Rochester Institute of Technology
College of Science
School of Physics and Astronomy

Chaotic Behavior of Double Pendulum Systems

Student: *Ramsey (Rayla) Phuc*

Course: *Advanced Computational Physics (PHYS-377.01)* Term: *2020-2021 Spring*
Professor: *Dr. Pratik Dholabhai* Due date: *May 04, 2021*

Abstract

The double pendulum is a classic example of a chaotic system that exhibits complex and unpredictable behavior. In this paper, we investigate the dynamics of the double pendulum and analyze its chaotic characteristics. We begin by formulating the equations of motion using Lagrangian mechanics, considering the gravitational forces and constraints on the system. Through numerical simulations and computational techniques, we explore the evolution of the double pendulum system under various initial conditions. Our analysis reveals that the double pendulum demonstrates sensitive dependence on initial conditions, commonly referred to as the "butterfly effect." Small variations in the initial conditions lead to significant deviations in the system's subsequent motion, making long-term predictions challenging.

1 Introduction and Background

1.1 Introduction

A dynamical system, in physics, typically describes the state of a particle that varies over time. One particular example of a dynamical system in physics is the double pendulum. A single pendulum consists of one object (a bob) that is connected to a fixed point by an non-stretchable string. A double pendulum is an extension of the single pendulum where you attach another bob at the end of the single pendulum. Unlike a single pendulum, the double pendulum has a strong sensitivity to its initial conditions, which causes the double pendulum to behave in a chaotic way.

1.2 Goals

1. Derive the potential, kinetic, and total energies of a generic double pendulum system.
2. Derive the system of equations that represents the motion for a generic double pendulum system.
3. Solve the system of equations numerically using a custom ODE solver. This solver uses the Runge-Kutta Order 4 method.
4. Create an animation of the double pendulum along with some animated graphs. These animated graphs are meant to be in sync with the double pendulum animation.
5. Show that the Double Pendulum system is a chaotic system by showing the differences a small change in a initial condition can do.

1.3 Fundamental Physics

Figure 1 captures a frame of the double pendulum. Here we will set the initial anchor at the origin O. x_i represents the horizontal distance from the origin to bob i, y_i represents the vertical distance from the origin to bob i, and θ_i represents the angle bob i makes between the anchor or the bob above it and the vertical.

We can derive the equations that are necessary to show the behavior of this chaotic system. Lets start by deriving the potential, kinetic, and the total energy of this system. From Figure 1, we can determine the positions of each bob. The x and y coordinates for bob 1 are:

$$\begin{aligned} x_1 &= l_1 \sin(\theta_1) \\ y_1 &= -l_1 \cos(\theta_1) \end{aligned} \tag{1}$$

and the x and y coordinates for bob 2 are:

$$\begin{aligned} x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_2) \\ y_2 &= -l_1 \cos(\theta_1) - l_2 \cos(\theta_2) \end{aligned} \tag{2}$$

Next, we will compute the total potential energy of the system. We know that the potential energy for bob i is $PE_i = m_i g h_i$. We also know that h_i is the vertical component of bob i, or y_i. Using these y components from Equations (1)-(2), we can compute the total potential energy of the system:

$$\begin{aligned} PE &= PE_1 + PE_2 \\ PE &= m_1 g h_1 + m_2 g h_2 \\ PE &= m_1 g y_1 + m_2 g y_2 \\ PE &= m_1 g\left(-l_1 \cos(\theta_1)\right) + m_2 g\left(-l_1 \cos(\theta_1) - l_2 \cos(\theta_2)\right) \end{aligned}$$

or

$$PE = -\left(m_1 + m_2\right) g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2) \tag{3}$$

Next, we will compute the first time derivatives of the equations in Equation 1 to get bob 1's velocity components:

$$\begin{aligned} v_{x_1} &= l_1 \dot{\theta_1} \cos(\theta_1) \\ v_{y_1} &= l_1 \dot{\theta_1} \sin(\theta_1) \end{aligned} \tag{4}$$

