Plonk: Smoothed particle hydrodynamics analysis and visualization with Python

Daniel Mentiplay[1]

1 School of Physics and Astronomy, Monash University

Summary

`Plonk` is a Python package for analysis and visualization of smoothed particle hydrodynamics data focussing on astrophysical applications.

Hydrodynamical simulations are a standard part of the astronomer's toolkit. The knowledge required to run these simulations and analyze the output can be specialized. Many of the codes written to run and analyze these simulations are written in compiled languages such as Fortran and C++. The advantage of using compiled languages is performance. One disadvantage is that expertise in these languages is becoming less widely available than other, typically interpreted, languages such as Python. Access to analysis and visualization of hydrodynamical simulations should be available to a wide audience. Much of the modern astronomy pipeline is written in Python.

Smoothed particle hydrodynamics (SPH) is a method for solving the equations of hydrodynamics (Gingold & Monaghan, 1977; Lucy, 1977; Price, 2012). The fluid is discretized as a set of massive particles. The density field is computed as a sum over particles weighted by a smoothing kernel. The particles are evolved by solving the equations of hydrodynamics written in Lagrangian form.

Visualization of SPH data is complicated by the fact that the data is represented on a set of particles arranged irregularly in space, as opposed to a regular grid. One visualization method is to plot the particles as a scatter plot, possibly with a color to represent some quantity defined on the particles. An alternative is to interpolate quantities defined on the particles to a two-dimensional pixel array. The latter approach provides an accurate representation of the data, in that it uses the same underlying approach to computing density as in the SPH method.

SPH simulations typically output files which contain quantities defined on the particles at a given snapshot in time. The size of these files scales with the number of particles. These snapshot files may also include metadata, such as physical units, time, and other physical and numerical parameters. In addition, output may also include auxiliary files containing globally-averaged quantities output more frequently than snapshot files.

`Plonk` can read data from these various data sets and provide access to the underlying data as `NumPy` arrays. Large arrays are loaded only as required. The entire data set does not need to be read into memory at once. After the array is loaded from file it is cached in memory for later use. `Plonk` can read particles representing gas, dust (including multiple species), or other fluids, as well as sink particles (representing compact objects like stars or planets).

`Plonk` has tools to perform typical tasks associated with the analysis and visualization of smoothed particle hydrodynamics data, such as: calculating extra quantities on the particles, calculating and plotting radial profiles, and visualization of any quantity defined on the particles via kernel density estimation. The interpolation component of the visualization package makes

use of `KDEpy`, a Python package for weighted kernel density estimation using fast Fourier transforms (Odland, 2018). As such it has performance on par with other (compiled) SPH visualization programs such as Splash (Price, 2007).

Given that `Plonk` is a Python package it can be used via a Jupyter notebook (Kluyver et al., 2016). The Jupyter notebook is a platform which allows for exploratory data analysis and visualization, combined with the ability to share the notebook, as a record of analysis, with collaborators. A user can install Plonk, using a package manager such as pip or Conda, on the supercomputer where their data resides, and analyze and visualize their data via a web browser running on their local machine. The expensive computations take place on the remote machine.

Similar existing Python packages for analysis and visualization of SPH (or N-body) data such as `pynbody` (Pontzen, Roškar, Stinson, & Woods, 2013) and `Py-SPHViewer` (Benitez-Llambay, 2015) focus on cosmological simulations. The, more general, scientific analysis and visualization package `yt` (Turk et al., 2011) was originally designed to work with gridded simulation output. The next major version of `yt` (version 4.0) will support analyzing SPH data directly on the particles, rather than via interpolating onto an octree data structure, as in the current version (3.5.1). In contrast, `Plonk` uses the particle data directly and is focussed on astrophysical fluid dynamics simulations, such as protoplanetary and black hole accretion disks, star formation, and molecular clouds. `Plonk` was originally designed with users of the Phantom SPH code (Price et al., 2018) in mind. At the time of writing, `Plonk` is the only open source Python package for analysis and visualization that can read Phantom SPH data.

Acknowledgements

References

Astropy Collaboration. (2013). Astropy: A community Python package for astronomy, *558*. doi:10.1051/0004-6361/201322068

Benitez-Llambay, A. (2015, July). Py-sphviewer: Py-sphviewer v1.0.0. doi:10.5281/zenodo.21703

Collette, A. (2013). *Python and HDF5*. O'Reilly.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars., *181*, 375–389. doi:10.1093/mnras/181.3.375

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides & B. Schmidt, Eds.). IOS Press. doi:10.3233/978-1-61499-649-1-87

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis., *82*, 1013–1024. doi:10.1086/112164

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

Odland, T. (2018). *Tommyod/kdepy: Kernel density estimation in python*. Zenodo. doi:10.5281/zenodo.2392268

Oliphant, T. E. (2006). *A guide to numpy*. USA: Trelgol Publishing.

Pontzen, A., Roškar, R., Stinson, G., & Woods, R. (2013, May). pynbody: N-Body/SPH analysis for python. Astrophysics Source Code Library.

Price, D. J. (2007). splash: An Interactive Visualisation Tool for Smoothed Particle Hydrodynamics Simulations, *24*(3), 159–173. doi:10.1071/AS07022

Price, D. J. (2012). Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, *231*(3), 759–794. doi:10.1016/j.jcp.2010.12.011

Price, D. J., Wurster, J., Tricco, T. S., Nixon, C., Toupin, S., Pettitt, A., Chan, C., et al. (2018). Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics, *35*, e031. doi:10.1017/pasa.2018.25

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., & Norman, M. L. (2011). yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data, *192*(1), 9. doi:10.1088/0067-0049/192/1/9

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science and Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2019). SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, arXiv:1907.10121. Retrieved from http://arxiv.org/abs/1907.10121

Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., et al. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453. doi:10.7717/peerj.453