Le génie pour l'industrie

PROJET DE SESSION
MINI PROJET B

GROUPE 5
ALEXIS MARKIEWICZ
GRANGE MARTIN
SCOTTU EMILIO

12 juin 2024 MGA802

Analyse de 3 méthodes d'intégration numérique

But du projet et contexte :

Nous allons étudier 3 méthodes d'intégration numérique : la méthode des rectangles, des trapèzes et de Simpson. Le but est de programmer chacune de ces méthodes en python de base, en numpy et d'utiliser leurs versions préprogrammées lorsqu'elles existent. Ainsi, nous pourrons comparer ces dernières entre-elles et observer l'impact du choix de programmation.

Le polynôme que l'on va étudier est le suivant : $f(x) = p_1 + p_2x + p_3x^2 + p_4x^3 = 1 - 200x - 300x^2 + 4x^3$

Sur l'intervalle $[a,b] = [0,100]$, on peut calculer la valeur exacte de l'intégrale de f : $\int_0^{100} f(x)dx = -999\,900$

Méthode des rectangles :

Premièrement on utilise la méthode des rectangles du point du milieu, c'est-à-dire que l'on approche l'aire sous la courbe par des rectangles de largeur définie (segment entre deux points successifs) et de hauteur la valeur de f évalué au milieu de ce segment. Ensuite, il suffit de sommer l'aire de chacun de ces rectangles afin d'obtenir une approximation de l'intégrale. De façon mathématique cela s'écrit de la manière suivante :

$$\int_a^b f(x)dx = \sum f\left(\frac{x_{i+1} + x_i}{2}\right) * (x_{i+1} - x_i)$$

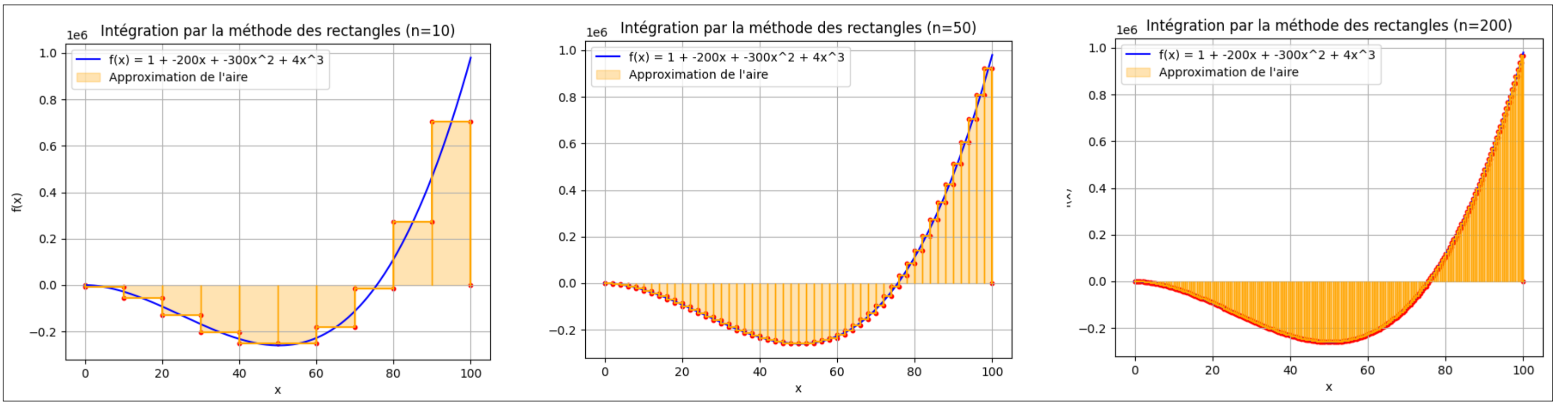

Figure 1 : Illustration de la méthode des rectangles du point du milieu pour n = 10, 50 et 200

	n = 10	n = 50	n = 200
Approximation de l'intégrale	-1 249 900	-1 009 900	-1 000 525

La méthode des rectangles du point du milieu a tendance à surestimer l'intégrale d'un polynôme. Sans surprise, plus on ajoute de segments, plus le résultat se rapproche de la valeur analytique.