Repeating this procedure on Equation 2 for bob 2 yields:

$$\begin{aligned} v_{x_2} &= l_1 \dot{\theta_1} \cos(\theta_1) + l_2 \dot{\theta_2} \cos(\theta_2) \\ v_{y_2} &= l_1 \dot{\theta_1} \sin(\theta_1) + l_2 \dot{\theta_2} \sin(\theta_2) \end{aligned} \tag{5}$$

We know that the velocity of bob i is $v_i^2 = v_{x_i}^2 + v_{y_i}^2$. Using the equations in Equation 4, we can compute the velocity of bob 1:

$$v_1^2 = l_1^2 \dot{\theta_1}^2 \tag{6}$$

And we can follow the same steps to compute the velocity of bob 2:

$$v_2^2 = l_1^2 \dot{\theta_1}^2 + l_2^2 \dot{\theta_2}^2 + 2 l_1 l_2 \dot{\theta_1} \dot{\theta_2} \cos(\theta_1 - \theta_2) \tag{7}$$

With this, we can now compute the total kintetic energy of the system. We know that the kinetic energy for bob i is $KE_i = \left(m_i v_i^2\right)/2$. With Equations (6)-(7), we can say that the total kinetic energy of the system is:

$$\begin{aligned} KE &= KE_1 + KE_2 \\ KE &= \frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2} \\ KE &= \frac{m_1\left(l_1^2 \dot{\theta_1}^2\right)}{2} + \frac{m_2\left(l_1^2 \dot{\theta_1}^2 + l_2^2 \dot{\theta_2}^2 + 2 l_1 l_2 \dot{\theta_1} \dot{\theta_2} \cos(\theta_1 - \theta_2)\right)}{2} \end{aligned}$$

or:

$$KE = \frac{(m_1+m_2)\, l_1^2 \dot{\theta_1}^2 + m_2 l_2^2 \dot{\theta_2}^2}{2} + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \tag{8}$$

With expressions for the total potential energy and the total kinetic energy, we can compute the total energy of the system. The total energy is the sum of all energies. Thus, we have $TE = KE + PE$ or:

$$TE = \frac{(m_1+m_2)\, l_1^2 \dot{\theta_1}^2 + m_2 l_2^2 \dot{\theta_2}^2}{2} + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) - (m_1+m_2)\, g l_1 \cos(\theta_1) - m_2 g l_2 \cos(\theta_2) \tag{9}$$

Now lets derive the equations that capture the motion of the system. To do this, we will need to determine the Lagrangian. The lagrangian of a system is $L = KE - PE$. For this system, the lagrangian is:

$$L = \frac{(m_1+m_2)\, l_1^2 \dot{\theta_1}^2 + m_2 l_2^2 \dot{\theta_2}^2}{2} + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) + (m_1+m_2)\, g l_1 \cos(\theta_1) + m_2 g l_2 \cos(\theta_2) \tag{10}$$

Now we will need to utilize the Euler-Lagrange equation, which states:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0 \tag{11}$$

We will apply the Euler-Lagrange equation to both θ_1 and θ_2. Applying the Euler-Lagrange equation to θ_1, we get:

$$\begin{aligned}
0 &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) - \frac{\partial L}{\partial \theta_1} \\
0 &= \frac{\mathrm{d}}{\mathrm{d}t}\Big((m_1+m_2)\, l_1^2 \dot{\theta}_1 + m_2 l_1 l_2 \dot{\theta}_2 \cos(\theta_1 - \theta_2)\Big) - \Big(-m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - g l_1 (m_1+m_2) \sin(\theta_1)\Big) \\
0 &= (m_1+m_2)\, l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g l_1 (m_1+m_2) \sin(\theta_1)
\end{aligned}$$

or:

$$(m_1+m_2)\, l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g (m_1+m_2) \sin(\theta_1) = 0 \tag{12}$$

