Lesson 18: Driven Damped Pendulum and RK4

The driven, damped pendulum is a relatively simple chaotic system that exhibits a range of interesting dynamical behaviors. There is no analytical solution for this system; it can only be investigated numerically or experimentally.

1 Equations of motion

Consider a plane pendulum with length ℓ and mass m, as shown in Figure 1. The angle

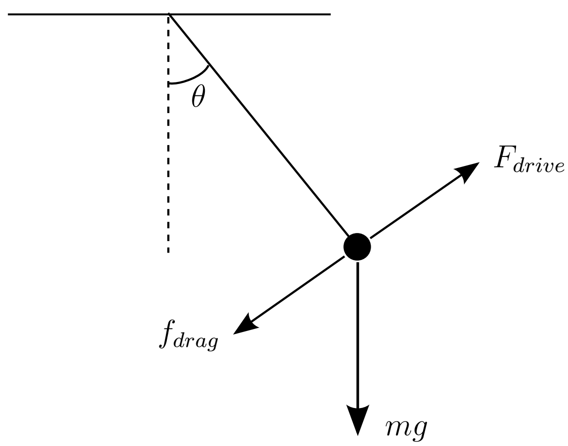

Figure 1: A pendulum consisting of a mass m suspended from a cord of length ℓ. The pendulum makes an angle θ with the vertical.

between the pendulum and the vertical is θ. The component of the gravitational force in the direction perpendicular to the cord is $-mg\sin\theta$. The pendulum is subject to a linear drag force $f_{drag} = -2m\ell\beta\,\dot{\theta}$ that opposes the motion, and an external driving force $F_{drive} = mg\gamma\,\cos(\omega t)$ that depends on time. (Here, β, γ, and ω are constants. The factors of $2m\ell$ and mg are included for later convenience.) Newton's second law equates the sum of forces to the product of mass m and acceleration. The component of Newton's second law in the direction perpendicular to the cord is

$$-mg\sin\theta - 2m\ell\beta\,\dot{\theta} + mg\gamma\,\cos(\omega t) = m\ell\ddot{\theta} \tag{1}$$

This simplifies to

$$\ddot{\theta} = -\omega_0^2\sin\theta - 2\beta\,\dot{\theta} + \gamma\omega_0^2\cos(\omega t) \tag{2}$$

where $\omega_0 \equiv \sqrt{g/\ell}$ is the "natural frequency".[1] Equation (2) is the equation of motion for the driven, damped pendulum. The degree of damping is controlled by β. The constants γ and ω determine the amplitude and frequency of the driving force. The natural period for the pendulum is $T_0 = 2\pi/\omega_0$. The drive period is $T = 2\pi/\omega$.

2 What are we looking for?

Pull the pendulum aside, to some initial angle $\theta(0)$. Now release it with some initial angular velocity $\dot{\theta}(0)$, and the pendulum will move. If $\gamma = 0$, so the driving force is turned off, then the damping will cause the pendulum to slow down and eventually come to rest at $\theta = 0$. This will happen for any initial conditions $\theta(0)$ and $\dot{\theta}(0)$. If $\gamma \neq 0$, we would expect that, in the long run, the pendulum's motion will be dictated by the driving force. That is, initially the pendulum's motion will depend on $\theta(0)$ and $\dot{\theta}(0)$, but in the long run the pendulum's motion will be independent of the initial conditions. It will move in some definite manner determined entirely by the driving force. The exact details of this long–term motion will depend on the parameter γ that controls the magnitude of the driving force. Our goal is to investigate the long–term motion of the pendulum for various values of γ.

3 Numerical implementation

The differential equation (2) can be written as two first–order equations,

$$\dot{\theta} = \Psi \tag{3a}$$
$$\dot{\Psi} = -\omega_0^2 \sin\theta - 2\beta\Psi + \gamma\omega_0^2 \cos(\omega t) \tag{3b}$$

Exercise 1: Write a code using the second order Runge–Kutta method to solve the system of equations for the driven, damped pendulum. Set $\omega = 2\pi$ so that the drive period is $T = 1$. Also set $\omega_0 = 1.5\omega$ and $\beta = \omega_0/4$. Show that different values of the drive amplitude γ lead to qualitatively different types of behavior, as in Figure 2.

In previous lessons, we checked our codes by comparing the numerical results to exact, analytical results. For example, in Lesson 9, you wrote an RK2 code for the radioactive decay problem and checked that the errors were proportional to Δt^2. If you had found that the errors were not proportional to Δt^2, then you would have known that your code contained a bug.

This type of test of a numerical code is called a *convergence test*. It not only serves to help identify bugs, but it also helps the user to understand the resolution required for a given level of accuracy.

We would like to perform a convergence test for the driven, damped pendulum. However, in this case, we can't compute the error since we don't have an analytical solution. It turns out that we don't need to know the exact answer to show how the error depends on resolution.

[1]The natural frequency is the frequency of small oscillations without any driving force or significant damping.

