

Séance 9 Intégrales numériques

Précédemment, nous nous sommes uniquement intéressés au calcul exact d'intégrales. Il peut être aussi très utile de calculer une intégrale numériquement lorsque sa valeur n'est pas calculable exactement, ou parfois simplement car seul le résultat numérique est intéressant.

Les méthodes d'intégration numérique ont pour but de créer des suites approchant la valeur d'une intégrale donnée, en maîtrisant l'erreur commise (si on ne connaît pas l'erreur, le calcul ne sert à rien), et de préférence avec le moins de calculs possibles.

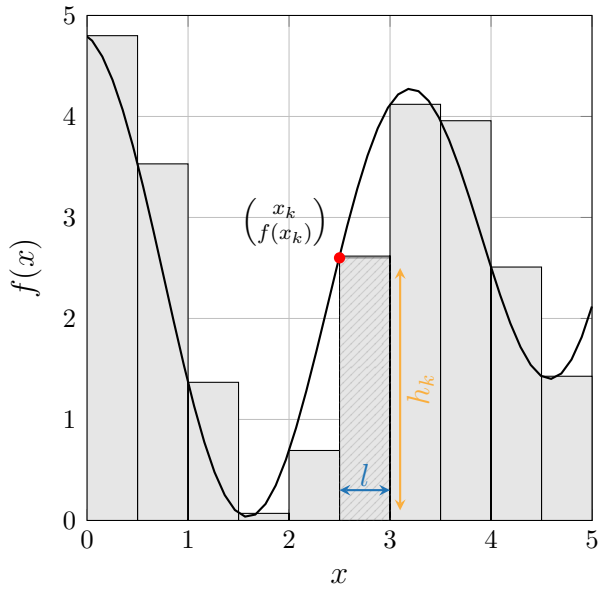
1 Retour sur la méthode des rectangles

Principe : On approche la courbe de la fonction f sur l'intervalle $[a, b]$ par des rectangles de largeur constante (en nombre n fixé à l'avance) et de hauteur égale à la valeur d'un réel dans l'intervalle. Ainsi, la largeur de chaque rectangle est $l = (b - a)/n$ et la hauteur du $k^{\text{ième}}$ rectangle est $h_k = f(x_k)$, où x_k est un réel choisi dans le $k^{\text{ième}}$ intervalle $([a + kl, a + (k + 1)l])$. La surface du $k^{\text{ième}}$ rectangle est donc $l \times h_k$. On peut alors approcher l'aire sous la courbe par la somme de chaque rectangle,

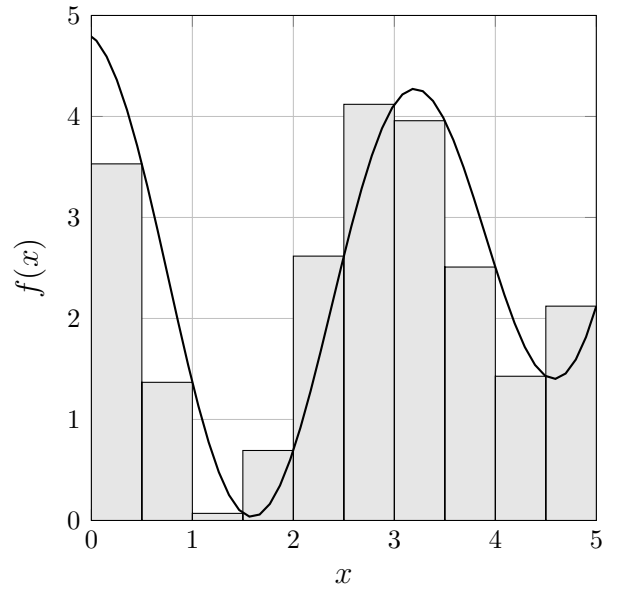
$$\int_a^b f(x) dx \simeq \sum_0^{n-1} lh_k = \frac{b-a}{n} \sum_{k=0}^{n-1} f(x_k).$$

En pratique on choisit la valeur à gauche de l'intervalle $x_k = a + \frac{k(b-a)}{n}$, ou la valeur à droite de l'intervalle $x_k = a + \frac{(k+1)(b-a)}{n}$.

