```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep  2 16:20:53 2022

@author: fabrice
"""
#équation à résoudre: x''(t)-ε(1-x**2)x'(t) +x = Asin(wt)


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


fig = plt.figure(figsize = (9,9))
ax = fig.add_subplot()

from matplotlib import rc
font_properties = {'size' : 14}
rc('font' , **font_properties)

##définition des styles de lignes, maximum 5 graphes
linestyles  = [{'ls':'-'} , {'ls':'--'} , {'ls':':'} , {'ls':'-.'} ,
               {'dashes':[2,4,2,4,8,4]}]


#paramètres de l'équation A=amplitude du forçage, w=fréquence du forçage
#essayer !
A , w = 1.2 , 2*np.pi/10

#paramètre d'amortissement ε = e faire des essais ! intéressant pour 8.53 (chaos)
e = 8.53

#conditions initiales deux valeurs pour x0: x01 et x02 afin de tester la sens
#ibilité aux conditions initiales
x01 , dx0 = 0 , 0
x011 = x01 , dx0

x02 , dx0 = 0.1 , 0
x022 = x02 , dx0

vect_init = x011 , x022

#fichier base de temps
tmin , tmax  = 0 , 200
deltat = 1000
t = np.linspace(tmin , tmax, deltat)

#fonction qui renvoie dx/dt et d2x/dt2 à l'instant t
#on pose x1=x et x2=dx/dt
def derive(t , y):
    x1 , x2 = y
    dx1dt = x2
    dx2dt = e*(1-x1**2)*x2  - x1 + A*np.sin(w*t)
```

```
    return dx1dt , dx2dt

j = 0 #compteur pour les graphiques (2 graphiques car 2 conditions initiales)

#boucle qui fait faire le calcul de x pour les deux valeurs de x0
for xx in vect_init:

    #intégration numérique de l'edo
    solution = solve_ivp(derive , (tmin, tmax) , xx , dense_output = True)

    #extraction des solutions x et sa dérivée sur la base des t
    x  = solution.sol(t)[0]

    #dessine les graphiques pour chaque x0: x(t)
    ax.plot(t , x , c = 'k' , label = 'x(t)  x0='+str(xx[0]) , **linestyles[j] )

    j+=1

#affiche les graphiques
ax.set_xlabel('temps (s)')
ax.set_ylabel('x(t) (a.u)')
ax.grid()
plt.legend(loc = 'upper right' )
plt.suptitle("""
             Amplitude du forçage A = 1.2
             Fréquence du forçage ω = 2π/10
             Paramètre d'amortissement ε = """+str(e)
             )
fig.tight_layout()
plt.show()
```