

## 8. 780.20 Session 8

### a. Brief Gnuplot Stuff

#### a.1 A Common Error with Continuation Lines

If you have a gnuplot plot file `my_plotfile.plt` and you get an error message like

```
"my_plotfile.plt", line 27: invalid character
```

when you try to load it, the most likely problem is that there is at least one blank space after a continuation character `\`. That `\` has to be the absolute last thing on a line or you'll get this error. Check it out!

#### a.2 Multiple Plot Windows

In gnuplot under Linux, the “terminal type” corresponding to the screen is called “x11” (for the X11 windowing system). You can have more than one plot window open at a time, however, by labeling them 1, 2, 3, and so on. For example, to plot  $\sin(x)$  in one window while plotting columns 1 and 2 from the file `data.dat` in another:

```
gnuplot> set term x11 1
gnuplot> plot sin(x)
gnuplot> set term x11 2
gnuplot> plot "data.dat" using 1:2
```

By default, the windows will have titles like “Gnuplot 1” and “Gnuplot 2” on their title bars. To get window 1 to have the title “Time Dependence”, use:

```
gnuplot> set term x11 1 title "Time Dependence"
gnuplot> plot sin(x)
```

You can close a windowing by first selecting it and then typing the letter `q`. (If you exit gnuplot entirely, all windows will close.) None of this works on Windows.

#### a.3 Piping to Gnuplot

In the `diffeq_pendulum.cpp` program, the output is sent to gnuplot more-or-less directly. The actual execution is not very elegant and may not work perfectly all the time. The C++ class in `GnuplotPipe.cpp` was adapted from a code found on the web. To see how to use it, look at the `GnuplotPipe.h` header file. There are many improvements possible!

## b. Follow-up to Session 7: Damping

Let's review the harmonic oscillator (the  $p = 2$  case from Session 6) and the mathematical conditions for underdamped, critically damped, and overdamped motion. At the same time, you should look at the handout showing phase-space plots for these cases.

We start with the differential equation for an undriven harmonic oscillator with a damping force proportional to the velocity:

$$F = Ma \implies \frac{d^2x}{dt^2} = -\omega_0^2 x - \frac{b}{M} \frac{dx}{dt}, \quad (8.1)$$

or

$$\frac{d^2x}{dt^2} + \frac{b}{M} \frac{dx}{dt} + \omega_0^2 x = 0. \quad (8.2)$$

Since this is a *homogeneous* equation, we look for solutions of the form  $x(t) = Ae^{i\omega t}$ , where  $\omega$  could be positive or negative, real or complex. Substitute  $x(t)$  into Eq. (8.2):

$$(i\omega)^2 Ae^{i\omega t} + \frac{b}{M}(i\omega)Ae^{i\omega t} + \omega_0^2 Ae^{i\omega t} = 0. \quad (8.3)$$

The factor  $Ae^{i\omega t}$  never vanishes, so we can divide it out, leaving

$$\omega^2 - \frac{ib}{M}\omega - \omega_0^2 = 0. \quad (8.4)$$

In general, this quadratic equation has two (complex) solutions, which will lead to the two independent solutions we expect for a 2nd-order differential equation. We'll denote the solutions  $\omega_+$  and  $\omega_-$ , where

$$\omega_{\pm} = \frac{\frac{ib}{M} \pm \sqrt{-\frac{b^2}{M^2} + 4\omega_0^2}}{2} = \frac{ib}{2M} \pm \sqrt{\omega_0^2 - \left(\frac{b}{2M}\right)^2}. \quad (8.5)$$

Now let's consider cases. For simplicity, we'll work with units in which  $M = 1$  and the spring constant  $k = 1$ , which means  $\omega_0 = \sqrt{k/M} = 1$ . For initial conditions, let's take  $x_0 = 0$ ,  $v_0 = 1$  (so if we're thinking of a spring, it starts at the equilibrium position with a nonzero speed).

1.  $b = 0 \implies$  "undamped"

From Eq. (8.5),  $\omega_{\pm} = \pm\omega_0$ . A general solution is therefore

$$x(t) = Ae^{i\omega_0 t} + Be^{-i\omega_0 t} \quad (8.6)$$

and the initial conditions  $x_0 = 0$  and  $v_0 = 1$  yield

$$A + B = 0, \quad i\omega_0(A - B) = 1, \quad (8.7)$$

or

$$A = \frac{1}{2i} \frac{1}{\omega_0}, \quad B = -\frac{1}{2i} \frac{1}{\omega_0}. \quad (8.8)$$

Recalling that

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i} \quad \text{and} \quad \cos x = \frac{e^{ix} + e^{-ix}}{2}, \quad (8.9)$$

we obtain the solutions

$$x(t) = \frac{1}{\omega_0} \sin(\omega_0 t), \quad v(t) = \cos(\omega_0 t). \quad (8.10)$$

The properties of trig functions tell us immediately that the phase-space plot of  $v$  vs.  $x$  will have the equation

$$\omega_0^2 x^2 + v^2 = 1, \quad (8.11)$$

which is an ellipse.

