

INTÉGRATION NUMÉRIQUE ET SÉRIES DE FOURIER

1. Objectifs

Dans ce TP, on programmera, dans un premier temps, quelques méthodes d'intégration classiques (cf. section 2) qui seront testées sur des fonctions simples et nous comparerons ces différentes méthodes en termes de précision de calculs. Parmi ces méthodes, nous présenterons la méthode de Simpson que nous utiliserons pour la deuxième partie.

Ainsi, dans la section 3, nous calculerons, grâce à cette méthode d'intégration de Simpson, les coefficients de Fourier $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}^*}$ d'une fonction f de \mathbb{R} dans \mathbb{R} et 2π -périodique, définis par

$$\left\{ \begin{array}{l} a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(t) dt, \\ \forall n \in \mathbb{N}^*, \quad a_n = \frac{1}{\pi} \int_0^{2\pi} \cos(nt) f(t) dt, \\ \forall n \in \mathbb{N}^*, \quad b_n = \frac{1}{\pi} \int_0^{2\pi} \sin(nt) f(t) dt. \end{array} \right. \quad (1)$$

Numériquement, nous étudierons l'erreur commise en remplaçant la fonction f par sa somme partielle de Fourier f_n définie par

$$\forall n \in \mathbb{N}, \quad \forall t \in \mathbb{R}, \quad f_n(t) = a_0 + \sum_{k=1}^n a_k \cos(kt) + b_k \sin(kt). \quad (2)$$

Plus précisément, en prenant des fonctions f particulières, nous constaterons expérimentalement, que l'erreur

$$\varepsilon(n) = \|f - f_n\|_\infty = \max_{t \in [0, 2\pi[} |f(t) - f_n(t)|, \quad (3)$$

tend vers zéro quand n tend vers l'infini et ce, d'autant plus rapidement que la fonction f est « régulière » (en un sens que nous préciserons).

Naturellement, les connaissances théoriques requises ont très simples (juste les définitions (1) et (2) des coefficients de Fourier d'une fonction et de sa somme partielle de Fourier) et dans ce TP, il sera laissé beaucoup de place à l'expérimentation numériques des concepts et leurs représentations graphiques, grandement facilitées par l'usage des fonctions graphiques de matlab.

Enfin, dans la section 4, la partie théorique consistera à montrer que l'erreur tend vers zéro de la façon suivante :

$$\forall n \in \mathbb{N}, \quad \max_{t \in [0, 2\pi[} |f(t) - f_n(t)| \leq \frac{M}{n^k}, \quad (4)$$

où M est une constante ne dépendant que de f et k est un entier naturel non nul, d'autant plus grand que f est « régulière ».

2. Mise en forme de méthodes d'intégration

Soient a, b deux réels tels que $a < b$, f une fonction de $[a, b]$ dans \mathbb{R} et N appartenant à \mathbb{N}^* . On considère le pas h de discrétisation défini par

$$h = \frac{b-a}{N}. \quad (5)$$

Pour calculer l'intégrale de f sur $[a, b]$, la méthode des rectangles consiste à approcher cette intégrale par

$$I_1 = h \sum_{i=0}^{N-1} f(a + ih). \quad (6)$$

La méthode du point milieu consiste à approcher cette intégrale par

$$I_2 = h \sum_{i=0}^{N-1} f(a + ih + h/2). \quad (7)$$

Le méthode des trapèzes consiste à approcher la fonction f sur chaque intervalle $[a + ih, a + (i + 1)h]$ pour $0 \leq i \leq N - 1$ par une fonction affine qui prend les mêmes valeurs que f aux points $a + ih$ et $a + (i + 1)h$ (cf. figure 1). Dans ce cas, l'intégrale est approchée par l'expression

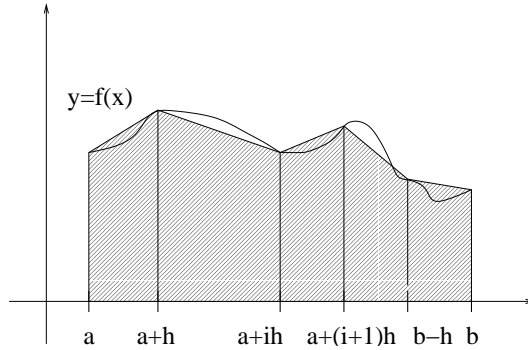


FIGURE 1. La méthode d'intégration par les rectangles

$$I_3 = \frac{h}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^{N-1} f(a + ih) \right). \quad (8)$$

