The use of Python in astrophysics for the post-processing and visualization of "star-in-a- box" simulations

J. Morin[1,2], B. Dintrans[2] and T. Gastine[3,2]

[1]Dublin Institute for Advanced Studies, School of Cosmic Physics, 31 Fitzwilliam Place, Dublin 2, Ireland
[2]IRAP-UMR5277, CNRS & Université de Toulouse, 14 avenue Edouard Belin, 31400 Toulouse, France
[3]Max-Planck-Institut für Sonnensystemforschung, Max-Planck-Straße 2, 37191 Katlenburg-Lindau, Germany

Contacts: jmorin@cp.dias.ie, dintrans@ast.obs-mip.fr, gastine@mps.mpg.de

Abstract

Hydrodynamic and magnetohydrodynamic direct numerical simulations are very valuable tools for the study of a number of astrophysical problems. We present here our use of Python, Scipy and various scientific libraries to analyze "star-in-a-box" simulations, prototype post-processing functions and produce publication-quality plots.

The Pencil Code

The Pencil Code (PC) is a high-order finite-difference code for compressible hydrodynamic flows with magnetic fields and particles [1] developed by a core team of ∼ 30 researchers plus additional contributors, and distributed under the GPL. The code is designed with scalability and modularity in mind, and is successfully used to deal with a variety of astrophysical problems such as driven MHD turbulence in a periodic box, accretion disc turbulence in the shearing sheet approximation, self-gravity, non-local radiation transfer, dust particle evolution with feedback on the gas, etc.

Star-in-a-box simulations

Global numerical simulations of fluid motion in stellar interiors are generally carried out in spherical geometries using spectral codes (physical quantities are decomposed on a spherical harmonics basis in angular directions and on Chebishev polynomials in the radial direction) providing a high computational accuracy. During the past decade a complementary approach has emerged, called star-in-a-box, that consists in running global stellar simulations in Cartesian geometry by embedding the star in a cubic box [2].

Spherical harmonics

- Stellar surface = surface of coordinates
- Spectral precision

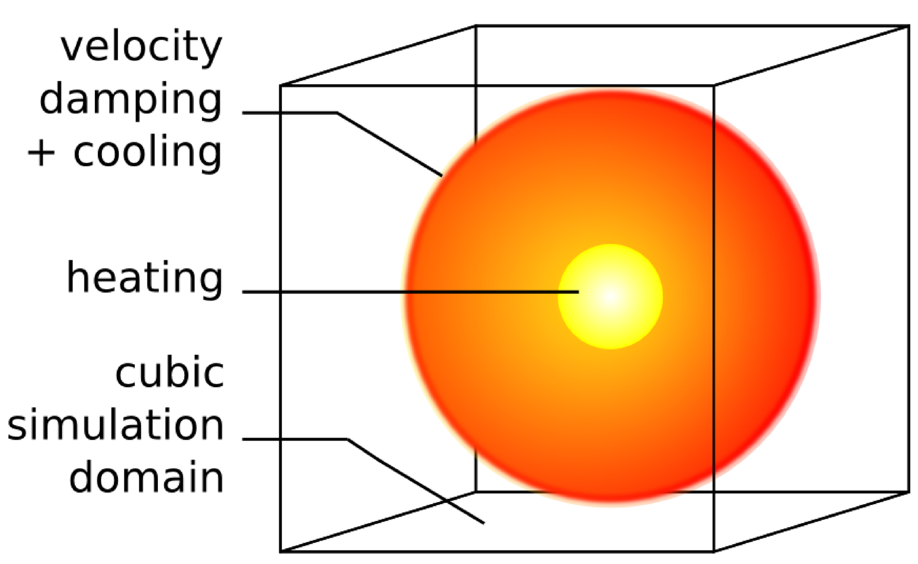
- Central singularity
- Computation of non-linear terms
- Complex parallelization

Star-in-a-box

- No singular point
- Non-linear terms are handled naturally
- Highly parallelizable

- ∼ half grid points lost
- Non-spectral precision
- Boundary conditions
- Heavier post-processing

This method is very well-suited to the study of fully-convective stars – young T Tauri stars, main sequence red dwarfs, or evolved cool giants – in particular the generation of magnetic fields by dynamo effect in these objects [3].



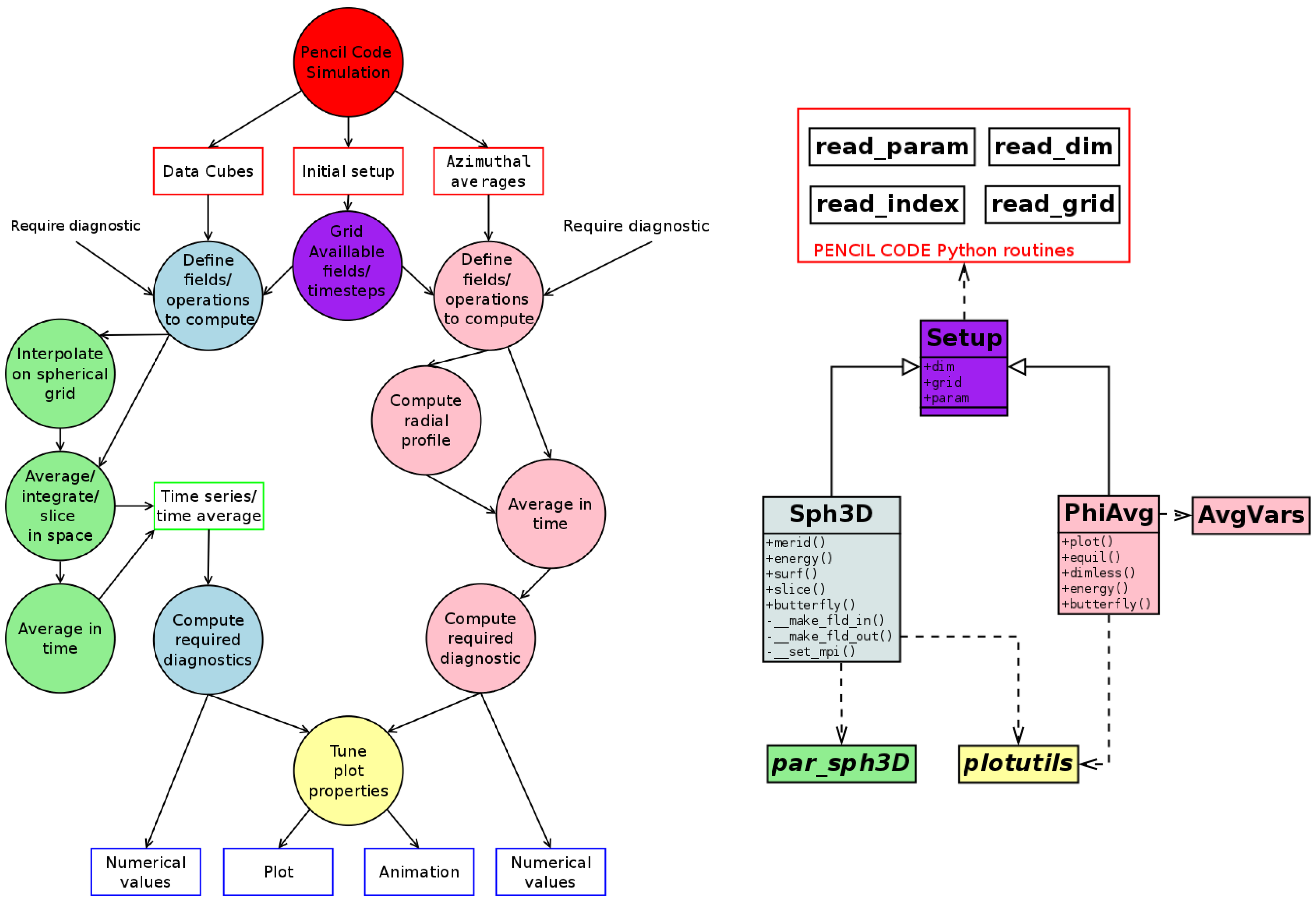
Developing post-processing routines in Python

The Pencil Code is provided with a number of post-processing and plotting routines mostly developed in IDL and for cartesian diagnostics. We focus here on the development of a set of routines dedicated to star-in-a-box simulations in Python, though since Python I/O routines to deal with PC data products are now provided Python-based post-processing is increasingly used in the PC community (*e.g.*, [4]). The crucial factors for the choice of Python are:

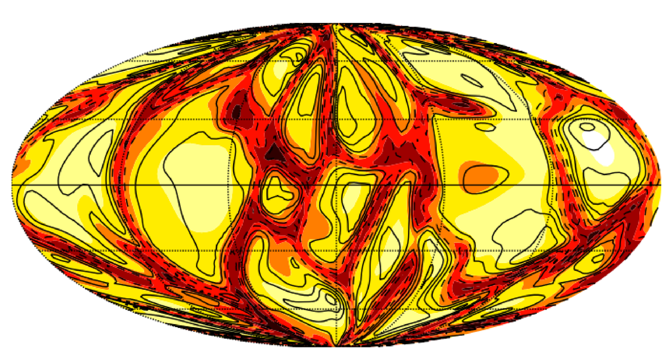
- Python and main libraries are free and libre software
- A number of high-quality libraries are available
  - Numpy/Scipy: array processing, interpolation, integration
  - Matplotlib + basemap toolkit: plots and spherical projections
  - PyPar: interface with MPI
- Performance is generally better than IDL

Data flow and software architecture

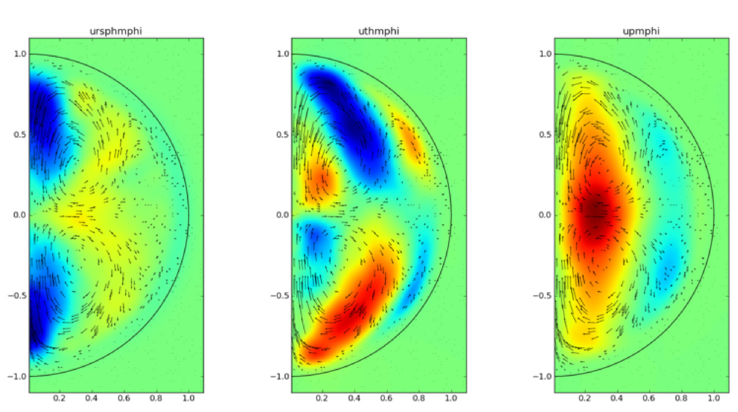
Two processing tools are available: directly from data cubes (`Sph3D`) or from azimuthal averages (`PhiAvg`) computed internally during PC runs. New diagnostics are prototyped with `Sph3D` on low-resolution runs and then implemented in the PC (computation part) and in (`PhiAvg`) (plotting part).



