

# Maddox Gonzalez

## Contents

---

- Project 9
- Due 4/13/25

## Project 9

---

### Due 4/13/25

---

```
% Clear command window, workspace, and close all figures
clc; clear; close all

% Define nonlinear 2nd-order pendulum ODE
function [qdot] = pendulum_nonlinear_ODE(~,q)
    % Define constant values for pendulum
    L = 15;           % Pendulum Length
    g = 9.81;          % Gravity
    global C_d;        % Coefficient of drag (reference)

    % Represent 2nd-order ODE as two 1st-order ODEs that are coupled
    qdot(1,1) = q(2);
    qdot(2,1) = -(g/L)*sin(q(1)) - C_d*q(2)*abs(q(2));
end

% Define coefficient of drag variable
global C_d;           % Define as global so ODE function can see it
drag = [0, 0.5];       % Case 1 is no drag, case 2 considers drag: [no, yes]

% Define the period of the pendulum
period = 8.3380;
T = 0:period:period*9;

% Define total time conditions for ODE
q0 = [pi/3, 0];         % [init_pos, init_vel]

% Define total time interval for our solution
t_range = [0, 200];      % [init_time, final_time]

% Define vector of tolerances for ode45 function
tol = [1e-3, 1e-6];

% Loop for different values of drag
for i = 1:length(drag)
    C_d = drag(i);
    t_eval = 12;
    if C_d ~= 0
        t_eval = 11.5;
    end

    % Loop for different tolerances
    for j = 1:length(tol)
        options = odeset('RelTol', tol(j), 'AbsTol', tol(j));
        [t, q] = ode45(@pendulum_nonlinear_ODE, t_range, q0, options);
```

```

% Convert numerical value of tolerance and drag to strings of text
temp = num2str(tol(j));
temp1 = num2str(C_d);

fprintf((strcat("Solutions implementing Cd = ",temp1," and tol = ", temp,"\\n")));
fprintf('Displacement: %.8f at t = %.2f s\\n', interp1(t, q(:,1), t_eval), t_eval);
fprintf('Velocity: %.8f at t = %.2f s\\n\\n', interp1(t, q(:,2), t_eval), t_eval);

figure, hold on

subplot(2,1,1)

if i == 1, plot(t,q(:,1),'b',T,q0(1),'o')
else plot (t,q(:,1),'b'),
end
grid on
title(['Displacement vs Time (Cd = ', temp1, ', tol = ', temp, ')'])
xlabel('Time (seconds)')
ylabel('Displacement (radians)')

subplot(2,1,2)
plot(t,q(:,2),'r')
title(['Velocity vs Time (Cd = ', temp1, ', tol = ', temp, ')'])
xlabel('Time (seconds)')
ylabel('Velocity (radians/second)')

figure;
plot(q(:,1), q(:,2), 'b');
title('Phase Plot');
xlabel('Displacement ');
ylabel('Velocity ');
grid on;
end
end

