RAPPORT – INTEGRATION NUMERIQUE

MGA 802 - SUJETS SPECIAUX I EN AERONAUTIQUE (É2024)

MINI PROJET B

AUTEURS – GROUPE 6 : ADRIEN HUYGHEBAERT (HUYA83280101) – BAPTISTE ROUANET (ROUB76310101) – CLELIA DURANDET (DURC86340001)

MONTREAL, LE 12 JUIN 2024

ÉCOLE DE TECHNOLOGIE SUPERIEURE

Introduction :

La fonction polynomiale choisie pour l'étude de comparaison entre les différentes méthodes est la suivante :

$$\mathbf{F(x)} = p_1 + \boldsymbol{p_2 x} + p_3 x^2 + p_4 x^3$$

Avec : p_1 **= 24** p_2 **= -30** p_3 **= -50** p_4 **= 3**

L'intervalle d'intégration choisi est **[a, b] = [-1 ; 1]** avec n le nombre de segments qui varie selon le cas d'étude.

1) Intégration par la méthode des rectangles

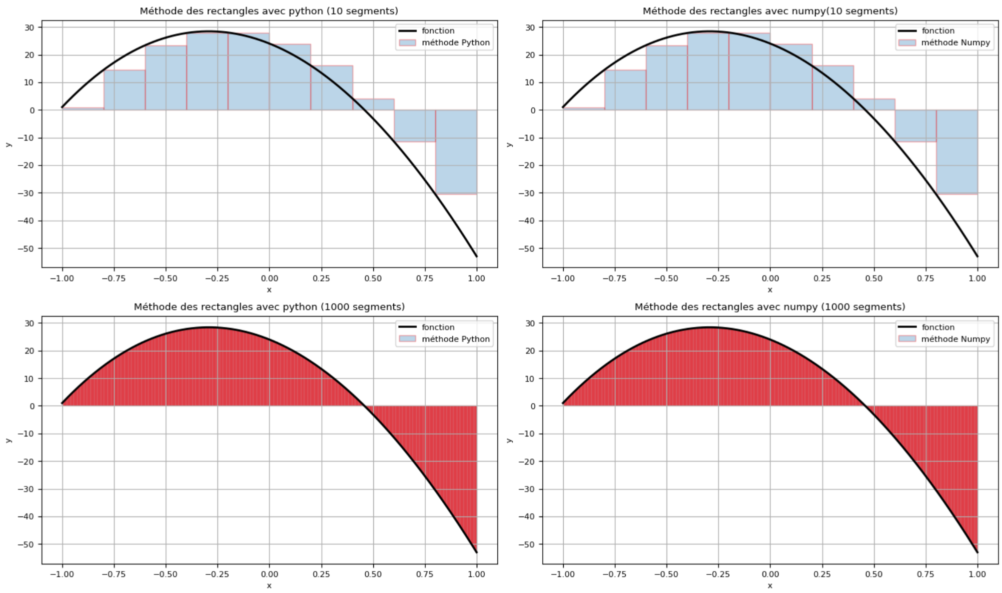

Figure 1 : Comparaison de la méthode des rectangles avec n=10 et n=100

On observe que les deux types de codage donnent des résultats très similaires. Il est nécessaire d'avoir beaucoup de segments pour pouvoir approximer correctement la solution analytique comme le montre le graphique ci-dessous. Avec 10000 segments, on commence à avoir un résultat satisfaisant (erreur = 10^{-2}).

