COURS N°4 PYTHON

L2 MATHÉMATIQUES

Application aux mathématiques

Vue pour impression

Auteur : Antoine Laurent

INTÉGRATION

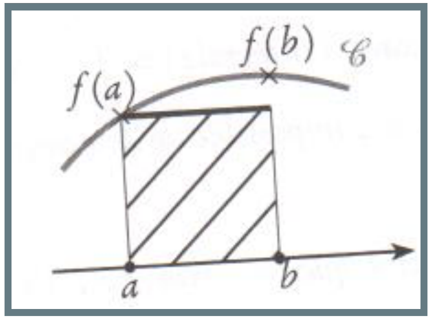
- Pour calculer de manière exacte l'intégrale $I = \int_a^b f x \, \mathrm{d}x$ on cherche une primitive F de f
- Il est souvent difficile, voire impossible de déterminer une primitive d'une fonction donnée
- On va voir trois méthodes pour évaluer l'intégrale de la fonction : $I = \int_0^1 \sqrt{1 - x^2} \, \mathrm{d}x$

INTÉGRATION

- On définit sur l'intervalle $[0; 1]$, la fonction f par : $f(x) = \sqrt{1 - x^2}$
- Il s'agit donc de calculer l'aire du domaine associé à f sur $[0; 1]$
- On découpe l'intervalle $[0; 1]$ en n intervalles $[\frac{k}{n}; \frac{k+1}{n}]$ pour $n >= 2$ et $0 <= k <= n - 1$
- On remplace la courbe C de la fonction par :
  - un segment horizontal (méthode des rectangles),
  - un segment de droite (méthode des trapèzes),
  - un morceau de parabole (méthode de Simpson)

MÉTHODE DES RECTANGLES

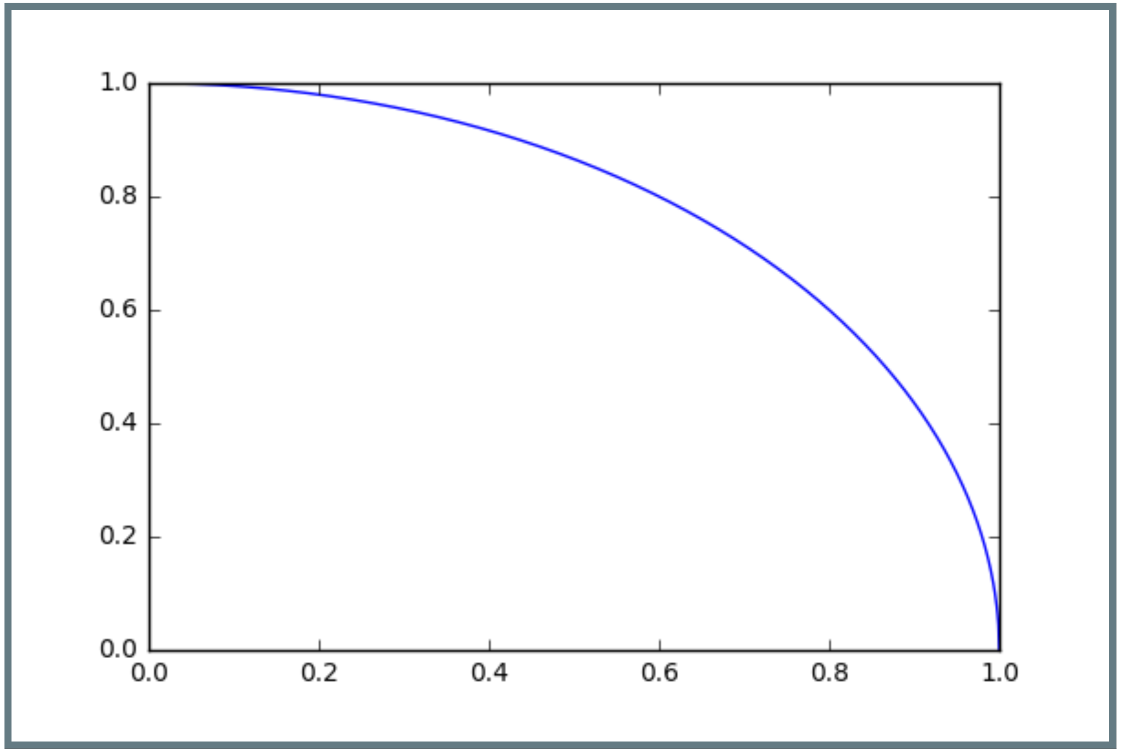
- Soit f la fonction continue et positive sur $[a, b]$ et C sa courbe représentative.
- Pour estimer $I = \int_a^b fx \, \mathrm{d}x$, on peut remplacer C sur $[a, b]$ par la droite d'équation $y = f(a)$