Rectangles à gauche

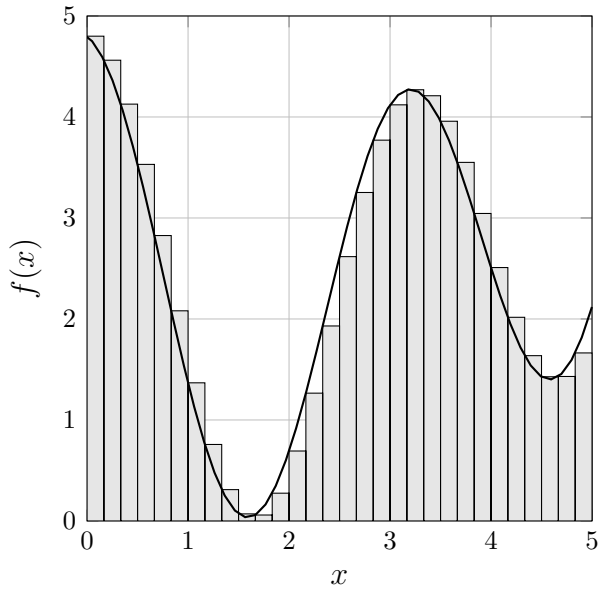


Rectangles à droite

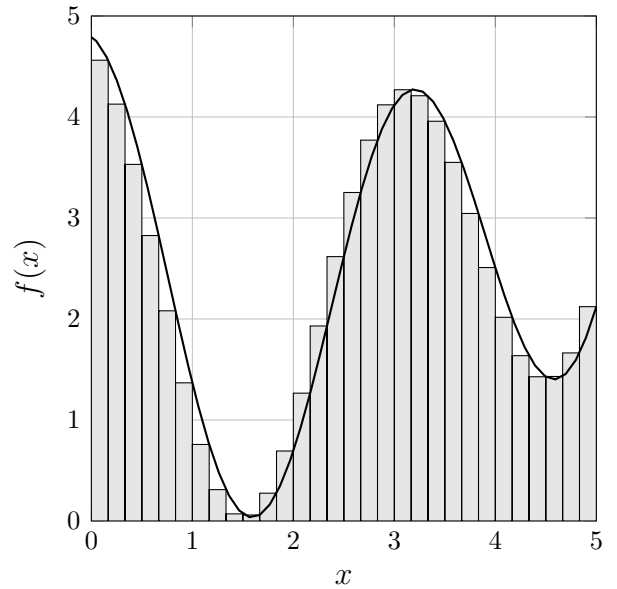


En augmentant le nombre de rectangles, on approxime de mieux en mieux l'aire sous la courbe

Rectangles à gauche



Rectangles à droite



Exemple de code numérique en Python.

Problème : intégrer la fonction $f(x) = x^2 + 2x + \cos(x)$ entre 0 et 1 avec 20 rectangles.

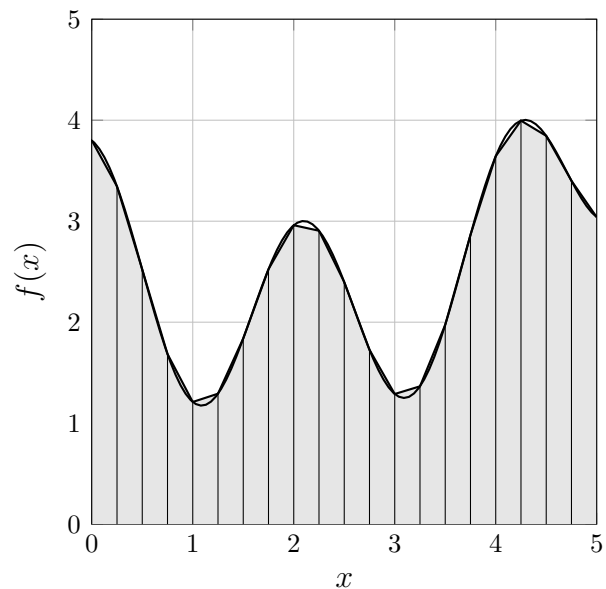
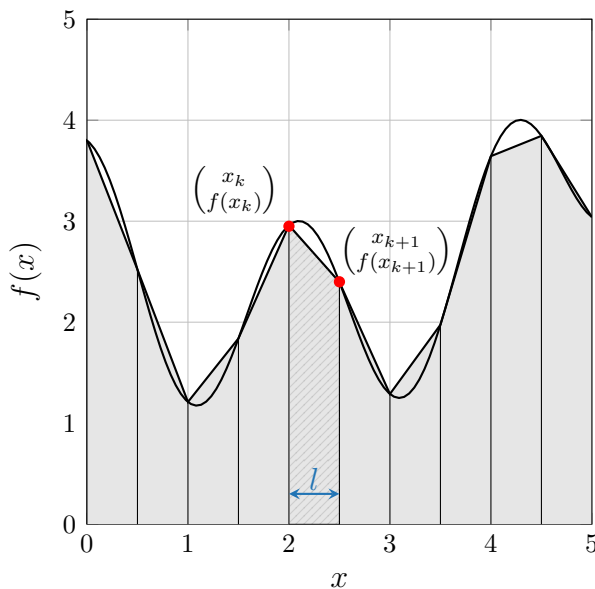
Librairies Python	<code>import numpy as np</code> <code>import matplotlib.pyplot as plt</code>	
Variables numériques	<code>xmin = 0</code> <code>xmax = 1</code> <code>nbx = 21</code> <code>nbi = nbx - 1</code>	<code># borne inférieure</code> <code># borne supérieure</code> <code># nombre de points</code> <code># nombre d'intervalles</code>
Évaluation numérique de la fonction	<code>x = np.linspace(xmin, xmax, nbx)</code> <code>y = x**2+2*x+np.cos(x)</code> <code>plt.plot(x,y,"bo-")</code>	<code># abscisses</code> <code># évaluation de $f(x)$</code>
Somme de l'aire des rectangles	<code>integrale = 0</code> <code>for i in range(nbi):</code> <code>integrale = integrale + y[i]*(x[i+1]-x[i])</code> <code># dessin du rectangle</code> <code>x_rect = [x[i], x[i], x[i+1], x[i+1], x[i]]</code> <code>y_rect = [0 , y[i], y[i] , 0 , 0]</code> <code>plt.plot(x_rect, y_rect,"r")</code>	<code># initialisation</code> <code># boucle for</code>
Résultats	<code>print("integrale =", integrale)</code> <code>plt.show()</code>	<code># affichage du résultat</code>

