The Double Pendulum

Andrew Rippy

Department of Physics, Wabash College, Crawfordsville, IN 47933

December 4th, 2020

Abstract

This write-up will focus on the dynamic system of the double pendulum. Specifically, the non-ideal double pendulum, which includes the moments of inertia of both arms, but does not include friction. The double pendulum is an oscillating mechanical system that is considered to be a chaotic system. The behavior of the system is quite complicated, but can be explained in reasonably simplistic terms through the equations of motion for both arms. There does not exist any analytical solutions to the double pendulum system, so we must rely on numerical solutions to model the motion of the system, using the derived equations of motion. As such, a simulation for the double pendulum was created to exemplify the motion of the double pendulum in a visual manner. In addition to the simulation, a real-life version of the double pendulum was created in order to compare its observed motion against the simulated predicted motion.

1 Derivation of the Equations of Motion

The Lagrangian Approach

First and foremost, let us define our variables:

- L_1 is the length from pivot 1 to pivot 2 (The length of arm 1)
- L_2 is the length of arm 2 of the pendulum
- R_1 is the length from pivot 1 to the center of mass of arm 1
- R_2 is the length from pivot 2 to the center of mass of arm 2
- θ_1 is the angle relative to the vertical of arm 1
- θ_2 is the angle relative to the vertical of arm 2
- x_1, x_2, y_1, y_2 are the x and y coordinates of the respective center of masses of the arms
- m_1,m_2 are the respective masses of each arm

