

TP noté : interpolation polynomiale

Instructions

Au début de la séance, créez par l'intermédiaire de l'éditeur un document que vous nommerez `nom.prenom.py` (avec vos propres nom et prénom bien sûr) et que vous sauvegarderez dans le répertoire MPSI4 qui se trouve dans le répertoire Documents. Si ce répertoire n'existe pas, c'est que votre ordinateur n'est pas relié au réseau ; dans ce cas, sauvegardez-le directement dans le répertoire Documents et appelez-nous à la fin de l'épreuve pour que nous copions sur une clé USB le résultat de votre travail.

Ce document devra contenir l'ensemble des définitions et des scripts qui vous seront demandés en respectant les noms et l'ordre des paramètres qui seront indiqués dans l'énoncé, de manière à permettre une vérification automatique de votre code. Vous pouvez, si vous l'estimez nécessaire, définir des fonctions supplémentaires, mais dans ce cas ces dernières devront impérativement être accompagnées de commentaires indiquant leur rôle. Pour rappel, toute ligne débutant par un croisillon `#` est un commentaire qui ne sera pas lu par l'interprète de commande.

Enfin, pensez à sauvegarder régulièrement votre travail, et n'éteignez pas votre ordinateur avant d'avoir obtenu confirmation que nous avons en notre possession votre copie numérique.

Mise en garde. Ce sujet est long et de difficulté croissante et il est prévisible que la troisième partie ne sera abordée que par peu d'étudiants (voire aucun). Il n'est donc pas nécessaire de résoudre le problème dans son entier pour obtenir la note maximale.

Introduction

Le but de cette séance est de définir un certain nombre d'outils spécifiques aux polynômes à coefficients réels, avant de s'intéresser plus spécifiquement à l'interpolation d'une fonction par un polynôme. Ces derniers seront représentés par des

listes ; plus précisément, le polynôme $P = \sum_{k=0}^n a_k X^k$ sera représenté par la liste $[a_n, a_{n-1}, \dots, a_1, a_0]$ de ses coefficients, rangés par ordre des degrés décroissants.

Par la suite, on identifiera un polynôme P et sa représentation par une liste p . Si $P \neq 0$, on dira que cette représentation est *normalisée* lorsque $\text{len}(p)$ est égale à $\deg P + 1$; la représentation normalisée du polynôme nul sera la liste vide.

Question 1. Définir une fonction `normalise` qui prend en argument une représentation quelconque d'un polynôme P et la normalise.

Désormais, on supposera que les polynômes qui seront passés en argument des fonctions seront normalisés, et on exigera de ces dernières qu'elles retournent des polynômes normalisés.

1. Opérations usuelles sur les polynômes

Question 2. Étant donné un polynôme non nul $P = \sum_{k=0}^n a_k X^k$ et un réel x , on considère la suite finie $(u_0, u_1, \dots, u_n)$ définie par les relations :

$$u_0 = a_n \quad \text{et} \quad \forall k \in \llbracket 1, n \rrbracket, u_k = x u_{k-1} + a_{n-k}$$

Déterminer la valeur de u_n et en déduire une fonction `eval` qui prend en arguments un polynôme P et un réel x et qui retourne le réel $P(x)$.

Question 3. Définir une fonction nommée `mult` qui prend en argument un scalaire a et un polynôme P et qui retourne le polynôme $a.P$.

Question 4. Définir une fonction nommée `somme` prenant en arguments deux polynômes P_1 et P_2 et retournant la représentation normalisée de $P_1 + P_2$.

Question 5. Définir de même la fonction nommée `produit` retournant le produit des deux polynômes P_1 et P_2 .

2. Interpolation polynomiale

2.1 Polynômes de Lagrange

Dans cette partie, $x_0, x_1, \dots, x_n$ désignent $n+1$ réels deux à deux distincts. On appelle *polynômes de Lagrange* associés aux *nœuds* (x_j) les polynômes L_i pour $i \in \llbracket 0, n \rrbracket$ définis par :

$$L_i = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}.$$

Question 6. Définir une fonction `lagrange` qui prend en arguments la liste des nœuds (x_j) et un entier i et qui retourne le polynôme L_i .

Question 7. À quoi est égal $L_i(x_j)$? En déduire pour tout $(n+1)$ -uplet $(y_0, \dots, y_n)$ l'existence d'un polynôme P de degré inférieur ou égal à n vérifiant : $\forall j \in \llbracket 0, n \rrbracket, P(x_j) = y_j$.

Il est facile de prouver que ce polynôme P est l'unique polynôme de degré inférieur ou égal à n vérifiant ces conditions d'interpolation ; il sera désormais appelé le *polynôme d'interpolation de Lagrange* associé aux points $(x_j, y_j)_{j=0, \dots, n}$.

Définir une fonction nommée `interpole` qui prend en arguments les deux listes (x_j) et (y_j) et qui retourne ce polynôme P .

2.2 Interpolation polynomiale d'une fonction

Étant donnée une fonction $f : [a, b] \rightarrow \mathbb{R}$ et $n+1$ réels $x_0, x_1, \dots, x_n$ deux à deux distincts de l'intervalle $[a, b]$, on appelle *polynôme interpolant de f* le polynôme d'interpolation de Lagrange associé aux points $(x_j, f(x_j))_{j=0, \dots, n}$. Celui-ci sera désormais noté $P_n f$.

