

TP3 Matplotlib

2024 - 2025

1^{er} avril 2025

1 Petits exercices avant la pratique

Matplotlib est une bibliothèque Python qui permet de faire toutes sortes de tracés. Avant tout, importez le module matplotlib à l'aide de :

```
1 import matplotlib.pyplot as plt # plt est un raccourci pour matplotlib.pyplot
```

Exercice 1 : Création d'une figure

A l'aide de la méthode `figure`, créez une figure de couleur bleue de 500 pixels en largeur et 300 pixels en hauteur. A noter que

$$h = dpi \times h_{inch} \text{ et } w = dpi \times w_{inch}$$

avec *dpi* dot per inch (pixels par pouce)

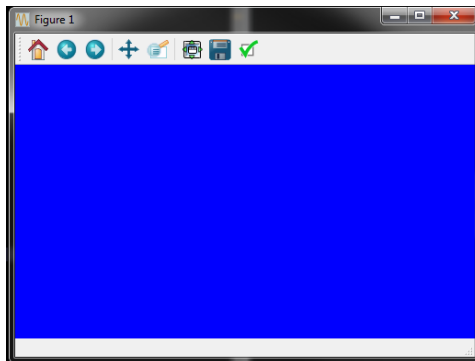


FIGURE 1 – figure Matplotlib

Exercice 2 : Création d'un axe

A l'aide de la méthode `axes`, créez un axe centré sur la figure. Pour information, la taille de la figure est unitaire (longueur = 1 et largeur = 1).

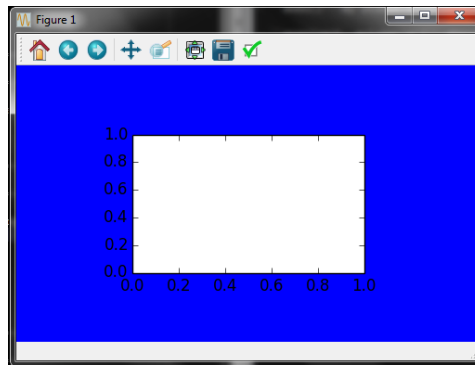


FIGURE 2 – figure Matplotlib avec un axe de taille 50% de la figure et centré.

Exercice 3 : La méthode subplot

A partir de l'aide de la méthode `subplot`, créez ensuite une figure analogue à la figure ci-dessous. Vous pouvez faire des tests pour vous aider.

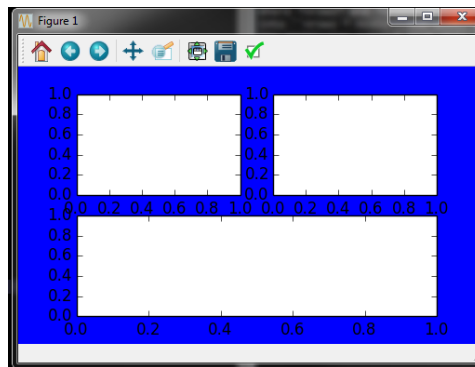


FIGURE 3 – figure Matplotlib avec 3 axes. Utilisation de la méthode `subplot`

2 Rappels et compléments sur NumPy

NumPy est une bibliothèque Python qui permet à la fois de manipuler des tableaux multidimensionnels mais aussi de faire des calculs sur ces objets.

- Création d'un tableau avec NumPy

```
1 import numpy as np
2 v = np.array([1, 2, 3.4])
```

- Faire des opérations algébriques sur les tableaux

```
1 w = 3*v - np.array([2, 2, 10.2]) # on obtient ?
```

- Afin de compléter votre formation veuillez vous référer au document synthétique ci-joint.

NumPy peut faire beaucoup plus que ça. Comme le module `math` de Python, `numpy` dispose de la plupart des fonctions mathématiques que vous connaissez. Cependant, les fonctions NumPy ont l'avantage de s'appliquer sur les tableaux (composantes par composantes).

