

Pendulum Coding Problem

Johnathan Corbin
Spaceflight Mechanics - Spring 2019

February 2, 2019

Contents

1	Formula Derivation	3
1.1	Analytic Approach	6
1.1.1	75 Degree Displacement	6
1.1.2	170 Degree Displacement	7
1.2	Numerical Approach	7
1.2.1	Without Aerodynamic Drag	7
1.2.2	With Aerodynamic Drag	7
2	Results	8
3	Matlab Code	11
3.1	Main	11
3.1.1	75 Degree Displacement - Numerical with Drag	11
3.1.2	75 Degree Displacement - Numerical without Drag	11
3.1.3	75 Degree Displacement - Analytical	11
3.1.4	170 Degree Displacement - Numerical with Drag	12
3.1.5	170 Degree Displacement - Numerical without Drag	12
3.1.6	170 Degree Displacement - Analytical	12
3.2	Functions	13
3.2.1	Numerical - No Drag	13
3.2.2	Numerical - Drag	13

1 Formula Derivation

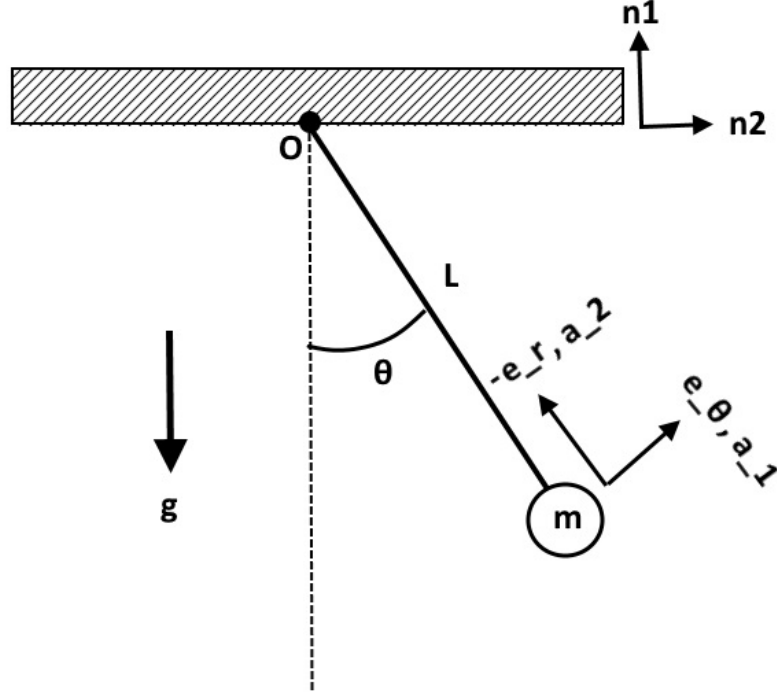


Figure 1: Simple Planar Pendulum System

Figure 1 is a diagram of a simple planar pendulum system. The pendulum swings in the plane of \hat{n}_1 and \hat{n}_2 . In this case, we will consider the rod to have no mass and as having a negligible width, so as not to experience any drag forces. In addition, the rod is of a constant length $L = 2.0\text{meters}$ from point O . The sphere has a mass $m = 0.04\text{kg}$ and a radius of $r = 0.05\text{meters}$.

The reference frame \hat{n}_1 and \hat{n}_2 is taken to be adequately inertial for the purposes of this problem. The entire system is acted upon by standard gravity g that acts in the direction shown in figure 1. The room has an atmosphere $\rho = 1.2250 \frac{\text{kg}}{\text{m}^3}$ and the mass has a coefficient of drag $c_d = 0.42$ that will be used in the calculation of the drag force acting on the mass.

