

```

# -*- coding: utf-8 -*-
"""
Created on Tue Oct  2 18:00:41 2018

@author: maxhu
"""
#Question 1 Part E showing the period graphed at different displacements and a small and large y'/y

import matplotlib.pyplot as plt
import numpy as np

# constants
v_0 = 0
t_0 = 0
m = 1
delta_t = 10**(-2)
y_0 = 0
k = 9*10**9
q1 = 1*10**-3
q2 = 1*10**-3
m2 = 1
g = 9.8
y_o = np.sqrt((k*q1*q2)/m2*g)
A_0 = .01
y_min = 0
y_max = 2

# lists to store data
yvalues = np.linspace(y_min, y_max, 1000)
T_values = []
T2_values = []

# nested loop
# for loop handles different lengths of pendulum
for n in range(len(yvalues)):
    # set initial conditions for looping calculations
    flag = True
    t_i = t_0
    v_i = v_0
    y_i = yvalues[n]
    A_i = A_0
    counter = 0

    # while loop handles omega and theta values
    while flag == True:
        v_f = g*delta_t*((1/(1+A_i)**2)-1)+v_i
        y_f = y_i + v_f*delta_t
        t_f = t_i + delta_t
        A_f = (y_i + v_f*delta_t)/y_o

    #         y_values.append(y_f)
    #         v_values.append(v_f)
    #         time_values.append(t_f)

    v_i = v_f
    y_i = y_f
    t_i = t_f

```

```

A_i = A_f

# when your current position is really close to the starting position
# exit the while loop because you've gone a full period
if (np.abs(y_i - yvalues[n]) < 10**(-3) and counter >= 50):
    flag = False

counter = counter + 1

T_values.append(t_i)
T2_values.append( T_values[n]**2 )

# This prints the values for periods at different iterations. The small
# displacement ones will be at the beginning of the lists.
#print(T_values)

fig = plt.figure(1)
my_fig = fig.add_subplot(1,1,1)
plt.grid(True)
plt.plot(yvalues, T_values, color = 'black', label = '$Y^{\prime}(\theta)/Y_o$ = .01', linestyle = '-')

my_fig.set_xlabel('Displacement')
my_fig.set_ylabel('T Values (1/s)')
my_fig.set_title('The Period at Various Displacements')
plt.legend(loc = 'lower right')

plt.savefig('TvDisplA01.png', dpi=300)

```