1 Introduction to Animation

Python has two basic modes: normal and interactive. The normal mode is the mode where scripts and complete .py files are run using the Python interpreter. Interactive mode is a command line environment which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole.

We will make animated plots of the normal modes of particles on a string, which we recently derived in class. To animate a plot we will turn on “interactive mode” in Python, and then repeatedly draw a graph so that each plot is slightly different than the previous. As long as we remove the previous graph before drawing the next one, and each graph isn’t too different than the previous, we will get a smooth animation.

Task 1: Type in this code and then run it:

```python
import numpy
import pylab

pylab.ion()  # interactive mode

xaxis = numpy.linspace(-2,10,100)
for i in range(100):
    pylab.clf()  # clear the graph
    yvals = numpy.cos(xaxis + i/10.)
    # yvals = numpy.cos(xaxis) * numpy.sin(i/20.)
    pylab.plot(xaxis, yvals)
    pylab.ylim([-1,1])  # set the vertical axis to [-1,1]
    pylab.draw()
```

Note that as the for loop iterates, the animation works to simulate the passage of time, \( t \). In this case, \( yvals \) is of the form \( f(x + vt) \). You may recall from Phys 212 that this represents the equation for a wave travelling to the left.

Task 2: Now, run it again where you switch which \( yvals \) line is commented out.

Compare the code to the output and identify the purpose of each line in the logical sequence of the code. Check with one of your instructors if you have any questions about how it works.

2 N-coupled Oscillators

Now we will modify the code for the case of a small number of coupled oscillators. Let’s begin with the 4-coupled oscillator case, which means that we need points at 0, 1, 2, 3, 4, and 5, i.e., \( p = 0, \ldots, N + 1 \).

You might think that we could set

\[
xaxis = [0,1,2,3,4,5]
\]
But we want to do arithmetic operations on \texttt{xaxis}, meaning that we want to operate on the list and have that operation apply to every element in the list.

\textbf{Task 3:} Type the above command into Python, and then try adding 1 to it, or multiplying it by 3, or multiplying it by 3.2. Discuss the results with someone sitting near you.

We need a different type of list, on which we can do arithmetic. One possibility, which is the method used last week, is to set:

\begin{verbatim}
    xaxis = numpy.linspace(0,5,6)
\end{verbatim}

Another option is to set:

\begin{verbatim}
    xaxis = numpy.asarray([0,1,2,3,4,5])
\end{verbatim}

Decide which of these methods you prefer, and use it in your code.

\textbf{Task 4:} We will use the formulae derived in class to modify the above code so that it animates one of the normal modes for the case of 4-coupled oscillator system. Sketch out how you will modify the code and discuss this with a neighbour and then with one of the instructors.

\textbf{Task 5:} Implement your code and show the animation to your instructors.

\textbf{Task 6:} Write a function that can be used to plot any normal mode for any number of particles. Show your instructor.

\textbf{Final report}

Modify your program to prompt the user for the number of particles and the normal mode, and then show an animation of that mode.