

# INTÉGRATION : DEUX AUTRES MÉTHODES

I. Méthode Monte-Carlo du rejet	2
II. Méthode de l'espérance	3
III. Approfondissement : comparaison de méthodes	3

Pour calculer l'aire sous une courbe d'une fonction  $f$ , nous avons vu la méthode de Riemann qui consiste à encadrer l'aire par deux séries de rectangles (borne inférieure et borne supérieure).

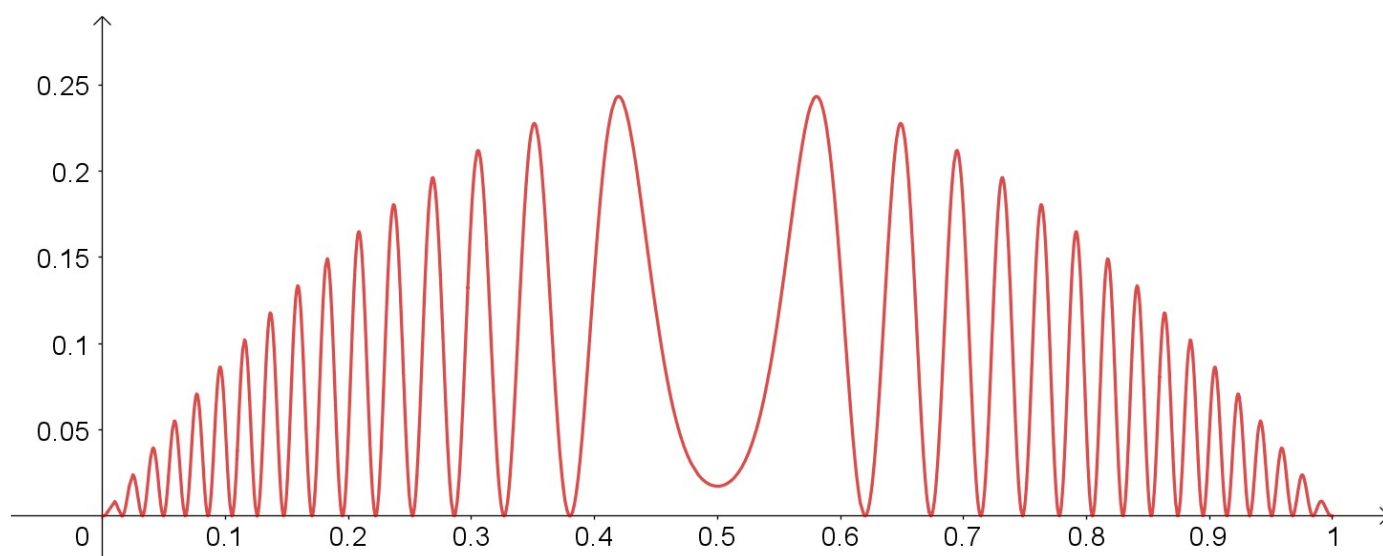
Nous avons aussi envisagé la « méthode des trapèzes »...

Ces deux méthodes partent de l'hypothèse que l'aire à calculer n'est pas trop « tourmentée » et que la surface peut être facilement découpée en rectangles ou trapèzes. Si l'aire est telle qu'il devient très compliqué d'utiliser la méthode des rectangles ou des trapèzes, on a souvent recours à une solution probabiliste : la méthode de Monte-Carlo, universellement exploitée dans le monde de la simulation pour sa puissance et son efficacité. On peut aussi l'utiliser pour simuler la désintégration d'une population de noyaux radioactifs, le refroidissement d'un verre, le comportement d'un gaz d'électrons, etc.

La méthode de Monte-Carlo est une méthode moins précise que la méthode des trapèzes : ce manque de précision est dû à l'usage obligatoire d'un générateur de nombres aléatoires, dont la qualité influe sur la précision du calcul. Il est en effet impossible de créer, avec un algorithme, une suite de nombres parfaitement aléatoire. On se contente donc de générer des suites pseudo-aléatoires avec un ordinateur ou une calculatrice. Une suite de nombres pseudo-aléatoire est une suite périodique, de période aussi grande que possible, dont les nombres, à l'intérieur de chaque période, sont aussi peu corrélés entre eux que possible (c'est à dire le plus indépendants les uns des autres). Celles que l'on implémente de façon standard sur les calculateurs grand public ont une période de l'ordre de  $2^{32}$ , soit environ 4,3 milliards.

Dans cette activité, nous allons calculer des valeurs approchées de l'intégrale sur  $[0;1]$  de la fonction suivante :  $f(x) = x(1-x)\sin^2[200x(1-x)]$ .

La primitive d'une telle fonction semble incalculable, et la méthode de Riemann semble difficile à mettre en œuvre.



# I. Méthode Monte-Carlo du rejet

A partir de l'algorithme ci-contre, expliquer cette méthode.

