

Université d'Évry Paris-Saclay

Faculté des Sciences

2ème année de Licence en Mathématiques



Diverses méthodes d'intégration

Réalisé par :
SAMER Ahmed Wahid

Année académique : 2023-2024

Table des matières

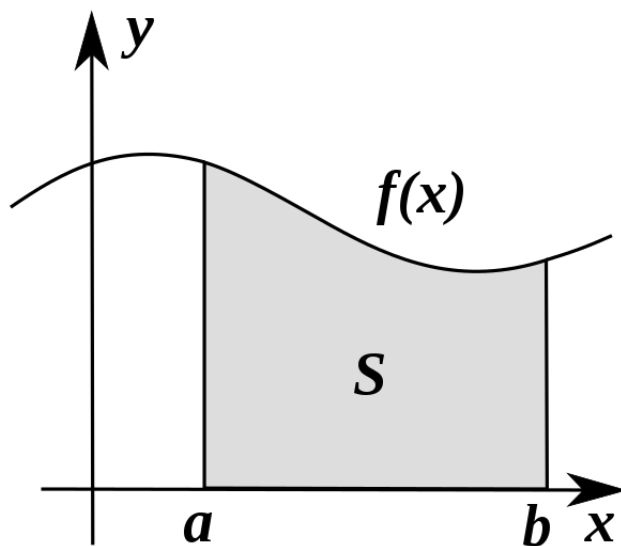
1	Introduction	2
2	La Méthode de Simpson	3
2.1	Procédure Détail de la Méthode	3
2.2	Somme des Contributions	3
2.3	Efficacité et Erreur	3
2.4	Mise en œuvre et exemples	4
3	La Méthode de Monte Carlo	5
3.1	Implémentation de la Procédure	5
3.2	Analyse de la Convergence	5
3.3	Discussion sur la Variabilité	5
3.4	Mise en œuvre et exemples	5
4	La Méthode des Rectangles	6
4.1	Schématisation de la Procédure	6
4.2	Exploration de Différentes Variantes	6
4.3	Évaluation de la Précision et de l'Erreur	6
4.4	Mise en œuvre et exemples	7
5	La Méthode des Trapèzes	7
5.1	Formalisation de la Méthode	7
5.2	Élaboration et Justification	7
5.3	Évaluation de l'Erreur	8
5.4	Mise en œuvre et exemples	8
6	La Méthode de Gauss	9
6.1	Formulation de la Méthode	9
6.2	Analyse d'Erreur	9
6.3	Mise en œuvre et exemples	9
7	La Méthode d'Euler	10
7.1	Formulation de la Méthode	10
7.2	Démonstration du Théorème	10
7.3	Analyse d'Erreur	10
7.4	Mise en œuvre et exemples	11
8	Références	12

1 Introduction

L'intégration est une des pierres angulaires de l'analyse mathématique, jouant un rôle crucial dans divers domaines tels que la physique, l'ingénierie, la statistique et l'économie. Elle permet de déterminer des aires sous des courbes, des volumes, des valeurs centrales et d'autres quantités importantes. Cependant, à l'exception de quelques fonctions simples, la plupart des intégrales ne peuvent pas être calculées exactement et nécessitent des méthodes d'approximation.

Dans ce contexte, plusieurs méthodes d'approximation ont été développées pour estimer les valeurs des intégrales. Ces méthodes, bien que basées sur des principes mathématiques variés, visent toutes à fournir une valeur approchée qui se rapproche le plus possible de la vraie valeur de l'intégrale. Parmi ces techniques, nous explorerons principalement la méthode des rectangles, la méthode des trapèzes, la méthode de Simpson, la méthode de Monte-Carlo, La méthode de Gauss ainsi que la méthode d'euler.

Ce travail vise à établir une comparaison critique entre ces différentes stratégies d'intégration. Nous développerons et présenterons des scripts Python détaillés pour chacune de ces méthodes, les appliquant à des intégrales de diverses dimensions. Notre objectif est de déterminer leur efficacité et précision relative, soulignant spécifiquement l'ampleur des erreurs d'approximation associées. Ce faisant, nous aspirons non seulement à éclairer les nuances théoriques mais aussi à offrir une perspective pratique sur l'application efficace de ces méthodes d'intégration numérique.



source : https://fr.wikiversity.org/wiki/Approche_th%C3%A9orique_du_calcul_int%C3%A9gral

2 La Méthode de Simpson

La méthode de Simpson repose sur l'utilisation de polynômes du second degré pour approximer la fonction f sur chaque sous-intervalle de l'intervalle d'intégration global $[a, b]$. En ajustant ces polynômes pour passer par trois points clés sur chaque segment, la méthode permet une estimation précise de l'intégrale, même avec un nombre réduit de subdivisions.

