Ethan Safra
May 18, 2025

galpy: From Galactic Parameters to Graphs

[1, 2] I chose galpy, a Python package for galactic dynamics, for my final project. I was first intrigued by galpy since it can compute a very wide range of astronomical computations. galpy works with a variety of potentials. It can plot rotation curves, compute orbit integrations, evaluate and sample distribution functions, and calculate action-angle coordinates. galpy can also easily perform a wide range of unit conversions.

[3] Before galpy's release, streamtools was most commonly used for galactic dynamics, namely for modeling stream dynamics in galaxies. In January 2014, galpy (version 0.1) was first released. In July 2022, streamtools' particle-spray model was incorporated into galpy (v1.8.0) for the `galpy.df.streamspraydf` model. I have installed and used version 1.10.2.

galpy is similar to Gala, another popular package for galactic dynamics. Gala also computes galactic potentials and performs orbit integrations. Additionally, Gala can perform dynamical coordinate transformations and compute chaos indicators for nonlinear dynamics.

[4, 7, 16, 17, 19] galpy has been maintained by Professor Jo Bovy, its original author. galpy.org provides some basic information about galpy. galpy's "highly complete documentation" (docs.galpy.org/en/stable/#, last updated March 3, 2025) can thoroughly guide users. galpy's GitHub (github.com/jobovy/galpy) allows users to view and contribute (see CONTRIBUTING.md) to the code. galpy also has a Slack community (galpy.slack.com/) to help others contribute.

galpy boasts "very high test coverage," and clicking the "Actions" tab on GitHub proves this to be true. However, there are a handful of discrepancies between what my code produced and what is on the Introduction page of galpy's documentation, suggesting that either the code or the documentation might not be perfectly maintained. Additionally, the "Module Index" hyperlink on the documentation's home page brings users to a 404 error page.

[5, 6] Installing galpy came with some minor complications. In accordance with the documentation's installation page, I installed galpy with the command

```
!conda install -c conda-forge gsl galpy
```

To get around a handful of errors, I had to open the terminal. There, I found that I needed to install gsl and update ca-certificates, certifi, conda, and OpenSSL in order to install galpy. At this point, installation became straightforward.

[8] galpy is used in many other programming packages. StreamTools, for example, was re-released, using galpy to model and analyze stellar streams. TorusMapper, a package that calculates celestial objects' positions and velocities given actions and angles, uses galpy to compute the galactic potentials. pynbody, a package that simulates and analyzes N-body simulations, also uses galpy to compute galactic potentials.

[9] galpy is a Python package, meaning it can be used only in Python scripts and Python notebooks. I have been using galpy in a JupyterLab Python notebook.

[10] Below is an example of how to use galpy to make simple calculations and unit conversions. As mentioned, there are some peculiar discrepancies between my results and the expected results listed in the documentation. Nonetheless, unit conversions with galpy are very straightforward and helpful.

```
# galpy's natural/internal units are such that circular velocity =1 at R=1

from galpy.util import conversion
# In natural units, orbital time at R=1 is 2pi
print("circular orbital time for v=220 km/s and R=8 kpc is ",
   2.*numpy.pi*conversion.time_in_Gyr(220.,8.), " Gyr")
# Should be 0.223405444283 (about 223 Myr
```

```
circular orbital time for v=220 km/s and R=8 kpc is  0.22340544439051707
Gyr
```

```
from galpy.potential import MWPotential2014, evaluatezforces
print("vertical force 1.1 kpc above the galactic plane (v=220 km/s, R=8 kpc)
   is ", -evaluatezforces(MWPotential2014, 1., 1.1/8.)
   *conversion.force_in_pcMyr2(220.,8.), "pc/Myr^2")
# expect 2.0259181908629933
```

```
vertical force 1.1 kpc above the galactic plane (v=220 km/s, R=8 kpc) is
2.0259181889046634 pc/Myr^2
```

```
print("vertical force 1.1 kpc above the galactic plane (v=220 km/s, R=8 kpc)
   is ", -evaluatezforces(MWPotential2014, 1., 1.1/8.)
   *conversion.force_in_2piGmsolpc2(220.,8.), "2pi*G*M_solar/pc^2")
# force/acceleration in terms of G, M_solar, pc
# expect 71.658016957792356
```

```
vertical force 1.1 kpc above the galactic plane (v=220 km/s, R=8 kpc) is
71.67605643105574 2pi*G*M_solar/pc^2
```

```
print("local dark matter density at R=8 kpc (v=220 km/s) is ",
      MWPotential2014[2].dens(1., 0.)*conversion.dens_in_msolpc3(220., 8.),
   " M_sol/pc^3")
# expect 0.0075419566970079373
```

```
local dark matter density at R=8 kpc (v=220 km/s) is  0.0075438553390854805
```

```
M_sol/pc^3
```