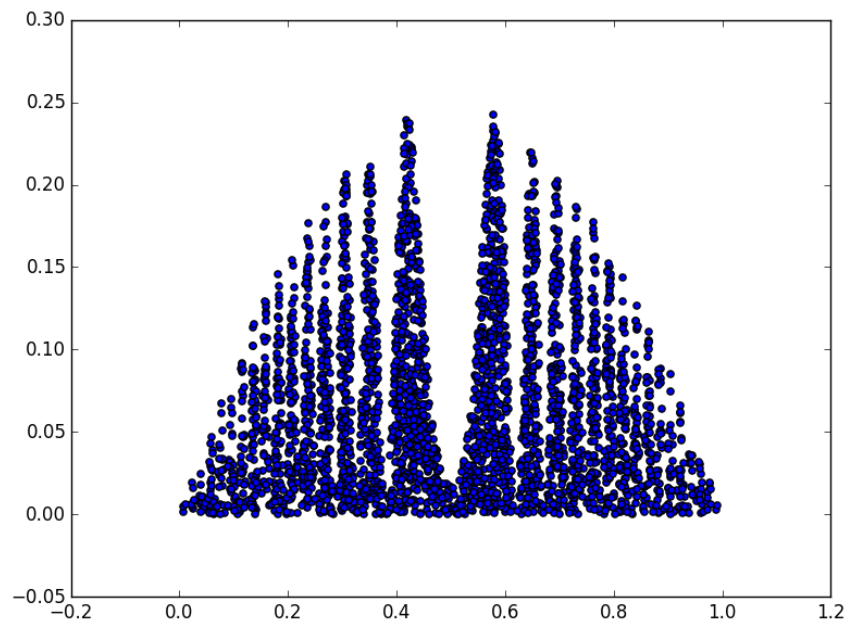
```
from random import *
from math import *
from numpy import *
from scipy.integrate import quad
import matplotlib.pyplot as plt

def mini(f,a,b):
    l=linspace(a,b,1000000)
    minimum = f(a)
    for i in l:
        if f(i)<minimum: minimum=f(i)
    return minimum

def maxi(f,a,b):
    l=linspace(a,b,1000000)
    maximum = f(a)
    for i in l:
        if f(i)>maximum: maximum=f(i)
    return maximum
```

```
def montecarlo(n,f,a,b):
    mi, ma = mini(f,a,b), maxi(f,a,b)
    eff=0
    x_list,y_list=[],[]
    for i in range(n):
        x=uniform(a,b)
        y=uniform(mi,ma)
        if y <= f(x):
            eff=eff+1
            x_list.append(x)
            y_list.append(y)
    plt.scatter(x_list,y_list)
    plt.savefig("MonteCarlo.png")
    airerectangle=(b-a)*(ma-mi)
    freq=eff/n
    phrase="Méthode Monte-Carlo rejet : "+str(freq*airerectangle)
    return phrase
```

```
N=int(input("Entrez n non nul : "))
a,b=float(input("Entrez a : ")),float(input("Entrez b : "))
f=lambda x:x*(1-x)*(sin(200*x*(1-x)))**2
print(montecarlo(N,f,a,b))
print("Avec quad (scipy) : ",quad(f,a,b))
```



## II. Méthode de l'espérance

DÉFINITION.

Soit  $f$  une fonction continue sur un intervalle  $[a; b]$ .

La **valeur moyenne de la fonction  $f$**  sur  $[a; b]$  est le réel  $\frac{1}{b-a} \int_a^b f(t) dt$ .

En notant  $\mu$  cette valeur moyenne et  $I$  l'intégrale entre  $a$  et  $b$  de  $f$ , on a donc :

$$\mu = \frac{1}{b-a} \times I \quad \text{donc} \quad I = (b-a) \times \mu.$$

Autrement dit, si on approche la valeur moyenne  $\mu$ , on aura une valeur approchée de  $I$ .

Pour approcher  $\mu$ , il suffit de se rappeler la définition de l'espérance :  $\int_a^b t f(t) dt$ . En subdivisant

l'intervalle  $[a; b]$  en  $N$  (le plus grand possible) morceaux, il suffira alors d'approcher cette intégrale par une somme finie :  $\sum_a^b t f(t)$ .

A l'aide de ces indications, écrire un programme Python qui calcule une valeur approchée de  $\int_a^b f(t) dt$  avec la méthode de l'espérance.

## III. Approfondissement : comparaison de méthodes

Écrire un programme Python qui :

- simule 1 000 000 de méthodes Monte-Carlo par rejet et calcule la moyenne des valeurs obtenues
- déterminer quelle méthode (rectangles / Monte-Carlo / espérance) donne la valeur approchée la plus proche de celle donnée par la fonction `quad`.

On rappelle l'algorithme de la méthode des rectangles (point médian) :

```
def rectangles(a, b, n, f):
    S = 0
    for i in range(0, n):
        x1 = a + (b - a)*i/n
        x2 = a + (b - a)*(i+1)/n
        S += f((x1 + x2)/2)*(x2 - x1)
    return S
```