Thème: Probabilités et statistiques

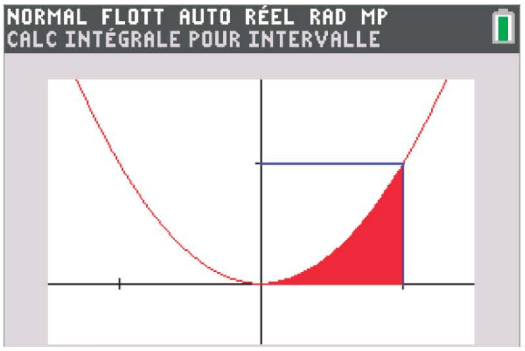
Méthode de Monte-Carlo

TI-83 Premium CE
Edition Python

Enoncé

Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = x^2$. On note C_f la courbe de la fonction f dans un repère orthonormé $(O;\vec{\imath},\vec{\jmath})$. On cherche à calculer A l'aire délimitée par C_f, l'axe (Ox) et les droites d'équations $x = 0$ et $x = 1$.

On a représenté cette aire en rouge ci-contre.



Pour calculer A on va utiliser la méthode de Monte-Carlo qui consiste à générer aléatoirement n points dans le carré de côté 1 contenant notre aire. La proportion des points à l'intérieur de notre aire nous donnera une valeur approchée de A.

1. Ecrire la fonction Python **f** qui prend comme argument **x** et qui renvoie **x²**. Exécuter le script et calculer les images de 1 et 1,5.

```
def f(x):
  y=.....
  return y
```

2. Compléter la fonction Python **graphe** qui représente graphiquement la fonction **f** sur l'intervalle $[0;1]$.

3. Compléter la fonction Python **mc** qui prend comme argument **n** et qui affiche au hasard **n** points dans le rectangle bleu puis dénombre les points à l'intérieur de l'aire recherchée et affiche la proportion des points à l'intérieur de l'aire A. Lancer la fonction en prenant $n = 1000$.

4. Vérifier votre calcul en utilisant l'outil calculs accessible dans 2nde trace (calculs f4). puis dans la page de calculs.

```
import ti_plotlib as plt
def graphe():
  plt.cls()
  plt.window(-0.5,1.5,-0.3,1.2)
  plt.axes("on")
  for i in range(500):
    plt.plot(i/500,.....,".")
  plt.color(0,0,255)
  plt.line(0,1,1,1,"")
  plt.line(1,1,1,0,"")
```

```
from random import *
def mc(n):
  graphe()
  s=0
  for i in range(n):
    x,y=random(),random()
    if y<=f(x):
      s=.....
      plt.color(255,0,0)
    else:
      plt.color(0,255,0)
    plt.plot(x,y,".")
  plt.text_at(1,str(s/n),"center")
  plt.show_plot()
```

1. Fonction f

x² s'écrit en Python **x**2**. (* correspond à la multiplication).

```
ÉDITEUR : MONTECAR
LIGNE DU SCRIPT 0004
def f(x):
  y=x**2
  return y
```

On obtient $f(1) = 1^2 = 1$ ainsi que $f(1{,}5) = 1{,}5^2 = 2{,}25$.

```
PYTHON SHELL

>>> # Shell Reinitialized
>>> # L'exécution de MONTECAR
>>> from MONTECAR import *
>>> f(1)
1
>>> f(1.5)
2.25
```

2. Fonction graphe

cls permet d'effacer l'écran ; **window** définit les valeurs extrêmes de la fenêtre ; **axes** affiche les axes (Ox) et (Oy) ; **plot** affiche un point, ici de coordonnées **(i/500;f(i/500))** ; **color** change la couleur en bleu (code **rgb** 0,0,255) et enfin on affiche les segments manquants encadrant la surface d'étude à l'aide de l'instruction **line**.

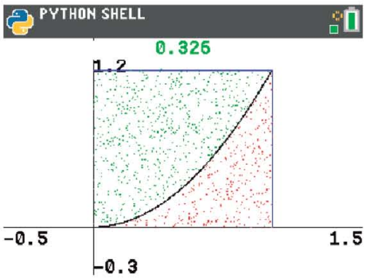
Toutes ces instructions sont accessibles dans math **ti_plotlib**.