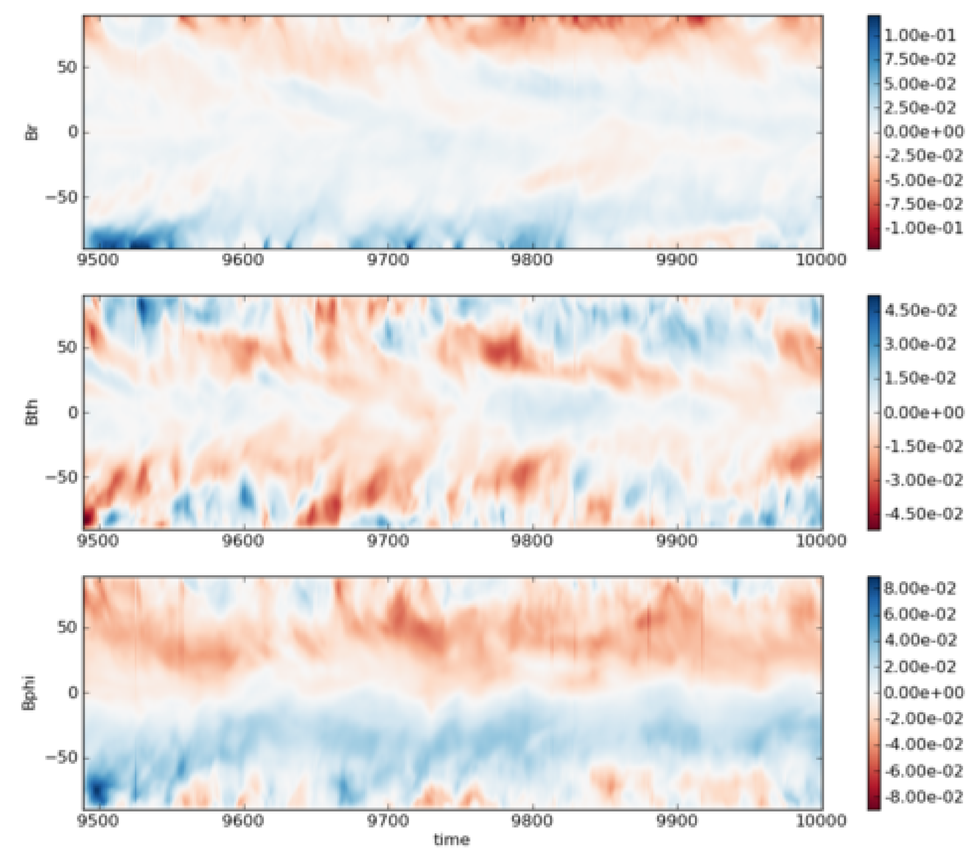
Examples of available plots



Interpolation on a spherical surface



Azimuthal and temporal averages



"Butterfly"-like diagram

See also [5] for additional examples.

Future plans

- Extend usage of PyPar to PhiAvg for high resolution runs.
- Implement spherical harmonics decomposition directly from Cartesian coordinates.
- Extend to 3-D visualization with Mayavi and Pencil Code vtk file converter.
- Release publicly.

Assessment of our Python experience

- Switching to Python/Numpy/SciPy is rather easy thanks to "natural" syntax.
- Array processing performance is good, provided appropriate syntax is used.
- PyPar allows simple data parallelization based on MPI.
- The weak typing of Python combined with optional arguments/default values makes the code very flexible.

References

[1] Brandenburg A., Dobler W., 2002, CoPhC, 147, 471
[2] Freytag B., Steffen M., Dorch B., 2002, AN, 323, 213
[3] Dobler W., Stix M., Brandenburg A., 2006, ApJ, 638, 336
[4] Gastine T., Dintrans B., 2008, A&A, 490, 743
[5] J. Morin, PhD thesis, http://tel.archives-ouvertes.fr/tel-00480428/

Acknowledgements

Calculations were carried out on the CalMip machine of the "Centre Interuniversitaire de Calcul de Toulouse", and at CINES (Centre Informatique National de l'Enseignement Supérieur). We are grateful to the Pencil Code and scientific Python communities for developing Free/Libre software.

Pencil Code resources

- Project homepage: http://code.google.com/p/pencil-code/
- SVN repository: http://pencil-code.googlecode.com/svn/trunk/
- Pencil Code user meeting: http://pencilcode11.obs-mip.fr/index.html