Méthode des trapèzes :

La méthode des trapèzes utilise la hauteur aux extrémités de chaque segment afin de former un trapèze, on calcule l'aire de l'ensemble des trapèzes et on somme ces aires pour obtenir une approximation de l'intégrale.

De façon mathématique cela s'écrit : $\int_a^b f(x)dx = \sum \frac{f(x_i+1)-f(x_i)}{2} * (x_{i+1} - x_i)$

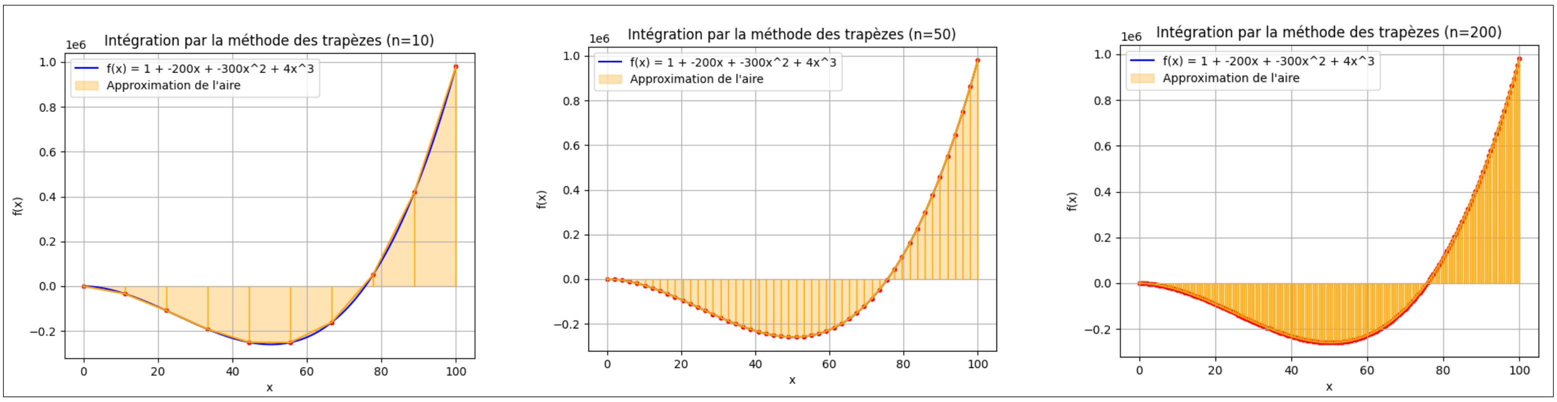

Figure 2 : Illustration de la méthode des trapèzes pour n = 10, 50 et 200

	n = 10	n = 50	n = 200
Approximation de l'intégrale	-499 900	-979 900	-998 650

Le polynôme choisi implique que de petits changements sont représentés par de grandes valeurs. Pour 10 segments, l'aire négative est sous-estimée et l'aire positive est, elle, sur estimée. Ainsi, quand on augmente le nombre de segments, l'intégrale devient plus précise, l'aire négative augmente et celle positive diminue. Le résultat de l'intégrale augmente alors négativement.

On remarque que lorsque la courbe se trouve en dessous de l'axe des abscisses, si elle est concave, la méthode sur estime l'aire et si la courbe est convexe, la méthode sous-estime l'aire au-dessus de la courbe. Lorsqu'on se trouve au-dessus de l'axe des abscisses, le comportement est inversé.

Méthode de Simpson :

La méthode de Simpson approche la courbe sur un segment en utilisant un polynôme du second ordre c'est-à-dire une parabole. L'aire de chaque segment est sommée afin d'obtenir une approximation de l'intégrale. De façon mathématique cela s'écrit de la manière suivante :

$$\int_a^b f(x)dx = \sum \frac{x_{i+1} - x_i}{6} * [f(x_i) + 4f\left(\frac{x_{i+1} + x_i}{2}\right) + f(x_i + 1)]$$