Applying the Euler-Lagrange equation to θ_2, we get:

$$\begin{aligned}
0 &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \frac{\partial L}{\partial \theta_2} \\
0 &= \frac{\mathrm{d}}{\mathrm{d}t}\Big(m_2 l_2^2 \dot{\theta}_2 + m_2 l_1 l_2 \dot{\theta}_1 \cos(\theta_1 - \theta_2)\Big) - \Big(m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) - m_2 g l_2 \sin(\theta_2)\Big) \\
0 &= m_2 l_2^2 \ddot{\theta}_2 + m_2 l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 g l_2 \sin(\theta_2) \\
0 &= l_2 \ddot{\theta}_2 + l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g \sin(\theta_2)
\end{aligned}$$

or:

$$l_2 \ddot{\theta}_2 + l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g \sin(\theta_2) = 0 \tag{13}$$

We will now combine Equations (12) and (13) to create a system of equations:

$$\begin{cases}
0 = (m_1+m_2)\, l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g (m_1+m_2) \sin(\theta_1) \\
0 = l_2 \ddot{\theta}_2 + l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + g \sin(\theta_2)
\end{cases}$$

We can rewrite the above system of equations to express $\ddot{\theta}_1$ and $\ddot{\theta}_2$ with respect to the other variables and parameters:

$$\begin{cases} \ddot{\theta}_1 = \dfrac{-g\left(2m_1+m_2\right)\sin\left(\theta_1\right) - m_2 g \sin\left(\theta_1 - 2\theta_2\right) - 2m_2\left(\dot{\theta}_2^2 l_2 + \dot{\theta}_1^2 l_1 \cos\left(\theta_1-\theta_2\right)\right)\sin\left(\theta_1-\theta_2\right)}{l_1\left(2m_1+m_2-m_2\cos\left(2\left(\theta_1-\theta_2\right)\right)\right)} \\ \ddot{\theta}_2 = \dfrac{2\left(\dot{\theta}_1^2 l_1\left(m_1+m_2\right) + \left(m_1+m_2\right) g\cos\left(\theta_1\right) + \dot{\theta}_2^2 l_2 m_2 \cos\left(\theta_1-\theta_2\right)\right)\sin\left(\theta_1-\theta_2\right)}{l_2\left(2m_1+m_2-m_2\cos\left(2\left(\theta_1-\theta_2\right)\right)\right)} \end{cases}$$

Finally, we can modify the system once more. We will let $\omega_i = \dot{\theta}_i$ to obtain a system of 4 ordinary differential equations:

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 \\ \dot{\omega}_1 = \dfrac{-g\left(2m_1+m_2\right)\sin\left(\theta_1\right) - m_2 g \sin\left(\theta_1 - 2\theta_2\right) - 2m_2\left(\omega_2^2 l_2 + \omega_1^2 l_1 \cos\left(\theta_1-\theta_2\right)\right)\sin\left(\theta_1-\theta_2\right)}{l_1\left(2m_1+m_2-m_2\cos\left(2\left(\theta_1-\theta_2\right)\right)\right)} \\ \dot{\omega}_2 = \dfrac{2\left(\omega_1^2 l_1\left(m_1+m_2\right) + \left(m_1+m_2\right) g\cos\left(\theta_1\right) + \omega_2^2 l_2 m_2 \cos\left(\theta_1-\theta_2\right)\right)\sin\left(\theta_1-\theta_2\right)}{l_2\left(2m_1+m_2-m_2\cos\left(2\left(\theta_1-\theta_2\right)\right)\right)} \end{cases} \tag{14}$$

2 Program Manual

2.1 Summary of the program

This program can be ran in two ways. You can either run the Jupyter Notebook file `main.ipynb` or run the python script `main.py`. Both files will generate the same output. The purpose of this program is to not only solve the equations of motion of the double pendulum system, but also generate an animation of a double pendulum system with different animated plots that are in sync with animation of the double pendulum system.

