Labo d'introduction à l'informatique

pour les mathématiques

Yann Thorimbert

Semaine 10

Graphiques avec matplotlib

Création de graphiques

- Nous allons utiliser certaines fonctionnalités de **matplotlib**, une librairie de visualisation de données pour Python inspirée de l'affichage des graphiques en MATLAB.

- Possibilités nombreuses, **personnalisables** ⇒ nous ne détaillons pas tout ici.

- Il est indispensable de savoir s'aider des **ressources en ligne** pour ce type de sujet.

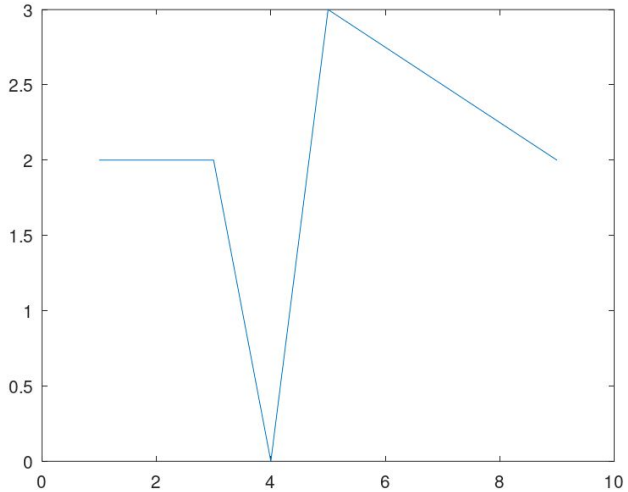
Importer matplotlib

- Nous aborderons formellement les importations de modules dans un autre cours.
- Tout comme vous avez déjà utilisé le module math, vous importerez ici le module matplotlib.
- Plus précisément, pour travailler nous créerons toujours un alias du sous-module pyplot :

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

Graphique minimaliste

```
import matplotlib.pyplot as plt
x = [1, 3, 4, 5, 9] # coordonnées x
y = [2, 2, 0, 3, 2] # coordonnées y
plt.plot(x,y)
plt.show()
```



Personnalisation d'un graphique

```
x = [1,   3, 4, 5, 9]
y = [2.5, 1, 3, 3, 2]
plt.plot(x,y)
plt.title("Mon graphe") # Ajout d'un titre (souvent pas nécessaire en science)
plt.xlabel('mon axe x') # On nomme l'axe x
plt.ylabel('mon axe y') # On nomme l'axe y
plt.xlim([min(x)-2, max(x)+2]) # On choisit l'étendue de l'axe x
plt.ylim([min(y)-3, max(y)+3]) # On choisit l'étendue de l'axe y
plt.grid() # On ajoute une grille
plt.show()
```

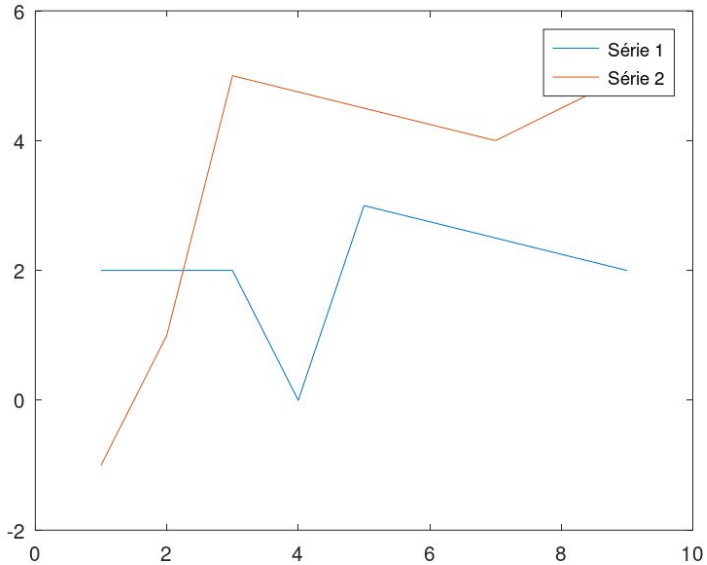
Affichage de plusieurs courbes

```
x = [1, 3, 4, 5, 9]
y = [2, 2, 0, 3, 2]
plt.plot(x, y, label="Série 1")

x2 = [1,2,3,7,9]
y2 = [-1,1,5,4,5]
plt.plot(x2, y2, label="Série 2")

plt.ylim([-2,6])

# on affiche également une légende :
plt.legend()
plt.show()
```

