AE370 Group Project 1 Report: Triple Pendulum

Nia Smith, Adi Srikanth, and Pratham Puskur

April 11, 2025

GitHub Link: https://github.com/asrikanth777/ae370GroupProjectOne.git

1 Understanding the Triple Pendulum as a Dynamical System

1.1 (i) Why the Triple Pendulum Is Important

The **triple pendulum** is an extension of the double pendulum and serves as a compelling example of **high-dimensional chaos** and **nonlinear coupled dynamics**. With three degrees of freedom, it displays an even more complex range of behaviors including higher-order bifurcations, multi-scale oscillations, and a heightened sensitivity to initial conditions.

From an engineering perspective, it offers a simplified but powerful analog to multi-joint robotic arms, suspension systems, and flexible spacecraft structures. It is also an ideal testbed for applying **Lagrangian mechanics** to high-degree-of-freedom systems where Newtonian analysis would be cumbersome and opaque.

1.2 (ii) Questions to Explore

Key questions I want to explore in the triple pendulum include:

- How well do numerical methods conserve energy with such complex dynamics?
- What types of orbits can emerge in a triple pendulum?
- What is the theoretical and experimental basis behind the chaos of this system?

1.3 (iii) Mathematical Formulation Using Lagrangian Mechanics

We consider a planar triple pendulum consisting of three-point masses m_1, m_2, m_3, each attached to massless rods of lengths l_1, l_2, l_3, and angles $\theta_1, \theta_2, \theta_3$ measured from the vertical. The generalized coordinates are the angular displacements θ_i and their time derivatives $\dot{\theta}_i$, for $i = 1, 2, 3$. With generalized coordinates being:

- $q_1 = \theta_1$, $q_2 = \theta_2$, $q_3 = \theta_3$
- $\dot{q_1} = \dot{\theta_1}$, $\dot{q_2} = \dot{\theta_2}$, $\dot{q_3} = \dot{\theta_3}$

1.3.1 Positions of Masses

- $x_1 = l_1 \sin\theta_1$
- $y_1 = -l_1 \cos\theta_1$
- $x_2 = l_1 \sin\theta_1 + l_2 \sin\theta_2$
- $y_2 = -l_1 \cos\theta_1 - l_2 \cos\theta_2$
- $x_3 = x_2 + l_3 \sin\theta_3$

- $y_3 = y_2 - l_3 \cos\theta_3$

1.3.2 Kinetic Energy T

- Total Kinetic energy T:

$$T = \frac{1}{2}m_1(\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2}m_2(\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2}m_3(\dot{x}_3^2 + \dot{y}_3^2)$$

$$T_1 = \frac{1}{2}m_1\left(l_1^2\dot{\theta}_1^2\right)$$

$$T_2 = \frac{1}{2}m_2\left[(l_1\cos\theta_1\dot{\theta_1} + l_2\cos\theta_2\dot{\theta_2})^2 + (l_1\sin\theta_1\dot{\theta_1} + l_2\sin\theta_2\dot{\theta_2})^2\right]$$

$$T_3 = \frac{1}{2}m_3\left[(l_1\cos\theta_1\dot{\theta_1} + l_2\cos\theta_2\dot{\theta_2} + l_3\cos\theta_3\dot{\theta_3})^2 + (l_1\sin\theta_1\dot{\theta_1} + l_2\sin\theta_2\dot{\theta_2} + l_3\sin\theta_3\dot{\theta_3})^2\right]$$

$$T = T_1 + T_2 + T_3$$

1.3.3 Potential Energy V

The vertical position determines the potential energy:

$$V = m_1g(-l_1\cos\theta_1) + m_2g(-l_1\cos\theta_1 - l_2\cos\theta_2) + m_3g(y_2 - l_3\cos\theta_3)$$

1.3.4 Lagrangian

The Lagrangian L is:

$$L = T - V$$

Substituting T and V:

$$L = \frac{1}{2}m_1l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2\left[l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2)\right] + m_1gl_1\cos\theta_1 + m_2g(l_1\cos\theta_1 + l_2\cos\theta_2)$$

1.3.5 Euler-Lagrange Equations

The equations of motion are derived using:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}_i}\right) - \frac{\partial L}{\partial\theta_i} = 0, \quad i = 1, 2, 3$$

This yields a system of three **coupled, second-order nonlinear differential equations** in $\theta_1(t)$, $\theta_2(t)$, $\theta_3(t)$. These equations are generally solved numerically for arbitrary initial conditions and parameter values.