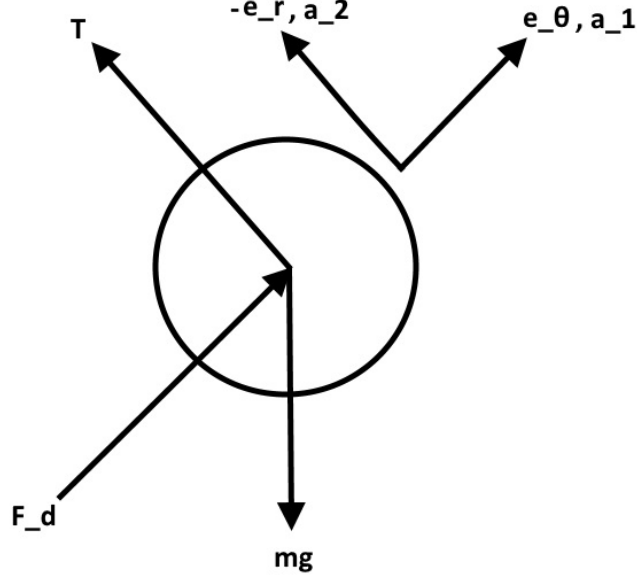


Figure 2: Free Body Diagram of Mass m

Analysis of this problem begins with the free body diagram shown in figure 2. Newton's second law provides a good starting point and is given by:

$$\sum \vec{F}^m = m\vec{a} \quad (1)$$

Using the free body diagram, it can be shown that for this system, Newton's second law becomes the following:

$$\sum \vec{F}^m = m\vec{a} = T\hat{a}_2 - mg\hat{n}_2 + F_d\hat{a}_1 \quad (2)$$

Inspection of figure 1 shows that the motion of the mass is restricted to rotating about point O at a constant radius of L . If we assume that point O is fixed and in an inertial frame, then the acceleration \vec{a} of m is given by the following equation:

$$\vec{a} = (\ddot{L} - L\dot{\theta}^2)\hat{e}_r + (L\ddot{\theta} + 2\dot{L}\dot{\theta})\hat{e}_\theta \quad (3)$$

In our case, L is a constant value. As such, $\dot{L} = 0$ and $\ddot{L} = 0$. Additionally, we need to remember that $\hat{a}_2 = -\hat{e}_r$ and $\hat{a}_1 = \hat{e}_\theta$. Substituting these into equation (3) results in the following definition for the forces on mass m :

$$\sum \vec{F}^m = m\vec{a} = T\hat{a}_2 - mg\hat{n}_2 + F_d\hat{a}_1 = m(L\ddot{\theta}\hat{a}_1 + L\dot{\theta}^2\hat{a}_2) \quad (4)$$

We now have a single vector equation to describe the acceleration of the mass. From this vector equation, we can determine two scalar equations, one for \hat{a}_1 and another for \hat{a}_2 . First, it is necessary for us to define the value for the drag force, F_d , and to also convert the \hat{n}_2 basis in our motion equation into the \hat{a}_1 and \hat{a}_2 basis.

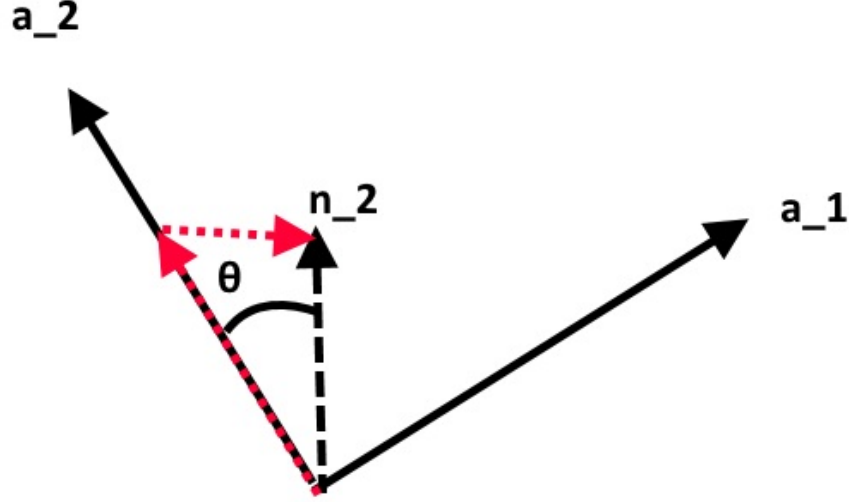


Figure 3: Basis Transform for \hat{n}_2

Using figure 3 as a guide, one can determine that $\hat{n}_2 = \sin(\theta)\hat{a}_1 + \cos(\theta)\hat{a}_2$. Additionally, $F_d = \frac{\rho\dot{\theta}^2 c_d A (-\text{sign}(\theta))}{2}$ where A is the frontal area of the mass and the $(-\text{sign}(\theta))$ function is essential to ensuring that the drag force acts to always oppose the direction of velocity, even if the velocity changes signs. Without this term, the drag force would always act in the same \hat{a}_1 direction, causing the velocity to increase for half the time and subsequently slow the mass down for the other half. In reality, the drag is always working to slow the mass. Once these two additions are taken into account, the equation of motion can be split into the following two scalar equations:

$$\hat{a}_1 : mL\ddot{\theta} = F_d - mg\sin(\theta) \quad (5)$$

$$\hat{a}_2 : mL\dot{\theta}^2 = T - mg\cos(\theta) \quad (6)$$

For this problem, we are interested in the motion of the mass, not the tension force in the rod. As such, equation (6) can be safely discarded. Equation (5) is more useful for our analysis and will be used for the rest of the derivations to determine the motion of the mass. Substituting the previous equation for F_d

into equation (5) yields the following:

$$\ddot{\theta} + \frac{\text{sign}(\dot{\theta})\rho c_d L \dot{\theta}^2 A}{2m} + \frac{g \sin(\theta)}{L} = 0 \quad (7)$$

Equation (7) is our full nonlinear equation of motion for the mass. To solve this equation analytically, several assumptions need to be made to simplify the system. In our case, we will assume that $c_d = 0$ in order to remove the drag force term and also that at small angles, $\sin(\theta) \approx \theta$. After these approximations, our system becomes the following:

$$\ddot{\theta} + \frac{g\theta}{L} = 0 \quad (8)$$

To analytically solve this system, we start by finding the characteristic equation and then finding the roots r_1 and r_2 . The characteristic equation is given by the following:

$$r^2 + \frac{g}{L} = 0 \quad (9)$$

From the characteristic equation, it can be found that there are two complex roots: $r_1 = \sqrt{\frac{g}{L}}i$ and $r_2 = -\sqrt{\frac{g}{L}}i$. If the complex roots of the characteristic equation are of the form $r_{1,2} = \lambda \pm \mu i$, then the general solution to the differential equation is as follows:

$$\theta(t) = c_1 e^{\lambda t} \cos(\mu t) + c_2 e^{\lambda t} \sin(\mu t) \quad (10)$$

In our case, $\mu = \sqrt{\frac{g}{L}}$ and $\lambda = 0$. The unknowns c_1 and c_2 will be determined from the given initial conditions for the system. The problem specifies two separate sets of initial conditions for us to consider. The first set of initial conditions is $[\theta_0, \dot{\theta}_0] = [75deg, 0]$ and the second set is $[\theta_0, \dot{\theta}_0] = [170deg, 0]$. Prior to finding the unknown constants, our general solution is of the form:

$$\theta(t) = c_1 \cos(\sqrt{\frac{g}{L}}t) + c_2 \sin(\sqrt{\frac{g}{L}}t) \quad (11)$$

1.1 Analytic Approach

1.1.1 75 Degree Displacement

At $t = 0$, we have the following initial conditions: $[\theta_0, \dot{\theta}_0] = [75deg, 0]$. Substituting these values into equation (11) and the derivative of equation (11) results in two equations with two unknowns. From this system of equations one can determine that $c_1 = 75deg = \frac{5\pi}{12}rad$ and $c_2 = 0$. This results in the following analytic solution for an initial displacement of 75 degrees:

$$\theta(t) = \frac{5\pi}{12} \cos(\sqrt{\frac{g}{L}}t) \quad (12)$$

1.1.2 170 Degree Displacement

At $t = 0$, we have the following initial conditions: $[\theta_0, \dot{\theta}_0] = [170deg, 0]$. Substituting these values into equation (11) and the derivative of equation (11) results in two equations with two unknowns. From this system of equations one can determine that $c_1 = 170deg = \frac{17\pi}{18}rad$ and $c_2 = 0$. This results in the following analytic solution for an initial displacement of 170 degrees:

$$\theta(t) = \frac{17\pi}{18} \cos(\sqrt{\frac{g}{L}}t) \quad (13)$$

1.2 Numerical Approach

From figures 4 and 5 it is apparent that the analytic results previously acquired were only accurate within a limited range. To improve our results, we need to adjust the simplifications that were made to the system. Previously we had assumed that $c_d = 0$ and $\sin(\theta) \approx \theta$. If we now consider the effects of drag and disregard the small angle approximation, we will have a more accurate model of our system, but it becomes significantly harder to find an analytic answer for. The next two sections will look at how removing these two assumptions and solving the systems numerically can improve the accuracy of our results.

