ASI 3

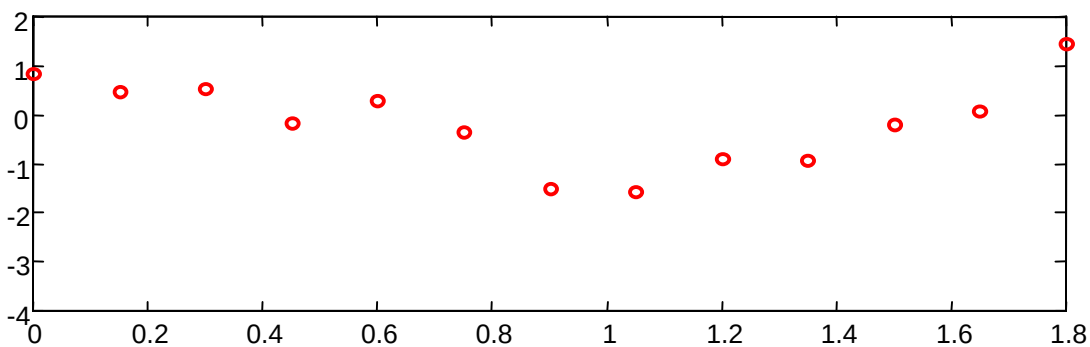
Méthodes numériques pour l'ingénieur

Interpolation

$f(x)$

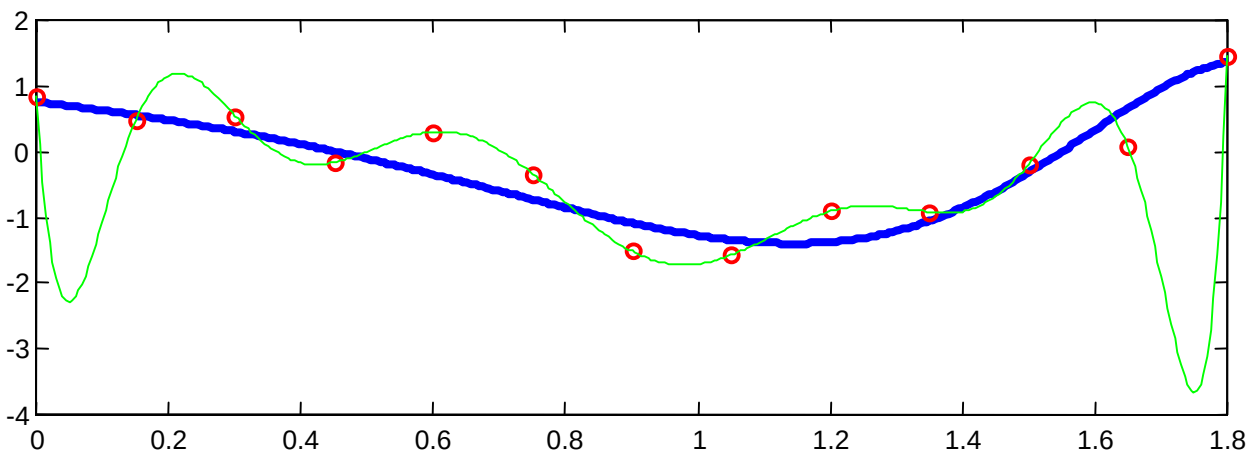
Approximation de fonctions

- Soit une fonction *f* (inconnue explicitement)
  - connue seulement en certains points $x_0, x_1 \ldots x_n$
  - ou évaluable par un calcul coûteux.
- Principe :
  - représenter *f* par une fonction simple, facile à évaluer
- Problème :
  - il existe une infinité de solutions !



Approximation de fonctions

- Il faut se restreindre à une famille de fonctions
  - polynômes,
  - exponentielles,
  - fonctions trigonométriques…



