Double Pendulum

Siyuan Chen, Ziyao Wang, Kaixuan Zhou

November 22, 2023

1 Introduction

Double pendulum, also known as chaotic pendulum, is the system that a pendulum attached to another pendulum's end shown below.

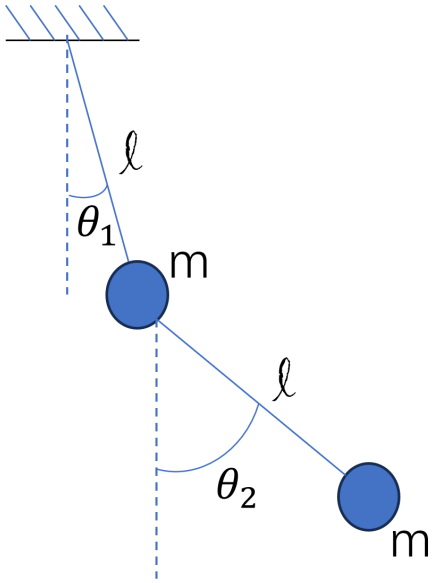

Figure 1: This is the double pendulum figure. l is the length of the massless arms between the two bobs with mass m. θ_1 and θ_2 are the angles between the arm and the vertical line.

Our project is designed to calculate the trajectory of the double pendulum and figure out the chaos of the system.

2 Preparation

The degree of freedom of this system is 2. We use the angle θ_1 and θ_2 as the generalized coordinates. We have chosen to measure the potential downwards from the level of the pivot and we get the following potential:

$$V = mgh_1 + mgh_2 = -mgl(2cos\theta_1 + \cos\theta_2), \tag{1}$$

The velocities of the two bobs are given by:

$$\begin{aligned} v_1 &= l\dot{\theta}_1 \\ v_2^2 &= l^2[\dot{\theta}_1^2 + \frac{1}{2}\dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2)] \end{aligned} \tag{2}$$

The total kinetic energy is

$$T = \frac{1}{2}mv_1^2 + \frac{1}{2}mv_2^2 = ml^2[\dot{\theta}_1^2 + \frac{1}{2}\dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2)] \tag{3}$$

The Lagrangian of the system is

$$L = T - V = \frac{1}{2}mv_1^2 + \frac{1}{2}mv_2^2 = ml^2[\dot{\theta}_1^2 + \frac{1}{2}\dot{\theta}_2^2 + \dot{\theta}_1\dot{\theta}_2 cos(\theta_1 - \theta_2)] + mgl(2cos\theta_1 + \cos\theta_2) \tag{4}$$

By Euler-Lagrangian equations, we get

$$\begin{aligned} 2\ddot{\theta_1} + \ddot{\theta_2}\cos(\theta_1 - \theta_2) + \dot{\theta_2^2}\sin(\theta_1 - \theta_2) + 2\frac{g}{l}\sin\theta_1 = 0, \\ \ddot{\theta_2} + \ddot{\theta_1}\cos(\theta_1 - \theta_2) - \dot{\theta_2^2}\sin(\theta_1 - \theta_2) + \frac{g}{l}\sin\theta_2 = 0. \end{aligned} \tag{5}$$

We can convert them into first order by changing variables $\omega_1 = \dot{\theta_1}$, $\omega_2 = \dot{\theta_2}$. Plugging in these new variables and rearranging the equations, we get

$$\dot{\omega_1} = \frac{-\omega_1^2\sin(\theta_1 - \theta_2)\cos(\theta_1 - \theta_2) + \frac{g}{l}\sin\theta_2 cos(\theta_1 - \theta_2) - \omega_2^2\sin(\theta_1 - \theta_2) + 2\frac{g}{l}\sin(\theta_1)}{2 - \cos^2(\theta_1 - \theta_2)} \tag{6}$$

Simplify it with triangle identities

$$\begin{aligned} \sin(\theta_1 - \theta_2)\cos(\theta_1 - \theta_2) = \frac{1}{2}\sin(2\theta_1 - 2\theta_2) \\ \cos^2(\theta_1 - \theta_2) = \frac{\cos(2\theta_1 - 2\theta_2) + 1}{2} \end{aligned} \tag{7}$$