2.2 Installation requirements

For this program to work, you need to be able to successfully run a Jupyter Notebook or a Python file. The rest of this subsection assumes you have the ability to open, edit, and run a Jupyter Notebook or a Python file. The required libraries you need in order to successfully run the program are `matplotlib` and `numpy`.

2.3 Details on the program

After the user has decided on the value of the initial conditions and the parameters, the user can run the program. Upon running the program, the first thing that the program will do is solve the system of equations in Equation (14). The python code that captures the system of equations can be found in Appendix B.1. The method I used to solve this system of differential equations is the Runge-Kutta order 4 method. My implementation of this method can be found in Appendix B.2. The code in Appendix B.2 has a function called `get_data()`. The purpose of this helper function is to gather the data we want more easily. The input is an $n \times m$ array and it transposes that array. Appendix B.3 shows the code for the function `get_data()`. Once the system of differential equations is solved, the program will generate five plots based the solution to the system of differential equations. These five plots will be saved as images for the user's reference and they are plotting:

1. θ_2 vs θ_1
2. ω_2 vs ω_1
3. θ_1, θ_2 vs time
4. ω_1, ω_2 vs time

5. Potential, Kinetic, and Total Energies vs time

After the plots are generated and saved as images, the program will create an animation. Using the constants, parameters, and initial conditions that the user initialized as well as the solution to the system of differential equations, this animation will involve:

- A simulation of the double pendulum system
- θ_2 vs θ_1
- ω_2 vs ω_1
- θ_1, θ_2 vs time
- ω_1, ω_2 vs time
- Potential, Kinetic, and Total Energies vs time

3 Results and Analysis

3.1 First set of initial conditions

Consider the case with the following parameters and initial conditions:

$$\begin{cases} m_1 = 1 \text{ kg} \\ m_2 = 1 \text{ kg} \end{cases}, \begin{cases} l_1 = 1 \text{ m} \\ l_2 = 1 \text{ m} \end{cases}, \begin{cases} \theta_{1,0} = 45^\circ \\ \theta_{2,0} = 0^\circ \end{cases}, \begin{cases} \omega_{1,0} = 0^\circ/\text{s} \\ \omega_{2,0} = 0^\circ/\text{s} \end{cases}, \; 0 \le t \le 10$$

where each $\theta_{i,0}$ and $\omega_{i,0}$ are the initial starting angles and angular velocities for each bob. Figure 2 is a collection of graphs that will be generated at the end of the program. Figures 2c and 2d are time plots for the position and angular velocity of each bob in the double pendulum system respectively. For a time of 10 seconds, it is clear that the double pendulum system under this set of initial conditions shows a consistent periodic behavior. These plots show that the double pendulum system has a period about 2.6383 seconds. Figures 2a and 2b are phase plane plots for the position and angular velocity of each bob in the double pendulum system respectively. It is more apparent in the animation when running the code, but these phase plane plots show that there is a consistent pattern in how the system behaves. Finally, Figure 2e is a time plot that plots the Potential, Kinetic, and Total energies against time. Here we can see that not only the consistent periodic behavior exist, but energy is conserved.

3.2 Slightly changing one initial condition

Lets use the same set up but change one of the initial conditions slightly, which is $\theta_{2,0} = 1^\circ$ and rerun the program. Figure 3 is a set of graphs that are generated under this set of initial conditions. Compared to Figures 2c and 2d, Figures 3c and 3d look very similar. However, we can see that the bobs' position and angular velocity starts to differ after each period in Figure 3c and 3d respectively. The system does show a periodic behavior, but each period is slightly different than the previous period. This is more apparent when we look at Figures 3a and 3b. These figures show that the system starts to be inconsistent while maintaining its periodic behavior over time. Just by increasing the initial angle of bob 2 by 1°, the system starts to diverge from its initial period. This shows that the Double Pendulum System is a chaotic system. Figure 3e shows how the energy is conserved under these initial conditions. While energy is conserved, the amount of Potential and Kinetic energy differs as time goes on, which aligns with the fact that the system behaves in a chaotic way.