Quelques méthodes d'approximation

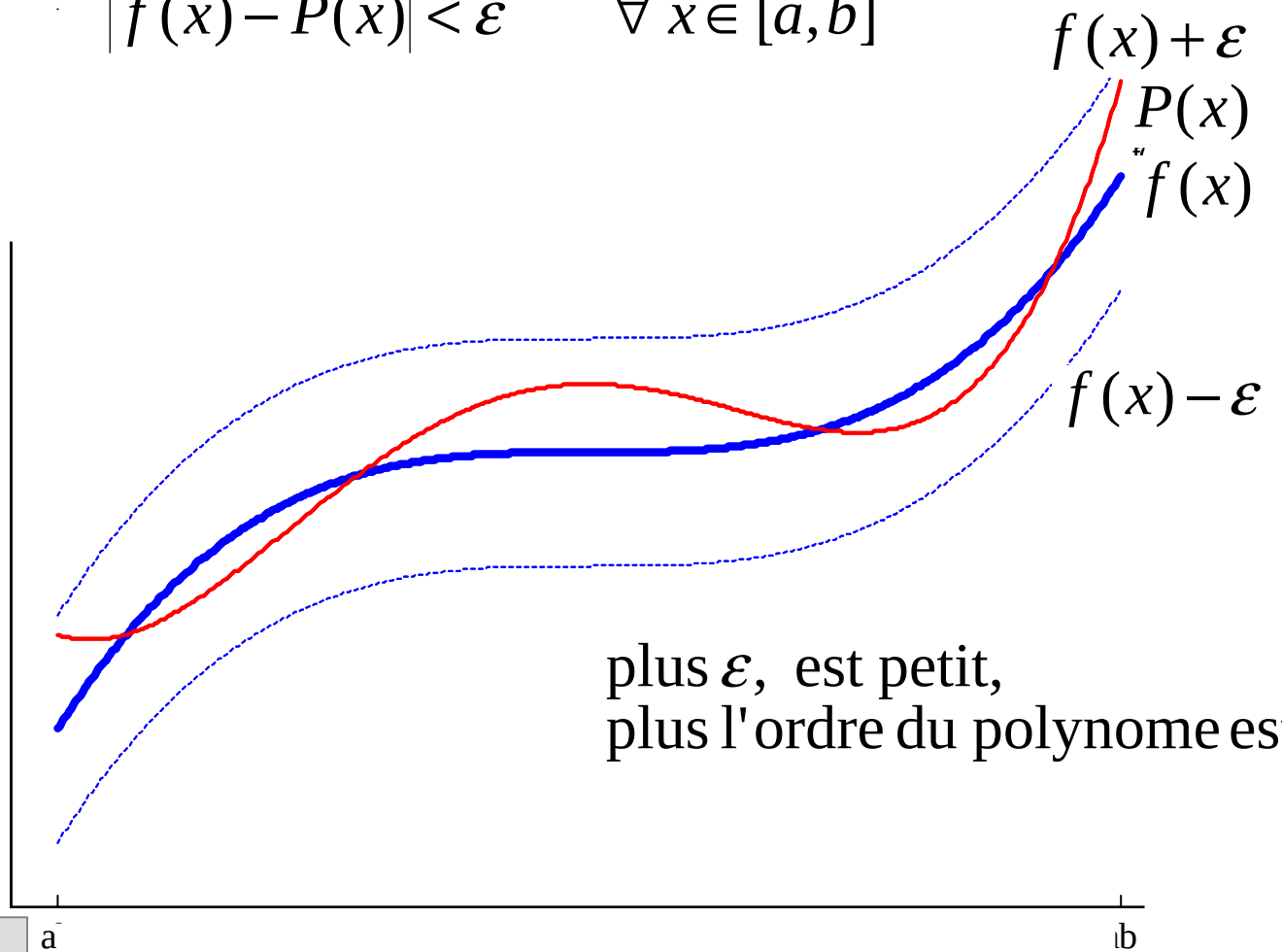
- Interpolation polynomiale
  - polynômes de degré au plus *n*
    - polynômes de Lagrange
    - différences finies de Newton
- Interpolation par splines
  - polynômes par morceaux

- Interpolation d'Hermite
  - informations sur les dérivées de la fonction à approcher
- ...voir le groupe de TT…

Théorème d'approximation de Weierstrass

soit f une fonction définie et continue sur l'intervalle $[a,b]$
Alors, $\forall \varepsilon > 0$, il existe un polynôme $P(x)$, définit sur $[a,b]$ tel que :

$$|f(x) - P(x)| < \varepsilon \qquad \forall\, x \in [a,b]$$



plus ε, est petit,
plus l'ordre du polynome est grand

Interpolation :

$n+1$ points, $n+1$ contraintes, $n+1$ équations, $n+1$ inconnues : ordre n

Interpolation polynomiale

- Le problème : les données, la solution recherchée

  $(x_0, y_0 = f(x_0)),\ldots,(x_i, y_i = f(x_i)),\ldots,(x_i, y_i = f(x_i))$

  $P(x)$ tel que $P(x_i) = f(x_i), i = 0, n$

- mauvaise solution : résoudre le système linéaire

  $P(x) = \sum_{i=0}^{n} a_i x^i \quad \Rightarrow \quad Va = y$ $\qquad V$: matrice de Vandermonde

- la combinaison linéaire de polynômes est un polynôme

  $P(x) = y_0 P_0(x) + \ldots + y_i P_i(x) + \ y_n P_n(x)$

  tel que $P_i(x_i) = 1$ et $P_i(x_j) = 0 \qquad j \neq i$

  ainsi $P(x_i) = y_0 P_0(x_i) + \ldots + y_i P_i(x_i) + \ y_n P_n(x_i)$

  $\qquad\qquad\qquad\qquad\qquad\qquad\quad \downarrow \qquad\qquad\qquad\quad \downarrow \qquad\qquad\quad \downarrow$

  $\qquad\qquad\qquad\qquad\qquad\qquad\quad 0 \qquad\qquad\qquad\quad 1 \qquad\qquad\quad 0$

Interpolation polynomiale : Lagrange

- Théorème
  - Soient $n+1$ points distincts x_i réels et $n+1$ réels y_i, il existe un unique polynôme $p \in P_n$ tel que $p(x_i) = y_i$ pour $i = 0$ à n

démonstration

  - Construction de p : $$p(x) = \sum_{i=0}^{n} y_i L_i(x)$$

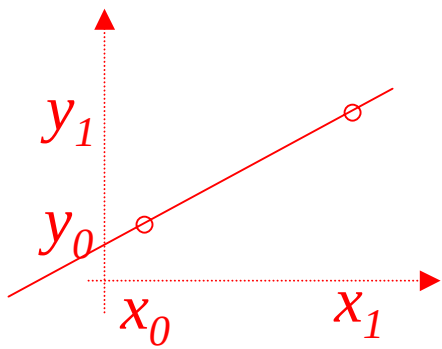
    avec L_i polynôme de Lagrange $$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}$$

