

Informatique

Tracer une courbe en Python

M. Marmorat

16 janvier 2025

On utilise le module `matplotlib.pyplot`. On souhaite tracer la courbe de la fonction f sur l'intervalle $[a, b]$, avec $a < b$. L'idée est de discrétiser l'intervalle $[a, b]$. Par exemple, la courbe de la fonction \exp sur $[-1, 1]$.

On a deux méthodes, avec ou sans le module `numpy`.

1 Avec Numpy

```
import numpy as np
import matplotlib.pyplot as plt

# a est la borne inferieure de l'intervalle
# b est la borne superieure de l'intervalle
# N est le nombre de points de discretisation
# f est la fonction dont on souhaite tracer la courbe
X = np.linspace(a, b, N)
Y = f(X)
plt.plot(X, Y)
plt.show()
```

Remarque 1. Attention, la fonction f , les paramètres a et b ainsi que N doivent être affectés (c'est-à-dire avec des valeurs concrètes). Par exemple

```
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(-1, 1, 100)
Y = np.exp(X)
plt.plot(X, Y)
plt.show()
```

2 Sans Numpy

```
import matplotlib.pyplot as plt

# a est la borne inferieure de l'intervalle
# b est la borne superieure de l'intervalle
# N est le nombre de points de discretisation
# f est la fonction dont on souhaite tracer la courbe
pas = (b-a)/(N-1)
X = [a + k*pas for k in range(N)]
Y = [f(x) for x in X]
plt.plot(X, Y)
plt.show()
```

Remarque 2. Attention, la fonction f , les paramètres a et b ainsi que N doivent être affectés (c'est-à-dire avec des valeurs concrètes). Par exemple

```
import math
import matplotlib.pyplot as plt

a = 1
b = -1
N = 100

pas = (b-a)/(N-1)
X = [a + k*pas for k in range(N)]
Y = [math.exp(x) for x in X]
plt.plot(X, Y)
plt.show()
```

3 Tracer un cercle

Pour tracer le cercle unité, on paramètre le cercle trigonométrique à l'aide des fonctions cos et sin.

```
import numpy as np
import matplotlib.pyplot as plt

N = 100
T = np.linspace(0, 2*np.pi, N)
X = np.cos(T)
Y = np.sin(T)

plt.close()
plt.plot(X, Y)
plt.show()
```

Pour tracer le cercle de centre (x_0, y_0) et de rayon $r > 0$:

```
import numpy as np
import matplotlib.pyplot as plt

N = 100
T = np.linspace(0, 2*np.pi, N)
X = x0 + r*np.cos(T)
Y = y0 + r*np.sin(T)

plt.close()
plt.plot(X, Y)
plt.show()
```