

---

## TP1 : Numpy et Matplotlib

---

Lorsqu'on souhaite faire du calcul numérique en Python, les bibliothèques **Numpy** et **Matplotlib** sont des incontournables.

- La bibliothèque **Numpy** introduit une classe d'objets spécifiques appelés les *array*, qui correspondent à des *tableaux de nombres*. Ces tableaux sont très utiles en calcul numérique, pour stocker des données statistiques, des signaux temporels, des matrices, etc.
- La bibliothèque **Matplotlib** permet de tracer des *graphiques* de manière simple et puissante.

### Partie 1 – Numpy

Pour utiliser les fonctionnalités de cette bibliothèque, il faut commencer par l'importer. Il est d'usage de désigner la bibliothèque Numpy sous le nom raccourci *np* :

```
import numpy as np
```

Numpy définit une classe très puissante appelée les *array*. Il s'agit de tableaux de nombres (avec autant de dimensions qu'on le souhaite, mais dans ce TP on se limite à une seule dimension), sur lesquels on peut effectuer facilement de nombreuses opérations mathématiques.

**Exercice 1** (Découverte des *array*). Ouvrez le fichier *exo1\_numpy\_généralités.py*, exécutez-le et répondez aux questions.

**Exercice 2** (Indexation des tableaux). Ouvrez le fichier *exo2\_numpy\_indexation.py*, exécutez-le et répondez aux questions.

## Partie 2 – Matplotlib

### Premier exemple

Dans ce cours, on utilisera principalement l'interface de Matplotlib dédiée à la représentation de courbes 2D. Cette interface s'appelle *pyplot*. Ouvrez un nouveau script et copiez-y l'exemple suivant :

```
import matplotlib.pyplot as plt

plt.figure()
X=[0,3,2,1]
Y=[0,0,1,-1]
plt.plot(X,Y,color='blue')
plt.axis('equal')
plt.title('Ma première figure')

plt.show()
```

Commentons ces quelques lignes :

- *plt.figure()* crée une nouvelle figure (mais sans forcément l'afficher à l'écran).
- *X=...* définit une liste de 4 nombres.
- *Y=...* définit une autre liste de 4 nombres.
- *plt.plot(X,Y,...)* affiche dans la figure 4 points, dont les 4 abscisses sont données par la liste *X* et les 4 ordonnées sont données par la liste *Y*. Les arguments suivants, optionnels, permettent de définir les caractéristiques du tracé.
- *plt.axis('equal')* impose que l'échelle de longueur soit la même en x et en y.
- *plt.title(...)* définit le titre du graphique.
- Enfin, *plt.show()* est nécessaire pour afficher la (ou les) figure(s) qu'on vient de créer.

### Superposer plusieurs commandes graphiques

On peut superposer les résultats de plusieurs commandes graphiques dans une même figure. Dans le code précédent, après le *plt.title(...)* mais AVANT le *plt.show()*, rajoutez les lignes suivantes :

```
A=[0,1,2,3,2,1,1.5]
B=[0,-1,-1,0,1,1,0]
plt.plot(A,B,color='blue',linestyle='none',marker='o',markerfacecolor='red',
         markersize=10)
```

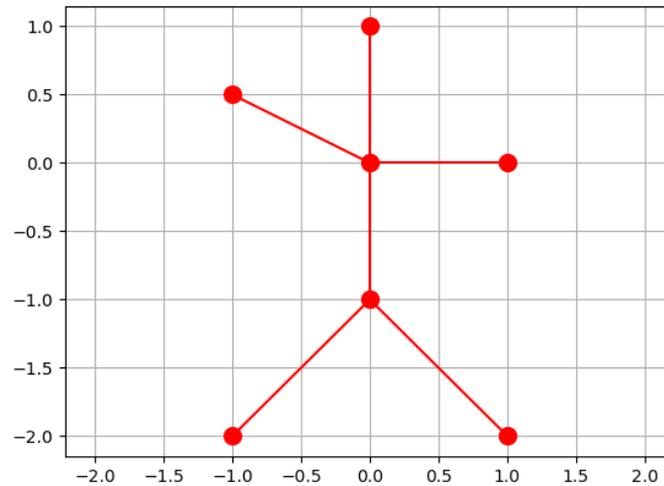
### Remarques concernant la commande *plt.show()*

- Si on tape simplement *plt.show()*, l'exécution du code est interrompue jusqu'à ce qu'on ait fermé toutes les figures affichées.
- C'est pourquoi il est pratique de n'écrire la commande *plt.show()* qu'à la toute fin du script. Pour tout fermer, il suffit ensuite de forcer l'interruption du script avec *Ctrl+C*.
- Une autre option plus élégante est de remplacer *plt.show()* par les lignes suivantes :

```
plt.show(block=False)
input()
```

