Phys 304: Homework 7

Mary Smith*
Haverford College Department of Physics
(Dated: March 29, 2024)

[This homework took me about 5 hours to do. I learned about how to apply the 4th order Runge-Kutta method to solve differential equations. The most interesting problem was the strange attractors problem, because I had never seen anything like it before. The problem set was just right. I collaborated with Melanie Santiago on this problem set.]

1. PROBLEM 1: THE NONLINEAR PENDULUM

A standard physical concept is the linear pendulum, where the behavior of a pendulum can be approximated by a linear differential equation that can be solved exactly. However, real pendulums are nonlinear.

The case that will be examined here is a nonlinear pendulum with an arm of length l holding a bob of mass m, where the angle of displacement of the arm from the vertical is θ. This means that the acceleration of the mass is $ld^2\theta/dt^2$ in the tangential direction. The force on the mass is vertically downward with magnitude mg, where $g = 9.81ms^-2$ is the acceleration due to gravity. Friction is ignored and the arm is assumed to be massless. The component of this force in the tangential direction is $mg\sin(\theta)$, always toward the rest point at $\theta = 0$.

Thus, Newton's second law gives an equation of motion for the pendulum of the form

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin(\theta). \tag{1}$$

Equation 1 is nonlinear and is very difficult to solve analytically. Instead, the equation can be broken into two first-order equations. A new variable for the angular velocity of the pendulum ω can be defined as

$$\frac{d\theta}{dt} = \omega \tag{2}$$

This means that Equation 1 becomes

$$\frac{d\omega}{dt} = -\frac{g}{l}\sin(\theta). \tag{3}$$

The variables θ and ω are combined into a single vector $\mathbf{r} = (\theta, \omega)$, and the fourth order Runge-Kutta method can be applied in vector form to solve the two equations simultaneously.

The pseudocode for this problem can be found in Figure 1.

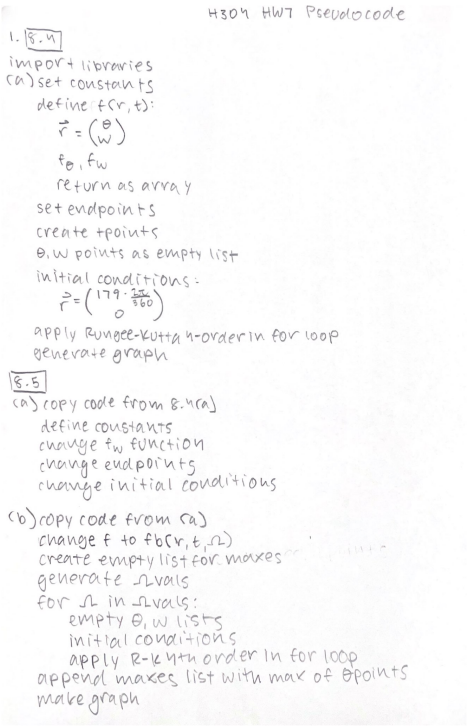

FIG. 1: [The pseudocode to calculate the motion of a non-driven and driven nonlinear pendulum.]

1.1. Exercise 8.4 part (a): Solving the first order equations

This problem was solved using Python code, according to Figure 1. The pendulum function was defined, for a vector $\mathbf{r} = (\theta, \omega)$ and time t. Empty lists were created for the values of θ and ω, and the initial conditions were set for $\mathbf{r} = (170 * \frac{2\pi}{360}, 0)$, given by the problem. Then, the fourth order Runge-Kutta method was applied to solve for θ and ω. These values were used to make a graph of θ as a function of time in Figure 2.

The plot in Figure 2 has some interesting features. There are some very well defined plateaus in the position of the pendulum. This is because when the bob is at higher angles from the vertical, the force on the bob is almost entirely in the vertical direction. There is very little force to pull the bob back to the $\theta = 0$ position, so

*Electronic address: masmith@haverford.edu

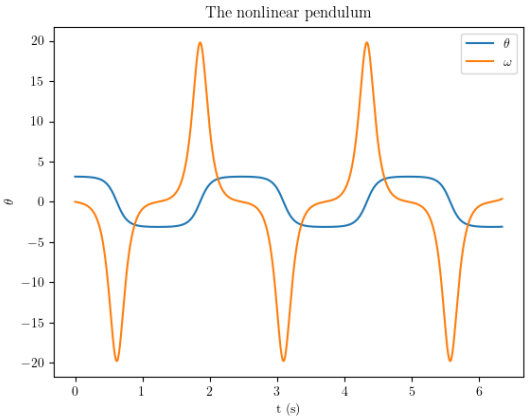

FIG. 2: [The position θ and angular velocity ω of the nonlinear pendulum.]

