Physics 331 Advanced Classical Mechanics Fall 2011

Problem B

We will make a bifurcation diagram for the driven damped pendulum, I had originally intended to make a template, but I think it’ll be more fun if you program your own. Here’s the steps:

1. Define \( \text{eqn} = \) the differential equation, with \( \omega = 2\pi, \omega_0 = 3\pi, \) and \( \beta = 3\pi/4. \) Be sure to leave \( \gamma \) as an unspecified variable.

2. We want to grow a list of \( (\gamma, \phi) \) pairs, so we need to start with an empty list. Type

   \[
   \text{list} = \{\}
   \]

3. Now we want to set up a for-loop to go from \( \gamma = 1.06 \) to \( \gamma = 1.087 \) in steps of size 0.0001. As an example, the for-loop syntax to take \( x \) from 1 to 10 in steps of 0.1 would be:

   \[
   \text{For}[x=1, x<=10, x+=0.1, \text{expressions}]
   \]

   Here \( \text{expressions} \) is as many mathematica expressions as you want, separated by semicolons (and carriage returns usually, for readability).

4. In the \( \text{expressions} \) part of the for-loop you will have three expressions. The first is the numerical solution command. Initially, just take the time range to be 0 to 20 until you have everything working. You need to set step limit to infinity, via

   \[
   \text{soln} = \text{NDSolve}[ ... \{t, 0, 20\}, \text{MaxSteps} \rightarrow \text{Infinity}].
   \]

5. The second expression does the strobe light: it extracts a set of \( \phi \) values at the driving period, using the \text{Table} function. Let me just give you the syntax:

   \[
   \text{newlist} = \text{Table}[\{ \gamma, \phi[t] /. \text{soln}[[1]] \}, \{t, 10, 20, 1\}];
   \]

   This would extract the values from \( t = 10 \) to 20 in increments of 1.

6. The third expression adds the new \( (\gamma, \phi) \) values stored in \( \text{newlist} \) to the ever-growing \text{list}, via the command

   \[
   \text{list} = \text{Union}[\text{list}, \text{newlist}]
   \]

7. Be sure to close up your for-loop with the closing square bracket and see if this runs. If it completes successfully, you can plot your data with

   \[
   \text{ListPlot}[\text{list}, \text{PlotStyle} \rightarrow \text{PointSize}[\text{Small}]]
   \]

8. Your plot probably looks bad, but if it has a bunch of points with the horizontal range between \( \gamma = 1.06 \) and 1.087, you’re probably in good shape. If not, be sure to empty the \text{list} variable again before re-trying.

9. Once it is looking decent, empty the list, increase the \text{NDSolve} solution out to time \( t = 400, \) and in the \text{Table} command, extract the values from \( t = 300 \) to \( t = 400. \)

   On a linux machine, this longer run takes around a minute. It might be a bit slower on Windows, but hopefully in that ballpark.
In principle, you should now have a beautiful bifurcation diagram. Please print out your whole notebook (perhaps after tidying it up) and not just the plot.

**Problem C**

Here are a couple of situations where we wouldn’t usually make a state space plot. But they are good for practice.

(a) Consider an object starting from rest and accelerating downward due to gravity alone (no air resistance). Make a plot (by hand is fine) of $\dot{y}$ versus $y$.

(b) Now consider the same situation except with air resistance. Make a plot that shows the trajectory from $t = 0$ into the time region where the object has reached terminal velocity.

**Problem D**

We’re going to make both a state space plot for the attractor plus a Poincare section for a variety of $\gamma$. Consider the driven damped pendulum with $\omega = 2\pi$, $\omega_0 = 3\pi$, and $\beta = 3\pi/4$. Use initial condition $\phi(0) = -\pi/2$ and $\dot{\phi}(0) = 0$. Here are the steps.

1. Define `eqn` as before. But in the `NDSolve` command, there is an important difference: type

   ```math
   soln = NDSolve[{eqn, \phi[0]==-\pi/2, \phi'[0]==0}, \phi, {t, 0, 100}, MaxSteps -> \text{Infinity}].
   ```

   The difference is in the middle: instead of defining the thing to solve for as $\phi[t]$, define it as $\phi$. I don’t know why, but it makes a difference.

2. For the parametric plot, use the command

   ```math
   ParametricPlot[{\phi[t], \phi'[t]}/.soln, {t, 30, 100}, AspectRatio->0.6]
   ```

   Notice how the `/soln` part comes after the curly braces.

3. For the Poincare plot, use the command

   ```math
   list = Table[{\phi[t], \phi'[t]}/.soln, {t, 30, 99, 1}];
   ListPlot[list]
   ```

   Similar syntax for the `/soln`. This takes values in increments of 1 from 30 to 99. For the Poincare plot, if you are seeing fewer points than you expect, it can be because choosing integer $t$ values isn’t optimal (it could be the place where the different phase space curves are close together). So there’s no reason not to try going from 30.4 to 99.4 in steps of 1, for example, in case it gives more “typical” data.

So with that procedure, make the state space plot and the Poincare section for the following, and comment whether the results are what you expected.

(a) $\gamma = 1.06$; (b) $\gamma = 1.078$; (c) $\gamma = 1.081$; (d) $\gamma = 1.085$; 
(e) Any other $\gamma$ value that seems to give you a nice chaotic behavior. Play around!