2.1 Procédure Détail de la Méthode

Pour appliquer la méthode de Simpson :

- Division de l'intervalle : Divisons l'intervalle $[a, b]$ en n sous-intervalles égaux, où n doit être un nombre pair. La largeur de chaque sous-intervalle est $h = \frac{b-a}{n}$.
- Points de Calcul : Pour chaque paire de sous-intervalles, nous utiliseront les points finaux et le point médian pour définir une parabole. Ces points sont x_0, x_1, \dots, x_n où $x_i = a + i \cdot h$.
- Application de la Formule : L'estimation de l'intégrale sur chaque paire de sous-intervalles $[x_{2i}, x_{2i+2}]$ est donnée par :

$$\int_{x_{2i}}^{x_{2i+2}} f(x) dx \approx \frac{h}{3} (f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2}))$$

où x_{2i+1} est le point médian.

2.2 Somme des Contributions

L'approximation totale de l'intégrale sur $[a, b]$ est alors la somme des contributions de toutes les paires de sous-intervalles :

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 4f(x_{n-1}) + f(x_n)]$$

2.3 Efficacité et Erreur

L'erreur dans la méthode de Simpson est proportionnelle à la quatrième puissance de la largeur des sous-intervalles h , ce qui en fait une méthode d'ordre élevé. Plus précisément, La méthode est d'ordre $N = 3$. On a :

$$\begin{aligned} K_3(t) &= \int_a^b (x-t)_+^3 dx - (b-a) \left(\frac{1}{6}(a-t)^3 + \frac{2}{3} \left(\frac{a+b}{2} - t \right)^3 + \frac{1}{6}(b-t)^3 \right) \\ &= \frac{(b-t)^4}{4} - (b-a) \left(\frac{2}{3} \left(\frac{a+b}{2} - t \right)^3 + \frac{1}{6}(b-t)^3 \right) \\ &= \begin{cases} \frac{(t-a)^3(3t-2b-a)}{12} & \text{si } a \leq t < \frac{a+b}{2}, \\ \frac{(b-t)^3(b+2a-3t)}{12} & \text{si } \frac{a+b}{2} \leq t \leq b, \end{cases} \end{aligned}$$

qui est de signe constant négatif sur $[a, b]$, en vertu des inégalités :

$$\frac{2a+b}{3} \leq \frac{a+b}{2} \leq \frac{a+2b}{3}.$$

On a donc la formule d'erreur suivante pour une fonction f de classe C^2 :

$$E(f) = \frac{f^{(4)}(c)}{6} \int_a^b K_3(t) dt = \frac{(b-a)^5}{2880} f^{(4)}(c)$$

Par conséquent,

$$|E(f)| \leq \frac{(b-a)^5}{2880} \|f^{(4)}\|_{\infty}.$$

Considérons maintenant une méthode de Simpson composée sur $[a, b]$, dans laquelle tous les segments $[a_i, a_{i+1}]$ ont même longueur $\frac{b-a}{p}$. Les p erreurs de quadratures élémentaires s'additionnent, et l'on trouve de la même manière que dans le cas du point milieu :

$$E(f) = \frac{(b-a)^5}{2880p^2} f^{(4)}(c).$$

2.4 Mise en œuvre et exemples

La mise en œuvre de la Méthode de Simpson en Python peut être réalisée en suivant ces étapes :

```

1 def simpson(f, a, b, n):
2     if n % 2 != 0:
3         raise ValueError("Le nombre n'est pas pair.")
4
5     h = (b - a) / n
6     x = [a + i * h for i in range(n+1)]
7     y = [f(x[i]) for i in range(n+1)]
8
9     integral = y[0] + y[-1]
10
11     for i in range(1, n, 2):
12         integral += 4 * y[i]
13
14     for i in range(2, n-1, 2):
15         integral += 2 * y[i]
16
17     integral *= h / 3
18     return integral

```

3 La Méthode de Monte Carlo

le principe de la Méthode de Monte Carlo est de l'échantillonnage aléatoire pour estimer l'espérance mathématique d'une fonction, qui est directement reliée à l'intégrale de la fonction sur un intervalle ou un domaine donné. L'idée est de tirer aléatoirement des échantillons et d'utiliser la moyenne des évaluations de la fonction sur ces échantillons pour approximer l'intégrale.

