MT12 - TP2 : Développements sur une base orthogonale

Dans ce TP on considère l'espace des fonctions continues sur $[-1,1]$ à valeurs dans $\mathbb{R}$, noté $H = C^0([-1,1];\mathbb{R})$, muni du produit scalaire

$$\langle f, g\rangle = \int_{-1}^{1} f(t)g(t)\,dt, \qquad f, g \in H, \tag{1}$$

et la norme associée $\|f\| = \sqrt{\langle f, f\rangle}$. Le but de cette séance est d'introduire une famille de polynôme orthogonaux, les polynômes de Legendre notés $(P_k)_{k\geq 0}$, qui forme une famille totale de H, i.e., $(P_k)_{k\geq 0}$ engendre un sous-espace vectoriel dense dans H. Plus précisément, pour toute fonction $f \in H$, pour tout $\varepsilon > 0$, il existe $n \in \mathbb{N}$ et $a_1, \dots, a_n \in \mathbb{R}$ tels que

$$\left\| f - \sum_{k=0}^{n} a_k P_k \right\| \leq \varepsilon.$$

Intuitivement, on peut voir les polynômes $(P_k)_{k\geq 0}$ comme une “base” de H. Il faut faire cependant attention, car ici H est de dimension infinie. La notion de famille totale généralise la notion de base algébrique d'un espace vectoriel de dimension finie. Nous allons voir dans cette séance des applications pratiques de cette notion de famille totale.

---

1 Exercice préliminaire : récursivité

On a vu dans le TP1 une manière de coder une suite, en ayant le terme générale de la suite. Ce n'est pas toujours possible. Quand ce n'est pas possible, il faut utiliser *la récursivité*. Pour faire simple, cela consiste à :

(a) Définir un terme de départ.

(b) Définir un terme en fonction du précédent.

Ainsi, la suite étudiée au TP1 s'écrit, par récurrence :

$$\begin{cases} U_1 = 1, \\ U_{n+1} = \frac{U_n}{2}. \end{cases}$$

En scilab on écrit :

```
function U=Suite(n)
    if n<1 then
        U = null
    elseif n == 1 then
        U = 1
    else
        U = Suite(n - 1) / 2 // Calcule U_n puis le divise par 2.
    end
endfunction
```

Note importante : si vous entrez un nombre négatif, par exemple -3, il faut que votre fonction renvoie « null ». En effet, si ce n'est pas le cas, votre machine va chercher à calculer Suite(-4), qui elle même va chercher à calculer Suite(-5), etc, et ne va jamais s'arrêter. Il faut donc s'assurer que la condition d'arrêt (calcul de U_1) est atteinte à un moment donné quand on écrit une fonction récursive.

Remarque : renvoyer *null* correspond à détruire la variable cible ! Dans l'exemple qui suit la variable u n'est pas définie.

```
u=Suite(-4) // u n'est pas defini!
if exists('u') then // la fonction exists permet de tester si une variable (ou une fonction) est definie
    disp(u)
end
```

1. Écrire une fonction `Scilab` qui code la suite
$$\begin{cases} U_0 = 1, \\ U_{n+1} = U_n + 1. \end{cases}$$

2. Écrire une fonction `Scilab` qui code la suite de Fibonacci
$$\begin{cases} U_0 = 1, \\ U_1 = 1, \\ U_{n+2} = U_{n+1} + U_n. \end{cases}$$

3. Écrire une fonction `Scilab` qui code la suite de fonctions
$$\begin{cases} f_0(x) = x, \\ f_{n+1}(x) = x \cdot f_n. \end{cases}$$

4. Quelle est cette suite de fonctions ? Affichez dans l'intervalle $[-1, 1]$ les trois premiers termes de cette suite.

2 Polynômes de Legendre

Les polynômes de Legendre sont des polynômes solutions de l'équation différentielle

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[(1-x^2)\frac{\mathrm{d}P_n(x)}{\mathrm{d}x}\right] + n(n+1)\,P_n(x) = 0, \qquad P_n(1) = 1.$$

On peut démontrer la formule de récurrence

$$(n+1)\,P_{n+1}(x) = (2n+1)\,x\,P_n(x) - n\,P_{n-1}(x), \tag{2}$$

pour tout entier $n \geq 1$, partant de $P_0(x) = 1$, $P_1(x) = x$. Calculez à la main les premiers polynômes de Legendre et vérifiez que

$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2},\ P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x,\ P_4(x) = \frac{1}{8}\left(35x^4 - 30x^2 + 3\right),$$

$$P_5(x) = \frac{1}{8}\left(63x^5 - 70x^3 + 15x\right),\ P_6(x) = \frac{1}{16}\left(231x^6 - 315x^4 + 105x^2 - 5\right).$$

Les polynômes de Legendre sont deux à deux orthogonaux (mais pas orthonormés) pour le produit scalaire de H, et on montre que

$$\langle P_n, P_m \rangle = \int_{-1}^{1} P_m(x) P_n(x)\, dx = \frac{2}{2n+1}\, \delta_{mn}. \tag{3}$$

La famille $(P_k/\|P_k\|)_{k\geq 0}$ est donc orthonormée dans H. On peut montrer que c'est une « base » (famille totale) de H. On remarque aussi que les polynômes P_{2p} sont paires et P_{2p+1} sont impaires.

