



Scientific python + IPython intro

Alexey Buzmakov

Shubnikov Institute of Crystallography RAS,
Moscow

28/11/2013, XFEL, Hamburg

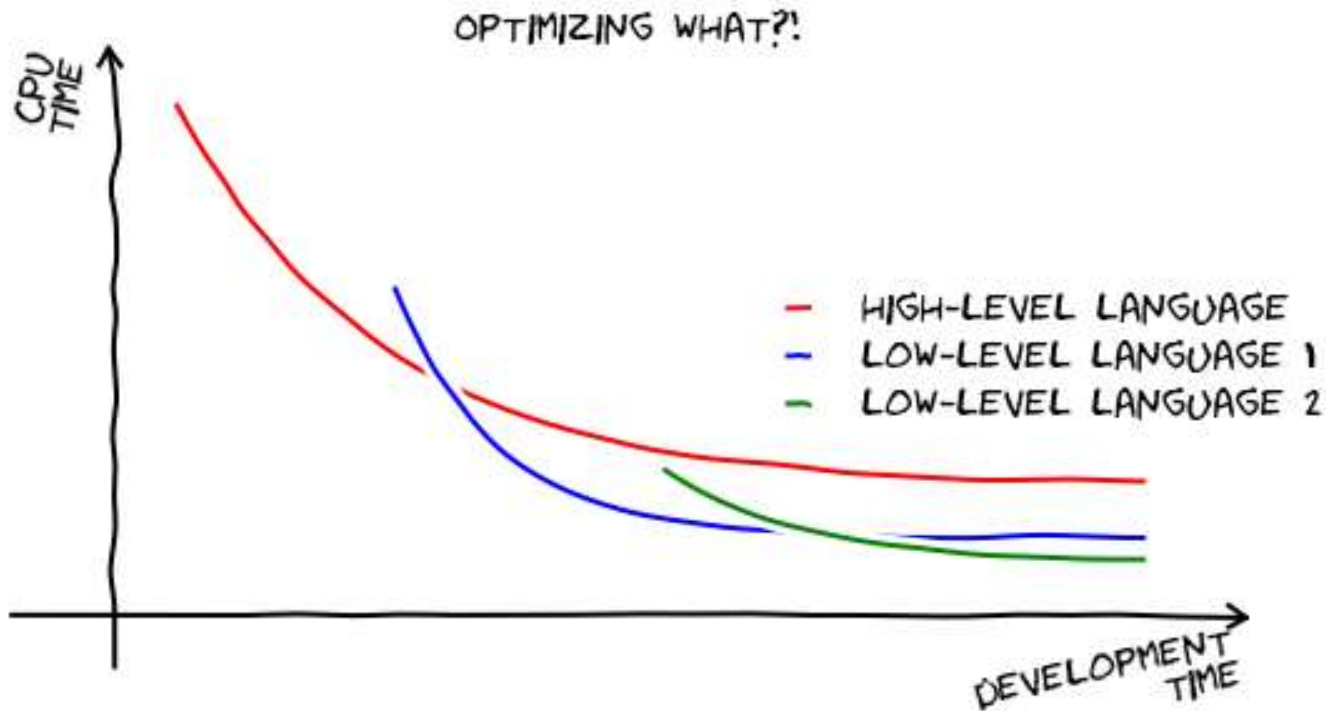
Headline

- Why python?
- What is python?
- How to use python and IPython?

The scientist's needs

- Get data (simulation, experiment control, data files)
- Manipulate and process data.
- Visualize results... to understand what we are doing!
- Communicate results: produce figures for reports or publications, write presentations.

- Compiled languages: C, C++, Fortran, etc.
- Scripting languages: Matlab
- Other scripting languages: Scilab, Octave, Igor, R, IDL, etc.



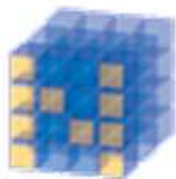
- General characteristics of Python:
 - **clean and simple language:** Easy-to-read and intuitive code, easy-to-learn minimalistic syntax, maintainability scales well with size of projects.
 - **expressive language:** Fewer lines of code, fewer bugs, easier to maintain.
- Technical details:
 - **dynamically typed:** No need to define the type of variables, function arguments or return types.
 - **automatic memory management:** No need to explicitly allocate and deallocate memory for variables and data arrays. No memory leak bugs.
 - **interpreted:** No need to compile the code. The Python interpreter reads and executes the python code directly.

- Advantages:
 - ➔ Very rich scientific computing libraries (a bit less than Matlab, though)
 - ➔ Well thought out language, allowing to write very readable and well structured code: we “code what we think”.
 - ➔ Many libraries for other tasks than scientific computing (web server management, serial port access, etc.)
 - ➔ Free and open-source software, widely spread, with a vibrant community.
- Drawbacks:
 - ➔ – less pleasant development environment than, for example, Matlab. (More geek-oriented).
 - ➔ – Not all the algorithms that can be found in more specialized software or toolboxes.

