[1] Evaluating the Efficiency of PySpecKit

Maria Nolan - May 18, 2025

[2] The PySpecKit package is a Python spectroscopic analysis and reduction toolkit designed for general use in optical, infrared, and radio spectra. PySpecKit allows for reading various file types, including FITS-standard and non-standard files like CLASS spectra. It also includes features for fitting models to spectra, baseline and continuum fitting, and equivalent width measurements. Furthermore, PySpecKit produces high-quality plots with TeX labels and annotations. PySpecKit is extensible, allowing users to add their own readers, writers, and fitting routines.

[3] I selected this package to review because I enjoy visualizing and analyzing spectral data. Furthermore, as a Physics & Astronomy double-major, I believe that knowing how to efficiently utilize this package in Python will be helpful later in my education and my career.

[4] PySpecKit's development began in 2009 with a script called "showspec" within the agpy package used in Google Code. This script was created by a graduate student—Adam Ginsburg—at the University of Colorado for plotting and fitting profiles to spectra in Python. Since it has been developed, Astropy was created and included similar features to PySpecKit. In 2017 PySpecKit became an affiliated package with Astropy. After doing further research, I discovered a package called "GalaPy" that is also a spectral modeling tool for galaxies.

[5] PySpecKit is still maintained (and actively under development) by the original authors—Adam Ginsburg and many others. If one wishes to contribute to this project, they can navigate to the GitHub repository for PySpecKit and go to the "Issues" tab to report any bugs or ideas. Furthermore, one can send an email to pyspeckit@gmail.com with any questions or ideas. I downloaded version 1.0.4.dev of PySpecKit.

[6, 7] PySpecKit was quite easy to install since installation is the same as any Python package, so long as one has pip. The install took less than one minute.

```
pip install
https://github.com/pyspeckit/pyspeckit/archive/master.zip
```

If one wishes to install PySpecKit but does not have pip, there is another option as well.

```
wget --no-check-certificate
https://github.com/pyspeckit/pyspeckit/archive/master.zip
unzip master.zip
```

```
cd pyspeckit
python setup.py install
```

[8] The source code for PySpecKit is available on its GitHub repository under the “pyspeckit” directory.
[9] PySpecKit is not used by other packages as a dependency. However, it is used alongside many other packages, such as SpectralCube and Astropy, in astronomical data analysis workflow.

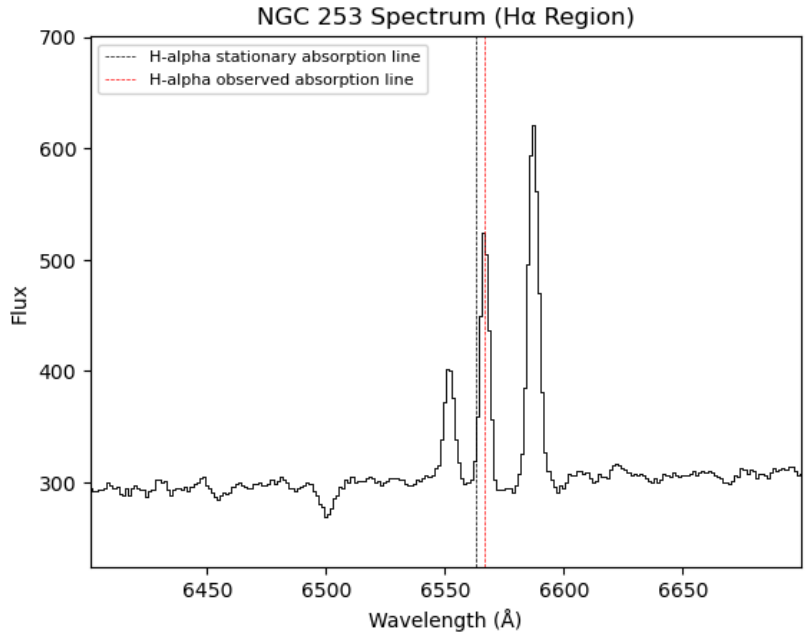
[10] PySpecKit is primarily used as a Python library, and is commonly used in Python scripts, Jupyter notebooks, and interactive GUI plotters via Python.

[11] The accompanying notebook shows how I have used PySpecKit. It is a Python notebook.