- Le rectangle obtenu a pour largeur $(b - a)$ et pour hauteur $f(a)$, donc l'aire du rectangle est $(b - a) \times f(a)$.
- Si on approche l'intégrale par l'aire du rectangle, alors $I \approx (b - a) \times f(a)$

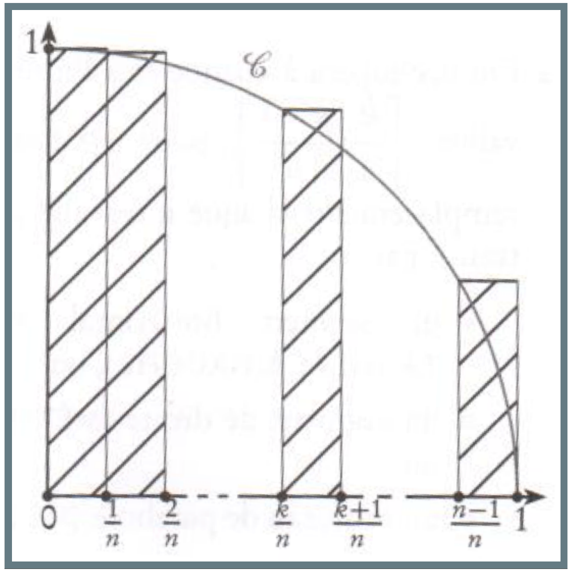
MÉTHODE DES RECTANGLES (2)

- Considérons maintenant la fonction f définie sur $[0;\, 1]$ par $f(x) = \sqrt{1 - x^2}$
- Quel est, par considération géométrique, sans le calcul, la valeur exacte de $I = \int_0^1 \sqrt{1 - x^2}\, \mathrm{d}x$?



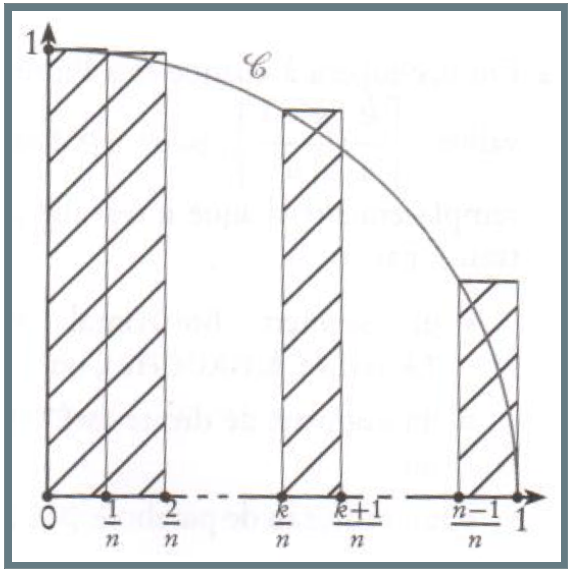
MÉTHODE DES RECTANGLES (3)

- I correspond à l'aire du quart de disque de centre 0 et de rayon 1, donc $I = \pi/4$
- f étant continue et positive sur $[0;1]$, $I = \int_{\frac{k}{n}}^{\frac{k+1}{n}} f(x)\,\mathrm{d}x$ peut être approchée par l'aire d'un rectangle de largeur $\frac{1}{n}$ et de hauteur $f(\frac{k}{n})$, soit $\frac{1}{n} \times \sqrt{1 - \frac{k^2}{n^2}}$, soit $\frac{1}{n^2} \times \sqrt{n^2 - k^2}$



MÉTHODE DES RECTANGLES (4)

- $I = \int_0^1 f(x)\,\mathrm{d}x$ peut être approximée par la somme des aires des rectangles :
  - $R_n = \frac{1}{n^2} \sum_{k=0}^{n-1} \sqrt{n^2 - k^2}$
- R_n approche par excès I



IMPLÉMENTATION EN PYTHON

- Définir la fonction f

```
from math import *
def f(x):
    return sqrt(1-x**2)
```

- Écrire une fonction rect ayant pour paramètres f, n pour calculer et afficher R_n

```
def rect(f,n):
    R=0
    t=0
    h=1/n
    while t<1: #on arrive à h
        R = R+h*f(t)
        t=t+h
    return R
```