We can get:

$$\dot{\omega_1} = -\frac{\omega_1^2\sin(2\theta_1 - 2\theta_2) + \frac{g}{l}[3\sin\theta_1 + \sin(\theta_1 - 2\theta_2)] + 2\omega_2^2\sin(\theta_1 - \theta_2)}{3 - \cos(2\theta_1 - 2\theta_2)} \tag{8}$$

$$\dot{\omega_2} = \frac{4\omega_1^2\sin(\theta_1 - \theta_2) + 2\frac{g}{l}[-\sin\theta_2 + \sin(2\theta_1 - \theta_2)] + \omega_2^2\sin(2\theta_1 - 2\theta_2)}{3 - \cos(2\theta_1 - 2\theta_2)} \tag{9}$$

The energy of the system in terms of θ_1,θ_2,ω_1,and ω_2:

$$E = T + V = \frac{1}{2}mv_1^2 + \frac{1}{2}mv_2^2 = ml^2[\omega_1^2 + \frac{1}{2}\omega_2^2 + \omega_1\omega_2 cos(\theta_1 - \theta_2)] - mgl(2cos\theta_1 + \cos\theta_2) \tag{10}$$

3 Python simulation

To test our simulation, we will need an initial condition here. Starting from here, we will use the initial condition of the textbook of Newman, 8.15: $\theta_1 = \theta_2 = \frac{\pi}{2}, m = 1, l = 0.4$, which implies holding the two bobs at the horizontal line and let it go freely. We will adopt two methods: 4th order Runge-Kutta and Leap-frog methods to simulate this process.

3.1 Runge-Kutta Method

We use the 4th order Runge-Kutta method and have it output to a file named 'resultRK.txt' which contains the angle position and angle velocity information at every timestep. Using the initial conditions mentioned above, we draw the evolution of θ_1 and the trajectory of two bobs in the same picture.

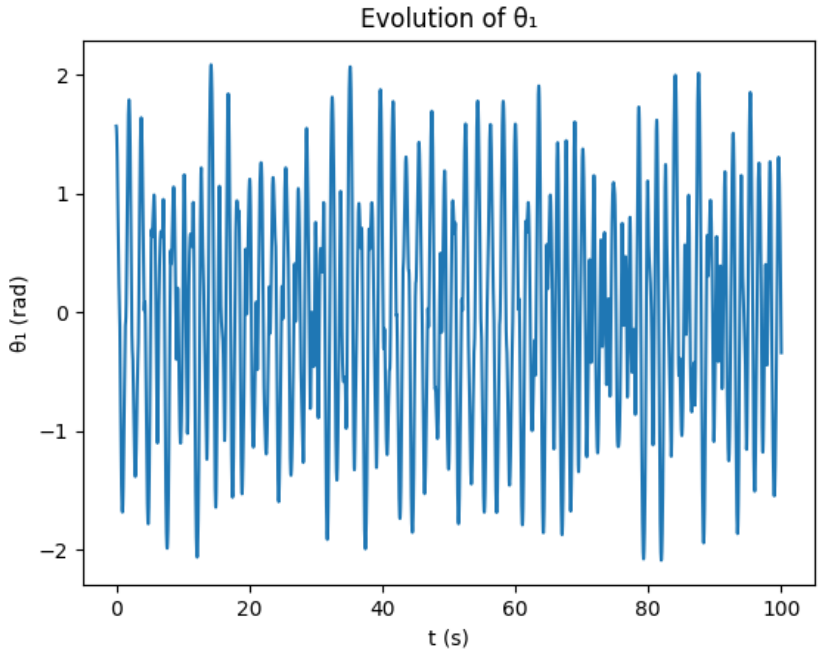

Figure 2: The evolution of two θ_1.

