

# monteCarlo

Documents d'accompagnement  
Algorithmique et programmation  
nouveau programme du lycée 2019

## 1 Approximation du nombre $\pi$ par la méthode de Monte-Carlo

### 1.1 Présentation de l'activité

- **Niveau de classe :**
- Classe de première de la voie générale (spécialité mathématiques)
- Classe de première de la voie technologique (tronc commun).
- **Référence au programme :**
- Spécialité mathématiques de première générale: *Méthode de Monte Carlo : estimation du nombre  $\pi$ .*
- **Description :** Activité de découverte d'une méthode probabiliste d'estimation : la méthode de Monte-Carlo. Le but ici est d'estimer l'aire du quart de disque de rayon 1 afin d'obtenir une approximation du nombre  $\pi$ .

### 1.2 Objectifs

Il est courant d'utiliser des nombres aléatoires pour estimer des quantités numériques. Nous allons ici utiliser des variables aléatoires de loi uniforme afin d'estimer le nombre  $\pi$ . Pour cela nous allons: - générer des points aléatoirement et uniformément dans un carré de côté 1; - calculer la proportion de points situés en dessous du quart de cercle de rayon 1. Cela nous permettra de donner une estimation de son aire; - approcher le nombre  $\pi$  sachant que l'aire d'un quart de disque de rayon 1 est  $\frac{\pi}{4}$ .

### 1.3 Estimation

La fonction `random` de la librairie `random` permet de générer un nombre aléatoirement et uniformément entre 0 et 1.

---

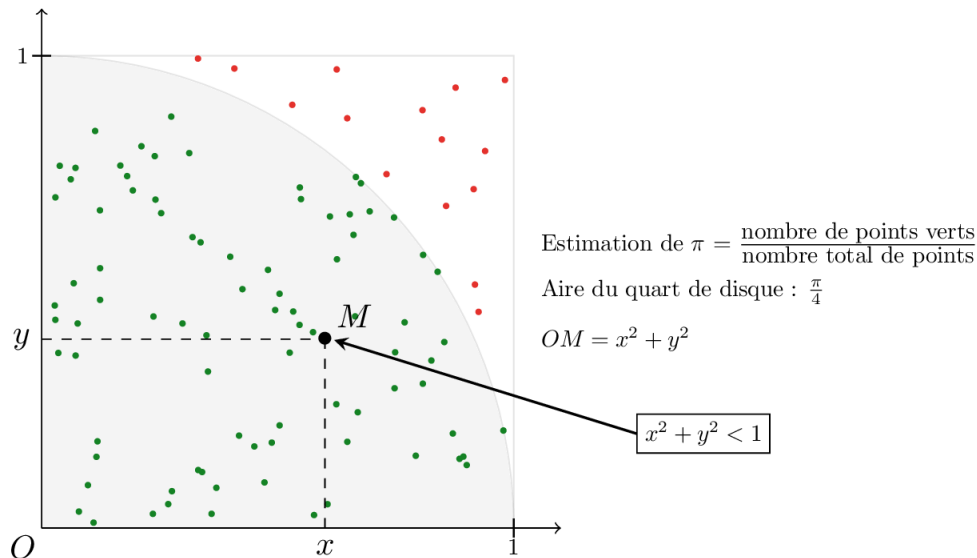
```
from random import random as rd

nbPoints = 1000

s = 0
for i in range(nbPoints):
    x,y = rd(),rd()
    if x**2+y**2 < 1:
        s=s+1
estimPi = 4*s/nbPoints

print(estimPi)
```

---



Monte Carlo

3.108

Suggestions pédagogiques

- **Expliquer un programme**
- Que représente la variable  $s$  ?
- Expliquer la ligne 8.
- **Compléter un programme**
  - Le programme précédent étant fourni en remplaçant la ligne 8 par `if ...`, demander aux élèves de compléter la ligne 8.
  - Le programme précédent étant fourni en remplaçant la ligne 10 par `estimPi = ...`, demander aux élèves de compléter la ligne 10.
- **Écrire un programme**  
 Écrire un programme générant des points de coordonnées aléatoires entre 0 et 1 et comptant la proportion de points situés dans le disque de centre  $(0,0)$  et de rayon 0,5
- **Tester** l'estimation du nombre  $\pi$  pour différents nombres de points générés.

