GALPY: Galactic Orbits in Python

Christopher Sosa - 5/18/2025

2. Package: galpy
Galpy is a Python package for galactic dynamics. It allows users to simulate, integrate, and visualize orbits in different gravitational potentials. It includes models for the Milky Way, integration tools, and analysis utilities.

3. I selected galpy because it's one of the most widely used galactic dynamics libraries in Python and is listed as 'approved' in the PHYS265 project list. It also provides accessible documentation and examples.

4. Galpy was first released in December of 2014. It was developed by Jo Bovy and is influenced by earlier codes used for integrating stellar orbits. The version I installed was obtained using `import galpy; print(galpy.__version__)` -> 1.8.1.

5. Yes, galpy is still maintained and regularly updated by Jo Bovy and other contributors. Contribution guidelines are available on its GitHub repository. The list of contributors is in the AUTHORS.txt document

6. It was straightforward to install via pip using `!pip install galpy` in a Jupyter notebook. On Windows, it worked without requiring compilation.

7. Yes, galpy installs cleanly using pip. No additional dependencies were needed for basic usage. Some more complex functions may need C copiled extensions but for the purposes of this project none were used.

8. Yes, the source code is available on GitHub at https://github.com/jobovy/galpy.

9. Galpy is used in other tools and packages related to galactic structure and orbit analysis. Example: The `gala` package cites and builds upon similar structures.

10. It is used via Python scripts or Jupyter notebooks. There is no command-line interface or GUI.

11. See Jupyter notebook attached -This code uses Galpy to simulate a test star orbiting in a simplified galactic potential (the Isochrone potential), and computes the orbital actions which are conserved quantities used in galactic dynamics.

12. Yes, galpy produces matplotlib-based plots natively using `Orbit.plot()` and related methods.

13. Yes. See Figure 1 at the bottom of the report or in the notebook. Two figures were produced in the example I made, but I chose the second one. The figure shows the shape of the test star's orbit as it wraps around the galaxy.

14. Galpy is pure Python but uses compiled extensions in C for performance. The pip installer manages some of this seamlessly. Others give an error when no compiled C extensions are available, which can limit functionality.

15. The input includes orbit initial conditions (position, velocity, etc.), gravitational potential models, time Array for Orbit Integration or optional inputs for Action and angles.

16. The output includes orbit trajectories, plots, and computed properties like eccentricity or radial velocity.

17. Yes, galpy includes automated tests and benchmarks. Tests are found in its GitHub repo under `/tests`.

18. The reliability is supported by the extensive test suite and its use in peer-reviewed research. There are amazing papers that use Galpy. See below 'Papers using galpy' in https://docs.galpy.org/en/v1.10.2/

19. Main dependencies: matplotlib, numpy, packaging, scipy. Found via documentation and `!pip shows galpy` again in the jupyterlab notebook.

20. Documentation is available at https://docs.galpy.org and was sufficient for this project.

21. The user is asked to cite the following paper: *galpy: A Python Library for Galactic Dynamics*, Jo Bovy (2015), *Astrophys. J. Supp.*, **216**, 29 (arXiv/1412.3451). It also asks the user to cite the github repository http://github.com/jobovy/galpy.

22. Other references: GitHub docs, ADS citation tool for galpy, galpy tutorials.

23. Two papers:
-Fritz, T. K., et al. *"The Orbit and Origin of the Ultra-faint Dwarf Galaxy Segue 1."* The Astrophysical Journal, vol. 860, no. 2, 2018, p. 164. https://arxiv.org/abs/1711.09097

-Marchetti, T., et al. *"The Gaia Challenge: Prospects for Detection of Hypervelocity Stars."* Monthly Notices of the Royal Astronomical Society, vol. 490, no. 1, 2019, pp. 1570–1583. https://ui.adsabs.harvard.edu/abs/2018MNRAS.476.4697M/abstract

24. Yes - especially astropy unit handling, orbit objects, and integration syntax were new.

25. This was my first time using galpy. All results and code were produced for this PHYS265 project.

FIGURE 1: