Introduction to Scientific Python
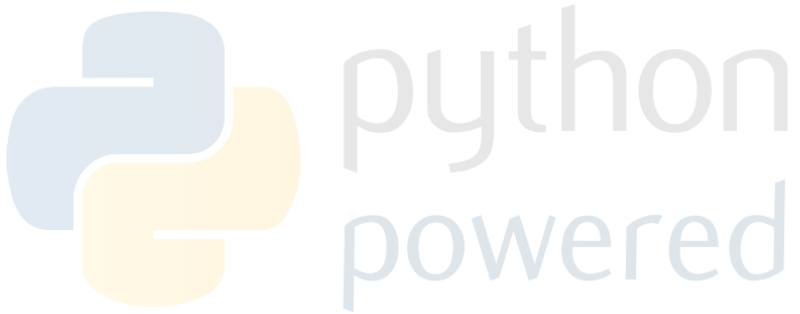
Iñigo Aldazabal Mensa – Centro de Física de Materiales (CSIC-UPV/EHU)

CFM Software Carpentry Tracks

San Sebastián, 4th May 2017

Why Python?

Why Python?

Python is inherently slow compared to C/C++ or FORTRAN, so why Python for Scientfic Computing?

Python is slow, but...

Syntactically, Python code looks like executable pseudo code. Program development using Python is 5-10 times faster than using C/C++...

The best approach is often to write only the performance-critical parts of the application in C++ or Java, and use Python for all higher-level control and customization.

– Guido van Rossum

Why Python?

We have (for free)

- A general purpose language with a huge spectrum of freely available libraries for almost anything you can think of.
- A very easy to learn (and read) language that smoothly interfaces with C/C++ and FORTRAN (eg. calculation kernels).
- Lots of wrappers for well stablished, fast and long time tested numerical packages.
- Lots of high level utility libraries for scientific computing: plotting, data analisys, parallelization, ...

Why Python? Batteries Included

PYTHON:
BATTERIES INCLUDED

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

Python Scientific Computing Environment

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

NumPy
Base N-dimensional array package

SciPy library
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

Python Scientific Computing Environment

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

NumPy

NumPy

Base
N-dimensional
array package

SciPy library
Fundamental
library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]: IPython
IPython Enhanced
Interactive Console

NumPy is the fundamental package for scientific computing with
Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number
  capabilities

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

NumPy

NumPy

Base
N-dimensional
array package

SciPy library

Fundamental
library for scientific
computing

Matplotlib

Comprehensive 2D
Plotting

IP[y]: IPython
IPython

Enhanced
Interactive Console

```python
import numpy as np

# Create a numpy array, x
X = np.array( [1.1, 1.3, 1.5] )
y = np.sin(a)

# create a random two dimensional numpy array, A
A = np.random.rand(3,3)

A.transpose()
A.trace()
```

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

SciPy

NumPy
Base
N-dimensional
array package

SciPy library
Fundamental
library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]: IPython
IPython Enhanced
Interactive Console

SciPy is a collection of mathematical algorithms and convenience functions built on the Numpy extension of Python.

Provides the user with high-level commands and classes for manipulating and visualizing data.

With SciPy an interactive Python session becomes a data-processing and system-prototyping environment rivaling sytems such as MATLAB, IDL, Octave, R-Lab, and SciLab.

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

SciPy

NumPy
Base
N-dimensional
array package

SciPy library
Fundamental
library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]: IPython
IPython Enhanced
Interactive Console

SciPy subpackages (some of them)

- constants: Physical and mathematical constants
- fftpack: Fast Fourier Transform routines
- integrate: Integration and ordinary differential equation solvers
- interpolate: Interpolation and smoothing splines
- linalg: Linear algebra
- optimize: Optimization and root-finding
- signal: Signal processing
- special: Special functions
- ...

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

SciPy

NumPy
Base
N-dimensional
array package

SciPy library
Fundamental
library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]:   IPython
IPython   Enhanced
          Interactive Console

```python
import numpy as np
from scipy.special import gamma

x = np.array( [1.1, 2., 3.] )
y = gamma(x)

print (y)
```

```
[ 0.95135077,  1. ,  2. ]
```

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

matplotlib

NumPy	SciPy library	Matplotlib	IPython
Base N-dimensional array package	Fundamental library for scientific computing	Comprehensive 2D Plotting	IPython Enhanced Interactive Console

matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

matplotlib can be used in Python scripts, the Python and IPython shell (ala MATLAB® or Mathematica®), web application servers, and several graphical user interface toolkits.

For simple plotting the `pyplot` interface provides a MATLAB-like interface, particularly when combined with the `IPython / Jupyter` environment.

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

SciPy

NumPy
Base
N-dimensional
array package

SciPy library
Fundamental
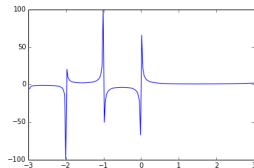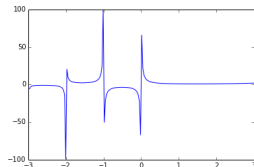library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]:
IPython
IPython
Enhanced
Interactive Console

```python
import numpy as np
from scipy.special import gamma
import matplotlib.pyplot as plt

x = np.linspace( start=-3., stop
    =3., num=200 )
y = gamma(x)

plt.plot(x,y)
plt.savefig("gamma.png")
```

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

SciPy

NumPy	SciPy library	Matplotlib	IPython
Base N-dimensional array package	Fundamental library for scientific computing	Comprehensive 2D Plotting	Enhanced Interactive Console

IP[y]: IPython
        Enhanced
        Interactive Console

```python
import numpy as np
from scipy.special import gamma
import matplotlib.pyplot as plt

x = np.linspace( start=-3., stop
    =3., num=200 )
y = gamma(x)

plt.plot(x,y)
plt.savefig("gamma.png")
```


Just google matplotlib images for examples!

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

IPython / Jupyter

NumPy
Base
N-dimensional
array package

SciPy library
Fundamental
library for scientific
computing

Matplotlib
Comprehensive 2D
Plotting

IP[y]:
IPython

IPython
Enhanced
Interactive Console

The **IPython** / **Jupyter Notebook** is a web-based interactive computing system that enables users to author documents that include live code, narrative text, LaTeX equations, HTML, images and video.

These documents contain a full record of a computation and its results and can be shared on email, Dropbox, version control systems (like git/GitHub) or with the Jupyter online notebook viewer nbviewer.jupyter.org.
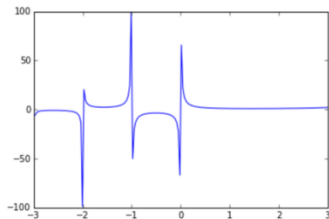
Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

IPython / Jupyter

Python

Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

IPython

Python
Python Scientific Computing Environment

NumPy
SciPy
matplotlib
IPython / Jupyter

Python Scientific Computing Environment

Let's try it!