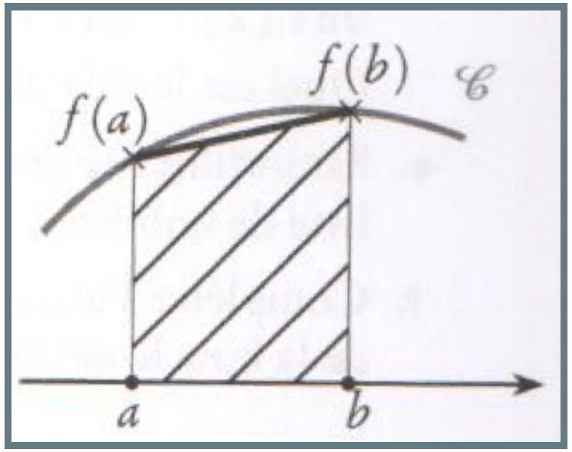
RÉSULTAT

```
for k in range(8):
    approx = rect(f, 10**k)
    print("k=%i, I=%f, I_approx=%f, Diff=%.12f" % (k, pi/4, approx,
    approx-pi/4))

#Resultat:
k=0, I=0.785398, I_approx=1.000000, Diff=0.214601836603
k=1, I=0.785398, I_approx=0.826130, Diff=0.040731419655
k=2, I=0.785398, I_approx=0.790104, Diff=0.004706094547
k=3, I=0.785398, I_approx=0.785889, Diff=0.000490703330
k=4, I=0.785398, I_approx=0.785448, Diff=0.000049706049
k=5, I=0.785398, I_approx=0.785403, Diff=0.000004990724
k=6, I=0.785398, I_approx=0.785399, Diff=0.000000499703
k=7, I=0.785398, I_approx=0.785398, Diff=0.000000050142
```

MÉTHODE DES TRAPÈZES

- Soit f la fonction continue et positive sur $[a, b]$ et C sa courbe représentative.
- Pour estimer $I = \int_a^b fx \,\mathrm{d}x$, on peut remplacer C sur $[a, b]$ par le segment $[AB]$ où $A(a, f(a)$ et $B(b, f(b)$



- L'aire du trapèze est :
  - base moyenne * hauteur
  - $\frac{f(a)+f(b)}{2} \times (b - a)$
  - Soit $I \approx \frac{f(a)+f(b)}{2} \times (b - a)$

MÉTHODE DES TRAPÈZES (2)

- Considérons maintenant la fonction f définie sur $[0;\,1]$ par $f(x) = \sqrt{1-x^2}$
- $I = \int_{\frac{k}{n}}^{\frac{k+1}{n}} f(x)\,\mathrm{d}x$ peut être approximée par l'aire d'un trapèze de hauteur $\frac{1}{n}$ et de base moyenne $\frac{\sqrt{n^2-(k+1)^2}+\sqrt{n^2-k^2}}{2n}$
- $I = \int_0^1 f(x)\,\mathrm{d}x$ peut être approximée par la somme des aires des trapèzes, soit $T_n = \frac{1}{2n^2}\sum_{k=0}^{n-1} \sqrt{n^2-(k+1)^2} + \sqrt{n^2-k^2}$

IMPLÉMENTATION PYTHON

- Construire une fonction trap avec pour paramètres la fonction f et le nombre de découpages, n.

```
from math import *
def trap(f,n):
    T=0
    h=1/n
    L=np.linspace(0,1,n, endpoint=False) #sans le dernier element
    #ou
    #L=np.linspace(0,1,n+1)[:-1]
    for t in L:
        T=T+(f(t)+f(t+h))*h/2
    return T
```

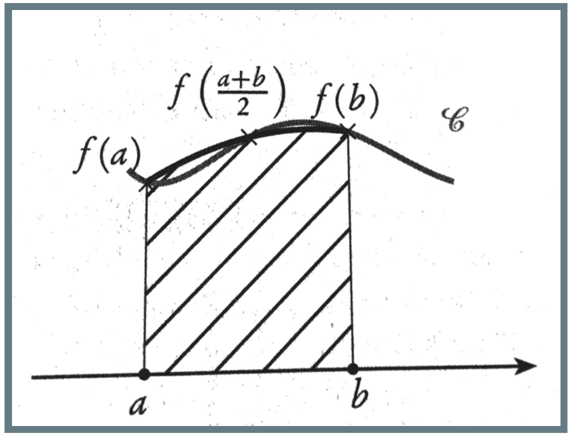
RÉSULTAT

```
for k in range(8):
    approx = trap(f, 10**k)
    print("k=%i, I=%f, I_approx=%f, Diff=%.12f" % (k, pi/4, approx,
    approx-pi/4))

#resultat
k=0, I=0.785398, I_approx=0.500000, Diff=-0.285398163397
k=1, I=0.785398, I_approx=0.776130, Diff=-0.009268581835
k=2, I=0.785398, I_approx=0.785104, Diff=-0.000293905453
k=3, I=0.785398, I_approx=0.785389, Diff=-0.000009296670
k=4, I=0.785398, I_approx=0.785398, Diff=-0.000000293995
k=5, I=0.785398, I_approx=0.785398, Diff=-0.000000009297
k=6, I=0.785398, I_approx=0.785398, Diff=-0.000000000294
k=7, I=0.785398, I_approx=0.785398, Diff=-0.000000000009
```