    L est un polynôme d'ordre n

  - Propriétés de L_i
    - $L_i(x_i)=1$
    - $L_i(x_j)=0 \quad (j \neq i)$

Lagrange : exemple n°1

- Exemple avec $n=1$
  - on connaît 2 points (x_0,y_0) et (x_1,y_1)
  - on cherche la droite $y=ax+b$ (polynôme de degré 1) qui passe par les 2 points :
    - $y_0 = a\,x_0 + b$ $\qquad a = (y_0 - y_1) / (x_0 - x_1)$
    - $y_1 = a\,x_1 + b$ $\qquad b = (x_0\,y_1 - x_1\,y_0) / (x_0 - x_1)$



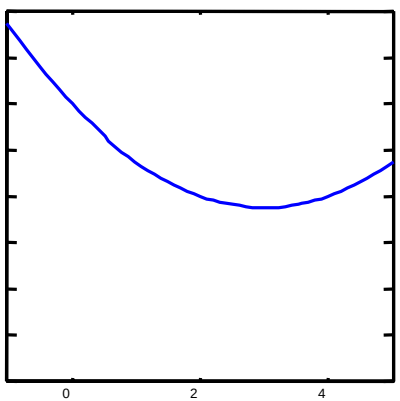
  - $$y = \frac{y_0 - y_1}{x_0 - x_1}\,x + \frac{x_0 y_1 - x_1 y_0}{x_0 - x_1}$$

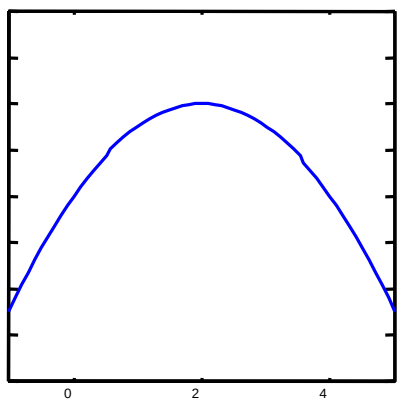
$$y = y_0\,\frac{x - x_1}{x_0 - x_1} - y_1\,\frac{x - x_0}{x_0 - x_1} = y_0\,\underbrace{\frac{x - x_1}{x_0 - x_1}}_{L_0(x)} + y_1\,\underbrace{\frac{x - x_0}{x_1 - x_0}}_{L_1(x)}$$

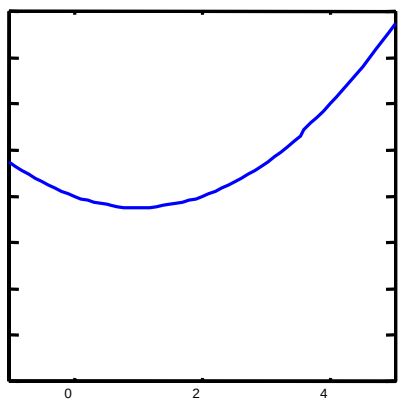
Lagrange : exemple n°2

- Exemple avec $n=2$
  - on connaît 3 points $(0,1)$, $(2,5)$ et $(4,17)$
  - polynômes de Lagrange associés :

$$L_0(x) = \frac{(x-2)(x-4)}{8} \qquad L_1(x) = \frac{x(x-4)}{-4} \qquad L_2(x) = \frac{x(x-2)}{8}$$



