University of Utah, Mathematics Department

# CHAOTIC MOTION
# IN
# DOUBLE PENDULUM SYSTEMS

## A Final Computing Project

Charles Godfrey    Tannon Warnick    Henry Whiting

**Abstract**

This project explores the chaotic dynamics of a double pendulum system, a classic example of how simple physical laws can yield complex, unpredictable behavior. Starting from the Lagrangian formulation, we derive the coupled non-linear equations of motion governing the system. We implement the fourth-order Runge-Kutta (RK4) numerical method to simulate the system's trajectory over time. The study demonstrates the system's extreme sensitivity to initial conditions the "butterfly effect" where infinitesimal differences in starting states lead to exponentially diverging outcomes.

December 5, 2025

# Contents



# List of Figures

# 1. Introduction

A single pendulum is a classical example of simple harmonic motion. When constrained to small angles the pendulum will swing periodically and consistently. By simply adding a second pendulum at the end of the first, the system transforms into a classical example of chaotic motion. Even though these two systems are governed by the same physical laws of motion and only being acted upon by one force (gravity), a double pendulum is *heavily* dependent on initial conditions. We will use an approximation to solve this equation to view the behavior; the approximation we will use is the fourth order Runge-Kutta method.

# 2. Mathematical Model

To simulate the double pendulum, we must first derive the equations of motion. Because the system involves multiple degrees of freedom, the Lagrangian formulation of mechanics is significantly more efficient than the Newtonian approach.

## 2.1. System Setup and Geometry

We consider a system of two point masses, $m_1$ and $m_2$, connected by massless rigid rods of length $l_1$ and $l_2$. The system is confined to a 2D plane and acted upon by a uniform gravitational field $g$ pointing downwards. The system can be visualized in Figure 2.1 on this page. Note: unlike in Figure 2.1, the mathematical (and numeric) system will not intersect with itself or other objects.

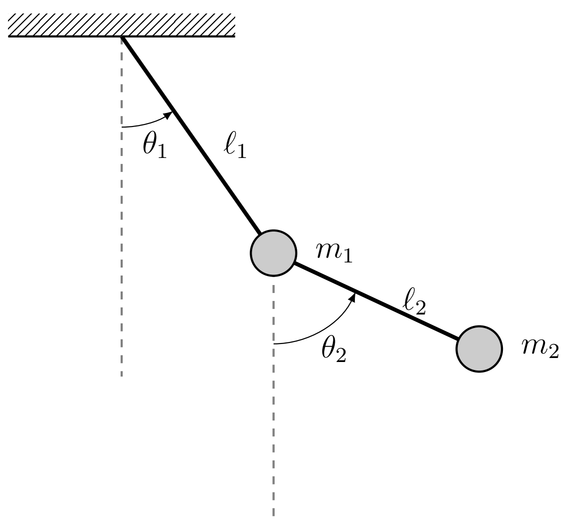


Figure 2.1: Double pendulum system

The system we will construct mathematically and depicted by Figure 2.1 is a simple pendulum, which means that there is a point mass at the end of a massless rod. A compound pendulum can be similarly derived by just substituting the position of the point mass by the position of the center of mass respectively.

We define the generalized coordinates as the angles $\theta_1$ and $\theta_2$, measured from the vertical axis. The origin $(0, 0)$ is the pivot point of the first pendulum.

$$x_1 = \ell_1 \sin\theta_1 \tag{2.1}$$
$$y_1 = -\ell_1 \cos\theta_1 \tag{2.2}$$
$$x_2 = \ell_1 \sin\theta_1 + \ell_2 \sin\theta_2 \tag{2.3}$$
$$y_2 = -\ell_1 \cos\theta_1 + \ell_2 \cos\theta_2 \tag{2.4}$$

## 2.2. The Lagrangian Formulation

The Lagrangian $\mathcal{L}$ is defined as the difference between the kinetic energy $(T)$ and the potential energy $(V)$ of the system:

$$\mathcal{L} = T - V$$

### 2.2.1. Kinetic Energy ($T$)

The total kinetic energy is the sum of the kinetic energies of the individual masses:

$$T = \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2$$

Calculating the velocities squared ($v^2 = \dot{x}^2 + \dot{y}^2$):

$$\begin{aligned}
v_1^2 &= (l_1\dot{\theta}_1)^2 \\
v_2^2 &= \dot{x}_2^2 + \dot{y}_2^2 \\
&= l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1\dot{\theta}_2(\sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2) \\
&= l_1^2\dot{\theta}_1^2 + l_2^2\dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2)
\end{aligned}$$

Thus, the total kinetic energy is:

$$T = \frac{1}{2}(m_1 + m_2) l_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2 l_2^2\dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) \tag{2.5}$$

### 2.2.2. Potential Energy ($V$)

Assuming potential energy is zero at $y = 0$, we have:

$$\begin{aligned}
V &= m_1 g y_1 + m_2 g y_2 \\
&= -(m_1 + m_2) g l_1 \cos\theta_1 - m_2 g l_2 \cos\theta_2
\end{aligned} \tag{2.6}$$

## 2.3. Equations of Motion

Applying the Euler-Lagrange equation for each coordinate $\theta_i$:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = 0$$

Solving these derivatives yields a system of two coupled, non-linear second-order differential equations.

