Zya Woodfork
PHYS265
18 May 2025

PySpecKit: Python Spectroscopic Toolkit

1. **Name of the package, describe what is the basic aim of what the package does or solve?**
   This package is called PySpecKit, and it is used to analyze spectroscopic data. It was made to act as a spectroscopic analysis tool that is user-friendly and applicable at multiple wavelengths.

2. **Why/how did you select this package?**
   I selected this package because I have a little bit of experience in analyzing star spectra and wanted to see how this could make a process like that easier.

3. **How old is the package? does it have a geneology, i.e. what related codes came before or after. are there other codes you can find that solve the same problem?**
   Development of PySpecKit started in 2009 with a script called “showspec”. Other similar tools include IRAF and SPLAT, but these aren’t as easy to use and only focus on one or a few wavelengths.

4. **Is it still maintained, and by the original author(s)? Are there instructions how to contribute to this project.**
   The documentation was last updated on September 16, 2023 by the original author, Adam Ginsburg. The github README says that if you have issues or suggestion, you can either email pyspeckit@gmail.com or post on the issues page.

5. **Evaluate how easy it was to install and use. What commands did you use to install?**
   To install PySpecKit, all I had to do was run the command *pip install pyspeckit*. Then I did an import statement, and the package was ready to use. The documentation was very useful in helping me figure out where to start.

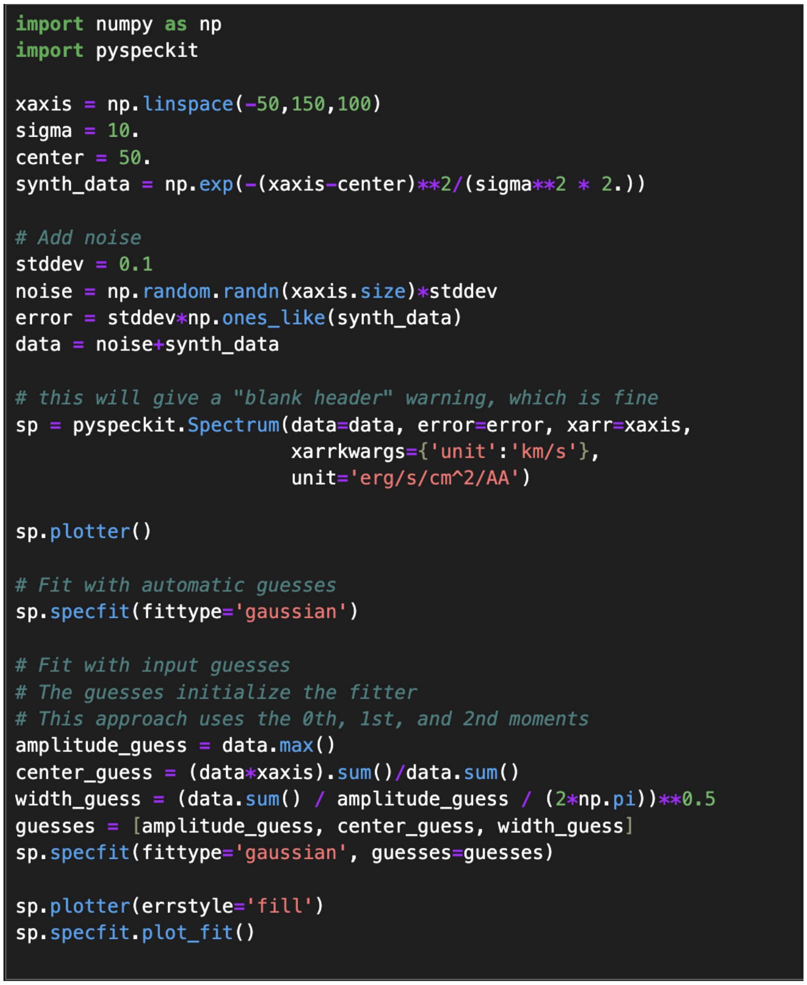
6. **Does it install via the "standard" pip/conda, or is it more complex?**
   This is installed with a standard pip install command.

7. **Is the source code available? For example, "pip install galpy" may get it to you, but where can you inspect the code?**
   Yes, source code is available at this link: https://github.com/pyspeckit/pyspeckit.

8. **Is the code used by other packages (if so, give one or two examples). ASCL codes have citations via their ADS link. See also 22.**
   The program Spectrum Iterative Fitter (SPIF) uses PySpecKit.

9. **How is the code used. Is it commandline, python script, or a jupyter notebook, or even a web interface?**
   PySpecKit's code is a Python script that you can edit and change depending on what you want to do with your data.

10. **Provide examples using the code. if you prefer to use a jupyter notebook instead of a python script, that's ok. See also 12.**
    This code provided in the documentation generates data and then plots and fits it to a gaussian fit.