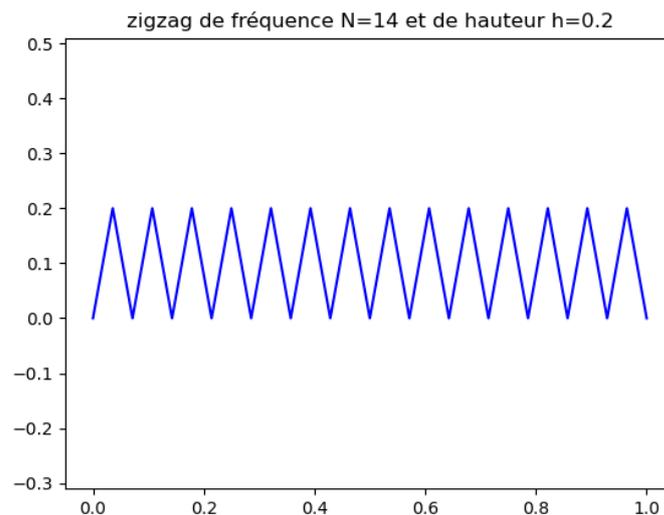
Dans ce cas, l'exécution du script ne se bloque pas au `plt.show`, elle continue jusqu'au `input()` (attente d'une entrée clavier). Pour terminer le script il suffit alors d'appuyer sur *Entrée* dans le terminal, et cela fermera automatiquement toutes les fenêtres créées.

**Exercice 3.** Reproduisez la figure suivante :



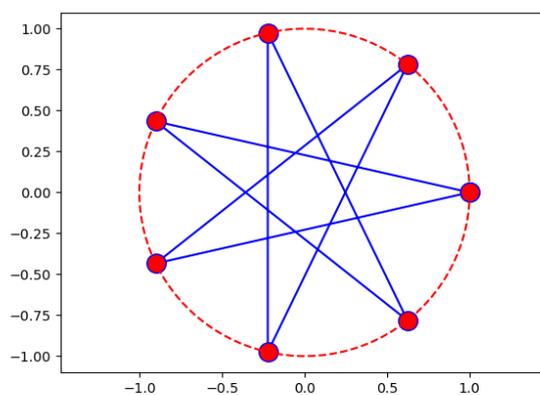
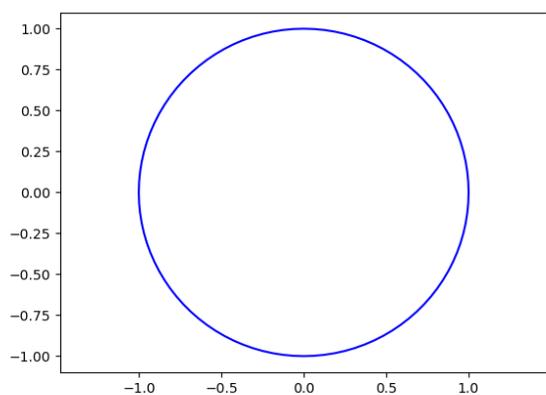
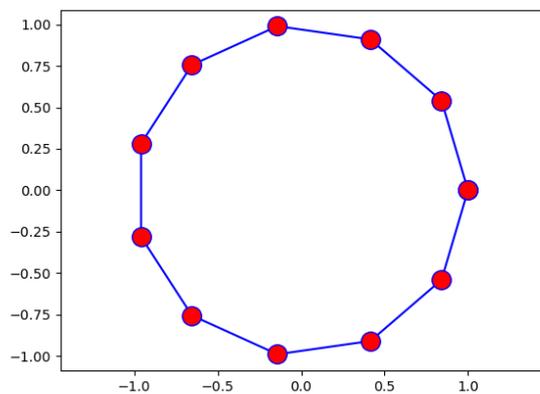
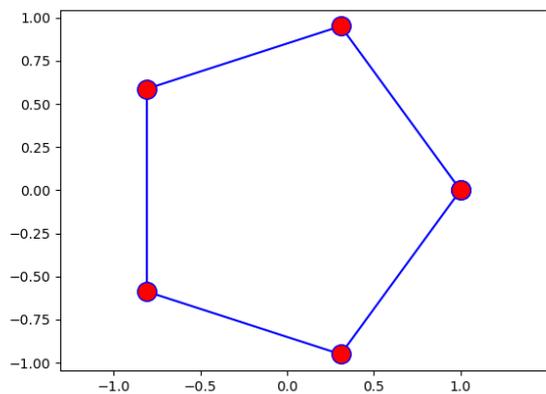
*Indice* : en anglais, une grille de coordonnées se dit *grid*.

**Exercice 4.** Reproduisez la figure suivante :



*Consignes* : En abscisses, le zigzag va toujours de  $x = 0$  jusqu'à  $x = 1$ . Par contre, les paramètres  $N$  (nombre de zig-zags) et  $h$  (hauteur du zig-zag) doivent être modifiables dans votre code. Testez votre code avec plusieurs valeurs de  $N$  et  $h$ .

**Exercice 5.** Reproduisez les figures suivantes :



*Indice* : tous les points à représenter se trouvent sur le cercle trigonométrique (de centre (0,0) et de rayon 1). Il reste juste à définir l'*angle* associé à chaque point.