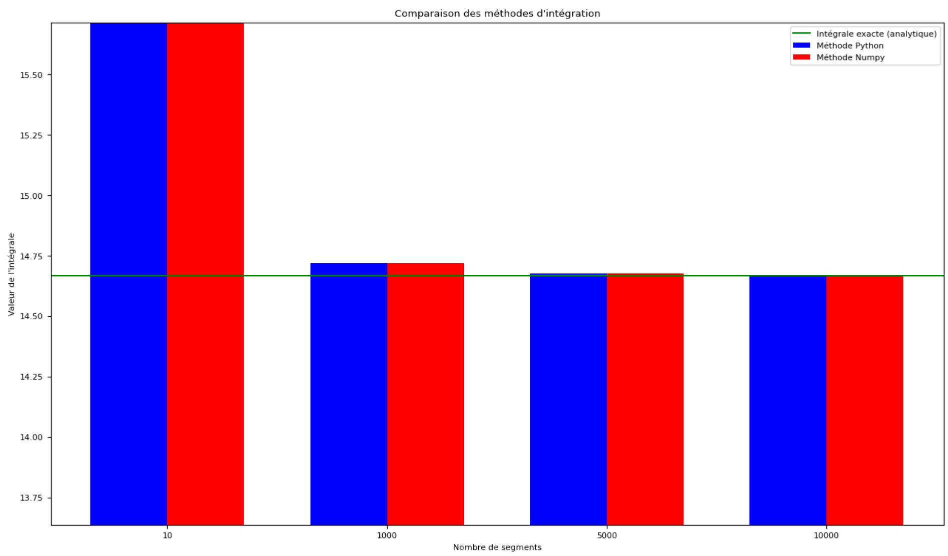

Figure 2 : Comparatif des valeurs d'intégrales par rapport à la valeur exacte (grossissement du graphique initial)

2) Comparaison avec la méthode des trapèzes7,93

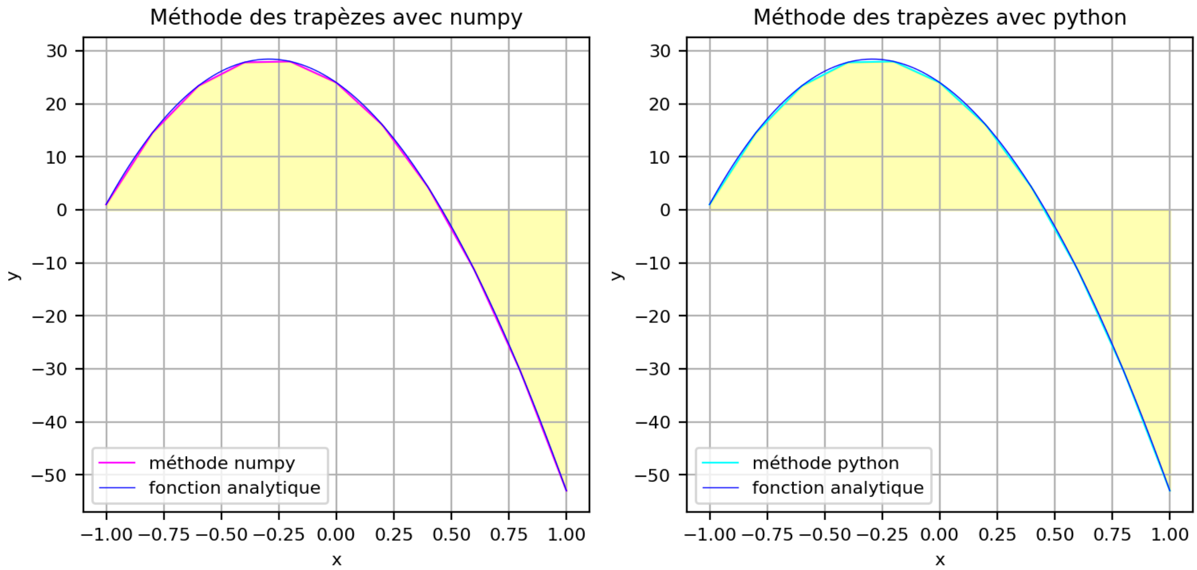

Figure 3 Comparaison de la méthode des trapèzes (python, numpy) avec n=10

On remarque que pour 10 segments la courbe est déjà très précise par rapport au tracé de la fonction polynomiale, avec n = 1000 l'aire est correcte à 10^{-3}. Pour n= 10, l'aire calculée avec les deux méthodes est très proche : $A = 13.99$ pour numpy et $A = 14.00$ pour python tandis que l'aire exacte vaut $A = 14.66$.