Exercice 4 : Que donne le code suivant ?

```
1 import numpy as np
2 v = np.array([1, 2, 3.4]) * np.pi / 4
3 w = np.sin(v)
4 print(w)
```

Exercice 5 : Que font les fonctions de NumPy suivantes

`zeros`, `ones`, `linspace`, `random.random`, `arange` ? Utilisez ces fonctions pour recréer l'exemple $y = \sin(x)$ effectué lors du TP n°1

3 NumPy et Matplotlib

Nous allons essentiellement l'utiliser pour tracer des fonctions. Les instructions de base pour se servir de Matplotlib sont illustrées ici :

```
1 import matplotlib.pyplot as plt
2 x=np.array([0,1,2,3])
3 y=np.random.randint(0,9,size=4)
4 plt.plot(x,y,"dg--",label="Mes_données")
5 plt.xlabel("axe_des_x")
6 plt.ylabel("axe_des_y")
7 plt.title("Titre_du_graphique")
8 plt.grid(True)
9 plt.legend()
```

Exercice 6 :

En utilisant les fonctions `linspace` et `cos` de numpy, tracez la fonction sur $y = \cos(x)$ avec $x \in [0, 10\pi]$. Quelle est l'influence du nombre de points passé à la fonction `linspace` ? Grâce à son second appel à la fonction `plt.plot` superposez le graphe de la fonction $y = \exp\left(-\frac{x}{10}\right) \cdot \cos(x)$ au précédent. Pensez à bien nommer les axes et les courbes.

Exercice 7 : En modifiant le code ci-dessous, tracez les courbes suivantes :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 t = np.linspace( XX , XX , XX ) # XX à remplacer
4 x = XXXXX # x(t)
5 y = XXXXX # y(t)
6 plt.plot(x,y)
```

- La Lemniscate de Bernoulli

$$\begin{cases} x = \frac{\sin(t)}{1 + \cos^2(t)} \\ y = \frac{\sin(t) \cdot \cos(t)}{1 + \cos^2(t)} \end{cases} \text{ avec } t \text{ sur } [0, 2\pi]$$

- La spirale d'Archimède

$$\begin{cases} x = t \cdot \cos(t) \\ y = t \cdot \sin(t) \end{cases} \text{ avec } t \text{ sur } [0, 10\pi]$$

- La courbe du cœur

$$\begin{cases} x = 16 \sin^3(t) \\ y = 13 \cos(t) - 5 \cos(2t) - 2 \cos(3t) - \cos(4t) \end{cases} \text{ avec } t \text{ sur } [0, 2\pi[$$

- Les cyclo-harmoniques

$$\begin{cases} x = \left\{ 1 + \cos\left(\frac{p}{q}t\right) \right\} \cdot \cos(t) \\ y = \left\{ 1 + \cos\left(\frac{p}{q}t\right) \right\} \cdot \sin(t) \end{cases} \quad \text{avec } t \text{ sur } [0, 2q\pi]$$

Exercice 8 : Mise place d'une légende

Tracer les fonctions linéaire, quadratique et cubique avec `x = np.linspace(0, 2, 100)` comme abscisse. En utilisant la méthode `legend`, placer la légende en haut à gauche de la figure. Complétez le programme pour tracer les n premières fonctions x^n en demandant la valeur de n .

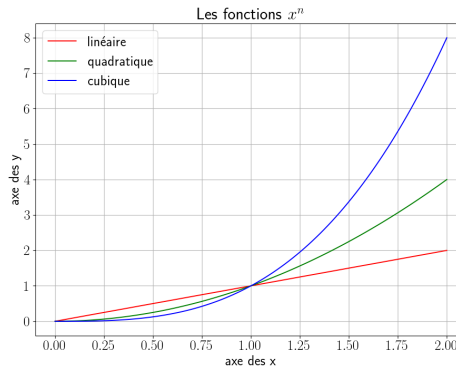


FIGURE 4 – Les fonctions puissance x^n

Exercice 9 : Tracer à partir d'un fichier .txt

Chargez le fichier texte intitulé `fichier_resultat.txt` à l'aide de la méthode `loadtxt` de NumPy. Tracez ensuite les données comme suit :

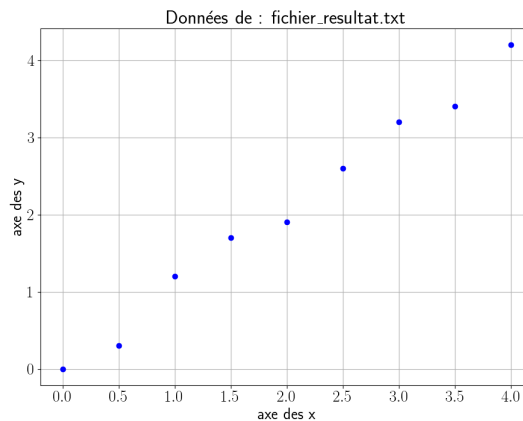


FIGURE 5 – Données issues du fichier texte : `fichier_resultat.txt`

Exercice 10 : Récapitulatif

Complétez le programme suivant afin d'obtenir la figure ci-dessous.

