-1} \cdot x^2 + -1.9362 \cdot x + 1.7898, & \text{si } x \in (1.5, 2], \\ -2.5680 \cdot 10^{-2} \cdot x^3 + 2.3112 \cdot 10^{-1} \cdot x^2 + -7.6498 \cdot 10^{-1} \cdot x + 1.0090, & \text{si } x \in (2, 3]. \end{cases}$$

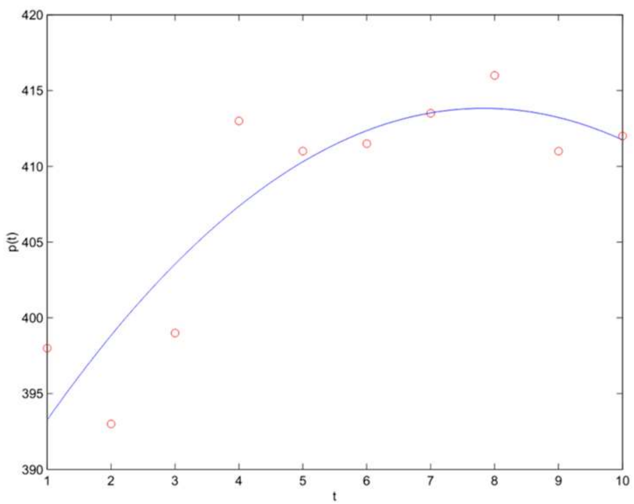


Approximation aux moindres carrés

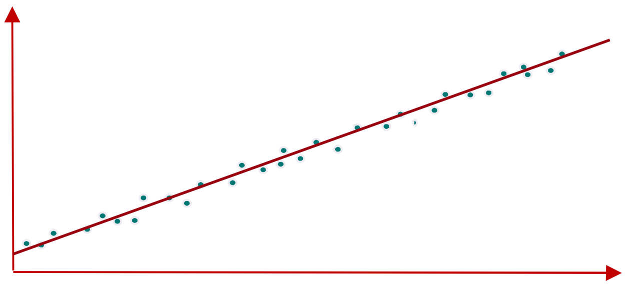
- Exemples

(1) Typiquement, on suppose disposer d'un jeu mesure (x_i, y_i)

on cherche f : $f(x, a_0, a_1, \ldots.a_n)$



(2) $g(x, a0, a1)= a_0+ a_1x$



- **Données** : un ensemble de points $(x_0, y_0),(x_1, y_1), \ldots, (x_r, y_r)$.
- **On cherche** : On cherche une fonction f dont la courbe approche au mieux tous les points.
  - Le modèle de f est fixée à l'avance (par exemple un polynôme de degré < r).
  - $f(x, a_0, a_1, \ldots, a_r)$
  - $a_0, a_1, \ldots, a_r$ sont des constantes à régler en fonction des points de l'ensemble.
- but : minimiser la distance entre f et l'ensemble des points.

Cas d'un polynôme

- Données : $(x_0, y_0),(x_1, y_1), \ldots, (x_n, y_p)$.

Si on représente la fonction par un polynôme de degré n ($n \leq p$)

$$f(x) = \sum_{k=0}^{n} a_k x^k$$

Si n = p : on est dans le cas de l'interpolation

(n+1) équations à (n+1) inconnues)

$$f(x_i) = y_i \Leftrightarrow \sum_{k=0}^{n} a_k x_i^k = y_i \qquad i = 1, \cdots, p$$

Cas d'un polynôme (suite)

$$f(x_i) = y_i \Leftrightarrow \sum_{k=0}^{n} a_k x_i^k = y_i \qquad i = 0, \cdots, p$$

Si $n < p$: on a une approximation

(p équations à n inconnues : plus d'équations que d'inconnues)

$$\min_A \sum_{i=0}^{p} (f(x_i) - y_i)^2$$

On minimise la somme des distances entre les valeurs théoriques $f(x_i)$ et les (p+1) données y_i (les carrés des erreurs)

Ce genre de problème intervient lorsque l'on cherche à modéliser à partir de données, les valeurs $(x_i \,;\, y_i)$ et bi sont souvent des résultats d'expériences ou de mesures, pouvant être entaches d'erreurs.

