Devoir de vacances de mathématiques et d'informatique

Les questions de maths seront rendues en classe. Les scripts Python sur CDP (transfert de copie). Ce devoir est un devoir de recherche, cela veut dire qu'il faudra prendre du temps pour le traiter, il demande un peu de recul car brasse beaucoup de notions vu dans des chapitres différents : polynômes, espaces pré-hilbertiens, intégration, dérivabilité, développements limités, mais aussi en informatique : graphisme, gestion des listes. En particulier, prenez bien le temps de lire les questions mais aussi les paragraphes qui introduisent les notions.

Objectifs

Le fil conducteur de ce sujet est le calcul approché d'intégrales. Les parties 1 et 2 peuvent être traitées de manière indépendante. La partie 3 utilise des résultats des parties 1 et 2. Les parties 1, 2 et 3 traitent de l'utilisation de polynômes interpolateurs pour le calcul approché d'intégrales : on présente le principe des méthodes de quadrature, dite de Newton-Cotes, ainsi qu'un raffinement avec la méthode de quadrature de Gauss.

Notations

- Si f est une fonction réelle bornée sur $[\,a\,;b\,]$ avec $a<b$, on pose $\|f\|_\infty = \sup_{x\in[\,a\,;b\,]} |f(x)|$.
- On note $\mathbb{R}_n[X]$ l'ensemble des polynômes à coefficients réels de degré inférieur ou égal à n. On pourra confondre les expressions "polynômes" et "fonctions polynomiales".

Partie 1 : Notion de polynôme interpolateur

Soit $f\colon [\,a\,;b\,]\to\mathbb{R}$ une fonction continue. On se donne $n+1$ points $x_0, x_1, \dots, x_n$ dans $[\,a\,;b\,]$, deux à deux distincts. On appelle polynôme interpolateur de f aux points x_i, un polynôme $P\in\mathbb{R}_n[X]$ qui coïncide avec f aux points x_i, c'est-à-dire tel que pour tout $i\in[\![\,0\,;n\,]\!]$, $P(x_i)=f(x_i)$.

Existence du polynôme interpolateur

Pour tout entier $i\in[\![\,0\,;n\,]\!]$, on pose $\ell_i(X)=\prod_{\substack{k=0\\k\neq i}}^{n}\frac{X-x_k}{x_i-x_k}\in\mathbb{R}_n[X]$. On pose : $L_n(f)=\sum_{i=0}^{n} f(x_i)\ell_i(X)$

1. Démontrer que $L_n(f)$ est un polynôme interpolateur de f aux points x_i, puis démontrer l'unicité d'un tel polynôme.

Un tel polynôme est appelé polynôme interpolateur de Lagrange.

Calcul effectif du polynôme interpolateur de Lagrange

2. *Informatique* : si $y_0,\dots,y_n$ sont des réels, le polynôme $P=\sum_{i=0}^{n} y_i\ell_i(X)$ est l'unique polynôme de $\mathbb{R}_n[X]$ vérifiant $P(x_i)=y_i$ pour tout i. Écrire une fonction `lagrange` qui prend en arguments x une liste de points d'interpolations x_i, y une liste d'ordonnées y_i de même longueur que x, a un réel, et qui renvoie la valeur de P en a.
   Par exemple, si `x=[-1,0,1]` et `y=[4,0,4]`, on montre que $P=4X^2$ et donc $P(3)=36$. Ainsi `lagrange(x,y,3)` renverra 36.

3. *Informatique* : chercher le polynôme interpolateur $P = a_0 + a_1X + \cdots + a_nX^n$ de f aux points x_i revient aussi à résoudre le système linéaire suivant d'inconnues $a_0, \ldots, a_n$:

$$\begin{cases} P(x_0) = f(x_0) \\ \vdots \\ P(x_n) = f(x_n) \end{cases} \iff V \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{pmatrix}$$

où V est une matrice carrée de taille $n+1$.
Déterminer la matrice V et indiquer la complexité du calcul en fonction de n, lorsque l'on résout ce système linéaire par la méthode du pivot de Gauss.

