Graphing with Pyplotlib

Thomas Schwarz

Introducing Python Modules

- Python is good for fast coding
  - But performance is not the best
- 1997 MIT graduate students need coding for Physics simulation:
  - Invent Jython (Python implementation on Java)
  - Create a module for numerical calculation
  - Which splits
- 2005: Several versions get united into SciPy
  - Still under development / extension

Introducing Python Modules

- Matlab is a standard tool in Engineering
    - Matplotlib is a Python module that emulates Matlab graphing standards in Python
    - Generates publication quality graphs

Introducing Python Modules

- Pandas
    - A package for data science
- Sympy
    - A package for simulation of physical systems
    - But really a symbolic computer algebra system

Installing Python Modules

- Python now comes with its own installer
  - pip or pip3 if you have Python2 and Python3
  - Always use pip3
- Open up a command window (terminal on Mac)
- Type `pip3 install numpy`
  - Might tell you to update pip (which you can do)
- Type `pip3 install matplotlib`
- Type `pip3 install scipy`

Using Python Modules

- You need to import modules in your code or in Idle

```
>>> from scipy import misc
>>> import matplotlib.pyplot as plt
>>> face = misc.face()
>>> plt.imshow(face)
<matplotlib.image.AxesImage object at 0x7fde31a951c0>
>>> plt.show()
```

Using Python Modules

- Importing modules
  - Version 1: `import numpy`
    - Now all of numpy is available
    - Need to prefix all elements of the module with `numpy.name`
  - Version 2: `import numpy as np`
    - Now all of numpy is available
    - Need to prefix all elements of the module with `np.name`

Using Python Modules

- Version 3: `from scipy import misc`
  - Now the submodule, function, or variable is available
  - No need for a prefix
  - But is a bit dangerous, since names in different modules might be the same

Using Python Modules

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

X = np.linspace(-5,10,1001)
plt.plot(X, np.exp(X)/10)
plt.plot(X, X**3-2)
plt.plot(X, 2000*np.sin(X))
plt.show()
```