



Python – Tracer des graphiques avec Matplotlib



La bibliothèque *matplotlib* doit être appelée pour utilisation des graphiques.

Dans la suite de la fiche, nous supposons que la ligne suivante a été insérée au début du script.

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

On suppose dans l'ensemble de la présente fiche que les listes *x* et *y* ont été déclarées au préalable avec les données à utiliser pour les graphiques.

Fonctions principales de *matplotlib*

(Consulter le site <https://matplotlib.org/> pour la notice complète)

| Fonctions | Actions réalisées |
|--|---|
| <code>plt.clf()</code> | Supprimer les tracés précédents |
| <code>plt.plot(x, y, styleDuGraphe, linewidth=1, label = 'y = f(x)')</code> | Tracer la courbe représentant <i>y</i> en fonction de <i>x</i> avec le style <i>styleDuGraphe</i> , l'épaisseur <i>linewidth</i> , le nom de la courbe à afficher dans la légende étant <i>label</i> |
| <code>plt.xlabel('x - axe des abscisses')</code> <code>plt.ylabel('y - axe des ordonnées')</code> | Ajouter des libellés sur les axes |
| <code>plt.axis([-5.5,5.5,0,10])</code> ou <code>plt.xlim(-5.5,5.5)</code> <code>plt.ylim(0,10)</code> | Définir des valeurs minimales et maximales pour les abscisses (-5.5 et 5.5) et les ordonnées (0 et 10) |
| <code>plt.title('Représentation de y en fonction de x')</code> <code>plt.title(r"\$\Delta E = \frac{h \times c}{\lambda}\$ (J)")</code> | Ajouter un titre au graphique <i>NB</i> : en ajoutant un <i>r</i> devant la chaîne de caractères, on peut afficher des formules mathématiques à l'aide de la syntaxe LATEX |
| <code>plt.grid()</code> | Ajouter une grille au graphique |
| <code>plt.text(2, 3.5, 'Point de fonctionnement')</code> <code>plt.annotate('Maximum', xy=(1.5, 1), xytext=(2,1.5), arrowprops=dict(facecolor='black', arrowstyle='->'))</code> | Ajouter du texte dans le graphe à la position souhaitée Ajouter une annotation à la position souhaitée <i>xytext</i> et trace une flèche jusqu'au point <i>xy</i> |
| <code>vecteur = plt.quiver(xVecteur, yVecteur, vecteurX, vecteurY, scale=echelleVecteur, color='r', angles='xy', units='xy')</code> <code>plt.quiverkey(vecteur, 0.1, 0.1, 2, label='échelle 2 m/s', coordinates='data')</code> | Tracer un vecteur au point d'application (<i>xVecteur</i> , <i>yVecteur</i>), <i>vecteurX</i> composante suivant <i>x</i> , <i>vecteurY</i> composante suivant <i>y</i> Trace l'échelle correspondant au vecteur <i>vecteur</i> , en position (0.1,0.1) sur le graphique et valeur de l'échelle. |
| <code>plt.legend()</code> | Ajouter une légende avec le nom des courbes |
| <code>plt.show()</code> | Afficher le graphe |

Enjoliver les graphes

✓ Paramètres de la fonction plot

`plt.plot(x, y, styleDuGraphe)` où `styleDuGraphe` est une chaîne de caractères qui regroupe la couleur de la courbe, le marqueur de point et le style de liaison entre les points.

| Chaîne | Marqueur de point |
|--------|--|
| . | <u>point</u> |
| , | <u>pixel</u> |
| o | <u>rond</u> |
| v | <u>triangle pointe en bas</u> |
| ^ | <u>triangle pointe en haut</u> |
| < | <u>triangle pointe à gauche</u> |
| > | <u>triangle pointe à droite</u> |
| 1 | <u>croix à 3 branches vers le bas</u> |
| 2 | <u>croix à 3 branches vers le haut</u> |
| 3 | <u>croix à 3 branches vers la gauche</u> |
| 4 | <u>croix à 3 branches vers la droite</u> |
| s | <u>carré</u> |
| p | <u>pentagone</u> |
| * | <u>étoile</u> |
| h | <u>hexagone</u> |
| H | <u>hexagone</u> |
| + | <u>plus</u> |
| P | <u>plus plein</u> |
| x | <u>croix</u> |
| X | <u>croix pleine</u> |
| d | <u>carreau</u> |
| D | <u>carreau plus grand</u> |
| | <u>barre verticale</u> |
| - | <u>barre horizontale</u> |

| Chaîne | Couleur en anglais | Couleur en français |
|--------|--------------------|---------------------|
| b | blue | bleu |
| g | green | vert |
| r | red | rouge |
| c | cyan | cyan |
| m | magenta | magenta |
| y | yellow | jaune |
| k | black | noir |
| w | white | blanc |

Voir la palette complète sur https://matplotlib.org/gallery/color/named_colors.html

| Chaîne | Style de ligne |
|--------|--------------------|
| - | ligne continue |
| -- | tirets |
| : | ligne en pointillé |
| -. | tirets points |

Exemple : `plt.plot(x, y, 'r+:')` → trace un graphe dont les points sont rouges, en forme de + et reliés par des lignes en pointillé.

À noter !

La fonction `plot` découpe l'option `styleDuGraphe` en morceaux :

- ✓ **Une couleur** (une seule lettre acceptée), que l'on peut aussi donner avec le mot clé `color=''` (plus de contrainte sur le nom de la couleur). De nombreuses autres fonctions de `pyplot` (`plt.grid`, `plt.xlabel`, `plt.ylabel`, `plt.title`, ...) utilisent ce mot clé permettant l'usage de toute la palette des couleurs.
- ✓ **Un style de marqueur de point**, que l'on peut aussi donner avec le mot clé `marker=''`
- ✓ **Un style de ligne**, que l'on donne avec le mot clé `linestyle=''`, souvent abrégé en `ls=''`

Si rien n'est précisé, Matplotlib utilise simplement le paramètre par défaut pour ces 3 styles.

✓ Types de graphes

- `plt.plot()` : pour tracer des courbes
- `plt.scatter()` : pour tracer des points
- `plt.bar()` : pour des diagrammes à barre
- `plt.pie()` : pour des camemberts
- `plt.hist()` : pour les histogrammes

✓ Créer des grilles de graphes

Il est possible de créer des grilles de graphes, solution très pratique pour empiler des graphes qui doivent être regardés ensemble mais qui n'ont pas les mêmes ordres de grandeurs en matière d'abscisses et d'ordonnées.

On utilise l'instruction `subplot()` qui va décrire une grille. Cette commande prend plusieurs arguments :

1. Nombre de lignes de la grille de graphe
2. Nombre de colonnes de la grille de graphe
3. Index du graphe dans la grille (la numérotation se fait de gauche à droite et de haut en bas)
4. Options

Exemple avec une grille de 1 colonne et 2 lignes (la couleur de fond du second graphe sera cyan) :

```
plt.subplot(1, 2, 1)
plt.plot(x, y, 'k-.', linewidth=2)
plt.subplot(1, 2, 2, facecolor='c')
plt.plot(t, y, 'r+:', linewidth=1)
plt.show()
```