4 Summary

What we have done is derived the necessary equations that capture the behavior of thee Double Pendulum system using simple mathematics, energy laws, and Lagrangian Mechanics. Using the code provided, we were able to solve the system of equations that captures the motion of the Double Pendulum system. Then, we were able to show that the Double Pendulum system is a chaotic system by showing that the behavior will change from a slight change in initial conditions.

Bibliography

[1] Erik Neumann. *Double pendulum.* Feb. 2002. URL: `https://www.myphysicslab.com/pendulum/double-pendulum-en.html`.

[2] Mark Newman. *Computational Physics*. First. University of Michigan. University of Michigan, 2013.

A Figures

A.1 Double Pendulum Details

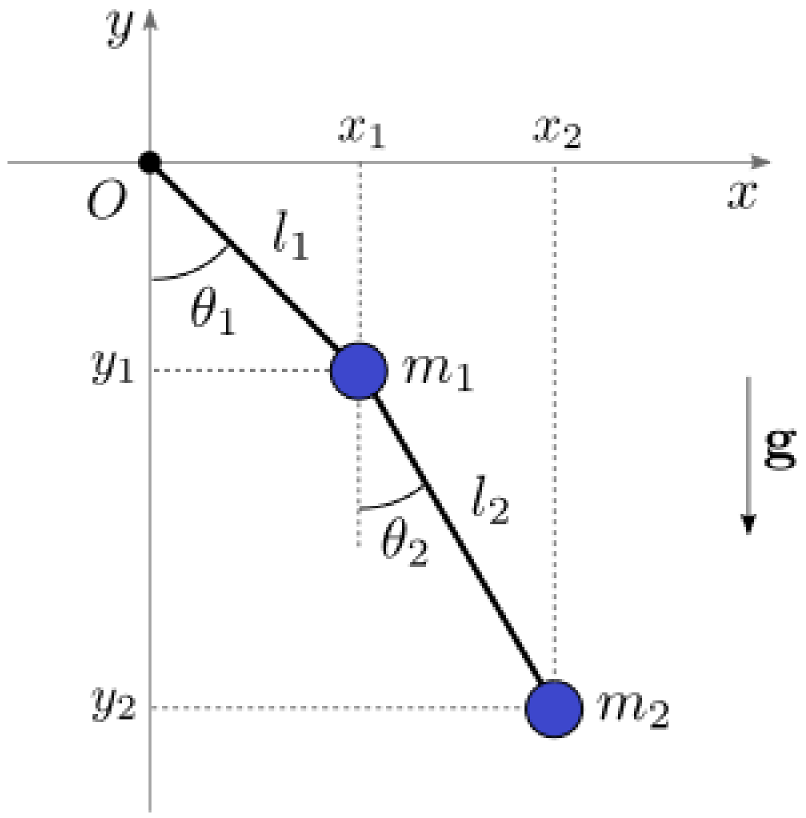
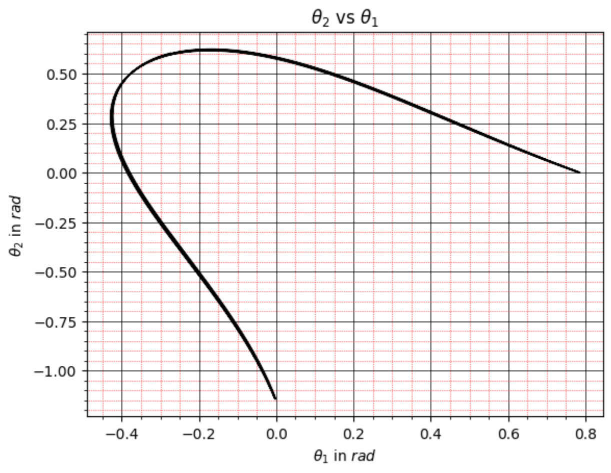
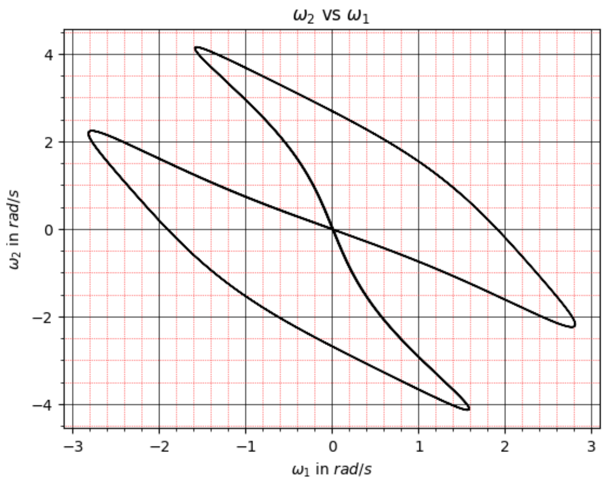