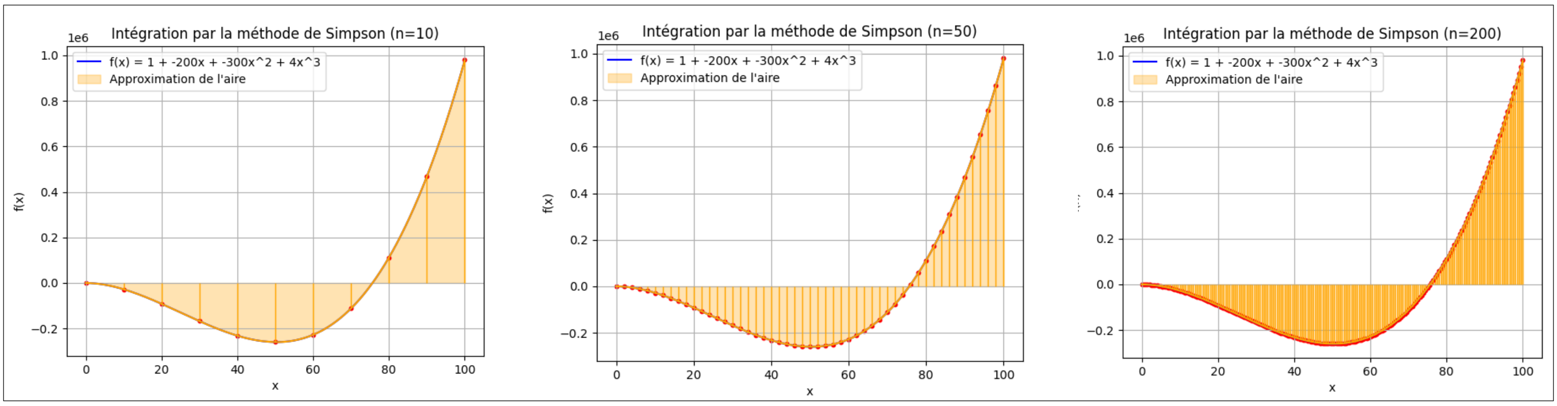

Figure 3 : Illustration de la méthode Simpson pour n = 10, 50 et 200

	n = 10	n = 50	n = 200
Approximation de l'intégrale	-999 900	-999 900	-999 900

La méthode de Simpson donne le même résultat que le calcul analytique dès n = 10. Elle est très efficace pour faire une approximation de polynômes de degrés 3 ou moins.

Comparaison des résultats :

Comparaison des versions pour la méthode des rectangles :

