

TP n° 7 : Calcul approché d'intégrales

Nous étudions dans ce TP plusieurs méthodes de calcul approché d'intégrales.

Exercice 1 – Calculs approchés d'intégrales par des méthodes élémentaires

1. Nous avons étudié dans le cours la méthode des rectangles, la méthode du point milieu, la méthode des trapèzes et la méthode de Simpson. Implémenter chacune de ces 4 méthodes. Elles prennent en paramètre la fonction à intégrer, les bornes de l'intervalle et le nombre de subdivisions.
2. La fonction `quad` du module `scipy.integrate` permet le calcul approché de l'intégrale d'une fonction entre deux bornes. Dans son utilisation la plus simple, elle prend 3 paramètres (fonction, borne inférieure, borne supérieure), et renvoie un tuple (valeur approchée de l'intégrale, erreur). Par comparaison à la valeur approchée renvoyée par `quad` et en tenant compte de l'erreur associée, déterminer combien de subdivisions sont nécessaires pour obtenir la valeur de $\int_0^1 e^{-t^2/2} dt$ à 10^{-4} près pour chacune des méthodes ci-dessous. À quel rang aller pour avoir une approximation à 10^{-8} et 10^{-10} pour les trois dernières ? Pousser jusqu'à 10^{-13} pour la méthode de Simpson.
3. Écrire une fonction `Phi` effectuant le calcul approché de $\Phi(x) = \int_0^x e^{-t^2/2} dt$ par la méthode de Simpson. On utilisera une subdivision en 100 intervalles. Que représente (à une constante multiplicative et additive près) cette fonction Φ ?
4. Tracer la courbe de Φ sur l'intervalle $[-2, 2]$. Utiliser pour cela la fonction `plot` du module `matplotlib.pyplot`. Cette fonction prend en paramètres deux listes $[a_1, a_2, \dots, a_n]$ et $[b_1, b_2, \dots, b_n]$ et traçant la ligne brisée reliant les points de coordonnées (a_i, b_i) .
5. Calculer numériquement une valeur approchée de $\int_{-\infty}^{+\infty} e^{-t^2/2} dt$. Comparer à la valeur théorique.

Exercice 2 – Méthode de Monte-Carlo

La méthode de Monte-Carlo pour le calcul de $I = \int_a^b f(t) dt$ consiste à tirer aléatoirement un grand nombre de réels $a_i \in [a, b]$ de façon uniforme, $i \in \llbracket 1, n \rrbracket$, et de faire la moyenne M_n des $f(a_i)$. Le second théorème de transfert permet d'affirmer que $E(M_n) = \frac{I}{b-a}$. La loi des grands nombres nous assure que plus n est grand, plus la probabilité que M_n soit une bonne approximation de $E(M_n)$ est forte. On obtient une méthode probabiliste de calcul d'une intégrale.

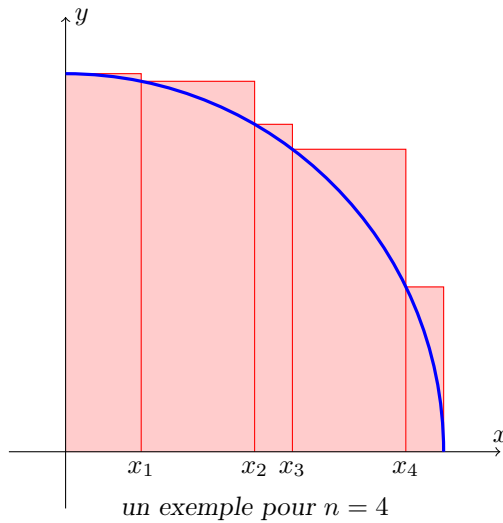
1. En déterminant la valeur de cette intégrale avec la fonction `quad` de `scipy.integrate`, déterminer le nombre moyen de tirages nécessaires pour obtenir une valeur approchée de $\int_0^1 e^{-t^2} dt$ à 10^{-6} près.
2. En généralisant cette méthode au cas d'une intégrale double, et en considérant sur $[0, 1]^2$ la fonction qui à (x, y) associe 1 si $x^2 + y^2 \leq 1$ et 0 sinon, déterminer une valeur approchée de π . La méthode fournit-elle une bonne valeur de π ?
3. Déterminer une valeur approchée du volume d'une boule de rayon 1. Comparer à la valeur théorique.

Exercice 3 –

On cherche à obtenir une approximation de π à l'aide de la méthode des rectangles. On considère donc la fonction

$f : x \mapsto \sqrt{1-x^2}$ qui vérifie : $\int_0^1 f(x) = \frac{\pi}{4}$.

On choisit ensuite une subdivision $(x_0 = 0, x_1, \dots, x_{n+1} = 1)$ *non nécessairement de pas régulier* et on calcule l'approximation par excès de $\frac{\pi}{4}$ qu'on obtient en appliquant la méthode des rectangles :



$$\frac{\pi}{4} \approx \sum_{k=0}^n (x_{k+1} - x_k) f(x_k)$$

Déterminez la meilleure approximation de π que l'on puisse obtenir lorsque $n = 200$. On donnera le résultat avec 10 chiffres significatifs.

Indication :

- On rappelle que si la fonction de n variables $F : (x_1, \dots, x_n) \mapsto \sum_{k=0}^n (x_{k+1} - x_k) f(x_k)$ admet un minimum en $A = (a_1, \dots, a_n)$, alors $\nabla F(A) = 0$, où ∇F est le vecteur gradient, constitué des dérivées partielles de F par rapport à chacune des variables x_i .
- On pourra montrer que F atteint un minimum en un point $A = (a_1, \dots, a_n)$ dont les coordonnées peuvent être déterminées de proche en proche en fonction de a_1 , puis minimiser la fonction en a_1 ainsi obtenue.
- Effectuer cette minimisation par un balayage de plus en plus fin au fur et à mesure qu'on s'approche de la valeur recherchée.