

RAPPORT ANALYSE NUMERIQUE

Introduction

Dans ce rapport, nous analyserons les résultats obtenus pour l'intégration numérique d'un polynôme de degré 3 à l'aide de plusieurs méthodes d'approximation : la méthode des rectangles, la méthode des trapèzes et la méthode de Simpson. L'objectif est d'évaluer la précision et l'efficacité computationnelle de ces méthodes en fonction du nombre de segments utilisés.

Trois axes principaux guideront notre étude :

1. L'erreur d'approximation : Nous comparerons la précision des différentes méthodes en mesurant l'écart entre l'intégrale exacte et les valeurs obtenues numériquement.
2. Le temps de calcul : Nous examinerons comment l'augmentation du nombre de segments influence le temps d'exécution des différentes méthodes.
3. L'impact de NumPy et Scipy : Nous mettrons en évidence les avantages des versions vectorisées en NumPy et Scipy par rapport aux implémentations classiques en Python.

Nous illustrerons ces observations à l'aide de graphiques montrant :

- L'évolution de l'erreur en fonction du nombre de segments.
- La variation du temps de calcul en fonction du nombre de segments.

Cette analyse nous permettra de mieux comprendre les compromis entre précision et performance pour chacune des méthodes d'intégration, et d'identifier les choix les plus efficaces en fonction des contraintes de calcul.

Voici les différents paramètres choisis pour les différents tests :