2) Comparaison avec la méthode de Simpson

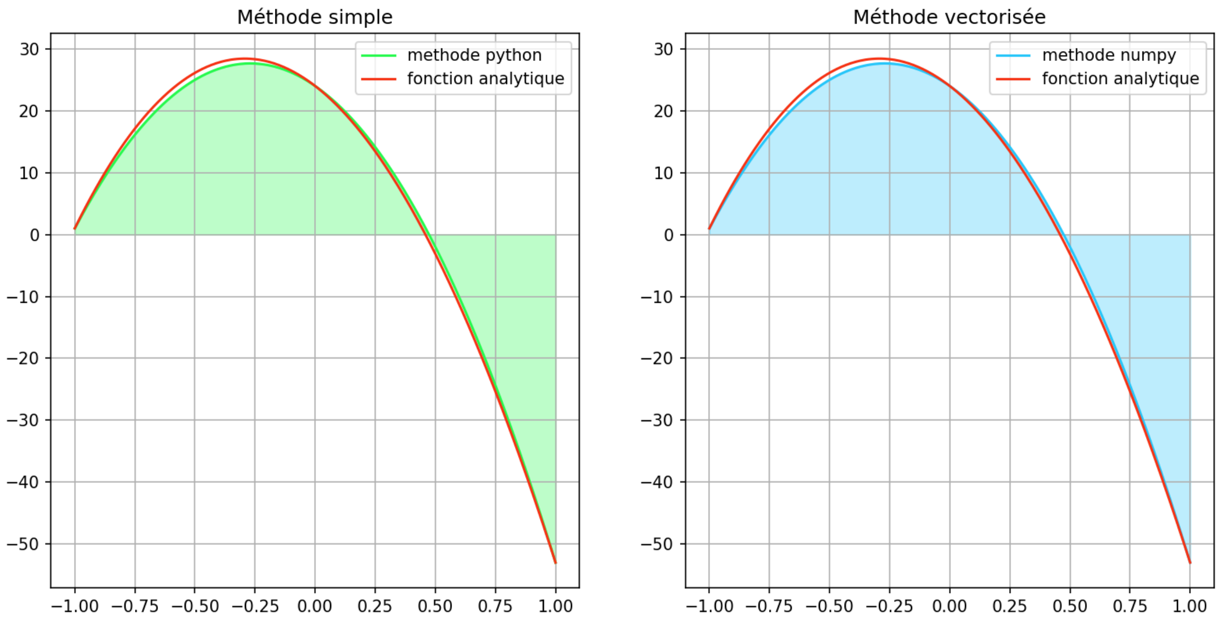

Figure 4. Comparaison des résultats entre la méthode python et la vectorisée (n=2)

On remarque que pour la configuration donnée dès n=2 on observe déjà une précision très élevée.

La précision étant très élevée due à l'étude d'une portion de la courbe en 2^{nd} degrés (on pourrait observer plus de différence pour du 3^{e} degrés)

3) Comparaison avec les méthodes préprogrammées (Scipy)

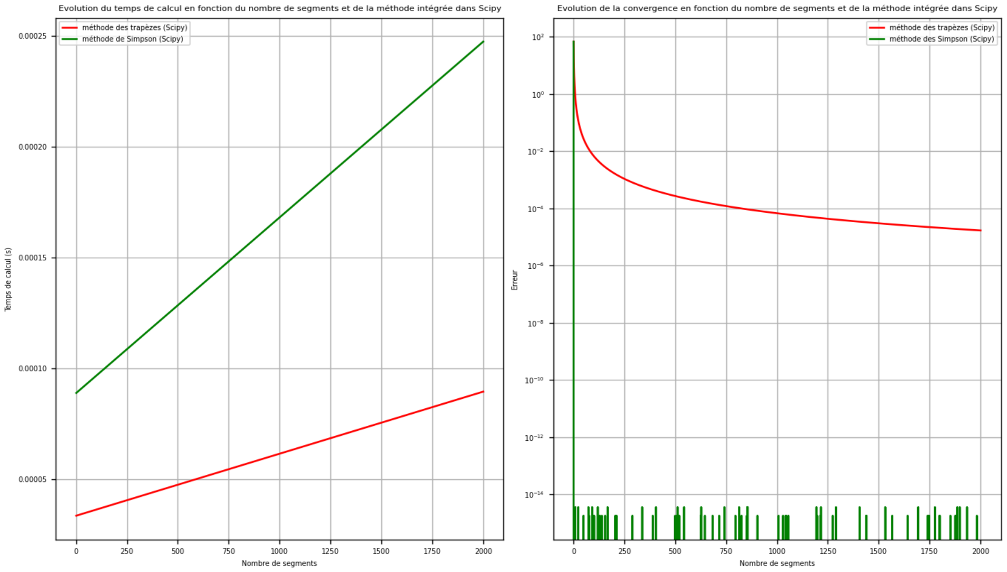

Figure 5. Comparaison des méthodes des trapèzes et de Simpson avec Scipy

La méthode des trapèzes de Scipy (en rouge) est plus rapide que celle de Simpson (en vert) mais elle est moins précise (erreur en 10^{-5} seulement).
On remarque également que la méthode Simpson est précise très rapidement (erreur = 10^{-8}) ce qui ne permet pas d'observer une convergence. Il faudrait tester la méthode sur une courbe plus complexe pour l'observer.