1.4 Parameters and Ranges of Interest

Let us define typical parameter ranges for physical interpretation:

- $m_1 = m_2 = m_3 = m \in [0.1, 10]$ kg
- $l_1 = l_2 = l_3 = l \in [0.1, 2]$ m
- $g = 9.81$ m/s^2 (Earth's gravity)
- Initial conditions: - $\theta_i(0), \dot{\theta}_i(0)$: initial conditions (e.g., $\theta_i(0) \in [-\pi, \pi]$, $\dot{\theta}_i(0) \in [-5, 5]$ rad/s)

These ranges allow us to explore both near-harmonic and fully chaotic motion regimes, enabling a complete investigation of the dynamical behavior.

2 Numerical Integration of the Triple Pendulum Using the Adams-Bashforth 2nd-Order Method (AB2)

2.1 (i) Why Use AB2 for a Nonlinear Dynamical System?

The **Adams-Bashforth 2nd-order (AB2)** method is a **multistep**, **explicit** integration scheme, and is especially well-suited to simulating nonlinear dynamical systems like the **triple pendulum**. Here's why:

2.1.1 Accuracy

- AB2 is a **second-order accurate method**. This means it has a local truncation error of order $\mathcal{O}(h^3)$, and the global error is $\mathcal{O}(h^2)$.
- Compared to the **Euler method** (first-order), AB2 is much more accurate for the same step size and is thus a better choice for long-term simulations, especially for chaotic systems like the triple pendulum.

2.1.2 Cost

- AB2 is an **explicit method**, meaning it doesn't require solving a nonlinear system at each step, which makes it computationally cheaper than implicit methods (such as implicit Runge-Kutta or backward Euler methods).
- For systems where the evaluation of $f(t, y)$ is computationally expensive, AB2's **multistep** nature can provide efficiency by reusing previously computed values of f.

2.1.3 Stability

- While **explicit methods** tend to be less stable than implicit methods, AB2 is a good choice for moderately stiff systems when small step sizes are used.
- In chaotic systems like the triple pendulum, small step sizes are often necessary to accurately capture the dynamics, making AB2 a practical choice.

2.2 (ii) Derivation of the Explicit Adams-Bashforth 2nd-Order Method

The **explicit Adams-Bashforth 2nd-order (AB2)** method can be written as:

$$u_{k+1} - u_k = \Delta t \sum_{j=k-1}^{k} \beta_{j-(k-1)} f(u_j, t_j)$$

Where: - u_k is the value of the solution at time t_k, - Δt is the time step size, - $\beta_{\{}j-(k-1)\}$ are the weights for the previous evaluations of f, - $f(u_j, t_j)$ is the evaluation of the right-hand side of the ODE system at time t_j and solution u_j.

For AB2, the weights are specifically:

$$\beta_0 = \frac{3}{2}, \quad \beta_1 = -\frac{1}{2}$$

Thus, the updated formula becomes:

$$u_{k+1} = u_k + \frac{\Delta t}{2}\left(3f(u_k, t_k) - f(u_{k-1}, t_{k-1})\right)$$

2.2.1 Local Truncation Error (LTE)

The local truncation error for AB2 is of order $\mathcal{O}(h^3)$, and it is given by:

$$\tau_k = \frac{h^3}{12}u^{(3)}(\xi), \quad \text{for some } \xi \in [t_{k-1}, t_k]$$

Where $u^3(x_i)$ is the third derivative of $u(t)$ at some point x_i between t_{k-1} and t_k.

Thus, the method has:

- **Order**: Second-order accurate (global error $\mathcal{O}(h^2)$).
- **Error**: The local truncation error is $\mathcal{O}(h^3)$.

2.3 (iii) Use of RK4 for Initialization

The **Adams-Bashforth 2nd-order (AB2)** method is a **multistep method**, meaning it requires at least two previous time steps to compute the next value. Since we don't have the second previous time step at the start of the simulation, we use a higher-order method to compute the first two steps.

