13. Calcul approché d'une intégrale

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.
Le but du TP est de calculer une valeur approchée de l'intégrale $I = \displaystyle\int_0^1 \frac{4}{1+x^2}\,\mathrm{d}x$.

1. Calculer la valeur exacte de l'intégrale I.

2. Créer en langage Python une fonction de la variable n nommée *subdivision* permettant de créer une liste de $n+1$ flottants de 0 à 1 en progression arithmétique.
   (par exemple pour l'entrée $n = 4$ la fonction doit renvoyer la liste $[0,\ 0.25,\ 0.5,\ 0.75,\ 1]$)

3. Créer en langage Python une fonction de la variable l nommée *image* permettant d'obtenir la liste des images par la fonction $f : x \mapsto \dfrac{4}{1+x^2}$ des valeurs d'une liste l.
   (par exemple pour l'entrée $l = [1,\ 2,\ 3]$ la fonction doit renvoyer la liste $[2.0,\ 0.8,\ 0.4]$)

4. Créer en langage Python une fonction des variables l, a et b nommée *somme* permettant de calculer la somme des valeurs d'indices compris entre a et b d'une liste l.
   (par exemple pour l'entrée $l = [0,\ 1,\ 2,\ 3,\ 4]$, $a = 1$ et $b = 3$, la fonction doit renvoyer la valeur 6)

5. Utiliser les fonctions *subdivision*, *image* et *somme* pour créer en langage Python une fonction de la variable n nommée *rectangle* permettant de calculer $\displaystyle\frac{1}{n}\sum_{k=0}^{k=n-1} f\left(\frac{k}{n}\right)$.
   Vérifier que $rectangle(100)$ approche I à 10^{-2} près.

6. Utiliser les fonctions *subdivision*, *image* et *somme* pour créer en langage Python une fonction de la variable n nommée *trapeze* permettant de calculer $\displaystyle\frac{1}{n}\left(\frac{f(0)+f(1)}{2} + \sum_{k=1}^{k=n-1} f\left(\frac{k}{n}\right)\right)$.
   Vérifier que $trapeze(100)$ approche I à 10^{-4} près.

7. Créer en langage Python une fonction des variables l, a et b nommée *pseudosomme* permettant de calculer la somme des valeurs d'indices compris entre a et b d'une liste l en comptant double celles d'indices impairs.

8. Utiliser les fonctions *subdivision*, *image* et *pseudosomme* pour créer en langage Python une fonction de la variable n nommée *simpson* permettant de calculer $\displaystyle\frac{1}{3n}\left(\frac{f(0)+f(1)}{2} + 2\sum_{k=0}^{k=n-1} f\left(\frac{2k+1}{2n}\right) + \sum_{k=1}^{k=n-1} f\left(\frac{2k}{2n}\right)\right)$.
   Vérifier que $simpson(50)$ approche I à 10^{-13} près.

Réponses

1) $I = 4\arctan(1) - 4\arctan(0) = \pi$.

2)
```
def subdivision(n):
    '''retourne une liste de n+1 flottants de 0 à 1 en progression arithmétique'''
    l=[]
    for k in range(0,n+1):
        l.append(k/n)
    return(l)
```

3)
```
def image(l):
    '''retourne les images des éléments de l par la fonction x->4/(1+x^2)'''
    L=[]
    for k in range(0,len(l)):
        L.append(4/(1+l[k]**2))
    return(L)
```

4)
```
def somme(l,a,b):
    '''calcul de la somme des valeurs de l d'indices compris entre a et b'''
    s=0
    for k in range(a,b+1):
        s=s+l[k]
    return(s)
```

5)
```
def rectangle(n):
    '''calcul d'une valeur approchée de pi par la méthode des n rectangles'''
    x=subdivision(n)
    y=image(x)
    s=somme(y,0,n-1)
    return(s/n)
```

6)
```
def trapeze(n):
    '''calcul d'une valeur approchée de pi par la méthode des n trapèzes'''
    x=subdivision(n)
    y=image(x)
    s=somme(y,1,n-1)
    return((s+(y[0]+y[n])/2)/n)
```

7)
```
def pseudosomme(l,a,b):
    '''calcul de la somme des valeurs de l d'indices compris entre a et b
    en comptant double les valeurs d'indices impairs'''
    s=0
    for k in range(a,b+1):
        if k%2==0:
            s=s+l[k]
        else:
            s=s+2*l[k]
    return(s)
```

8)
```
def simpson(n):
    '''calcul d'une valeur approchée de pi par la méthode de simpson'''
    x=subdivision(2*n)
    y=image(x)
    s=pseudosomme(y,1,2*n-1)
    return((s+(y[0]+y[2*n])/2)/(3*n))
```