Enfin, la méthode de Simpson (ou des paraboles) consiste à remplacer la fonction f par une fonction parabolique qui prend les mêmes valeurs que f aux points $a + ih$, $a + (i + 1/2)h$ et $a + (i + 1)h$. Dans ce cas, l'intégrale est approchée par l'expression

$$I_4 = \frac{h}{6} \left(f(a) + f(b) + 2 \sum_{i=1}^{N-1} f(a + ih) + 4 \sum_{i=0}^{N-1} f(a + ih + h/2) \right). \quad (9)$$

Écrire une fonction matlab qui calcule les quatre valeurs approchées I_1 , I_2 , I_3 et I_4 de la fonction sinus. Les paramètres de cette fonction seront a , b , N et un entier prenant quatre valeurs différentes

(et qui renverra, selon ces valeurs, l'une des quatre valeurs I_1 , I_2 , I_3 et I_4). On utilisera les tableaux de matlab du type $a : h : b - h$ et la fonction *sum*.

Avec matlab, si *fcn* est une chaîne représentant une fonction (built-in ou déclarée dans un fichier .m) et si X est un vecteur *feval*(*fcn*, X) renvoie un vecteur Y de même taille, dont chacune des composantes est égale à l'évaluation de la fonction *fcn* sur chacune des composantes de X . Avec cette fonction, il est possible de passer le nom de la fonction comme paramètre de la fonction qui intègre. On écrira donc une fonction matlab **intfcn**(c, N, a, b, fcn) qui renverra l'une des quatre valeurs I_1 , I_2 , I_3 et I_4 , égale à l'intégrale approchée de la fonction *fcn* sur $[a, b]$ avec N points d'intégration et c sera un entier appartenant à l'ensemble $\{1, 2, 3, 4\}$.

Testez cette fonction pour calculer les intégrales suivantes

$$\int_0^1 \cos(x) dx, \quad \int_0^1 \sqrt{x} dx \text{ et } \int_0^1 \frac{1}{1+x^2} dx, \quad (10)$$

que l'on comparera à leurs valeurs exactes (en faisant prendre à N des valeurs de plus en plus grandes).

On comparera aussi la précision de ces différentes méthodes entre elles, en calculant I_1 , I_2 , I_3 et I_4 , pour une même valeur de N . Que constatez-vous ? Quelle méthode vous semble être la plus précise ?

3. Étude des coefficients de Fourier d'une fonction périodique

Par la suite, nous intégrerons numériquement les fonctions intervenant dans le calcul des coefficients de Fourier par la méthode de Simpson.

3.1. Calcul des coefficients de Fourier et des sommes partielles

Soit maintenant une fonction f de \mathbb{R} dans \mathbb{R} et 2π -périodique. Écrire une fonction matlab qui calcule les coefficients de Fourier de f définis par (1). Cette fonction **coeffourier**(n, N) renverra, pour $(n, N) \in \mathbb{N} \times \mathbb{N}^*$, les coefficients de Fourier $(a_k)_{0 \leq k \leq n}$ et $(b_k)_{1 \leq k \leq n}$, chacun d'eux étant calculés avec la méthode d'intégration de Simpson à N points. Pouvez-vous écrire une fonction similaire **coeffourier**(n, N, fcn) où le nom de la fonction f est passé par variable (dans la chaîne de caractère *fcn*) ?

Par la suite, on voudrait pouvoir stocker sur le disque dur, sous forme de fichier .mat, ces coefficients de Fourier ; ainsi, ils seront calculés une fois pour toute (pour $n = nmax$, un « grand » nombre donné à l'avance) et pour calculer la somme partielle f_n (avec $n \leq nmax$), on n'aura plus qu'à les charger. En utilisant l'instruction *save*, écrire un script permettant de stocker ces coefficients de Fourier dans un fichier .mat.

