

IPython and the Scientific Python Ecosystem

Fernando Pérez

<http://fperez.org>, @fperez_org

`Fernando.Perez@berkeley.edu`

Helen Wills Neuroscience Institute, UC Berkeley

Geilo Winter School on Reproducibility

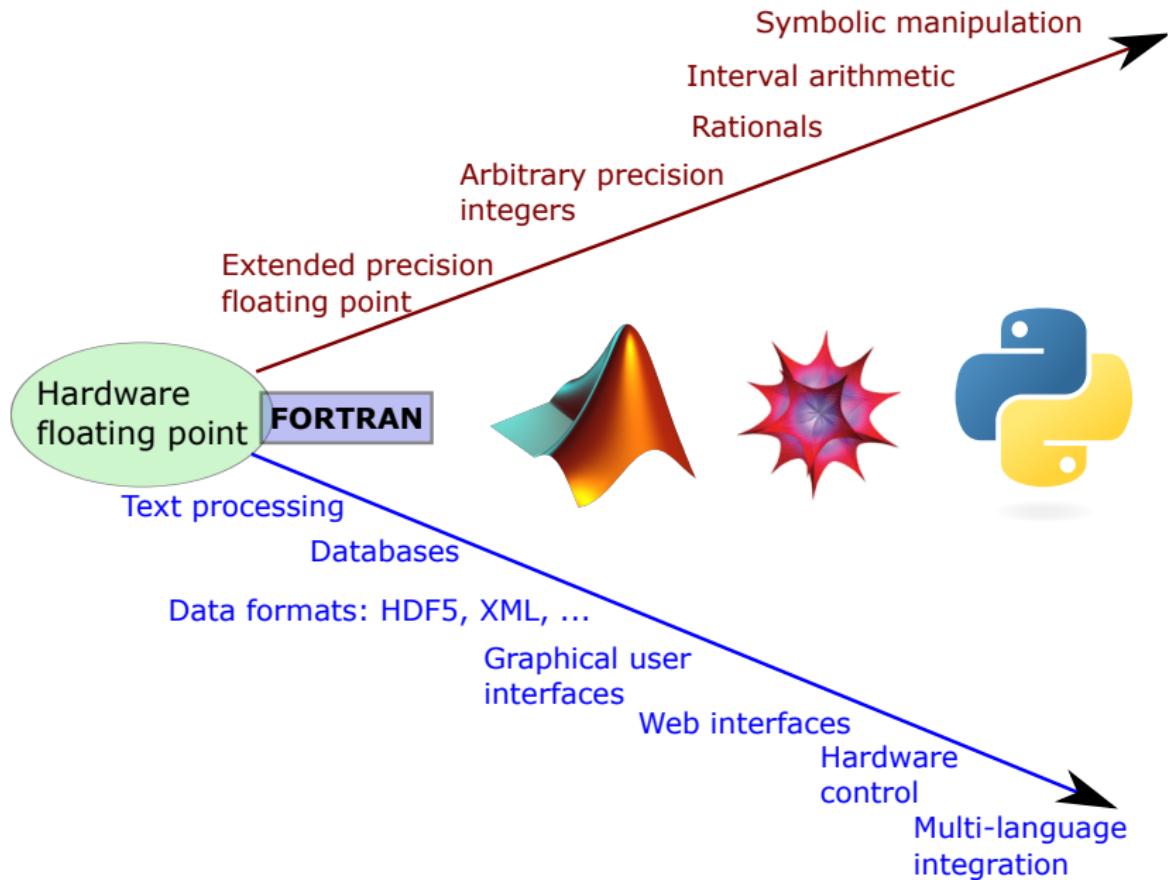
Jan 21, 2013

Outline

1 Scientific Python

2 IPython: Interactive Python

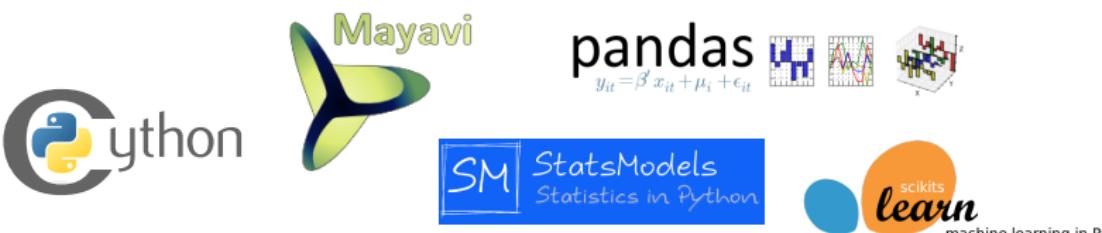
Beyond (Floating Point) Number Crunching





- A flexible, efficient, multidimensional array object.
- Convenient syntax: `c = a+b.`
- Math library that operates on arrays: `y = sin(k*t).`
- Basic scientific functionality:
 - Linear algebra
 - FFTs
 - Random number generation

Scientific Python: a Rich Ecosystem



Outline

1 Scientific Python

2 IPython: Interactive Python

Computing is **not the 'third branch' of science...**

It is now the backbone of theory *and* experiment!

Computing in science **must** improve drastically before we can really call it **scientific**.

Computing is **not the 'third branch' of science...**

It is now the backbone of theory *and* experiment!

Computing in science **must** improve drastically before we can really call it **scientific**.

The Lifecycle of a Scientific Idea (schematically)

- ① Individual exploratory work
- ② Collaborative development
- ③ Production work (HPC, cloud, parallel)
- ④ Publication (with reproducible results!)
- ⑤ Education
- ⑥ Goto 1.

The Problem with most tools

Barriers and discontinuities in workflow in between all the steps

The Lifecycle of a Scientific Idea (schematically)

- ① Individual exploratory work
- ② Collaborative development
- ③ Production work (HPC, cloud, parallel)
- ④ Publication (with reproducible results!)
- ⑤ Education
- ⑥ Goto 1.

