

## Python 8 : Représentation graphique

On peut représenter graphiquement en Python des suites, ainsi que des fonctions. On aura besoin de la bibliothèque matplotlib qui permet de faire des tracés :

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

Soient  $X=[x_1 \ x_2 \ \dots \ x_n]$  et  $Y=[y_1 \ y_2 \ \dots \ y_n]$  deux matrices lignes (ou listes) à  $n$  éléments.

La commande `plt.plot(X,Y)` permet de représenter les valeurs de  $Y$  en fonction de celle de  $X$ . Plus précisément elle trace une ligne brisée qui relie les points de coordonnées  $(x_i, y_i)$ .

En pratique il faut ajouter la commande `plt.show()` à la fin du programme pour afficher le graphique à l'écran.

### 1) Tracer les termes d'une suite

Pour tracer une suite, on va donc utiliser :

- Le vecteur  $X=[0 \ 1 \ 2 \ \dots \ n]$  qu'on peut déclarer avec `np.arange(0,n+1)`, ou encore avec `list(range(0,n+1))`
- Le vecteur  $Y=[u_0 \ u_1 \ u_2 \ \dots \ u_n]$  qui contient les termes de la suite à représenter.

Il faut donc remplir  $Y$  avec les termes de la suite. Voyons sur des exemples :

Exemple 1 :

On définit une suite récurrente par :  $u_0 = 3$  et, pour tout entier  $n$ ,  $u_{n+1} = 2u_n + 4$ .

$n=15$  # on choisit de représenter les 15 premiers termes

$Y=np.zeros(n+1)$  # ce vecteur va accueillir les valeurs de la suite ( $u_n$ )

$Y[0]=3$  # On rentre donc la valeur de  $u_0$  pour la première coordonnée de  $Y$

for  $k$  in `range(0,n)`:

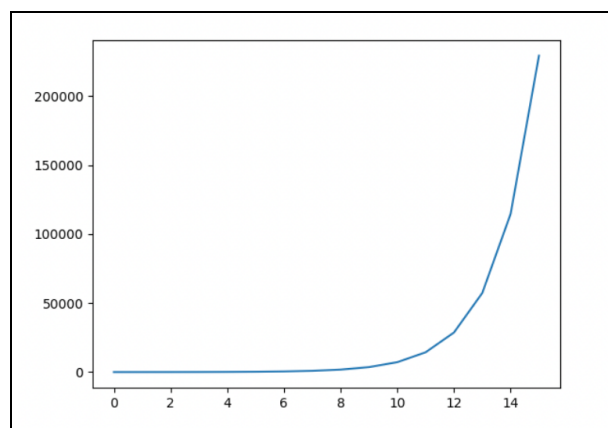
$Y[k+1]=2*Y[k]+4$  #On remplit  $Y$  avec les termes de la suite de manière successive

$X=np.arange(n+1)$

`plt.plot(X,Y)`

`plt.show()`

Il s'affiche (les points sont reliés) :



## 2) Tracer la courbe d'une fonction

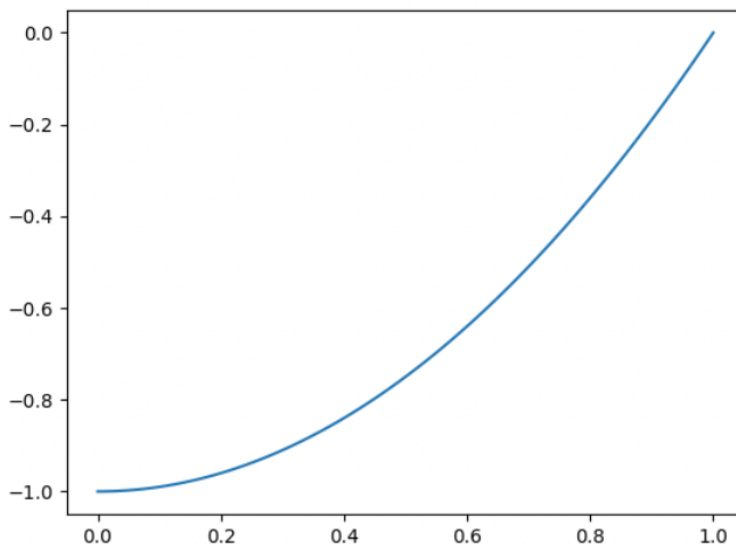
Si on veut tracer une fonction  $f$  sur le segment  $[a,b]$  on peut procéder ainsi :

```
X=np.linspace(a,b,100)      #100 est un grand nombre, 200 ou 1000 marchent aussi
plt.plot(X,f(X))
plt.show()
```

Par exemple :

```
def f(x):                    # on déclare la fonction avant, c'est plus simple
    x=x**2-1
    return x
X=np.linspace(0,1,100)      # on la représente sur le segment [0,1]
plt.plot(X,f(X))
plt.show()
```

Affiche ceci (la courbe de la fonction  $f(x) = x^2 - 1$  sur le segment  $[0,1]$ ) :



### ➤ Comment cela fonctionne ?

La commande `linspace(a,b,n)` renvoie un vecteur ligne qui contient  $n$  nombres régulièrement espacés entre  $a$  et  $b$ , donc cela imite le segment  $[a,b]$  si on met une grande valeur de  $n$ , ainsi, si on représente  $x$  en fonction de  $f(x)$  on va représenter toutes les images du segment  $[a,b]$ , ce qui exactement la définition de la courbe d'une fonction.

### ➤ Quelques astuces pratiques :

- Si on veut tracer une fonction sur un intervalle non fermé, par exemple  $]0,1]$  on peut simplement utiliser `linspace(0.001,1,100)` pour éviter le point de non définition
- Pour tracer la fonction inverse d'une fonction  $f$ , il suffit d'écrire `plot(f(x),x)`