TP16

Représentations graphiques

Préparer les parties I et II.1

I Les listes de numpy

Nous savons créer des listes, mais les calculs numériques sur les listes ne sont pas naturels : si `L1 = [a0, a1, a2]` et `L2 = [b0, b1, b2]`, les listes

`[a0+b0, a1+b1, a2+b2]` ou `[a0*b0, a1*b1, a2*b2, a3*b3]`

ne sont pas faciles à obtenir (il faudrait faire une boucle for).
Rappel : `L1 + L2 = [a0, a1, a2, b0, b1, b2]` (concaténation).

On va donc introduire un nouveau type de listes pour pouvoir faire ces calculs.

I.1 Commande np.array

La bibliothèque **numpy** se charge avec

`import numpy as np`

La fonction **array** de numpy permet de transformer une liste normale en une **liste numpy** (ou tableau à une dimension).

`np.array( [a0, a1, ..., an] )`

Le résultat est une variable de type array (tableau).

```
>>> import numpy as np
>>> L = np.array([1,2,3])
>>> type(L)
<class 'numpy.ndarray'>
>>> L
array([1, 2, 3])
```

I.2 Listes particulières (très utiles pour ce TP)

- `np.linspace(a, b, N)` : liste de $N \in \mathbb{N}$ valeurs régulièrement espacées, de $a \in \mathbb{R}$ à $b \in \mathbb{R}$ inclus ($a \leqslant b$).

```
>>> np.linspace(1,3,10)
array([1.        , 1.22222222, 1.44444444, 1.66666667, 1.88888889,
       2.11111111, 2.33333333, 2.55555556, 2.77777778, 3.        ])
```

- `np.arange(a, b, p)` : liste des nombres de a inclus à b exclu, espacées d'un pas de p. C'est la même chose que `range(a,b,p)` mais avec la possibilité de mettre des valeurs a, b, p non entières.

```
>>> np.arange(1 , 3, 0.1)
array([1. , 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1,
       2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9])
```

I.3 Calculs avec les listes de numpy

Soient `L1` et `L2` deux **listes numpy** et soient k un nombre et n un entier.

Opération	Syntaxe Python	Résultat
Addition terme à terme	`L1 + L2`	`array( [a0 + b0, ..., an + bn] )`
Ajout d'un nombre	`L1 + k`	`array( [a0 + k, ..., an + k] )`
Multiplication terme à terme	`k * L1`	`array( [k * a0, ..., k * an] )`
	`L1 * L2`	`array( [a0 * b0, ..., an * bn] )`
Division terme à terme	`L1 / L2`	`array( [a0 / b0, ..., an / bn] )`
Puissance terme à terme	`L1 ** k`	`array( [a0 ** k, ..., an ** k] )`
	`L1 ** L2`	`array( [a0 ** b0, ..., an ** bn] )`
	`k ** L1`	`array( [k ** a0, ..., k ** an] )`

Exercice 1 Créer la liste
$$x = [0\ ,\ 0.5\ ,\ 1\ ,\ 1.5\ ,\ \dots\ ,\ 5]$$
puis en déduire la liste
$$y = [0^3 - 2\times 0\ ,\ 0.5^3 - 2\times 0.5\ ,\ 1^3 - 2\times 1\ ,\ 1.5^3 - 2\times 1.5\ ,\ \dots , 5^3 - 2\times 5].$$

Fonctions usuelles

Les fonctions usuelles de numpy permettent le calcul sur les listes.

```
>>> import numpy as np

>>> L1 = np.array([0, 1, 2])

>>> np.exp(L1)
array([ 1., 2.71828183,  7.3890561])  # liste [exp(0), exp(1), exp(2)]
```

II Tracer une courbe

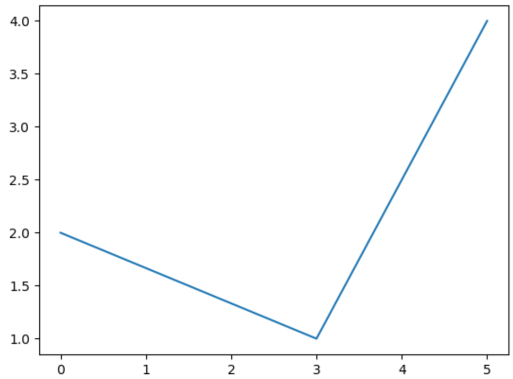
La bibliothèque permettant de tracer des graphiques est **matplotlib.pyplot**

```
import matplotlib.pyplot as plt
```