## 1.4 Convergence

Nous allons maintenant utiliser les listes afin de stocker au fur et à mesure des estimations de  $\pi$  obtenues en faisant varier le nombre de points générés.

---

```

%matplotlib inline
import matplotlib.pyplot as plt

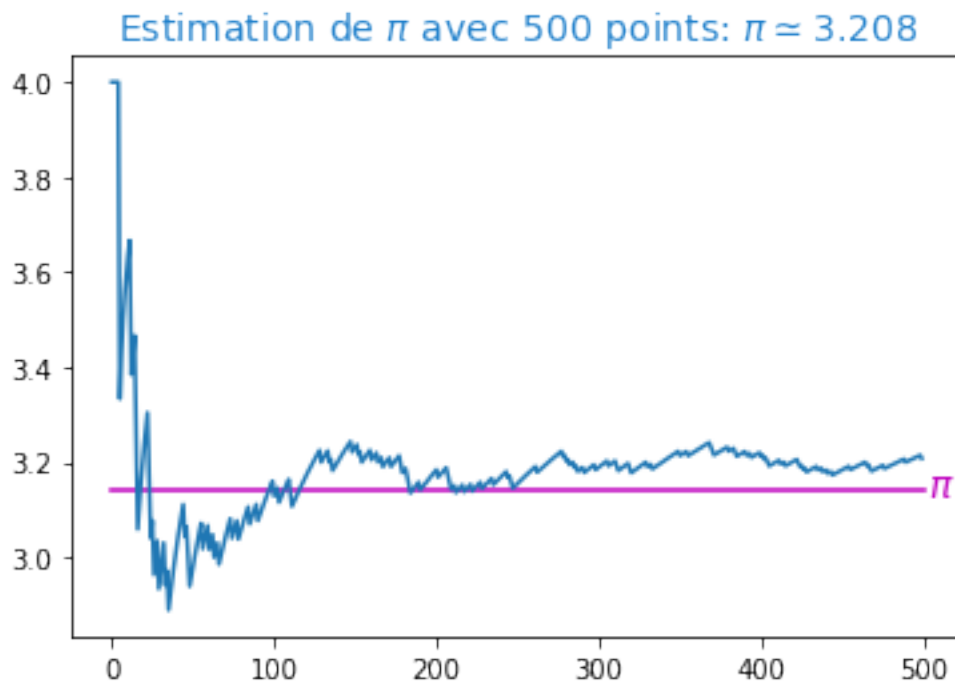
### cte ###
N=500

estimation = []
somme = 0
for i in range(N):
    (x,y) = (rd(),rd())
    if x**2+y**2<1:
        somme = somme + 1
    estimation.append(4*somme/(i+1))

plt.plot([0,N],[3.14,3.14],color='m')
plt.text(N+2,3.13,"$\pi$",color='m',fontsize=14)
plt.title("Estimation de $\pi$ avec {} points: $\pi \simeq \{ }$".format(N,estimation[-1]))
plt.plot(estimation)
plt.show()

```

---



Suggestions pédagogiques

- **Expliquer un programme**  
Que contient la liste estimation ?

- **Compléter un programme**

Le programme précédent étant fourni en remplaçant la ligne 13 par `estimation.append(...)`, demander aux élèves de compléter la ligne 13.

Remarque

Il peut être intéressant d'utiliser la courbe en mode 'pas à pas' pour montrer aux élèves que lorsqu'un point vert apparaît la courbe monte et que lorsqu'un point rouge apparaît, elle descend.