The Journal of Open Source Software

EXP: a Python/C++ package for basis function expansion methods in galactic dynamics

Michael S. Petersen [1] **and Martin D. Weinberg** [2]

1 University of Edinburgh, UK **2** University of Massachusetts/Amherst, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Galaxies are ensembles of dark and baryonic matter confined by their mutual gravitational attraction. The dynamics of galactic evolution have been studied with a combination of numerical simulations and analytical methods (Binney & Tremaine, 2008) for decades. Recently, data describing the positions and motions of billions of stars from the *Gaia satellite* (Gaia Collaboration, 2016, 2018) depict a Milky Way (our home galaxy) much farther from equilibrium than we imagined and beyond the range of many analytic models. Simulations that allow collections of bodies to evolve under their mutual gravity are capable of reproducing such complexities but robust links to fundamental theoretical explanations are still missing.

Basis Function Expansions (BFE) represent fields as a linear combination of orthogonal functions. BFEs are particularly well-suited for studies of perturbations from equilibrium, such as the evolution of a galaxy. For any galaxy simulation, a biorthogonal BFE can fully represent the density, potential and forces by time series of coefficients. The coefficients have physical meaning: they represent the gravitational potential energy in a given function. The variation the function coefficients in time encodes the dynamical evolution. The representation of simulation data by BFE results in huge compression of the information in the dynamical fields; for example, 1.5 TB of phase space data enumerating the positions and velocities of millions of particles becomes 200 MB of coefficient data!

For optimal representation, the lowest-order basis function should be similar to the mean or equilibrium profile of the galaxy. This allows for use of the fewest number of terms in the expansion. For example, the often-used basis set from Hernquist & Ostriker (1992) matches the Hernquist (1990) dark matter halo profile, yet this basis set is inappropriate for representing the cosmologically-motivated Navarro et al. (1997) profile. The EXP software package implements the adaptive empirical orthogonal function (EOF; see Figure 1) basis strategy originally described in Weinberg (1999) that matches any physical system close to an equilibrium model. The package includes both a high performance N-body simulation toolkit with computational effort scaling linearly with N (Petersen et al., 2022), and a user-friendly Python interface called pyEXP that enables BFE and time-series analysis of any N-body simulation dataset.

Statement of need

The need for methodology that seamlessly connects theoretical descriptions of dynamics, N-body simulations, and compact descriptions of observed data gave rise to EXP. This package provides recent developments from applied mathematics and numerical computation to represent complete series of *Basis Function Expansions* that describe the variation of *any* field in space. In the context of galactic dynamics, these fields may be density, potential, force, velocity fields or any intrinsic field produced by simulations such as chemistry data. By combining the coefficient information through time using multichannel singular spectral analysis (mSSA; Golyandina et al. (2001)), a non-parametric spectral technique, EXP can deepen our understanding by discovering the dynamics of galaxy evolution directly from simulated, and by analogy, observed data.

EXP decomposes a galaxy into multiple bases for a variety of scales and geometries and is thus able to represent arbitrarily complex simulation with many components (e.g., disk, bulge, dark matter halo, satellites). EXP is able to efficiently summarize the degree and nature of asymmetries through coefficient amplitudes tracked through time and provide details at multiple scales. The amplitudes themselves enable ex-post-facto dynamical discovery. EXP is a collection of object-oriented C++ libraries with an associated modular N-body code and a suite of stand-alone analysis applications.

pyEXP provides a full Python interface to the EXP libraries, implemented with `pybind11` (Jakob et al., 2017), which provides full interoperability with major astronomical packages including Astropy (Astropy Collaboration, 2013) and `Gala` (Price-Whelan, 2017). Example workflows based on previously published work are available and distributed as accompanying examples and tutorials. The examples and tutorials flatten the learning curve for adopting BFE tools to generate and analyze the significance of coefficients and discover dynamical relationships using time series analysis such as mSSA. We provide a full online manual hosted by ReadTheDocs.

The software package brings published – but difficult to implement – applied-math technologies into the astronomical mainstream. EXP and the associated Python interface pyEXP accomplish this by providing tools integrated with the Python ecosystem, and in particular are well-suited for interactive Python (Pérez & Granger, 2007) use through (e.g.) Jupyter notebooks (Kluyver et al., 2016). EXP serves as the scaffolding for new imaginative applications in galactic dynamics, providing a common dynamical language for simulations and analytic theory.

Features and workflow

The core EXP library is built around methods to build the best basis function expansion for an arbitrary data set in galactic dynamics. The table below lists some of the available basis functions. All computed bases and resulting coefficient data are stored in HDF5 (The HDF Group, 2000-2010) format.