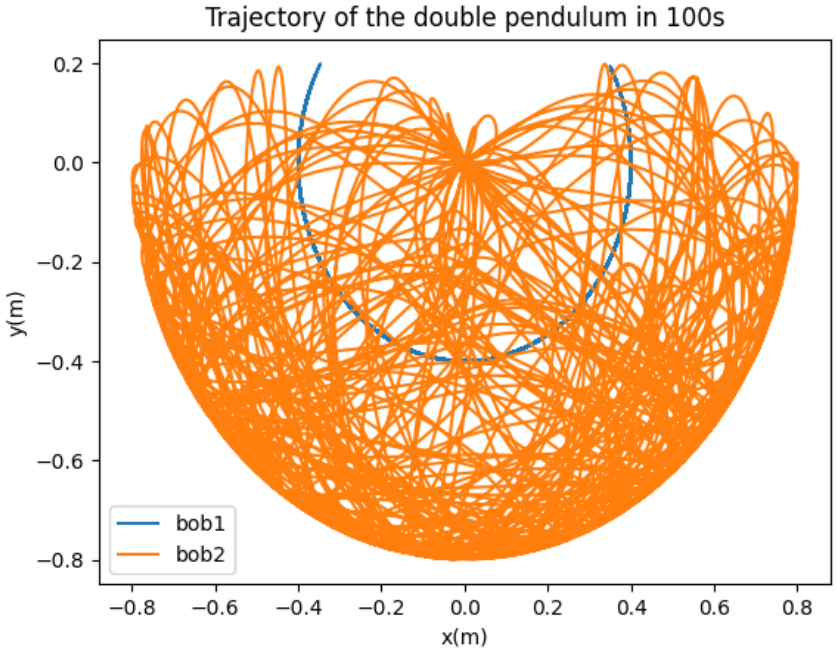

Figure 3: The trajectory of two bobs.

In order to see the process directly, we use matplotlib.animation to make an animation (pendulumRK.gif) in which we can see the position of the two bobs as the time evolves.

3.1.1 Convergence of time

To test the convergence of the integrator, we choose three different time step size: 0.002, 0.004, 0.008. The runga-kutta method is accurate to terms of order h^4 and carries an error of order h^5. The accuracy of the final calculation carries an error of order h^4. Therefore, the differences in the results of step size=0.008 and step size=0.004 should be 2^4 larger than the differences between step size=0.004 and step size=0.002. We test this in our code and get the answer that is closed to 16.

3.1.2 Conservation of Energy

To test the conservation of energy, the time step we use is 0.02 in which the variation of energy is no more than 10^{-5}. We draw a picture to show how the total energy changes with time. The initial energy is zero. We can see that the energy is decreasing instead of being exactly 0 all the time, which is because the Runga-Kutta method carries an error of order h^5.

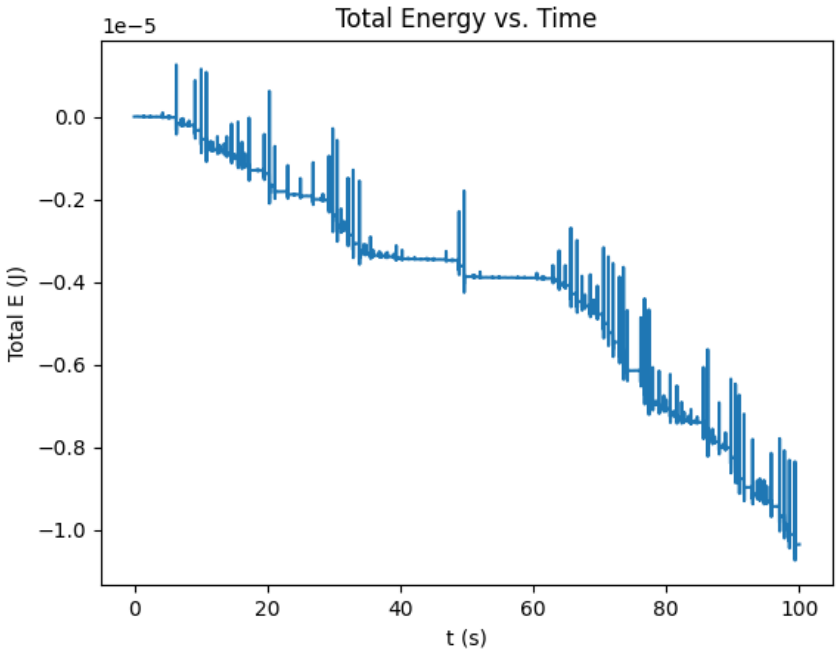

Figure 4: Total energy when using Runge-Kutta method.

3.2 Leapfrog Method

For the leapfrog method, we are getting some problems. We are getting very different plots for leapfrog method and RK4. We plot both the integer steps and half steps for leapfrog and find them diverge abruptly. It is also very strange that the results for using one euler step to get the former half step, i.e. using the 0th and $-\frac{1}{2}$th steps as our initial conditions, and using two euler steps to get the 1st and the $\frac{1}{2}$th steps as initial condisions are very different, which is not what we are expecting as these initial condisions should be almost the same. This may be something wrong with our coding, or leapfrog method is just so sensitive in this case as the integrands Eq. 8 and 9 are very complex.