

Final Project: Chaos!

The non-linear driven pendulum

Background

You are probably aware that many physical systems and types of natural phenomena can exhibit the highly complex, unpredictable behavior known as ‘chaos’. Examples of inherently chaotic phenomena include the weather (no surprise there!), population dynamics, economic systems, fluid flow, the motions of various solar system bodies.

Most of these examples are complex, highly non-linear systems, so you may not be too surprised that they are chaotic. However, even relatively simple physical systems can exhibit such behavior. A loose definition of a chaotic system is one that is both deterministic (i.e., can be described by precise mathematical formula) and unpredictable in the sense that no matter how precisely one specifies the initial conditions, the remaining uncertainty ensures that the future state of the system cannot be predicted. (This is often summarized by saying that the system shows extreme sensitivity to initial conditions.) The aim of this project is to simulate the motion of a simple physical system – a driven pendulum subject to damping, in order to demonstrate that it is possible for a system to be both deterministic and chaotic.

You should be familiar with the simple pendulum – ubiquitous in introductory Physics textbooks as an example of simple harmonic motion. The textbook simple pendulum differs from real life in at least two important ways: (i) the only forces acting are gravity and tension in the string (damping forces such as friction in the bearing, air resistance etc. are ignored); (ii) the motion of the pendulum is restricted to small angles, such that $\sin\theta \approx \theta$. Assuming also that the string is massless, the equation of motion is

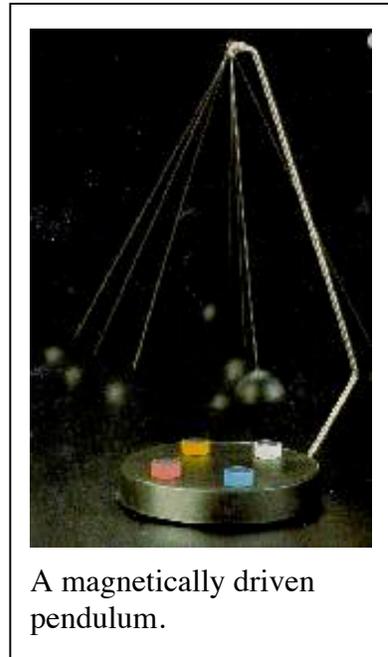
$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\theta$$

where L is the length of the string and θ is the angle that the string makes with the vertical. It is easy to verify that this equation has the general solution

$$\theta = \theta_0 \sin(\Omega_s t + \phi)$$

where the angular frequency of the oscillation for the simple pendulum is $\Omega_s = \sqrt{g/L}$, and θ_0 and ϕ are constants that depend on the initial conditions (initial displacement and velocity of the pendulum). According to this solution, the pendulum oscillates sinusoidally and the oscillations continue forever without decaying (there’s no damping force).

It’s a matter of experience that real pendulums don’t behave this way. To make a more realistic simulation, we at least need to include a **damping force**. For example, the drag from



A magnetically driven pendulum.

air resistance at low speeds is proportional to the speed. Since the speed of the pendulum is $v=L\omega$ where $\omega \equiv d\theta/dt$ is the angular velocity, we can write the damping force, F_d , due to air resistance as follows:

$$F_d = -bL\omega$$

where b is a damping constant. The inclusion of damping means that the oscillations will decay exponentially, so in order to keep our pendulum swinging, we also need to apply a driving force. For example, a force, F_D , varying sinusoidally with time of amplitude A and angular frequency Ω_D :

$$F_D = AL\sin(\Omega_D t)$$

For our realistic pendulum, we will also want to eliminate the assumption that the motion is restricted to small angles. To keep things relatively simple, we'll assume that the "string" is not only massless but also rigid (think "rod" rather than "string"), so that the tension force exactly balances the radial component of the gravitational force (parallel to the "string"), even if this requires a "negative" tension. We can now use Newton's 2nd Law to obtain an equation of motion for the angular displacement of our (sort of) realistic pendulum.

The program

Design, write and test a PYTHON function to compute and plot the trajectory and angular velocity of driven pendulum subject damping, as functions of time [i.e., $\theta(t)$ and $\omega(t)$]. The first thing you will need to do is formulate the equations of motion for both θ and ω . You are advised to show your equations to your instructor before you start work on the program itself.

A good starting point is the `baseball` program of Project 5. However, you should use the **Euler-Cromer** method (see "Lecture 8": Simulating Orbital Motion on myCourses), rather than the Euler method. The values of L , $b'=b/m$, $A'=A/m$, and Ω_D should be passed as input arguments. You will also need to specify initial values of θ and ω .