```
1 def f(t) :
2     return np.exp(-t)*np.cos(2*np.pi*t)
3
4 t1=np.linspace(0,5,0.1)
5 t2=np.linspace(0,5,0.02)
6
7 plt.figure(1)
8 plt.____(____)
9 plt.plot(t1,f(t1),'bo',t2,f(t2),'k-')
10 plt.grid(____)
```

```

11 plt.margins(____)
12 plt.____(____)
13 plt.plot(t2,np.cos(2*np.pi*t2),'r--')
14 plt.grid(____)
15 plt.margins(____)
16 plt.____(____)
17 plt.plot(t2,np.sin(2*np.pi*t2),'b-')
18 plt.grid(____)
19 plt.margins(____)

```

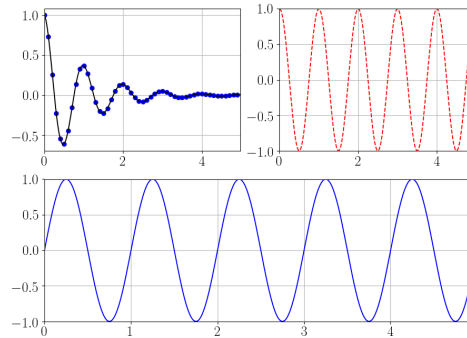


FIGURE 6 – Figure à reproduire

Exercice 11 : Histogramme

Le but est de tracer l'histogramme ci-dessous. Il représente les scores de 5 groupes différents composés d'hommes et de femmes.

- hommes : 20, 35, 30, 35, 27
- femmes : 25, 32, 34, 20, 25

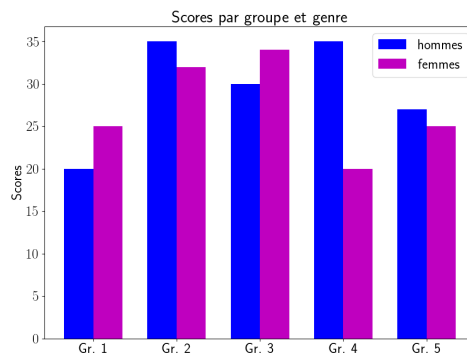


FIGURE 7 – Histogramme

On trace un histogramme à l'aide de la méthode `bar` de matplotlib. Vous prendrez une largeur (`width`) de 0.35. Vous pouvez remarquer que l'axe des abscisses n'est pas composé de chiffres mais de chaînes de caractère ; la fonction `xticks` permet de les modifier.

Exercices 12 : Camembert

A l'aide de la méthode `pie` de matplotlib, tracez le camembert ci-dessous.

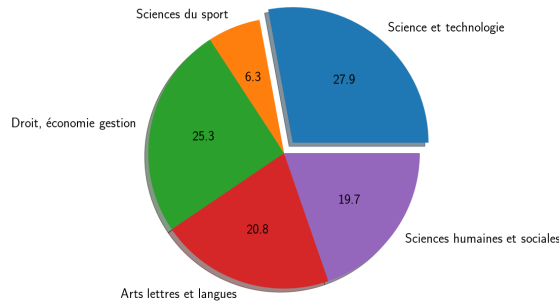


FIGURE 8 – Camembert