Écrire une fonction matlab **sommefourier**($t, n, nmax, a0, A, B, fcn$) qui calcule $f_n(t)$ (défini par (2)) où t est un réel, n et $nmax$ sont deux entiers tels que $n \leq nmax$, a_0 est un réel correspondant au premier de coefficient de Fourier de f , A et B sont deux tableaux de réels de taille $(1, nmax)$, représentant les coefficients de Fourier $(a_k)_{1 \leq k \leq nmax}$ et $(b_k)_{1 \leq k \leq nmax}$, et *fcn* une chaîne de caractère contenant le nom de la fonction f .

Ainsi, les coefficients de Fourier étant stockés définitivement sur disque dur, on les récupère en utilisant l'instruction *load* ; grâce à la fonction **sommefourier**($t, n, nmax, a0, A, B, fcn$) définie ci-dessus, on peut donc calculer la somme partielle f_n .

On calculera et stockera sur disque dur les coefficients de Fourier des fonctions E , C , B_1 , B_2 , B_3 , B_4 , 2π périodiques et définies respectivement par, pour tout $x \in [0, 2\pi[$

$$E(x) = \begin{cases} 1 & \text{si } x \leq \pi, \\ -1 & \text{si } x > \pi, \end{cases} \quad (11a)$$

$$C(x) = \begin{cases} x/\pi & \text{si } x \leq \pi, \\ 2 - x/\pi & \text{si } x > \pi, \end{cases} \quad (11b)$$

$$B_1(x) = \left(\frac{x}{2\pi}\right) - \frac{1}{2}, \quad (11c)$$

$$B_2(x) = \left(\frac{x}{2\pi}\right)^2 - \left(\frac{x}{2\pi}\right) + \frac{1}{6}, \quad (11d)$$

$$B_3(x) = \left(\frac{x}{2\pi}\right)^3 - \frac{3}{2}\left(\frac{x}{2\pi}\right)^2 + \frac{1}{2}\left(\frac{x}{2\pi}\right), \quad (11e)$$

$$B_4(x) = \left(\frac{x}{2\pi}\right)^4 - 2\left(\frac{x}{2\pi}\right)^3 + \left(\frac{x}{2\pi}\right)^2 - \frac{1}{30}. \quad (11f)$$

3.2. Représentation graphique de f et de f_n

Écrire un script qui trace, sur une même figure, les fonctions f et f_n sur l'intervalle $[0, 2\pi]$, pour différentes valeurs de n . Exécutez ce script pour chacune des six fonctions définies par (11).

Prendre pour n des valeurs de plus en plus grandes. Que constate-t-on graphiquement.

Quel phénomène présentent les fonction E et B_1 que ne présentent pas les autres fonctions définies par (11) ; on parle de phénomène de Gibbs. Quelle est la particularité de E et B_1 par rapport aux autres fonctions de (11).

Quelle propriété les fonctions B_i ont elles l'air de présenter (par rapport à l'approximation de f par f_n) ?

3.3. Étude expérimentale de la décroissance vers zéro des coefficients de Fourier

Si la fonction f possède certaines hypothèses de régularité (cf. partie théorique 4), alors la limites des coefficients de Fourier a_n et b_n quand n tend vers l'infini est égale à zéro. Pour en mesurer l'ordre de convergence, on considère, le cas échéant, le réel α défini par

$$\forall n \in \mathbb{N}^*, \quad |a_n| \approx \frac{M}{n^\alpha} \text{ et } |b_n| \approx \frac{M'}{n^\alpha}, \quad (12)$$

où M et M' sont deux constantes qui ne dépendent que de la fonction f . Montrer que si les points définis par $(\log_{10}(n), \log_{10}(|a_n|))$, et $(\log_{10}(n), \log_{10}(|b_n|))$ (avec n décrivant $\{1, \dots, n_{max}\}$) décrivent une droite, la mesure de la pente de cette droite permet de déterminer l'ordre α .

Estimez cet ordre pour chacune des six fonctions définie par (11). Que constatez vous ?

