INSA Département GM : TP interpolation polynomiale et MNT

Dans ce TP, on se propose d'étudier la construction d'un *Modèle Numérique de Terrain* (voir MNT ou le cours). Étant donné une série de mesures sur une grille cartésienne, c'est à dire des valeurs $z_{i,j}$ en chaque point $(x_i = ih,\ \ y_j = jh)$, on souhaite construire à l'aide de l'interpolation polynomiale une fonction $p(x, y)$ interpolant ces données.

Vous rendrez pour deux semaines après la séance de TP vos **codes** et vos **rapport au format pdf**. Je vous invite à regrouper ces fichiers dans un document .zip.

Cas des fonctions 1D

Pour commencer à aborder ce problème, revoyons l'interpolation pour un fonction d'une variable. Étant donné $(n + 1)$ points $\{x_0, \cdots, x_n\}$, l'objectif est de déterminer le polynôme $p_n(x)$ de degré n vérifiant $p_n(x_i) = y_i$.

(Théorie)

1. Expliquer comment à l'aide de l'algorithme des *différences-divisées* et l'algorithme de *Horner* on peut évaluer le polynôme d'interpolation en x donné. Expliquer en particulier l'algorithme de *Horner* rappelé ci-après.

(Code)

1. Compléter dans le fichier *algo.py* la fonction *NevilleAitken* implémentant l'algorithme de *Neville-Aitken*. Attention, l'argument des points d'évaluation est un vecteur !

2. a) Compléter dans le même fichier la fonction *DiffDiv* implémentant l'algorithme des *Différence Divisées*.

   b) Implémenter ensuite la méthode de *Horner* dont l'algorithme est rappeler ci-après.

3. Dans le fichier *data.py*, modifier la fonction f pour avoir $f(x) = \cos(2\pi x)$:

   a) Tester à l'aide du fichier *main.py*, en choisissant les paramètres $a = 0$ et $b = 1$, la méthode de *Neville-Aitken*.

   b) De même, tester la méthode *Différence-Divisée - Horner*.

   c) Dans les deux cas ci-dessus, tester également les points *équi-répartis* et de *Chebychev*. Faites varier le nombre de points d'interpolation. Commenter.

4. Dans le fichier *data.py*, modifier la fonction f pour avoir $f(x) = |x|$:

   a) Tester comme précédemment, en choisissant les paramètres $a = -1$ et $b = 1$ et les points *équi-répartis*, et les méthodes de *Neville-Aitken* et *Différence-Divisée - Horner*. Commenter.

   b) Tester maintenant avec les points de *Chebychev*. Commenter.

Application au MNT

Voyons maintenant l'application au cas des MNT. On considère que le fichier *mesure.data* contient la mesure de la profondeur de l'océan en différents points sur une grille cartésienne (par exemple pour cartographier les abysses, zone moins bien connue que la surface lunaire !).

L'objectif va être de déterminer le polynôme d'interpolation $p_n(x, y) \in \mathbb{Q}_n$ où

$$\mathbb{Q}_n = \text{vect}\ \{x^i y^j \quad \forall (i, j) \in [0, n]^2\}$$

vérifiant $p_n(x_i, y_j) = z_{i,j}$ pour tout points $(x_i, y_j) = [ih - a, jh - a]$ avec $h = 2a/n$ et $(i, j) \in [0, n]^2$.

(Théorie)

1. En supposant que le problème d'interpolation ci-dessus admet une solution, montrer qu'elle est unique.
   Indication : On pourra procéder par l'absurde en supposant que deux polynômes p_n^1 et p_n^2 sont solutions et montrer que la différence $v = p_n^1 - p_n^2$ est nécessairement nulle.

2. a) Expliquer le rôle et le fonctionnement de la méthode *NevilleAitken2D* dans le fichier *algo.py*.

b) Peut-on faire une méthode d'évaluation de $p_n(x, y)$ se basant sur les *différences-divisées* et la méthode de *Horner* ? Si oui, expliquer comment. Si non, expliquer ce qui pose problème !

(Code)

1. a) Commencer par tester à l'aide du programme *main.py* l'affichage de la grille de mesures pour construire le MNT.

   b) Ensuite, tester la construction du MNT.

2. Construire un nouveau jeu de données, sur le même format que le fichier *mesures.data* (i.e. un tableau de taille 5×5) et tester la construction du MNT pour ce nouveau jeu de données.

3. **(Bonus)** Implémenter et tester une méthode d'évaluation de $p_n(x, y)$ basée sur les *différences-divisées* et la méthode de *Horner*.

Quelques remarques de conclusion

L'un des objectifs de ce TP a été de vous présenter une application de l'interpolation polynomiale. Au travers des MNT, nous avons abordé l'interpolation multivariée qui soulève de nombreuses difficultés, en particulier si la grille de données n'est pas cartésienne. Noter que ce type de problématique apparaît également en météorologie lorsqu'on souhaite dessiner une carte des températures ou encore en redimensionnement d'images.

Enfin pour conclure, il est important de savoir qu'on utilise davantage des courbes splines pour résoudre les problèmes d'interpolation, ce que vous découvrirez en détails l'année prochaine !

Rappel algorithme

[Entrées / Sorties :]
Entrées : coeff : **Tableau** ; ptinterp : **Tableau** ; x : **Tableau**
Sortie : res : **Tableau**

[Initialisation :]
n ← longueur(coeff)
res ← coeff(n-1) + coeff(n)× (x - ptinterp(n-1))

[Itérations :]
Pour i **de** 1 **à** n-2 **faire**
  | res ← coeff(n-1-i) + (x - ptinterp(n-1-i))×res
Fin Pour

Algorithme 1: Algorithme de Horner pour évaluer p_n en x