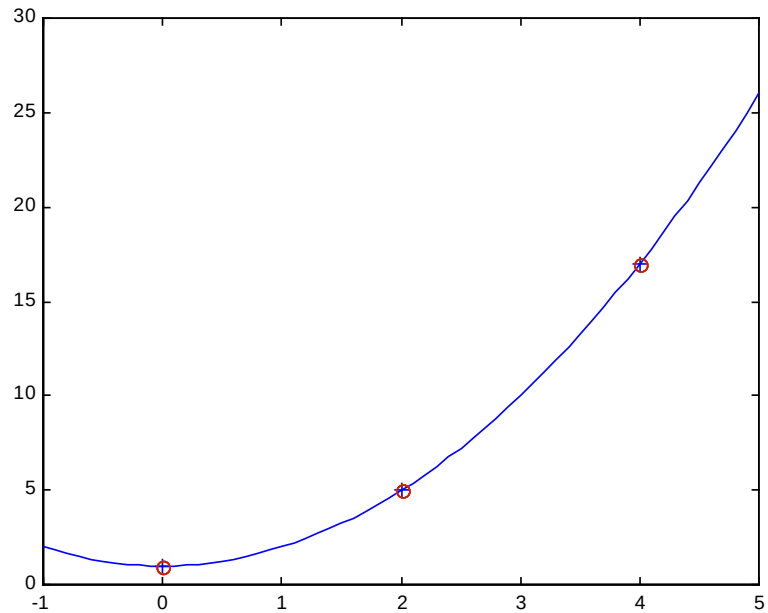




Lagrange : exemple n°2

– calcul du polynôme d'interpolation

- $p(x)=L_0(x) + 5\ L_1(x) + 17\ L_2(x)$



- en simplifiant, on trouve $p(x)=x^2+1$

Lagrange : l'algorithme

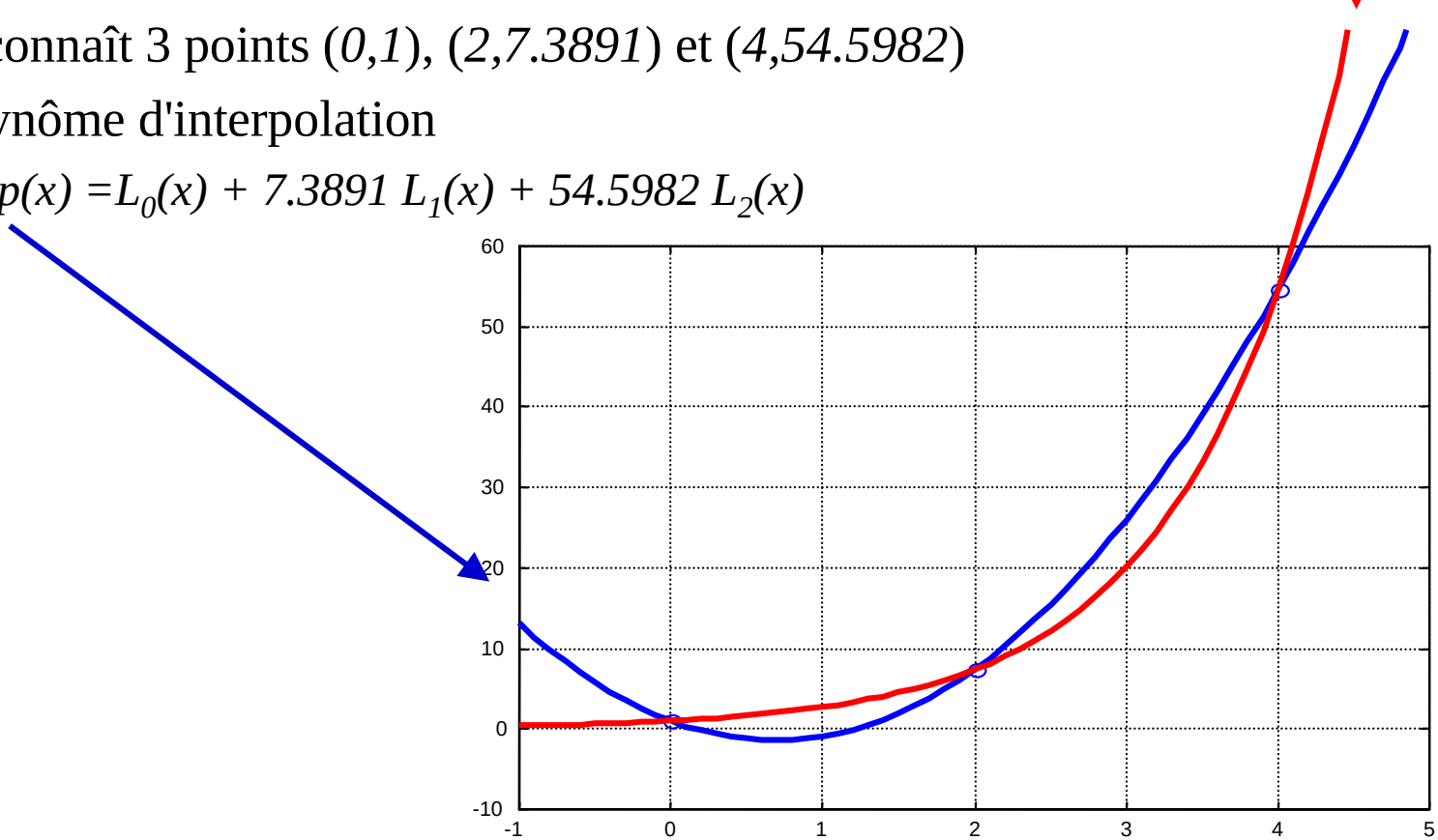
Fonction y = lagrange(x,x_i,y_i)

pour $i = 1$ jusqu'à n
  pour $j = 1$ jusqu'à n, $j \neq i$;
    $l \leftarrow l * \dfrac{x - x_i(j)}{x_i(i) - x_i(j)}$
  fait
  $y \leftarrow y + y_i * l$
fait

