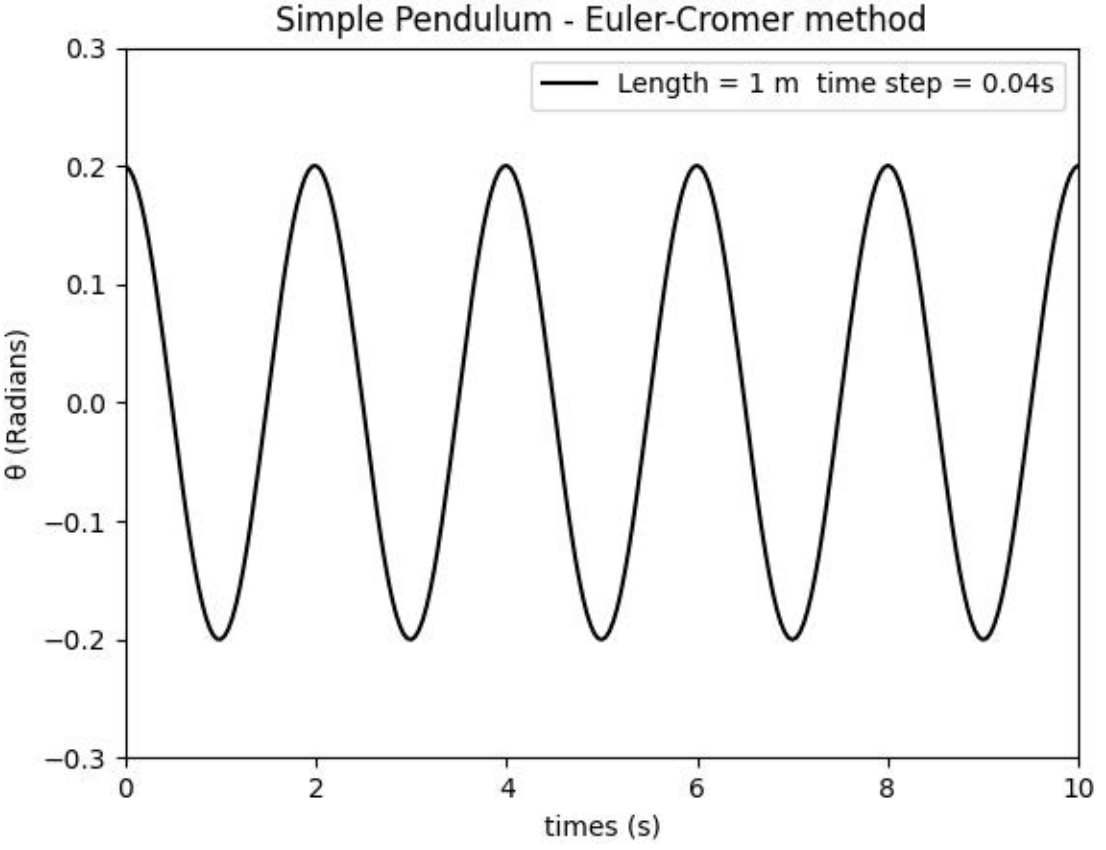
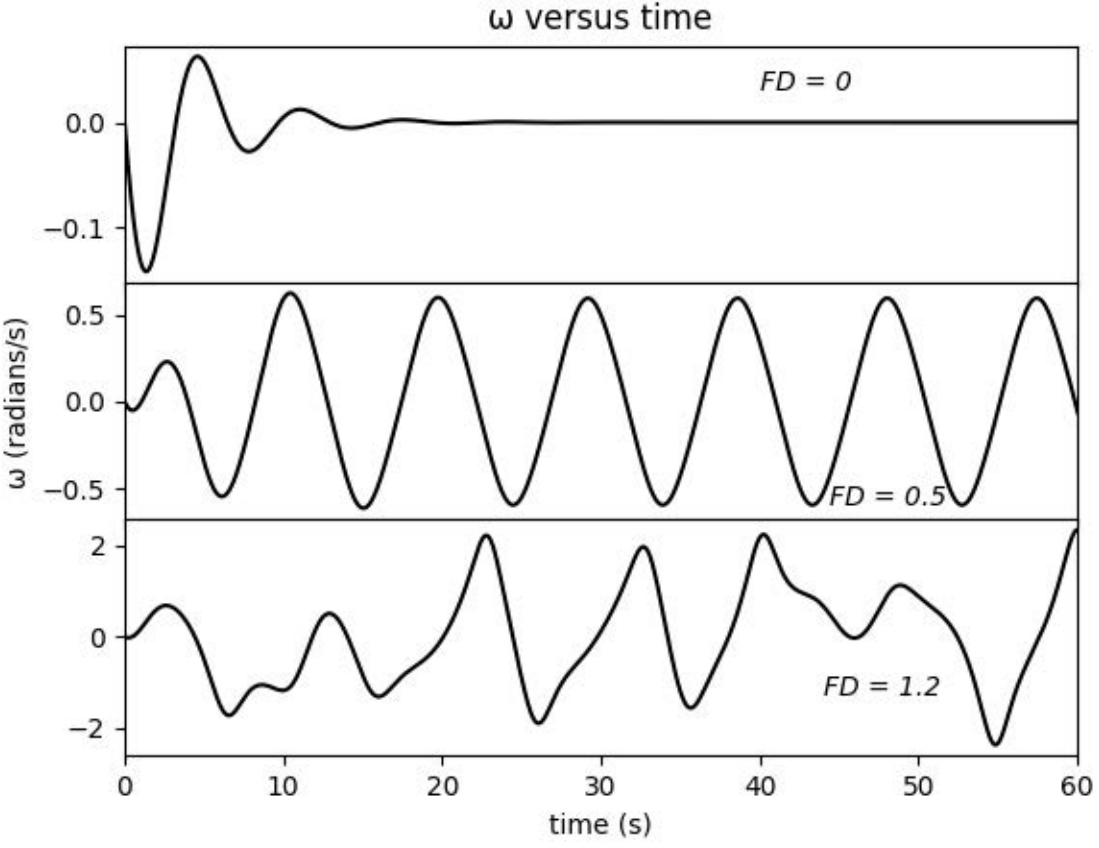
```
import matplotlib.pyplot as plt
import math

W = 0
THETA = 1
TIME = 2
N = 2000


def main(problem=0, F=0):
    if(problem == 3.3):
        for i in range(0, N):
            w_i = data[i][W] - (g/l)*data[i][THETA]*dt
            θ_i = data[i][THETA] + w_i * dt
            t_i = data[i][TIME] + dt
            data[i + 1] = [w_i, θ_i, t_i]
    else:
        for i in range(0, N):
            w_i = data[i][W] + (-(g/l) * math.sin(data[i][THETA]) -
                                q*data[i][W] + F * math.sin(Ω * data[i][TIME])) * dt
            θ_i = data[i][THETA] + w_i * dt
            if(θ_i < -math.pi):
                θ_i = θ_i + 2*math.pi
            elif(θ_i > math.pi):
                θ_i = θ_i - 2*math.pi

            t_i = data[i][TIME] + dt
            data[i + 1] = [w_i, θ_i, t_i]


def plot(x, y, title, labelx, labely, style='-', legendLabel=FileNotFoundError):
    if(labelx is not None):
        plt.xlabel(labelx)
    if(labely is not None):
        plt.ylabel(labely)
    if(title is not None):
        plt.title(title)
    line = plt.plot(x, y, style, label=legendLabel, color='k')
    if(legendLabel is not None):
        plt.legend()
    return line


if __name__ == '__main__':
    ω = 0
    θ = 0.2
    t = 0
    g = 9.8
    l = 1
    dt = 0.04

    # Amplitude
    F = [0, 0.5, 1.2]
    # Angular Frequency
    Ω = (2/3)
    # strength of damping