Scientific Python building blocks



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

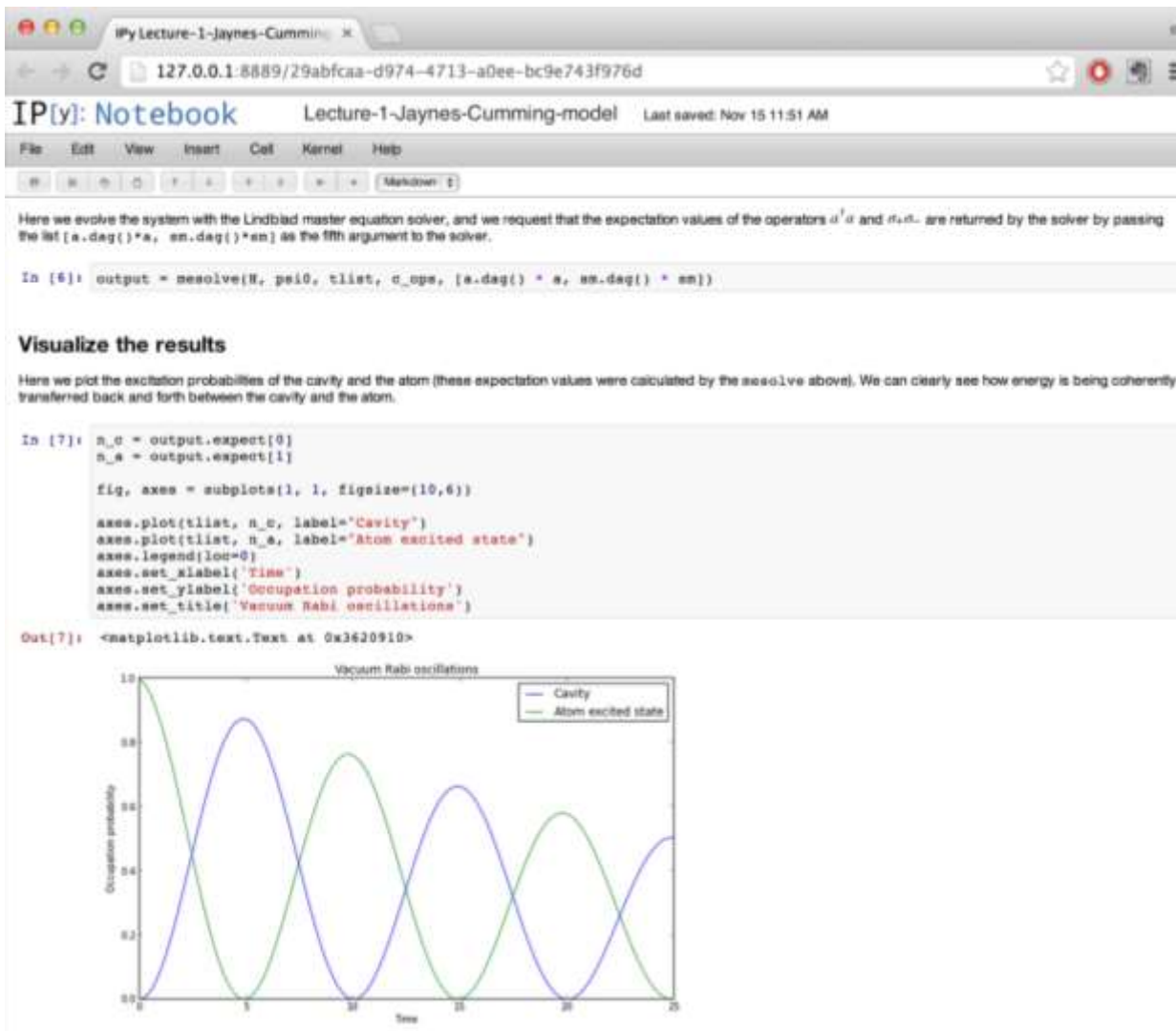
- There are currently two versions of python:
 - **Python 2** and **Python 3**.
 - Python 3 will eventually supercede Python 2, but it is not backward-compatible with Python 2.
 - A lot of existing python code and packages has been written for **Python 2**, and it is still the most wide-spread version.
- To see which version of Python you have, run
 - `$ python --version`
- Several versions of Python can be installed in parallel

- For example, to run a file my-program.py that contains python code from the command prompt, use:
 - `$ python my-program.py`
 - or run interpreter `$python`



```
rob:~$ python
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> █
```

- In scientific computing, we typically **don't know what we're doing**
- IPython is an **I**nteractive shell that addresses the limitation of the standard python interpreter, and it is a work-horse for scientific use of python.
- It provides an interactive prompt to the python interpreter with a greatly improved user-friendliness.



- <http://bit.ly/WPG-wiki> -> Tutorial files -> Ipython presentation
- Download it
- Run in this folder
 - ipython notebook

- Generate html and pdf reports
- Free online publishing service
<http://nbviewer.ipython.org/>

- Python <http://www.python.org/>
- Numpy + Scipy + matplotlib <http://www.scipy.org/>
- H5py <http://www.h5py.org/>
- Numpy for Matlab users
http://wiki.scipy.org/NumPy_for_Matlab_Users
- Free tutorials (some materials was taken in this tutorials)
 - <http://scipy-lectures.github.io/>
 - <http://jrjohansson.github.io/>
- Free binary distribution with a lots of scientific stuff (especially for windows users)
 - <https://store.continuum.io/cshop/anaconda/>
- Free scientific python IDE (Matlab like)
 - <https://code.google.com/p/spyderlib/>