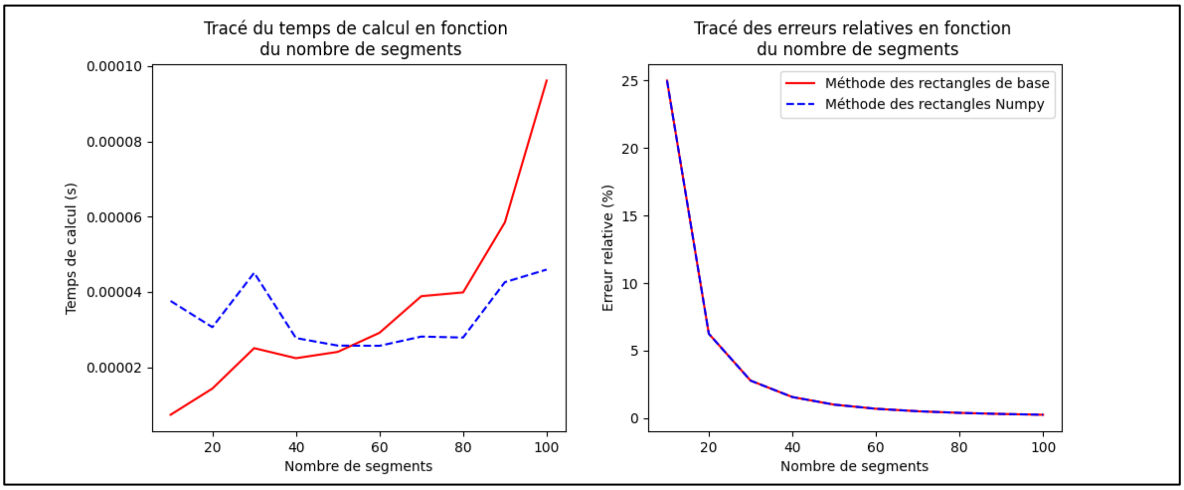

Figure 4 : Temps de calcul et erreur relative pour la méthode des rectangles

Pour la méthode des rectangles, l'implémentation avec les outils de base de python est 2 à 4 fois plus rapide pour des petits nombres de segments mais au bout de 60 segments, la méthode utilisant numpy devient plus rapide. Plus encore, la méthode utilisant numpy semble s'exécuter en autant de temps peu importe le nombre de segments. Par ailleurs, les deux implémentations ont la même erreur relative. Il est donc plus intéressant d'utiliser l'implémentation avec numpy dans ce cas.

Comparaison des versions pour la méthode des trapèzes :

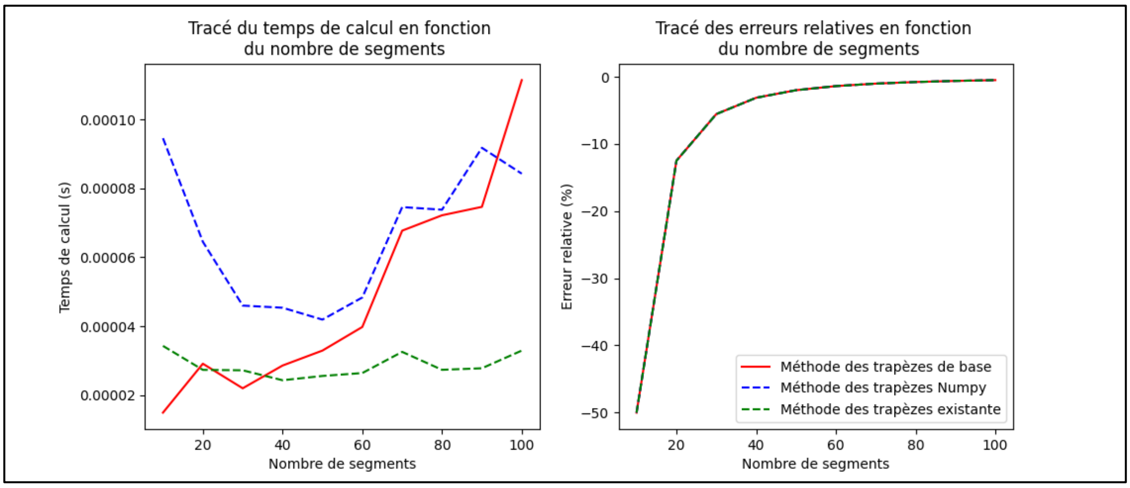

Figure 5 : Temps de calcul et erreur relative pour la méthode des trapèzes

Pour la méthode des trapèzes, les erreurs sont identiques peu importe la manière dont on l'a implémentée. Au niveau des temps de calcul on remarque que la méthode déjà existante dans le module numpy est la plus rapide. La méthode implémentée avec les outils basique de python est presque aussi rapide que la méthode déjà implémentée jusqu'à 40 segments, ensuite elle devient 2 fois plus lente. On peut donc conclure que la méthode déjà existante est la plus intéressante.

Comparaison des versions pour la méthode de Simpson :