```

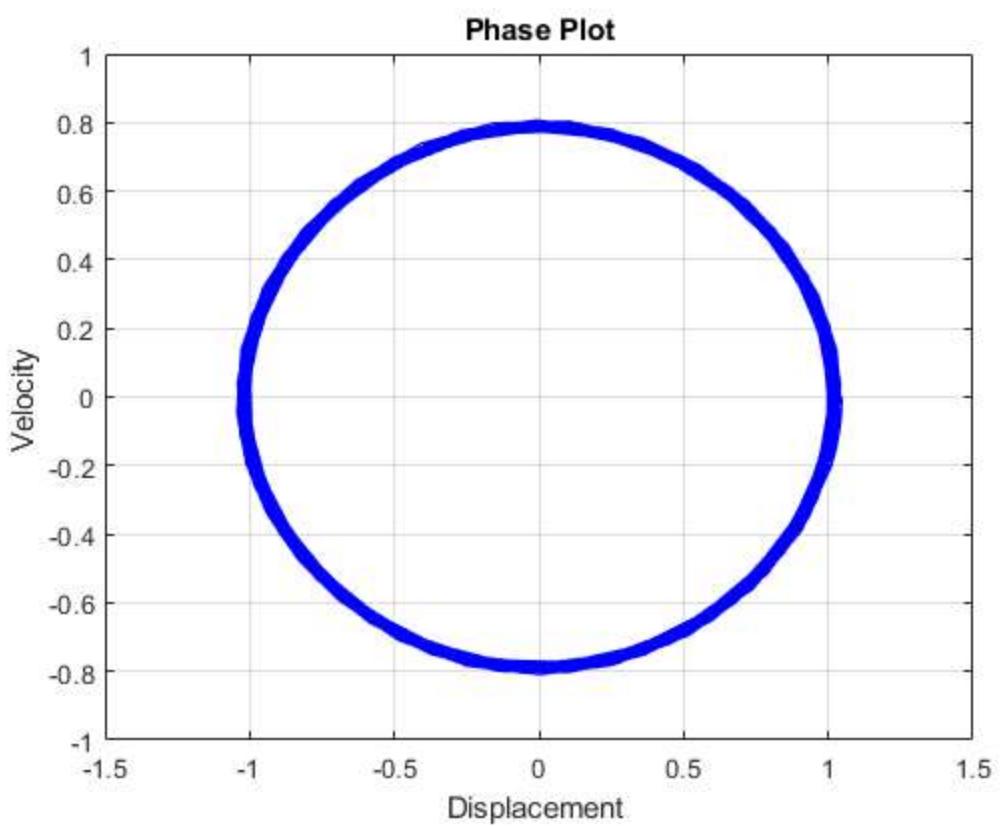
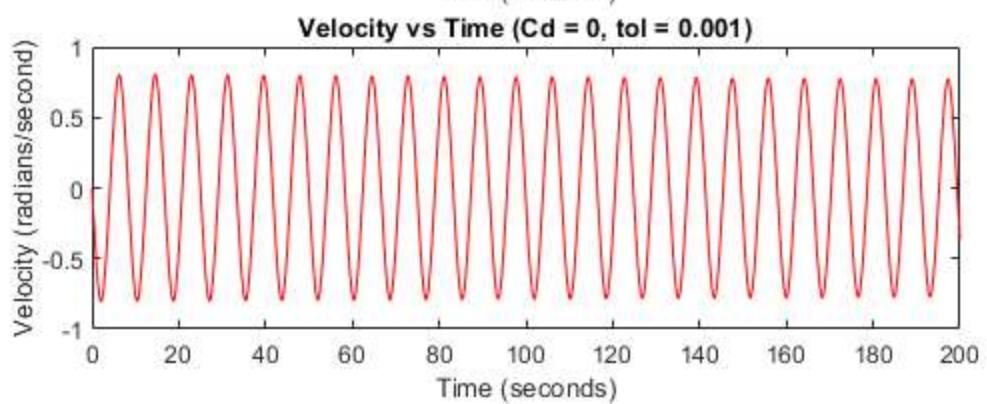
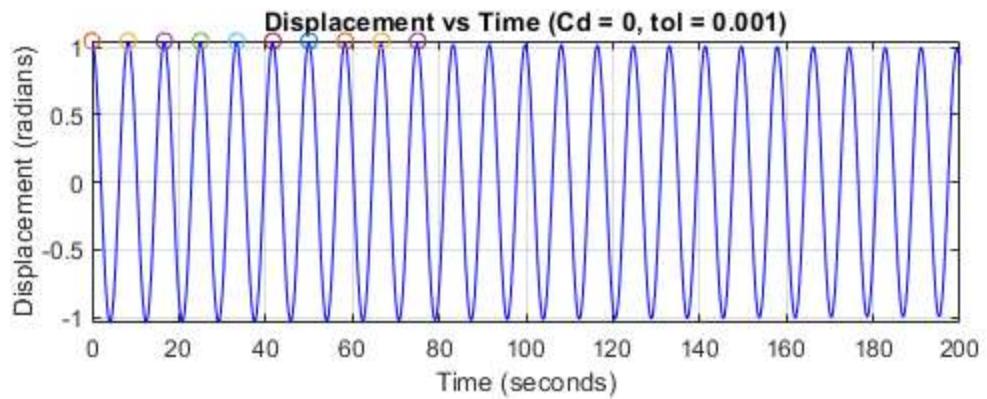
---