Cas d'un polynôme (suite)

$$f(x_i) = \sum_{k=0}^{n} a_k x_i^k \qquad pour\ (i = 0, \cdots, p)$$

n le degré du polynôme. En développant les équations on obtient :

$$\begin{cases} a_0 x_0^0 + a_1 x_0^1 + a_2 x_0^2 + \cdots + a_n x_0^n = y_o \\ a_0 x_1^0 + a_1 x_1^1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1 \\ \vdots \\ a_0 x_j^0 + a_1 x_j^1 + a_2 x_j^2 + \cdots + a_n x_j^n = y_j \\ \vdots \\ a_0 x_p^0 + a_1 x_p^1 + a_2 x_p^2 + \cdots + a_n x_p^n = y_p \end{cases}$$

Cas d'un polynôme (suite)

On note : $A = \{a_0, \cdots a_{n-1}\}$, on cherche le polynôme :

$$\min_{a} \sum_{i=0}^{p} (f(x_i) - y_i)^2 = \min_{A} \varphi(A)$$

$$\varphi(A) = \sum_{i=0}^{p} \left(\sum_{k=0}^{n} (a_k x_i^k - y_i) \right)^2$$

$$A^* = \underset{A}{\text{argument}}\, \varphi(A) \iff \frac{\partial \varphi}{\partial a_k}(A^*) = 0 \quad k = 0, \cdots, n$$

Le minimum est atteint au point où les dérivées s'annulent.

Remarque : on minimise par rapport aux coefficients a_k

$$\frac{\partial \varphi}{\partial a_j} = 2 \sum_{i=0}^{n} \left(\sum_{k=0}^{p} (a_k x_i^{k-1} - y_i) \right) x_i^{j-1} = 0$$

Cas d'un polynôme (suite)

Le minimum est atteint au point où les dérivées s'annulent.

Remarque : on minimise par rapport aux coefficients a_k

$$\frac{\partial \varphi}{\partial a_j} = 2 \sum_{i=0}^{n} \left(\sum_{k=0}^{p} \left(a_k x_i^{k-1} - y_i \right) \right) x_i^{j-1} = 0 \quad (*)$$

$$\Updownarrow$$

$$\sum_{k=0}^{n} a_k \left(\sum_{i=0}^{p} x_i^{k-1} x_i^{j-1} \right) = \sum_{i=0}^{n} y_i x_i^{j-1} \quad (**)$$

$(*)$ et $(**)$ Dérivée d'un polynôme d'ordre 2

Cas d'un polynôme (suite)

Détail du calcul :

$$\mathrm{f}\,(x, a_0, \ldots, a_r) = a_0 + a_1 x + \cdots + a_r x^{\mathrm{p}}.$$

Distance :

$$\varphi(a_0, \ldots, a_r) = \sum_{i=0}^{n} (y_i - (a_0 + a_1 x_i + \cdots + a_{\mathrm{p}} x_i^{\mathrm{p}}))^2.$$

Dérivée pour $k = 0, \ldots, r$:

$$\frac{\varphi(a_0, \ldots, a_{\mathrm{p}})}{\partial a_k} = \sum_{i=0}^{n} \left[2 \cdot (y_i - (a_0 + a_1 x_i + \cdots + a_{\mathrm{p}} x_i^{\mathrm{p}}))\,(-x_i^k)\right] = 0$$

On réécrit l'expression de la dérivée en isolant les a_j (pour $k = 0, \cdots n$)

$$\left(\sum_{i=0}^{p} x_i^k\right) a_0 + \left(\sum_{i=0}^{p} x_i^{k+1}\right) a_1 + \cdots + \left(\sum_{i=0}^{p} x_i^k\right) a_p = \left(\sum_{i=0}^{p} y_i x_i^k\right)$$

On réécrit l’expression de la dérivée en isolant les a_j (pour $k = 0, \cdots n$)

$$\left(\sum_{i=0}^{p} x_i^k\right) a_0 + \left(\sum_{i=0}^{p} x_i^{k+1}\right) a_1 + \cdots + \left(\sum_{i=0}^{p} x_i^k\right) a_n$$

On obtient le système linéaire suivant :

$$\underbrace{\begin{pmatrix} \sum i & \sum x_i & \sum x_i^2 & \cdots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \cdots & \sum x_i^{n+1} \\ & \vdots & \vdots & \cdots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \cdots & \sum x_i^{2n} \end{pmatrix}} \times \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix}} = \begin{pmatrix} \sum y_i \\ \sum y_i x_i \\ \vdots \\ \sum y_i x_i^n \end{pmatrix}$$

Cas d'un polynôme (suite)

On peut aussi le voir sous forme matricielle on peut écrire :

