Ruben Dasgupta
05/18/2025

PySpecKit: Toolkit for the Stars

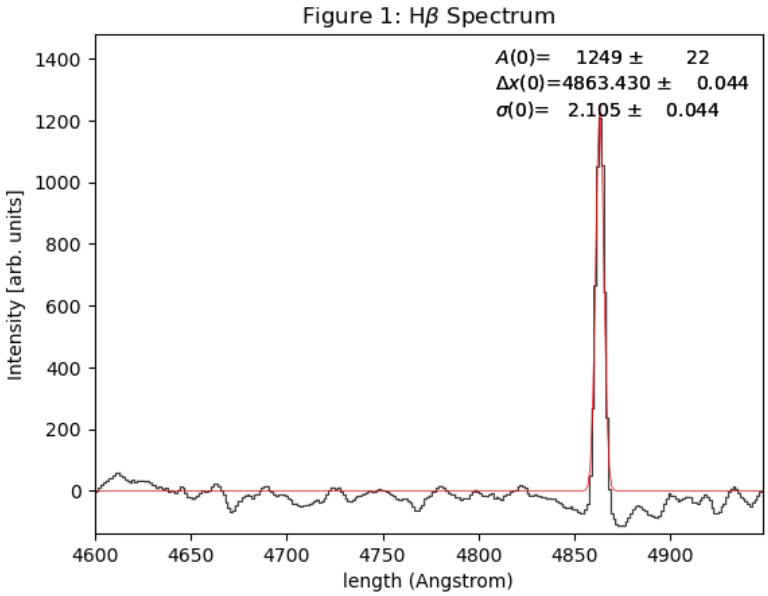
1) PySpecKit is a python package that aims to provide an all-in-one spectroscopic toolkit for astronomy and other applications. The package visualizes and fits models to different empirical spectra.
2) I selected this package because I was interested in gaining experience with a spectrum fitting package that's applicable to optical, infrared, and radio spectra. I thought it would be useful to learn such a package for my physics career.
3) PySpecKit's origin dates back to 2009. However, the official publication that released PySpecKit was in The Astronomical Journal in May of 2022. I have version 1.0.4 installed on my device.
4) PySpecKit is still maintained by its original author Adam Ginsburg. On the pyspeckit.readthedocs.io website there a detailed instructions on how to contribute to pyspeckit.
5) To install pyspeckit I followed the instructions on github. The command was: "!pip install https://github.com/pyspeckit/pyspeckit/archive/master.zip" but "!pip install pyspeckit" also works and is shown in my notebook. These were very straightforward instructions and installation with good documentation.
6) This does install by the standard pip methods as shown above.
7) The source code can be found on the github at pyspeckit/pyspeckit: Python Spectroscopic Toolkit. The inspect package can also give some more details about pyspeckit on jupyterlab, but it still directs the user to the github.
8) The code is used in a few different packages. Two of the ones I found are: GAUSSPY+, and MUSEpack.
9) The code is normally used as a python script or jupyter notebook. I use a jupyter notebook.
10) Here is a short example of my use of PySpecKit. See the jupyter notebook for further information.

```
xaxis = angstrom
baseline = np.poly1d([650,720])(xaxis)

# Can add noise if interested. The data provided by prof Teuben has no noise so that inclusion is not done here.

data = intensity
# There is a blank header warning that pops up, we can disregard per the pyspeckit authors, it is normal.
sp = pyspeckit.Spectrum(data=data, xarr=xaxis,
                        xarrkwargs={'unit':'Angstrom'},
                        unit='Intensity [arb. units]')

sp.plotter()
```

11) The package produces its own figures but they are matplotlib based, and very similar to the old IRAF plotting. The command sp.plotter() is the necessary plotting command.
12) Below is the figure:

Ruben Dasgupta
05/18/2025

Figure 1, to the right, is the Hβ spectral line in NGC253. The gaussian fit routine of PySpecKit generates the red line and the black line is the data. PySpecKit also returns labels of the fit details in TeX format on the top right. The amplitude, center, and width/sigma are reported. The y-axis is set to zero, the original data had an offset ~600 in arbitrary units of Intensity or Flux.



13) The package is pure python.
14) The input to package is normally datasets rather than parameters. This is so that the user can easily enter data from a file and use a built-in routine to visualize and fit. However, PySpecKit allows the user to create new fitting models/parameters within their package.
15) The output of the package is parameters of the fitting process which can be tuned, but it also gives publication quality graphs which is the screen ouput portion.
16) The code does have tests and benchmarks shown in the github. These can be found under the test folder at: pyspeckit/pyspeckit-tests at c11ed9b3747a7b5bc18a48340ca3c2f207cd8c85.
17) I feel quite confident about the code's ability to produce a reliable result. There are a few components to this. One includes the multitude of tests and test-cases demonstrated on the github. A second less rigorous verification is that I have plotted the NGC253 HB data and fit it using the package to achieve reasonable results.
18) The main python packages PySpecKit depends upon is Numpy and Astropy. PySpecKit was integrated in the Astropy affiliation in 2015 and most of the units and conventions are native to Astropy. In fact, I believe Astropy is a requirement for installation and use. I found this information by reading the documentation and looking at examples which extensively used Astropy, and numpy to a lesser extent for smaller tasks.
19) The documentation provided by the package was sufficient because of the plethora of example cases. On readthedocs.io there was a general overview and description of classes, functions, and function parameters but the examples were the most helpful and intuitive.
20) They do give a preferred citation method for use of this code in a publication, which is the 2022 release. The URL is: Pyspeckit: A Spectroscopic Analysis and Plotting Package - Astrophysics Data System.
21) Reference links used for implementation
    a) Basic Plotting Guide — pyspeckit v1.0.2.dev232

Ruben Dasgupta
05/18/2025

b) ASCL.net - PySpecKit: Python Spectroscopic Toolkit
c) pyspeckit/pyspeckit: Python Spectroscopic Toolkit
d) Galactic H₂CO Densitometry. I. Pilot Survey of Ultracompact H II Regions and Methodology - Astrophysics Data System

22) Citation style references

[1] J. Negus, J. M. Comerford, and F. Müller Sánchez, A Catalog of Broad Hα and Hβ Active Galactic Nuclei in MaNGA, ApJ **971**, 92 (2024).

[2] A. Ginsburg, A. T. Barnes, C. D. Battersby, A. Bulatek, S. Gramze, J. D. Henshaw, D. Jeff, X. Lu, E. A. C. Mills, and D. L. Walker, JWST Reveals Widespread CO Ice and Gas Absorption in the Galactic Center Cloud G0.253+0.016, The Astrophysical Journal **959**, 36 (2023).

23) I did not need to learn new python methods to use this package. The class was enough to get through this project

24) I have no prior experience with this package. The work was done solely by me; I did not work in a group.