Algorithmique 4

Interpolation et approximation

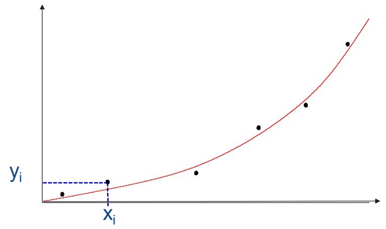
Problème

- Données :
  - une fonction f
  - un ensemble de points connus (x_i, $f(x_i)$)
- But : déterminer un ”modèle” mathématique pour f
  - réduire f en une formule utilisable (exemple : régression)
  - bonnes propriétés : dérivabilité, etc.
- Dans quels cas ?
  - définir un modèle mathématique à partir d’un nombre discret de mesures
  - analyser un phénomène étudié de manière empirique
  - remplacer une équation de courbe “compliquée” par une fonction polynomiale par exemple.

Approximation

les (x_i, y_i) sont des mesures inexactes

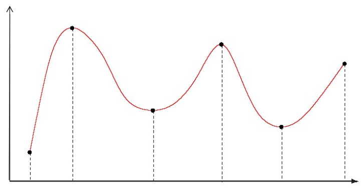
Objet de l’étude : déterminer la courbe s’approchant au mieux des points (x_i, $f(x_i)$)



Interpolation

les x_i, sont des mesures exactes

➡ donc la courbe passe par tous les $(x_i, f(x_i))$



On se donne

- une fonction f : R → R inconnue et continue sur un intervalle fermé [a, b].
- un ensemble de points connus $(x_i, f(x_i))$, pour **i∈[0, n].**

$\{x_0, x_1, . . . , x_n\}$ est le support de l'interpolation.

On cherche

une fonction j : R → R telle que

$j(x_i) = f(x_i)$, $i \in [0, n]$.

- En pratique, φ est une somme de fonctions

$$\varphi(x) = \sum_{j=0}^{n} a_j \varphi_j(x)$$

vérifiant

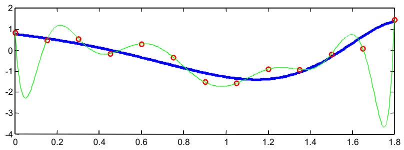
$$f(x_i) = \varphi(x_i) \text{ avec } (x_i) \in R^n \quad (1)$$

φ_j: fonction de base qui doit se prêter aux traitements numériques courants.

Problème : déterminer les a_j pour vérifier (1) et l'unicité de la solution donc de a_j

Approximation de fonctions

- En général, on se restreint à une famille de fonctions connues
  - polynômes,
  - exponentielles, logarithme
  - 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 (ce chapitre ne sera pas traité)
  - informations sur les dérivées de la fonction à approcher

Théorème d'approximation de Weierstrass

soit f fct continue sur $[a, b]$

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 polynôme est grand

Interpolation :

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

Interpolation polynomiale

- Le problème : Solution recherchée
- Données --> $(x_0, y_0 = f(x_0)), \cdots, (x_i, y_i = f(x_i)), \cdots, (x_n, y_n = f(x_n))$
- Solution --> $P(x) : P(x_i) = y_i \;\; i = 0, \cdots, n$

- mauvaise solution : résoudre le système linéaire

$$P(x) = \sum_{i=0}^{i=n} a_i x^i$$

- la combinaison linéaire de polynômes est un polynôme

→ Idée de Lagrange

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

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

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

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

$$0 \qquad 1 \qquad 0$$

Méthode de Lagrange pour l'interpolation polynômiale

→ Idée changer de base pour les polynômes pour faciliter le calcul

$$\boxed{L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n+1} \frac{(x - x_j)}{(x_i - x_j)}}$$

L_i est un polynôme d'ordre n

- **Théorème**
  - Soient *n+1* points distincts de coordonnée $(x_i \,,\, y_i)$

avec $x_i \,,\, y_i$ réels

il existe un unique polynôme $P \in P_n$ tel que $P(x_i) = y_i \; i = 0, \cdots, n$

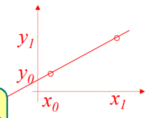
- **Construction de *p* :**
  avec L_i polynôme de Lagrange

Idée de démonstration

- **Propriétés de L_i**
  - $l_i(x_i) = 1$
  - $l_i(x_j) = 0 \quad j \neq i$
  - $P(x) = \sum_{i=0}^{n} y_i L_i(x)$

Lagrange : degré 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 :
    - $y_0 = a\,x_0 + b$ $\qquad a = (y_0 - y_1) / (x_0 - x_1)$
    - $y_1 = a\,x_1 + b$ $\qquad b = (x_0\,y_1 - x_1\,y_0) / (x_0 - x_1)$



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

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

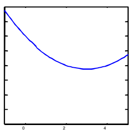
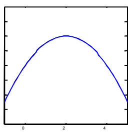
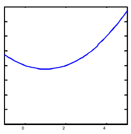
75