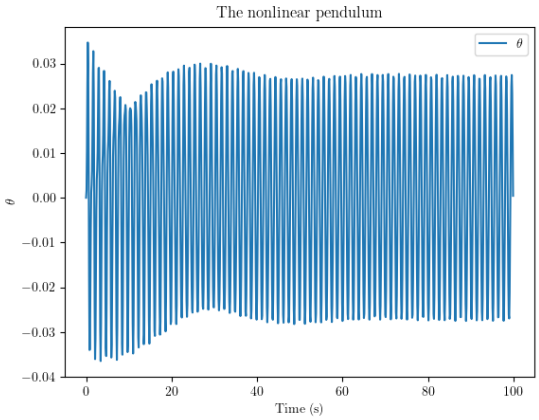

FIG. 3: [The position θ of the driven nonlinear pendulum as a function of time.]

it stalls for a short amount of time until the horizontal force can pull it back to center. The angular velocity curve has peaks at the times where there are the greatest increase or decrease of position, which makes sense.

1.2. Exercise 8.5: The driven pendulum

A pendulum like the one in Section 1.1 can be driven by, for example, exerting a small oscillating force horizontally on the mass. This makes the equation of motion for the pendulum

$$\frac{d^2\theta}{dt^2} == \frac{g}{l}\sin(\theta) + C\cos(\theta)\sin(\Omega t), \tag{4}$$

where C and Ω are constants.

1.2.1. Part (a): Position as a function of time

A program was written in Python to solve Equation 4 for θ as a function of time with $l = 10m$cm, $C = 2s^-2$, and $\Omega = 5s^-1$. This was then used to make a plot of θ as a function of time from $t = 0$ to $t = 100$s. The pendulum was started at rest with $\theta = 0$ and dθ/dt=0.

The pendulum function was defined, for a vector $\mathbf{r} = (\theta, \omega)$ and time t. Empty lists were created for the values of θ and ω, and the initial conditions were set for $\mathbf{r} = (0, 0)$, given by the problem. Then, the fourth order Runge-Kutta method was applied to solve for θ and ω. These values were used to make a graph of θ as a function of time for the driven pendulum in Figure 4.

1.2.2. Part (b): Finding the resonant frequency

By changing the value of Ω, while keeping C the same, a value for which the pendulum resonates with the driving force and swings widely from side to side was able to be found. This value is called the resonant frequency. A program was written in Python to find this frequency and generate a plot, according to the psuedocode in Figure 1.

A new function of $\mathbf{r}$, t, and Ω was created. Then, an empty list was created for the amplitudes of the position θ, and a list for the values of Ω was created and filled in an acceptable range. Then, for each Ω in the values list, the amplitude list was appended for the maximum position θ. The resonant frequency is then the value of Ω for the maximum amplitude, which was found to be ≈ 9.5095 Hz. A plot of driving frequency Ω versus amplitude was then created in Figure **??**

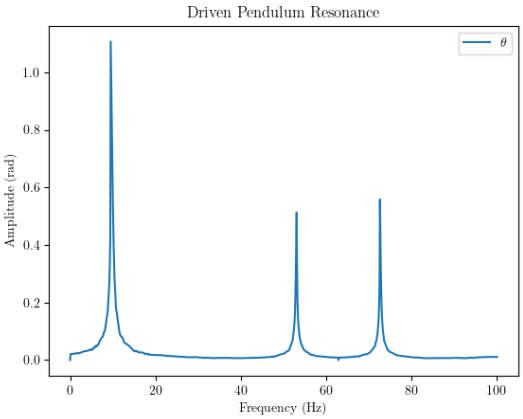

FIG. 4: [Amplitude versus driving frequency of the driven nonlinear pendulum. The resonant frequency is the value of the driving frequency at the maximum value of the amplitude.]

