Université d'Avignon - UFR Sciences
Licence de mathématiques - S5
Analyse numérique 2016-2017

Léo Brunswic
leo.brunswic.fr

TP n°2
Polynômes orthogonaux, résolution d'équation et interpolation

Regarder l'aide obtenue en appuyant sur F1 doit être votre premier réflexe

Ce TP est consacré à la construction de "bons" points d'interpolation. Quelques mots clefs utiles : *produit scalaire, espace pre-hilbertien, racines, orthogonalisation de Schmidt*

1 Polynômes orthogonaux

Soit $\omega :]a, b[\to \mathbb{R}_+^*$ continue telle que $\int_a^b \omega < +\infty$,

$$E = \mathbb{R}[X] \qquad \langle P, Q \rangle = \int_a^b P(x)Q(x)\omega(x)\mathrm{d}x$$

On appelle ω le poid.
Question 1. On se place dans l'espace vectoriel $E = \mathbb{R}[X]$ muni du produit scalaire ci-dessus. Montrer qu'il existe une unique suite de polynôme $(P_n^\omega)_{n\in\mathbb{N}}$ telle que
— P_n^ω est de degré n.
— P_n^ω est unitaire.
— $\langle P_i^\omega, P_j^\omega \rangle = 0$ si $i \neq j$
Montrer la relation de récurrence suivante sur les P_n^ω :

$$P_{n+2}^\omega = XP_{n+1}^\omega - \frac{\langle P_{n+1}^\omega, XP_{n+1}^\omega \rangle}{\langle P_{n+1}^\omega, P_{n+1}^\omega \rangle} P_{n+1}^\omega - \frac{\langle P_n^\omega, XP_{n+1}^\omega \rangle}{\langle P_n^\omega, P_n^\omega \rangle} P_n^\omega$$

Definition 1 *P_n^ω est le n-ième polynôme orthogonal associé au poid ω.*

Question 2. Ecrire un script qui prend entrée un intervalle, un poid sur cet intervalle et un entier n et renvoit le n-ième polynôme orthogonal associé à ce poid.

Entrées	Sortie
$a, b \in \mathbb{R}$ $n \in \mathbb{N}$ $\omega : [a, b] \to \mathbb{R}_+^*$	P_n^ω

Question 3. Application numérique

$$a = -1, b = 1, \omega = 1, n = 1, 2, 3, \dots, 10$$

Ces polynômes sont les polynômes de Lagrange. Attention, si vous recherchez ces polynômes, sur internet par exemple, ils peuvent différer d'une constante multiplicative. Nous normalisons nos polynômes en leur demandant d'être unitaires, d'autres sources les normalisent en demandant $(P_n^\omega)_{n\in\mathbb{N}}$ d'être une base orthonormale.

Question 4. Montrer que pour tout $n \in \mathbb{N}$, P_n^ω est un polynôme scindé à racines simples et que ses racines sont dans l'intervalle $[a, b]$.

Ecrire une fonction qui prend en entrée un polynôme et renvoie la liste de ses racines. On va supposer que le polynôme en entrée est toujours scindé à racine simple dans $\mathbb{R}$. Vous pouvez implémenter une methode par dichotomie ou de Newton au choix.

Entrées	Sortie
$P \in \mathbb{R}[X]$	$[\lambda_1, \cdots, \lambda_n]$

Question 5. Application numérique

$$P = x^6 - 14x^5 + 70x^4 - 140x^3 + 49x^2 + 154x - 120$$

Definition 2 *On note $\lambda^{(n,\omega)}$ le vecteur des racines de P_n^ω classés par ordre croissant.*

Question 6. Ecrire un script qui prend en entrée un intervalle, un ordre et un poid et retourne la liste des racines du n-ième polynôme orthogonal associé au poid.

Entrées	Sortie
$a, b \in \mathbb{R}$ $n \in \mathbb{N}$ $\omega : [a, b] \to \mathbb{R}_+^*$	$[\lambda_1^{(n,\omega)}, \cdots, \lambda_n^{(n,\omega)}]$

2 Interpolation par la méthode de Newton aux racines des polynômes orthogonaux

Definition 3 *Soit $f : [a, b] \to \mathbb{R}$ une fonction et x un vecteur de points distincts dans $[a, b]$. On note $P_{f,x}$ l'interpolation de Lagrange f aux points $[x_1, \cdots, x_n]$.*

Question 7. Ecrire un script qui prend en entrée les points d'interpolations et une fonction et retourne son polynôme d'interpolation aux points donnés. On vous demande de le construire par la méthode de Newton.

Entrées	Sortie
n $[x_1, \cdots, x_n]$ f	$P_{f,x}$

Question 8. Application numérique :

$$n = 10, \quad \forall i \in [[1, n]], x_i = -1 + \frac{2i}{n}, \quad f = \frac{1}{1 + 25x^2}$$

Question 9. Ecrire un script qui prend en entrée un intervalle,un poid, un entier n et un fonction et qui retourne le polynôme d'interpolation de la fonction en les racines du n-ième polynôme orthogonal associé au poid.

Entrées	Sortie
a, b	
n	$P_{f,\lambda^{(n,\omega)}}$
ω	
f	

Question 10. Application numérique :

$$a = -1, b = 1, \omega = \frac{1}{\sqrt{1-x^2}}, f = \frac{1}{1+25x^2}$$

Question 11. On prend $a = -1, b = 1$ et $\omega(x) = \frac{1}{\sqrt{1-x^2}}$. Montrer que $P_n^{\omega}(x) = \arccos(n\cos(x))$ et en déduire une formule pour $\lambda_i^{(n)}$.

En remarquant que $\lambda_{2i}^{(2n)} = \lambda_i^{(n)}$, écrire un script efficace pour calculer le 2^p-ième polynôme d'interpolation d'une fonction f donné pour les points λ^{2^p}.

3 Compléments le calcul du produit scalaire

Dans la partie 1, le produit scalaire avait été très succintement implémenté en utilisant une méthode des rectangles (à gauche).

Question 12. Implémenter le produit scalaire de deux polynômes en utilisant la méthode des trapèzes.

Entrées	Sortie
ω	
P	$\int_a^b \omega(t)P(t)\mathrm{d}t$
a, b	

On remarque un gain de précision notable en utilisant cette méthode plutot que la méthode des rectangles à gauche. Cela permet notamment un gain dans la rapidité du programme construisant l'interpolation d'une fonction aux racines des polynômes orthogonaux.

Question 13. Ecrire un script calculant le produit scalaire de deux fonctions de manière approchée en les remplaçant par leur polynômes d'interpolation aux racines de P_n^{ω} pour un certain n.

Question 14. Que pourrait-on faire pour significativement améliorer la rapidité du dernier script et utiliser pleinement les ressources de scilab ?

Pour lundi 21/11 écrire un script qui applique cette méthode au calcul d'interpolation aux racines de polynômes orthogonaux de votre choix. Donner également des représentations graphiques légendée illustrant les polynômes orthogonaux que vous avez choisis. Donner une table des polynômes orthogonaux que vous avez construit.

La note dépendra du degré du plus grand polynôme orthogonal que vous aurez réussi à calculer de manière fiable.