**For $\theta_1$:**

$$(m_1 + m_2)l_1\ddot{\theta}_1 + m_2l_2\ddot{\theta}_2\cos(\theta_1 - \theta_2) + m_2l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) + (m_1 + m_2)g\sin\theta_1 = 0 \tag{2.7}$$

**For $\theta_2$:**

$$m_2l_2\ddot{\theta}_2 + m_2l_1\ddot{\theta}_1\cos(\theta_1 - \theta_2) - m_2l_1\dot{\theta}_1^2\sin(\theta_1 - \theta_2) + m_2g\sin\theta_2 = 0 \tag{2.8}$$

These two equations fully describe the motion of the double pendulum. In Section 3, we will rearrange these terms to solve for $\ddot{\theta}_1$ and $\ddot{\theta}_2$ explicitly to implement the RK4 algorithm.

# 3. Preparing for Numerical Model

Implementing RK4 requires transforming the coupled second-order differential equations derived in Section 2 into a system of first-order differential equations.

## 3.1. Matrix Formulation of Motion

We begin with the coupled Euler-Lagrange equations (2.7) and (2.8). By grouping the angular acceleration terms $(\ddot{\theta}_1, \ddot{\theta}_2)$ on the left-hand side and the remaining velocity and potential terms on the right, we obtain the following linear system:

$$(m_1 + m_2)l_1\ddot{\theta}_1 + m_2l_2\cos(\theta_1 - \theta_2)\ddot{\theta}_2 = -m_2l_2\dot{\theta}_2^2\sin(\theta_1 - \theta_2) - (m_1 + m_2)g\sin\theta_1, \tag{3.1}$$

$$m_2l_1\cos(\theta_1 - \theta_2)\ddot{\theta}_1 + m_2l_2\ddot{\theta}_2 = m_2l_1\dot{\theta}_1^2\sin(\theta_1 - \theta_2) - m_2g\sin\theta_2. \tag{3.2}$$

This system can be written in matrix form as $M(\theta)\ddot{\theta} = b(\theta, \omega)$, where $\omega_i = \dot{\theta}_i$:

$$\underbrace{\begin{pmatrix} (m_1 + m_2)l_1 & m_2l_2\cos(\theta_1 - \theta_2) \\ m_2l_1\cos(\theta_1 - \theta_2) & m_2l_2 \end{pmatrix}}_{M(\theta)} \underbrace{\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}}_{\ddot{\theta}} = \underbrace{\begin{pmatrix} -m_2l_2\omega_2^2\sin(\theta_1 - \theta_2) - (m_1 + m_2)g\sin\theta_1 \\ m_2l_1\omega_1^2\sin(\theta_1 - \theta_2) - m_2g\sin\theta_2 \end{pmatrix}}_{b(\theta,\omega)}.$$

## 3.2. Explicit Acceleration Terms

To solve for the angular accelerations $\ddot{\theta}$, we invert the mass matrix $M$. For a $2 \times 2$ matrix, the inverse is explicitly given by:

$$\ddot{\theta} = M^{-1}b = \frac{1}{\det M}\begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Defining the angle difference as $\Delta = \theta_1 - \theta_2$, the determinant is $\det M = m_2l_1l_2(m_1 + m_2\sin^2\Delta)$. Carrying out the matrix multiplication yields the explicit accelerations:

$$\ddot{\theta}_1 = \frac{1}{\det M}\left[m_2 l_2 b_1 - m_2 l_2 \cos(\Delta) b_2\right], \tag{3.3}$$

$$\ddot{\theta}_2 = \frac{1}{\det M}\left[-m_2 l_1 \cos(\Delta) b_1 + (m_1 + m_2) l_1 b_2\right]. \tag{3.4}$$

Substituting the components of vector $b$ (from (3.1) and (3.2)) into these expressions provides the functions necessary for the numerical solver.

### 3.3. State-Space Representation

Finally, we convert the two second-order equations into a system of four first-order equations. We introduce the state vector $X$:

$$X = \begin{pmatrix} \theta_1 & \theta_2 & \omega_1 & \omega_2 \end{pmatrix}^T.$$

The time evolution of the system is governed by $\dot{X} = F(X)$:

$$\dot{X} = \frac{d}{dt}\begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \ddot{\theta}_1(\theta_1, \theta_2, \omega_1, \omega_2) \\ \ddot{\theta}_2(\theta_1, \theta_2, \omega_1, \omega_2) \end{pmatrix},$$

where $\ddot{\theta}_1$ and $\ddot{\theta}_2$ are determined by equations (3.3) and (3.4). This vector field $F(X)$ is passed directly to the RK4 algorithm.

## 4. Implementing Numerical model

The full Julia code used to execute the system used in this derivation can be found in Appendix A on the current page. Now that we have a first order differential equation, we can implement it into a Runge-Kutta model. These kinds of methods are incredibly accurate even for chaotic dynamical systems.

## 5. Results

## 6. Conclusion

## A. Julia Code