

**Numpy : import numpy as np**

`np.array(L)` ---- Transforme la liste  $L$  en matrice numpy    `np.transpose(M)` ----- transposée de  $M$     `np.min(M)` ----- le plus petit élément de  $M$   
`np.zeros([n,m])` ----- matrice nulle de taille  $n \times m$     `np.dot(M,P)` ----- produit matriciel  $MP$     `np.shape(M)` ----- format de la matrice  $M$   
`np.eye(n)` ----- matrice unité de taille  $n$     `np.sum(M)` ----- somme de tous les éléments de  $M$     `np.arange(a,b,p)` ---- nombres ente  $a$  et  $b$  avec un pas  $p$ .  
`np.diag(L)` ----- matrice diagonale    `np.max(M)` ----- le plus grand élément de  $M$

**Liste**

`[]` ----- une liste vide    `L.remove(a)` ----- enlève une fois  $a$  à  $L$   
`[a] * n` - une liste avec  $n$  fois l'élément  $a$     `max(L)` ----- le plus grand élément de  $L$   
`L.append(a)` ----- Ajoute  $a$  à la fin de  $L$     `min(L)` ----- le plus petit élément de  $L$   
`L1 + L2` --- Concatène les listes  $L1$  et  $L2$     `sum(L)` ----- somme des éléments de  $L$   
`len(L)` -- nombre d'éléments de la liste  $L$   
`L.pop(k)` --- enlève l'élément d'indice  $k$

**random : import random as rd**

`rd.random()` ----- un nombre au hasard entre 0 et 1, 1 exclus  
`rd.randint(a,b)` ----- un nombre entier au hasard entre  $a$  et  $b$  inclus  
`rd.choice(L)` ----- Choisit aléatoirement un élément de la liste.

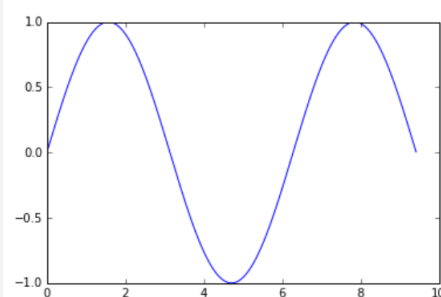
**math : import math as m**

`m.atan(x)` -----  $\arctan(x)$     `m.exp(x)` -----  $e^x$   
`m.factorial(n)` -----  $n!$  si  $n \in \mathbb{N}$     `m.sin(x)` -----  $\sin x$   
`m.sqrt(x)` -----  $\sqrt{x}$  si  $x \geq 0$     `m.cos(x)` -----  $\cos x$   
`m.log(x)` -----  $\ln(x)$  si  $x > 0$

**Tracé de courbe**

```
import matplotlib.pyplot as plt

def f(x):
    return math.sin(x)
X = np.arange(0, 3*np.pi, 0.01)
Y = [ f(x) for x in X ]
plt.plot(X, Y)
plt.show()
```

**numpy.linalg import numpy.linalg as la**

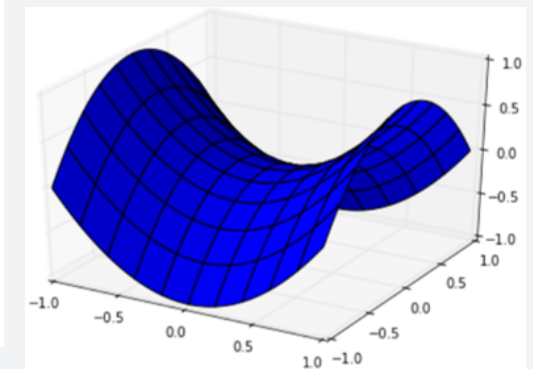
`la.inv(M)` ---- l'inverse de la matrice  $M$     `la.matrix_rank(M)` ----- rang de  $M$   
`la.eig(M)` liste des valeurs propres de  $M$     `la.det(M)` ----- déterminant de  $M$   
 et matrice de passage associée

**Tracé de surfaces**

```
from mpl_toolkits.mplot3d
    import Axes3D
import matplotlib.pyplot as plt

ax = Axes3D(plt.figure())
def f(x,y):
    return x**2- y**2
X = np.arange(-1,1,0.02)
Y = np.arange(-1,1,0.02)
X, Y = np.meshgrid(X, Y)
Z = f(X, Y)
```

```
ax.plot_surface(X, Y, Z)
plt.show()
```

**Tracé de lignes de niveau**

```
import matplotlib.pyplot as plt
plt.show()

def f(x,y):
    return x**2+ y**2+ x*y
X = np.arange(-1,1,0.01)
Y = np.arange(-1,1,0.01)
X, Y = np.meshgrid(X, Y)
Z = f(X, Y)
plt.axis('equal')
plt.contour(X, Y, Z,
    [0.1,0.4,0.5])
```

