I Représentation de courbes

I.1 Graphe point par point

Python fait du dessin point par point et les relie avec des segments de droite. Pour obtenir une courbe lisse, il faut un nombre élevé de points, par contre pour une droite 2 suffisent !
Dans un premier temps, on importe les modules nécessaires : *import numpy as np* et *import matplotlib.pyplot as plt*.

La fonction *plt.plot()* est la plus importante pour tracer des graphiques ! Elle prend en argument deux listes de même taille :

$$plt.plot([x_1, \cdots, x_n], [y_1, \cdots, y_n])$$

produit l'affichage d'une ligne brisée reliant les points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$.
Pour voir cette courbe, on utilise l'instruction $plt.show()$.
En clair la première liste est celle des abscisses et la seconde celle des ordonnées.

Exercice 1: Tapez les instructions suivantes :

☞ $x = [-2, 1, 2, 4, 6]$ ☞ $y = [4, 2, -3, 5, 2]$ ☞ $plt.plot(x, y)$ ☞ $plt.show()$
☞ $plt.plot([-1, 2, 5, 1, 4, 2, 3, 0])$ ☞ $plt.show()$

Après avoir affiché un graphique, il faut fermer la fenêtre pour en faire un nouveau. Si on enchaîne les instructions *plot(...)* sans les afficher, les graphiques vont se superposer en changeant de couleur au moment de l'affichage. Si l'on ne précise pas de liste y, Python considère que par défaut la première liste est $[0, 1, \cdots, n-1]$.

Si l'on veut définir plusieurs figures, on peut les numéroter avec la commande suivante $plt.figure(n)$ où n est le numéro de la figure.

Exercice 2: Tapez les instructions suivantes :

☞ $plt.figure(1)$ ☞ $x = [1, 2, 6]$ ☞ $y = [-4, 2, -1]$ ☞ $plt.plot(x, y)$ ☞ $plt.show()$
☞ $plt.figure(2)$ ☞ $u = [-1, 3, 5]$ ☞ $v = [0, 4, 0]$ ☞ $plt.plot(u, v)$ ☞ $plt.show()$

La commande $plt.close()$ ferme la figure en cours, la commande $plt.close('all')$ ferme toutes les fenêtres.

I.2 Graphe d'une fonction

On commence par créer un tableau des abscisses avec au choix les deux commandes suivantes :

$x = np.arange(xmin, xmax, pas)$ ou $x = np.linspace(xmin, xmax, nbpoints)$

Avec la commande *arange*, on crée un tableau de nombres compris entre a (inclus) et b (exclus) avec un pas de r, ie les nombres $a, a+r, a+2r, \cdots$ en s'arrêtant à $a+kr$ tel que $a+kr < b \inf a+(k+1)r$
Avec la commande *linspace*, on crée un tableau de n nombres compris entre a (inclus) et b (inclus) avec un pas de $pas = \dfrac{b-a}{n-1}$, ie les nombres $a, a+pas, a+2*/pas, \cdots, a+(n-1)*pas = b$.
Avec des valeurs entières, la fonction *arange* est similaire à la fonction *range* mais renvoie un tableau au lieu d'une liste.
Avec un pas non entier, il est conseillé d'utiliser la fonction *linspace*.
Par défaut, le nombre de points est égal à 50.

Exercice 3: Tapez les instructions suivantes :

☞ $t1 = np.arange(1, 17, 2)$ ☞ $t2 = np.arange(0.2, 4.5)$
☞ $t3 = np.linspace(1, 25, 12)$ ☞ $t4 = np.linspace(-pi, pi)$

On crée alors le tableau des ordonnées par l'instruction : $y = f(x)$

Pour créer le graphique de la fonction f, on utilise la fonction plot : plt.plot(x, y)

Exercice 4: Editez les instructions suivantes et testez-les :

```
1 x=np.arange(0,4*pi,0.01)
2 y1=np.sin(x)
3 y2=np.cos(x)
4 plt.plot(x,y1,'r') # trace en rouge
5 plt.plot(x,y2)
6 plt.xlim(0,4*pi) ; plt.ylim(-1,1) # délimite le cadre
7 plt.xlabel("Abscisses")
8 plt.title("Fonctions circulaires")
9 plt.show()
```

Exercice 5: Ecrire un code qui dessine la fonction exponentielle sur l'intervalle $[-2, 2]$. Sur le même graphe, on dessinera les fonctions polynômiales suivantes de degrés respectifs 0,1, 2 et 3 :

$$f_0 : x \mapsto 1, \quad f_1 : x \mapsto 1 + x, \quad f_2 : x \mapsto 1 + x + \frac{x^2}{2}, \quad f_3 : x \mapsto 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$

Que peut-on constater au voisinage de 0 ?

La fonction plot possède des options, concernant la couleur, le style des points ou des droites, en tapant help(plt.plot) dans le shell, vous pourrez faire des essais sur les différentes possibilités offertes.
On peut par exemple placer plusieurs graphiques sur une même fenêtre avec la commande *plt.subplot* (en français : sous-graphique). Dans *plt.subplot* on précise 3 chiffres : le nombre de lignes, le nombre de colonnes et le numéro de la case qu'on ouvre.

Exercice 6: Editez les instructions suivantes et testez-les :

```
1  plt.close('all')
2  plt.figure(1)
3  x=np.linspace(0,1,100) # Création d'un tableau d'abscisses
4  plt.subplot(2,2,1) # on coupe la fenêtre en 4 : premier graphique
5  plt.plot([1,1],linewidth=10) # on force le trait
6  plt.xlim(0,1) ; plt.ylim(0,1)
7  plt.subplot(2,2,2) # Deuxième graphique
8  plt.plot(x,x,'ro') # Un peu de couleur et des gros points
9  plt.subplot(2,2,3) # Troisième graphique
10 plt.plot(x,x**2,color=[0.8,0.1,0.7]) # on peut fabriquer la couleur que l'on
   veut
11 plt.legend(["Carrée"],loc=0) # Légende
12 plt.subplot(2,2,4) # Dernier graphique
13 plt.plot(x,x**3,'g–',linewidth=0.5)
14 plt.text(0.2,0.5,'Fonction $x\mapsto x^3$') # on place du texte où l'on
   veut, et même du code LaTeX

15 plt.show()
```

II Exercices :

Exercice 7: Créer une fonction monomes(N) qui représente les fonctions $x : x \mapsto x^k$ pour $k = 1, \cdots, n$, si possible avec une légende.

Exercice 8: Exemple de suite récurrente : $u_{n+1} = f(u_n)$ avec $f : I \to I$ et $u_0 \in I$ (I intervalle de $\mathbb{R}$).

Créez une fonction recurrente(f,u0,N,a,b) qui produit la représentation en escalier de la suite (u_n) :
Plus précisément cette fonction devra :
- tracer le graphe de f entre a et b.
- tracer la bissectrice d'équation $y = x$.
- tracer les deux axes (Ox) et (Oy) en noir.
- afficher une grille de graduation.
- tracer, avec une ligne continue et des marqueurs, l'escalier ou l'escargot représentant la suite (u_n)

On peut que les points en question ont pour coordonnées successives :

$$(u_0, u_0), (u_0, u_1), (u_1, u_1), (u_1, u_2), \cdots, (u_{n-1}, u_{n-1}), (u_{n-1}, u_n), (u_n, u_n)$$

On testera notre fonction avec les fonctions suivantes : $f = \cos$, $f(x) = \frac{1}{4}x^2 + \frac{3}{4}, \cdots$.