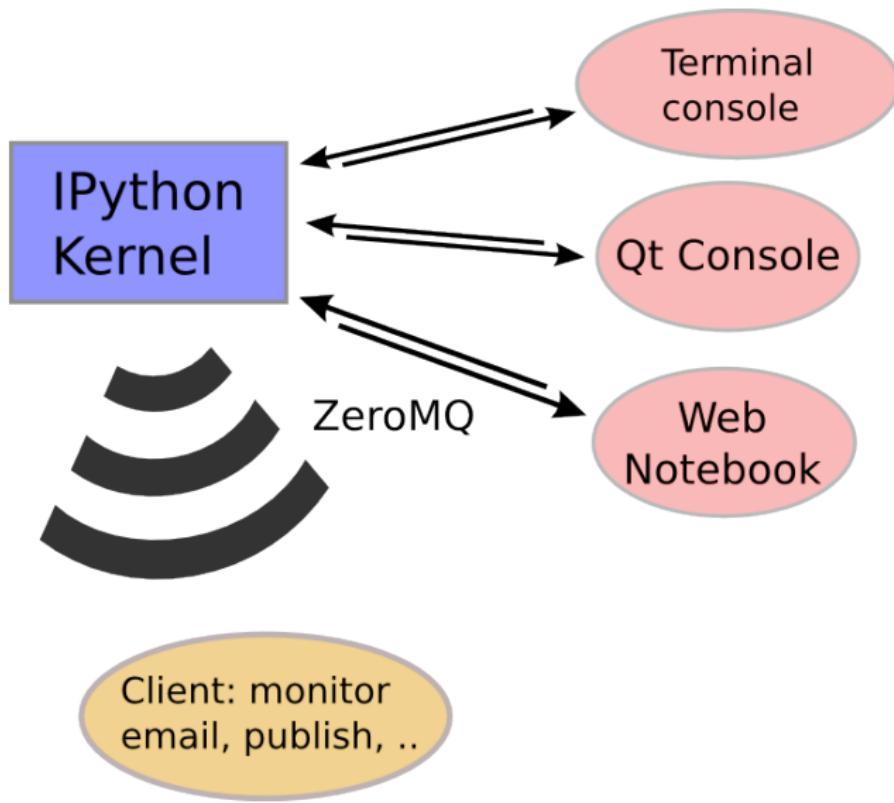
The Problem with most tools

Barriers and discontinuities in workflow in between all the steps

IPython's goal:
Fluid transitions in all these steps

Demo

Pillar #1: An architecture for interactive computing



Pillar #2: the Notebook Format

- **JSON** but version control-friendly
- Easy for machine processing, **fixable by hand if need be.**
- Lots of hooks for **metadata**
- **Not Python-specific** (R and Ruby notebooks exist, Julia planned)
- Produce Markdown, reST, L^AT_EX, HTML, etc...

An open format for sharing, publishing and archiving executable computational work

Documented protocols and formats: a growing ecosystem around IPython

Microsoft Visual Studio 2010 integrated console

Dino Viehland and Shahrokh Mortazavi (Microsoft); <http://pytools.codeplex.com>

The screenshot shows the Microsoft Visual Studio 2010 IDE interface with the following components visible:

- Title Bar:** MpiDemo - Microsoft Visual Studio
- Menu Bar:** File, Edit, View, Refactor, Project, Build, Debug, Team, Data, Tools, Architecture, Test, Analyze, Window, Help
- Toolbar:** Standard development tools like Open, Save, Print, Find, Replace, etc.
- Toolbox:** Standard .NET development tools.
- Solution Explorer:** Shows the Solution 'MpiDemo' (1 project) with files MpiDemo, Search Path, and Program.py.
- Properties Explorer:** Standard .NET properties window.
- Python 2.7 Interactive Window:** Displays the following Python code and its execution results:

```
In [1]: from IPython.parallel import Client
In [2]: rc = Client()
In [3]: rc.ids
[0, 1, 2, 3]
In [4]: dview = rc[:] # use all engines
In [5]: serial_result = map(lambda x:x**10, range(32))