3.4. Étude expérimentale de l'erreur $\|f - f_n\|_\infty$ - *Partie facultative*

La fonction `fmin` de matlab est très pratique : elle détermine le minimum d'une fonction f : si le nom de la fonction f est stocké dans la chaîne `fcn`, `fmin(fcn, c, d, [1 tol])` calcule le minimum de la fonction f dans l'intervalle $[c, d]$ avec la tolérance `tol`. En remarquant que

$$\|f - f_n\|_\infty = \max_{t \in [0, 2\pi[} |f(t) - f_n(t)| = - \min_{t \in [0, 2\pi[} (-|f(t) - f_n(t)|),$$

écrire une fonction matlab, qui, pour f et n donnés, calcule $\|f - f_n\|_\infty$. Comme dans la section 3.3, on cherche α l'ordre de convergence de f_n vers f défini par

$$\forall n \in \mathbb{N}^*, \quad \varepsilon(f, n) = \|f - f_n\|_\infty \approx \frac{M}{n^\alpha}. \quad (13)$$

En traçant de même l'ensemble des valeurs $(\log_{10}(n), \log_{10}(\varepsilon(f, n)))$, (n décrit $\{1, \dots, n_{max}\}$), on est capable d'estimer l'ordre de convergence α .

Calculer α pour chacune des six fonctions définie par (11). Que constatez vous ?

4. Majoration théorique de l'erreur

4.1. Étude théorique de la décroissance vers zéro des coefficients de Fourier

Soit f une fonction de \mathbb{R} dans \mathbb{R} et 2π -périodique. Citez une condition suffisante sur f pour que

$$\lim_{n \rightarrow +\infty} a_n = 0 \text{ et } \lim_{n \rightarrow +\infty} b_n = 0.$$

Démontrez le.

On cherche maintenant à démontrer le résultat constaté expérimentalement par (12). Soient f une fonction de \mathbb{R} dans \mathbb{R} 2π périodique et k un entier naturel. Montrez les deux propriétés suivantes :

Proposition 4.1. *La fonction f est de classe \mathcal{C}^k sur \mathbb{R} si et seulement si elle vérifie les deux assertions suivantes :*

- 1) *f est de classe \mathcal{C}^k sur $[0, 2\pi[$.*
- 2) *Pour tout $l \in \{0, \dots, k\}$, $f^{(l)}(2\pi - 0)$ existent et sont égales à $f^{(l)}(0 + 0)$, c'est-à-dire, les valeurs de f et de ses k premières dérivées coïncident aux extrémités du segment $[0, 2\pi]$.*

Proposition 4.2. *Si f est de classe \mathcal{C}^k sur \mathbb{R} , alors, pour $n \rightarrow +\infty$,*

$$a_n = o\left(\frac{1}{n^k}\right) \text{ et } b_n = o\left(\frac{1}{n^k}\right). \quad (14)$$

Si, de plus, $f^{(k+1)}$ est continue par morceau et dérivable par morceaux sur $[0, 2\pi[$, alors on a

$$a_n = \mathcal{O}\left(\frac{1}{n^{k+2}}\right) \text{ et } b_n = \mathcal{O}\left(\frac{1}{n^{k+2}}\right). \quad (15)$$

On pourra faire k intégrations par partie successives. On rappelle que, pour deux suites réelles $(u_n)_{n \in \mathbb{N}}$ et $(v_n)_{n \in \mathbb{N}}$,

$$u_n = \mathcal{O}(v_n)$$

est équivalent à :

$$\exists M \in \mathbb{R}^+, \quad \forall n \in \mathbb{N}, \quad |u_n| \leq M |v_n|.$$

La constante M est indépendante de n .

Ces résultats semblent ils être en cohérence avec ce que vous avez constatés dans la section 3.3 ? De quelles classes sont les fonctions E , C , B_1 , B_2 , B_3 et B_4 , définies par (11).

4.2. Étude théorique de la décroissance vers zéro de l'erreur $\|f - f_n\|_\infty$

On a le résultat suivant :

Proposition 4.3. *Si f est de classe \mathcal{C}^k sur \mathbb{R} , et si $f^{(k+1)}$ est continue par morceau et dérivable par morceaux sur $[0, 2\pi[$, alors*

$$\|f - f_n\|_\infty = \mathcal{O}\left(\frac{1}{(k+1)n^{k+1}}\right). \quad (16)$$

On démontrera ce résultat en justifiant et utilisant l'égalité suivante :

$$\forall n \in \mathbb{N}^*, \quad \forall t \in [0, 2\pi[, \quad f(t) - f_n(t) = \sum_{p=n+1}^{+\infty} a_p \cos(pt) + b_p \sin(pt).$$

Ce résultat vous semble-t-il être corroboré par les résultats numériques de la section 3.4 ?