Figure 1: A visualization of a double pendulum

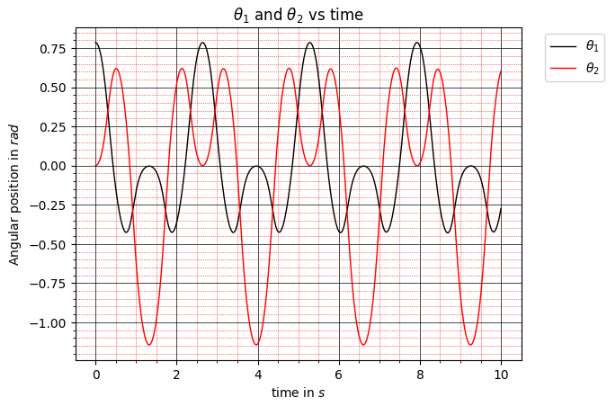
A.2 Results from the first set of initial conditions



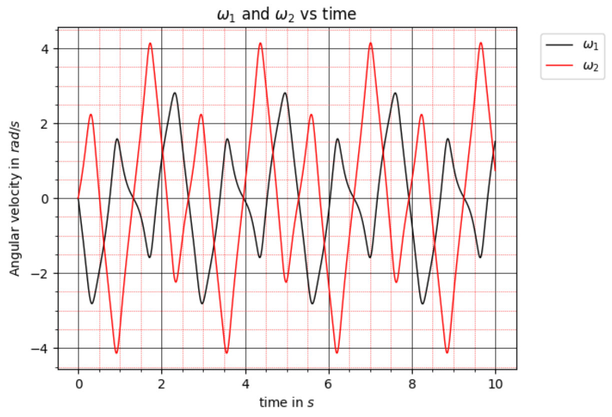
(a) θ_2 vs θ_1.



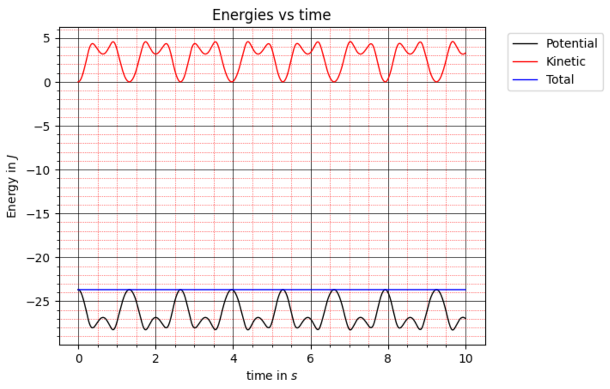
(b) ω_2 vs ω_1.



(c) θ_1, θ_2 vs time.