Complexité du calcul : n^2

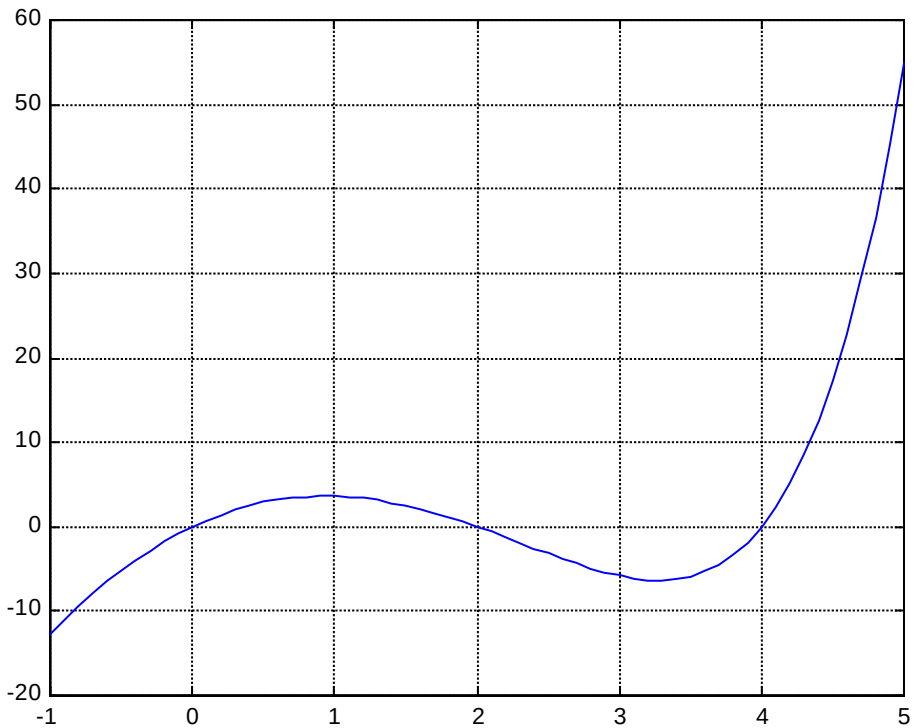
Lagrange : exemple n°3

- Exemple avec $n=2$ (fonction à approcher $y=e^x$)
  - on connaît 3 points $(0,1)$, $(2,7.3891)$ et $(4,54.5982)$
  - Polynôme d'interpolation
    - $p(x) = L_0(x) + 7.3891\ L_1(x) + 54.5982\ L_2(x)$



Lagrange : exemple n°3

– Erreur d'interpolation $e(x) = f(x) - p(x)$



Lagrange : erreur d'interpolation

- Théorème :
  - si f est $n+1$ dérivable sur $[a,b]$, $\forall\ x \in\ [a,b]$, notons :
    - I le plus petit intervalle fermé contenant x et les x_i
    - $\phi(x)=(x-x_0)(x-x_1)\ldots(x-x_n)$
  - alors, il existe $\xi \in\ I$ tel que $e(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\phi(x)$
  - NB : ξ dépend de x

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

Lagrange : choix de n

- Supposons que l'on possède un nb élevé de points pour approcher f … faut-il tous les utiliser ?
  - (calculs lourds)

- Méthode de Neville :
  - on augmente progressivement n
  - on calcule des L_i de manière récursive
  - on arrête dès que l'erreur est inférieure à un seuil

    (d'ou l'utilité du calcul de l'erreur)

La méthode de Neuville

- Définition

  $P_{m_1,m_2,\ldots,m_k}(x)$ polynôme de Lagrange calculé sur les k points $(x_{m_1}, y_{m_1}), (x_{m_2}, y_{m_2}), \ldots, (x_{m_k}, y_{m_k})$

- Théorème

$$P(x) = \frac{(x - x_j)P_{0,1,\ldots,j-1,j+1,\ldots,n}(x) - (x - x_i)P_{0,1,\ldots,i-1,i+1,\ldots,n}(x)}{x_i - x_j}$$

- Démonstration

$$P(x_i) = f(x_i); P(x_j) = f(x_j) \quad \text{et} \quad P(x_k) = f(x_k)$$

- Application systématique $\quad Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}$

$x_0 \quad P_0 = Q_{0,0}$

$x_1 \quad P_1 = Q_{1,0} \quad P_{0,1} = Q_{1,1}$

$x_2 \quad P_2 = Q_{2,0} \quad P_{1,2} = Q_{2,1} \quad P_{0,1,2} = Q_{2,2}$

$x_3 \quad P_3 = Q_{3,0} \quad P_{2,3} = Q_{3,1} \quad P_{1,2,3} = Q_{3,2} \quad P_{0,1,2,4} = Q_{3,3}$

L'algorithme de Neuville

Fonction y = Neuville(x,x_i,y_i)

pour $i = 1$ jusqu'à n
    $Q(i,0) \leftarrow y_i(i)$
fait
pour $i = 1$ jusqu'à n
    pour $j = 1$ jusqu'à i
        $$Q(i,j) \leftarrow \frac{(x - x_i(i-j))Q(i,j-1) - (x - x_i(i))Q(i-1,j-1)}{x_i(i) - x_i(i-j)}$$
    fait
    $y \leftarrow Q(n,n)$
fait

Complexité du calcul : n^2

Interpolation polynomiale : Newton

- Polynômes de Newton :
  - base = $\{1, (x-x_0), (x-x_0)(x-x_1), \ldots, (x-x_0)(x-x_1)\ldots(x-x_{n-1})\}$
  - on peut ré-écrire $p(x)$:
    $p(x)=a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)+\ldots+ a_n(x-x_0)(x-x_1)\ldots(x-x_{n-1})$
  - calcul des a_k : méthode des différences divisées

Newton : différences divisées

- Définition :
  - Soit une fonction *f* dont on connaît les valeurs en des points distincts *a*, *b*, *c*, …
  - On appelle **différence divisée** d'ordre *0*, *1*, *2*,...,*n* les expressions définies par récurrence sur l'ordre *k* :
  - *k=0* $f[a] = f(a)$
  - *k=1* $f[a,b] = (f[b] - f[a]) / (b - a)$
  - *k=2* $f[a,b,c] = (f[a,c] - f[a,b]) / (c - b)$
  - … $f[X,a,b] = (f[X,b] - f[X,a]) / (b - a)$ $a \notin X$, $b \notin X$, $a \neq b$

Newton : différences divisées

- Théorèmes :
  - détermination des coefficients de $p(x)$ dans la base de Newton :

