

```

# -*- coding: utf-8 -*-
"""
Created on Fri Aug 28 11:31:56 2020

@author: Fabrice
"""
##résolution de l'équadiff z''=g-lambda/m*z' chute libre dans milieu avec
##coefficient de frottement lambda

import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

#définition zone graphique
fig=plt.figure(figsize=(8,8))
ax=fig.add_subplot(111)

##chute libre avec frottements
#grandeurs physiques masse et gravité
m = 1
g = 9.81
lambda=0.4

#conditions initiales en t=0 pour la fonction odeint position0 et vitesse0
position0, vitesse0 = 0 , 0
initial=[position0, vitesse0]

#base de temps
t = np.linspace(0, 10, 101)

#paramètres dz/dt = [vitesse , accélération] pour la fonction odeint
def equadiff(z, t,g,lambda,m):
    position , vitesse = z

    #fichier à 2D: vitesse et vitesse'=g-(lambda/m)*vitesse
    dzdt=[vitesse , g-(lambda/m)*vitesse]

    return dzdt

#liste des coefficients de frottement
liste=[0.4]

#boucle qui dessine les graphiques (possibilité de mettre plusieurs éléments
## dans liste !)
for lambda in liste:

    #intégration
    sol = odeint(equadiff , initial, t , args=(g,lambda,m))

    #position = sol[:,0] vitesse = sol[:,1]
    ax.plot(t,sol[:,0], marker='o', markersize = 2,
            label='position z, coefficient de frottement= '+str(lambda))
    ax.plot(t,sol[:,1], marker='o', markersize = 2,
            label='vitesse v, coefficient de frottement= '+str(lambda))

#paramètres des graphiques
ax.grid(which='both')
ax.set_xlabel('temps de chute (s)')
ax.set_ylabel('déplacement (m) / vitesse (m/s)')
ax.set_title('Chute libre avec frottement, horaire vitesse et position')

#dessin des graphiques
fig.tight_layout()
plt.legend()
plt.show()

```