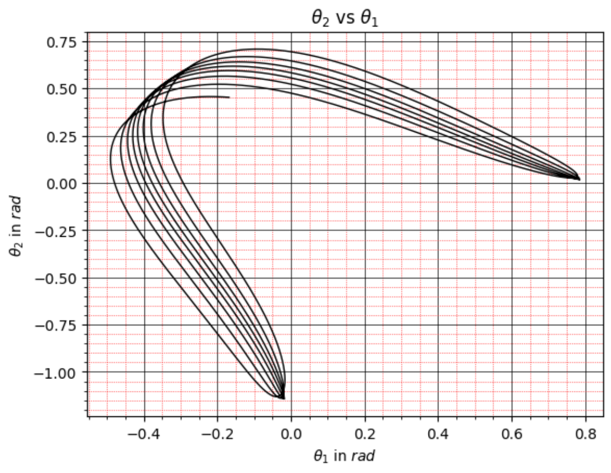
(d) ω_1, ω_2 vs time.



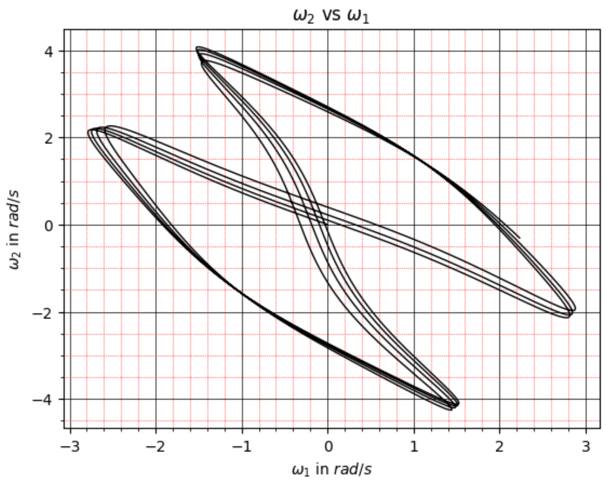
(e) Potential, Kinetic, and Total Energies vs time.

Figure 2: Solving the system from $t_1 = 0$ s to $t_2 = 10$ s where $m_1 = 1$ kg, $m_2 = 1$ kg, $l_1 = 1$ m, $l_2 = 1$ m, $\theta_{1,0} = 45°$, $\theta_{2,0} = 0°$, $\omega_{1,0} = 0°/\text{s}$, $\omega_{2,0} = 0°/\text{s}$.

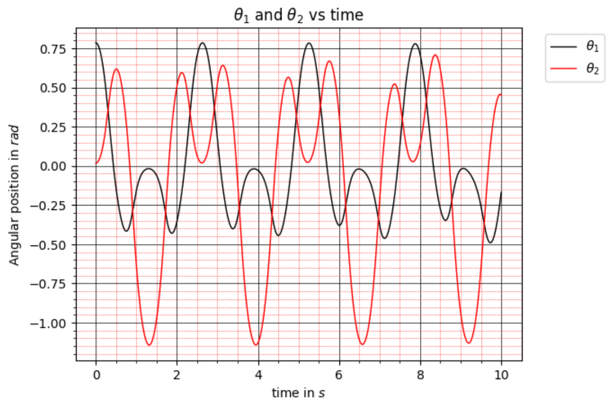
A.3 Results from the second set of initial conditions



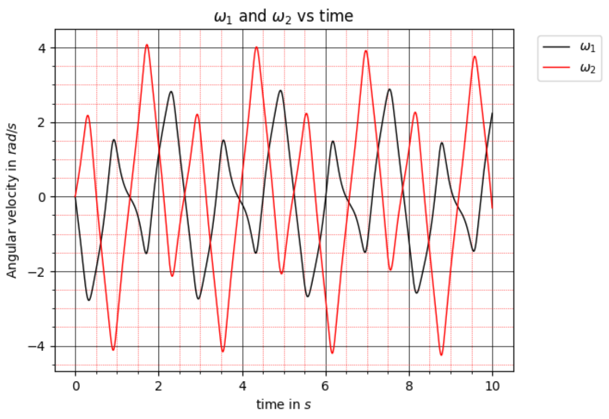
(a) θ_2 vs θ_1.



