Scientific Python

Preparatory Course Information

23 - 24 November 2015

Introduction

This document provides information that will help you to get the most out of the Scientific Python course.

1. What the course will cover

In this two-day course we will introduce the core Python packages used for scientific computing. Here is the course outline:

- **Lecture 1** – Introducing the course, Python on ARCHER, Using the IPython shell, brief recap of core Python basics (see **a.**).
- **Lecture 2** – Introduction to NumPy, using arrays in Python
- **Lecture 3** – Introduction to Matplotlib, basic plotting and plotting for publication
- **Lecture 4** – Introduction to SciPy, exploring some of the available functionality
- **Lecture 5** – Introduction to f2py and how to call Fortran and C functions in Python

3. Prerequisites

We will assume students are familiar with the core Python basics listed below in (**a**). We will also assume that students will use their own laptops for the course, see (**b**).

a. What you should know

You will need a grasp of the core Python basics. There are many online resources, for example Code Academy's basic Python course, or the SciPy introductory lectures notes:

Code Academy
Scipy lectures notes

Following Code Academy's basic Python course, you should be familiar with:

- **Python syntax** : importance of whitespace, basic data types (integer, float, string), variables, basic mathematical operations, how to use `print`

- **Strings & console output**
  Creating and using strings, converting numerical types to strings, string concatenation, printing with `%`

- **Conditionals and control flow**
  Comparators (==, !=, <=, >=) and Boolean operators, truth table (use of `AND`, `NOT`, `OR`), using `if..elif..else`

- **Functions**
  How to define your own function, how to import modules and particular functions, avoid universal import, built-in functions such as `abs(), min() max(), len() and type()`

- **Lists**
  Mutatble data types/data structures, how to create lists e.g. using `range()`, iterating over a list, create a list of random integers with `randint()`, accessing a list by indexing and slicing (e.g. `list[start:end:step]`), list functions (using dot notation) such as `.insert(), .append()`

- **Loops**
  The `for… :` loop, the `while… : do` loop.

b. What you should bring

We strongly encourage you to use your own laptop for the course. You will also need access to the following:

- Python, version 2.7
- Python packages `NumPy, SciPy`, `Matplotlib` and `IPython`
- compilers for Fortran and C
- A text Editor
- GNU Make
- A Secure Shell (SSH) client

To have access to all the above, you have two options.

i. Set up your own laptop

This option requires a bit of work, but provides long term access beyond the course. To avoid the potential difficulty of installing Python and all the necessary packages and software tools separately, we recommend the Anaconda distribution which you can download and install from here.

You will need a C compiler (e.g. **gcc**) and a Fortran compiler (e.g. **gfortran**) to be able to compile external code used with `f2py`:

Linux users should have gcc installed on their machine by default. It should be easy to add the Fortran compiler gfortran.

Mac users can get gcc can install Xcode Command Line tools by doing:

```
  xcode-select  --install
```

in a terminal window. For more information, see: OSX Daily Blog. Once you have Xcode Command Line tools, you can get gcc and then gfortran from the web.

Windows users can download and install MinGW to get access to gcc and gfortran. Alternatively you can install Git Bash.

You will also need GNU Make, a text Editor and a Secure Shell (SSH) client (see **ii.** for information about SSH clients).

You may find the set up information on this Software Carpentry course page useful.

Please note: we are unlikely to be able to spend any time helping you resolve issues with your set up during the course. Fear not, however, you have the back-up option of using ARCHER.

ii. Use ARCHER

This is the simplest option. Python, all necessary packages, software tools and compilers for the course are installed on ARCHER. We will be using the Anaconda Python distribution, which is available on ARCHER, see here.

All you need is a Shell (SSH) client installed on your laptop to use ARCHER.

For Linux/Mac OSX
The Linux terminal comes with a SSH client.

For Windows
You could install one of the following popular SSH clients:

• PuTTY	http://www.putty.org/
• MobaXterm	http://mobaxterm.mobatek.net/

We will cover how to connect to ARCHER in the course, but you can find more information before then on the ARCHER website e.g. here

c. Optional

Jupyter Notebooks

Please note: we do not recommend running notebooks on ARCHER. There may be provision for running notebooks remotely on an alternative server. However, we cannot guarantee its availability for the course.

Jupyter (formerly IPython) notebooks are web browser-based interactive documents that allow you to execute Python code, visualize results and much more, all in one document. The `jupyter` and `notebook` packages come with the Anaconda distribution:

For more information on how to get started with Jupyter notebooks, see:

http://jupyter.org

http://ipython.org/

http://jupyter-notebook.readthedocs.org/en/latest/examples/Notebook/rstversions/

http://ipython.org/notebook.html

http://johnlaudun.org/20131228-ipython-notebook-keyboard-shortcuts/