[1] PySpecKit: A Python Toolkit for Spectral Analysis

Riley Lutz

May 18, 2025

[2] PySpecKit is a toolkit written in Python that is designed to aid physicists and astronomers in analyzing spectral data. It is primarily meant to provide tools for spectral line fitting, including gaussian and voigt profiles, and baseline-subtraction routines. PySpecKit is able to simplify daunting spectral models by offering known and easily usable fitting functions.

[3] I selected this package because I am interested in pursuing a career in astronomy and this package gives me the opportunity to practice methods that I may use frequently in the future. I expect that I will frequently analyze data sets, so to combine these two interests seems very interesting. I also selected this package because I knew I would have access to the necessary data to complete it.

[4] PySpecKit was originally developed by a graduate student in 2009 with a script called "showspec" in the agpy package. This toolkit became more developed over time, and by 2011 it was finally moved into its own repository, and then into GitHub by 2012. Since it has been developed, astropy was created and has included features that made some of PySpecKit redundant. In 2017, PySpecKit became an affiliated package with astropy. I am using version 1.0.4.

[5] The primary author listed on GitHub is Adam Ginsburg with assistance from 25 other contributors. Adam Ginsburg is still the primary contributor to this package, although his commits over time have drastically decreased since 2020. This package encourages edits and additions to the code. Contributors are instructed to navigate to the "issues" tab in the repository and post their cases.

[6, 7, 8] The installation process for this package is very simple, it only requires "!pip install pyspeckit". There are no complex dependencies for this package, only standard Python scientific libraries such as matplotlib, numpy, and astropy (new to me but easy to understand). This source code is openly available on GitHub and allows users to modify, inspect, and extend the toolkit.

[9] Other packages do not depend on PySpecKit, but it can naturally be used concurrently with Astropy and SpectralCube. All these packages are useful for spectral analysis.

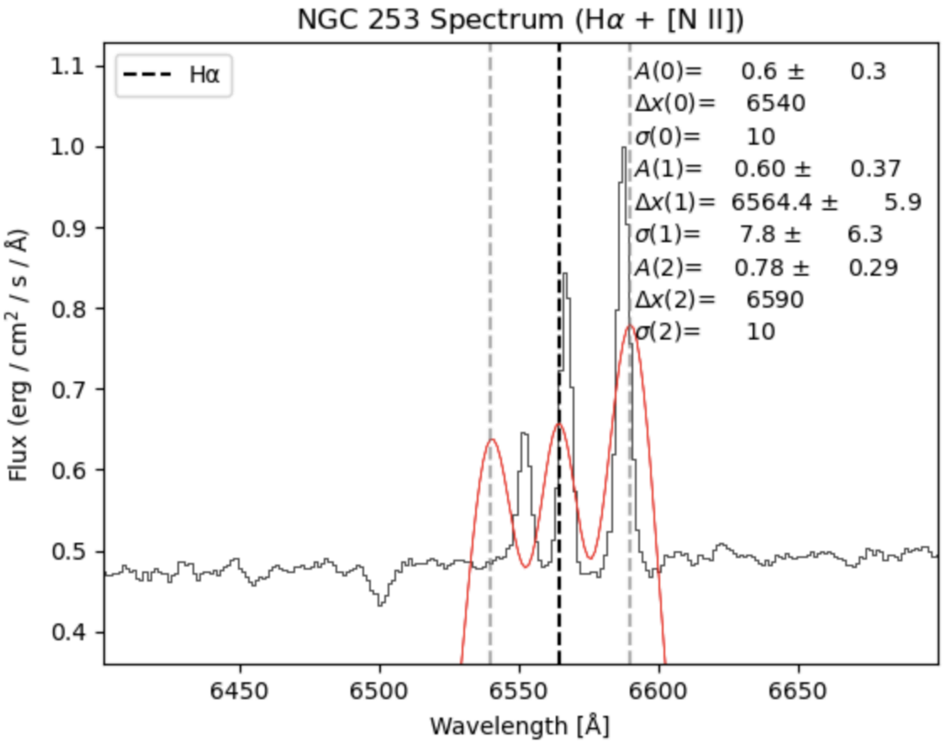
[10] PySpecKit is primarily used as a Python library within Jupyter notebooks or Python scripts. By itself, PySpecKit does not provide a web UI or command-line interface.

[11] In the notebook below, the normalized spectrum of NGC 253's Hα region with a multi-Gaussian fit is shown by using PySpecKit's "specfit" module.

[12] PySpecKit utilizes matplotlib to create figures. PySpecKit adds features to these plots that are otherwise more difficult to produce. Below is a small snippet of code to demonstrate how I plotted and fitted the data:

```
spectrum_ha = psk.Spectrum(data=flux_ha, xarr=xarr_ha, xunit='angstroms',
unit='erg / (cm^2 s angstrom)', header=header_ha)
spectrum_ha.data /= np.max(spectrum_ha.data)
guesses = [
    0.2, 6548.0, 2.5,   # [N II] 6548
    1.0, 6563.0, 3.0,   # Hα
    0.4, 6583.0, 2.5    # [N II] 6583
]
spectrum_ha.specfit(fittype='gaussian', guesses=guesses,
limited=[(True,True)]*9,
                    limits=[(0,2), (6540,6555), (0,10),
                            (0,2), (6555,6570), (0,10),
                            (0,2), (6570,6590), (0,10)])
spectrum_ha.specfit.plot_fit(color='red')
```

[13] Here is a figure that shows the NGC 253 Spectrum in the Hα region with observed flux versus wavelength. The black line is the normalized spectral data, and the red curve shows the Gaussian components fitted to the emission lines.



[14] PySpecKit is written solely in Python and does not require any additional coding language.

[15, 16] PySpecKit is meant to be compatible with several forms of spectroscopic data. This could include inputs of ASCII text files, FITS files, and programmatically generated spectral data. The outputs of the package come in many forms such as fitted model parameters, fit statistics/uncertainties, plots that overlay spectra, and residuals.

[17, 18] PySpecKit includes unit tests that ensure accuracy in its core functions. Because it is reliant on numerical libraries like NumPy and SciPy, PySpecKit is very reliable. This package also uses characteristics of Astropy for units and coordinate systems which limits common errors. It also forces you to think critically when making initial guesses and parameter constraints for fitting. With this being said, the package does not include formal regression or benchmarking tools.

[19] PySpecKit depends on standard scientific libraries such as NumPy, SciPy, Matplotlib, and Astropy. I discovered this from the requirements tab in the PySpecKit repository. NumPy is heavily used for numerical array handling, and matplotlib is necessary for producing plots which is vital to this project. Astropy deals in astronomy-specific utilities.

[20] There is a comprehensive list of documentation that is very user-friendly. The package documentation can be found in the GitHub repository. There is a detailed README 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 was sufficient in guiding me on this project and gave me a thorough enough understanding to apply my knowledge.

[21, 22] A preferred citation method is linked into the ASCL.net website. The preferred citation method is: **https://ui.adsabs.harvard.edu/abs/2022AJ....163..291G/abstract**. I used no other references.

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

[24] This was my first time using PySpecKit and Astropy and handling this type of spectral data. Because it heavily relies on NumPy and Matplotlib, I did not have to learn too much to complete this project. Using the package documentation, I was able to use my knowledge from the class and complete this project.

[25] I did not have prior experience in this package or this data, it is all new to me. I collaborated with Maria Nolan.