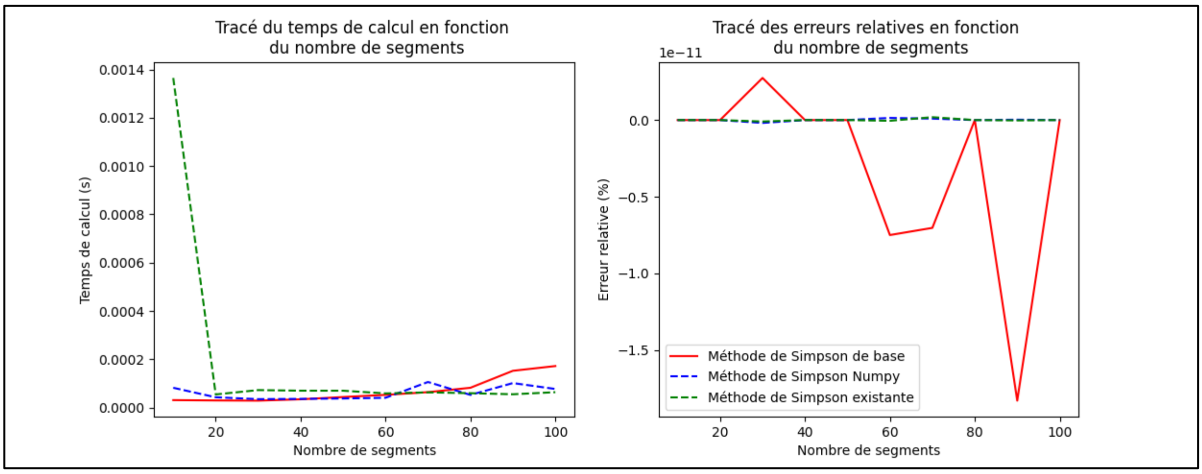

Figure 6 : Temps de calcul et erreur relative pour la méthode de Simpson

Le point fou obtenu pour 10 segments proviendrait d'abord du fait que nos ordinateurs ne sont pas RealTime mais aussi du fait qu'à l'appel de la fonction, certains modules se chargent et rendent la première mesure du temps plus longue. Pour la méthode de Simpson, les erreurs varient très peu selon l'implémentation. D'un autre côté les temps de calcul sont quasiment identiques pour l'implémentation utilisant numpy et celle déjà existante. La méthode utilisant les outils basiques de python est la plus rapide jusqu'à 40 segments, à partir de là elle devient plus lente. Les méthodes numpy et existante semblent alors les plus intéressantes. Les erreurs peuvent varier selon le nombre de segments car les approximations d'ordre deux épousent plus ou moins bien la courbe réelle. On reste sur des erreurs nulles ($10^{-11}\%$).

Comparaison des méthodes implémentées avec python :

Nous pouvons dès lors comparer les méthodes entre-elles selon les versions utilisées. Dans le cas où seuls les outils de bases de python sont utilisés, on remarque que la méthode Simpson à un temps de calcul relativement identique à la méthode des trapèzes, qui est 2 fois plus grand que le temps de calcul de la méthode des rectangles. Cela n'est pas surprenant car la méthode des rectangles effectue moins de calculs que les deux autres méthodes. Pour ce qui est des erreurs relatives, la méthode Simpson est largement plus précise, alors que les autres ont des erreurs atteignant 50 %. Les méthodes convergent toutes vers le bon résultat avec l'augmentation du nombre de segments. La méthode Simpson semble donc être celle à privilégier lorsqu'on vise la précision. Le temps de calcul étant de l'ordre de 10^{-5} secondes, on peut juger qu'il est acceptable.

