

TP16

REPRÉSENTATIONS GRAPHIQUES

*Corrigé***Exercice 1** Créer la liste

$$x = [0, 0.5, 1, 1.5, \dots, 5]$$

puis en déduire la liste

$$y = [0^3 - 2 \times 0, 0.5^3 - 2 \times 0.5, 1^3 - 2 \times 1, 1.5^3 - 2 \times 1.5, \dots, 5^3 - 2 \times 5].$$

```
x = np.arange(0, 5.5, 0.5)
# ou
x = np.linspace(0, 5, 11)

y = x**3 - 2*x
```

Exercice 2

```
x = [0, 3, 5]
y = [2, 1, 4]
plt.plot(x, y)
plt.show()
```

Exercice 3 Tracer la courbe de la fonction $f : x \mapsto x^2 + x + 1$ sur $[0, 5]$.

```
# méthode 1 : directe
x = np.linspace(0, 5, 100)
y = x**2 + x + 1
plt.plot(x, y)
plt.show()
```

```
# méthode 2 : la fonction f a
# été définie
def f(x):
    return x**2 + x + 1

x = np.linspace(0, 5, 100)
y = f(x)
plt.plot(x, y)
plt.show()
```

Exercice 4 Tracer la courbe de la fonction $g : x \mapsto \ln(x) - 2x\sqrt{x}$ sur $]0, 5]$.

```
# méthode 1 : directe
x = np.linspace(0.01, 5, 100)
# même si numpy gère le problème,
y = np.log(x) - 2*x*sqrt(x)
plt.plot(x, y)
plt.show()
```

```
# méthode 2 : avec la fonction g
def g(x):
    return np.log(x) - 2*x*sqrt(x)

x = np.linspace(0.01, 5, 100)
y = g(x)
plt.plot(x, y)
plt.show()
```

Exercice 5 Tracer **sur le même graphique**, les courbes des fonctions \exp et $f: x \mapsto \sqrt{6 + \sqrt{3x}}$ sur $[0, 3]$. On ajoutera une légende et un titre.

```
# Courbe de exp
x = np.linspace(0,3,100)
y = np.exp(x)
plt.plot(x,y, label = "Courbe de exp")

# Courbe de f
x = np.linspace(0,3,100)
y = np.sqrt(6+np.sqrt(3*x))
plt.plot(x,y, label = "Courbe de f")

# Affichage du graphique
plt.title("Courbes de exp et f") # titre
plt.legend() # affichage de la légende
plt.show() # affichage du graphique
```

Exercice 6 On considère la suite définie par
$$\begin{cases} u_0 = 1 \\ u_{n+1} = \sqrt{2 + u_n} \end{cases}$$

1. Écrire un programme qui crée la liste (classique) $U = [u_0, u_1, \dots, u_{50}]$.

```
u = 1
U = [u]
for k in range(50):
    u = np.sqrt(2+u)
    U.append(u)
print(U)
```

2. Représenter alors les points de coordonnées (k, u_k) pour $k \in \llbracket 0, 50 \rrbracket$.

```
# Liste des abscisses
n = range(0,51) # ou np.arange(0,51,1) ou np.linspace(0,50,51)
# graphique
plt.plot(n,U, "+")
# Le "+" permet d'obtenir un nuage de points marqués +
plt.show()
```

Exercice 7

```
x = np.linspace(3,7,100)
y = 3*np.sqrt(1-x**2/49)
plt.plot(x,y)

x = np.linspace(-7,-3,100)
y = 3*np.sqrt(1-x**2/49)
plt.plot(x,y)

x = np.linspace(4,7,100)
y = -3*np.sqrt(1-x**2/49)
plt.plot(x,y)

x = np.linspace(-7,-4,100)
```

```

y = -3*np.sqrt(1-x**2/49)
plt.plot(x,y)

x = np.linspace(0.75,1,100)
y = 9-8*np.abs(x)
plt.plot(x,y)

x = np.linspace(-1,-0.75,100)
y = 9-8*np.abs(x)
plt.plot(x,y)

x = np.linspace(0.5,0.75,100)
y = 0.75+3*np.abs(x)
plt.plot(x,y)

x = np.linspace(-0.75,-0.5,100)
y = 0.75+3*np.abs(x)
plt.plot(x,y)

# Attention, f5 est une fonction constante
x = [-0.5,0.5] # ou x = np.linspace(-0.5,0.5,100)
y = [2.25, 2.25] # y = 2.25 + x*0
# x et y doivent être des listes de même taille.
plt.plot(x,y)

x = np.linspace(-4,4,100)
y = np.abs(x/2)-(3*np.sqrt(33)-7)/112*x**2 + np.sqrt(1-( np.abs( np.abs(x)-2) -1)**2) - 3
plt.plot(x,y)

x = np.linspace(-3,-1,100)
y = 6*np.sqrt(10)/7 + 1.5-0.5*np.abs(x) - 6*np.sqrt(10)/14*np.sqrt(3+2*np.abs(x)-x**2)
plt.plot(x,y)

x = np.linspace(1,3,100)
y = 6*np.sqrt(10)/7 + 1.5-0.5*np.abs(x) - 6*np.sqrt(10)/14*np.sqrt(3+2*np.abs(x)-x**2)
plt.plot(x,y)

plt.show()

```

