Climate School. Chris Budd and Matt Griffith, Bath

Some basic Python tools and a group exercise

Python is a general purpose, and easy to use, programming language. It is free, runs on most systems, and has extensive internet support. It is also an excellent mathematical calculator and graph plotter. This is a very short set of notes to allow you to run the Python code needed for the climate work sheet. Python can do MUCH more than this. A great reference is the book *Python for Scientists.* by John Stewart. At the end of the notes there is a whole group exercise to try your skills on.

1. *Starting up* To start a Python session you need to load some libraries to do mathematics, and plotting respectively. To do this type the following instructions after the prompt

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
```

2. *Basic operations* Python can be used as a scientific calculator. To do this you need to use commands from the numpy library. These can be used to do operations such as

```
>>> x = 1
>>> y = 2
>>> z = x+ y
>>> w = np.sin(2*np.pi*z**5/7)
```

In this case the last instruction calculates $w = \sin(2\pi z^5/7)$.

3. *Vectors* A powerful feature of Python is its ability to work with whole vectors (a feature that it shares with other languages such as Matlab). The `np.linspace` allows you to set up a vector such as

```
>>> x = np.linspace(1,10,num=10)
```

will set up the vector $x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. You can then work (pointwise) on the whole vector. Thus the expression

```
>>> y = x**2
>>> z = y + 2*x
```

gives the vectors $y = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]$, $z = [3, 8, 15, 24, 35, 48, 63, 80, 99, 120]$. Python can also do a wide variety of other vector operations, including solving linear equations and a wide variety of matrix operations.

4. *Basic plotting* Python has an impressive range of graphical facilities, most of which which we won't look at here. For the exercises we will need to do some basic plotting. Suppose that you want to plot two functions, say $y = \sin(x)$ with $0 \leq x \leq 10$ and $z = \cos(3 * x) - \sin(2 * x)$ over the same range. We want the first graph to have a blue solid line and the second a red dashed line. An easy way to do this is to use the `np.linspace` command to set up the vector of x values that you use. The following commands will do this and do the plot that you need. In this case we sub-divide the interval $[0, 10]$ into 1000 points.

```
>>> x = np.linspace(0,10,num=1000)
>>> y = np.sin(x)
>>> z = np.cos(3*x)-np.sin(2*x)
>>> plt.plot(x,y,'b')
```

```
>>> plt.plot(x,z,'r--')
>>> plt.show()
```

This plots the two graphs and then shows you the results at the end. You can add x and y labels before you show the graphs as follows

```
>>> np.xlabel('x')
>>> np.ylabel('This is a graph of y')
```

For more plotting (such a plotting in three dimensions, movies etc.) see the book by Stewart.

5. *Loops* In the exercises we will be working with loops where we repeat similar operations many times. Suppose that we want to display the square of numbers $7, 8, \ldots 22$. We can do this as follows

```
>>> for n in range(7,23):
....  x = n**2
....  x
>>>
```

NOTES

1. The colon : is very important in the line `for n in range(7,23):`
2. After the .... put in a space. Python requires indentation to mark the start and end of for loops and so an indent of 2 or 4 spaces is crucial.
3. Note that the last number in the range is one more than the maximum number that you want to square.

After the Autumn school has finished, learn more about Python and experiment. For example you can use Python to solve differential equations. These are often used in climate models.

Group Exercise: Is Climate Chaotic?

One model for the evolution of climate uses the *Logistic equation* to predict the temperature of the Earth. In this model the temperature in the nth year is called T_n. The temperature in the (n+1)st year can be predicted from that in the nth year through the equation:

$$T_{n+1} = c\, T_n(1 - T_n).$$

This model works with *normalised temperatures* T_n lying in the range $0 < T_n < 1$. With $T_n = 0$ being cold and $T_n = 1$ being hot. This model starts with an *initial state* $T_0 = Tinit$ and evolves this forward in time. The type of climate that we see then depends on the value of c. In all cases we will take $Tinit = 0.2$ and we will look at different values of c. We can implement this model by using the vector and loop features of Python. We will take T to be the vector of all of the temperature values. In this each element of the vector will represent the temperature of the Earth in a given year. Suppose that the years are numbered $n = 0 \ldots 49$, then we can initialise a vector of temperatures $T[n]$ by using the command

```
>>> T=np.zeros(50)
```

This tells Python to expect to work with a vector T of the form $(T[0], T[1], \ldots, , T[49])$. (Note that Python indexes vectors starting from the index zero.)

We now want to see how the values of T_n vary from year to year starting from some initial value `Tinit` . To do this we need to use a simple loop. One way to do this is as follows

```
>>> x = Tinit
>>> for n in range(0,50):
....   T[n] = x
....   x = c*x*(1-x)
>>>
```

You can plot the vector T as a line graph as follows

```
>>> plt.plot(T)
>>> plt.show()
```

You need to close the plot when you have finished looking at it. You may prefer to plot both a line graph and points. One way to do this is

```
>>> plt.plot(T,'r')
>>> plt.plot(T,'bo')
>>> plt.show()
```

Experiment with the plotting style that suits you best.

Now perform the following exercises.

(i) Set $c = 2.8$, and using a Python loop (see notes above) to generate a sequence of values of T_n for $n = 0, 1, \ldots 49$ with $T_0 = Tinit$. Now plot the whole vector T (see notes again). What do you notice? This represents a situation where the climate tends to a steady state. This type of behaviour is very predictable.

(ii) Set $c = 3.2$ and repeat your calculation in *(i)*. What do you now see? This represents a case where the temperature of the Earth can swing between two values (so called hot and snow ball Earth).

(iii) Set $c = 3.5$ and repeat. What do you see? How might you describe this climate?

(iv) Set $c = 4$ and repeat. What do you see? This type of behaviour is called *chaotic*. It is very hard to predict. How might you advise a government in this case? Fortunately climate is not really like this, as we can see from historical records.

CJB/MG