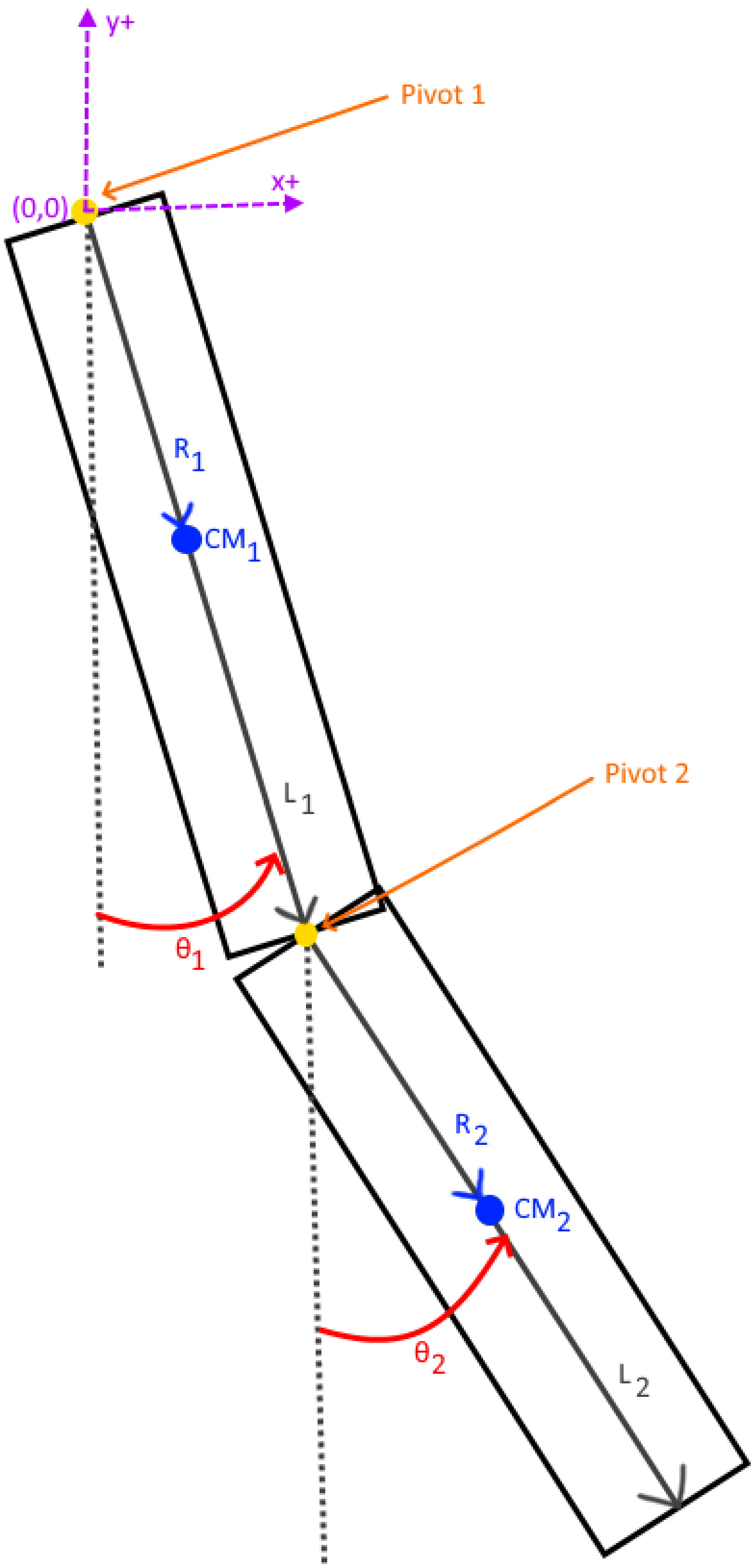


Figure 1: A Diagram of the Double Pendulum

The Lagrangian is given as follows:

$$\mathscr{L} = T - U \tag{1}$$

Where T and U are the kinetic and potential energy of the pendulum respectively. T is composed of the following components; The linear kinetic energy of the center of mass of arm 1, $\frac{1}{2}m_1(\dot{x_1}^2+\dot{y_1}^2)$, the rotational kinetic energy of arm 1, $\frac{1}{2}I_1\dot{\theta_1}^2$, the linear kinetic energy of the center of mass of arm 2, $\frac{1}{2}m_2(\dot{x_2}^2+\dot{y_2}^2)$, and the rotational kinetic energy of arm 2, $\frac{1}{2}I_2\dot{\theta_2}^2$. U is composed of the following components; The gravitational potential energy of the center of mass of arm 1, m_1gy_1, and the gravitational potential energy of the center of mass of arm 2, m_2gy_2. Putting all of these components together, our Lagrangian is as follows:

$$\mathscr{L} = \frac{1}{2}\Big(m_1(\dot{x_1}^2+\dot{y_1}^2) + I_1\dot{\theta_1}^2 + m_2(\dot{x_2}^2+\dot{y_2}^2) + I_2\dot{\theta_2}^2\Big) - m_1gy_1 - m_2gy_2 \tag{2}$$

Next, we need to get the Lagrangian entirely in terms of θ. We can derive the following relations from the diagram:

$$\begin{aligned}
x_1 &= R_1\sin(\theta_1)\\
y_1 &= -R_1\cos(\theta_1)\\
x_2 &= L_1\sin(\theta_1) + R_2\sin(\theta_2)\\
y_2 &= -L_1\cos(\theta_1) - R_2\cos(\theta_2)
\end{aligned}$$

Taking the first derivatives with respect to time, gives us the following equations:

$$\begin{aligned}
\dot{x_1} &= \dot{\theta}_1R_1\cos(\theta_1)\\
\dot{y_1} &= \dot{\theta}_1R_1\sin(\theta_1)\\
\dot{x_2} &= \dot{\theta}_1L_1\cos(\theta_1) + \dot{\theta}_2R_2\cos(\theta_2)\\
\dot{y_2} &= \dot{\theta}_1L_1\sin(\theta_1) + \dot{\theta}_2R_2\sin(\theta_2)
\end{aligned}$$