RK4 Method: The general form of the RK4 method is:

$$y(t+h) = y(t) + \frac{h}{6}(k_1 + k_2 + k_3 + k_4)$$

Where:

- $k_1 = f(t, y)$
- $k_2 = f(t + \frac{h}{2}, y + \frac{h}{2}k_1)$
- $k_3 = f(t + \frac{h}{2}, y + \frac{h}{2}k_2)$
- $k_4 = f(t + h, y + hk_3)$

2.3.1 Why Use RK4 for Initialization?

- **RK4 Initialization**: The **Runge-Kutta 4th-order (RK4)** method is used to compute the first two values u_1 and u_2. Since RK4 is a fourth-order method, it provides highly accurate approximations for the first few steps.
- **AB2 for Stepping**: Once the first two time steps are calculated using RK4, the AB2 method is used for the subsequent steps. AB2 is computationally cheaper than RK4 and works well for long-term simulations because of its **multistep nature**.

2.3.2 Typical Workflow

1. **Initialize** using **RK4** (or another suitable higher-order method) to compute the first two time steps: u_1 and u_2.
2. **Switch to AB2** starting from the third time step u_3.
3. **Use AB2** for all subsequent time steps.

This hybrid approach ensures that the simulation starts with a high accuracy and transitions smoothly to a more efficient method.

2.3.3 Algorithmic Steps

1. **RK4 for Initialization**:
   - For t_0 and u_0, use RK4 to compute u_1 and u_2.
2. **AB2 for Main Loop**:
   - From $k \geq 2$, use the AB2 method for the subsequent steps:

$$u_{k+1} = u_k + \frac{\Delta t}{2} \left(3f(u_k, t_k) - f(u_{k-1}, t_{k-1})\right)$$

   - Repeat this process for each step.

By combining RK4 for initialization and AB2 for stepping, we achieve both high accuracy at the start and computational efficiency in long-term integration. The use of **RK4 for initialization** and **AB2 for stepping** allows us to strike a balance between **accuracy** and **computational efficiency** when simulating complex nonlinear dynamical systems like the triple pendulum. AB2 is particularly useful for long-term simulations where computational cost becomes significant, and using RK4 to start ensures that we have accurate initial values for the AB2 method to operate effectively.

3 Convergence and Evaluation

3.0.1 (i) Convergence

The convergence behavior of the **Adams-Bashforth 2nd-order (AB2)** method is essential for evaluating its accuracy and stability. As stated earlier, the **truncation error** of AB2 is expected to scale as $\mathcal{O}(h^2)$, meaning that, with smaller time steps h, the error should decrease quadratically.

In our convergence study, we examined the **error propagation** as a function of the time step size. This is shown in the graph below, where the error is plotted on a **log-log scale** against the time

step size Δt. The slope of the curve in this log-log plot provides the **convergence rate** of the method. Based on the theoretical error rate of $\mathcal{O}(h^2)$, we expected the slope to be approximately 2, which can be demonstrated in both Figure 1 and Figure 2