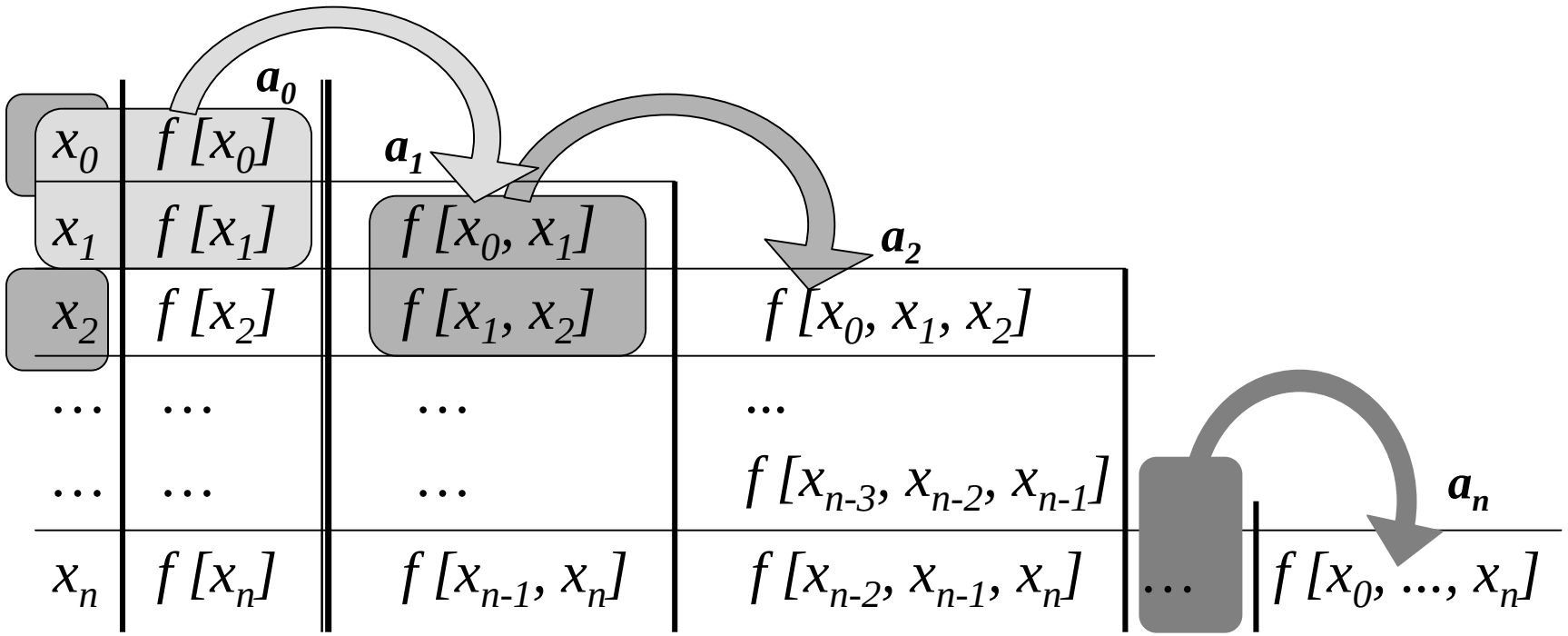
$$f[x_0, x_1, \ldots, x_k] = a_k \quad \text{avec} \quad k = 0 \ldots n$$

  - erreur d'interpolation :

$$e(x) = f[x_0, x_1, \ldots, x_n, x]\ \phi(x)$$

Newton : différences divisées

- Calcul pratique des coefficients :



Newton : exemple

- (ex. n°2) : $n=2$ $(0,1)$, $(2,5)$ et $(4,17)$

		a_0		
0	$f[x_0]=1$			
2	$f[x_1]=5$	$f[x_0, x_1]$ $=(1-5)/(0-2)=2$	a_1	
4	$f[x_2]=17$	$f[x_1, x_2]$ $=(5-17)/(2-4)=6$	$f[x_0, x_1, x_2]$ $= (2-6)/(0-4)=1$	a_2

$p(x)=1 + 2x + x(x-2)$ (et on retombe sur $p(x) = 1 + x^2$)

Newton : l'algorithme

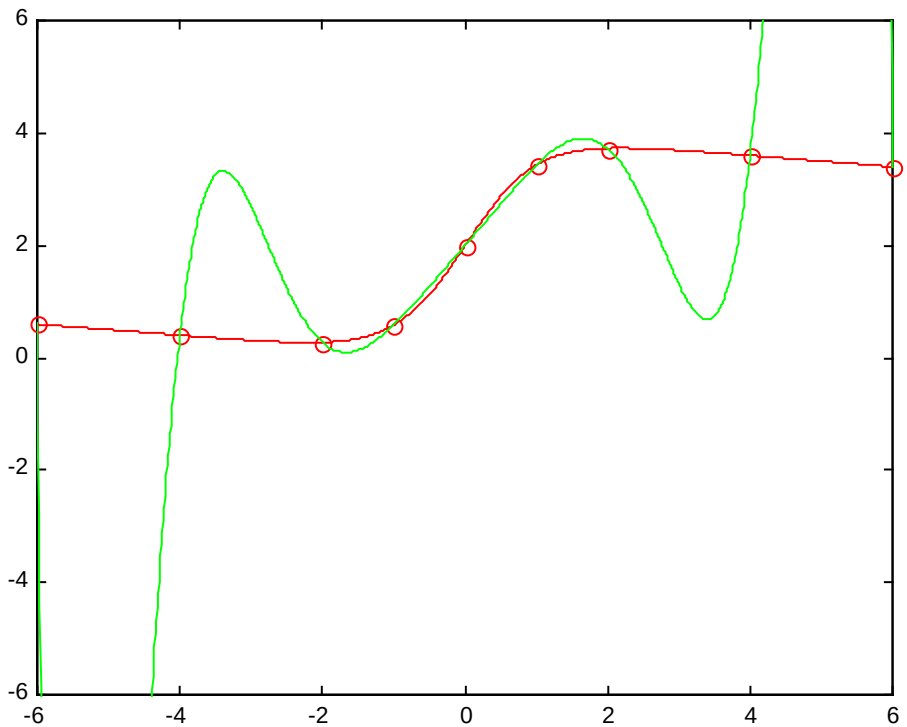
Fonction a = Newton(x_i,y_i)