3.1 Implémentation de la Procédure

Supposons que l'on souhaite estimer l'intégrale de la fonction f sur un intervalle $[a, b]$. On procède comme suit :

- Génération de Points Aléatoires : On génère N points aléatoires x_i uniformément répartis dans l'intervalle.
- Évaluation de la Fonction : On calcule $f(x_i)$ pour chaque point x_i .
- Calcul de la Moyenne : On calcule la moyenne des valeurs obtenues : $\bar{f} = \frac{1}{N} \sum_{i=1}^N f(x_i)$.
- Estimation de l'Intégrale : L'estimation de l'intégrale est donnée par $(b - a) \cdot \bar{f}$.

3.2 Analyse de la Convergence

L'erreur de l'estimation par la méthode de Monte Carlo diminue en $O(N^{-1/2})$, où N est le nombre de points échantillonnés. Cette convergence est garantie par la loi des grands nombres et le théorème central limite, indépendamment de la dimension de l'intégrale.

3.3 Discussion sur la Variabilité

L'erreur statistique associée à la méthode de Monte Carlo est proportionnelle à la variance de $f(x)$ sur l'intervalle considéré. Pour réduire cette variance, et donc l'erreur, on peut utiliser des techniques comme le "stratified sampling" ou le "importance sampling", qui visent à améliorer l'efficacité de l'échantillonnage.

3.4 Mise en œuvre et exemples

La mise en œuvre de la méthode de Monte-Carlo en Python peut être réalisée en suivant ces étapes :

```
1 import random
2 def monte_carlo(f, a, b, n):
3     t = 0
4     for _ in range(n):
5         x = random.uniform(a, b)
6         t += f(x)
7     carlo = t / n
8     return (b - a) * carlo
```

4 La Méthode des Rectangles

Cette technique approche l'intégrale d'une fonction continue en subdivisant de parties égales, ce qui génère de multiples rectangles dont l'aire combinée sert d'approximation à l'intégrale de la fonction.

4.1 Schématisation de la Procédure

L'intervalle $[a, b]$ est divisé en n sous-intervalles de largeur $h = \frac{b-a}{n}$. Pour chaque sous-intervalle $[x_{i-1}, x_i]$, on considère un rectangle avec une base h et une hauteur $f(x_i)$ où x_i peut être le point gauche, le point droit ou tout autre point intermédiaire du sous-intervalle, selon la variante de la méthode choisie (point milieu, point gauche, point droit).

L'estimation de l'intégrale devient alors la somme des aires de ces rectangles :

$$\int_a^b f(x) dx \approx h(f(x_0) + f(x_1) + \cdots + f(x_{n-1}))$$

4.2 Exploration de Différentes Variantes

Méthode du Point Gauche : Chaque x_i est le point gauche du sous-intervalle, donc $x_i = x_{i-1}$.

Méthode du Point Droit : Chaque x_i est le point droit, donc $x_i = x_i$.

Méthode du Point Milieu : Chaque x_i est le milieu du sous-intervalle, donc $x_i = \frac{x_{i-1} + x_i}{2}$.

4.3 Évaluation de la Précision et de l'Erreur

La précision de cette méthode dépend grandement de la valeur de n , le nombre de subdivisions. Une valeur plus élevée de n réduit la largeur h des rectangles et augmente généralement la précision de l'approximation. L'erreur de cette méthode pour une fonction continûment dérivable est proportionnelle à la première puissance de h , et peut être explicitée par une expression impliquant la dérivée de la fonction sur l'intervalle : La méthode est d'ordre $N = 1$.

