

M5 E2 : Analyse Numérique (Cours magistral - Partie 2)

Dr. Yassine Sabbar

High school of technology
Ibn Zohr University
Agadir, Morocco

Email : yassine.sabbar@edu.uiz.ac.ma

2 mai 2019

Chapitre 5 :

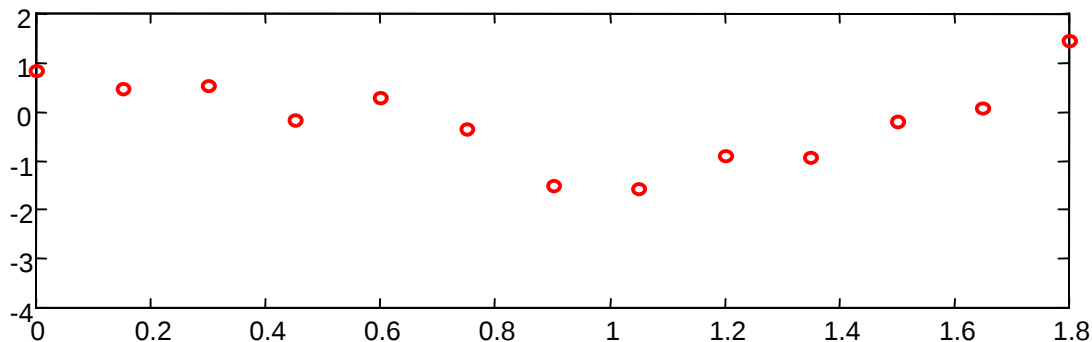
Interpolation polynomiale

Problème de l'interpolation

- 1 En mathématiques, en analyse numérique, l'interpolation polynomiale est une technique d'interpolation d'un ensemble de données ou d'une fonction par un polynôme.
- 2 Étant donné un ensemble de points (obtenu, par exemple, à la suite d'une expérience), on cherche un polynôme qui passe par tous ces points.
- 3 Approcher une fonction f consiste à la remplacer par une autre fonction g dont la forme est plus simple et dont on peut se servir à la place de f .
- 4 L'approximation a pour but de remplacer f par une fonction plus simple en vue d'un calcul numérique d'intégrale ou de dérivée. De plus, le but est d'avoir une représentation synthétique de données expérimentales (dont le nombre peut être très élevé).

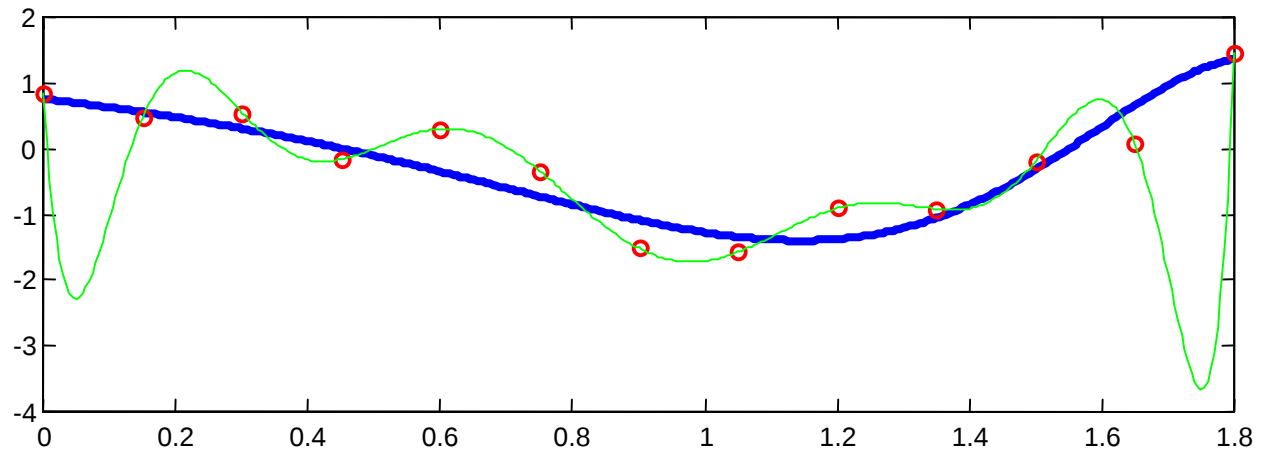
Approximation de fonctions

- Soit une fonction f (inconnue explicitement)
 - connue seulement en certains points $x_0, x_1 \dots x_n$
 - 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...