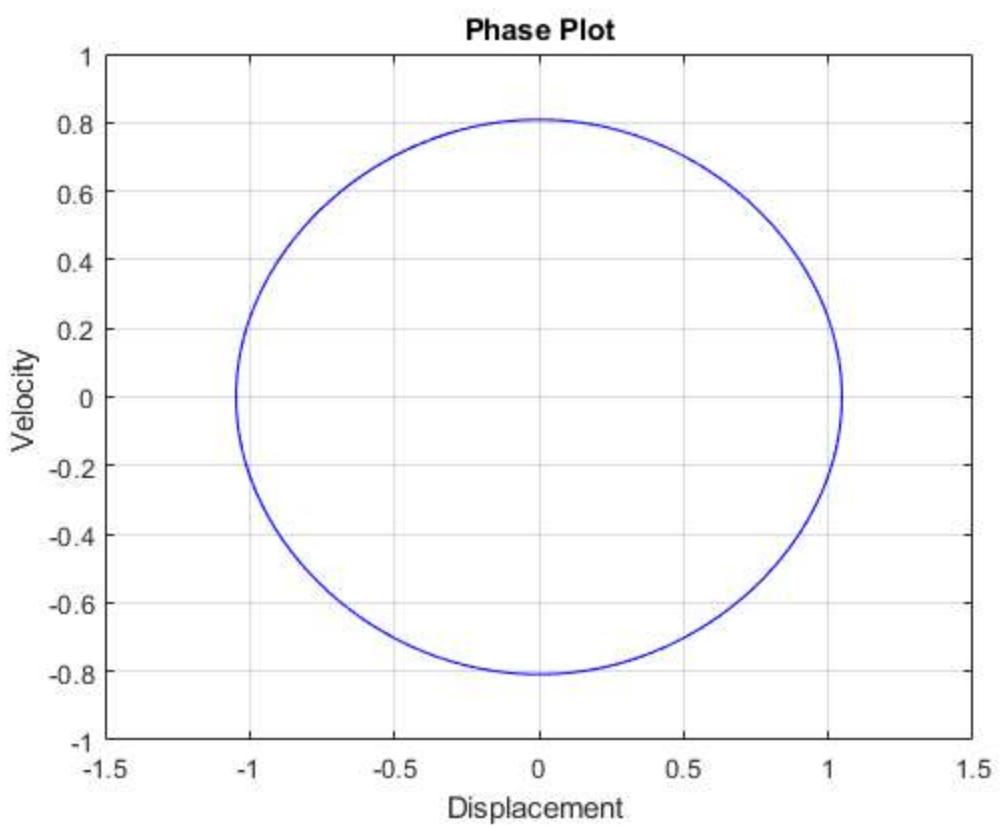
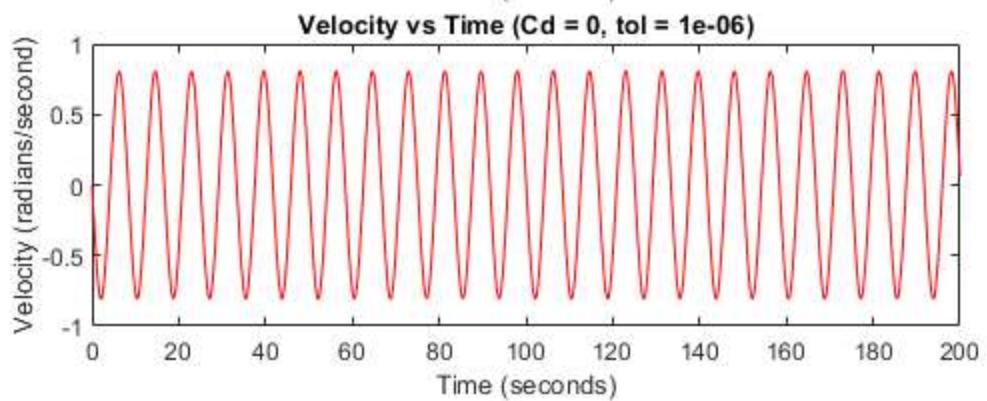
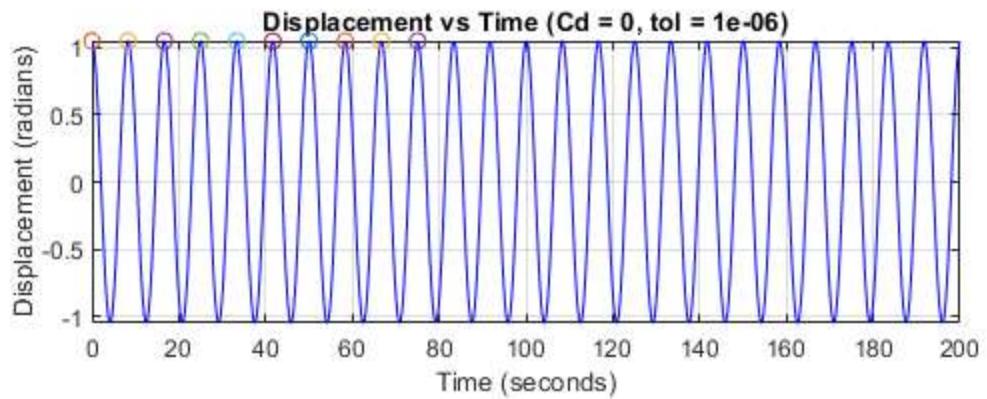
Solutions implementing Cd = 0 and tol = 0.001  
 Displacement: -0.97247026 at t = 12.00 s  
 Velocity: -0.27648093 at t = 12.00 s