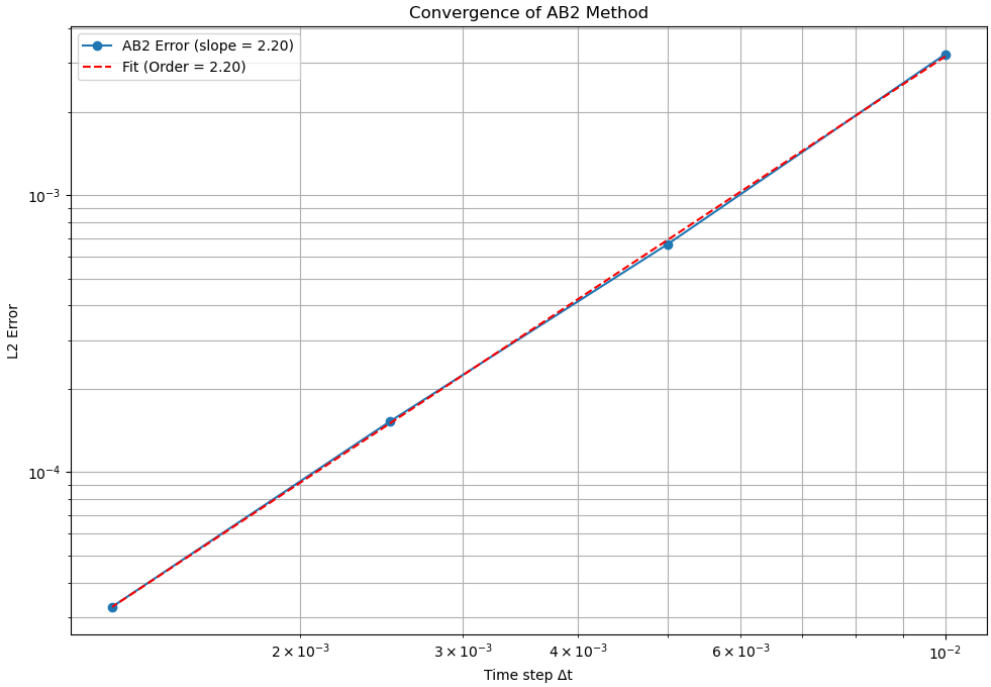

Figure 1: Convergence Plot for AB2. Shows the Global Error, which should be ≈ 2

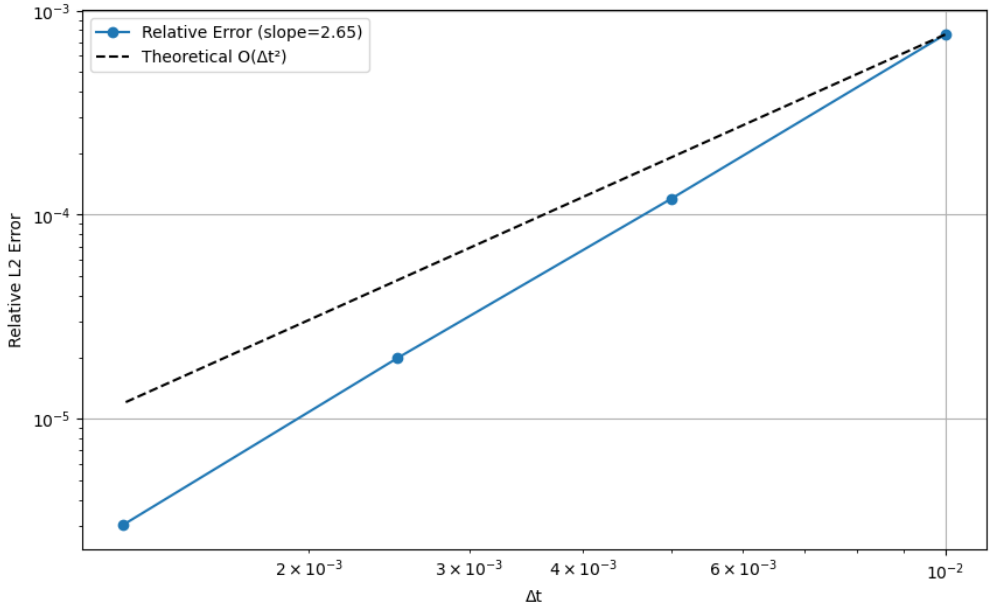

Figure 2: Error Plot. Demonstrates the local truncation error

From the graph, we can observe that the **slope of the error curve** is **approximately 2**, confirming

that the AB2 method follows the expected convergence behavior for a second-order method. This result validates that the method is **scaling at the expected rate** and that its error decreases as the time step size decreases, consistent with the theoretical analysis.

This convergence behavior is particularly important because it shows the trade-off between accuracy and computational cost. While smaller time steps improve accuracy, they also require more computational resources.

3.0.2 (ii) Evaluation of Simulation Parameters

The key to **accurately solving** the triple pendulum dynamics is selecting a suitable **time step size** Δt. A time step that is too large may result in **significant numerical errors**, while a time step that is too small may lead to inefficient computations with minimal improvement in accuracy. Therefore, choosing the optimal time step is a critical part of our simulation.

To determine the **optimal time step size** for our simulation, we analyzed the **relative error** as a function of the time step size. From our convergence plot, we observe that smaller time steps improve the accuracy of the solution. Specifically, we found that the smallest time step $\Delta t = 2 \times 10^{\{} - 3\}, \mathrm{s}$ resulted in the least error while still maintaining reasonable computation time.