Le polynome : $f(x) = 1 + 2x + 3x^2 + 4x^3$

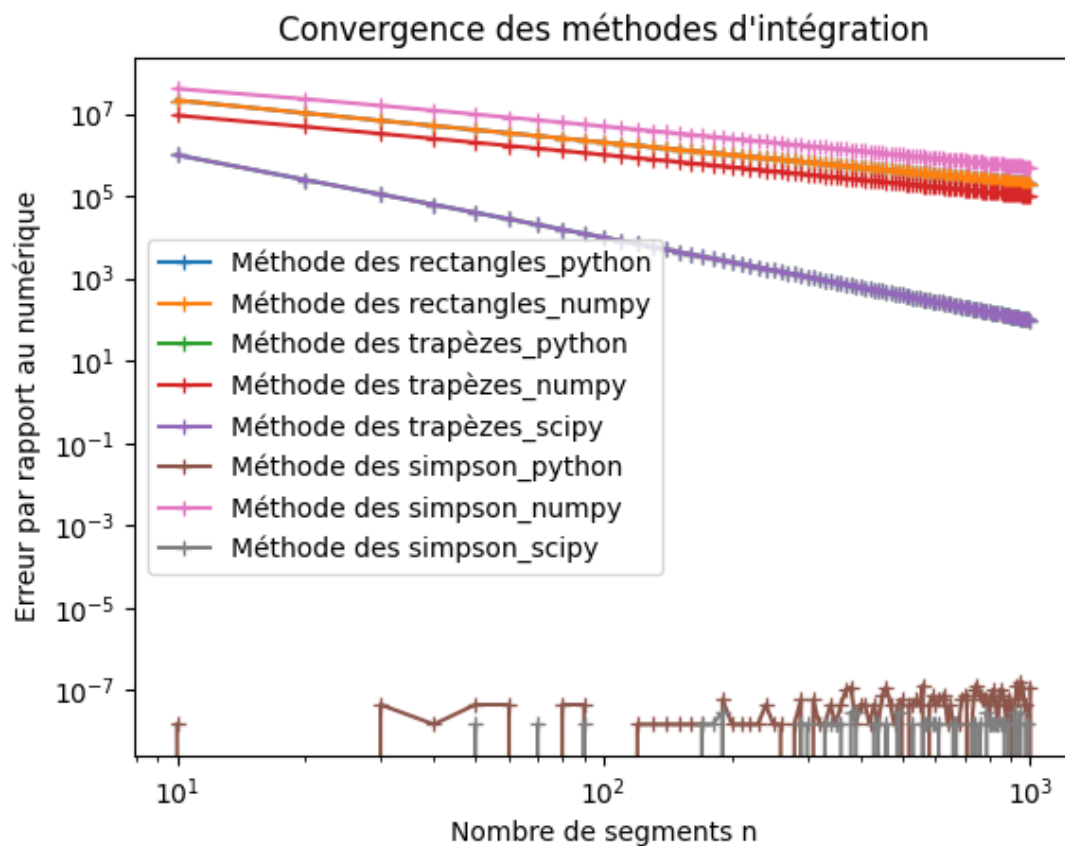
Le nombre de segments maximum : 1000

La borne inférieure : 1

La borne supérieure : 100

Analyse des Résultats

1. Convergence et Erreur en fonction du nombre de segments

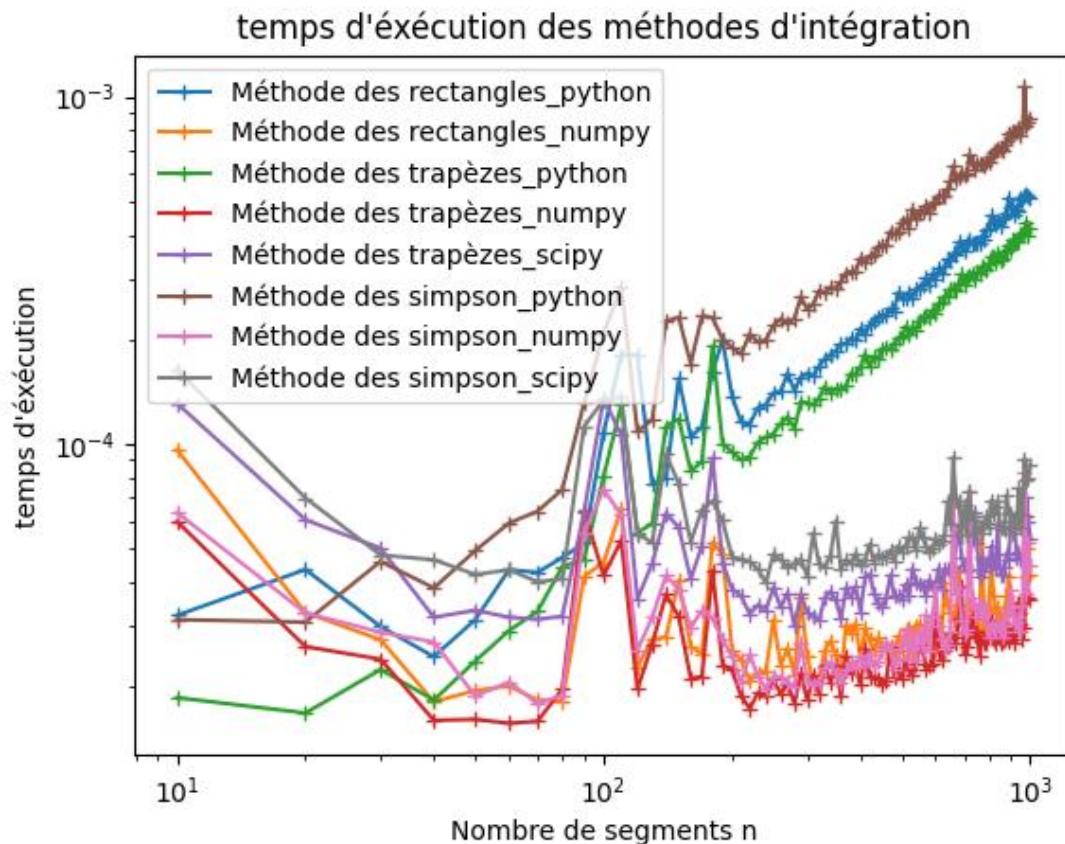


Graphique montrant l'erreur et la convergence en fonction du nombre de segments

Le premier graphique illustre L'évolution de l'erreur en fonction du nombre de segments. On peut observer les points suivants :

- **Tendance générale :** Plus le nombre de segments augmente, plus l'intégrale numérique se rapproche de la solution exacte.
- **Différence entre méthodes :**
 - La méthode des rectangles converge plus lentement, car elle repose sur une approximation rudimentaire.
 - La méthode des trapèzes améliore légèrement la précision en tenant compte des valeurs aux extrémités des sous-intervalles.
 - La méthode de Simpson offre une convergence beaucoup plus rapide, grâce à son approximation par des paraboles qui réduit significativement l'erreur.
- **Impact du nombre de segments :** On note que la convergence s'améliore rapidement lorsque le nombre de segments passe de faible à modéré, puis la précision stagne lorsque l'on atteint un nombre de segments élevé (loi des rendements décroissants).

2. Temps de calcul en fonction du nombre de segments



Graphique montrant le temps d'exécution en fonction du nombre de segments

Le deuxième graphique montre comment le temps d'exécution varie avec l'augmentation du nombre de segments. Voici les observations clés :

- **Tendance générale** : Le temps de calcul augmente avec le nombre de segments, mais à des rythmes différents selon la méthode employée.
- **Comparaison entre méthodes** :
 - La méthode des trapèzes est visiblement la méthode la plus rapide.
 - La méthode des rectangle reste performante.
 - La méthode de Simpson, bien que plus précise, demande plus de calculs et son temps d'exécution augmente plus rapidement.
- **Impact de NumPy et SciPy** : Si on test les versions optimisées avec NumPy, on observe un gain de performance, surtout pour un grand nombre de segments. L'utilisation de tableaux et d'opérations vectorisées permet d'éviter les boucles et d'accélérer le traitement.

Conclusions Générales

Antonin Chiarotto
Victor Garcia

- **Équilibre précision/temps** : La méthode de Simpson est la plus précise mais aussi la plus coûteuse en calcul.
- **Choix optimal** : Pour un compromis entre précision et rapidité, la méthode des trapèzes avec NumPy est souvent un bon choix.
- **Avantages de NumPy et Scipy** : L'utilisation de NumPy et Scipy peut significativement réduire le temps d'exécution tout en maintenant une bonne précision.