Petite indication : `x` correspond au pourcentage, `explode` vous permet de sélectionner la part de camembert que vous souhaitez mettre en avant, `labels` correspond à l'intitulé de chaque part, `autopct` vous permet d'afficher le pourcentage, `shadow` vous permet de rajouter une ombre et `startangle` vous permet d'afficher le camembert avec un certain angle. Vous aurez besoin de tous les paramétrer. Enfin, pour obtenir un beau croissant, il vous faudra rajouter `plt.axis('equal')`.

Exercices 13 : Coordonnées polaires

Le but de cet exercice est de tracer la figure ci-dessous.

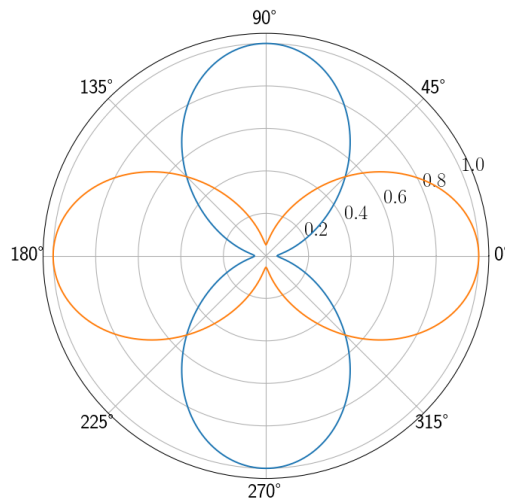


FIGURE 9 – Coordonnées polaires

Voici l'équation paramétrique :

$$r(\theta) = \left| \sqrt{\frac{5}{16\pi}} \cdot 3 \cdot \cos^2(\theta) - 1 \right| \text{ avec } \theta \text{ sur } [0, 2\pi]$$

Pour tracer en coordonnées polaires, il faut utiliser la méthode `polar`.

Exercice 14 : Contour et 3D

Le but de cet exercice est de tracer les 2 figures suivantes :

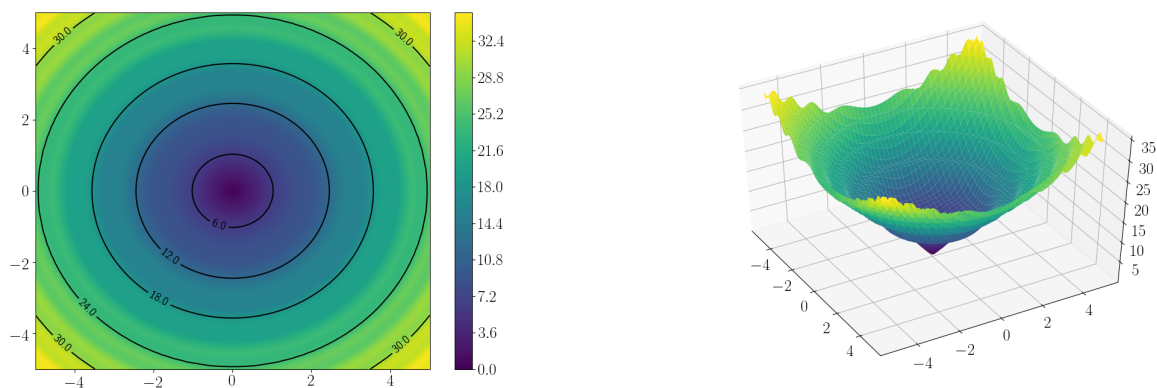


FIGURE 10 – Contour et visualisation 3D

L'équation utilisée est la suivante :

$$z(x, y) = 5 \cdot \sqrt{x^2 + y^2} + \sin(x^2 + y^2)$$

Vous prendrez 100 valeurs sur x et y (`linspace`) entre -5 et 5. Vous réaliserez un maillage X, Y à l'aide de la fonction `meshgrid`. Les contours de la fonction s'obtiennent avec `contour` et `contourf`. La fonction `colorbar` permet d'ajouter la barre de couleur sur le côté. Pour la visualisation 3D vous utiliserez `ax.plot_surface`.