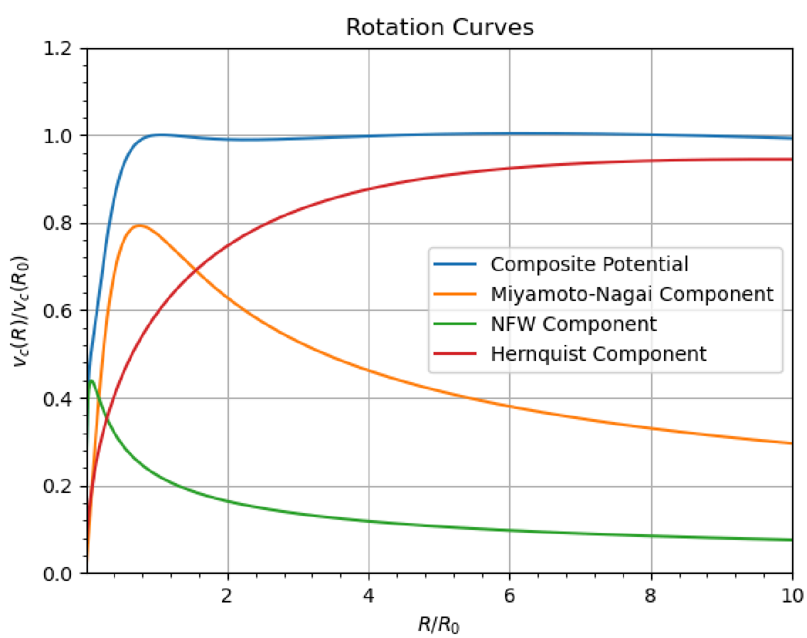
[11, 12] Below is an example of how to use galpy to plot rotation curves. galpy makes these figures using matplotlib. Defining potentials is rather straightforward, and figure axes are labeled automatically. However, galpy's plotting methods do not accept any inputs to adjust the figure visuals (titles, colors, labels, etc.). This can be done manually, nonetheless, by using `matplotlib.pyplot` to override the default conditions.

```
from galpy.potential import NFWPotential, HernquistPotential
mp = MiyamotoNagaiPotential(a=0.5, b=0.0375, normalize=.6)
np = NFWPotential(a=4.5, normalize=.35)
hp = HernquistPotential(a=0.6/8, normalize=0.05)
# normalize values sum to 1 --> circular velocity =1 at R=1

from galpy.potential import plotRotcurve
plotRotcurve(hp+mp+np, Rrange=[0.01,10.], grid=1001, yrange=[0.,1.2])
# can replace hp+mp+np with [hp,mp,np] or with MWPotential2014 if you import
   it from galpy.potential

mp.plotRotcurve(Rrange=[0.01,10.], grid=1001, overplot=True)
# overplot=True ensures all curves are on one figure
hp.plotRotcurve(Rrange=[0.01,10.], grid=1001, overplot=True)
np.plotRotcurve(Rrange=[0.01,10.], grid=1001, overplot=True)

import matplotlib.pyplot as plt
plt.title("Rotation Curves")
plt.grid()
plt.legend(['Composite Potential', 'Miyamoto-Nagai Component', 'NFW
   Component', 'Hernquist Component'])
```



[13] galpy primarily utalizes Python code, but galpy also C code in order to make certain calculations more efficient.

[14, 15] galpy takes parameters, rather than datasets, as input and outputs numbers or figures. From these figures, one could use various attributes or methods to capture specific data points.

[18] The documentation explains that, in order to install galpy, one must have the packaging, numpy, scipy, and matplotlib packages installed. The documentation also mentions the following optional dependencies: astropy for units and universal constants, astroquery for `Orbit.from_name` initialization, tqdm to display progress bars, numexpr to plot arbitrary expressions of Orbit quantities, numba to speed up the evaluation of certain functions, JAX for constant-anisotropy distribution functions in `galpy.df.constantbetadf`, and pynbody for `SnapshotRZPotential` and `InterpSnapshotRZPotential`. I found that astropy is a very helpful tool to keep track of units.

[19, 23] The material taught in PHYS265, in conjunction with galpy's documentation page (the Introduction page in particular), sufficiently allowed me to navigate galpy. I did not need to learn any new Python methods or strategies to complete this project.

[20] One who uses galpy should cite the galpy paper, Bovy's 2014 paper describing how and why he developed galpy. The AAS style citation for the galpy paper is as follows:

`galpy`: A Python Library for Galactic Dynamics, Jo Bovy (2015), *Astrophys. J. Supp.*, **216**, 29

[21] References

- The galpy paper - https://iopscience.iop.org/article/10.1088/0067-0049/216/2/29
- Gala documentation - https://gala.adrian.pw/en/latest/
- Google Gemini (via Google Search and gemini.google.com/app)
    - Helped me find sources and essentially skim through galpy's documentation in order to answer questions
- ChatGPT - https://chatgpt.com/
    - Taught me how to title galpy figures using matplotlib.pyplot
    - Provided the citation style for the galpy paper citation above
- matplotlib documentation - https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.legend.html

[22] galpy has been used in over 950 astrophysical publications, many of which are listed on the bottom of galpy's documentation page. In 2023, a team led by Yaqian Wu published a paper using galpy on the formation of thin discs:

Timing the formation of the galactic thin disc with asteroseismic stellar ages,
Yaqian Wu, Maosheng Xiang, Gang Zhao, Yuqin Chen, Shaolan Bi, et al. (2023), Mon. Not. Roy. Astron. Soc. 520, 1913

In 2020, a team led by Francois Hammer published a paper using galpy arguing against the presence of dark matter in Milky Way dwarf spheroidals:

Orbital evidences for dark-matter-free Milky Way dwarf spheroidal galaxies,
Francois Hammer, Yanbin Yang, Frederic Arenou, et al. (2020), Astrophys. J. 892, 3

[24] This marks my first time using galpy. The above code report was completed on my own.