ECE 351 Lab 1

Andrew Hartman

September 2019

Contents

1 Introduction

This lab was designed to introduce the tools used for programming in Python 3.x as well as some functions and syntax of Python that will be used. To complete the lab the program Jupyter Notebook or Spyder are used as the python interpreters. Both programs complete the same task, being a development environment and running the code.

2 The Jupyter Notebook Environment

Jupyter Notebook is the application the labs in this course are designed for. It is both a python interpreter and Tex editor. There are two types of cells within Jupyter Notebook, the first being code cells, the code within these cells are interpreted and ran by python, the other cells are called markdown cells, these are used for latex and report writing. In both Jupyter and other latex editors it is possible to include equations formatted nicely by either including them inside of $'s or between \beginequation \endequation.

3 Using Python 3.x

3.1 Variables, Arrays, and Matrices

In python variables do not need to be defined using a specific type. The interpreters is able to interpret the variable type when it is assigned a value. Python also does not require end of line characters like other languages such as C/C++. In order to view a variable the print() command can be used. Python 3.x has a built in way to deal with arrays, lists, as well as a different way inside the numpy library, arrays. These are defined as follows:

```
list1 =[0,1,2,3]
array1=numpy.array([0,1,2,3])
```

The numpy arrays provide a few benefits that python lists do not. It is easier to manage multi-dimensional arrays. If a list is 2 dimensional it can not be individually addressed, whereas arrays allow this.

```
array2 = np.array([[1,2,3,4,5],[6,7,8,9,10]])
list2 = list(array2)
print('array2:',array2[0,2],array2[1,4])
print('list2:',list2[0],list2[1])

array2: 3 10
list2 : [1 2 3 4 5] [ 6 7 8 9 10]
```

Numpy arrays also provide other functions to use when working with arrays. numpy.zeros() and numpy.ones() can be use to create arrays of all 0's or 1's. numpy.arrange() and numpy.linspace() can be used to create linearly stepped arrays of values within bounds or up to a limit to use for sequencing.

3.2 Plotting Functions

Python 3.x has a package built for it to easily create and modify plots called matplotlib.pyplot. To use this library first a new figure needs to be created, and then define a subplot.

```
matplotlib.pyplot.figure(figsize=(12,8))
matplotlib.pyplot.subplot(3,1,1)
```

The parameters passed into the subplot function define the row of the figure to show up in, then the column, and finally what number the subplot is. Next a function needs to be plotted. An array of values for the x axis and a function for the y need to be defined. These can then be plotted using matplotlib.pyplot.plot().

```
x = np.arange(-2,2+steps,steps)
y1 = x + 2
plt.plot(x,y1)
```

To change the title of the whole figure the matplotlib.pyplot.title() function can be used. The subplots can be labels using matplotlib.pyplot.ylabel(). Finally to display the plots the function matplotlib.pyplot.show() has to be called. Plots can also be created on individual figures, but it is general practice to make subplots within one figure if the plots relate to eachother in some way.

3.3 Complex Numbers

The python numpy library also provides a simple implementation of complex numbers. Python uses 'j' as the imaginary number. The numpy functions .real() and .imag() allow you to extract the real or imaginary sections of an imaginary form. The python abs() function returns the magnitude of the complex number and the numpy.angle() function returns the angle in radians. If a square root in python returns a negative number it can cause an issue and return Nan. To avoid this add a +0j to the value inside a square root to force the python interpreter to use complex numbers.

4 Additional Helpful Commands

This last section of lab lists some useful packages and functions that are commonly used in Python. The majority are functions to enable manipulation of numpy arrays. This includes combining, inserting, reshaping, transposing, finding the length of, and dimensions of arrays among other things.

5 Questions

1. For which course are you most excited in your degree? Which course have you enjoyed the most so far?

I am very excited for ECE 340 and probably 440, working with microcontrollers and systems interests me. SO far the course I have enjoyed the most was one outside of my degree, it was my ceramics course.

2. Leave any feedback on the clarity of the purpose, deliverables, and tasks for this lab.

This lab was easy to follow on what the purpose and deliverables should be. The tasks were simple due to just being copying code and seeing what it does. It may have been beneficial if I did not know some python beforehand to have a few small tasks to use the newly learned python skills.

6 Conclusion

In this lab a rudimentary base for python 3.x was developed. Good explanations for how to use arrays, plots, and complex numbers provides a good set of knowledge to further develop understanding of how to apply python to actual problems.