2.  $\frac{b}{2M} < \omega_0 \implies$  “underdamped”

Now we have a real and an imaginary part to  $e^{i\omega t}$ . The general solution is:

$$x(t) = e^{-\frac{b}{2M}t} \left( A e^{i\sqrt{\omega_0^2 - (b/2M)^2}t} + B e^{-i\sqrt{\omega_0^2 - (b/2M)^2}t} \right). \quad (8.12)$$

The decaying exponential on the outside will form an *envelope*  $\pm e^{-\frac{b}{2M}t}$  about the oscillations. We can read off the oscillation frequency (if the oscillations die slowly) as

$$\omega = \sqrt{\omega_0^2 - (b/2M)^2}. \quad (8.13)$$

The phase-space trajectory is a *spiral* into a *fixed point* at the origin (that is, it eventually stops at the equilibrium point!). What would a plot of the time dependence look like? (Predict and then look at the corresponding plot on the handout.)

3.  $\frac{b}{2M} = \omega_0 \implies$  “critically damped”

The solution for  $x(t)$  is a degenerate form now, which does not look like the previous forms since the oscillating parts vanish. Rather, we have a general solution

$$x(t) = (A + Bt)e^{-\frac{b}{2M}t}. \quad (8.14)$$

For the initial conditions  $x_0 = 0$  and  $v_0 = 1$ , we find

$$x(t) = te^{-\frac{b}{2M}t}, \quad v(t) = e^{-\frac{b}{2M}t} \left( 1 - \frac{b}{2M}t \right). \quad (8.15)$$

(See the plots to help visualize these.) Note that  $x(t)$  is always positive but  $v(t)$  goes negative. What would the plots look like for initial conditions  $x_0 = 1$  and  $v_0 = 0$ ?

4.  $\frac{b}{2M} > \omega_0 \implies$  “overdamped”

Now we find that  $\omega$  is purely imaginary, which implies exponential decay. The general solution is

$$x(t) = e^{-\frac{b}{2M}t} \left( A e^{\sqrt{\omega_0^2 - (b/2M)^2}t} + B e^{-\sqrt{\omega_0^2 - (b/2M)^2}t} \right). \quad (8.16)$$

Except for unusual values of  $b$ , the “ $B$ ” term dies much faster, while the “ $A$ ” term dies off slowly once  $B$  is no longer relevant because of cancellation between the exponentials. In that region,  $v(t) \propto x(t)$  so the phase-space trajectory is close to a straight line. Is it possible for  $v(t)$  to go to zero without turning negative?

Try to reproduce these different regimes in real time using `diffeq_pendulum.cpp`. When we now add a *driving* force

$$\frac{f_{\text{ext}}}{M} \cos(\omega_{\text{ext}} t) \quad (8.17)$$

(we've chosen the phase to be zero at  $t = 0$ ), then the above solutions to the *homogeneous* equation ("transients") are added to a specific solution to get the general solution. The transients die out (hence the name!), leaving the specific solution in the long-time limit.

### c. C++ Strings and Things

When we have names or a phrase or a sentence in C, we store it as a *char* type, or actually a pointer to an array of a fixed length. For example, the declaration

```
char filename[30];
```

In C++, we can use C-style string and associated commands (such as `sprintf` or `strcmp`) or we can use C++ strings. I recommend C++ strings for several reasons:

- they can be any length, and adjust automatically (i.e., we do *not* declare the length);
- they work with `==` and `+` as you would hope and expect;
- they are overall safer.

Here's a quick guide to the use of C++ strings, with the goal of constructing filenames. The `filename_test.cpp` program in Session 8 illustrates the summary given here.

Action	How to do it
include file	<code>#include &lt;string&gt;</code>
declaring	<code>string my_name;</code> <code>string my_name = "Furnstahl";</code>
assigning	<code>my_name = "Smith";</code>
comparing	<code>if (my_name == "Furnstahl") { (do something) }</code>
joining	<code>string my_full_name;</code> <code>first_name = "Dick", last_name = "Furnstahl";</code> <code>my_full_name = first_name + last_name;</code>
printing	<code>cout &lt;&lt; "My name is " &lt;&lt; my_full_name &lt;&lt; endl;</code>
converting to char	<code>my_full_name.c_str();</code>

Suppose we want to open a file with the name stored in the string `my_full_name`. Here's how:

```
ofstream my_file; // declare the output file stream
file.open(my_full_name.c_str()); // notice the use of c.str()
```