$X \cdot\ A = Y$ avec $A = \{a_0, \cdots a_{n-1}\}$

$$\overbrace{\begin{bmatrix} x_0^0 & x_0^1 & x_0^2 & \cdots & x_0^{n-1} \\ x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^{n-1} \\ & & \vdots & & \\ x_j^0 & x_1^1 & x_j^2 & \cdots & x_j^{n-1} \\ & & \vdots & & \\ x_p^0 & x_p^1 & x_1^2 & \cdots & x_p^{n-1} \end{bmatrix}}^{X} \cdot A = Y = \overbrace{\begin{bmatrix} 1 & x_0^1 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1^1 & x_1^2 & \cdots & x_1^{n-1} \\ & & \vdots & & \\ 1 & x_1^1 & x_j^2 & \cdots & x_j^{n-1} \\ & & \vdots & & \\ 1 & x_p^1 & x_p^2 & \cdots & x_p^{n-1} \end{bmatrix}}^{x_j^0 = 1}$$

On a un système Linéaire surdéterminé voir la suite

Cas particulier : régression Linéaire

C'est une approximation par un polynôme de degré 1

$g(x) = g(x, x_0, x_1) = a_0 + a_1 x$

Le système devient :

$$\begin{pmatrix} n+1 & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}$$

Ou encore

$$\begin{pmatrix} n+1 & \sum x_i \\ \bar{x} & \overline{x^2} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \bar{y} \\ \overline{xy} \end{pmatrix}$$

Avec $\bar{x}, \bar{y}, \overline{x^2}, \overline{xy}$ désigne les moyenne de $x_i, y_i, \ldots.$

On obtient alors :

$$a_0 = \frac{\bar{y}.\overline{x^2} - \bar{x}.\overline{y.x}}{\overline{x^2} - (\bar{x})^2} \qquad a_1 = \frac{\overline{x.y} - \bar{x}.\bar{y}}{\overline{x^2} - (\bar{x})^2}$$

On vérifie aisément que la droite passe par le point moyen :

$$\bar{y} = g(\bar{x}, a_0, a_1)$$

Cas général d'un système linéaire

Soit à estimer un système linéaires : on a un système surdéterminé : *plus d'équations que d'inconnues (n<P)*

$$\begin{cases} m_{11}a_1 & + & a_{12}a_2 & +\ldots\ldots+ & a_{1n}a_n & = & y_1+r_1 \\ m_{21}a_1 & + & m_{22}a_2 & +\ldots\ldots+ & m_{2n}a_n & = & y_2+r_2 \\ : & : & : & : & : & : & : \\ : & : & : & : & : & : & : \\ m_{p1}x_1 & + & m_{p2}a_2 & +\ldots\ldots+ & m_{pn}a_n & = & y_p+r_p \end{cases}$$

m_{ik} : coefficients des équations de mesures ; y_k : mesures,

r_k *: erreurs de mesures*

n données m équations avec m > n

Cas général d'un système linéaire

En notation Matricielle $Ax = Y + R$ avec R vecteur résidu ou erreur

$$\begin{bmatrix} m_{11} & m_{12} & \ldots\ldots & m_{1n} \\ m_{21}x_1 & m_{22} & \ldots\ldots & m_{2n} \\ : & : & : & : \\ : & : & : & : \\ m_{m1} & m_{m2} & \ldots\ldots & m_{pn} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ : \\ : \\ a_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ : \\ : \\ y_p \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ : \\ : \\ r_p \end{bmatrix}$$

La méthode des M.C.
Déterminer a_k tq les sommes des carrés des résidus soit minimales.

Cas d'un système linéaire

En notation Matricielle Ax = Y+R

$$M \cdot A = Y + R \Leftrightarrow M \cdot A - Y = R$$

$$S = R^t R = (M \cdot A - Y)^{t(M \cdot A - Y)} \quad (*)$$

Avec $A = [a_0, \cdots, a_n]^t$

La minimisation de S par rapport à a_k avec k = 1, …n entraine les conditions nécessaires : $\dfrac{\partial S}{\partial a_k} = 0$

Cas d'un système linéaire

En revenant à la notation matricielle ($*$), les conditions s'écrivent (voir explication donnée en cours) :

$$(M^t \cdot M)\tilde{A} = M^t y \;\; (**)$$

avec $\tilde{A}$ les valeurs de A minimisant $(*)$
On arrive à un système cohérent n équation à n inconnues. On résout le système en faisant appel aux méthodes de résolution *d'un système linéaire (triangulation, QR, LU, ….)*

Par exemple la solution des moindres carrés $\tilde{A}$ est donnée :

$$\tilde{A} = (M^t M)^{-1} M^t Y$$

Régression exponentielle

- L'exemple le plus connu est la modélisation de la radioactivité d'un déchet nucléaire ou la modélisation de l'évolution de la population !