4. En déduire que cette matrice V est inversible.

Expression de l'erreur d'interpolation

On suppose, en plus dans cette partie, que $f \in \mathscr{C}^{n+1}([\,a\,;b\,],\mathbb{R})$. On rappelle que $L_n(f)$ est son unique polynôme interpolateur aux points x_i. On note $\sigma = \{x_0, \ldots, x_n\}$ l'ensemble des points d'interpolations et $\pi_\sigma = \prod_{i=0}^{n}(X - x_i) \in \mathbb{R}[X]$. On veut démontrer pour tout réel $x \in [\,a\,;b\,]$, la propriété suivante notée $\mathcal{P}_x$:

$$\exists c_x \in]\,a\,;b\,[\qquad f(x) - L_n(f)(x) = \frac{f^{(n+1)}(c_x)}{(n+1)!}\pi_\sigma(x)$$

5. Quel est le degré de π_σ ?
6. Résultat préliminaire : soit $\phi\colon [\,a\,;b\,] \to \mathbb{R}$ une fonction dérivable, et $x_1, x_2, \ldots, x_p \in [\,a\,;b\,]$ tel que $x_1 < x_2 < \ldots < x_p$ tels que pour tout $i \in [\![\,1\,;p\,]\!]$, $\phi(x_i) = 0$. Prouver que ϕ' s'annule (au moins) $p-1$ fois sur $[\,a\,;b\,]$.
7. Résultat préliminaire (suite et fin) : soit $p \in \mathbb{N}^*$. Démontrer que si $\phi\colon [\,a\,;b\,] \to \mathbb{R}$ est une fonction p-fois dérivable qui s'annule $p+1$ fois, alors il existe $c \in]\,a\,;b\,[$ tel que $\phi^{(p)}(c) = 0$.
8. Justifier que pour tout $x \in \sigma$, la propriété $\mathcal{P}_x$ est vraie.

On fixe $x \in [\,a\,;b\,]\backslash\sigma$. Soit λ un réel. On définit sur $[\,a\,;b\,]$ une application F par

$$F(t) = f(t) - L_n(f)(t) - \lambda\pi_\sigma(t)$$

9. Déterminer un réel λ de sorte que $F(x) = 0$. On choisira alors λ de cette façon.
10. Démontrer que F s'annule $n+2$ fois et en déduire que $\mathcal{P}_x$ est vraie.
11. Justifier que la fonction $f^{(n+1)}$ est bornée sur $[\,a\,;b\,]$ et en déduire un réel positif K indépendant de n tel que

$$\forall x \in [\,a\,;b\,] \qquad |f(x) - L_n(f)(x)| \leqslant \frac{K^{n+1}}{(n+1)!}\|f^{(n+1)}\|_\infty$$

12. On définit f sur $[\,-1\,;1\,]$ par $f(x) = \dfrac{1}{1+x^2}$. Donner le développement limité de f en 0, en déduire $f^{(k)}(0)$. Puis en déduire que

$$\forall k \in \mathbb{N} \qquad \|f^{(2k)}\|_\infty \geqslant (2k)!$$

Cette dernière inégalité montre que la quantité $\|f^{(n+1)}\|_\infty$ peut être grande et cela peut empêcher parfois la convergence de la suite de polynômes interpolateurs. Ceci est appelé le phénomène de Runge.

13. *Informatique* : illustrer le phénomène de Runge grâce à un script Python.

Partie 2 : famille de polynômes orthogonaux

On munit $\mathbb{R}[X]$ l'espace des polynômes à coefficients réels du produit scalaire $\langle \quad , \quad \rangle$ défini par : pour tous polynômes P et Q de $\mathbb{R}[X]$,

$$\langle P, Q\rangle = \int_{-1}^{1} P(t)Q(t)\,\mathrm{d}t$$

14. Démontrer que $\langle \quad , \quad \rangle$ est bien un produit scalaire sur $\mathbb{R}[X]$.
15. *Informatique :* en Python, un polynôme P de $\mathbb{R}_N[X]$ est codé comme la liste exhaustive de ses coefficients $a_0, a_1, \ldots, a_N$. Par exemple le polynôme $P = -1+2X+4X^3$ est entré sous la forme `P=[-1,2,0,4,0,...,0]` (avec autant de 0 que nécessaire afin que la liste ait $N+1$ éléments). Écrire une fonction `multiplication` qui prend en argument deux polynômes P et Q sous forme de liste et renvoie la liste des coefficients correspondant au polynôme produit `R` de $\mathbb{R}_{2N}[X]$.