```

```
q = 0.5
# the mass
m = 1

data = dict()
data[0] = [ω, θ, t]
main(problem=3.3)

radians = []
time = []
for key in data.keys():
    radians.append(data[key][THETA])
    time.append(data[key][TIME])
plot(time, radians, "Simple Pendulum - Euler-Cromer method", "times (s)",
     "θ (Radians)", legendLabel="Length = "+str(l)+" m  time step = "+str(dt)+"s")
plt.xlim([0, 10])
plt.ylim([-0.3, 0.3])
plt.show()

# Change the inital value of the length
l = 9.8

fig = plt.figure()
gs = fig.add_gridspec(3, hspace=0)
axs = gs.subplots(sharex=True, sharey=False)

i = 0
for damping in F:
    data.clear()
    data[0] = [ω, θ, t]
    main(problem=3.6, F=damping)

    w_list = []
    time = []
    for key in data.keys():
        w_list.append(data[key][W])
        time.append(data[key][TIME])
    axs[i].plot(time, w_list, '-', color='k')
    i = i + 1
axs[0].text(40, 0.03, 'FD = 0', style ='italic', color ="black")
axs[1].text(44.4, -0.6, 'FD = 0.5', style ='italic', color ="black")
axs[2].text(44, -1.3, 'FD = 1.2', style ='italic', color ="black")
axs[1].set_ylabel('ω (radians/s)')
plt.xlabel('time (s)')
plt.xlim([0, 60])
axs[0].set_title('ω versus time')
plt.show()
```


Simple Pendulum - Euler-Cromer method
Length = 1 m time step = 0.04s
θ (Radians)
times (s)
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
2
4
6
8
10


ω versus time
FD = 0
FD = 0.5
FD = 1.2
ω (radians/s)
time (s)
0.0
−0.1
0.5
0.0
−0.5
2
0
−2
0
10
20
30
40
50
60