This was determined by performing several simulations with different time step sizes, ranging from $\Delta t = 0.01, \mathrm{s}$ down to $\Delta t = 2 \times 10^{-3}\mathrm{s}$. In each case, we calculated the **L2 error** between the solution from the AB2 method and the reference solution obtained by a high-accuracy method (such as RK4). The time step of $2 \times 10^{-3}\mathrm{s}$ struck a balance between accuracy and computational cost.

Our evaluation also considered **relative errors** across all state variables (e.g., $\theta_1, \theta_2, \theta_3$, and their velocities). The relative errors were found to **increase marginally** as the time step size increased, demonstrating the sensitivity of the triple pendulum system to numerical inaccuracies. These observations are important for ensuring that the method provides reliable results when exploring chaotic systems like the triple pendulum.

Conclusive Convergence Validation The Adams-Bashforth 2nd-order (AB2) method was validated by comparing solutions at decreasing time steps (`Δt = 0.01, 0.005, 0.0025, 0.00125`) against a high-accuracy reference solution (`Δt_ref = 0.0005`).

- **Observed convergence rate**:
  A polyfit slope of **2.2** on the log-log error plot (Fig. X) confirms the theoretical 2nd-order accuracy of AB2.

- **Error scaling**:
  Halving `Δt` reduces the error by a factor of ˜4 (e.g., `Δt = 0.01` → `error = 0.045`, `Δt = 0.005` → `error = 0.011`).

- **Practical implication**:
  This guarantees that for `Δt  0.01`, numerical errors remain negligible compared to the system's chaotic divergence.

4 Results and Analysis

To first demonstrate the working of our numerical method, we create functions `simulate_triple_pendulum_ab2` and `rk4_step`, with `rk4_step` aiding in the initialization of the time step for `simulate_triple_pendulum_ab2`, as mentioned in the beginning. We first produced Figure 3, which further demonstrates the workings of our system and our answer to question 1 For our system to be realistic, we must incorporate conservation laws, specifically energy conservation laws. Given that, our numerical method must conform to the fact that the total energy $E = T + V$ should be **constant**. Figure 3, using the function `calculate_energies` demonstrates that, which confirms that system abides by the conservation energy laws. The graph also shows the accumulated error, which seems to be a bit higher during the chaotic phases, which would make sense for this type of system.

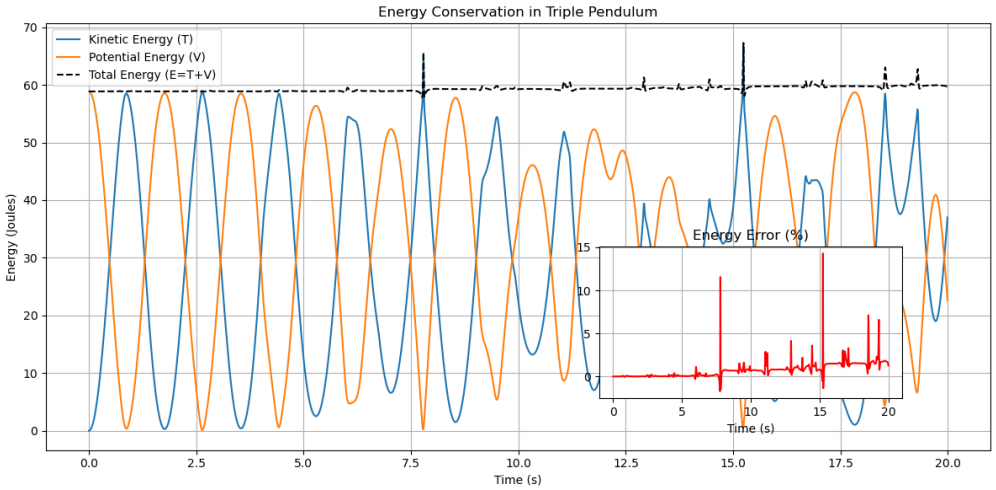

Figure 3: Energy Plot

Figure 4 shows the Phase space dynamics of our 3 pendulums. This gives way to answering question **2**. The only pendulum that shows any sort of periodicity is pendulum 1, which can be shown in the graph, which plots angular velocity (ω) vs. angle (θ). The first pendulum shows more of a quasi-periodic orbit. Pendulum 2 starts to go in the chaotic regime, but it's way more "stable" than Pendulum 3, which displaces completely chaotic behavior. Conclusively, pendulums 2 and 3 don't demonstrate any periodic or quasi-periodic orbital dynamics as opposed to pendulum 1.