On rappelle que si $P = \sum_{n=0}^{N} a_n X^n$ et $Q = \sum_{n=0}^{N} b_n X^n$ alors :

$$R = P \times Q = \sum_{n=0}^{2N} c_n X^n \text{ avec } c_n = \sum_{k=0}^{n} a_k b_{n-k}$$

On aura donc $c_0 = a_0.b_0$, $c_1 = a_0.b_1 + a_1.b_0$, $c_2 = a_0b_2 + a_1b_1 + a_2b_0$, etc.

16. *Informatique :* écrire une fonction `Intégrale` qui prend en argument une liste de coefficients correspondant à un polynôme P de $\mathbb{R}_N[X]$ ainsi que deux réels a et b et renvoie en sortie la valeur exacte de $\displaystyle\int_a^b P(t)\,\mathrm{d}t$.
(*on ne demande pas une valeur approchée obtenue par une quelconque méthode d'approximation mais bien la valeur exacte.*)
17. *Informatique :* En déduire une fonction `ProdScal` qui prend en argument deux listes de coefficients correspondants aux polynômes P et Q de $\mathbb{R}_N[X]$ et renvoie la valeur exacte du produit scalaire $\langle P, Q\rangle$ défini au début de la partie 2.

Pour $N \in \mathbb{N}^*$, on applique l'algorithme de Gram-Schmidt à la famille $(1, X, X^2, \ldots, X^N)$ de $\mathbb{R}[X]$. On obtient donc une famille orthonormée de polynômes $(P_0, P_1, P_2, \ldots, P_N)$ vérifiant

$$\forall k \in [\![\, 0\,;N\,]\!] \quad \text{vect}(1, X, \ldots, X^k) = \text{vect}(P_0, P_1, \ldots, P_k)$$

Le polynôme P_n s'appelle le polynôme de Legendre d'indice n.

18. Dans le cas où $N = 2$ (seulement pour cette question), calculer explicitement P_0, P_1 et P_2 grâce à l'algorithme de Gram-Schmidt
19. *Informatique :* écrire une fonction `Base_Canonique` qui prend en entrée un entier `N` et renvoie la liste `B` de taille $N+1$ où pour $k \in [\![\, 0\,;N\,]\!]$, `B[k]` est la liste des coefficients du monôme de X^k. (*Le monôme X^k est codé par la liste de taille $N+1$ où tous les éléments sont nuls sauf le $k-$ième qui vaut* 1.)
20. *Informatique :* écrire une fonction `Gram_Schmidt` qui prend en argument un entier N et renvoie l'expression (toujours sous forme de liste des coefficients) de la famille des polynômes de Legendre $(P_0, P_1, \ldots, P_n)$ (*il s'agit donc ici de programmer l'algorithme de Gram-Schmidt*). Le résultat renvoyé est donc une liste `F` de taille $n+1$ où pour $k \in [\![\, 0\,;n\,]\!]$, `F[k]` est la liste des coefficients du polynôme de Legendre P_k (utiliser les fonctions `Base_Canonique` et `ProdScal`). Tester cette fonction avec $N = 2$, les résultats sont-ils cohérents avec votre réponse à la question 18.
21. Justifier que pour $n \in [\![\, 1\,;N\,]\!]$, le polynôme P_n est orthogonal à $\mathbb{R}_{n-1}[X]$. Démontrer que le polynôme P_n est de degré n.

On prend $n \geqslant 1$. On veut démontrer que P_n admet n racines simples dans $[\,-1\,;1\,]$.

22. Justifier que $\displaystyle\int_{-1}^{1} P_n(t)\,\mathrm{d}t = 0$ et en déduire que P_n admet au moins une racine dans $[\,-1\,;1\,]$.

On suppose par l'absurde que P_n admet strictement moins de n racines simples. Si P_n admet des racines $t_1, \dots, t_p$ de multiplicité impaire avec $p < n$, on pose $Q = (X - t_1)\dots(X - t_p)$; sinon, on pose $Q = 1$. On considère enfin le polynôme $H = QP_n$.

