Informatique Appliquée au Calcul Scientifique 1
Séance 12

TP 6 : Intégration Numérique par les Méthodes des Rectangles et des Trapèzes
Évaluation

Table des matières

Objectifs

Ce TP vise à :

- Comprendre et appliquer les méthodes des rectangles et des trapèzes pour l'intégration numérique.
- Implémenter ces méthodes en Python avec des fonctions commentées.
- Comparer les résultats numériques avec la valeur exacte de l'intégrale.
- Analyser l'influence du pas sur la précision des méthodes.

Contexte

On souhaite calculer l'intégrale de la fonction $f(x) = x^2 + 1$ sur l'intervalle $[0, 2]$.
La valeur exacte de cette intégrale sera à calculer.
Vous utiliserez les méthodes des rectangles (à gauche) et des trapèzes pour approcher cette intégrale, d'abord manuellement, puis par programmation, et analyserez la précision obtenue.

Exercices

Exercice 1 : Calculs manuels

Calculez à la main l'approximation de l'intégrale de $f(x) = x^2 + 1$ sur $[0, 2]$ avec un pas $h = 0{,}5$ (soit 4 sous-intervalles) pour :
a) La méthode des rectangles (utilisez les points à gauche des sous-intervalles).
b) La méthode des trapèzes. Comparez vos résultats à la valeur exacte que vous aurez calculé au préalable et calculez l'erreur absolue pour chaque méthode.

Exercice 2 : Programmation en Python

Écrivez un programme Python qui :

a) Définit une fonction `integration_rectangle(f, a, b, h)` qui calcule l'intégrale par la méthode des rectangles (points à gauche) pour une fonction f sur un intervalle $[a\,;\,b]$ avec un pas de h.

b) Définit une fonction `integration_trapeze(f, a, b, h)` qui calcule l'intégrale par la méthode des trapèzes.

c) Teste ces fonctions pour $f(x) = x^2 + 1$ sur $[0, 2]$ avec les pas $h = 0{,}5$, $h = 0{,}1$, et $h = 0{,}01$.

d) Affiche les résultats et les erreurs par rapport à la valeur exacte.

<u>Commentez clairement votre code (rôle des paramètres, étapes du calcul, etc.).</u>

Exercice 3 : Analyse de la précision

Déterminez, par programmation, une valeur approximative du pas h pour laquelle l'erreur absolue des deux méthodes est inférieure à 10^{-4} . Justifiez vos résultats et commentez l'influence du pas sur la précision.

Instructions

- Fournissez vos calculs manuels détaillés pour l'exercice 1.
- Fournissez le code Python complet et commenté pour l'exercice 2.
- Pour l'exercice 3, incluez une analyse des résultats (par exemple, comment l'erreur évolue avec h).

- Vous pouvez utiliser la bibliothèque numpy pour les calculs numériques.

Rendu

Rédigez un rapport clair incluant :

- Les calculs manuels de l'exercice 1.
- Le code Python et les résultats de l'exercice 2.
- L'analyse de l'exercice 3 avec une discussion sur la précision.

Le rapport doit être soumis en format PDF ou Word et vos fichiers python seront joins. Vous ferez attention à bien faire figurer votre nom et votre prénom dans le nom des fichiers.

Rappel Python :

Création d'une liste de n points sur l'intervalle [a ; b] :

```
import numpy as np
x = np.linspace(a, b, n)
```

Garder la partie entière d'un nombre avec int():

```
n = int((b - a) / h)
```

Création d'une fonction mathématique :

```
f = lambda x: x^2 + 1
```

ou

```
def f(x):
    return x^2 + 1
```

Bonus :

Exercice 4 : Visualisation de l'erreur

Modifiez votre programme pour tracer, à l'aide de la bibliothèque matplotlib, un graphique représentant l'évolution de l'erreur absolue des deux méthodes en fonction du pas h.
Testez des valeurs de h allant de 1 à 0.001 (par exemple, h = 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001). Utilisez une échelle logarithmique pour les axes pour mieux visualiser la convergence.
Commentez le graphique : que remarque-t-on sur la vitesse de convergence des deux méthodes ?

Point Python :

Échelle logarithmique avec matplotlib :

```
import matplotlib.pyplot as plt
plt.loglog(pas_visu, erreurs_rect, marker='o', label='Méthode des rectangles')
```