

FICHE : TRACER DES COURBES

Le module `numpy` contient des outils permettant de générer et de manipuler des tableaux de nombres.

```
import numpy as np
```

Générer des listes d'abscisses

```
L1 = list(range(n)) .....
```

```
L2 = np.linspace(a,b,n) .....
```

```
L2 = np.arange(a,b,p) .....
```

Le module `matplotlib.pyplot` contient des outils permettant de tracer des graphiques.

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

Placer des points ou relier des points sur un graphique

```
plt.plot(a,b,'o') .....
```

```
plt.plot([x1,x2],[y1,y2],'o') .....
```

```
plt.plot([x1,x2],[y1,y2]) .....
```

```
plt.plot(L1,L2) .....
```

```
plt.show() .....
```

Remarque : On retient que pour tracer un segment, il suffit de donner les coordonnées de deux points à relier.

Exemple : `plt.plot([-5,5],[0,0])`

Pour tracer une courbe, on construit donc en fait un nuage de points (reliés entre eux) : on commence toujours par générer une liste (ou un tableau) contenant des valeurs d'abscisses et un tableau contenant les ordonnées correspondantes.

Tracer la courbe représentant une fonction f

```
X=np.linspace(a,b,n)
Y=[np.log(t) for t in X] ou Y=np.log(X) .....

plt.plot(X,Y)
plt.show()
```



Attention, pour « appliquer » une fonction mathématique à un tableau (X), il faut que celle-ci provienne du module **numpy** et non du module **math**.

On peut ajouter un troisième argument (optionnel) à la fonction **plot** pour préciser

- le symbole utilisé pour les points

'.' : 'h' : 'x' :

'o' : '+' : '*' :

- la couleur

'r' : 'g' : 'm' :

'b' : 'c' : 'k' :

- le type de ligne

'-' : '- -' : '-.' :

- le nom que l'on souhaite donner à la courbe si on décide d'afficher une légende

label='courbe 1'

On peut aussi ajouter du texte sur le graphique à une certaine position :

```
plt.text(x,y,'blabla',color='b') .....
```

insère le texte « blabla » (en bleu) au niveau du point de coordonnées (x,y) .

Il peut enfin être agréable de modifier les paramètres de la fenêtre graphique ou d'ajouter une légende.

Quelques options d'habillage du graphique

```
plt.axis('equal') ou plt.axis([xmin,xmax,ymin,ymax]) .....

plt.grid() .....

plt.xlabel('blabla') .....

plt.ylabel('blabla') .....

plt.title('blabla') .....

plt.legend() .....
```