Now we use these to rewrite the Lagrangian in terms of theta:

$$\begin{aligned}
\mathscr{L} = \frac{1}{2}\Big(m_1((\dot{\theta}_1R_1\cos(\theta_1))^2 + (\dot{\theta}_1R_1\sin(\theta_1))^2) + I_1\dot{\theta_1}^2 +\\
m_2((\dot{\theta}_1L_1\cos(\theta_1) + \dot{\theta}_2R_2\cos(\theta_2))^2 + (\dot{\theta}_1L_1\sin(\theta_1) + \dot{\theta}_2R_2\sin(\theta_2))^2) + I_2\dot{\theta_2}^2\Big) +\\
m_1gR_1(\cos(\theta_1)) + m_2g(L_1\cos(\theta_1) + R_2\cos(\theta_2))
\end{aligned}$$

Expanding out,

$$\begin{aligned}
\mathscr{L} = \frac{1}{2}\Big(m_1\dot{\theta_1}^2R_1^2\cos^2(\theta_1) + m_1\dot{\theta_1}^2R_1^2\sin^2(\theta_1) + I_1\dot{\theta_1}^2 +\\
m_2\dot{\theta_1}^2L_1^2\cos^2(\theta_1) + m_2\dot{\theta_2}^2R_2^2\cos^2(\theta_2) + 2m_2L_1R_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1)\cos(\theta_2) +\\
m_2\dot{\theta_1}^2L_1^2\sin^2(\theta_1) + m_2\dot{\theta_2}^2R_2^2\sin^2(\theta_2) + 2m_2\dot{\theta}_1\dot{\theta}_2L_1R_2)\sin(\theta_1)\sin(\theta_2) + I_2\dot{\theta_2}^2\Big) +\\
m_1g(R_1\cos(\theta_1)) + m_2g(L_1\cos(\theta_1) + R_2\cos(\theta_2))
\end{aligned}$$

Now, grouping like terms and factoring, we have:

$$\mathscr{L} = \frac{1}{2}\Bigg(\dot{\theta}_1^{\,2}\Big(m_1R_1^2\cos^2(\theta_1) + m_1R_1^2\sin^2(\theta_1) + I_1 + m_2L_1^2\cos^2(\theta_1) + m_2L_1^2\sin^2(\theta_1)\Big) +$$
$$\dot{\theta}_2^{\,2}\Big(m_2R_2^2\cos^2(\theta_2) + m_2R_2^2\sin^2(\theta_2) + I_2\Big) +$$
$$2m_2L_1R_2\dot{\theta}_1\dot{\theta}_2\Big(\cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2)\Big)\Bigg) +$$
$$\Big(m_1gR_1 + m_2gL_1\Big)\cos(\theta_1) + \Big(m_2gR_2\Big)\cos(\theta_2)$$

Next, using trig identities $\sin^2(\alpha)+\cos^2(\alpha) = 1$ and $\cos(\alpha)\cos(\beta)+\sin(\alpha)\sin(\beta) = \cos(\alpha-\beta)$, we simplify further:

$$\mathscr{L} = \frac{1}{2}\Bigg(\dot{\theta}_1^{\,2}\Big(m_1R_1^2 + I_1 + m_2L_1^2\Big) + \dot{\theta}_2^{\,2}\Big(m_2R_2^2 + I_2\Big)\Bigg) + \dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2)\Big(m_2L_1R_2\Big) +$$
$$\Big(m_1gR_1 + m_2gL_1\Big)\cos(\theta_1) + \Big(m_2gR_2\Big)\cos(\theta_2)$$

Finally, for the sake of compact notation, let us define the following constants:

$$A = m_1R_1^2 + I_1 + m_2L_1^2 \tag{3}$$
$$B = m_2R_2^2 + I_2 \tag{4}$$
$$C = m_2L_1R_2 \tag{5}$$
$$D = m_1gR_1 + m_2gL_1 \tag{6}$$
$$E = m_2gR_2 \tag{7}$$

Finally, we can rewrite our Lagrangian into a compact form:

$$\mathscr{L} = \frac{1}{2}\Big(A\dot{\theta}_1^{\,2} + B\dot{\theta}_2^{\,2}\Big) + C\dot{\theta}_1\dot{\theta}_2\cos(\theta_1-\theta_2) + D\cos(\theta_1) + E\cos(\theta_2) \tag{8}$$

