

TP n°4 : Représentations graphiques avec Python

Dans ce TP, nous allons avoir besoin d'utiliser deux nouveaux modules Python :

- `numpy` qui vous permettra de créer des tableaux (array) et d'accéder aux fonctions usuelles ;
- `matplotlib` qui vous permettra d'effectuer des représentations graphiques de fonctions à l'aide de la bibliothèque `pyplot`.

1 Le module Numpy et fonctions prédéfinies

- Ce module contient toutes les fonctions et constantes usuelles.
- Ce module permet également la création et la manipulation de tableaux (array en anglais). Dans ce TP nous travaillerons avec des tableaux de dimension 1 (aussi appelés vecteurs). Plus tard dans l'année, on sera amené à travailler avec des tableaux de dimension 2 (aussi appelés matrices).

Pour l'importation du module, on utilisera la syntaxe :

```
import numpy as np
```

Exercice n°1 Fonctions usuelles et constantes avec Numpy

Comprendre les fonctions et commandes suivantes des calculs proposés que vous taperez dans la console :

```
import numpy as np
a=np.exp(10)
b=np.log(a)
c=np.e
c
d=np.log(e)
np.sqrt(16)
np.sqrt(-2)
np.abs(-3)
np.floor(1.1)
np.floor(-0.8)
```

En fait l'importation d'un module peut se réaliser selon l'une des deux syntaxes :

```
from numpy import* ou import numpy as np.
```

La première syntaxe charge le module, l'exécute et place toutes les fonctions et constantes du module dans l'espace des noms. Si deux modules possèdent des fonctions portant le même nom, seule la fonction du dernier module importé sera exécutable (puisque son importation aura écrasé la précédente). C'est pourquoi on préférera la seconde syntaxe qui réalise une importation en notation pointée permettant de différencier les fonctions de deux modules portant le même nom.

Exercice n°2 Création de vecteurs avec les fonctions `arange` et `linspace`

Expliquer le fonctionnement des fonctions `np.arange(a,b,p)` et `np.linspace(a,b,n)` à l'aide des calculs proposés que vous taperez dans la console :

```
np.arange(2,12,2)
A=np.arange(0,1.3,0.2)
A
np.arange(-2,2.1)
np.linspace(2,12,5)
B=np.linspace(2,12,10)
B
np.linspace(-2,2.1)
A**2+1
np.log(B)
```

Expliquez les deux dernières commandes.

2 Le module Matplotlib

2.1 Tracé simple

Pour importer ce module, on utilisera la syntaxe suivante : `import matplotlib.pyplot as plt`.

Pour tracer une fonction $f : [a; b] \rightarrow \mathbb{R}$ avec Python, on suivra les étapes suivantes :

1. **On construit un vecteur x de valeurs appartenant à l'intervalle $[a; b]$.**

Il y a deux possibilités :

- avec un pas p constant : `x=np.arange(a,b,p)`
- avec un nombre de points n fixés : `x=np.linspace(a,b,n)`

2. **On définit la fonction $f : [a; b] \rightarrow \mathbb{R}$**

Il y a deux possibilités :

- soit elle est déjà définie (`exp`, `log`...)
- soit il faut la définir avec la syntaxe `def`.

3. **On trace la courbe représentative \mathcal{C}_f de f en utilisant la commande `plt.plot(x,f(x))` puis `plt.show(x,f(x))`**

