

TP5 : CORRIGÉ

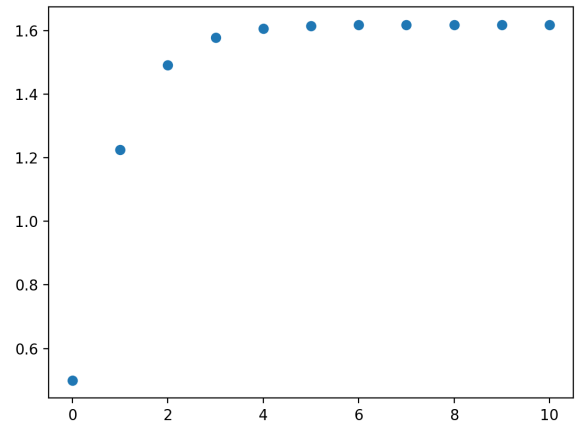
REPRÉSENTER LES PREMIERS TERMES D'UNE SUITE

```
import numpy as np
import matplotlib.pyplot as plt

def termes(n) :
    '''Renvoie la liste [u0,...,un] pour u0=0.5
    et u(n+1)=racine(1+u(n))'''
    u = 0.5
    L = [u]
    for i in range(1,n+1) :
        u = np.sqrt(1+u)
        L.append(u)
    return L

def nuage(n) :
    '''Affiche le nuage des points (n,u(n))'''
    Abs = list(range(n+1))
    Ord = termes(n)
    plt.plot(Abs,Ord, 'o')
    plt.show()

# Il semblerait que la suite (un) converge vers
# 1,6 environ
```



TRACER UN GRAPHIQUE : PREMIERS EXEMPLES

```
import numpy as np
import matplotlib.pyplot as plt

X=np.linspace(-2,1,50)

Y1=[np.exp(t) for t in X]
Y2=[-t for t in X]

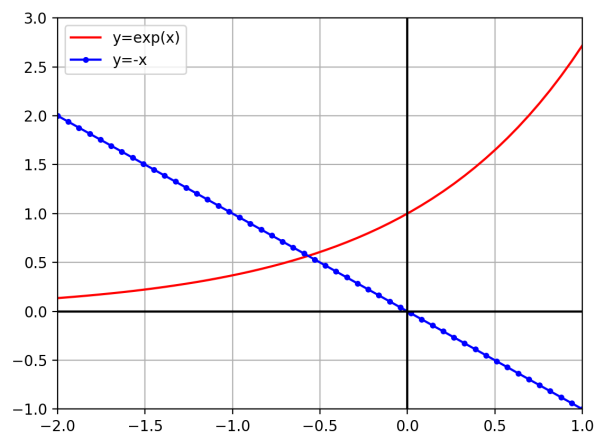
plt.plot(X, Y1, 'r', label='y=exp(x)')
plt.plot(X, Y2, 'b-', label='y=-x')

plt.axis([-2,1,-1,3])
#plt.axis('equal')

plt.plot([-2,1],[0,0], 'k') # axe des abscisses
plt.plot([0,0],[-1,3], 'k') # axe des ordonnées

plt.grid()
plt.legend()

plt.show()
```



REPRÉSENTER UNE FAMILLE DE FONCTIONS

```
import numpy as np
import matplotlib.pyplot as plt

# Données globales

V0 = 10
h = 2
g = 9.8

def hauteur(x, alpha) :
    """
    x est un réel correspondant à l'abscisse du projectile
    alpha est l'angle de lancer
    Renvoie la hauteur du projectile lorsqu'il a pour abscisse x
    """
    return -(g/(2*V0**2*np.cos(alpha)**2))*x**2 + x*np.tan(alpha) + h

def trajectoire(alpha) :
    """
    Génère la trajectoire du projectile si l'angle de lancer est alpha
    """
    X = np.linspace(0,15,200)
    Y = [hauteur(t, alpha) for t in X]
    Xplus = [t for t in X if hauteur(t, alpha)>=0]
    Yplus = [hauteur(t, alpha) for t in X if hauteur(t, alpha)>=0]
    plt.plot(Xplus, Yplus, label='alpha = '+str(round(alpha,2)))

def plusieursTrajectoires(L) :
    """
    Pour chaque élément alpha de la liste de nombres L :
    Génère et affiche sur le même graphique les trajectoire du projectile si l'angle de
    lancer est alpha
    """
    for alpha in L :
        trajectoire(alpha)
    plt.plot([0,15],[0,0], 'k')
    plt.axis([0,15,-1,6])
    plt.legend()
    plt.xlabel('Distance (m)')
    plt.ylabel('Hauteur (m)')
    plt.grid()
    plt.show()
```

