Chapitre 4

Polynômes : évaluation et interpolation

4.1 Evaluation

4.1.1 Evaluation de Horner

Entrée: $a_0, \ldots, a_n$, coefficients du polynôme P ; t, réel.
Sortie: $val = P(t)$
$val \leftarrow a_n$;
pour i *de* $n-1$ *à* 0 **faire**
| $val \leftarrow a_i + t \times val$;
fpour
retourner *val;*

Algorithme 1: Evaluation de Horner

L'algorithme est de complexité linéaire en n.

Exercice 4.1.1 *Programmez l'évaluation d'Horner en Python et en xcas, verifiez avec la fonction xcas déja programmée.*

4.1.2 Evaluation de Newton-Horner

Théorème 4.1.2 *Soit $c_0, \ldots, c_{n-1}$ des réels (distincts ou non).*
Alors les polynômes $P_0 = 1$, $P_i = (X - c_0) \ldots (X - c_{i-1})$ $(1 \leq i \leq n)$ forment une base de $\mathbb{R}_n[X]$.

L'écriture d'un polynôme dans cette base s'appelle *forme de Newton*.

Entrée: $c_0, \ldots, c_{n-1}$, centres de Newton ; $d_0, \ldots, d_n$, coefficients de Newton du polynôme P ; t, réel
Sortie: $val = P(t)$
$val \leftarrow d_n$;
pour i *de* $n-1$ *à* 0 **faire**
| $val \leftarrow d_i + (t - c_i) \times val$;
fpour
retourner *val;*

4.2 Le polynôme d'interpolation

Théorème 4.2.1 *Soit $(c_0, y_0), \ldots, (c_n, y_n)$ une suite de points, avec les c_i deux à deux distincts. Il existe un unique polynôme d'interpolation de degré au plus n :*

$$L_n(x) = \sum_{i=0}^{n} y_i . \frac{\prod\limits_{0 \leq j \leq n, j \neq i} (x - c_j)}{\prod\limits_{0 \leq j \leq n, j \neq i} (c_i - c_j)}$$

Soit f la fonction qu'on cherche à interpoler aux centres $c_0, c_1, \ldots$ ($y_i = f(c_i)$).

Le k-ième coefficient de Newton du polynôme d'interpolation est noté $d_k = f[c_0, \ldots, c_k]$, et on l'appelle la différence divisée relative à $c_0, \ldots, c_k$.

$$L_n(x) = \sum_{k=0}^{n+1} d_k.P_k \tag{4.1}$$

Proposition 4.2.2 *On*[1] *a* $f[c_0, \ldots, c_n] = \sum_{0 \leq i \leq n} \frac{f(c_i)}{\prod_{0 \leq j \leq n, j \neq i} (c_i - c_j)}$

Corollaire 4.2.3 *On a :* $d_0 = y_0$, $d_1 = \frac{y_1 - y_0}{c_1 - c_0}$,

$$d_k = f[c_0, \ldots, c_k] = \frac{f[c_1, \ldots, c_k] - f[c_0, \ldots, c_{k-1}]}{c_k - c_0}$$

Détermination des différences divisées

Entrée: n, entier ; $c_0, \ldots, c_n$, centres distincts ; $y_0, \ldots, y_n$, valeurs.
Sortie: $d = (d_0, \ldots, d_n)$, vecteur des différences divisées.

```
pour i de 0 à n faire
 | d_i ← y_i;
fpour
pour i de1 à n faire
 | pour j de n à i faire
 | | d_j ← (d_j − d_{j−1}) / (c_j − c_{j−i}) ;
 | fpour
fpour
```

Algorithme 2: Calcul des différences divisées

Majoration de l'erreur

Théorème 4.2.4 *Soit f une fonction C^{n+1} sur $[a, b]$, notons L_n le polynôme d'interpolation de f en $n+1$ points distincts (c_i) de $[a, b]$. Alors, pour tout x de $[a, b]$, il existe $\xi_x \in [a, b]$ tel que :*

$$|f(x) - L_n(x)| \leq \left| \frac{f^{(n+1)}(\xi_x)}{(n+1)!}.P_{n+1}(x) \right|$$

où $P_{n+1}(x) = \prod_{i=0}^{n} (x - c_i)$.

Exercice 4.2.5 *(Preuve du th). On pose : $E_n = f - L_n$, et pour un x fixé distinct des (c_i) dans $[a, b]$, on pose aussi $G(t) = E_n(t) - \frac{P_{n+1}(t)}{P_{n+1}(x)}.E_n(x)$. Montrer que G est C^{n+1} sur $[a, b]$ et y possède $n+2$ zéros. Déduire le résultat du théorème des valeurs intermédiaires*

Exercice 4.2.6 *(Idées de projets) Etudiez l'interpolation avec des points égaux, ou bien étudiez le cas de la dimension 2, ou de l'interpolation par morceaux. Splines.*

Réference : Quarteroni-Sacco-Saleri : "méthodes numériques"

1. en identifiant le coefficient de x^n dans la formule : 4.1