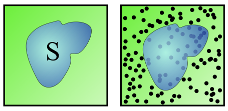
Méthode de Monte-Carlo

Introduction

La méthode de Monte-Carlo est une méthode d'approximation de l'aire d'une surface basée sur une approche probabiliste. On souhaite estimer l'aire de la surface S ci-dessous incluse dans un rectangle d'aire connue.



On considère l'expérience aléatoire consistant à définir N points aléatoires uniformément répartis sur la surface du rectangle. La probabilité qu'un point aléatoire soit inclus dans la surface S est alors $p = \dfrac{\textit{Aire de } S}{\textit{Aire du rectangle}}$.

Lorsque N prend de grandes valeurs, la fréquence observée des points inclus dans S est proche de p donc :

$$\text{le nombre } f_{\text{observée}} = \frac{\textit{Nombres de points inclus dans } S}{\textit{Nombre total de points}} \text{ est proche de } \frac{\textit{Aire de } S}{\textit{Aire du rectangle}}.$$

Une estimation de l'aire de S est donc donnée par : $\textit{Aire du rectangle} \times f_{\text{observée}}$.

Partie A Testons la méthode de Monte-Carlo dans un cas particulier

Nous allons pour cela estimer l'aire d'une surface simple par la méthode de Monte-Carlo et confronter l'approximation obtenue à la valeur exacte de l'aire que nous savons calculer.

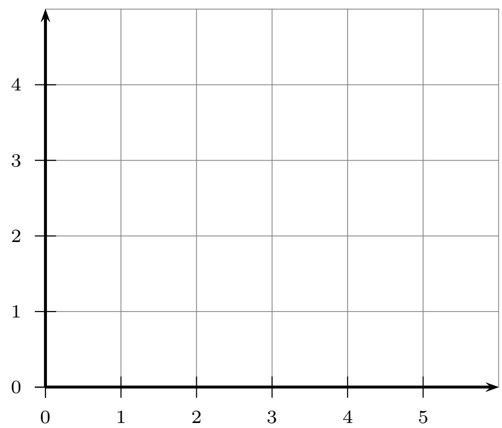
1. On considère la fonction f définie sur l'intervalle $[0;6]$ par $f(x) = \dfrac{1}{2}x$. Représenter la courbe C_f dans le repère orthonormé ci-contre.

2. Hachurer la surface S_1 délimitée par C_f, l'axe des abscisses et la droite d'équation $x = 6$.

3. Calculer l'aire de la surface hachurée.

....

4. Expliquer ce que fait la fonction **montecarlo()** dans le programme ci-dessous.



....

....

```python
from random import*

def f(x):
    return x/2

def montecarlo():
    x=uniform(0,6)      # abscisse aléatoire dans l'intervalle [0;6]
    y=uniform(0,5)      # ordonnée aléatoire dans l'intervalle [0;5]
    if y<f(x):
        return True
    else:
        return False
```

5. On souhaite appeler 1000 fois la fonction **montecarlo()** et compter le nombre de points inclus dans la surface étudiée. Compléter et saisir le programme ci-dessous :

```
from random import*

def f(x):
    return x/2

def montecarlo():
    x=uniform(0,6)      # abscisse aléatoire dans l'intervalle [0;6]
    y=uniform(0,5)      # ordonnée aléatoire dans l'intervalle [0;5]
    if y<f(x):
        return True
    else:
        return False

compteur=0

for i in range(1000):
    if montecarlo()== ..... :
        compteur= ..............
print(compteur/1000)
```

6. A quoi correspond l'affichage obtenu en sortie de programme ?

....

7. D'après l'introduction, une estimation de l'aire de S_1 est donnée par : *Aire du rectangle* $\times f_{\text{observée}}$.
Modifier le programme pour qu'il affiche une estimation de l'aire de S_1.

8. Calculer $|\text{valeur affichée} - 9|$ et interpréter le résultat.

....

....

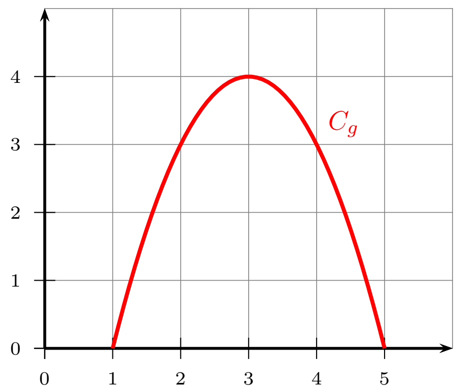
Partie B Application de la méthode dans un cadre général

On souhaite à présent appliquer la méthode de Monte-Carlo pour estimer l'aire d'une surface que nous ne savons pas calculer. On considère la fonction g définie sur $[1;5]$ par $g(x) = -x^2 + 6x - 5$.
On note S_2 la surface délimitée par C_g et l'axe des abscisses.

1. Hachurer la surface S_2.

2. Modifier le programme pour obtenir une estimation de l'aire de S_2.

3. Le programme ci-dessous permet d'afficher un point rouge et un bleu dans un repère. En vous inspirant de cet exemple, compléter le programme de la question 2 pour obtenir un graphique où les points inclus dans la surface S_2 sont affichés en rouge et les autres en bleu.




```
from matplotlib.pyplot import*          # module graphique

plot(1,2,'ro')       # point de coordonnées (1;2) motif 'o' de couleur rouge
plot(2,3,'bo')       # point de coordonnées (2;3) motif 'o' de couleur bleue

savefig("figure")        # sauvegarde et affichage du graphique
```