import matplotlib.pyplot as plt

import numpy as np

class ODESolver(object):

    def \_\_init\_\_(self, omega\_0 = 0, theta\_0 = 10, eta=0.01, n\_iter=10):

        self.omega\_0 = omega\_0

        self.theta\_0 = theta\_0

        self.eta = eta

        self.n\_iter = n\_iter

    def euler(self,alpha):

        self.time\_ = np.zeros(self.n\_iter)

        self.omega\_ = np.zeros(self.n\_iter)

        self.theta\_ = np.zeros(self.n\_iter)

        self.omega\_[0] = self.omega\_0

        self.theta\_[0] = self.theta\_0\*np.pi/180.0

        for i in range(self.n\_iter-1):

            self.time\_[i+1] = self.time\_[i] + self.eta

            self.omega\_[i+1] = self.omega\_[i] + self.eta\*alpha(self.theta\_[i])

            self.theta\_[i+1] = self.theta\_[i] + self.eta\*self.omega\_[i]

        return self

    def verlet(self,alpha):

        self.time\_ = np.zeros(self.n\_iter)

        self.theta\_ = np.zeros(self.n\_iter)

        self.theta\_[0] = self.theta\_0\*np.pi/180.0

        self.time\_[1]= self.eta

        self.theta\_[1] = self.theta\_[0]+self.omega\_0\*self.eta +0.5\* (self.eta\*\*2)\*alpha(self.theta\_[0])

        for i in range(self.n\_iter-2):

            self.time\_[i+2] = self.time\_[i+1] + self.eta

            self.theta\_[i+2] = 2.0\*self.theta\_[i+1] -self.theta\_[i] + (self.eta\*\*2)\*alpha(self.theta\_[i+1])

        return self

def alpha(x):

    return -np.sin(x)

time=ODESolver(omega\_0 = 0, theta\_0 = 10, eta=0.1, n\_iter=300).euler(alpha).time\_

theta=ODESolver(omega\_0 = 0, theta\_0 = 10, eta=0.1, n\_iter=300).euler(alpha).theta\_

plt.plot(time,theta\*180/np.pi,lw=3,color='red')

plt.xlabel('time(s)',size=13)

plt.ylabel('angle (deg)',size=13)

plt.title('Euler Method',size=13)

plt.show()

time=ODESolver(omega\_0 = 0, theta\_0 = 10, eta=0.1, n\_iter=300).verlet(alpha).time\_

theta=ODESolver(omega\_0 = 0, theta\_0 = 10, eta=0.1, n\_iter=300).verlet(alpha).theta\_

plt.plot(time,theta\*180/np.pi,lw=3,color='blue')

plt.xlabel('time(s)',size=13)

plt.ylabel('angle (deg)',size=13)

plt.title('Verlet Method',size=13)

plt.show()