Name	Description
sphereSL	Sturm-Liouville basis function solutions to Poisson's equation for any arbitrary input spherical density
bessel	Basis constructed from eigenfunctions of the spherical Laplacian
cylinder	EOF solutions tabulated on the meridional plane for distributions with cylindrical geometries
flatdisk	EOF basis solutions for the three-dimensional gravitational field of a razor-thin disk
cube	Trigonometric basis solution for expansions in a cube with boundary criteria
field	General-purpose EOF solution for scalar profiles
velocity	EOF solution for velocity flow coefficients

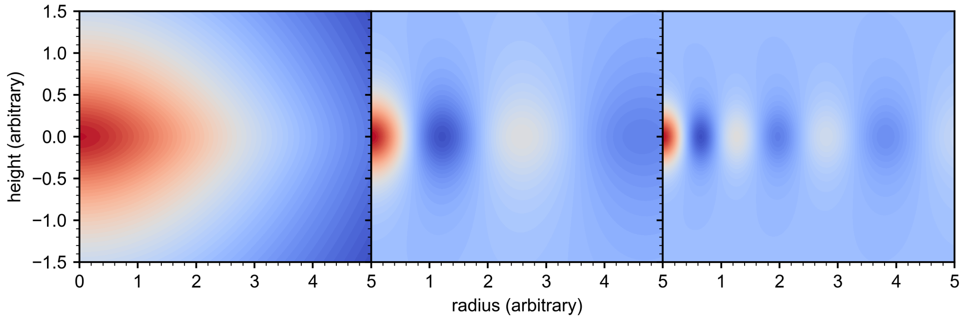

Figure 1: Example cylinder basis functions, where the color encodes the amplitude of the function, for an exponential disk with a scalelength of 3 and a scaleheight of 0.3 in arbitrary units. We select three functions at low, medium, and higher order (corresponding to the number of nodes). The color scale has been normalised such that the largest amplitude is unity in each panel.

N-body simulation

Computing the gravitational potential and forces from a collection of N particles is typically an expensive endeavour. EXP reduces the cost by using BFE to compute the potential and forces such that computational effort scales with the number of particles. Other modern N-body codes use direct summation (Wang et al., 2015) or tree-based solutions (Springel et al., 2021), which have computational effort that scales as N^2 and N log N, respectively. The trade off for BFE solutions comes in the form of restricted degrees of freedom; for many problems in near-equilibrium galactic dynamics this is not a problem, but rather a feature.

Our design includes a wide choice of run-time summary diagnostics, phase-space output formats, dynamically loadable user libraries, and easy extensibility. Stand-alone routines include the EOF and mSSA methods described above, and the modular software architecture of `EXP` enables users to easily build and maintain extensions. The `EXP` code base is described in published papers (Petersen et al., 2022; Weinberg, 2023) and has been used, enhanced, and rigorously tested for nearly two decades.

The design and implementation of the N-body tools allows for execution on a wide variety of hardware, from personal laptops to high performance computing centers, with communication between processes handled by MPI (Message Passing Interface Forum, 2023) and GPU implementations in CUDA (NVIDIA et al., 2020). Owing to the linear scaling of computational effort with N and the novel GPU implementation, the N-body methods in `EXP` deliver performance in collisionless N-body simulations previously only accessible with large dedicated CPU clusters.

The flexible N-body software design allows users to write their own modules for on-the-fly execution during N-body integration. Modules enable powerful and intricate dynamical experiments in N-body simulations, further reducing the gap between numerical simulations and analytic dynamics. The package ships with several examples, including imposed external potentials, as well as a basic example that can be extended by users.

Using pyEXP to represent simulations

`pyEXP` provides an interface to many of the classes in the `EXP` C++ library, allowing for both the generation of all bases listed in the table above as well as coefficients for an input data set. Each of these tools are Python classes that accept `numpy` (Harris et al., 2020) arrays for immediate interoperability with `matplotlib` (Hunter, 2007) and Astropy. We include a verified set of stand-alone routines that read phase-space files from many major cosmological tree codes (for example, Springel et al. (2021)) and produce BFE-based analyses. The code suite includes adapters for reading and writing phase space for many of the widely used cosmology codes, with a base class for developing new ones. There are multiple ways to use the versatile and modular tools in pyEXP, and we anticipate pipelines that we have not yet imagined. The flexibility of the basis sets available in EXP greatly enhances the number of available basis sets implemented in Python (see, e.g. Price-Whelan (2017)).