Le noyau de Peano est donné par la formule :

$$\begin{aligned} K_1(t) &= \int_a^b (x-t)_+ dx - \left(\frac{b-a}{2} - \left(\frac{a+b}{2} - t \right) \right) \\ &= \frac{(b-t)^2}{2} - \frac{b-a}{2} + \left(\frac{a+b}{2} - t \right)_+ \\ &= \begin{cases} \frac{(t-a)^2}{2} & \text{si } a \leq t < \frac{a+b}{2}, \\ \frac{(b-t)^2}{2} & \text{si } \frac{a+b}{2} \leq t \leq b, \end{cases} \end{aligned}$$

car

$$\int_a^b (x-t)_+ dx = \frac{(b-t)^2}{2}.$$

Le noyau de Peano est donc de signe constant positif sur $[a, b]$, d'où la formule d'erreur pour une fonction f de classe C^2 :

$$E(f) = f''(c) \int_a^b K_1(t) dt$$

$$\begin{aligned}
&= f''(c) \left(\int_a^{\frac{a+b}{2}} \frac{(t-a)^2}{2} dt + f''(c) \int_{\frac{a+b}{2}}^b \frac{(b-t)^2}{2} dt \right) \\
&= \frac{(b-a)^3}{24} f''(c).
\end{aligned}$$

4.4 Mise en œuvre et exemples

En pratique, la méthode des rectangles peut être implémentée de manière simple en Python. Voici un exemple d'algorithme :

```

1 def rectangle(f, a, b, n):
2     h = (b - a) / n
3     integral = 0
4     for i in range(n):
5         x = a + i * h
6         integral += f(x)
7     integral *= h
8     return integral

```

5 La Méthode des Trapèzes

Lorsqu'on souhaite estimer l'intégrale d'une fonction la méthode des trapèzes remplace la courbe de la fonction par une série de segments linéaires qui relient les points calculés de la fonction. Ce processus transforme l'aire sous la courbe en une somme d'aires de trapèzes.

5.1 Formalisation de la Méthode

Considérons l'intervalle $[a, b]$ divisé en n sous-intervalles égaux de largeur $h = \frac{b-a}{n}$. Les points x_0, x_1, \dots, x_n sont définis tels que $x_i = a + i \cdot h$. L'aire sous la courbe de f sur chaque sous-intervalle $[x_i, x_{i+1}]$ est approximée par l'aire du trapèze formé par les points $(x_i, f(x_i))$ et $(x_{i+1}, f(x_{i+1}))$.

La somme des aires de ces trapèzes donne l'estimation de l'intégrale :

$$\int_a^b f(x) dx \approx \frac{h}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n))$$

5.2 Élaboration et Justification

L'expression dérivée se fonde sur le calcul de l'aire d'un trapèze isolé, où h est la hauteur ou la largeur du sous-intervalle, et $f(x_i)$ et $f(x_{i+1})$ sont les longueurs des bases parallèles du trapèze. Cette formule garantit que chaque point, sauf les extrémités, contribue deux fois, reflétant leur partage entre deux trapèzes adjacents.

5.3 Évaluation de l'Erreur

L'erreur commise par la méthode des trapèzes est liée à la seconde dérivée de la fonction f sur l'intervalle $[a, b]$. Plus précisément, l'erreur est proportionnelle à la largeur des sous-intervalles au carré, donnant lieu à une erreur globale qui décroît lorsque n augmente. Pour une fonction f deux fois dérivable, La méthode est d'ordre $N = 1$. Le noyau de Peano est donné par la formule :

$$\begin{aligned} K_1(t) &= \int_a^b (x-t)_+ dx - \left((b-a) \left(\frac{a+b}{2} - t \right) \right) \\ &= \left[\frac{(x-t)^2}{2} \right]_{x=a}^{x=b} - \frac{b-a}{2} ((a-t) + (b-t)) \\ &= \frac{(b-t)^2}{2} - \frac{b-a}{2} (b-t) \\ &= \frac{(b-t)(a-t)}{2} \end{aligned}$$

qui est de signe constant négatif sur $[a, b]$, d'où la formule pour une fonction f de classe C^2 :

$$E(f) = f''(c) \int_a^b \frac{(b-t)(a-t)}{2} dt = -\frac{(b-a)^3}{12} f''(c).$$

Par conséquent,

$$|E(f)| \leq \frac{(b-a)^3}{12} \|f''\|_{\infty}.$$

5.4 Mise en œuvre et exemples

Pour implémenter la méthode des trapèzes en Python, Voici un exemple d'algorithme :

```
1 def trapezes(f, a, b, n):
2     h = (b - a) / n
3     integral = (f(a) + f(b)) / 2
4     for i in range(1, n):
5         integral += f(a + i * h)
6     integral *= h
7     return integral
```

6 La Méthode de Gauss

La quadrature de Gauss est une approche qui améliore significativement l'efficacité des méthodes d'intégration numérique standard comme la méthode des trapèzes ou de Simpson. Elle est basée sur l'idée de choisir les points d'échantillonnage de la fonction et les poids de manière optimale pour maximiser la précision de l'approximation de l'intégrale.