To find the two equations of motion, we must take the following derivatives:

$$\frac{d}{dt}\frac{\partial\mathscr{L}}{\partial\dot{q}} = \frac{\partial\mathscr{L}}{\partial q} \tag{9}$$

Eq. of motion for θ_1

$$\frac{\partial\mathscr{L}}{\partial\dot{\theta}_1} = A\dot{\theta}_1 + C\dot{\theta}_2\cos(\theta_1-\theta_2)$$
$$\frac{d}{dt}\frac{\partial\mathscr{L}}{\partial\dot{\theta}_1} = A\ddot{\theta}_1 + C\ddot{\theta}_2\cos(\theta_1-\theta_2) - C(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_2\sin(\theta_1-\theta_2)$$
$$\frac{\partial\mathscr{L}}{\partial\theta_1} = -C\dot{\theta}_1\dot{\theta}_2\sin(\theta_1-\theta_2) - D\sin(\theta_1)$$
$$A\ddot{\theta}_1 + C\ddot{\theta}_2\cos(\theta_1-\theta_2) - C(\dot{\theta}_1-\dot{\theta}_2)\dot{\theta}_2\sin(\theta_1-\theta_2) = -C\dot{\theta}_1\dot{\theta}_2\sin(\theta_1-\theta_2) - D\sin(\theta_1)$$

Rearranging and simplifying we solve for $\ddot{\theta}_1$

$$\ddot{\theta}_1 = \frac{1}{A}\Big(-C\ddot{\theta}_2\cos(\theta_1-\theta_2) - C\dot{\theta}_2^{\,2}\sin(\theta_1-\theta_2) - D\sin(\theta_1)\Big) \tag{10}$$

Eq. of motion for θ_2

$$\frac{\partial \mathscr{L}}{\partial \dot{\theta}_2} = B\dot{\theta}_2 + C\dot{\theta}_1 \cos(\theta_1 - \theta_2)$$

$$\frac{d}{dt}\frac{\partial \mathscr{L}}{\partial \dot{\theta}_2} = B\ddot{\theta}_2 + C\ddot{\theta}_1 \cos(\theta_1 - \theta_2) - C(\dot{\theta}_1 - \dot{\theta}_2)\dot{\theta}_1 \sin(\theta_1 - \theta_2)$$

$$\frac{\partial \mathscr{L}}{\partial \theta_2} = C\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) - E\sin(\theta_2)$$

$$B\ddot{\theta}_2 + C\ddot{\theta}_1 \cos(\theta_1 - \theta_2) - C(\dot{\theta}_1 - \dot{\theta}_2)\dot{\theta}_1 \sin(\theta_1 - \theta_2) = C\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) - E\sin(\theta_2)$$

Rearranging and simplifying we solve for $\ddot{\theta}_2$

$$\ddot{\theta}_2 = \frac{1}{B}\Big(- C\ddot{\theta}_1 \cos(\theta_1 - \theta_2) + C\sin(\theta_1 - \theta_2)\dot{\theta}_1{}^2 - E\sin(\theta_2)\Big) \tag{11}$$