Nous allons nous intéresser à la question de la convergence uniforme du polynôme interpolant lorsque le nombre de nœuds n tend vers $+\infty$; autrement dit, nous allons chercher à déterminer dans quelle mesure la quantité :

$$M_n = \sup \{ |f(t) - P_n f(t)| \mid t \in [a, b] \}$$

tend vers 0 lorsque n tend vers $+\infty$.

Question 8. Définir une fonction baptisée *norme* prenant en arguments une fonction g et deux réels a et b et qui retourne la quantité : $\max \{ |g(t_k)| \mid 0 \leq k \leq 1000 \}$ avec $t_k = a + k \frac{b-a}{1000}$. Par la suite, cette fonction nous servira à évaluer la quantité M_n .

• Répartition uniforme des nœuds

Comme le polynôme interpolant dépend de la distribution des nœuds d'interpolation, il est nécessaire de préciser la répartition de ces derniers. Nous allons pour l'instant faire le choix d'une répartition uniforme des nœuds en posant :

$$\forall j \in \llbracket 0, n \rrbracket, \quad x_j = a + j \frac{b-a}{n}.$$

On considère tout d'abord la fonction $f : t \mapsto \sin(\pi t)$ sur l'intervalle $[-1, 1]$.

Question 9. Rédiger un script produisant l'affichage dans une même fenêtre graphique du graphe de la fonction f et de ses polynômes interpolants $P_n f$ pour $n = 3, 4, 5, 6$.

Question 10. Rédiger ensuite un script pour déterminer la plus petite valeur de n pour laquelle $M_n \leq 10^{-6}$.

• Phénomène de Runge

On considère désormais la fonction $f : t \mapsto \frac{1}{1+8t^2}$, toujours sur l'intervalle $[-1, 1]$.

Question 11. Rédiger un script produisant l'affichage dans une même fenêtre graphique du graphe de la fonction f et de ses polynômes interpolants $P_n f$ pour $n = 10, 12, 14, 16$.

La divergence de l'interpolation que l'on observe au voisinage des extrémités de l'intervalle porte de nom de *phénomène de Runge*. Cet effet fâcheux peut être évité en choisissant pour nœuds les *points de Tchebychev* ;

$$x_j = \cos\left(\frac{(2j+1)\pi}{2(n+1)}\right) \quad j = 0, \dots, n.$$

Question 12. Reprendre la même question mais en choisissant cette fois pour nœuds les points de Tchebychev.

Question 13. Avec ces points, pour quelle valeur minimale de n obtient-on $M_n \leq 10^{-3}$?

3. Splines cubiques

Nous n'avons pour l'instant attaqué le problème de l'approximation d'une fonction f sur un intervalle $[a, b]$ par un polynôme qu'en un sens global, c'est-à-dire en cherchant à n'utiliser qu'une seule expression analytique de l'interpolant sur $[a, b]$. Pour obtenir une approximation plus précise, on n'a dès lors d'autre choix que d'augmenter le degré du polynôme d'interpolation. L'exemple de Runge évoqué ci-dessus montre que la convergence uniforme n'est cependant pas garantie pour toute distribution arbitraire des nœuds.

Une alternative à cette première approche est de construire une partition de l'intervalle $[a, b]$ en sous-intervalles sur chacun desquels on emploie une interpolation polynomiale de bas degré (3 en ce qui nous concerne) qui porte le nom de *spline*.

Question 14. Étant donnée une fonction $f : [u, v] \rightarrow \mathbb{R}$ de classe $\mathcal{C}^1$, trouver l'unique polynôme $S_{u,v}$ de degré inférieur ou égal à 3 qui vérifie les conditions d'interpolation :

$$S_{u,v}(u) = f(u), \quad S'_{u,v}(u) = f'(u), \quad S_{u,v}(v) = f(v), \quad S'_{u,v}(v) = f'(v).$$

Indication. On pourra chercher $S_{u,v}$ sous la forme : $S_{u,v} = a(X-u)^3 + b(X-u)^2(X-v) + c(X-u)(X-v)^2 + d(X-v)^3$.

En déduire une fonction baptisée *spline* qui prend en arguments la fonction f , la fonction f' , u et v et qui retourne le polynôme $S_{u,v}$.

L'approximation par les splines cubiques d'une fonction f de classe $\mathcal{C}^1$ sur l'intervalle $[a, b]$ consiste à subdiviser régulièrement l'intervalle $[a, b]$ en posant $x_j = a + j \frac{b-a}{n}$, $j = 0, \dots, n$ et à approcher sur chaque intervalle $[x_j, x_{j+1}]$ la fonction f par le spline $S_{x_j, x_{j+1}}$.

Question 15. On considère la fonction $f : t \mapsto \sin(t^2)$ définie sur l'intervalle $[0, 4]$. Rédiger un script permettant de visualiser dans une même fenêtre graphique la fonction f , le polynôme interpolant de f pour $n = 12$, et son approximation par les splines cubiques, toujours pour $n = 12$.