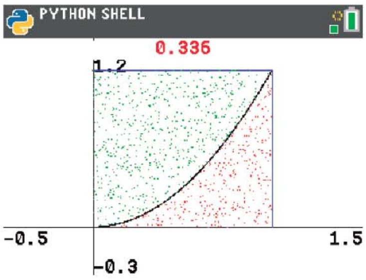
```
import ti_plotlib as plt
def graphe():
  plt.cls()
  plt.window(-0.5,1.5,-0.3,1.2)
  plt.axes("on")
  for i in range(500):
    plt.plot(i/500,f(i/500),".")
  plt.color(0,0,255)
  plt.line(0,1,1,1,"")
  plt.line(1,1,1,0,"")
```

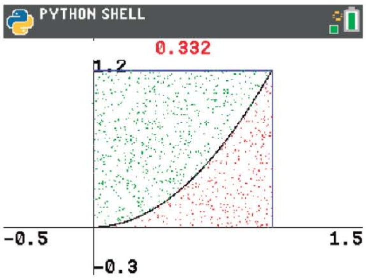
3. Algorithme Monte-Carlo

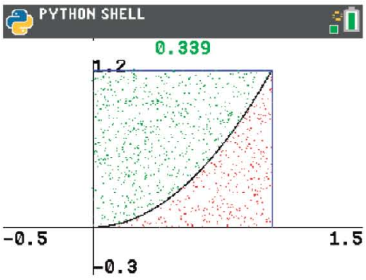
s dénombre les points situés à l'intérieur de l'aire A (ces points sont en rouge sur les figures ci-dessous). On obtient les valeurs approchées de A suivantes : 0,326 ; 0,336 ; 0,332 et 0,339.

```
from random import *
def mc(n):
  graphe()
  s=0
  for i in range(n):
    x,y=random(),random()
    if y<=f(x):
      s=s+1
      plt.color(255,0,0)
    else:
      plt.color(0,255,0)
    plt.plot(x,y,".")
  plt.text_at(1,str(s/n),"center")
  plt.show_plot()
```



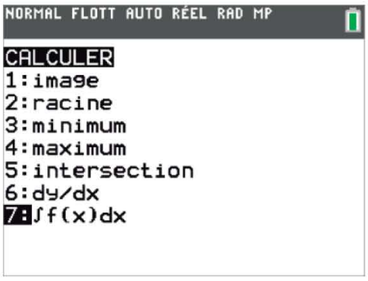






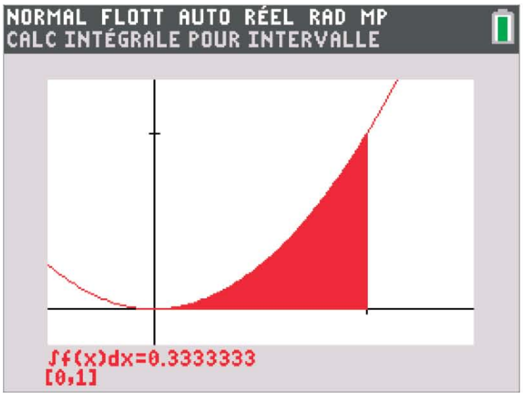
4. Calcul de l'aire

Méthode graphique : On peut obtenir une valeur approchée de l'aire en utilisant la fonction **calcul** de la calculatrice accessible dans [2nde] [trace] (calculs f4).



Après avoir choisi **∫f(x)dx** il faut entrer les bornes : 0 (valider en appuyant sur [entrer]), puis 1 (et valider).

On obtient une aire $A = 0{,}333$ ce qui est proche de ce qu'on avait obtenu lors de l'algorithme de Monte-Carlo.



STL : Pour trouver la valeur exacte de l'intégrale, on appuie sur [2nde] [fraction] et on sélectionne le symbole [∫ d□].