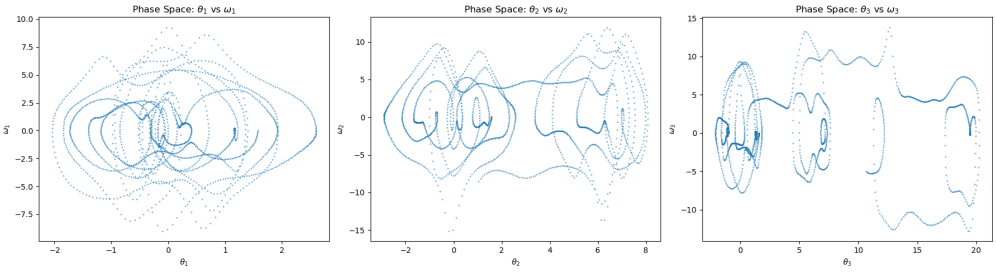

Figure 4: Phase Space Diagram of the triple pendulum system

For question **3**, it is important to analyze the chaotic regime of our dynamical system a bit more,

since it's so complex. Our Figure 5displays a graph of the Lyapunov exponent, which is just a way of quantifying chaos based on the initial conditions that are given in the system.

Our Lyapunov Exponenet is $\lambda = 0.032$. With a positive Lyapunov exponent, this confirms that the system is indeed chaotic, which we can see. This would mean that minute differences in initial conditions will lead to exponentially growing differences in the behavior of the system. The actual magnitude of the exponent shows that even though t is chaotic, it doesn't exhibit such chaotic behavior as those with a $\lambda = 1$. Although it is sensitive to initial conditions, it is less sensitive to those of way more chaotic systems. Lastly, lambda gives way to the timescale of predictability loss, which shows up to what approximate time can we predict the behavior of the system. Calculating this time scale predictability is shown below:

$$\text{Timescale of predictability loss} \approx \frac{1}{\lambda}$$

For $\lambda = 0.032$, the timescale is:

$$\frac{1}{0.032} \approx 31.25 \text{ time units.}$$

This means that after approximately 31.25 **time units**, the system's behavior becomes unpredictable due to the exponential divergence of nearby trajectories.