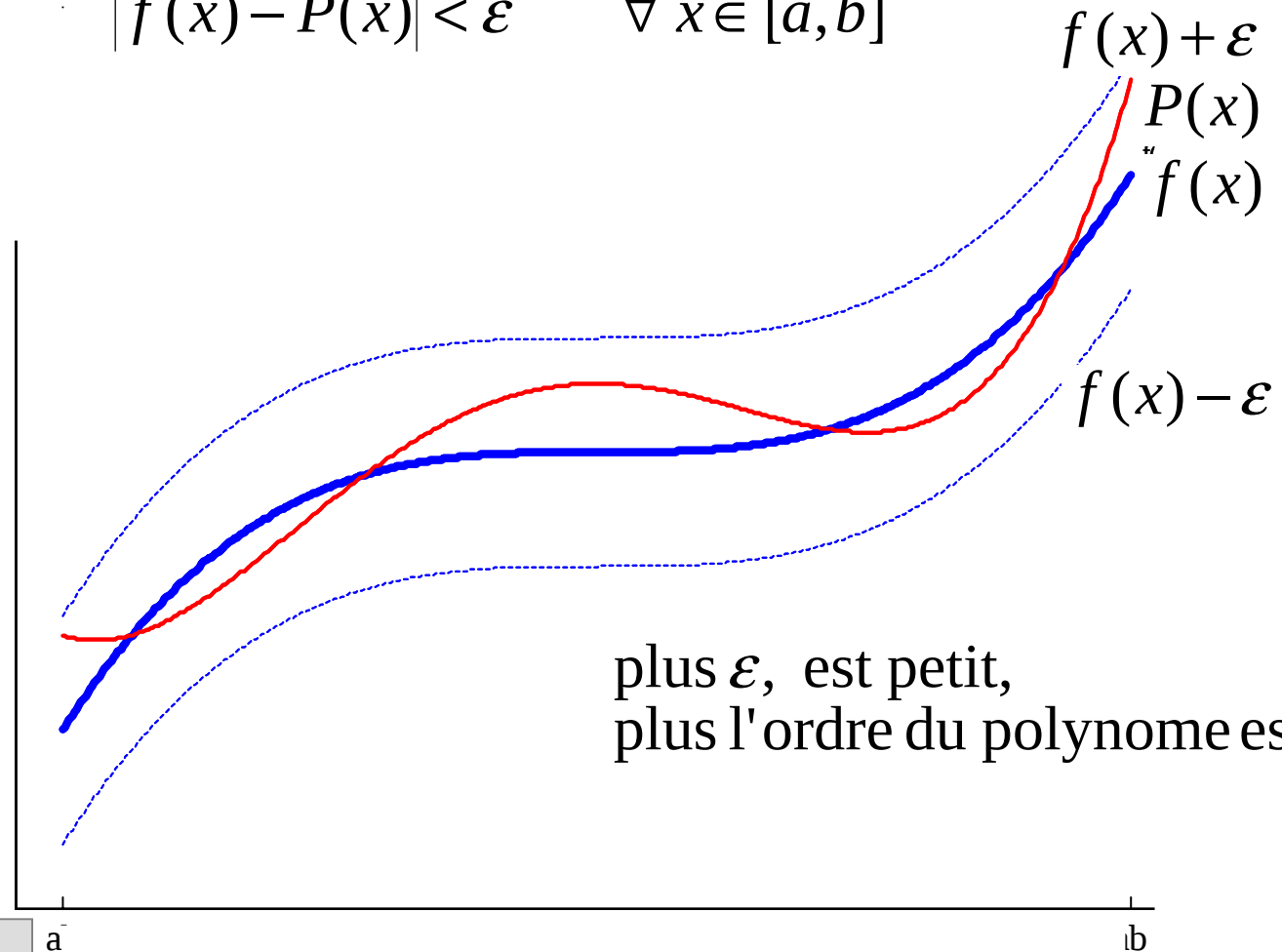
Quelques méthodes d'approximation

- 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 $P(x)$, défini sur $[a, b]$ tel que :