MÉTHODE DE SIMPSON

- Soit f la fonction continue et positive sur $[a, b]$ et C sa courbe représentative.
- Pour estimer $I = \int_a^b f x \, \mathrm{d}x$, on peut remplacer C sur $[a, b]$ par une parabole qui passe par les points de C d'abscisses a, b, et $m = \frac{a+b}{2}$ respectivement.



- $I \approx \frac{b-a}{6}(f(a) + 4f(\frac{a+b}{2}) + f(b))$

MÉTHODE DE SIMPSON (4)

- Considérons maintenant la fonction f définie sur $[0;\, 1]$ par $f(x) = \sqrt{1 - x^2}$
- $I = \int_{\frac{k}{n}}^{\frac{k+1}{n}} f(x)\,\mathrm{d}x$ peut être approximée par l'aire sous la courbe :

$$I = \int_{\frac{k}{n}}^{\frac{k+1}{n}} f(x)\,\mathrm{d}x \approx \frac{1}{6n}(f(\frac{k}{n}) + 4f(\frac{2k+1}{2n}) + f(\frac{k+1}{n}))$$

Soit entre 0 et 1 par la somme :

$$I \approx \frac{1}{6n} \sum_{k=0}^{n-1}(f(\frac{k}{n}) + 4f(\frac{2k+1}{2n}) + f(\frac{k+1}{n}))$$

IMPLÉMENTATION EN PYTHON

```
def simps(f,n):
    S=0
    h=1/n
    L=np.linspace(0,1,n, endpoint=False) #sans le dernier element
    for t in L:
        S=S+(f(t)+4*f(t+h/2)+f(t+h))*h/6
    return S
```

RÉSULTAT

```
for k in range(8):
    approx = simps(f, 10**k)
    print("k=%i, I=%f, I_approx=%f, Diff=%.12f" % (k, pi/4, approx,
    approx-pi/4))

#Résultat :
k=0, I=0.785398, I_approx=0.744017, Diff=-0.041381227541
k=1, I=0.785398, I_approx=0.784112, Diff=-0.001286397333
k=2, I=0.785398, I_approx=0.785358, Diff=-0.000040601099
k=3, I=0.785398, I_approx=0.785397, Diff=-0.000001283669
k=4, I=0.785398, I_approx=0.785398, Diff=-0.000000040592
k=5, I=0.785398, I_approx=0.785398, Diff=-0.000000001284
k=6, I=0.785398, I_approx=0.785398, Diff=-0.000000000041
k=7, I=0.785398, I_approx=0.785398, Diff=-0.000000000001
```

COMPARAISON DES 3 MÉTHODES

```
for k in range(8):
    approxR = rect(f, 10**k)
    approxT = trap(f, 10**k)
    approxS = simps(f, 10**k)
    print("k=%i, Diffs=%.12f / %.12f / %.12f" % (k, approxR-pi/4,
    approxT-pi/4, approxS-pi/4))

#Résultat :
k=0, Diffs=0.214601836603 / -0.285398163397 / -0.041381227541
k=1, Diffs=0.040731419655 / -0.009268581835 / -0.001286397333
k=2, Diffs=0.004706094547 / -0.000293905453 / -0.000040601099
k=3, Diffs=0.000490703330 / -0.000009296670 / -0.000001283669
k=4, Diffs=0.000049706049 / -0.000000293995 / -0.000000040592
k=5, Diffs=0.000004990724 / -0.000000009297 / -0.000000001284
k=6, Diffs=0.000000499703 / -0.000000000294 / -0.000000000041
k=7, Diffs=0.000000050142 / -0.000000000009 / -0.000000000001
```

ON TRACE LES ÉCARTS PAR RAPPORT À $\frac{\pi}{4}$

- On vectorize nos fonctions (voir notebook)