5. Écrire une fonction `Scilab` qui code les polynômes de Legendre de degré n au moyen de la formule de récurrence (2). Pour cela il suffit de compléter le code suivant

```
function y=LegendreP(n,x)
    if n==... then
        y=...;
    elseif n==... then
        y=...;
    elseif n>... then
        y=...;
    end
endfunction
```

On souhaite désormais calculer la meilleure approximation de la fonction $f : x \mapsto \exp(-x)$ sur $[-1,1]$ sur le sous-espace vectoriel engendré par les polynômes de Legendre de degré inférieur à n. Plus précisément, en reprenant les notations du chapitre 2, si on note $\mathcal{P}_n = \mathrm{Vect}\,\langle P_0, \ldots, P_n \rangle \subset H$, on cherche un polynôme P de $\mathcal{P}_n$ réalisant la meilleure approximation de f dans $\mathcal{P}_n$. Autrement dit on cherche un polynôme $P \in \mathcal{P}_n$ solution du problème de minimisation suivant

$$\|f - P\| = \min\{\|f - Q\|,\ Q \in \mathcal{P}_n\}.$$

En reprenant les calculs du chapitre 2 on montre que l'unique polynôme, noté $\pi_n f$, solution de ce problème est donné par la formule

$$\pi_n f(x) = \sum_{k=0}^{n} a_k \frac{P_k(x)}{\|P_k\|}, \qquad a_k = \langle f, \frac{P_k}{\|P_k\|} \rangle$$

où $\|P_k\| = \sqrt{\langle P_k, P_k \rangle}$.

6. On liste ci-dessous la valeur des a_k pour $k = 0, \ldots, 6$

```
a_0=sqrt(2)*sinh(1),
a_1=-sqrt(6)/exp(1)
a_2=(exp(2)-7)/exp(1)*sqrt(5/2)
a_3=(5*exp(2)-37)/exp(1)*sqrt(7/2)
a_4=3*sqrt(2)*(18*exp(2)-133)/exp(1)
a_5=(329*exp(2)-2431)/exp(1)*sqrt(11/2)
a_6=(3655*exp(2)-27007)/exp(1)*sqrt(13/2)
```

Écrivez un script qui propose une vérification de l'inégalité de Bessel

$$\|\pi_6 f\|^2 = \sum_{k=0}^{6} a_k^2 \leq \|f\|^2.$$

Pour ce faire on calculera la valeur exacte de $\|f\|^2$.

7. On considère une discrétisation uniforme de l'intervalle $[-1, 1]$ avec 400 points :

```
x = linspace(-1,1,400);
```

Sur un graphique, représentez les 7 premiers polynômes de Legendre P_k. Sur un autre graphique, tracer la fonction $x \mapsto f(x)$. Calculer et tracer sur le même graphique $\pi_6 f(x)$. On pourra aussi afficher la fonction d'erreur de projection

$$\varepsilon(x) = f(x) - \pi_6 f(x).$$

8. On souhaite faire de même avec la fonction $g : x \mapsto |x|$. Donnez l'expression des a_k pour cette fonction. Prouvez que ces a_k sont nuls pour k impair et simplifiez l'expression pour les k pairs.
9. Proposez une fonction qui pour un k donné, renvoie le a_k correspondant. Pour calculer l'intégrale, on pourra regarder la documentation de *integrate* en utilisant *help integrate*. On souhaite que la fonction se comporte différemment selon que k est pair ou impair.
10. Sachant que $a_0 = 1/\sqrt{2}$, $a_2 = \sqrt{10}/8$ et $a_4 = -\sqrt{2}/16$, vérifiez le bon fonctionnement de votre fonction.
11. Affichez sur une même figure les polynômes $\pi_n g$ pour $n = 2$, $n = 4$ et $n = 6$ (dans la partie haute) et l'erreur d'approximation (dans la partie basse).
12. Commentez la figure obtenue. Cette méthode d'approximation semble-t-elle converger rapidement ?