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



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

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

$$P(x) \text{ tel que } P(x_i) = f(x_i), i = 0, n$$

- mauvaise solution : résoudre le système linéaire

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

- la combinaison linéaire de polynômes est un polynôme

$$P(x) = y_0P_0(x) + \dots + y_iP_i(x) + \dots + y_nP_n(x)$$

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

$$\text{ainsi } P(x_i) = y_0 \underset{\substack{\downarrow \\ 0}}{P_0}(x_i) + \dots + y_i \underset{\substack{\downarrow \\ 1}}{P_i}(x_i) + y_n \underset{\substack{\downarrow \\ 0}}{P_n}(x_i)$$

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

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

- Propriétés de L_i

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

L est un polynôme d'ordre n

Lagrange : exemple n°1

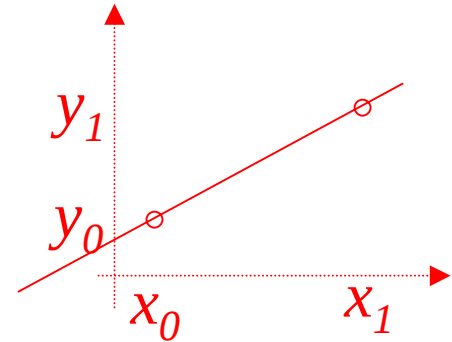
- Exemple avec $n=1$

- 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 :

- $y_0 = a x_0 + b$ $a = (y_0 - y_1) / (x_0 - x_1)$
 - $y_1 = a x_1 + b$ $b = (x_0 y_1 - x_1 y_0) / (x_0 - x_1)$

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

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



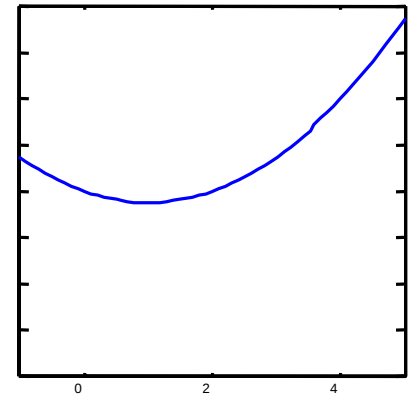
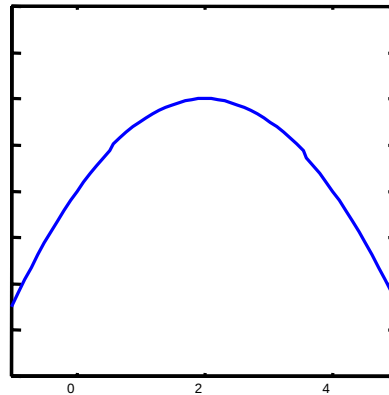
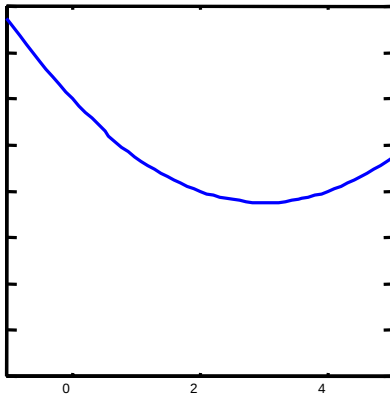
Lagrange : exemple n°2

