

# Fonctions\_Matplotlib

May 28, 2024

## 1 Quelques fonctions de matplotlib

```
[ ]:
```

### 1.1 Objets de figures et d'axes

#### 1.1.1 Créer une figure

La figure contient tous les éléments du tracé. Le principal moyen de créer une figure dans matplotlib consiste à utiliser le `pyplot`.

```
[ ]: import matplotlib.pyplot as plt

fig = plt.figure()
```

Vous pouvez éventuellement fournir un numéro que vous pouvez utiliser pour accéder à une image précédemment créée.

Si aucun numéro n'est fourni, l'ID du dernier personnage créé sera incrémenté et utilisé à la place; les chiffres sont indexés à partir de 1 et non de 0.

```
[ ]: import matplotlib.pyplot as plt

fig = plt.figure()
fig = plt.figure(1) # True
```

Au lieu d'un nombre, les chiffres peuvent également être identifiés par une chaîne.

Si vous utilisez un backend interactif, cela définira également le titre de la fenêtre.

```
[ ]: import matplotlib.pyplot as plt

fig = plt.figure('image')
```

Pour choisir l'utilisation de la figure

```
[ ]: plt.figure(fig.number) # or

plt.figure(1)
```

### 1.1.2 Créer un axe

Il existe deux manières principales de créer un axe dans matplotlib:

- utiliser pyplot
- utiliser l'API orientée objet.

En utilisant pyplot:

```
[ ]: import matplotlib.pyplot as plt

ax = plt.subplot(3, 2, 1) # 3 rows, 2 columns, the first subplot
```

En utilisant l'API orientée objet:

```
[ ]: import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(3, 2, 1)
```

La fonction de commodité `plt.subplots()` peut être utilisée pour produire une figure et une collection de sous-parcelles dans une commande:

```
[ ]: import matplotlib.pyplot as plt

fig, (ax1, ax2) = plt.subplots(ncols=2, nrows=1) # 1 row, 2 columns
```

```
[ ]:
```

## 1.2 Insertion de formules TeX dans les parcelles

Les formules TeX peuvent être insérées dans le tracé à l'aide de la fonction `rc`

Exemple

```
[ ]:
```

```
[ ]: import matplotlib.pyplot as plt

plt.rc(usetex = True)

#Utilisation

plt.xlabel('\alpha')
plt.xlabel(r'\alpha')
```

Exemple de code

```
[ ]: import matplotlib.pyplot as plt

plt.rc(usetex = True)
```

```

x = range(0,10)
y = [t**2 for t in x]
z = [t**2+1 for t in x]

plt.plot(x, y, label = r'\beta=\alpha^2$')
plt.plot(x, z, label = r'\beta=\alpha^2+1$')
plt.xlabel(r'\alpha$')
plt.ylabel(r'\beta$')
plt.legend(loc=0)
plt.show()

```

### 1.2.1 Enregistrement et exportation de tracés utilisant TeX

Afin d'inclure les tracés créés avec matplotlib dans les documents TeX, ils doivent être enregistrés sous pdf fichiers pdf ou eps . Tout texte du tracé (y compris les formules TeX) est rendu sous forme de texte dans le document final.

```

[ ]: plt.savefig('my_pdf_plot.pdf') # Saving plot to pdf file

plt.savefig('my_eps_plot.eps') # Saving plot to eps file

plt.savefig("my_png_plot.png")

```

### 1.3 Fermer une fenêtre de figure

Syntaxe

```

[ ]: plt.close() # ferme la figure active en cours

plt.close(fig) # ferme la figure avec la poignée 'fig'

plt.close(num) # ferme le numéro de chiffre 'num'

plt.close(nom) # ferme la figure avec l'étiquette 'name'

plt.close('all') # ferme tous les chiffres

```

Exemple

```

[ ]: import matplotlib.pyplot as plt

fig1 = plt.figure() # création de la première figure

plt.plot([0, 1], [0, 1])

fig2 = plt.figure() # création de la seconde figure

```

```
plt.plot([0, 1], [0, 1])
```

```
plt.close(fig1) # ferme la première figure mais la seconde reste active
```