In [6]: parallel_result = dview.map_sync(lambda x:
x**10, range(32))
In [7]: serial_result==parallel_result
True
In [8]: parallel_result
[0,
 1,
 1024,
 59049,
 1048576,
 9765625,
 60466176,
 282475249,
 1073741824,
 3486784401L,
 10000000000L,
 25937424601L,
 61917364224L,
 137858491849L,
 289254654976L,
 576650309625L]
```

- Code Editor:** Shows the contents of Program.py, which includes imports for random, mpi4py, MPI, and numpy, and defines a computePi function using MPI's parallel map.
- Status Bar:** Shows the current file (Program.py), line (Ln 6), column (Col 14), character (Ch 14), and mode (INS).

A vim client to control an IPython kernel/console

Paul Ivanov (Berkeley), <https://github.com/ivanov/vim-ipython>

The screenshot shows a terminal window with two panes. The left pane is a vim editor displaying Python code in `test.py`. The right pane is an IPython console.

Vim Editor Content (`test.py`):

```
pi@ykyc: ~/code/vim... test.py (vim) IPython
- Visual Mode
# It also works blockwise in Visual Mode.
# Select the next block and send it to IPython

import this,math # secret decoder ring
a,b,c,d,e,f,g,h,i = range(1,10)
code =(c,a,d,a,e,i)
msg = ...jrer myy frag sebz Ivv.\nIvv+wCigubamhfy) +this.s.split()[g]
decode=lambda x:"\n"+''.join([this.d.get(c,c) for c in x])+''
format=lambda x: These lines\n +"\n ".join([l for l in x.splitlines()])
secret_decoder = lambda a,b: format(a)+decode(msg)%str(b)[-1]
'Md *len(code)%code == str(int(math.pi*1e5))

# Then, go to the qtconsole and run this line
print secret_decoder(_,_)

##
+-- 4 lines: Running whole files
##
+-- 22 lines: IPython's object? Functionality
+-- 33 lines: IPython's tab-completion Functionality
```