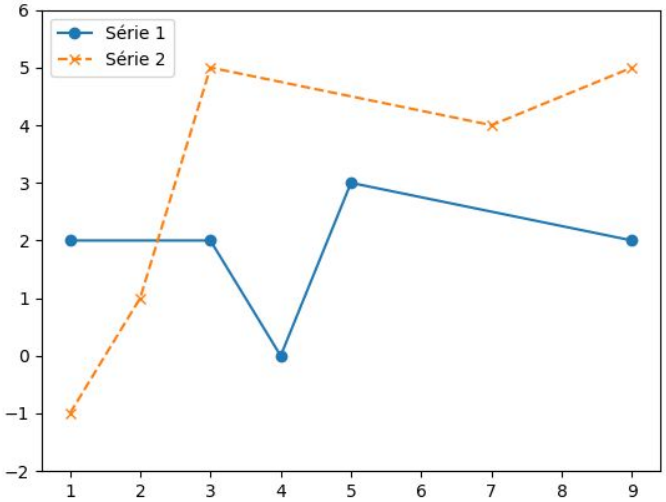


Styles | **Marker et style de ligne**

```
x = [1, 3, 4, 5, 9]
y = [2, 2, 0, 3, 2]
plt.plot(x, y, "o-", label="Série 1")

x2 = [1, 2, 3, 7, 9]
y2 = [-1, 1, 5, 4, 5]
plt.plot(x2, y2, "x--", label="Série 2")

plt.ylim([-2,6])
plt.legend()
plt.show()
```

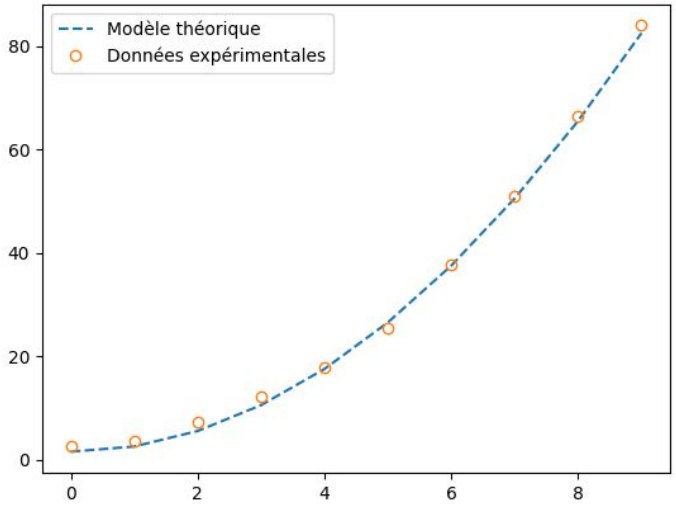


Styles | **Remplissage de marker**

```
plt.plot(x, y, "--",
         label="Modèle théorique")

plt.plot(x2, y2, "o",
         mfc="none", #Marker Face Color
         label="Données expérimentales")


plt.legend()
plt.show()
```



Styles | **Séparation des paramètres**

```
plt.plot(x, y,
         marker="s", #carré
         linestyle="-.", #pointillé-traitillé
         linewidth=2, #épaisseur de la ligne
         markersize=12, #taille du marqueur
         color="b") #bleu
```

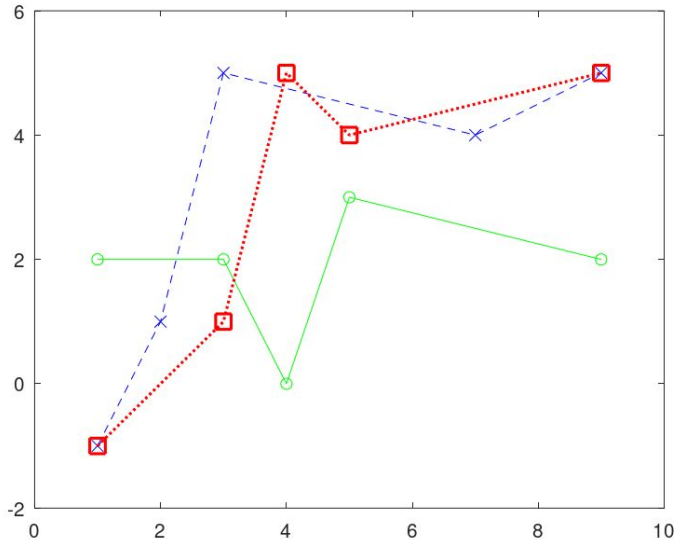
NB : expérimentez par vous-même avant tout, car matplotlib décide par défaut d'une alternance de couleurs

Styles de ligne (2)

```
# g → green, o → circle, - → trait
plt.plot(x,y,"go-")

# b → blue, x → croix, -- → traitillés
plt.plot(x2,y2,"bx--")

# r → red, s → square, : → pointillés
plt.plot(x3,y3, "rs:")
```