It would be a good idea to break program development up into several stages.

- Start with the simple pendulum (no drag; no driving force). You should check your program by comparing your results with the well-known analytical solution.
- Next add the damping force. Again there is an analytical solution to compare with (look it up!)
- Finally, include the driving force. Test the program with different values of A . Do the results seem sensible as you gradually increase A ?

Your program should

- Compute θ and ω from $t = 0 \rightarrow t_{end}$, where t_{end} is a user-specified end time with a timestep Δt that is also specified by the user.
- Produce plots of θ versus t , ω versus t , and θ versus ω .

Programming issues: you will need to use arrays to record the values of t , θ and ω at each timestep, for plotting. Also note that the pendulum is allowed to swing all the way around its

pivot point, so $|\theta| > \pi$ is possible. However, it may be more illuminating to make plots where you “reset” θ (i.e., add or subtract 2π) so that it remains in the range $|\theta| < \pi$.

Investigation

There are four parameters to adjust and therefore many different experiments you can try. Probably the most interesting parameters to play with are A' and b' , so at least to start with, keep L and Ω_D fixed. A useful trick is to set $L = 9.81$ m, which makes the angular frequency of the equivalent simple pendulum $\Omega_S = 1$ rad/s. Its period is then $T_S = 2\pi/\Omega_S = 2\pi$. As a value for the angular frequency of the driving force, try $\Omega_D = 2/3 \Omega_S$. Choose sensible values for the initial conditions (θ and ω at $t = 0$). Here are some suggestions for investigation.

First, just explore the effect of damping by running simulations with different values of the damping constant but no driving force ($A' = 0$). You may need to start with a reasonably large initial value of θ (say 0.5). Estimate the time constant for the decay of the oscillations. Is it consistent with the approximate analytical theory?

Fix the damping constant at $b' = 0.5$ and investigate the effect of increasing the driving force. For example, try $A' = 0.1, 0.5, 0.99, 1.2$. What happens to the oscillations? Pay particular attention to the trajectory in the phase-space plot (ω vs θ). Can you interpret your results? [In these simulations, it will be useful to have an option to begin your plots only once transients due to the initial conditions have died away and the pendulum has settled down into its **steady-state behavior**, i.e., you will want to run the simulation for many periods of the driving force before beginning to plot. This is most easily accomplished with a new user-specified parameter t_{skip} that specifies how long to run the simulation before starting to plot.]

Run simulations for identical pendulums (i.e., L, b', A' and Ω_D are the same) but slightly different initial values of θ . For example, try $\Delta\theta = 0.000001$ rad. What happens when the driving force is weak (e.g., $A' = 0.5$) and strong (e.g., $A' = 1.2$)? Make a plot of $|\Delta\theta|$ vs t for these two cases (being sure that your method of determining $\Delta\theta$ preserves the full numerical precision available in MATLAB); it is probably most enlightening to use a logarithmic scale for $|\Delta\theta|$. What are the differences that you see for these two cases and what does this say about the sensitivity to the initial conditions? Why do you think it might be difficult for weather prediction models to accurately forecast the weather more than a week or so into the future?

Modify your program to construct Poincaré sections. This plots ω vs θ but only for times that are in phase with the driving force. That is, only plot points at times for which $\Omega_D t = n\pi$, where n is an integer, $n = 0, 1, 2, 3, \dots$ (note that in practice, you will have to find the nearest timestep or, better yet, interpolate between the nearest timesteps). Construct Poincaré sections for the **steady-state behavior** in cases where the driving force is weak and strong. What differences do you see? Can you explain them?

With the other parameters the same as before, look at the **steady-state behavior** for $A' = 1.4, 1.44, 1.465, 1.476, \text{ and } 1.480$. This sequence is an example of what is called a “period-doubling route to chaos”. Can you see why?

Resources

Young, H. D. & Freedman, R. A., *University Physics*, 13th ed., Pearson (2012), Chapter 14 especially 14.5 – 14.8. You will find similar discussions of simple harmonic motion, pendulums, damping, and forced oscillations in most other university physics textbooks.

<http://hyperphysics.phy-astr.gsu.edu/hbase/pend.html#c1> (The basic physics of the pendulum.)

http://www.robert-doerner.de/en/driven_pendulum/driven_pendulum.html (a discussion with animations of the driven pendulum)

<http://order.ph.utexas.edu/educational.html> (an introduction to Chaos with more Physics applications)

<http://hypertextbook.com/chaos/> (...and a more mathematical treatment.)