Using pyEXP to analyze time series

The EXP library includes multiple time series analysis tools, documented in the manual. Here, we briefly highlight one technique that we have already used in published work: mSSA (Johnson et al., 2023; Weinberg & Petersen, 2021). Beginning with coefficient series from the previous tools, mSSA summarizes signals *in time* that describes dynamically correlated responses and patterns. Essentially, this is BFE in time and space. These temporal and spatial patterns allow users to better identify dynamical mechanisms and enable intercomparisons and filtering for features in simulation suites; e.g. computing the fraction galaxies with grand design structure or hosting bars. Random-matrix techniques for singular-value decomposition ensure that analyses of large data sets is possible. All mSSA decompositions are saved in HDF5 format for reuse.

Acknowledgements

We acknowledge the ongoing support of the B-BFE collaboration. We also acknowledge the support of the Center for Computational Astrophysics (CCA). The CCA is part of the Flatiron Institute, funded by the Simons Foundation. We thank Robert Blackwell for invaluable help with HPC best practices.

References

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

Binney, J., & Tremaine, S. (2008). *Galactic Dynamics: Second Edition*. Princeton University Press. http://adsabs.harvard.edu/abs/2008gady.book.....B

Gaia Collaboration. (2016). The Gaia mission. *Astronomy and Astrophysics*, *595*. https://doi.org/10.1051/0004-6361/201629272

Gaia Collaboration. (2018). Gaia Data Release 2. Mapping the Milky Way disc kinematics. *A&Ap*, *616*, A11. https://doi.org/10.1051/0004-6361/201832865

Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. A. (2001). *Analysis of time series structure: SSA and related techniques*. CRC press.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hernquist, L. (1990). An Analytical Model for Spherical Galaxies and Bulges. *356*, 359. https://doi.org/10.1086/168845

Hernquist, L., & Ostriker, J. P. (1992). A Self-consistent Field Method for Galactic Dynamics. *386*, 375. https://doi.org/10.1086/171025

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jakob, W., Rhinelander, J., & Moldovan, D. (2017). *pybind11 – seamless operability between c++11 and python*.

Johnson, A. C., Petersen, M. S., Johnston, K. V., & Weinberg, M. D. (2023). Dynamical data mining captures disc-halo couplings that structure galaxies. *MNRAS*, *521*(2), 1757–1774. https://doi.org/10.1093/mnras/stad485

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & Jupyter Development Team. (2016). Jupyter Notebooks—a publishing format for reproducible computational workflows. In *IOS press* (pp. 87–90). https://doi.org/10.3233/978-1-61499-649-1-87

Message Passing Interface Forum. (2023). *MPI: A message-passing interface standard version 4.1*. https://www.mpi-forum.org/docs/mpi-4.1/mpi41-report.pdf

Navarro, J. F., Frenk, C. S., & White, S. D. M. (1997). A Universal Density Profile from Hierarchical Clustering. *490*(2), 493–508. https://doi.org/10.1086/304888

NVIDIA, Vingelmann, P., & Fitzek, F. H. P. (2020). *CUDA, release: 10.2.89*. https://developer.nvidia.com/cuda-toolkit

Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science and Engineering*, *9*(3), 21–29. https://doi.org/10.1109/MCSE.2007.53

Petersen, M. S., Weinberg, M. D., & Katz, N. (2022). EXP: N-body integration using basis function expansions. *510*(4), 6201–6217. https://doi.org/10.1093/mnras/stab3639

Price-Whelan, A. M. (2017). Gala: A python package for galactic dynamics. *The Journal of Open Source Software*, *2*(18). https://doi.org/10.21105/joss.00388

Springel, V., Pakmor, R., Zier, O., & Reinecke, M. (2021). Simulating cosmic structure formation with the GADGET-4 code. *506*(2), 2871–2949. https://doi.org/10.1093/mnras/stab1855

The HDF Group. (2000-2010). *Hierarchical data format version 5*. http://www.hdfgroup.org/HDF5.

Wang, L., Spurzem, R., Aarseth, S., Nitadori, K., Berczik, P., Kouwenhoven, M. B. N., & Naab, T. (2015). NBODY6++GPU: ready for the gravitational million-body problem. *450*(4), 4070–4080. https://doi.org/10.1093/mnras/stv817

Weinberg, M. D. (1999). An Adaptive Algorithm for N-Body Field Expansions. *117*(1), 629–637. https://doi.org/10.1086/300669

Weinberg, M. D. (2023). New dipole instabilities in spherical stellar systems. *MNRAS*, *525*(4), 4962–4975. https://doi.org/10.1093/mnras/stad2591

Weinberg, M. D., & Petersen, M. S. (2021). Using multichannel singular spectrum analysis to study galaxy dynamics. *MNRAS*, *501*(4), 5408–5423. https://doi.org/10.1093/mnras/staa3997