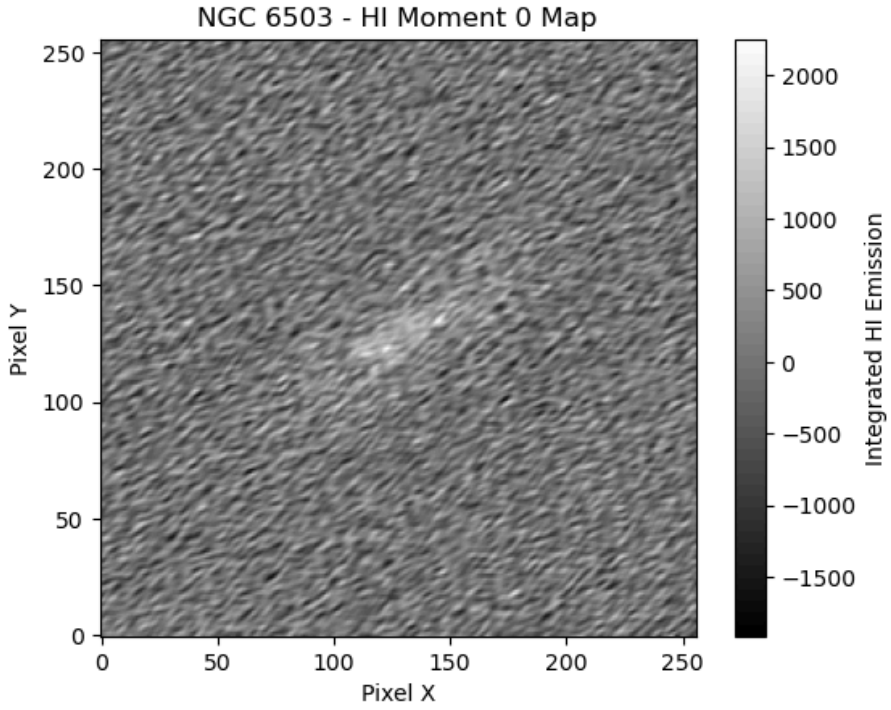
[12] PySpecKit is able to produce figures using Matplotlib. PySpecKit extends Matplotlib by wrapping spectral plotting and labelling tools for astronomy. An example of some code I used to plot data was:

```
spec = pyspeckit.Spectrum(xarr=wavelength_ha, data=flux_ha,
xarrkwargs={'unit':'angstrom'})
spec.plotter()
spec.plotter.plot()
```

[13] Here is a figure of a NGC 253’s spectrum in the Hα region using .tab data and the example code from the previous question.



In addition, I utilized the spectral-cube import to plot the FITS file containing information about NGC 6503. I plotted the spectral cube of the NGC's HI Moment 0 Map.



[14] PySpecKit is written entirely in Python. It does not need any accompanying C/C++/Fortran Code.

[15] PySpecKit accepts both datasets (e.g. spectra from FITS or .tab data) and user-defined parameters. In addition, NumPy arrays can also be utilized to generate inputs from scratch

[16] The main outputs of the PySpecKit package includes visual plots of spectrum and fit, best-fit parameters, residuals, and full access to model data arrays. Outputs can be printed, saved, or exported for further analysis.

[17, 18] PySpecKit provides unit tests in the "tests/" directory primarily using "pytest". However, it does not currently include formal regression or benchmarking tools, although many unit tests effectively catch regressions by checking expected numerical values and behaviors. We can trust the results this package produces because it includes automated tests (e.g., spectral fitting, data loading, plotting), is built on trusted libraries, and has been used in peer-reviewed scientific research.

[19] PySpecKit mainly depends on standard scientific libraries such as NumPy, numpydoc, Matplotlib, OrderedDict, Astropy, and spectral-cube. I was able to find the requirements by going to the PySpecKit GitHub repository, then clicking on the "REQUIREMENTS" page.

[20] The package documentation can be found in the PySpecKit GitHub repository. There is a detailed README file and a link under documentation that provides a thorough list of supported file types, guides, classes, and features. https://pyspeckit.readthedocs.io/en/latest/
This documentation sufficiently gave me an understanding of how to use this package and apply it to the project.

[21, 22]

- PySpecKit Preferred Citation: https://ui.adsabs.harvard.edu/abs/2022AJ....163..291G
- ASCL: ASCL.net
- PySpecKit GitHub Repository: https://github.com/pyspeckit/pyspeckit

[23] Using ADS, I was able to find only one source identified in the ASCL reference: https://ui.adsabs.harvard.edu/abs/2011ApJ...736..149G/abstract. This is the only source that references PySpecKit in ADS.

[24] This is my first time using PySpecKit, so I had to learn how to use some new techniques, like spectral-cube. I have worked with Astropy before and spectral data, as well as NumPy and Matplotlib, so I didn't have to learn much else with that to complete this project. Using the package documentation, as well as prior knowledge from this class, I was able to complete this project.

[25] I did not have prior experience with this package or data. In addition, I worked with Riley Lutz on the project.