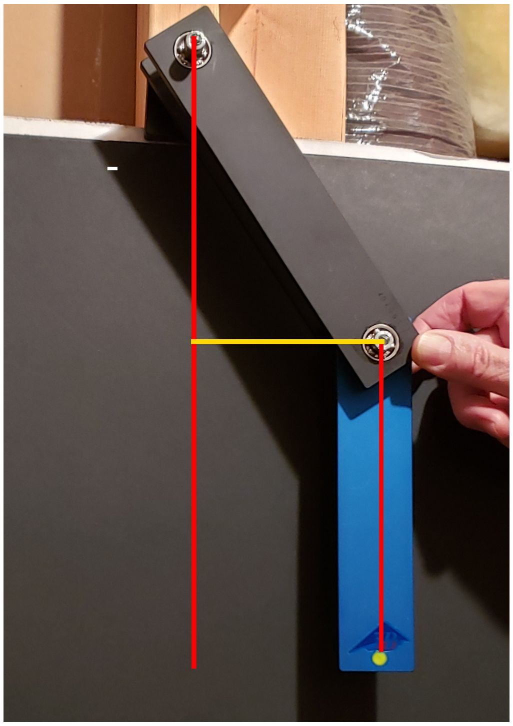
2 The Simulation and Real Life Pendulum

For the simulation, the equations of motion were integrated using numerical integration in Python via the Runge Kutta 4 method, and in Mathematica with Mathematica's NDsolve. The Python simulation file *DP.py* and the Mathematica notebook *doublependulum.nb* are included with this write-up showing an interactive, animated simulation. There is however, a downfall of numerical integration with respect to this mechanical system. According to a study[1] done on numerical integration in the context of mechanical systems, it is found that though RK4 and many other numerical integration techniques are stable in terms of progressing the system, they are not stable in terms of conserving energy. This causes more problems as the error in energy continues to compound the longer the simulation runs. The measurements are as follows:

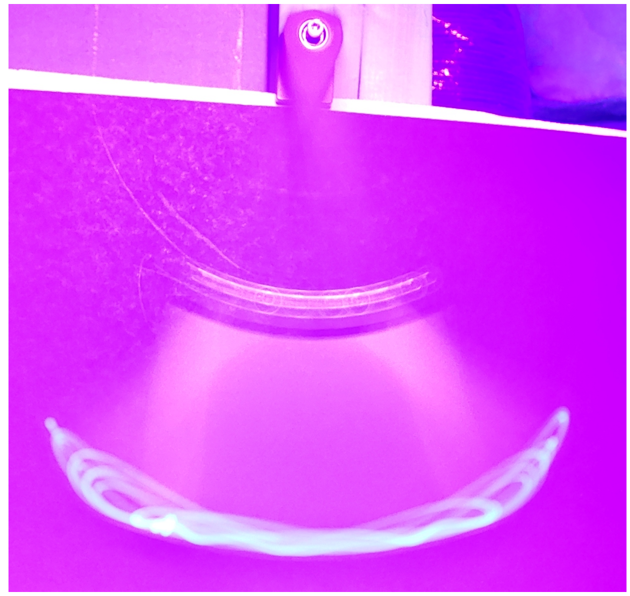
- $L_1 = 0.178$ m $\pm$ 0.0048 m 95 % CI
- $L_2 = 0.165$ m $\pm$ 0.0048 m 95 % CI
- $R_1 = 0.089$ m $\pm$ 0.0048 m 95 % CI
- $R_2 = 0.083$ m $\pm$ 0.0048 m 95 % CI
- $m_1 = 0.244$ kg $\pm$ 0.00048 kg 95 % CI
- $m_2 = 0.107$ kg $\pm$ 0.00048 kg 95 % CI
- $I = \frac{1}{12}m(L^2 + W^2)$ where L and W are the length and width respectively of the plate
- $W = 0.0381$ m $\pm$ 0.0048 m 95 % CI (Both plates were the same width)

Several trials with their real life motion along with their predicted simulated motion are shown below.