To numerically solve the new equations of motion, the `ode45` function in Matlab will be utilized. All code used in this report can be found in the Matlab Code portion (section 3).

1.2.1 Without Aerodynamic Drag

In this section, the effects of the small angle approximation are going to be examined. We will still assume that $c_d = 0$ for now. By removing the small angle approximation, our new equation of motion is as follows:

$$\ddot{\theta} + \frac{g \sin(\theta)}{L} = 0 \quad (14)$$

Section 3.1.2 contains the code used to analyze equation (14) for an initial displacement of 75 degrees. Section 3.1.5 contains the code used to analyze equation (14) for an initial displacement of 170 degrees. In addition, both methods utilized the function in section 3.2.1 to supply `ode45` with the proper time derivatives. The results for both can be seen in figures 4 and 5.

1.2.2 With Aerodynamic Drag

In section 1.2.1 we examined how the small angle approximation affects the accuracy of our results. Now, let's examine how the addition of a drag force can get us closer to an accurate model of the system. As previously stated, we are going to assume that $c_d = 0.42$ for the mass. After accounting for drag, our

new equation of motion is as follows:

$$\ddot{\theta} + \frac{\text{sign}(\dot{\theta})\rho c_d L \dot{\theta}^2 A}{2m} + \frac{g \sin(\theta)}{L} = 0 \quad (15)$$

For equation (15) it is important to note the $\text{sign}(\dot{\theta})$ function in the drag term. Without this function, the drag will not oppose the velocity of the mass at all times. Matlab has a built-in function for $\text{sign}()$. The $\text{sign}(\dot{\theta})$ function can be described as working as follows:

$$\text{sign}(\dot{\theta}) = \begin{cases} -1, & \dot{\theta} < 0 \\ 0, & \dot{\theta} = 0 \\ 1, & \dot{\theta} > 0 \end{cases} \quad (16)$$

Section 3.1.1 contains the code used to analyze equation (15) for an initial displacement of 75 degrees. Section 3.1.4 contains the code used to analyze equation (15) for an initial displacement of 170 degrees. In addition, both methods utilized the function in section 3.2.2 to supply *ode45* with the proper time derivatives. The results for both can be seen in figures 4 and 5.

2 Results

Figures 4 and 5 are graphs showing the results from all three different methods of analysis. The legend accompanying each graph distinguishes which results belong to the analytic result, numerical without drag, and numerical with drag.

Upon inspection, it is apparent that early in the time scale at 75 degrees, all three methods give very similar answers that start to lose accuracy around the 2 second mark. However, for 170 degrees, the analytic result gives a very poor estimate of the position of the mass. This result is expected and is a consequence of the small angle approximation we made in order to simplify the system. The larger the angle that we input into our analytic response, the worse the small angle approximation we made becomes. This is why we see such a poor approximation of the position for the large initial angle of 170 degrees in the analytic result.

In addition to the error resulting from the small angle approximation, we can see that in the numerical result that accounts for drag is slowly dampened over time. The drag force that opposes the velocity of the mass is what is responsible for slowing the motion of the mass. In the analytic result, we neglected drag altogether, which is why the system appears to oscillate forever without slowing down. The same is true of the numerical result we determined for the system with no drag. The only difference between the two is the phase of the oscillations.

In reality, the mass obviously will not oscillate forever. As such, our analytic result and numerical result with no drag are poor approximations of how the system truly behaves. They are only accurate on very small time scales,

when the drag force hasn't had enough time to significantly slow the mass. As such, the analytic response is really only valuable for showing which variables affect the system and how they can be changed to achieve different results from the system. If we wanted to determine the position of the mass after a long position of time, we would instead use the numerical result that incorporated drag, rather than the analytic result.