4) Résumé

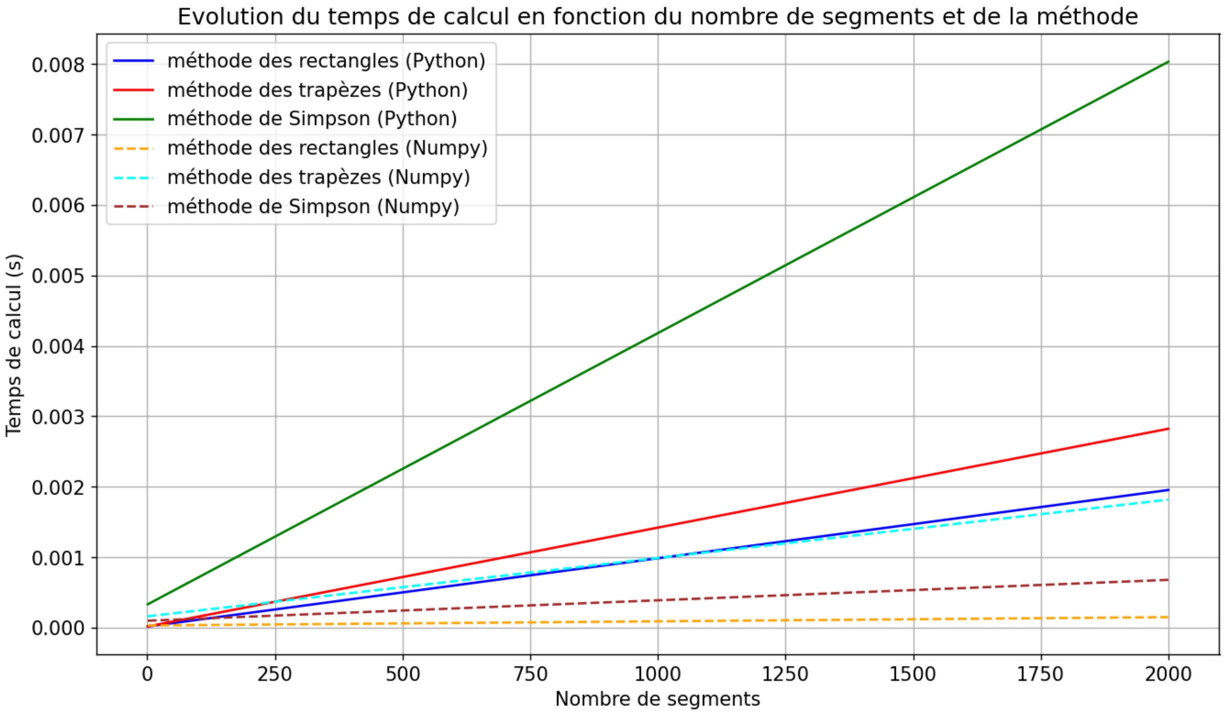

Figure 5. Comparaison des temps de calcul des 3 méthodes

Le temps de calcul augmente linéairement avec le nombre de segments pour les deux types de codage. Avec python, la méthode de Simpson est la moins rapide. Alors qu'avec Numpy c'est la méthode des trapèzes. Comme on s'y attend, la méthode des rectangles, moins complexe, est la plus rapide dans le cas du code Numpy pour un grand nombre de segments. De plus, elle semble rester stable malgré l'augmentation du nombre d'éléments. L'écart du temps de calcul est d'autant plus grand que le nombre de segment augmente, entre la méthode Simpson et les deux autres méthodes (à n = 200 par exemple). La méthode de vectorisation avec numpy devient plus efficace lorsque le nombre de segments est grand (> 100). Par ailleurs, on observe que les méthodes préprogrammées avec le package Scipy sont plus rapides que celles que nous avons codés nous même, même lorsque nous les avons vectorisées. Cela est dû au fait que le package Scipy réalise toutes les étapes en C alors que pour nos méthodes vectorisées, une partie de notre code est écrit en Python et Numpy derrière est en C. C'est cette partie écrite en Python qui fait que les méthodes préprogrammées sont plus rapide.