IPython Console Output:

```
In [4]: <module> US FROM /usr/lib/python2.6/os.pyc >
In [3]: from vim
-----
NameError: name 'from_vim' is not defined
Traceback (most recent call last)
/home/pi/code/vim-ipython/<ipython-input-3-72ba696b9008> in <module>()
----> 1 from_vim

NameError: name 'from_vim' is not defined

In [5]: from vim
Out[5]: 1

In [7]: from vim
Out[7]: 14

In [8]:
18:44@vim-ipython master $ gvim README.rst
18:45@vim-ipython master $ ipython qtconsole
[IPKernelApp] To connect another client to this kernel, use:
[IPKernelApp] --existing --shell=35153 --iopub=41748 --stdin=50006 --hb=43406
```

Notebooks on Windows Azure Cloud

Shahrokh Mortazavi (Microsoft), B.G., F.P.: <http://bit.ly/JQeojd>.

The screenshot shows a Microsoft Edge browser window. The address bar displays the URL <https://www.windowsazure.com/en-us/develop/python/tutorials/ipython-notebook/>. The page content is titled "IPython Notebook on Windows Azure". It includes a video thumbnail with a play button, social sharing icons for Facebook, Twitter, and LinkedIn, and a "Free trial" button. On the left, there's a sidebar with sections for "WINDOWS AZURE INTRO", "TUTORIALS" (listing "Web with Django", "Web with Django + MySQL", "Django with Visual Studio", "IPython Notebook", and a "show all" link), "HOW TO GUIDES" (listing "Blob Service", "Table Service", "Queue Service", "Service Bus Queues", "Service Bus Topics", "Command Line Tools", and a "show all" link), "COMMON TASKS" (listing "Install Python", "SQL Database Management", "Custom DNS", "Enable Remote Desktop", "Enable SSL", "CDN", "Staging Deployment", and a "show all" link), and "BEST PRACTICES" (listing "Troubleshooting", "Security", and "Performance"). The main content area shows a screenshot of the IPython Notebook interface with a title "IP[y]: Notebook" and a section titled "Simple spectral analysis".

IPython Notebook on Windows Azure

For a quick overview of installation and IPython, please watch:

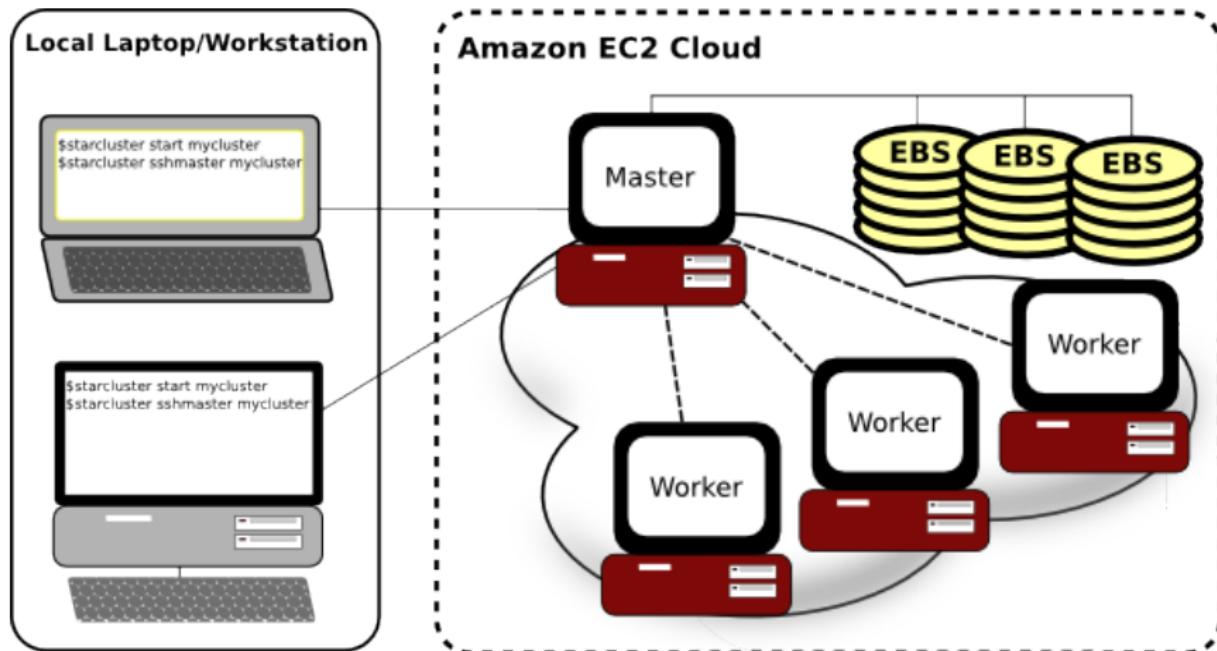
The [IPython project](#) provides a collection of tools for scientific computing that include powerful interactive shells, high-performance and easy to use parallel libraries and a web-based environment called the IPython Notebook. The Notebook provides a working environment for interactive computing that combines code execution with the creation of a live computational document. These notebook files can contain arbitrary text, mathematical formulas, input code, results, graphics, videos and any other kind of media that a modern web browser is capable of displaying.

Whether you're absolutely new to Python and want to learn it in a fun, interactive environment or do some serious parallel/technical computing, the IPython Notebook is a great choice. As an illustration of its capabilities, the following screenshot shows the IPython Notebook being used, in combination with the SciPy and matplotlib packages, to analyze the structure of a sound recording:

The screenshot shows the IPython Notebook interface running in a browser. The title bar says "IP[y]: Notebook" and "spectrogram". The URL in the address bar is <https://127.0.0.1:8888/bed6aac-d118-4521-96e7-3f5054b6641d>. The notebook interface has a menu bar with File, Edit, View, Insert, Cell, Kernel, Help. Below the menu is a toolbar with various icons. A section titled "Simple spectral analysis" is visible, showing a plot of a spectrogram.

Star Cluster: IPython parallel+Notebook on Amazon EC2

Justin Riley (MIT): <http://web.mit.edu/star/cluster>



One-click single notebook on Amazon EC2

Carl Smith (UK): <https://notebookcloud.appspot.com>.

The screenshot shows the NotebookCloud web interface. At the top, there's a navigation bar with links for "NBCloud Docs", "IPython Docs", "Account Details", and "Delete Account". On the right side of the bar, the email "fdo.perez@gmail.com" is displayed. Below the navigation bar, there's a section titled "Instance Launcher" with a grid of buttons for selecting instance types. The buttons are arranged in three rows: the first row contains "Micro", "Small", "Medium", and "Large"; the second row contains "XLarge"; the third row contains "High Memory", "High Memory x 2", "High Memory x 4", and "High CPU Medium". Below these, another row of buttons includes "High CPU XLarge", "GPU Cluster", "Super Cluster", "Death Star", and "Hello World". A note on the right side of the launcher says: "Select any instance type to see more information about it. If you are unsure about your options, please consult the documentation. If you are using NotebookCloud for learning programming, you should select Hello World." At the bottom left, a section titled "Your NotebookCloud Servers" displays the message "No instances (launched from your NotebookCloud account) exist on your AWS account." At the very bottom of the page, there are links for "AWS Website", "IPython Website", "Python Website", and "Carl Smith".

