Lab 4 - Nonlinear driven pendulum

Thursday 18 September - Due: Thursday 25 September

In this lab we will build a code to simulate a simple non-linear driven pendulum along the lines of the one we discussed in class. This will also serve as an introduction to the `visual.graph` module for 2D plots and also additional Vpython objects such as cylinder. We will be able to see chaotic behavior for some values of the driving force.

1 A basic code

1. Download the code `pendulum.py` from the LABS page. Paste it into an IDLE window and save a copy to your filespace.
2. Run the code. What do you see ? Can you explain why there is no motion in the Vpython display ? Does the code do anything at this point ? The angle turned through by the pendulum bob is called `ball.angle` in the code. What is the name given to the pendulum bob's angular velocity in the code ?
3. To see something interesting we need to open up a display to show the angle turned through by the pendulum as a function of time. Do this by
   (a) Adding the initial lines of code after the import statement.
   ```
   pic=gdisplay()
   angleplot=gcurve(gdisplay=pic)
   ```
   (b) Also, add a new line directly after `from visual import *` which reads
   ```
   from visual.graph import *
   ```
   (c) Finally, add the following line of code (correctly indented!) at the end of the `while` loop
   ```
   angleplot.plot(pos=(t,ball.angle)
   ```
4. Run the code and make a screen capture of what you see. Also show a listing of the modified code at this point.
5. Run the code for 200 units of time. What do you notice about the magnitude of the angle displayed ? How many radians constitute one entire revolution ?

6. To fix this add the following code after the statement which updates the variable `ball.angle` in the `while` loop. This piece of code keeps the angle within the range $-\pi \rightarrow \pi$ at all times.

```
if (ball.angle>math.pi):
    ball.angle=ball.angle-2*math.pi
if (ball.angle<-math.pi):
    ball.angle=ball.angle+2*math.pi
```

7. Take a new screen capture of the output after 100 units of time.

2 Visualizing the motion

1. Now let's add code to animate the motion of the pendulum bob. To do this we need to build the entire pendulum out of objects in Vpython. We already are using a `sphere()` object to hold the bob. Let's use a `cylinder` object to hold the string. Add the following line of code after the definition of the `ball` object.

```
string=cylinder(pos=(0,0,0),axis=ball.pos,color=color.green,radius=0.1)
```

   This creates a cylinder object called `string` with one end at the origin of coordinates and with axis pointing towards the center of `ball`.

2. Run the code. Do you see the string ? The problem is that we have not yet assigned any coordinates to `ball` – so by default it sites at the origin of coordinates obscuring the `string`!

3. Let us assume that the string is of length `L=10.0` and one end is anchored at the origin and the other is at the coordinates of the bob.

4. Work out a mathematical expression for the coordinates of the bob as a function of the angle made by the string with the vertical and the length of the string. Write a line of code to assign these coordinates to `ball.pos` eg.

```
ball.pos=vector(....
```

   You need to place this line of code after the initial creation of `ball`

5. Now run the code. What do you see ? It should be clear what the problem is – the pendulum is created and drawn correctly at the beginning but its position (and the `axis` argument of the cylinder) are not updated after each time step. Correct this and paste in a new code listing.

3 Chaos

1. Run the code for $F = 1.35$ and make a screen capture of the angle plot. What is the period of the motion ? How does this compare to the period of the driving force (which parameter in the code does the latter correspond to ?) Now change $F = 1.44$ and make another screen capture. What is the new period of the motion. How about for $F = 1.465$?

2. Modify your code to plot out not $(t, angle)$ but $(angle, anglevel)$. What does the latter plot look like for regular motion (eg $F = 0.5$ or $F = 1.35$) What about for chaotic motion $F = 1.2$? Show screen captures.