pour $i = 1$ jusqu'à n
    $F(i,0) \leftarrow y_i(i)$
fait
pour $i = 1$ jusqu'à n
    pour $j = 1$ jusqu'à i
        $F(i,j) \leftarrow \dfrac{F(i,j-1) - F(i-1,j-1)}{x_i(i) - x_i(i-j)}$
    fait
fait
pour $i = 1$ jusqu'à n
    $a(\mathrm{i}) \leftarrow F(n,i)$
fait

Complexité du calcul : n^2

A bas les polynômes

- Ex : $y=2(1+\tanh(x)) - x/10$ avec 9 points
  - entre les points, le polynôme fait ce qu'il veut…
    et plus son degré est élevé plus il est apte à osciller !



Interpolation par splines cubiques

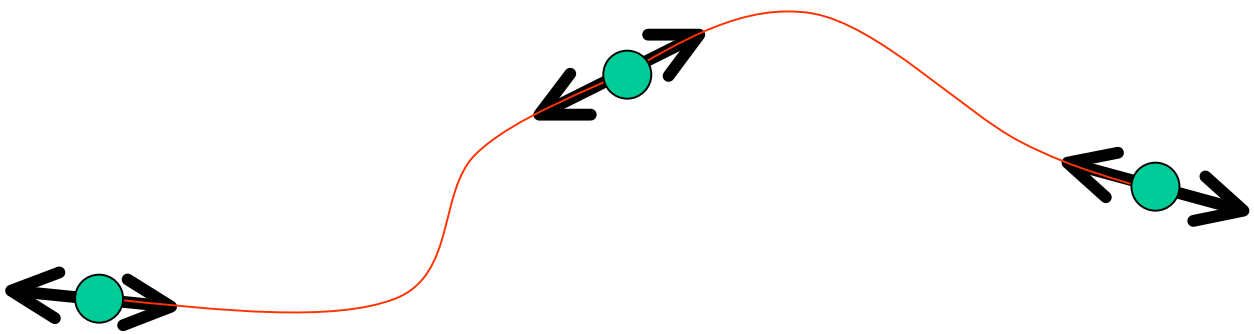
- Principe :
  - on approche la courbe par morceaux (localement)
  - on prend des polynômes de degré faible (3) pour éviter les oscillations

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 continûment dérivable),
    - g coïncide sur chaque intervalle $[x_i; x_{i+1}]$ avec un polynôme de degré inférieur ou égal à 3,
    - $g(x_i) = y_i$ pour $i = 0 \ldots n$
- Remarque :
  - Il faut des conditions supplémentaires pour définir la spline d'interpolation de façon unique
  - Ex. de conditions supplémentaires :
    - $g''(a) = g''(b) = 0$ spline naturelle.

Splines : illustration

$$P_2(x)=a_2\,(x-x_2)^3+b_2\,(x-x_2)^2+c_2\,(x-x_2)+d_2$$



$$P_1(x)=\alpha_1 x^3+\beta_1 x^2+\chi_1 x+\delta_1$$

$$=a_1\,(x-x_1)^3+b_1\,(x-x_1)^2+c_1\,(x-x_1)+d_1$$

Splines cubiques : détermination

- Détermination de la spline d'interpolation
  - g coïncide sur chaque intervalle $[x_i; x_{i+1}]$ avec un polynôme de degré inférieur ou égal à 3

  ➔ g'' est de degré 1 et est déterminé par 2 valeurs:
    - $m_i = g''(x_i)$ et $m_{i+1} = g''(x_{i+1})$ (**moment** au noeud n°i)

  - Notations :
    - $h_i = x_{i+1} - x_i$ pour $i = 0 \ldots n-1$
    - $\delta_i = [x_i; x_{i+1}]$
    - $g_i(x)$ le polynôme de degré 3 qui coïncide avec g sur l'intervalle δ_i

Splines cubiques : détermination

- $g''_i(x)$ est linéaire :
  - $\forall\, x \in \delta_i$ $\quad g_i''(x) = m_{i+1}\dfrac{x - x_i}{h_i} + m_i\dfrac{x_{i+1} - x}{h_i}$
  - on intègre (a_i constante) $\quad g_i'(x) = m_{i+1}\dfrac{(x - x_i)^2}{2h_i} - m_i\dfrac{(x_{i+1} - x)^2}{2h_i} + a_i$
  - on continue (b_i constante) $\quad g_i(x) = m_{i+1}\dfrac{(x - x_i)^3}{6h_i} + m_i\dfrac{(x_{i+1} - x)^3}{6h_i} + a_i(x - x_i) + b_i$