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

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

$$L_1(x) = \frac{x(x-4)}{-4}$$

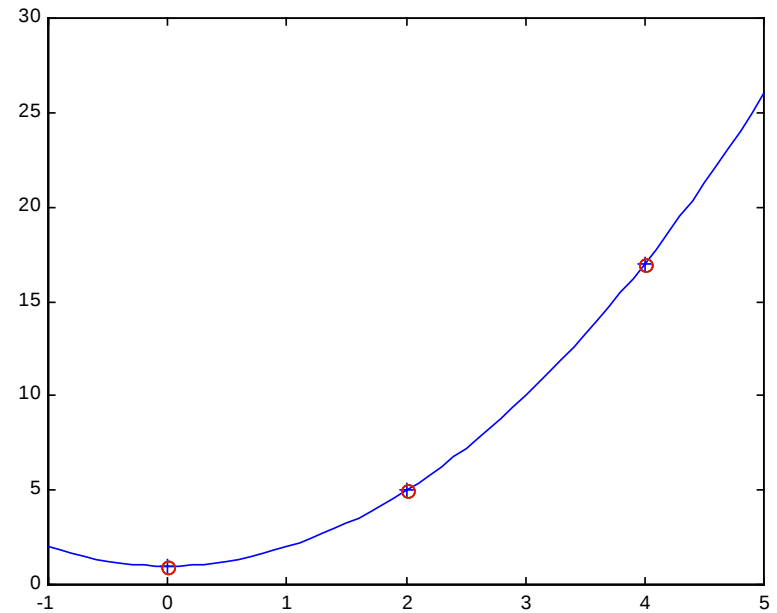
$$L_2(x) = \frac{x(x-2)}{8}$$



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

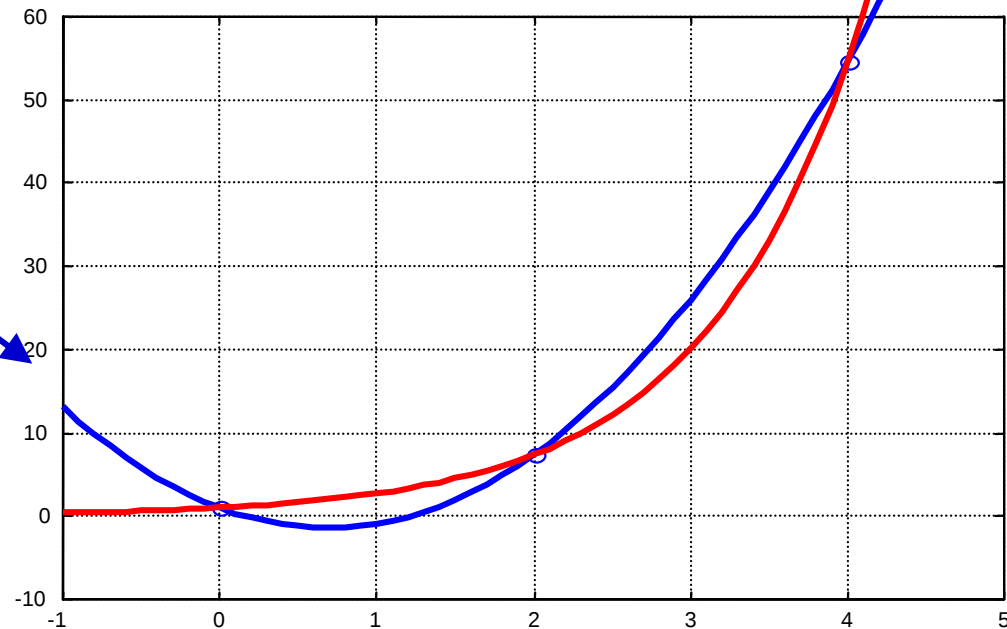
Fonction $y = \text{lagrange}(x, x_i, y_i)$