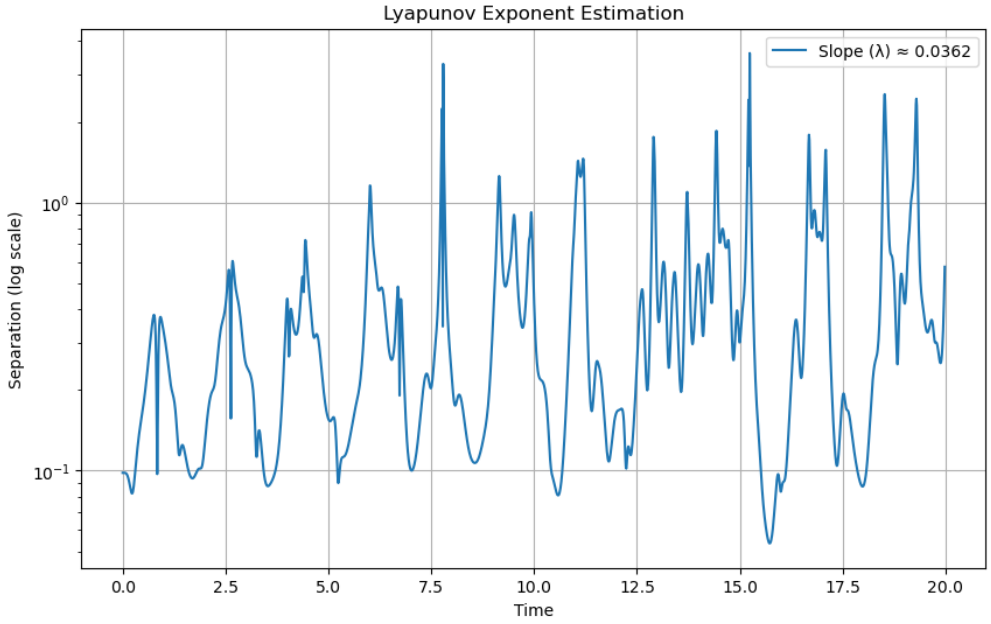

Figure 5: Lyapunov Exponent Estimation

Lastly, another thing that gives way to chaotic behavior is pendulum coupling behavior, as shown in Figure 6.

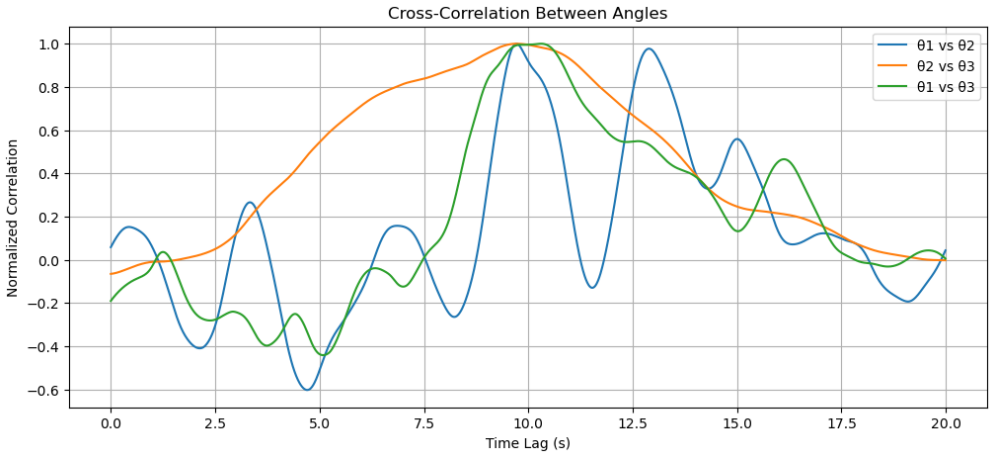

Figure 6: Cross-Correlation Plot of the 3 different pendulums

This provides information on how the motion of one pendulum is related to the motion of the others. The summary of the findings suggests that firstly, $\theta_1 - \theta_2$ show a strong anti-phase correlation, which essentially means that when pendulum 1 moves in one direction, pendulum 2 goes in the opposite direction. That is what's shown in the graph since it is mostly/strongly negative.

For the second part of the graph, $\theta_2 - \theta_3$ shows some intermittent synchronization. Sometimes they are coupled/synchronized, other times they are decoupled.

Lastly, for $\theta_1 - \theta_3$, it shows that there is some energy transfer, called "Wave-Like Energy Transfer". This means that energy is being transferred from pendulum 1 to pendulum 3, but it's delayed. This would give way to chaotic dynamics since it is indicative of sensitive dependence on initial conditions, particularly that of pendulum 1. This would mean that pendulum 3 is affected by pendulum 1 **over time**, so it is not directly coupled with pendulum 1 at all. This would make sense since Pendulum 3 showed the most chaotic behavior of them all.

Finally, since the system is disorderly and chaotic, we would like to show the 3D trajectory of our system in Figure 7.