Lagrange : degré 2

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

    → Espace vectoriel : avec $\{L_i\}$ base de l'interpolation

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

76

Lagrange : degré 2

- calcul du polynôme d'interpolation

points : (0,1), (2,5) et (4,17)

- $p(x)=L_0(x) + 5\,L_1(x) + 17\,L_2(x)$
- en réduisant l'expression , on trouve $p(x)=x^2+1$
- *On trouve la même expression en résolvant le système :*

  $y_i = a_2 x_i^2 + a_1\,x_i + a_0\,x_i$ *avec* $(x_i, y_i) = \overbrace{\{(0\,,1);(2\,\,,\,\,5);(4\,,17)\}}$

77

Lagrange : Algorithme

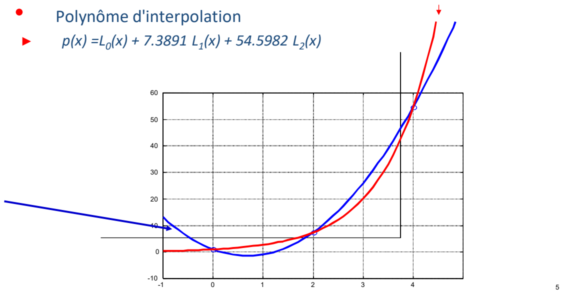
Fonction y = Lagrange (x, x_i, y_i)

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

Complexité du calcul : ?? À estimer

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 : erreur d'interpolation

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

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

$$e(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\varphi(x)$$

- NB : ξ dépend de x

Utilité = on contrôle l'erreur d'interpolation donc la qualité de l'interpolation (voir exercice proposé en TD)

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

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

- Piste de démonstration

$P(x_i) = f(x_i); P(x_j) = f(x_j)$ et $P(x_k) = f(x_k)$

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

La méthode de Neville (suite)

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

- Application systématique

$$\begin{array}{lllll}
x_0 & P_0 = Q_{0,0} & & & \\
x_1 & P_1 = Q_{1,0} & P_{0,1} = Q_{1,1} & & \\
x_2 & P_2 = Q_{2,0} & P_{1,2} = Q_{2,1} & P_{0,1,2} = Q_{2,2} & \\
x_3 & P_3 = Q_{3,0} & P_{2,3} = Q_{3,1} & P_{1,2,3} = Q_{3,2} & P_{0,1,2,4} = Q_{3,3}
\end{array}$$

L'algorithme de Neville

Fonction y = Neville(x,x_i,y_i)

pour $i = 1$ à n
  $Q(i, 0) \leftarrow y_i(i)$
fin pour
pour $i = 1$ à n
  pour $j = 1$ à i
    $Q(i,j)$
$\leftarrow \dfrac{(x - x_i(i-j))Q(i,j-1) - (x - x_i(i))Q(i-1,j-1)}{x_i(i) - x_i(i-j)}$
  fin pour
  $y \leftarrow Q(n, n)$
fin pour

Complexité du calcul : n^2

84

Méthode de Newton

- Polynômes de Newton :

  Base $=\{(1, (x - x_0), (x - x_0)(x - x_1), \cdots, (x - x_0)(x - x_1)\cdots(x - x_{n-1})\}$

  - on peut écrire $P(x)$:

    $$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

  - Calcul des a_k : méthode des différences divisées

85

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 à l'ordre k :

  - $k = 0 \quad f\,[a] = f\,(a)$
  - $k = 1 \quad f\,[a, b] = (f\,[b] - f\,[a]) / (b - a)$
  - $k = 2 \quad f\,[a, b, c] = (f\,[a, c] - f\,[a, b]) / (c - b)$

    …
  - $f\,[X, a, b] = (f\,[X, b] - f\,[X, a]) / (b - a)$

    $a \notin X, \quad b \notin X, \quad a \neq b$

86

Newton : différences divisées

- Détermination des coefficients de *p(x)* dans la base de Newton :

Théorèmes

calcul des coefficients de newton

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

Calcul de l'erreur d'interpolation : soit $\tilde{x} \in [x_0, x_n]$

$\exists(\xi) \in [x_0, x_n]$

$$e(\tilde{x}) = (\tilde{x} - x_0)(\tilde{x} - x_1) \cdots (\tilde{x} - x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}$$

87

Newton : différences divisées

- Calcul pratique des coefficients :

		a_1	a_2	a_n
x_0	$f[x_0]$ a_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	$f[x_n]$	$f[x_{n-1}, x_n]$ ….	$f[x_{n-2}, x_{n-1}, x_n]$	$f[x_0, \dots, x_n]$

88

Newton : exemple

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

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

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

89

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

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

  fait

fait

pour $i = 1$ jusqu'à n

  $a(\mathrm{i}) \leftarrow F(n,i)$

fait

Complexité du calcul : n^2