If we tried `file.open(my_full_name)`; we would get *many* errors. Now suppose we want to create a filename with the current value of the integer `i` as part of the name. We do this with an "output string stream" or `ostringstream`. Here's what it looks like:

```
int i = 5;    // define our integer
ostream my_filename_stream;    // declare the output string stream
my_filename_stream << "output_" << i;    // just like using cout
string my_filename = my_filename_stream.str(); // convert stream to a string
file.open (my_filename.c_str()); // open the file, again using c.str()
```

You could also skip defining the string `my_filename` and open the file using:

```
file.open (my_filename_stream.str().c_str());
```

#### d. Damped Driven Pendulum

There is an apt quote from physicist/mathematician Stanislaus Ulam about nonlinear phenomena:

“Calling the subject nonlinear dynamics is like calling zoology ‘non-elephant studies’.”

Indeed, most macroscopic real-world phenomena is *nonlinear*, unlike the impression you might receive from typical physics curricula, which emphasize linear problems.

In Session 8, we consider another simple yet rich example of nonlinear dynamics: the damped driven pendulum. We have in mind a physical pendulum with moment of inertia  $I$  for which the dependent variable is the angle  $\theta$ . It is subject to gravity, a damping force proportional to  $d\theta/dt \equiv \dot{\theta}$ , and an external periodic driving torque. Newton’s second law for the torque can be rearranged to give the differential equation:

$$\frac{d^2\theta}{dt^2} + \alpha \frac{d\theta}{dt} + \omega_0^2 \sin \theta = f_{\text{ext}} \cos(\omega_{\text{ext}} t) . \quad (8.18)$$

The presence of  $\sin \theta$  rather than  $\theta$  makes this problem inherently nonlinear.

To study this system, we’ll look at plots of  $\theta$  versus  $t$  and the phase-space plot  $\dot{\theta}$  vs.  $\theta$ . Under what conditions will these look like the results for the  $p = 2$  harmonic oscillator from Session 7? Try to decide what range of  $\theta_0$  gives harmonic behavior, given  $\dot{\theta}_0 = 0$ .

Let’s think about *chaos*. Consider the following brief discussion as a teaser. Here are some characteristics of chaos:

- The system does not repeat past behavior (cf. periodic behavior).
- An uncertainty (or variation) in initial conditions grows *exponentially* (rather than linearly) in time. The consequence is that the system is deterministic (as opposed to having a random component) but not predictable, since there is a finite precision in specifying the initial conditions (e.g., think about round-off error!).
- The system has a distributed power spectrum (more next time!).

The following are necessary conditions for chaotic behavior:

- a) The system has at least *three* independent dynamical variables. That is, the system can be written as

$$\frac{dy_0}{dt} = F_0(y_0, \dots, y_n), \quad (8.19)$$

$$\frac{dy_1}{dt} = F_1(y_0, \dots, y_n), \quad (8.20)$$

$$\vdots \quad (8.21)$$

$$\frac{dy_n}{dt} = F_n(y_0, \dots, y_n), \quad (8.22)$$

$$(8.23)$$

with  $n \geq 3$ .

- b) The equations of motion contain nonlinear term(s) that couple several of the variables.

You might not think that our pendulum example qualifies, since there only seem to be two independent dynamical variables,  $\theta$  and  $\omega \equiv \dot{\theta}$ . We find the third by introducing  $\phi$  as

$$\phi = \omega_{\text{ext}} t \quad \Longrightarrow \quad \frac{d\phi}{dt} = \omega_{\text{ext}}. \quad (8.24)$$

Thus, the three necessary equations are

$$\frac{d\theta}{dt} = \omega, \quad (8.25)$$

$$\frac{d\omega}{dt} = -\alpha\omega - \omega_0^2 \sin \theta - f_{\text{ext}} \cos \phi, \quad (8.26)$$

$$\frac{d\phi}{dt} = \omega_{\text{ext}}. \quad (8.27)$$

Now we satisfy a) with  $\theta$ ,  $\omega$ , and  $\phi$ , and we satisfy b) since the  $\sin \theta$  and  $\cos \phi$  terms couple the equations nonlinearly. So we should be able to find chaos!

## e. References

- [1] R.H. Landau and M.J. Paez, *Computational Physics: Problem Solving with Computers* (Wiley-Interscience, 1997).
- [2] M. Hjorth-Jensen, *Computational Physics*. These are notes from a course offered at the University of Oslo. See the 780.20 webpage for links.
- [3] W. Press *et al.*, *Numerical Recipes in C* (Cambridge, 1992). Individual chapters are available online from <http://lib-www.lanl.gov/numerical/bookcpdf.html>. There are also versions for Fortran and C++.