23. Justifier que $\displaystyle\int_{-1}^{1} H(t)\mathrm{d}t = 0$, puis conclure (remarquer que H est de signe constant sur $[\,-1\,;1\,]$).

Partie 3 : méthodes de quadrature

Dans cette partie, nous allons voir comment les polynôme interpolateurs de Lagrange peuvent être utilisés pour estimer $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ pour $f \in \mathscr{C}^0([\,a\,;b\,],\mathbb{R})$. Pour cela, on choisit d'abord une subdivision $a = x_0 < x_1 < \cdots < x_N = b$ de l'intervalle $[\,a\,;b\,]$. A cause du phénomène de Runge, si N est grand, le polynôme interpolateur de f aux points x_i n'est pas forcément une bonne approximation de f. Approximer $\displaystyle\int_a^b f(x)\,\mathrm{d}x$ par $\displaystyle\int_a^b L_N(f)(x)\,\mathrm{d}x$ n'est donc pas forcément pertinent. Nous allons en fait approximer f par un polynôme d'interpolation sur chaque petit intervalle $[\,x_k\,;x_{k+1}\,]$. D'après la relation de Chasles, on a

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x$$

24. Justifier que

$$\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x = \frac{x_{k+1}-x_k}{2}\int_{-1}^{1} g(t)\,\mathrm{d}t \quad \text{avec} \quad g(t) = f\left(x_k + (t+1)\frac{x_{k+1}-x_k}{2}\right)$$

On est donc ramené à estimer $\displaystyle\int_{-1}^{1} g(t)\mathrm{d}t$ où $g \in \mathscr{C}^0([\,-1\,;1\,],\mathbb{R})$. On se donne $n+1$ points $t_0, t_1, \dots, t_n$ dans $[\,-1\,;1\,]$, deux à deux distincts. On rappelle que $L_n(g) = \displaystyle\sum_{i=0}^{n} g(t_i)\ell_i(X)$ est le polynôme interpolateur de g aux points t_i et on pose

$$J(g) = \int_{-1}^{1} L_n(g)(t)\,\mathrm{d}t = \sum_{i=0}^{n} \alpha_i g(t_i) \quad \text{avec} \quad \alpha_i = \int_{-1}^{1} \ell_i(t)\,\mathrm{d}t$$

Lorsqu'on approxime $\displaystyle\int_{-1}^{1} g(t)\,\mathrm{d}t$ par $J(g)$, c'est-à-dire : $\displaystyle\int_{-1}^{1} g(t)\,\mathrm{d}t \approx \sum_{i=0}^{n} \alpha_i g(t_i)$. On dit que J est une méthode de quadrature associée aux points $t_0, \dots, t_n$ et aux poids $\alpha_0, \dots, \alpha_n$.

25. *Informatique :* pour $i \in [\![\,0\,;n\,]\!]$, on suppose que l'expression des polynômes d'interpolation de Lagrange ℓ_i est stockée (toujours sous la forme de liste des coefficients) dans l'élément `L[i]` d'une liste `L` de taille $n+1$.
Écrire une fonction `Integrale_Lagrange` qui prend en entrée une liste `T` comportant $n+1$ valeurs $t_0, t_1, \dots, t_n$ et une fonction g (définie au préalable sous Python) et renvoie une valeur approchée de $J(g)$. Pour calculer les α_i on pourra se servir du programme `Integrale` de la question 16.

26. Justifier que pour tout polynôme $P \in \mathbb{R}_n[X]$, on a $\displaystyle J(P) = \int_{-1}^{1} P(t)\,\mathrm{d}t$.

On dit que la méthode de quadrature J est d'ordre au moins n car la formule approchée est exacte pour les polynômes de degré inférieur ou égal à n.

27. Exemple : on prend $n = 1$, $t_0 = -1$ et $t_1 = 1$. Déterminer α_0 et α_1. Expliquer à l'aide d'un graphique en prenant g positive pourquoi, dans ce cas, la méthode J s'appelle la "méthode des trapèzes".

