Galpy: Dynamic Galactic Dynamics
Erik Siegwarth
5/9/2025

[1]
Galpy is a software package designed for galactic dynamics calculations. It is an associate package of astropy and is primarily written in python, with some parts in C. galpy's main goal is to accelerate certain calculations needed for galactic dynamics, such as orbit integrations, orbital frequencies, and calculating gravitational potentials and forces.
[2]
I selected this package because I thought it covered an interesting subject matter, galactic dynamics, and I also originally had wanted to do my final project on GALA but it proved too difficult to install. And Professor Teuben suggested galpy as an alternative. For my project I am using galpy, version 1.10.2
[3]
The package was first created in july 2010, and I could not identify any related code that came before or after, though it does rely on numpy, scipy, and matplotlib to function. As mentioned earlier GALA can also do many of the same functions as galpy. Galpy has plotting functions imbedded, such as plotrotcurve, but it relies on matplotlib axes.
[4]
It is still maintained by the original authors. They document how to contribute to the project on the github page, where any details can be found.
[5/6]
It was simple to install only taking a few minutes, I used the recommended windows installation method of:

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

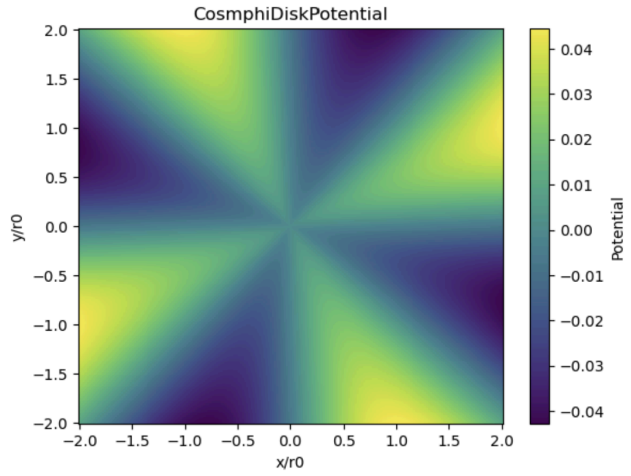
On the galpy github page they detail that for mac and linux you can use the simple pip install:

```
python -m pip install --only-binary galpy galpy
```

Which is recommended if you are using a non conda coding environment.

[7]On galpy's documentation page there is a link to the github webpage where all the source code can be found.
[8]
Astropy, an affiliated package, uses galpy related functions for orbit integrations.
[9]
In my examples i used a jupyter notebook.
[10/12]In galpy's documentation they provided an example of modeling the potential using the Miyamoto Nagai Potential. I chose to modify this example to model a 2d non axisymmetric potential, the Cos(m phi) Disk Potential. I first imported galpy, numpy, and matplotlib, and then the CosmphiDiskPotential. And I then defined my function for the potential with an amplitude of 2, an orientation angle of .43 radians, and other parameters taken from galpy's website. I applied my function over a grid of radius, and phi values. And plot it using a colormesh to see the potential at differing x and y points.
I also provide an example in my code of the graph of the rotation curves of two separate potentials, the Double Exponential Disk Potential and the Ring Potential at different radius'.

[11]
Galpy requires simple matplotlib plotting functions.



[12]
As described in the example above this figure represents the potential of a 2d non axisymmetric potential, given by a model and parameters taken from galpy's website. It is graphed for differing x and y values. Both of which have been normalized to be a distance x or y over a Ro of 8 KPC.
[13]Galpy, which is mainly written in python with some C, needs accompanying packages like numpy, matplotlib, and scipy; which are written in a mix of python, C, and Fortran.
[14]
Both parameters and datasets can be inputted into the package; as seen above parameters can be used for potential function, which would generate a theoretical data set. And you can also fit the potential model to real data that would need to be imported. For example as described in the second example above, you would create a theoretical model of an objects rotation curve in a given gravitational potential and then compare that with the observed data.
[15]
Both datasets and parameters can be outputted.
[16]
Yes, on galpy's github page there are a number of unit tests listed that can be used. I could not find any documentation of regression tests or benchmarking.
[17]
Besides using the unit tests, one could compare the results generated from galpy with results from other software packages like gala.
[18]
On galpy's documentation page it says it relies on numpy, scipy, and astropy. Some of galpys main functions rely on these packages, orbit integration uses odeint and rotation curves rely on astropy for proper units.
[19]
There was extensive documentation on the galpy website and on the github page.

[20]
Preferred citation: *galpy: A Python Library for Galactic Dynamics*, Jo Bovy (2015), *Astrophys. J. Supp.*, **216**, 29 (arXiv/1412.3451)
[21]
Astropy: https://www.astropy.org/acknowledging.html
Galpy: https://ascl.net/1411.008
[22]
Chemodynamics of the Milky Way. I. The first year of APOGEE data
Dynamical Modeling of Tidal Streams
[23]
I did not need to learn any new python methods. For my colormesh above I used all methods we had learned in class, and for my second example
[24]
I had no prior experience and did not work in a group.