NotebookCloud

<https://notebookcloud.appspot.com>

NBCloud Docs IPython Docs Account Details Delete Account fdo.perez@gmail.com

Micro Small Medium Large

XLarge

High Memory High Memory x 2 High Memory x 4 High CPU Medium

High CPU XLarge

GPU Cluster Super Cluster Death Star Hello World

Instance Launcher

Select any instance type to see more information about it. If you are unsure about your options, please consult the documentation.

If you are using NotebookCloud for learning programming, you should select Hello World.

Your NotebookCloud Servers

No instances (launched from your NotebookCloud account) exist on your AWS account.

AWS Website IPython Website Python Website Carl Smith

Other projects using IPython

Scientific

- **Software Carpentry**
- **EPD**: Enthought Python Distribution.
- **Continuum**: Anaconda, Wakari.
- **Sage**: open source mathematics.
- **PyRAF**: Space Telescope Science Institute
- **CASA**: Nat. Radio Astronomy Observatory
- **Ganga**: CERN
- **PyMAD**: neutron spectrom., Laue Langevin
- **Sardana**: European Synchrotron Radiation
- **ASCEND**: eng. modeling (Carnegie Mellon).
- **JModelica**: dynamical systems.
- **DASH**: Denver Aerosol Sources and Health.
- **Trilinos**: Sandia National Lab.
- **DoD**: baseline configuration.
- **Mayavi**: 3d visualization, Enthought.
- **NiPype**: computational pipelines, MIT.

Web/Other

- **Visual Studio 2010**: MS.
- **Django**.
- **Turbo Gears**.
- **Pylons** web framework
- **Zope** and **Plone** CMS.
- Axon Shell, BBC Kamaelia.
- **Schevo** database.
- **Pitz**: distributed task/bug tracking.
- **iVR** (interactive Virtual Reality).
- **Movable Python** (portable Python environment).
- ...

(Incomplete) Cast of Characters

- **Brian Granger** - Physics, Cal State San Luis Obispo
- **Min Ragan-Kelley** - Nuclear Engineering, UC Berkeley
- **Matthias Bussonnier** - Physics, Institut Curie, Paris
- **Brad Froehle** - Mathematics, UC Berkeley
- **Paul Ivanov** - Neuroscience, UC Berkeley.
- **Robert Kern** - Enthought
- **Thomas Kluyver** - Biology, U. Sheffield
- **Jonathan March**- Enthought
- **Evan Patterson** - Physics, Caltech/Enthought
- **Jörgen Stenarson** - Elect. Engineering, Sweden.
- Stefan van der Walt - UC Berkeley
- John Hunter - TradeLink Securities, Chicago.
- Prabhu Ramachandran - Aerospace Engineering, IIT Bombay.
- Satra Ghosh- MIT Neuroscience
- Gaël Varoquaux - Neurospin (Orsay, France)
- Ville Vainio - CS, Tampere University of Technology, Finland
- Barry Wark - Neuroscience, U. Washington.
- Ondrej Certik - Physics, U Nevada Reno
- Darren Dale - Cornell
- Justin Riley - MIT
- Mark Voorhies - UC San Francisco
- Nicholas Rougier - INRIA Nancy Grand Est
- Thomas Spura - Fedora project
- **Many more!** (~150 commit authors)

- **Enthought**, Austin, TX: **Lots!**
- **Microsoft**: WinHPC support, Visual Studio integration, Azure (thanks to [Shahrokh Mortazavi](#)).
- **DoD/DRC Inc**: funding through Sept. 2012 (thanks to [Jose Unpingco](#) and [Chris Keees](#)).
- **NIH**: via NiPy grant
- **NSF**: via Sage compmath grant
- **Google**: summer of code 2005, 2010.
- **Tech-X Corp.**, Boulder, CO: Parallel/notebook (previous versions)



ALFRED P. SLOAN FOUNDATION

2-year funding, core team (7 people)

Thanks!!

- Support the development of multiple projects.
- Community-created and driven.
- A neutral ground for industry, academia and government to support scientific open source.
- 501(c)3 - donations are tax-exempt in the USA