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

Complexité du calcul : n^2

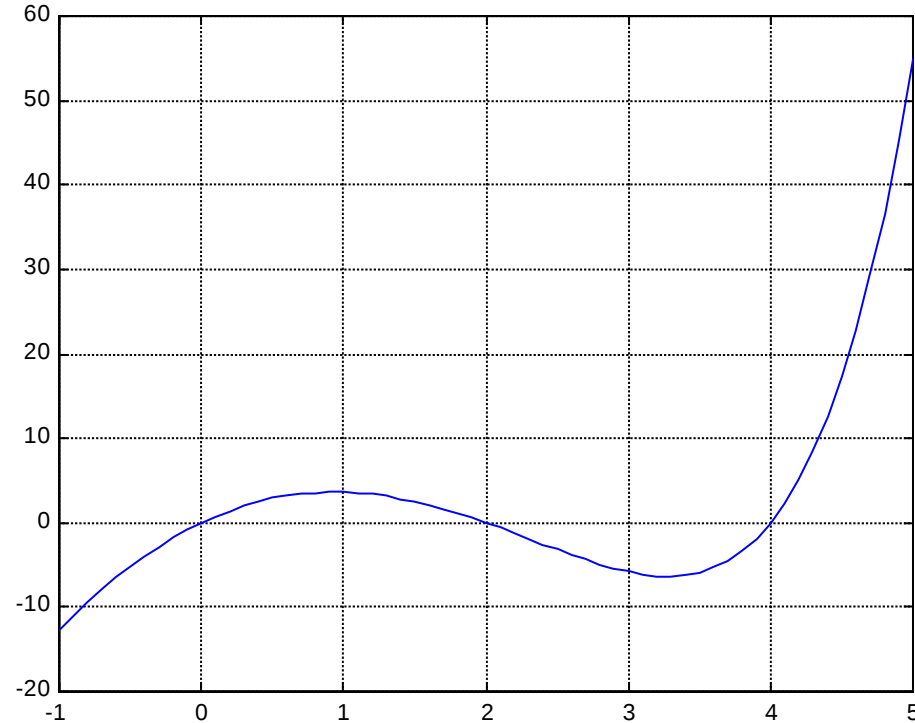
Lagrange : exemple n°3

- 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
 - $\phi(x)=(x-x_0)(x-x_1)\dots(x-x_n)$
 - 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'où l'utilité du calcul de l'erreur)

La méthode de Neville

- Définition

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

- Théorème

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

- Démonstration

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

- Application systématique

$$Q_{i,j} = P_{i-j,i-j+1,\dots,i-1,i}$$

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

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

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

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

L'algorithme de Neville

Fonction $y = \text{Neville}(x, x_i, y_i)$

```
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 = $\{1, (x-x_0), (x-x_0)(x-x_1), \dots, (x-x_0)(x-x_1)\dots(x-x_{n-1})\}$
 - on peut ré-écrire $p(x)$:
$$p(x)=a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1)+\dots+ a_n(x-x_0)(x-x_1)\dots(x-x_{n-1})$$
 - 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 $a, b, c, \dots$
 - On appelle **différence divisée** d'ordre $0, 1, 2, \dots, n$ 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 :

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

- erreur d'interpolation :

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

Newton : différences divisées

- Calcul pratique des coefficients :

x_0	$f[x_0]$				
x_1	$f[x_1]$	$f[x_0, x_1]$			
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
...	...	...	...		
...	...	...	$f[x_{n-3}, x_{n-2}, x_{n-1}]$		
x_n	$f[x_n]$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	...	$f[x_0, \dots, x_n]$

Newton : exemple

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

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

$$p(x)=1 + 2x + x(x-2)$$

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

Newton : l'algorithme

Fonction $a = \text{Newton}(x_i, y_i)$

```
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  $i$ 
        
$$F(i, j) \leftarrow \frac{F(i, j-1) - F(i-1, j-1)}{x_i(i) - x_i(i-j)}$$

    fait
fait
pour  $i = 1$  jusqu'à  $n$ 
     $a(i) \leftarrow F(n, i)$ 
fait
```

Complexité du calcul : n^2