Exercice n°3 Premières représentations graphiques

1. Tapez le script suivant et commenter (comprendre l'intérêt de chaque ligne) :

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(-3,3)
def f(x):
    y=x**2-1
    return y
y=f(x)
plt.plot(x,y)
plt.show()
```

Exécutez le script.

2. Modifier le vecteur x pour que ses coefficients aillent de -3 à 3 avec un pas de 0.1 .
3. Modifier le vecteur x pour qu'il comporte 100 coefficients compris entre -1 et 3 .

2.2 Fonctionnalités pour le tracé

Il existe des commandes spécifiques pour améliorer les représentations graphiques, celles-ci sont résumé dans le tableau ci-dessous :

Commandes	Signification
<code>plt.xlim(xmin,xmax)</code> <code>plt.ylim(ymin,ymax)</code> <code>plt.xlabel("nom axe abscisses")</code> <code>plt.ylabel("nom axe ordonnées")</code> <code>plt.title("nom du graphique")</code> <code>plt.legend()</code> <code>plt.axis("equal")</code> <code>plt.grid()</code> <code>plt.savefig("nom du fichier")</code>	restreint la représentation graphique à l'intervalle $[xmin,xmax]$ en abscisse restreint la représentation graphique à l'intervalle $[xmin,xmax]$ en ordonnée donne un titre à l'axe des abscisses donne un titre à l'axe des ordonnées donne un titre au graphique affiche la légende définie par <code>label="nom"</code> dans <code>plt.plot</code> pour que le repère soit orthonormé ajoute une grille au graphique sauvegarde la figure dans le dossier courant

- Pour créer la légende, il faut "labelliser" chaque tracé en écrivant : `plt.plot(x,y,label="nom de ma fonction")`

2.3 Couleurs et styles pour le tracé

On peut également changer les couleurs et le style des tracés, voici les commandes à utiliser :

Couleurs	Syntaxe
Rouge	"r"
Vert	"g"
Bleu	"b"
Cyan	"c"
Magenta	"m"
Jaune	"y"
Noir	"k"

Style	Syntaxe
Pointillés	":"
Lignes en pointillés	"--"
Traits/points	"_."
Cercle	"o"
Triangle	"v"
Carré	"s"
Signe +	"+"
Croix	"x"

- Exemple d'utilisation de ces commandes : `plt.plot(x,y,"r:",label="nom de ma fonction")`

Exercice n°4 Utilisation des fonctionnalités

Reprendre le script de l'exercice 3.

1. Changer la couleur de la courbe (par exemple en bleu).
2. Ajouter un titre et une légende.
3. Ajouter un nom à l'axe des abscisses et des ordonnées.
4. Changer les valeurs de la fenêtre graphique afin que la fenêtre fasse apparaître la courbe sur l'intervalle [2; 3].
5. Ajouter une grille à la représentation graphique.

2.4 Tracé multiple

Exercice n°5 Tracés multiples

1. Taper ce script :

```
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(0,2*np.pi,100)
y1=np.cos(x)
y2=np.sin(x)
plt.plot(x,y1,'b')
plt.plot(x,y2,'r')
plt.show()
```

2. Ajouter un titre.
3. Ajouter une légende pour chacun des deux courbes.

3 Exercices

Exercice n°6 Fonctions usuelles

1. Tracer l'une après l'autre les fonctions suivantes :
 - La fonction carrée sur l'intervalle $[-3; 3]$.
 - La fonction cube sur l'intervalle $[-3; 3]$.
 - La fonction valeur absolue sur l'intervalle $[-3; 3]$.
 - La fonction racine carrée sur l'intervalle $[0; 3]$.
 - La fonction inverse sur l'ensemble $[-2; 2] \setminus \{0\}$.
2. Ajouter un titre (le nom de la fonction à chaque tracé obtenu).

Exercice n°7

1. Représentez avec deux couleurs différentes sur l'intervalle $[-2; 2]$ avec un pas de 0.1, la fonction exponentielle et sa tangente au point d'abscisse 0
2. Ajouter un titre et une légende des deux courbes.

Exercice n°8

1. Tracer dans une fenêtre graphique et avec quatre couleurs et/ou styles différents les fonctions polynomiales suivantes sur l'intervalle $[-5; 5]$:
 - $f_1(x) = x^2 - 4x + 4$,
 - $f_2(x) = x^2 + 5x + 4$,
 - $f_3(x) = 2x^2 + x + 2$,
 - $f_4(x) = -x^2 + 5x + 4$,
2. Ajouter un titre et une légende.
3. Dans chacun des cas, déduire à partir du tracé obtenu le signe du discriminant de chaque polynôme.