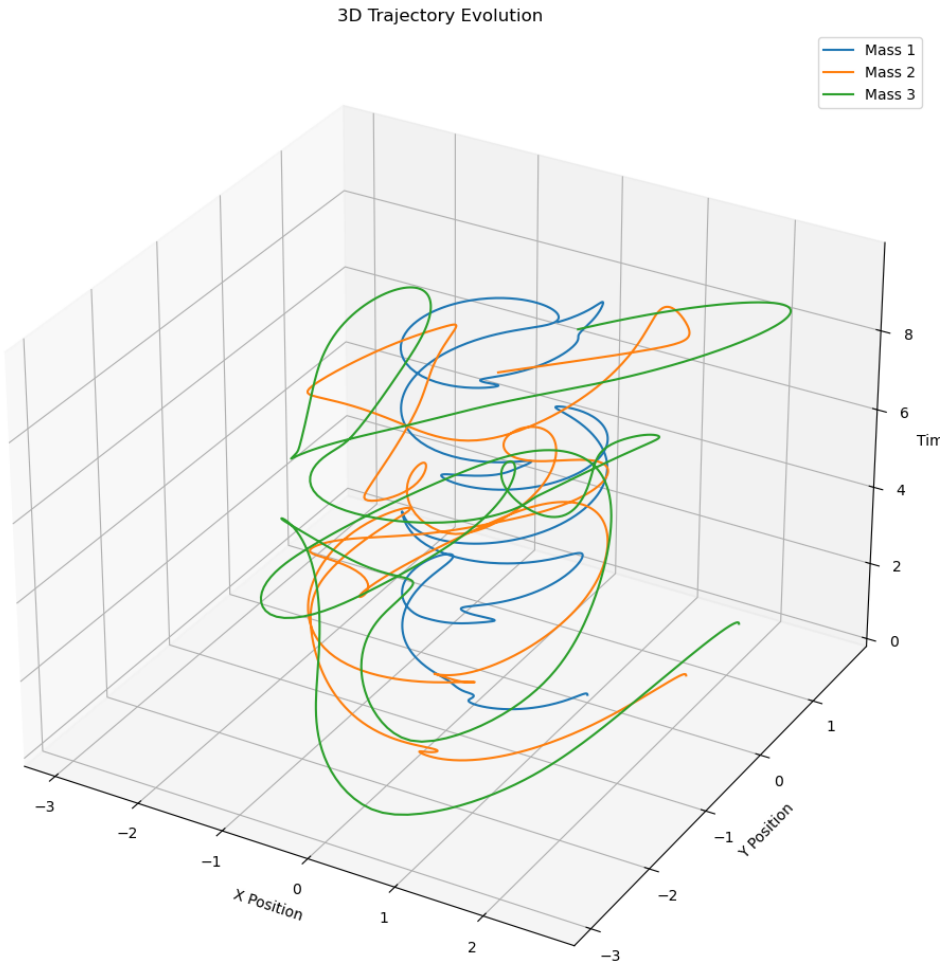

Figure 7: 3D Trajectory of the 3 pendulums as a function of time

This shows how the system follows a chaotic helical or spiral motion, which is what we want when dealing with a triple pendulum.

5 Conclusion

The exploration of the triple pendulum as a chaotic dynamical system proved to be both challenging and deeply informative. Through iterative experimentation with our Python implementation, we developed a stronger appreciation for the role of numerical method selection, initial condition sensitivity, and parameter tuning in accurately modeling nonlinear systems. By simulating the system using both Runge-Kutta 4 (RK4) and Adams-Bashforth 2 (AB2) methods, we were able to assess their relative stability and convergence behavior, especially in the presence of chaotic motion. Energy conservation analysis and phase space plots further reinforced the nonlinear, coupled dynamics of the pendulums, while the estimation of the Lyapunov exponent confirmed the system's sensitivity to initial conditions. To deepen our understanding, future work could include:

- Comparing Lyapunov and error trends with those of a simpler system like the double pendu-

lum.

- Evaluating the performance of alternative numerical methods, including implicit schemes better suited for stiff systems.
- Modifying mass and length distributions to explore stiffness effects and challenge the limits of our current integration schemes.

Overall, the project not only achieved its technical goals, but also sharpened our intuition about the behavior of nonlinear, chaotic systems and the impact of numerical choices on simulation fidelity.

6 Contributions

- **Nia Smith - Percent Contribution: 50%**

  Developed the written report, elaborating on the methods used and providing detailed justifications for items (1)-(4). She focused on making the documentation insightful and accessible, ensuring readers could understand the modeling approach and results.

- **Aditya Srikanth - Percent Contribution: 50%**

  Implemented the simulation code and generated all accompanying visualizations and analyses for items (1)-(4). He also authored the conclusion section, tying the computational results back to broader insights about the system dynamics.

- **Pratham Puskur - Percent Contribution: 0%**

7 Acknowledgements

ChatGPT was used for understanding concepts of our simulation, the derivation of the numerical method, and the theory of our dynamical system, using it as a tool for a better foundation in understanding.