Les fonctions en **bleu** correspondent aux représentations graphiques du calcul numérique. Attention à l'indentation de la boucle for, en Python, c'est l'indentation qui délimite l'intérieur de la boucle for.

2 Méthode des trapèzes

Principe : Au lieu d'approximer la fonction par une constante sur un petit intervalle, on va maintenant l'approximer par une droite. De manière similaire à la méthode des rectangles, on découpe l'intervalle d'intégration en n segments de largeur $l = (b - a)/n$. Maintenant, on approche désormais l'intégrale du $k^{\text{ième}}$ segment par l'aire du trapèze passant par les deux points $(x_k, f(x_k))$ et $(x_{k+1}, f(x_{k+1}))$. L'aire du $k^{\text{ième}}$ trapèze vaut $l(f(a_k) + f(a_{k+1}))/2$. On approche l'intégrale par la somme de l'aire de tous les trapèzes,

$$\begin{aligned}
 \int_a^b f(x)dx &\simeq \sum_0^{n-1} \frac{l(f(a_k) + f(a_{k+1}))}{2} = \frac{b-a}{2n} \sum_{k=0}^{n-1} (f(x_k) + f(x_{k+1})) \\
 &= \frac{b-a}{2n} \left(\sum_{k=0}^{n-1} f(x_k) + \sum_{k=0}^{n-1} f(x_{k+1}) \right) \\
 &= \frac{b-a}{2n} \left(\sum_{k=0}^{n-1} f(x_k) + \sum_{k=1}^n f(x_k) \right) \\
 &= \frac{b-a}{2n} \left(f(a) + \sum_{k=1}^{n-1} f(x_k) + \sum_{k=1}^n f(x_k) \right) \\
 &= \frac{b-a}{2n} \left(f(a) + \sum_{k=1}^{n-1} f(x_k) + \sum_{k=1}^{n-1} f(x_k) + f(b) \right) \\
 &= \frac{b-a}{2n} \left(f(a) + f(b) + 2 \sum_{k=1}^{n-1} f(x_k) \right)
 \end{aligned}$$



La méthode des trapèzes permet de converger plus rapidement (on a besoin de considérer moins d'intervalles) vers la valeur de l'intégrale que la méthode des rectangles.

3 Tests de convergence numérique

Lorsque l'on calcule une intégrale de manière analytique, on obtient un résultat exact. Au contraire, lorsque l'on calcule une intégrale de manière approchée avec n'importe quelle méthode numérique, on obtient une valeur approchée. Comment peut-on être certain que cette valeur est correcte ?

Avec une approximation numérique, il faut d'abord se poser une contrainte de précision : Combien désire-t-on de chiffres significatifs pour considérer comme correcte la valeur de l'intégrale ?

Lorsque l'on a fixé N chiffres significatifs pour le calcul de l'intégrale, on choisit un nombre d'intervalles de départ, et on calcule numériquement l'intégrale. On obtient un résultat R_1 avec N chiffres significatifs. On augmente ensuite le nombre d'intervalles (on peut par exemple

doubler le nombre d'intervalles), et on estime une nouvelle valeur d'intégrale R_2 avec N chiffres significatifs.

On compare un à un les N chiffres significatifs de R_1 et R_2 . Si les N chiffres sont identiques, on considère le résultat numérique valide. Sinon, on itère la procédure de raffinement de la description de la fonction en augmentant le nombre d'intervalles. On trouve R_3 . Si la comparaison des N chiffres significatifs est bonne on stoppe la procédure, sinon on itère en augmentant le nombre d'intervalles.

Problème Calcul de l'intégrale de $f(x)$ sur un segment avec N chiffres significatifs

Initialisation Choix d'un nombre d'intervalles de départ

Itération 1 Calcul de l'approximation numérique de l'intégrale 1, Résultat R_1 .

Itération 2 Augmentation du nombre d'intervalles (on double par exemple).
Calcul de l'approximation numérique de l'intégrale 2, Résultat R_2 .
Comparaison des N chiffres significatifs de R_1 et R_2 .
Si identiques STOP, sinon ITÉRATION.

⋮

⋮

Itération i Augmentation du nombre d'intervalles (on double par exemple).
Calcul de l'approximation numérique de l'intégrale i , Résultat R_i .
Comparaison des N chiffres significatifs de R_{i-1} et R_i .
Si identiques STOP, sinon ITÉRATION.

Conclusion Le résultat de l'intégrale avec N chiffres significatifs est R_i