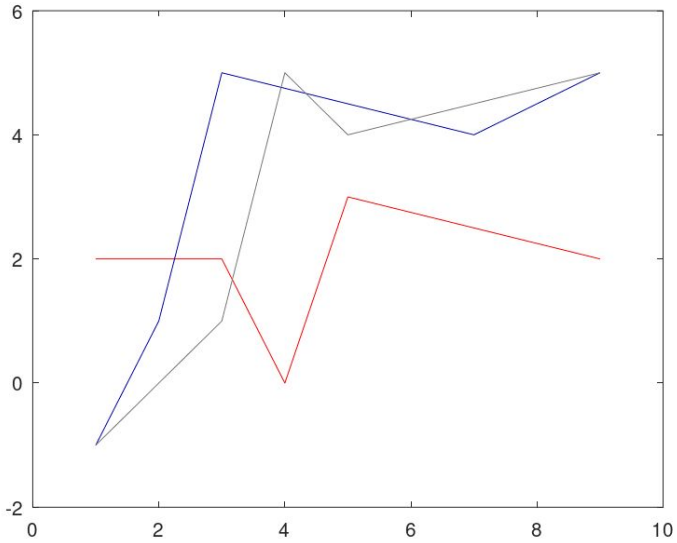

```
# Raccourcis possibles : "k", blacK, "r", Red,
"g",Green, "b", Blue, "y", Yellow, "m", Magenta,
"c",Cyan, "w", White, . . .
```

Note sur les couleurs | **Code RGB hexadécimal**

```
plt.plot(x,y, color="#ff0000") # rouge

plt.plot(x2,y2,color="#0000aa") # bleu sombre

plt.plot(x3,y3, color="#777777") # gris
```

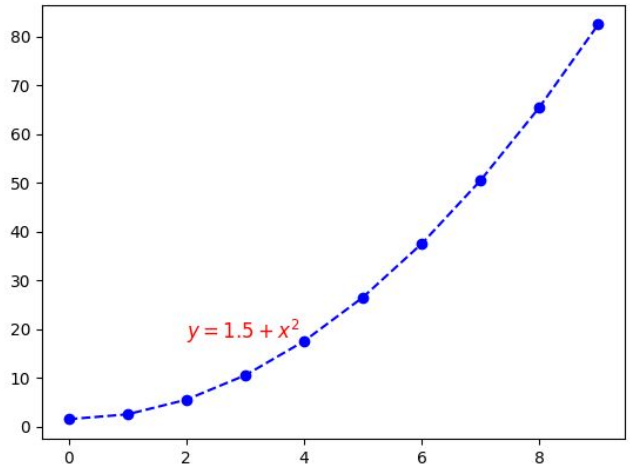


Ajout de texte et possibilité d'utiliser du LaTeX

```
plt.plot(x, y, "bo--")

plt.text( 2, 18, #coordonnées du texte
          "$y = 1.5 + x^2$", #$ pour LaTex
          fontsize=12,
          color="red")

plt.show()
```



Sauvegarde de fichier

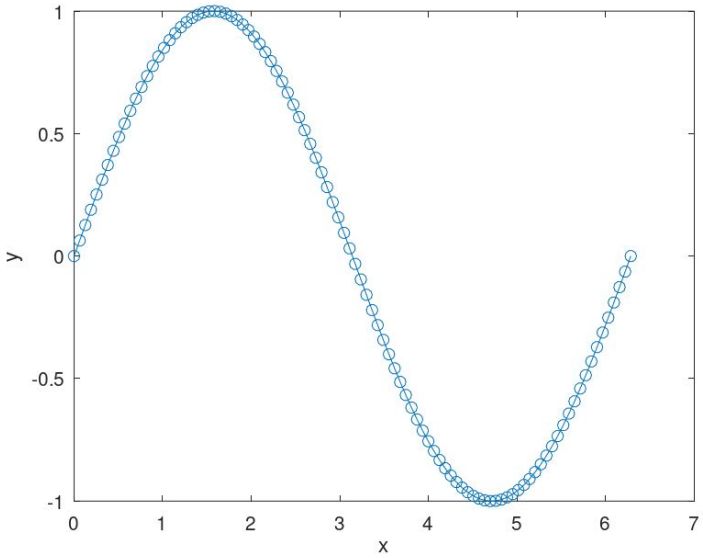
```
x = list(range(60))
y = []
for val in x:
    y.append(1.5 + (val/100)**2)

plt.plot(x, y, "b--")
plt.xlabel("t [s]")
plt.ylabel("x [m]")
plt.grid()
plt.savefig("Mon_graphe.png") #sauve dans le dossier courant
```

Utilisation de l'approche vectorielle | **À venir**

Prochainement nous verrons comment utiliser numpy pour compactifier l'écriture.

```
x = np.linspace(0, 2*np.pi, 100)
plt.plot(x, sin(x), "o-")
```



Voir la documentation et les exemples en ligne !

- Il existe d'autres types de graphiques (histogrammes, gâteaux, champs…)

- Pour ce type de problème, les ressources en ligne et les outils d'IA sont particulièrement efficaces ; savoir les utiliser de façon adéquate fait partie des compétences utiles en science.