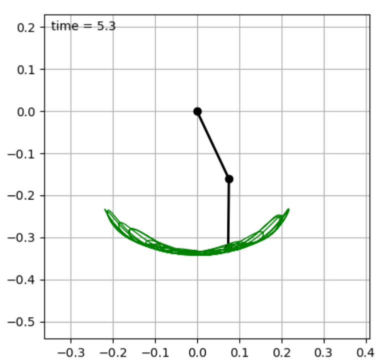
[1]https://arxiv.org/pdf/1909.13215.pdf



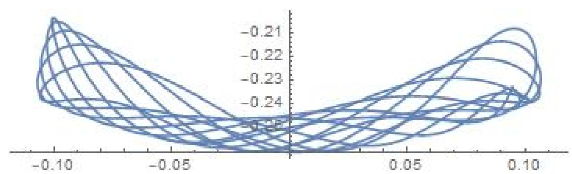
Initial Condition



Long Exposure of Path

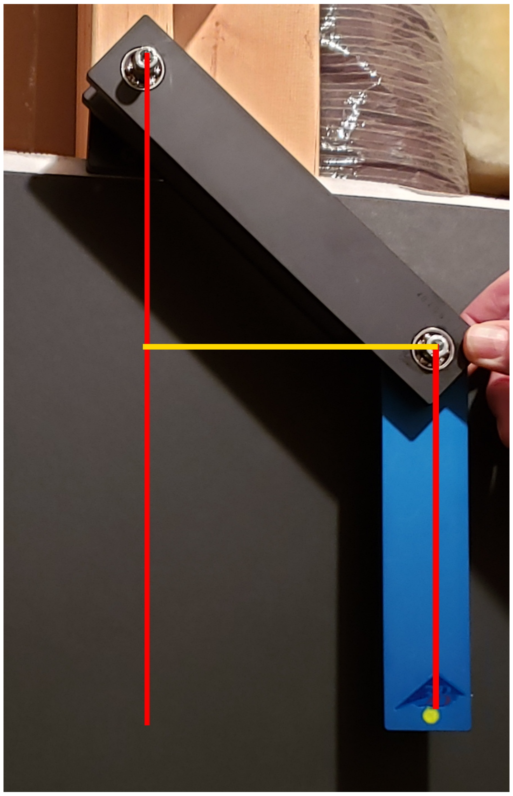


Python Simulation of Path

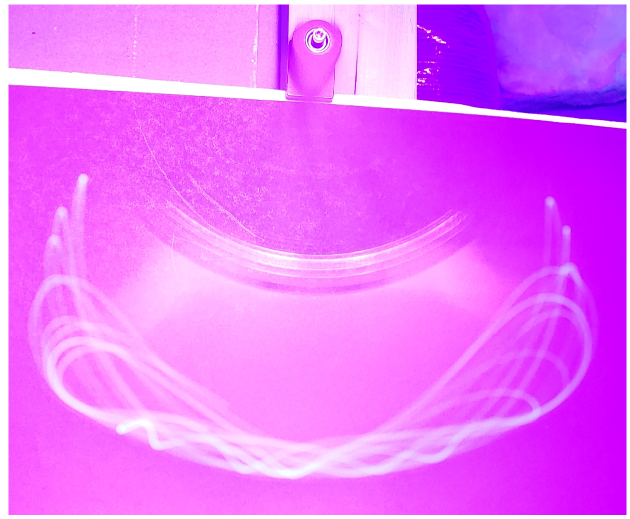


Mathematica Simulation of Path

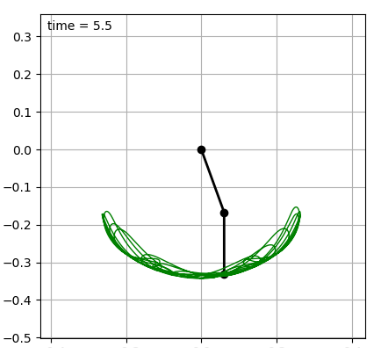
Figure 2: $\theta_1 = 32.2^\circ \pm 0.48^\circ 95$ % CI $\theta_2 = 0$