6.1 Formulation de la Méthode

Pour une fonction f continue sur l'intervalle $[a, b]$, l'intégrale de f est approximée par :

$$\int_a^b f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

où x_i sont les zéros du n -ième polynôme orthogonal de Legendre $P_n(x)$, et w_i sont les poids calculés par :

$$w_i = \frac{2}{(1 - x_i^2)[P'_n(x_i)]^2}$$

6.2 Analyse d'Erreur

L'erreur de la méthode de Gauss peut être exprimée en utilisant le $(2n)$ -ième dérivé de la fonction intégrée, ce qui montre que la méthode est très efficace pour les fonctions qui peuvent être bien approchées par des polynômes de bas degré sur l'intervalle d'intégration. L'erreur est donnée par :

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b [P_n(x)]^2 dx$$

pour un certain ξ dans $[a, b]$.

6.3 Mise en œuvre et exemples

Pour implémenter la Méthode de Gauss en Python, Voici un exemple d'algorithme :

```
1 import numpy as np
2
3 def gauss(f, a, b, n):
4     x, y = np.polynomial.legendre.leggauss(n)
5     x_s = 0.5 * (b - a) * x + 0.5 * (b + a)
6     integral = np.sum(y * f(x_s)) * 0.5 * (b - a)
7     return integral
```

7 La Méthode d'Euler

La méthode d'Euler est l'une des plus simples méthodes numériques pour résoudre les équations différentielles ordinaires (EDO) du premier ordre qui peuvent être représentées sous la forme

$$\frac{dy}{dt} = f(t, y).$$

Cette méthode est particulièrement utile pour approcher des solutions là où les solutions analytiques sont difficiles ou impossibles à obtenir.

7.1 Formulation de la Méthode

Soit une équation différentielle donnée par :

$$\frac{dy}{dt} = f(t, y)$$

où $y(t_0) = y_0$ est la condition initiale. La méthode d'Euler approxime la solution en progressant par des pas discrets de taille h .

Pour chaque pas n , la valeur suivante y_{n+1} est prédite par la formule :

$$y_{n+1} = y_n + h \cdot f(t_n, y_n) \quad t_{n+1} = t_n + h.$$

7.2 Démonstration du Théorème

Le développement principal repose sur l'approximation de la dérivée par une différence finie :

$$\frac{dy}{dt} \approx \frac{y_{n+1} - y_n}{h}$$

En substituant dans l'équation différentielle, nous obtenons :

$$\frac{y_{n+1} - y_n}{h} = f(t_n, y_n)$$

qui réarrangée donne la formule d'Euler :

$$y_{n+1} = y_n + h \cdot f(t_n, y_n)$$

7.3 Analyse d'Erreur

L'erreur de troncature locale à chaque pas est donnée par le développement de Taylor :

$$y(t_{n+1}) = y(t_n) + h \cdot y'(t_n) + \frac{h^2}{2} \cdot y''(\xi), \quad \xi \in [t_n, t_{n+1}]$$

L'erreur introduite à chaque étape est donc de l'ordre de $O(h^2)$, ce qui rend cette méthode d'ordre un en termes de convergence. Cependant, l'erreur peut s'accumuler sur de nombreux pas, nécessitant un soin dans le choix du pas de temps h .

7.4 Mise en œuvre et exemples

Pour implémenter la Méthode d'Euler en Python, Voici un exemple d'algorithme :

```
1 def euler(f, x0, y0, h, n):
2
3     x_v = [x0]
4     y_v = [y0]
5
6     for i in range(1, n+1):
7         x = x0 + i * h
8         y = y_v[-1] + h * f(x_v[-1], y_v[-1])
9         x_v.append(x)
10        y_v.append(y)
11
12    return x_v, y_v
```

8 Références

Livre : Mathématiques Tout-en-un pour la Licence 2, 2e édition, Sous la direction de Jean-Pierre RAMIS et André WARUSFEL : pages 928 a 945

Tout le Cours - Integration Numerique ,L2 Analyse numérique 1 S3, Alexandre Vidal