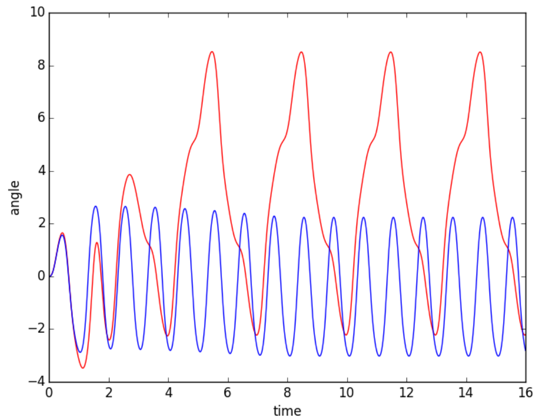

Figure 2: Example behavior for the driven, damped pendulum. These curves use the same initial conditions, but different values of the driving amplitude γ.

Here is the idea. Run a simulation to some final time T_{final}, using a timestep Δt, and let $\theta_{\Delta t}$ denote the final angle. Run the simulation two more times, to the same final time T_{final}, using timesteps $\Delta t/2$ and $\Delta t/4$. Denote the final angles by $\theta_{\Delta t/2}$ and $\theta_{\Delta t/4}$. Now, let's assume that the leading order term in the error is proportional to Δt^n. Then

$$\theta_{\Delta t} = \theta_{exact} + A\Delta t^n \tag{4a}$$
$$\theta_{\Delta t/2} = \theta_{exact} + A(\Delta t/2)^n \tag{4b}$$
$$\theta_{\Delta t/4} = \theta_{exact} + A(\Delta t/4)^n \tag{4c}$$

where θ_{exact} is the (unknown) exact answer and A is a constant. From these equations it is easy to show that

$$\frac{\theta_{\Delta t} - \theta_{\Delta t/2}}{\theta_{\Delta t/2} - \theta_{\Delta t/4}} = 2^n \tag{5}$$

For RK2, the errors should be proportional to Δt^2. Thus, a calculation of the expression $(\theta_{\Delta t} - \theta_{\Delta t/2})/(\theta_{\Delta t/2} - \theta_{\Delta t/4})$ should yield $2^2 = 4$.

The result will not be exactly equal to 4 because Eqs. (4) are not exact. Recall that higher order terms have been dropped from these equations. Thus, you may need to run the simulation at many different resolutions, producing results $\theta_{\Delta t}$, $\theta_{\Delta t/2}$, $\theta_{\Delta t/4}$, $\theta_{\Delta t/8}$, *etc.* The convergence calculation (5) can be carried out for each set of three successive θ values. This should show that, in the limit of high resolution, the value approaches 4.

Exercise 2: Carry out a convergence test for your code, using $\gamma = 0.8$ and a total run time of $20T$, where $T = 1$ is the drive period. Show that the calculation (5) approaches 4. About how many timesteps do you need for accuracy to, say, three decimal places? Four decimal places?

4 Fourth order Runge–Kutta

We want to investigate the long–term behavior of the driven damped pendulum, and we need the results to be highly accurate. It is difficult to acheive the desired level of accuracy with RK2. (Said another way, with RK2 we would need a very small Δt, and a very long run time.) It is worth the extra effort to implement a higher–order integration scheme, like fourth–order Runge–Kutta (RK4).

Let $u(t)$ denote the vector of dependent variables, with t the dependent variable, and write the system of equations (3) as $\dot{u} = f(u,t)$. That is,

$$u(t) \equiv \begin{pmatrix} \theta(t) \\ \Psi(t) \end{pmatrix} \tag{6}$$

and $f(u,t)$ is the vector of functions of θ and Ψ given by

$$f(u,t) \equiv \begin{pmatrix} \Psi \\ -\omega_0^2 \sin\theta - 2\beta\Psi + \gamma\omega_0^2\cos(\omega t) \end{pmatrix} \tag{7}$$

Now let $u_i \equiv u(t_i)$ denote u at time t_i, and let $u_{i+1} \equiv u(t_i + \Delta t)$ denote u at time $t + \Delta t$. Here is the RK4 scheme:

$$u_a = u_i + f(u_i, t_i)\Delta t/2 \tag{8a}$$
$$u_b = u_i + f(u_a, t_h)\Delta t/2 \tag{8b}$$
$$u_c = u_i + f(u_b, t_h)\Delta t \tag{8c}$$
$$u_d = u_i + f(u_c, t_{i+1})\Delta t \tag{8d}$$
$$u_{i+1} = \frac{1}{3}(u_a + 2u_b + u_c + u_d/2) - \frac{1}{2}u_i \tag{8e}$$

where $t_h \equiv t_i + \Delta t/2$ and $t_{i+1} \equiv t_i + \Delta t$. Although it is far from obvious, RK4 is a *fourth–order* scheme; that is, the errors are proportional to Δt^4.

Homework: Write a code to solve the equations for the driven damped pendulum using RK4. Carry out a convergence test for your code, using $\gamma = 0.8$ and a total run time of $20T$, where $T = 1$ is the drive period. Show the results of your test. Did you get the expected result? About how many timesteps do you need for accuracy to, say, four decimal places? Five decimal places?

You should realize that the number of timesteps needed for a given level of accuracy depends *strongly* on both the total run time and the values of the parameters like γ. When you change the run time, or change γ, there is really no way to predict how many timesteps will be needed for a given level of accuracy. Said another way, you cannot tell from a single simulation how accurate the results are. As a general rule, you need to run each simulation two or more times, using different resolutions, before you can begin to judge the accuracy of the results.