TD n° 4 : Méthodes d'intégration numérique

Il existe de nombreuses méthodes d'intégration numérique. Dans un but purement pédagogique, nous allons utiliser trois méthodes. La première s'appelle méthode des trapèzes et repose sur des considérations géométriques. Cette première technique se révélant très vite insuffisante lorsqu'il s'agit d'intégrer des fonctions fortement variables, nous proposons d'étudier une deuxième méthode : il s'agit de la méthode d'intégration dite de Simpson qui est, par rapport à la méthode des trapèzes, le degré supérieur de sophistication dans l'intégration numérique. Nous finirons par une implémentation très simple de la méthode de Monte Carlo qui est parfois utilisée en Physique pour des intégrations de fonctions oscillantes dans des espaces multi-dimensionnels.

1 Méthode des trapèzes

Considérons une fonction $f(x)$ continue sur un intervalle $[a, b]$ de $\mathbb{R}$. L'intégrale de $f(x)$ entre $x = a$ et $x = b$ est approximée par :

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} A_i \tag{1}$$

où $A_i = \dfrac{f(x_{i-1}) + f(x_i)}{2}(x_i - x_{i-1})$, est l'aire du trapèze hachuré représentée sur la figure ci–dessous.

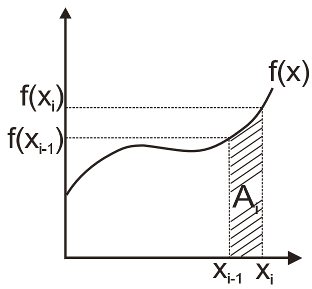

Figure 1 – Illustration de la méthode des trapèzes. Voir page 41 du polycopié de cours de Python pour deux autres schémas d'intégration.

Cette méthode revient à remplacer dans l'intervalle $[x_{i-1}, x_i]$, la courbe f(x) par un segment de droite joignant $f(x_{i-1}) = f_{i-1}$ et $f(x_i) = f_i$ et à calculer l'aire du trapèze de la figure 1. Pour calculer l'intégrale entre a et b, on somme les aires A_i de tous les trapèzes. L'intervalle d'intégration $[a, b]$ est divisé en N segments réguliers :

$$h = x_i - x_{i-1} = \frac{b-a}{N}.$$

La valeur de l'intégrale de $f(x)$ sur l'intervalle $[a, b]$ s'obtient par :

$$\boxed{\int_a^b f(x)dx \simeq \frac{h}{2}\left[f_0 + f_N + 2\sum_{i=1}^{N-1} f_i\right].} \tag{2}$$

2 Méthode de Simpson

On cherche à obtenir la valeur de l'intégrale : $I = \displaystyle\int_a^b f(x)dx$ par la méthode de Simpson. Pour cela on utilise une partition de points équidistants de l'intervalle d'intégration $[a, b]$: $x_i = a + ih$, où

$0 \leq i \leq N$ où N doit être pair pour cette méthode. Montrons que si f_i est la valeur de $f(x_i)$, on a :

$$A_i = \int_{x_{i-1}}^{x_{i+1}} f(x)dx = h\left(\frac{1}{3}f_{i-1} + \frac{4}{3}f_i + \frac{1}{3}f_{i+1}\right) + O(h^5 f^{(4)}).$$

(On remarque que l'intervalle d'intégration est de largeur $2h$.) Pour obtenir ce résultat, on effectue deux développements limités de la primitive F de f évaluée respectivement en x_{i+1} et en x_{i-1} [1] :

$$\begin{aligned} F_{i+1} &= F_i + hF_i^{'} + \frac{h^2}{2}F_i^{''} + \frac{h^3}{6}F_i^{(3)} + \frac{h^4}{4!}F_i^{(4)} + O(h^5 F^{(5)}), \\ F_{i-1} &= F_i - hF_i^{'} + \frac{h^2}{2}F_i^{''} - \frac{h^3}{6}F_i^{(3)} + \frac{h^4}{4!}F_i^{(4)} + O(h^5 F^{(5)}). \end{aligned}$$

En soustrayant la deuxième ligne à la première, il vient :

$$A_i = F_{i+1} - F_{i-1} = 2hF_i^{'} + 2\frac{h^3}{6}F_i^{(3)} + O(h^5 F^{(5)}).$$

