Scientific Python

Andy Turner, ARCHER CSE Team
a.turner@epcc.ed.ac.uk

Reusing this material

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

Scientific Computing Requirements

- Generate data
  - Usually from simulation on HPC facilities
  - (Also from experiment!)
- Process data
  - Generate appropriate results from simulation data
- Visualise results
  - To understand the significance of our work and gain scientific understanding
- Communicate results
  - Through publications, presentations, web, etc.

archer
epcc
THE UNIVERSITY OF EDINBURGH

Why Python?

- Rich set of scientific computing functionality
  - Powerful numerical and scientific libraries
  - Rich plotting functionality
  - Excellent support for interfacing to existing Fortran/C/C++ code
  - Interactive and scripting interface
- Simple to learn and code is very readable
  - Scientists are usually self-taught programmers
  - Syntax enables clarity in algorithms (in a similar way to Fortran)
- Free and Open Source
  - Widely-available so code is portable

Useful packages

- IPython
  - Advanced Python shell
- Matplotlib
  - Rich featured plotting (2D and 3D)
- Numpy
  - Tools for manipulating numerical arrays efficiently
- Scipy
  - High-level scientific routines for common algorithms: optimisation, Fourier transform, linear algebra and others
- f2py
  - Interface external code with Python
- mpi4py
  - Message passing parallel programming

Python: Interactive and Programs

- Python can be used as an interactive tool
  - For example, when producing simple plots to quickly analyse data
  - The IPython shell adds additional useful functionality
- It can also be used for writing programs
  - These can range from quick-and-dirty single use scripts to full programs
  - Can interface to C/C++ and Fortran code

IPython Shell

- IPython extends the standard Python shell with a number of useful things, including:
  - Tab completion
  - Interactive help
  - Built-in debugging and profiling
  - Pasting of code snippets from websites
  - Saving of sessions
- `quickref` command gives a summary of capabilities