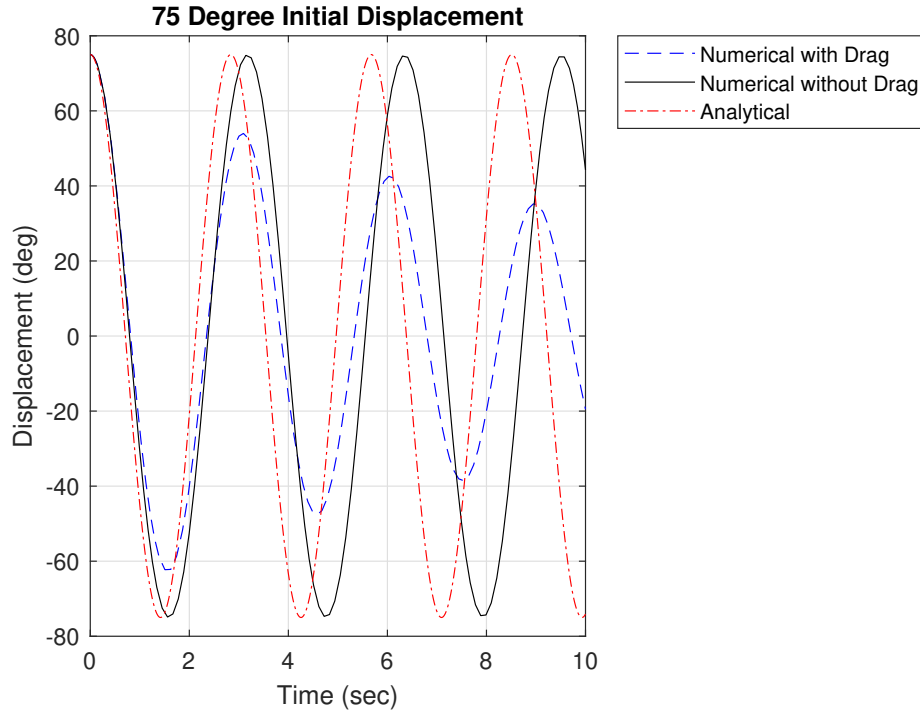


Figure 4: Plots of the Analytic and Numerical Results for 75 Degree Displacement

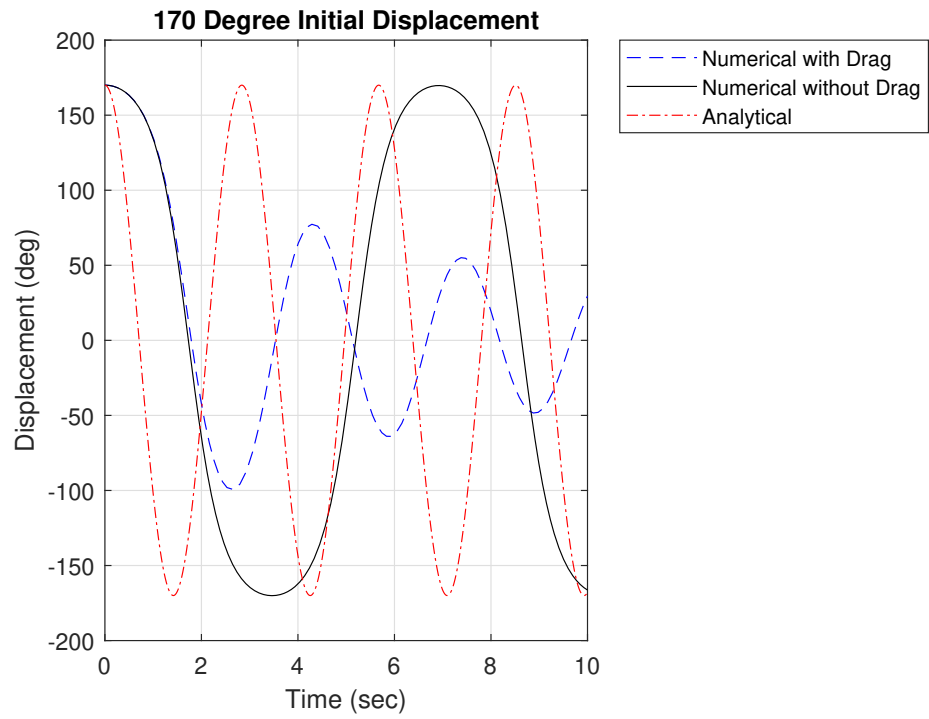


Figure 5: Plots of the Analytic and Numerical Results for 170 Degree Displacement