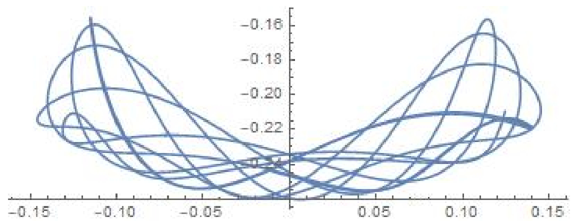
Initial Condition



Long Exposure of Path

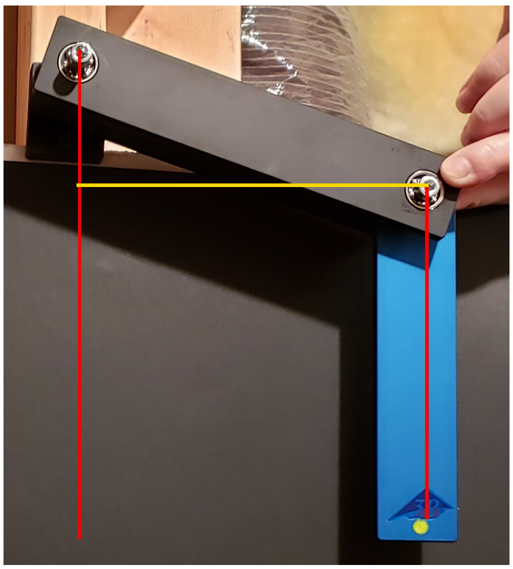


Python Simulation of Path



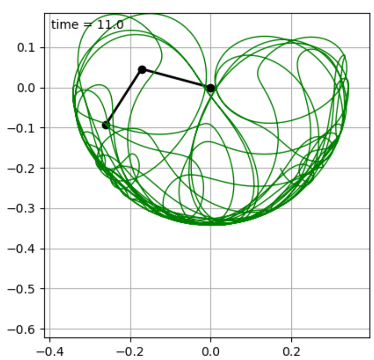
Mathematica Simulation of Path

Figure 3: $\theta_1 = 44.4° \pm 0.48°$ 95% CI $\theta_2 = 0$

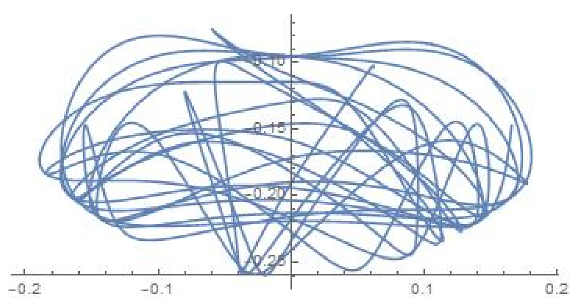


Initial Condition

Long Exposure Path



Python Simulation of Path



Mathematica Simulation of Path

Figure 4: $\theta_1 = 68.6° \pm 0.48°$ 95% CI $\theta_2 = 0$

3 The Process

In order to create the simulation in Python, I took advantage of the built in animator. Essentially what this does is it calls an animate(i) function at every ith frame, where it updates the pendulum with a step() function, and then redraws the new graph frame for each. It computes the steps in real time based on a chosen dt step, rather than pre-computing and saving as a video file or image. The pendulum object contained all the relevant measurements needed and the step() function used these values along with the derived equations of motion with a Runge Kutta 4 numerical integration method to compute each subsequent step. The Mathematica simulation used Mathematica's NDSolve to pre-compute a list of position values, then graph them onto a single frame. In order to capture the real-life motion of the pendulum, I used florescent paint along with an ultra-violet black light in a long exposure shot to capture the movement of the point.

4 Closing Remarks

Overall, the simulations worked well given the problems associated with numerical integration. For smaller angles, the real motion of the pendulum closely matched the expected motion from the simulations. However, once the angles became large enough, evidenced in Fig. 4, the simulations as well as the real motion deviated from one another. This is likely due to the real system having friction, the compounding error of the numerical integration, and the different integration methods employed by each simulation. It is also likely due to the uncertainty of the angle measurement, as since this is a chaotic system, it is *incredibly* sensitive to the initial conditions, and even a small fraction of an angle difference could mean vastly different motion outcomes.

5 Acknowledgements

I'd like to acknowledge the help and support of the Wabash Physics Department. I'd also like to acknowledge Dr. Brown for helping me with the equations of motion.