Chapitre 2 : Afficher ses résultats sous forme graphique

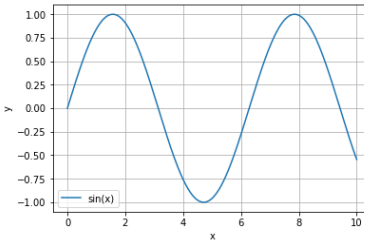
La bibliothèque maptplotlib.pyplot possède de nombreuses fonctions déjà écrites permettant d'effectuer des tracés. L'importation des fonctions de ce module s'effectue par la commande suivante :

```
import matplotlib.pyplot as plt# Préfixe plt à utiliser pour importer toutes les fonctions
```

a) Tracé d'une courbe

Afin de tracer une fonction dont la définition est connue, il est nécessaire de définir son intervalle d'étude ainsi que son pas en abscisse :

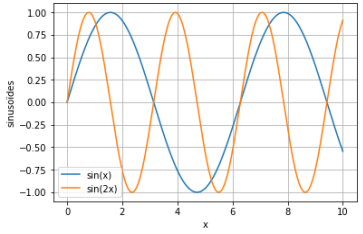
```
x=np.linspace(0,10,1000)#on génère les valeurs en abscisse
y=np.sin(x)# fonction à tracer
plt.xlabel("x")#nom de l'abscisse
plt.ylabel("y")#nom de l'ordonnée
plt.plot(x,y,label ='sin(x)')#les points (x,y) sont reliés par des segments
plt.legend()#permet de faire apparaître la légende (ici sin(x))
plt.grid()#grille
plt.show()#montre le graphe courant
```



b) Tracés de plusieurs courbes

Il est courant de devoir comparer plusieurs courbes :

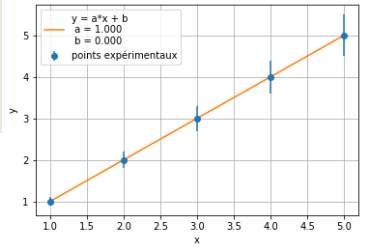
```
y1=np.sin(x)# fonction à tracerplt.grid()#grille
y2=np.sin(2*x)
plt.xlabel("x")#nom de l'abscisse
plt.ylabel("sinusoïdes")#nom de l'ordonnée
plt.plot(x,y1,label ='sin(x)')#on indique la fonction y1(x) à tracer
plt.plot(x,y2,label ='sin(2x)')#on indique la fonction y1(x) à tracer
plt.legend()#permet de faire apparaître la légende (ici sin(x))
plt.grid()#grille
plt.show()#montre le graphe courant
```



c) Représentations graphiques de résultats expérimentaux et test d'une loi affine

Lors de mesures, on obtient des couples (x_i, y_i) avec souvent une incertitude-type $u(y_i)$ à prendre en compte (barres d'erreurs). Une série de N mesures (x_i, y_i) peut permettre de tester une droite de régression $y(x)$:

```
X2=np.array([1,2,3,4,5])
Y2=X2
u_Y2=Y2*0.1#10% d'erreur
plt.xlabel("x")#nom de l'abscisse
plt.ylabel("y")#nom de l'ordonnée
plt.errorbar(X2,Y2,linestyle="",marker='o',yerr=u_Y2,label="points expérimentaux")#points exp (sans segment) + barres d'erreur
p=np.polyfit(X2,Y2,1)#Régression linéaire : on trouve le polynôme d'ordre 1 (ax+b) avec p tableau [a,b]
a=p[0]#pente
b=p[1]#Y(0)
plt.plot(X2,a*X2+b,label="y = a*x + b \n a = {0:.3f} \n b = {1:.3f}".format(a,b))#f pour indiquer que la valeur est flottant à 3 décimales
plt.legend()#permet de faire apparaître la légende (ici sin(x))
plt.grid()#grille
plt.show()#montre le graphe courant
```



d) Lignes de champ

La cartographie de certains champs de vecteurs (champ magnétique, champ des vitesses d'un écoulement, …) nécessite l'utilisation des fonctions quiver et streamplot. Voici l'exemple du champ des vitesses d'un solide en rotation autour d'un axe Oz fixe et observé dans un plan perpendiculaire : $\vec{v}(r) = r\omega\overrightarrow{u_\theta} = -\omega y\overrightarrow{u_x} + \omega x\overrightarrow{u_y}$

```
X3=np.linspace(-1,1,10)#zone d'observation
Y3=X3
Xm,Ym=np.meshgrid(X3,Y3)# Xm est un tableau indiquant l'abscisse de tous les pts, Ym pour les ordonnées
Vx=-Ym# w=1rad/s
Vy=Xm# w=1 rad/s
plt.streamplot(Xm,Ym,Vx,Vy)#lignes de champs
plt.quiver(Xm,Ym,Vx,Vy)#vecteurs champs
plt.show()
```