3 Matlab Code

3.1 Main

```
clear, clc

global L m g rho drag_coefficient interval area

%Declaring constants
L = 2; %length of rod [meters]
m = .04; %mass of sphere [kg]
radius = .05; %radius of sphere [m]
g = 9.81; %acceleration due to gravity [m/s^2]
rho = 1.225; %density of air [kg/m^3]
drag_coefficient = .42; %coefficient of drag
interval = [0,10]; %time interval [s]

area = pi * radius^2; %frontal area of sphere [m^2]

3.1.1 75 Degree Displacement - Numerical with Drag

initial_conditions = [75 * pi / 180; 0]; %initial conditions [deg deg/s]

fname = 'PendulumFuncDrag'; %Setting name of the .m file for the state
%variable time derivatives

[time, y] = ode45(fname, interval, initial_conditions);

figure(1)
plot(time, y(:,1) * 180 / pi, '--b');
ylabel('Displacement (deg)')
xlabel('Time (sec)')
hold on
grid on
```

3.1.2 75 Degree Displacement - Numerical without Drag

```
fname = 'PendulumFunc';

[time, y] = ode45(fname, interval, initial_conditions);

plot(time, y(:,1) * 180 / pi, 'k');
```

3.1.3 75 Degree Displacement - Analytical

```
time = 0:.01:10; %time interval (sec)
theta0 = 75 * pi / 180;
```

```

theta = theta0 * cos(sqrt(g / L) * time);

plot(time, theta * 180 / pi, '-.r')
legend('Numerical with Drag', 'Numerical without Drag', 'Analytical',...
       'Location', 'bestoutside')
title('75 Degree Initial Displacement')

```

3.1.4 170 Degree Displacement - Numerical with Drag

```

initial_conditions = [170 * pi / 180; 0]; %initial conditions [deg deg/s]

fname = 'PendulumFuncDrag'; %Setting name of the .m file for the state
%variable time derivatives

[time, y] = ode45(fname, interval, initial_conditions);

figure(2)
plot(time, y(:,1) * 180 / pi, '--b');
ylabel('Displacement (deg)')
xlabel('Time (sec)')
hold on
grid on

```

3.1.5 170 Degree Displacement - Numerical without Drag

```

fname = 'PendulumFunc';

[time, y] = ode45(fname, interval, initial_conditions);

plot(time, y(:,1) * 180 / pi, 'k');

```

3.1.6 170 Degree Displacement - Analytical

```

time = 0:.01:10; %time interval (sec)
theta0 = 170 * pi / 180;

theta = theta0 * cos(sqrt(g / L) * time);

plot(time, theta * 180 / pi, '-.r')
legend('Numerical with Drag', 'Numerical without Drag', 'Analytical',...
       'Location', 'bestoutside')
title('170 Degree Initial Displacement')

```

3.2 Functions

3.2.1 Numerical - No Drag

```
function dydt = PendulumFunc(t, y)
% Program determine the first order time derivatives of variable vector y = [y1,y2] for
% the current values of y and time t.
global L g
% can be easily used by the the dydt script

dydt = zeros(size(y)); % initialize to zero and makes column vector

dydt(1) = y(2);

dydt(2) = -(g * sin(y(1))) / L;
```

3.2.2 Numerical - Drag

```
function dydt = PendulumFuncDrag(t, y)
% Program determine the first order time derivatives of variable vector y = [y1,y2] for
% the current values of y and time t.
global L m g rho drag_coefficient area
% can be easily used by the the dydt script

dydt = zeros(size(y)); % initialize to zero and makes column vector

dydt(1) = y(2);

dydt(2) = (-sign(y(2)) * rho * L * drag_coefficient * y(2)^2 * area) / (2 * m) - (g * sin(y(1)))
```