

Python for Matlab Users

Computational Seminars @ GSE

Dieter Werthmüller

2nd March 2023

⇒ github.com/prisae/Python4MatlabUsers

- **What this workshop is**
 - A quick primer for Matlab users: What is the same, what is different?
- **What this workshop IS NOT**
 - An introduction to programming
 - An introduction to Python
 - Best practices and all the other stuff
- Ultimate objective of today
 - **YOU start to translate one of your scripts from Matlab to Python**

Rough comparison

Python

- Late 80's (Guido Rossum, then at CWI)
- Free and open-source
- General programming language;
scientific computing came later (late 90's)
- Many ways to install / use

Matlab

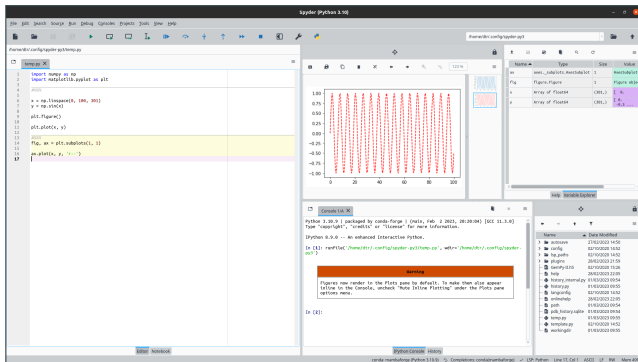
- Late 70's
- Proprietary and very costly
- Matrix computation
- One way to install / use

Both are multi-paradigm (e.g., *procedural*, *object-oriented*).

In Python, *everything* is an object.

⇒ Easiest for transition, the most “*Matlab-ish*” experience.

- Editor with
 - cells
 - linting
 - visual debugging
- Variable explorer
- Can run notebooks (spyder-notebook)
- Line profiler (spyder-line-profiler)



Terminal, Plain Python, IPython (Atom, VSCode, PyCharm, Sublime, ...)

Terminal based solutions

```
(base) ~/> python
Python 3.10.9 | packaged by conda-forge | (main, Feb 2 2023, 20:20:04) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import numpy as np
>>> x = 4
>>> np.sqrt(4)
2.0
>>> █
```

Plain Python

```
IPython: home/dtr
(base) ~/> ipython
Python 3.10.9 | packaged by conda-forge | (main, Feb 2 2023, 20:20:04) [GCC 11.3.0]
Type 'copyright', 'credits' or 'license()' for more information
IPython 8.9.0 -- An enhanced Interactive Python. Type '?' for help.
In [1]: import numpy as np

In [2]: x = 4

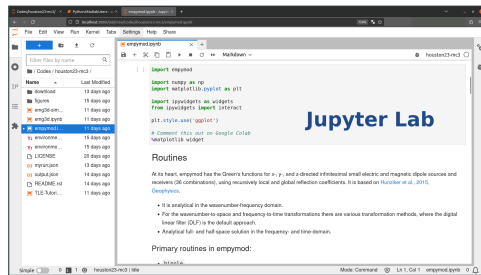
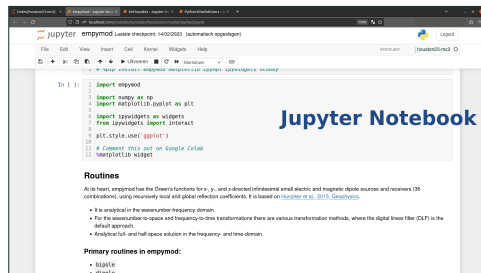
In [3]: np.sqrt(x)
Out[3]: 2.0

In [4]: _3
Out[4]: 2.0

In [5]: x = np.linspace(10, 400, 31)
```

IPython

Jupyter ecosystem (language agnostic!)



Distributions and Package Manager

Package Manager

- `pip` (`pip install package`)
- `conda` (`conda install package`)
- `mamba` (`mamba install package`)
- `conda`/`mamba` are language agnostic.
- `mamba` is the newer/faster `conda`.

Distributions

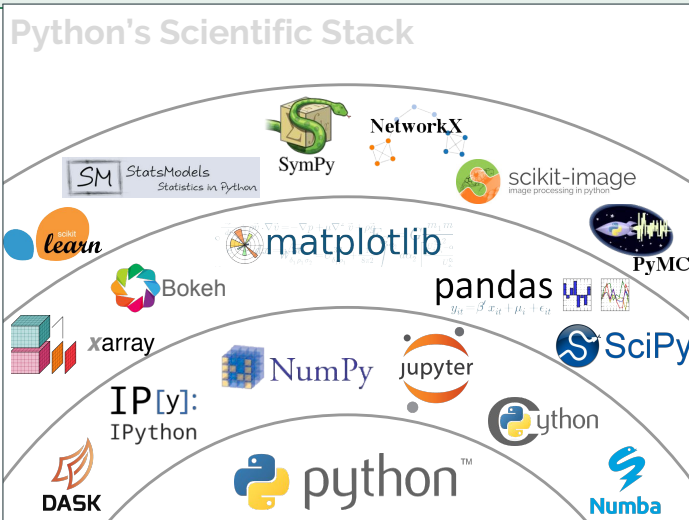
- There are many (e.g., Anaconda, conda-forge, Python(x,y); PyPy, EDM, WinPython)
- I recommend mambaforge:
github.com/conda-forge/miniforge#mambaforge

Make a note for the future: Once you translated your first scripts, get familiar with **environments**; crucial for **reproducibility**!

conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Why use Python? It is SO SLOW! (I hear it every day)

- In scientific computing, Python is GLUE
- High-level syntax wraps low-level C/Fortran libraries, which is (mostly) where the computation happens.
- AOT & JIT: Cython, PyPy, Numba, Pythran, ...



Jake VanderPlas, PyCon2017, The Unexpected Effectiveness of Python in Science

- \Rightarrow Bookmark this: numpy.org/doc/stable/user/numpy-for-matlab-users.html
- More in depth: enthought.com/wp-content/uploads/2019/08/Enthought-MATLAB-to-Python-White-Paper_.pdf

- Th, 2 Mar, 2–4 pm, Room 1.21: **Python for Matlab Users**
- Th, 6 Apr, 2–4 pm, Room 1.93: **Rust**
- Th, 1 May, 2–4 pm, Room TBA: **JupyterLite**