TOP 08

MÉTHODE DES TRAPÈZES POUR LE CALCUL APPROCHÉ D'INTÉGRALE.

Les fichiers TOP sont réservées aux étudiants qui préparent le Top 5. Ils sont plus difficiles et demandent déjà une bonne maîtrise du reste du programme (cours, exercices, TD et méthodes). Même si le contenu de ces exercices dépasse le cadre du programme de ECG2, ils peuvent inspirer une série de questions d'un texte de concours.

Cet exercice concerne le chapitre 07 (intégration) et les fonctions Python de base.

Exercice 1.-
Dans cet exercice, on discute deux méthodes de calcul numérique d'une intégrale et on compare leur efficacité. La première méthode, dite méthode des rectangles consiste à approcher une intégrale par ses sommes de Riemann ; la seconde dite, méthode des trapèzes consiste à approcher une intégrale par une somme d'aires de trapèzes.

Partie I : méthode des rectangles.

1. Nous avons vu dans le cours une méthode d'intégration approchée par les sommes de Riemann. Vérifier que la méthode du cours est équivalente au programme suivant (noter en particulier la présence de la fonction linspace qui permet de discrétiser un intervalle $[a, b]$ en n points répartis uniformément).

```
import numpy as np

def rectangle_gauche(f, a ,b, n) :
  h = (b-a)/(n-1)
  x = np.linspace(a, b, n)
  y = [f(t) for t in x]
  r = h * np.sum(y)
  return r
```

2. Pourquoi a-t-on appelé cette fonction rectangle_gauche ? En s'inspirant de la fonction ci-dessus, écrire une fonction rectangle_droite qui donne une autre approximation de l'intégrale de f.
3. Définir en Scilab la fonction réelle $\text{truc}(t) = \frac{4}{1+t^2}$. Trouver, grâce aux fonctions précédentes une valeur approchée de son intégrale sur $[0, 1]$. Comparer la valeur obtenue avec le nombre π.
4. **Difficile.** On suppose ici que f est de classe $\mathcal{C}^1$ et on note $I(a, b, f)$ l'intégrale de f sur $[a, b]$ et $R_g(f, a, b, n) =$ rectangle_gauche(f, a, b, n) [1]. Avec le théorème des accroissements finis, montrer que

$$|I(f, a, b) - R_g(f, a, b, n)| \leqslant \frac{(b-a)^2}{2(n-1)} M,$$

1. On définit de même R_d avec la fonction qui approche l'intégrale par les rectangles droits

où M est une constante dont on justifiera l'existence. On a la même formule pour R_d (la preuve est identique).

Partie II : méthode des trapèzes.

Dans cette partie, l'intégrale est approchée par la somme des aires des trapèzes obtenus en reliant les points $(x_i, f(x_i))$ situés sur la courbe de f, ce qui donne la formule d'approximation de l'intégrale par

$$T(f,a,b,n) = \frac{b-a}{n}\sum_{k=0}^{n-1}\frac{f\left(a+k\frac{b-a}{n}\right)+f\left(a+(k+1)\frac{b-a}{n}\right)}{2}$$

1. Faire un dessin de ce que représente $T(f,a,b,n)$.
2. Écrire en Scilab une fonction trapeze(f,a,b,n) qui permet de calculer l'intégrale approchée de f sur $[a,b]$ par cette méthode.
3. Montrer que $T(f,a,b,n) = \frac{1}{2}\left(R_g(f,a,b,n) + R_d(f,a,b,n)\right)$ et en déduire une autre façon de programmer trapeze.
4. **Encore plus difficile.** On suppose ici que f est de classe $\mathcal{C}^2$. Montrer que

$$|I(f,a,b) - T(f,a,b,n)| \leqslant \frac{(b-a)^2}{12(n-1)^2}M,$$

où M est une constante dont on justifiera l'existence.

Partie III : conclusion.

À l'aide des questions **4.** des parties **I.** et **II.**, quelle méthode semble la meilleure pour calculer une intégrale approchée. Vérifier votre hypothèse sur des exemples. [2]

Pour que cet exercice soit profitable, il n'est pas recommandé d'utiliser la correction trop tôt. La *recherche* de la solution a beaucoup plus d'intérêt que la solution elle-même.

Correction 1.-
Partie I : méthode des rectangles.

1. Déjà vu en cours.
2. On construit dans la fonction précédente les rectangles "par la gauche" c'est-à-dire que la hauteurs de chaque rectangle est donnée par la valeur de la fonction à gauche de l'intervalle. Pour les rectangles à droite, on écrirait plutôt :

```
import numpy as np

def rectangle_droite(f, a ,b, n) :
  h = (b-a)/(n-1)
  x = np.linspace(a+1/n, b+1/n, n)
  y = [f(t) for t in x]
  r = h * np.sum(y)
  return r
```

3.
```
def truc(t) :
  return 4/(1+t*t)


rectangle_gauche(truc,0,1,10000)
```

La valeur est très proche de π.