Solutions implementing Cd = 0 and tol = 1e-06  
 Displacement: -0.97461724 at t = 12.00 s  
 Velocity: -0.28297895 at t = 12.00 s

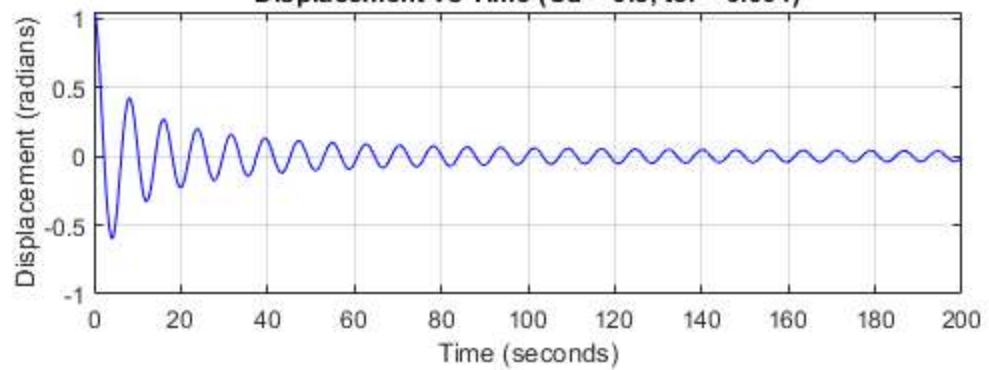
Solutions implementing Cd = 0.5 and tol = 0.001  
 Displacement: -0.28921804 at t = 11.50 s  
 Velocity: -0.12735675 at t = 11.50 s

Solutions implementing Cd = 0.5 and tol = 1e-06  
 Displacement: -0.28863729 at t = 11.50 s  
 Velocity: -0.12858239 at t = 11.50 s

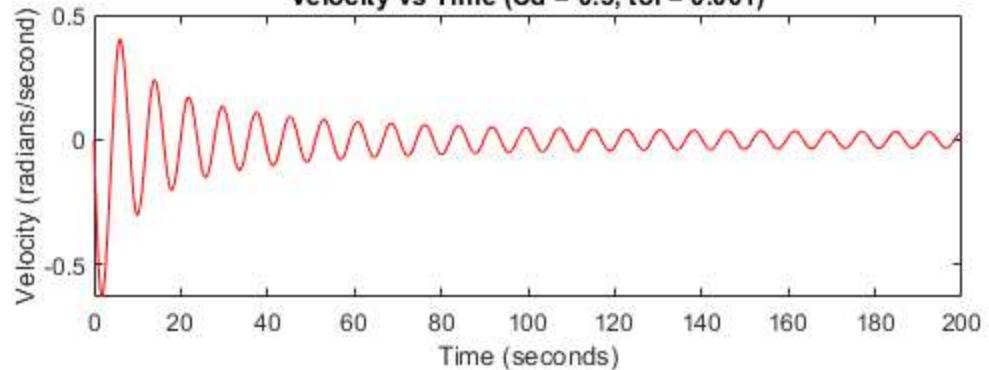




**Displacement vs Time ( $C_d = 0.5$ , tol = 0.001)**



**Velocity vs Time ( $C_d = 0.5$ , tol = 0.001)**



**Phase Plot**