2. PROBLEM 2: STRANGE ATTRACTORS AND CHAOS

One of the most celebrated sets of differential equations in physics is the Lorenz equations:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz, \end{aligned} \tag{5}$$

where σ, r, and b are constants. These equations were first studied by Edward Lorenz, who derived them from a simplified model of weather patterns. They were one of the first incontrovertible examples of deterministic chaos, which is the occurrence of apparently random motion even though there is no randomness built into the equations.

The pseudocode for this problem can be found in Figure 5.

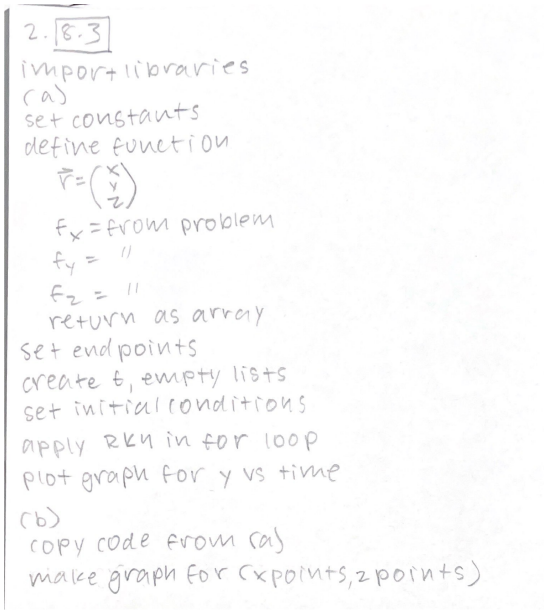

FIG. 5: [The pseudocode to calculate the Lorenz equations.]

2.1. Part (a): Solving for y

A Python program was written to solve the Lorenz equations, according to the pseudocode in Figure 5. A function $f(\mathbf{r}, t)$ was created, where $\mathbf{r} = (x, y, z)$. Empty lists were created, and then appended according to the Runge-Kutta 4th order method. This was done for the case where $\sigma = 10$, $r = 28$, and $b = \frac{8}{3}$ in the range from $t = 0$ to $t = 50$ with initial conditions $(x, y, z) = (0, 1, 0)$. This was then used to make a plot of y as a function of time in Figure 6.

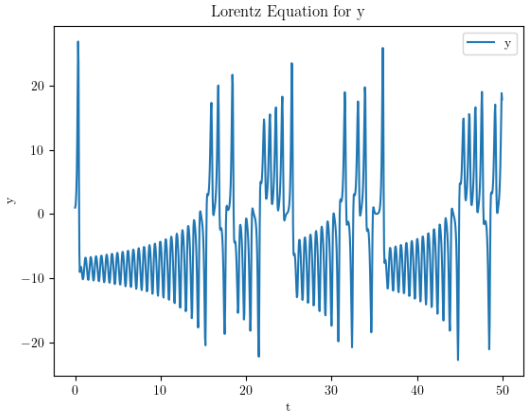

FIG. 6: [The y Lorenz equation in Equation 5, solved as a function of time for the initial conditions given. The time evolution of the graph is incredibly unpredictable.]

2.2. Part (b): Creating the strange attractor

The program in Section 2.1 was modified to produce a plot of z against x, according to the pseudocode in Figure 5. This graph in Figure 7 depicts this "strange attractor" of the Lorenz equations.

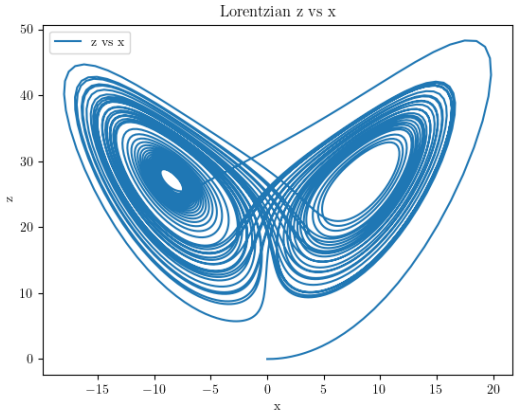

FIG. 7: [A graph of z against x for the Lorenz equations in Equation 5. This is the famous "strange attractor" of the Lorenz equations, a lop-sided butterfly-shaped plot that never repeats itself.]