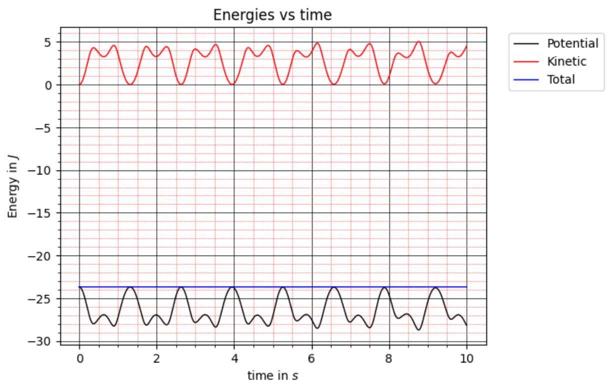
(b) ω_2 vs ω_1.



(c) θ_1, θ_2 vs time.



(d) ω_1, ω_2 vs time.



(e) Potential, Kinetic, and Total Energies vs time.

Figure 3: Solving the system from $t_1 = 0$ s to $t_2 = 10$ s with the same intial conditions from Figure 2 but with $\theta_{2,0} = 0°$ instead.

B Python code

B.1 System of Ordinary Differential Equations

```python
def df(F, t, params) -> np.array:
    # Unpack variables
    theta_1: float = F[0]
    theta_2: float = F[1]
    omega_1: float = F[2]
    omega_2: float = F[3]

    # Unpack parameters
    m_1: float = params[0]
    m_2: float = params[1]
    l_1: float = params[2]
    l_2: float = params[3]
    g: float = params[4]

    # System of differential equations
    theta_1_t: float = omega_1
    theta_2_t: float = omega_2

    # Breaking down each component since the expression is too big
    A1: float = -g * (2 * m_1 + m_2) * np.sin(theta_1)
    A2: float = - m_2 * g * np.sin(theta_1 - 2 * theta_2)
    A3: float = - 2 * m_2 * np.sin(theta_1 - theta_2)
    A4: float = (omega_2 ** 2) * l_2 + (omega_1 ** 2) * l_1 * np.cos(theta_1 - theta_2)
    B: float = l_1 * (2 * m_1 + m_2 - m_2 * np.cos(2 * (theta_1 - theta_2)))
    omega_1_t: float = (A1 + A2 + A3 * A4) / B

    # Breaking down each component since the expression is too big
    C1: float = 2 * np.sin(theta_1 - theta_2)
    C2: float = (omega_1 ** 2) * l_1 * (m_1 + m_2)
    C3: float = g * (m_1 + m_2) * np.cos(theta_1)
    C4: float = (omega_2 ** 2) * l_2 * m_2 * np.cos(theta_1 - theta_2)
    D: float = l_2 * (2 * m_1 + m_2 - m_2 * np.cos(2 * (theta_1 - theta_2)))
    omega_2_t: float = C1 * (C2 + C3 + C4) / D
    return np.array([theta_1_t, theta_2_t, omega_1_t, omega_2_t], dtype=float)
```

B.2 My Implementation of the Runge-Kutta Order 4 Method

```python
def rk4(df, r, ts, h, params):
    rs = [r]
    for i in range(1, len(ts)):
        f1 = df(rs[i - 1], ts[i - 1], params)
        f2 = df(rs[i - 1] + (h / 2) * f1, ts[i - 1] + (h / 2), params)
        f3 = df(rs[i - 1] + (h / 2) * f2, ts[i - 1] + (h / 2), params)
        f4 = df(rs[i - 1] + h * f3, ts[i - 1] + h, params)
        r = rs[i - 1] + (h / 6) * (f1 + 2 * f2 + 2 * f3 + f4)
        rs.append(r)
        pass
    return get_data(rs)
```

B.3 Helper Function

```python
def get_data(data):
    rs, cs = len(data), len(data[0])
    temp = np.empty((cs, rs), dtype=float)
    for r in range(0, rs):
        for c in range(0, cs):
            temp[c][r] = data[r][c]
            pass
        pass
    if len(temp) == 1:
        return temp[0]
    return temp
```