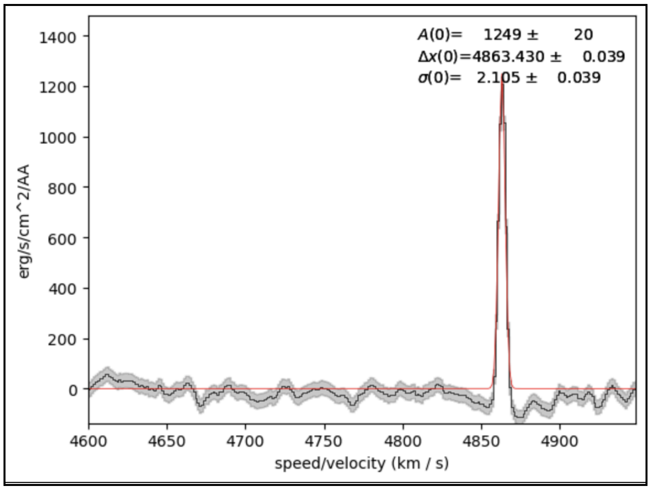


```
import numpy as np
import pyspeckit

xaxis = np.linspace(-50,150,100)
sigma = 10.
center = 50.
synth_data = np.exp(-(xaxis-center)**2/(sigma**2 * 2.))

# Add noise
stddev = 0.1
noise = np.random.randn(xaxis.size)*stddev
error = stddev*np.ones_like(synth_data)
data = noise+synth_data

# this will give a "blank header" warning, which is fine
sp = pyspeckit.Spectrum(data=data, error=error, xarr=xaxis,
                        xarrkwargs={'unit':'km/s'},
                        unit='erg/s/cm^2/AA')

sp.plotter()

# Fit with automatic guesses
sp.specfit(fittype='gaussian')

# Fit with input guesses
# The guesses initialize the fitter
# This approach uses the 0th, 1st, and 2nd moments
amplitude_guess = data.max()
center_guess = (data*xaxis).sum()/data.sum()
width_guess = (data.sum() / amplitude_guess / (2*np.pi))**0.5
guesses = [amplitude_guess, center_guess, width_guess]
sp.specfit(fittype='gaussian', guesses=guesses)

sp.plotter(errstyle='fill')
sp.specfit.plot_fit()
```

11. **Does the package produce figures, or are you on your own? Is matplotlib used?**

This package uses matplotlib to produce its figures.

12. **Your code and report should show at least one figure, and create a nice figure caption explaining what it shows. You notebook should show how the figure was made (i.e. be reproducable). Second figure is optional, but only use it when you need to illustrate something extra.**



Spectrum of NGC253 in H-beta with a gaussian fit.

13. **Is the package pure python? or does it need accompanying C/C++/Fortran code?**
    PySpecKit is pure Python.

14. **What is the input to the package? Just parameters, or dataset(s), or can they be generated from scratch?**
    You can either generate data or use data files.

15. **What is the output of the package? Just parameters, or dataset(s)?, or just a screen output you would need to capture**
    The output is a graph of the generated/imported data with a fit over it.

16. **Does the code provide any unit tests, regression or benchmarking?**
    In the actions section of the github, it shows tests that have been run. I believe these are to make sure all the code still works.

17. **How can you feel confident the code produce a reliable result? (see also previous question)**
    I can tell if the code works correctly by examining how closely the gaussian fit seems to fit the data.

18. **What (main) python package(s) does it use or depend on (e.g. numpy, curve_fit, solve_ivp) - how did you find this out?**
    When I go to the requirements page of the github, it gives a list of required packages, which includes numpy, matplotlib, and any version of astropy greater than v1.

19. **What kind of documentation does the package provide? Was it sufficient for you?**
    The documentation provides many pre-written Python scripts to demonstrate the package's abilities. These were very helpful in showing me what PySpecKit can do and how to get it to do those things.

20. **If you use this code in a paper, do they give a preferred citation method?**
    The documentation says to cite PySpecKit, you should use these two links: https://ui.adsabs.harvard.edu/abs/2022AJ….163..291G/abstract and http://adsabs.harvard.edu/abs/2011ascl.soft09001G, which are two papers that detail the history and capabilities of PySpecKit.

21. **Provide any other references you used in your report.**
    I used this paper to get some background information about PySpecKit's development: https://iopscience.iop.org/article/10.3847/1538-3881/ac695a/pdf.
    Other than that, all I used was PySpecKit's documentation: https://pyspeckit.readthedocs.io/en/latest/

22. **Can you find two other papers that used this package. E.g. use ADS citations for ASCL based code. See also 8.**
    - A Large Systematic Search for Close Supermassive Binary and Rapidly Recoiling Black Holes. IV. Ultraviolet Spectroscopy
    - Physical Conditions of the Ionized Superwind in NGC 253 with VLT/MUSE

23. **Did you have to learn new python methods to use this package? Or was the class good enough to get you through this project.**
    I did not have to do anything new to use this package. PHYS265 was good enough to get me through the project.

24. **Final Disclaimer: You need to state if you have prior experience in using the package or the data, or this is all new to you. In addition, if you collaborated in a group, as long as this is your work.**

I had no prior experience in using PySpecKit. I have done some work with spectral data in the past. I worked in a group with Debika Biswas, Jasmin Mohammadi, and Margaret Haswell.