II.1 Méthode générale

```
x = liste (normale ou array) des abscisses
y = liste (normale ou array) des ordonnées
# calcul du graphique
plt.plot(x,y)
# affichage
plt.show()
```

Exercice 2 Tracer ce graphique.




```
x =

y =

plt.plot(x,y)
plt.show()
```

II.2 Courbe d'une fonction simple

Mise à part le cas des fonctions affines (représentées par des droites qui peuvent être tracées comme ci-dessus), on procèdera de la manière suivante :

- On construit la liste x avec `np.linspace(a, b, 100)` (exemple courant de 100 points, à augmenter si le tracé n'est pas suffisamment lisse).
- On calcule ensuite y avec les opérations vues précédemment.

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

```
# méthode 1 : directe

x = np.linspace(...., ....., 100)

y =

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(..., ..., 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
```

```
# méthode 2 : avec la fonction g
```

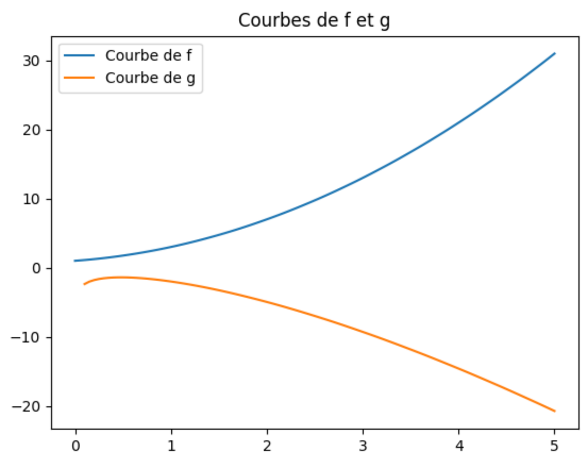
Toujours fermer la fenêtre du graphe entre deux exécutions.

II.3 Afficher deux courbes sur le même graphique, avec une légende

```
# Courbe de f
x = np.linspace(0,5,100)
y = x**2 + x + 1
plt.plot(x,y, label = "Courbe de f")

# Courbe de g
x = np.linspace(0.1,5,100)
y = np.log(x) - 2 * x * np.sqrt(x)
plt.plot(x,y, label = "Courbe de g")

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



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.
Inutile de recopier le script sur papier.

III Représentation des termes d'une suite

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) $U = [u_0, u_1, \dots, u_{50}]$.

```
```

2. Représenter alors les points de coordonnées (k, u_k) pour $k \in [\![0, 50]\!]$.

```
# Liste des abscisses
n =
# graphique
plt.plot(... , ... , "+")
# Le "+" permet d'obtenir un nuage de points marqués +
plt.show()
```

3. Conjecturer la limite de la suite (u_n).

IV Pour les plus rapides

Exercice 7 Tracer sur un même graphique les courbes des fonctions suivantes :

- $f_1 : x \mapsto 3\sqrt{1-\dfrac{x^2}{49}}$ sur $[3\,;7]$ et sur $[-7;-3]$
- $f_2 : x \mapsto -3\sqrt{1-\dfrac{x^2}{49}}$ sur $[4\,;7]$ et sur $[-7;-4]$
- $f_3 : x \mapsto 9-8|x|$ sur $[0.75;1]$ et sur $[-1;-0.75]$
- $f_4 : x \mapsto 0.75+3|x|$ sur $[0.5;0.75]$ et sur $[-0.75;-0,5]$
- $f_5 : x \mapsto 2.25$ sur $[-0.5;0,5]$
- $f_6 : x \mapsto \left|\dfrac{x}{2}\right| - \dfrac{3\sqrt{33}-7}{112}x^2 + \sqrt{1-\left(\big||x|-2\big|-1\right)^2} - 3$ sur $[-4;4]$
- $f_7 : x \mapsto \dfrac{6\sqrt{10}}{7} + (1.5-0.5|x|) - \dfrac{6\sqrt{10}}{14}\sqrt{3+2|x|-x^2}$ sur $[-3;-1]$ et sur $[1;3]$.