MPSI 2017-2018

Informatique

TP 5 : Calcul approché d'intégrales.

Partie I. Implémentation

1. Réécrire la fonction `rectangles_g(f,a,b,n)`, prenant pour argument deux flottants `a` et `b`, une fonction `f` et un entier `n`, et calculant l'approximation de $\int_a^b f$ donnée, par la méthode des rectangles à gauche avec `n` rectangles.
2. Écrire de même les fonctions `rectangles_d(f,a,b,n)` (rectangles à droite) et `trapezes(f,a,b,n)` (trapèzes).

Partie II. Analyse de l'erreur commise

1. Que vaut $\int_0^1 e^x dx$?
2. Calculer l'erreur commise (en valeur absolue) lorsqu'on approche cette intégrale à l'aide de la fonction `rectangles_g` pour les valeurs de n suivantes : $n = 10$, $n = 100$, $n = 1000$, $n = 10000$, $n = 100000$, $n = 1000000$.
3. Faire de même pour les fonctions `rectangles_d(f,a,b,n)` et `trapezes(f,a,b,n)`.
4. Quels théorèmes mathématiques admis semblent confirmés par ces tests ?

Partie III. Analyse graphique

On souhaite observer plus visuellement cette confirmation sur cet exemple et d'autres. On rappelle qu'on peut tracer des courbes (qu'on approche par des lignes brisées) à l'aide des fonctions `plot` et `show` de la bibliothèque `mathplotlib.pyplot`).

1. Exécutez-le programme suivant. Que fait-il ?

```
x = [i/100 for i in range(0,200)]
y = [x[i]**2 for i in range(0,200)]
plot(x, y)
show()
```

2. Écrire une fonction `erreur_commise_exp_recg(n1,n2)` qui trace la courbe de la fonction donnant <u>l'inverse</u> de l'erreur commise sur l'exemple de la partie II, en fonction de $n \in [n_1, n_2 - 1]$ lorsqu'on utilise la méthode des rectangles à gauche.

3. Récidiver avec `erreur_commise_exp_trap(n1,n2)` pour les trapèzes. Tester. Convainquant ?

4. Écrire `erreur_commise_recg(f,a,b,theorique,n1,n2)` et `erreur_commise_trap(f,a,b,theorique,n1,n2)` faisant la même chose mais pour une fonction f quelconque intégrée sur un intervalle $[a, b]$ quelconque, étant entendu que l'utilisateur doit donner la valeur exacte de `theorique` $= \int_a^b f$. Tester. Convainquant ?

Partie IV. Fonction d'erreur

La *fonction d'erreur* notée *erf* est une fonction importante en analyse et en probabilités. Elle ne peut pas s'exprimer à l'aide de sommes/différences/produits/quotients/composées de fonctions usuelles [1] mais est définie par une intégrale :

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

On se propose d'écrire une fonction `erf(x)` qui prend comme argument un flottant `x` et retourne l'approximation de $\operatorname{erf}(x)$ calculée à l'aide de la méthode des trapèzes pour $n = 100\lfloor x + 1 \rfloor$.

1. Pourquoi $100\lfloor x + 1 \rfloor$ plutôt que, par exemple, 100000 ?

2. Écrire une fonction `erf(x)` qui prend en argument un flottant `x` et retourne l'approximation de $erf(x)$ calculée comme proposé.

3. Écrire une fonction `trace_erf()` qui ne prend pas d'argument et trace la courbe représentative de erf. Un tracé sur $[-5, 5]$ est largement suffisant : pourquoi ?
   Attention, il y a au moins deux façons de procéder. L'une des deux est bien plus pertinente que l'autre !

1. Ça se démontre. Mais c'est dur.