Quadrature de Gauss

Dans les trois questions suivantes, on prend pour points d'interpolation $t_0, t_1, \dots, t_n$ les $(n+1)$ racines du polynôme de Legendre P_{n+1} introduit dans la partie 2. On fait la division euclidienne de P par P_{n+1}, on note respectivement Q le quotient et R le reste de cette division : $P = QP_{n+1} + R$

28. Démontrer que $J(QP_{n+1}) = \displaystyle\int_{-1}^{1} Q(t)P_{n+1}(t)\,\mathrm{d}t$, puis conclure que $J(P) = \displaystyle\int_{-1}^{1} P(t)\,\mathrm{d}t$.
29. Démontrer que les poids $\alpha_0, \dots, \alpha_n$ associés à la quadrature de Gauss sont strictement positifs et calculer leur somme.
30. Montrer qu'il existe un polynôme $P \in \mathbb{R}_{2n+2}[X]$ tel que $J(P) \neq \displaystyle\int_{-1}^{1} P(t)\,\mathrm{d}t$

La question 28 prouve que la méthode de Gauss est d'ordre d'au moins $2n+1$ améliorant considérablement le résultat de la question 26, tandis que le résultat de la question 30 prouve que la méthode de Gauss ne peut pas être d'ordre $2n+2$. Ainsi, la méthode de Gauss est exactement d'ordre $2n+1$.

C'est tout ? Je vais m'ennuyer moi ! ! !

Ok, ok, proposons des bonus :

31. *Informatique :* améliorons le résultat de la question 13 : créez une image .gif où on voit le phénomène de Runge s'amplifier au fur et à mesure que l'on interpole avec une subdivision ayant le plus de points.
32. *Informatique :* à la question 25, on a supposé que les polynômes ℓ_i étaient déjà définies dans une liste `L`, créer une fonction `ListeL` qui va vraiment créer cette liste `L` (attention on demande des polynômes sous forme de liste de coefficients, et non une fonction qui renvoie $\ell_i(x)$ quand x est un paramètre). Cette fonction `ListeL` prendra en argument la liste des réels x_0, x_1, …, x_n.
33. *Informatique :* notre façon de coder des polynômes sous Python est paradoxale. En effet, si $P = X^2 \in \mathbb{R}_2[X]$, alors sous Python, `P=[0,0,1]`, tandis que si $Q = X^2 \in \mathbb{R}_4[X]$, alors sous Python, `Q=[0,0,1,0,0]`. Mathématiquement, $P = Q$, bien sûr ! Cependant, si on rentre `P==Q` sous Python, il retournera `False`. Créer une fonction qui, étant donné deux polynômes rentrés sous Python, renvoie `True` si et seulement si ces deux polynômes sont égaux mathématiquement.

Indications (après avoir réfléchi sur la question)

1 Calculer $\ell_i(x_j)$ pour $(i,j) \in [\![0\,;n]\!]^2$. Montrer l'unicité du polynôme interpolateur en considérant un second polynôme interpolateur P et étudier les racines de $L_n(f) - P$.
2 On pourra créer une fonction annexe, qui calculera $\ell_i(a)$.
4 On pourra étudier le noyau de V en utilisant la question 1.
6 Rolle
7 Récurrence
8 Si $x \in \sigma$, calculer $f(x) - L_n(x)$ et $\pi_\sigma(x)$
10 $F(x) = 0$, calculer $F(x_i)$ puis appliquer la question 7.
11 Théorème des bornes atteintes, puis utiliser la question 8 pour majorer.
13 On pourra utiliser la fonction $f : x \mapsto \dfrac{1}{1+25x^2}$ sur $[\,-1\,;1\,]$, l'interpoler avec une subdivision régulière de 20 points.

14 On détaillera bien le caractère défini.
15 On commencera par "rallonger" les coefficients de P et de Q avec des 0 de sorte que la liste soit de longueur $2N+1$ avant de calculer les coefficients c_n
21 Ne pas oublier que pour tout $k \in [\![1\,;N]\!]$, $\text{vect}(1, X, \dots, X^k) = \text{vect}(P_0, P_1, \dots, P_k)$
22 Utiliser que $1 \perp P_n$. Que peut-on dire du signe de P_n sur $[\,-1\,;1\,]$ si P_n ne s'annule pas ?
23 Utiliser que $Q \in \mathbb{R}_{n-1}[x]$ et utiliser la question 21.
24 Changement de variable
26 Écrire P comme une combinaison linéaire des ℓ_i
27 Calculer ℓ_0, ℓ_1, α_0 et α_1, $J(g)$