En revenant de F à f et en utilisant le fait que :

$$F_i^{(3)} = f_i^{''} = \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2} + O(h^2 f^{(4)}),$$

on retrouve bien le résultat annoncé, soit :

$$A_i = h\left(\frac{1}{3}f_{i-1} + \frac{4}{3}f_i + \frac{1}{3}f_{i+1}\right) + O(h^5 f^{(4)}).$$

On déduit la valeur de I en sommant tous les termes A_i, avec i impair (et donc N doit être pair) car A_i correspond à l'intégration de $f(x)$ sur un intervalle de largeur $2h$, soit :

$$\boxed{I = \tfrac{h}{3} \textstyle\sum_{i\ impair}^{N-1} \left[f_{i-1} + 4f_i + f_{i+1}\right].} \tag{3}$$

Commentaires : cette méthode n'est rien d'autre que la généralisation de la méthode des trapèzes, en interpolant la fonction f, non pas par une droite, mais par un arc de parabole passant par trois points successifs de la courbe. Elle est donc exacte si f est un polynôme de degré ≤ 2.

3 Programmation et tests des deux méthodes

La fonction sinus cardinal,

$$f(x) = \frac{\sin x}{x} \text{ avec } f(0) = 1 \tag{4}$$

représente l'amplitude du motif de diffraction à travers une fente. On cherche à calculer la valeur I de l'intégrale de f entre a et b, soit,

$$\boxed{I = \int_a^b \frac{\sin x}{x}dx.} \tag{5}$$

Pour cela on utilise successivement les méthodes des trapèzes et de Simpson. Pour les deux méthodes, on prend $a = 0$ et $b = 2\pi$.

1. On passe sous silence les conditions de dérivabilité requises pour f.

3.1 Cahier des charges (consignes générales pour l'ensemble des questions qui suivront)

On chargera le module `numpy`. Vous devrez respecter les consignes suivantes :

1. Dans la fonction principale, on stockera les abscisses des noeuds x_i ($0 \leq i \leq n$) dans un tableau `x` de type `numpy.ndarrray` et on stockera les valeurs de $f(a + ih)$ ($0 \leq i \leq n$) dans un tableau `y` de type `numpy.ndarrray` dont la dimension est fixée à n=`10` pour commencer.
2. Vous devrez créer au moins 4 fonctions :
   (a) la fonction `poly2` qui renvoie la valeur d'un polynôme d'ordre 2. Son argument sera le tableau `x`.
   (b) La fonction `sin_card` qui renvoie la valeur de $f(x)$ (cf. Eq. (4)). Son argument sera le tableau `x`.
   (c) La fonction `trapeze` qui renvoie la valeur estimée de l'intégrale de f entre `a` et `b` en utilisant la méthode des trapèzes avec `n` intervalles (cf. équation 2). Elle doit avoir deux arguments : `x` et `y`.
   (d) La fonction `simpson` fera de même en utilisant la méthode de Simpson avec `n` intervalles (cf. équation (3)).
3. On appellera `It` et `Is` les valeurs de l'intégrale obtenues respectivement par la méthode des trapèzes et par celle de Simpson.
4. Pour étudier la convergence de `It` et de `Is` en fonction du nombre d'intervalles `n` entre `a` et `b`, on fait varier `n` de 20 à 1000 par pas de 20. Les valeurs des intégrales en fonction de `n` sont stockées dans des tableaux `It[50]` et `Is[50]`. Écrire les résultats dans un fichier «`resultat.dat`» sous la forme d'un tableau à 3 colonnes : `n`, `It[n/20-1]` et `Is[n/20-1]`. Tracer les courbes correspondantes et sauver la figure.

3.2 Mise au point et test (les questions)

3.2.1 Avec un polynôme d'ordre ≤ 2

Utiliser la fonction `poly2` pour mettre au point et tester vos deux fonctions `trapeze` et `simpson`. Faire afficher le résultat de l'intégration analytique de $x^2/3$ entre $a = 0$ et $b = 2$. Comparer le avec les résultats fournis par `trapeze` et `simpson` pour 2, 4, et 10 intervalles pour ces bornes d'intégration. Commenter.