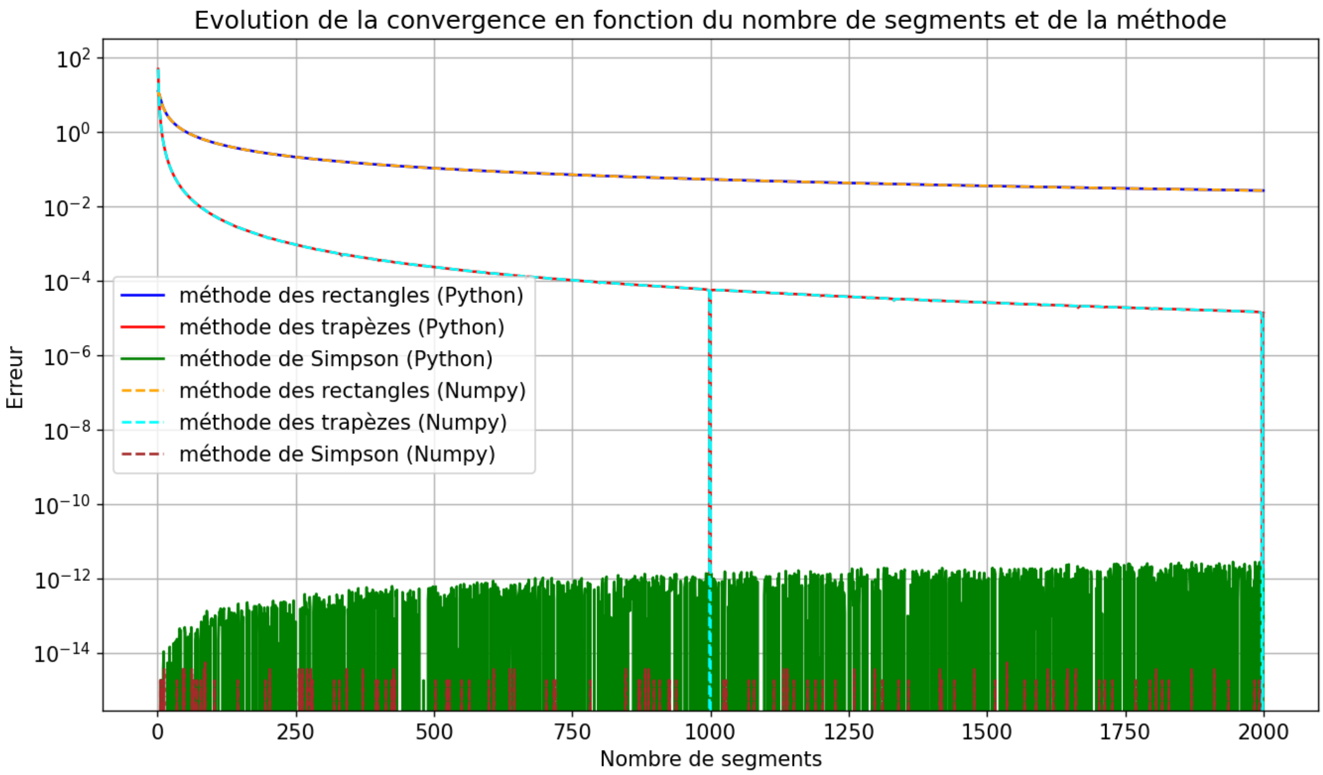

Figure 6. Comparaison de la convergence des 3 méthodes

La méthode des rectangles est la moins précise comme on s'y attend. Elle propose une erreur beaucoup plus importante que les autres et même avec 10000 éléments (erreur = 10^{-2}), on atteint pas le niveau de précision des autres méthodes avec 250 segments. Celle de Simpson, déjà très précise, a une erreur qui varie très peu (de l'ordre de 10^{-8}), elle converge dès le début. La méthode des trapèzes converge très rapidement également (erreur de 10^{-5}). On remarque pour toutes les méthodes, que la convergence avec le codage en Python et en Numpy sont très similaires (les courbes sont superposées).

On observe que la convergence est difficile à obtenir pour la méthode de Simpson. On observe même une divergence dans le cas de la fonction python. En effet l'accumulation de plusieurs petites erreurs avec l'augmentation du nombre de segments crée une augmentation de l'erreur globale.