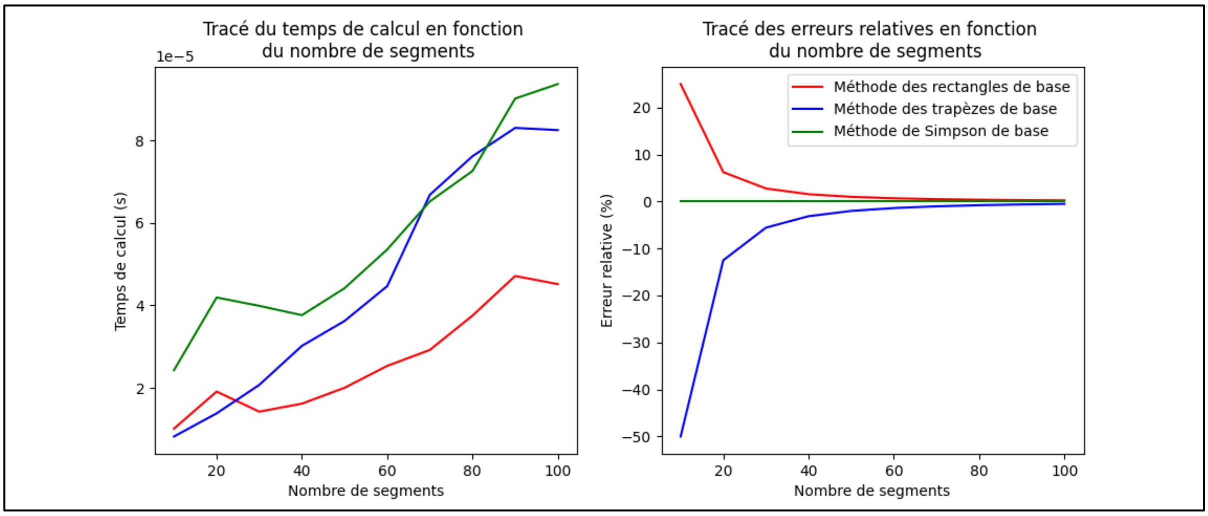

Figure 7 : Temps de calcul et erreur relative pour les méthodes implémentées sur python de base

Comparaison des méthodes implémentées avec numpy :

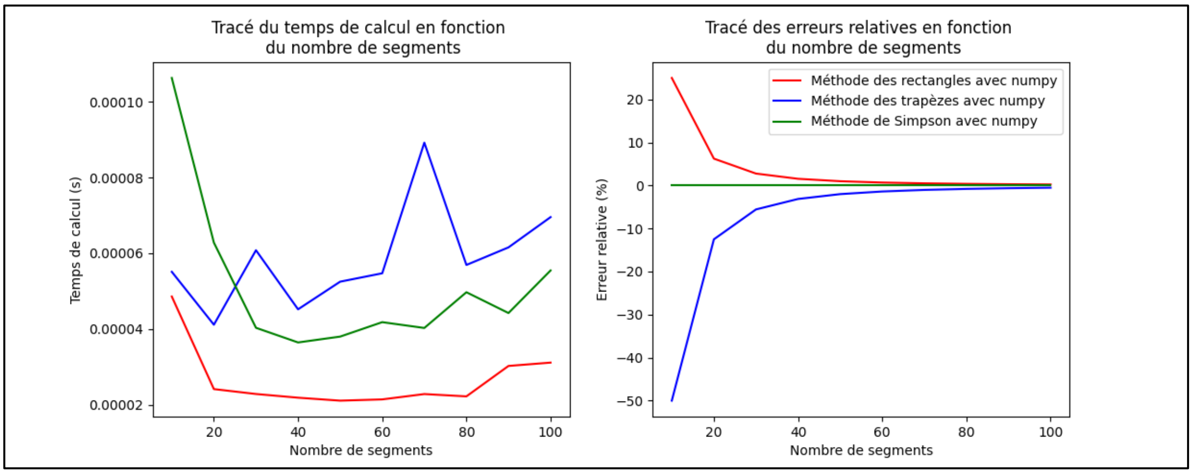

Figure 8 : Temps de calcul et erreur relative pour les méthodes implémentées avec numpy

Lorsqu'on compare les méthodes implémentées avec numpy, on observe les mêmes variations d'erreur relative que précédemment. En revanche le temps de calcul est modifié. Ainsi la méthode la plus rapide reste la méthode des rectangles mais la méthode des trapèzes est maintenant 1.5 fois plus lente que la méthode Simpson. La méthode Simpson devient alors encore plus intéressante dans ce cas. Cependant, à partir de 30 segments, l'erreur relative de la méthode des rectangles devient inférieure au 10 % ce qui la rend plus intéressante à utiliser lorsqu'on ne cherche pas une grande précision.

Nuances et Interprétation des Résultats

Il est important de comprendre que pour les méthodes des rectangles et des trapèzes, plus le nombre de segments est grand, plus l'intégrale numérique se rapproche du résultat analytique. Cependant, pour la méthode de Simpson, il est intéressant d'observer que les résultats obtenus jusqu'ici sont relatifs. On ne peut pas tirer de conclusions générales car les performances de la méthode dépendent premièrement du polynôme donné en entrée et surtout des bornes spécifiées. En effet, entre certaines bornes, le polynôme du troisième degré s'apparente à un trinôme du second degré. Dans ce cas, la méthode de Simpson est parfois plus performante pour un petit nombre de segments que pour un plus grand nombre.