- $g_i(x_i) = y_i \Rightarrow y_i = \dfrac{m_i h_i^2}{6} + b_i$ (1)
- $g_i(x_{i+1}) = y_{i+1} \Rightarrow y_{i+1} = \dfrac{m_{i+1} h_i^2}{6} + a_i h_i + b_i$ (2)

Splines cubiques : détermination

– $g'(x)$ est continue : $g_i'(x_i) = -m_i \frac{h_i}{2} + a_i = m_i \frac{h_{i-1}}{2} + a_{i-1} = g_{i-1}'(x_i)$ 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

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

– Rappel : on cherche les m_i (n+1 inconnues)

➔ on a seulement n-1 équations grâce à 4

➔ il faut rajouter 2 conditions, par exemple

➔ $m_0 = m_n = 0$ (spline naturelle)

Splines cubiques : détermination

4

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

– Ex de résolution avec $h_i = x_{i+1} - x_i$ 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 :

$$Tm = f$$

$$\begin{pmatrix} 4 & 1 & & & 0 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ \\ \dots \\ \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} f_1 \\ \\ \dots \\ \\ f_{n-1} \end{pmatrix}$$

- T inversible (diagonale strictement dominante)

Splines cubiques : l'algorithme

pour $i = 2; n-1$

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

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

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

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

fait

$T \leftarrow T(2{:}n\text{-}1, 2{:}n\text{-}1)$

$m \leftarrow T \setminus f$

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

pour $i = 1; n-1$

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

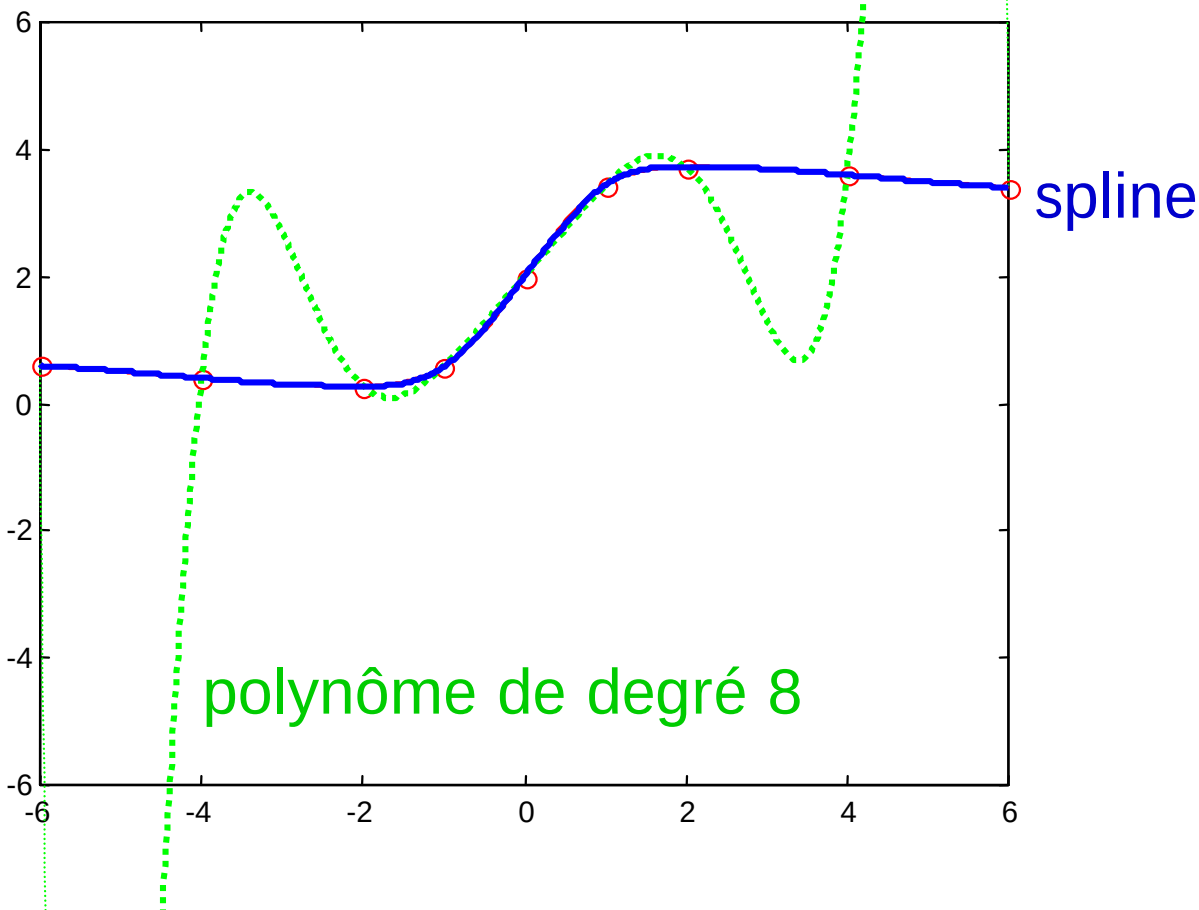
$$b(i) \leftarrow y(i) - \frac{m_i h_i}{6}$$

fait

Complexité du calcul : complexité du solveur

Splines cubiques : exemple

- Ex : $y=2(1+\tanh(x)) - x/10$ avec 9 points



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
  - spline cubique d'approximation (on régularise)
  - b spline
  - spline généralisée : splines gausiènnes (multidimensionelle)
- approximation - apprentissage