$$g(t, a, b) \quad = \quad be^{-a}$$

On a à résoudre un système non-linéaire avec des exponentielles.
Cette résolution peut-être réalisée,
soit en adaptant une méthode de résolution de système non linéaire dans un cadre multidimensionnel, ou en utilisant que

$$\log_b xy = \log_b x + \log_b y$$

$$\log_b \frac{x}{y} = \log_b x - \log_b y$$

$$\log_b x^p = {}^p \log_b x$$

Régression exponentielle

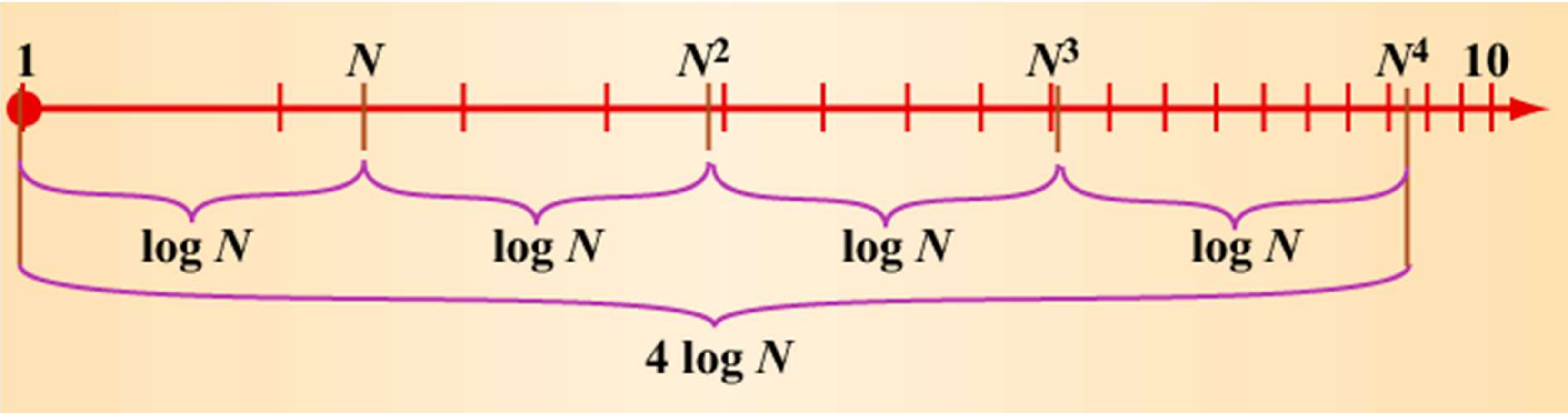
- L'exemple le plus connu est la modélisation de la radioactivité d'un déchet nucléaire ou la modélisation de l'évolution de la population !

$$g(t, a, b) = be^{-at}$$

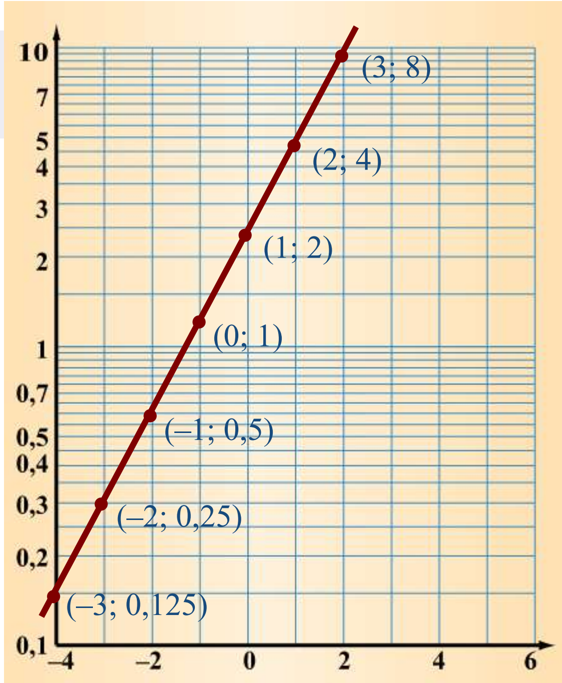
$$\log g(t, a, b) = \log b - at,$$

et donc, trouver une régression exponentielle pour les points (x_i, y_i)
Ou bien une régression linéaire pour les points $(x_i, \log y_i)$. Les constantes sont alors reliées par

$$b = \log a_0 \text{ et } a = -a_1.$$



Lien entre variables



Fonction logarithmique

Une fonction logarithmique est de la forme :

$$y = a \log x + b \text{ (ou } y = a \ln x + b\text{)}$$

On revient à un système linéaire

On voit directement qu'il doit y avoir une relation affine entre *y* et log *x* que l'on peut écrire :

$y = AX + B$, où $X = \log x$ et *A et B sont des coefficients réels.*

On remarque une telle relation sur un repère « semi-log » en représentant la variable x sur l'échelle logarithmique. Si le nuage forme une droite, le modèle est logarithmique.