3.3 Exploitation

3.3.1 Etude de la convergence numérique

On intégre maintant la fonction sinus cardinale codée dans `sin_card`. Etudiez la convergence des deux méthodes par rapport au nombre `n` d'intervalles de longueur `h` entre `a=0` et `b=`2π. Tracer les courbes `It=f(n)` et `Is=f(n)`. Conclure sur la vitesse de convergence de chacune des deux méthodes d'intégration.

3.3.2 Application à l'étude du rayonnement solaire

Dans cette application, on va calculer la puissance totale issue du rayonnement solaire reçue en moyenne par mètre carré aux latitudes de 40^o Nord de la terre [2]. (voir la figure 2).

2. Les données originales sont issues du site internet officiel `https://www.nrel.gov/grid/solar-resource/spectra.html`

1. On dispose de l'éclairement "spectral" reçu en watt par mètre carrés par nanomètres dans le fichier `ASTMG173_Interpolated.dat`. Les données sont réparties à intervalle régulier de longueurs d'onde exprimées en nanomètres. Par intégration numérique, calculer la puissance totale issue du rayonnement solaire reçue.
2. On veut faire de même à partir maintenant du fichier `ASTMG173.dat` où les données ne sont plus réparties à intervalle régulier de longueurs d'onde. Adaptez la formule des trapèzes pour pouvoir traiter ce cas. Puis, calculer de nouveau la puissance totale issue du rayonnement solaire reçue sur terre.

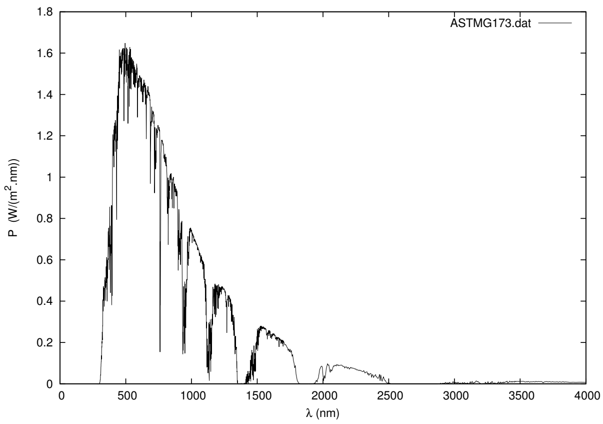

FIGURE 2 – Spectre solaire en watt par mètre carrés par nanomètres.

4 Méthode de Monte Carlo

Cette méthode d'intégration numérique est dans un certain sens très similaire à la méthode des rectangles mais les points d'échantillonnage de la fonction à intégrer sont choisis au hasard au lieu d'être équi-répartis. Nous nous intéresserons uniquement au cas où l'échantillonnage est effectué selon une loi de probabilité uniforme sur l'intervalle d'intégration, des techniques de Monte Carlo plus élaborées et plus performantes s'affranchissent de cette caractéristique.
Il peut être démontré que la limite de l'intégrale calculée numériquement par la méthode de Monte Carlo tend vers la valeur théorique de l'intégrale quand le nombre de points d'échantillonnage tend vers l'infini. Rajouter à votre programme :

1. La fonction `monte-carlo` qui renvoie la valeur estimée de l'intégrale de f entre `a` et `b` en utilisant la méthodes des trapèzes avec `n` intervalles (cf. équation (1)). Elle doit avoir deux arguments : `x` et `y`. Les positions des points d'échantillonnage seront calculés dans la fonction principale juste avant l'appel de la fonction `monte-carlo`. On utilisera la fonction `numpy.random.random` pour générer les nombres (pseudo)-aléatoires.
2. L'étude de la convergence en fonction de n (voir la section 3 pour les intégrales : $\int_0^2 x^2/3\,dx$ et $\int_0^{2\pi} f(x)\,dx$ et tracer les courbes correspondantes. Qu'en concluez-vous ?

5 Utilisation des fonctions d'intégration incluses dans `Numpy` et `Scipy`

1. En utilisant la documentation en ligne, choisir et tester trois d'entre elles sur $\int_0^{2\pi} f(x)\,dx$, et vérifier vos résultats.