2. Il existe une méthode encore meilleure pour le calcul approché d'intégrale. La technique consiste cette fois à approcher l'intégrale par une somme d'aires sous la courbe de certaines paraboles : voir la page Wikipédia

4. On a

$$\begin{aligned}
I(f,a,b)-R_g(f,a,b,n+1) &= \int_a^b f(t)dt - \frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a+k\frac{(b-a)}{n}\right)\\
&= \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} f(t)dt - \frac{b-a}{n}\sum_{k=0}^{n-1} f\left(a+k\frac{(b-a)}{n}\right)\\
&= \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} f(t)dt - \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} f\left(a+k\frac{b-a}{n}\right)dt\\
&= \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} \left[f(t)-f\left(a+k\frac{b-a}{n}\right)\right]dt
\end{aligned}$$

Puis en passant à la valeur absolue et en appliquant l'inégalité triangulaire

$$|I(f,a,b)-R_g(f,a,b,n+1)| \leqslant \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} \left|f(t)-f\left(a+k\frac{b-a}{n}\right)\right|dt$$

On applique maintenant le théorème de accroissements finis sur chaque intervalle $\left[a+k\frac{b-a}{n}, a+(k+1)\frac{b-a}{n}\right]$ de longueur $\frac{b-a}{n}$ Puisque la fonction f est $\mathcal{C}^1$, sa dérivée (est continue et donc) majorée par une constante M. On a

$$\left[f(t)-f\left(a+k\frac{b-a}{n}\right)\right] \leqslant \frac{b-a}{n}M.$$

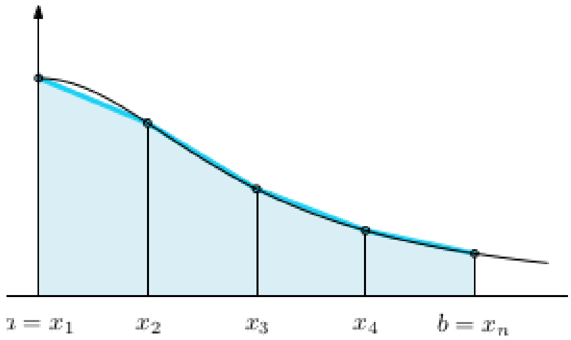
Il reste à intégrer ces inégalités : par croissance de l'intégrale, on a

$$|I(f,a,b)-R_g(f,a,b,n+1)| \leqslant \sum_{k=0}^{n-1}\int_{a+k\frac{b-a}{n}}^{a+(k+1)\frac{b-a}{n}} \frac{b-a}{n}M dt = \frac{(b-a)^2}{n}M.$$

L'inégalité de l'énoncé s'en déduit en modifiant légèrement la constante M (ou en apppliquant légèrement différemment le théorème des accroissements finis) et en remplaçant n par $n-1$. Attention à l'indice n : dans les questions **1.** et **2.**, on a découpé l'intervalle $[a,b]$ en n points seuleument alors qu'ici, on a découpé l'intervalle en $n+1$ points.

Partie II : méthode des trapèzes.

1.



2.

```python
import numpy as np

def trapeze(f, a ,b, n):
  h = (b-a)/(n-1)
  x1 = np.linspace(a, b, n)
  y1 = [f(t)/2 for t in x1]
  x2 = np.linspace(a+1/n, b+1/n, n)
  y2 = [f(t)/2 for t in x1]
  r = h * (np.sum(y1)+np.sum(y2))
  return r
```

3. On a

$$\begin{aligned} T(f,a,b,n) &= \frac{b-a}{n}\sum_{k=0}^{n-1}\frac{f\left(a+k\frac{b-a}{n}\right)+f\left(a+(k+1)\frac{b-a}{n}\right)}{2} \\ &= \frac{b-a}{2n}\left(\sum_{k=0}^{n-1} f\left(a+k\frac{b-a}{n}\right)+\sum_{k=0}^{n-1} f\left(a+(k+1)\frac{b-a}{n}\right)\right) \\ &= \frac{1}{2}\left(R_g(f,a,b,n)+R_d(f,a,b,n)\right) \end{aligned}$$

On en déduit que

```python
def trapeze(f,a,b,n)
  return (rectangle_gauche(f,a,b,n)+rectangle_droite(f,a,b,n))/2
```

donne le même résultat.

4. C'est une question vraiment difficile que je vous conseille d'admettre. Pour les plus courageux, l'explication est donnée par exemple dans ce texte : http://bruneau.u-cergy.fr/enseignement/L3ananum/cours-analyse_numerique_2014.pdf. Il s'agit de faire une double intégration par parties intelligemment.

Conclusion.

On en déduit que pour les grandes valeurs de n, l'erreur commise dans le calcul de l'intégrale est plus faible lorsqu'on la remplace par la somme des aires des trapèzes que par la somme des aires des rectangles. Cela provient de fait que

$$\frac{1}{(n-1)^2} \underset{n\to+\infty}{=} \mathbf{o}\left(\frac{1}{n-1}\right).$$

Il est donc plus économique de calculer l'intégrale d'une fonction par la méthode des trapèzes.