PHYS 265 Final Lab Report

May 18, 2025 ■ Erika Falco

APLpy: Plotting FITS Files is Easy as Py

[1] This report explores the capabilities of APLpy version 2.0.0. APLpy, or the Astronomical Plotting Library in Python, is a python package whose primary function is to turn Flexible Image Transport System (FITS) files into images with labeled axes. This package can be used as a tool to plot astronomical data, allowing users to plot grayscale, colorscale, or RGB images of FITS files, slice multidimensional data cubes, and overlay elements like coordinate grids and contours onto plots. This tool allows astronomers to quickly and efficiently plot images.

[2] I selected this package because I am interested in how astronomers use python to plot their data. I find the idea of FITS files really interesting with respect to how they are created, with images being constructed from electron deposits per pixel. I also enjoy getting a visual output from my code, so choosing a plotting package seemed like a good fit.

[3] APLpy is 16 years old, with the first version having been published in 2009. The package is not derived from any specific previous code, though it does rely on a number of other packages to function. APLpy has always been reliant on Matplotlib to render figures and was developed using PyFITS, which was the original FITS file handler in Python. APLpy came before Astropy, but later versions of APLpy incorporate Astropy's WCSAxes package, among other Astropy packages like fits. APLpy has now been updated to meet the requirements to be considered an Astropy-affiliated package. Though Astropy's WCSAxes package is incorporated into APLpy, it can also be used directly to serve roughly the same purpose as APLpy. WCSAxes is ideal for plotting astronomical images when more complex or detailed plots are required. APLpy's main appeal is its efficiency and simplicity – if you are looking to plot a FITS file quickly, APLpy is the way to go.

[4] APLpy was created by Thomas Robitaille and Eli Bressert. It is still being maintained by Robitaille, as well as a number of new contributors, and the latest version (2.2.0) was released on November 14, 2024. Though the documentation does not provide clear instructions on how to contribute to this package, you can file an issue or submit a pull request through APLpy's github. The large number of contributors listed on github suggests that many people have contributed through this route.

[5] APLpy was very easy to install with pip install, using the command "pip install aplpy."

[6] This package is installed via the standard pip.

[7] The source code for APLpy is available on their github, at https://github.com/aplpy/aplpy.

[8] APLpy code is not used by other python packages.

[9] I used APLpy in a jupyter notebook, but it can also be used in the terminal through non-interactive mode.

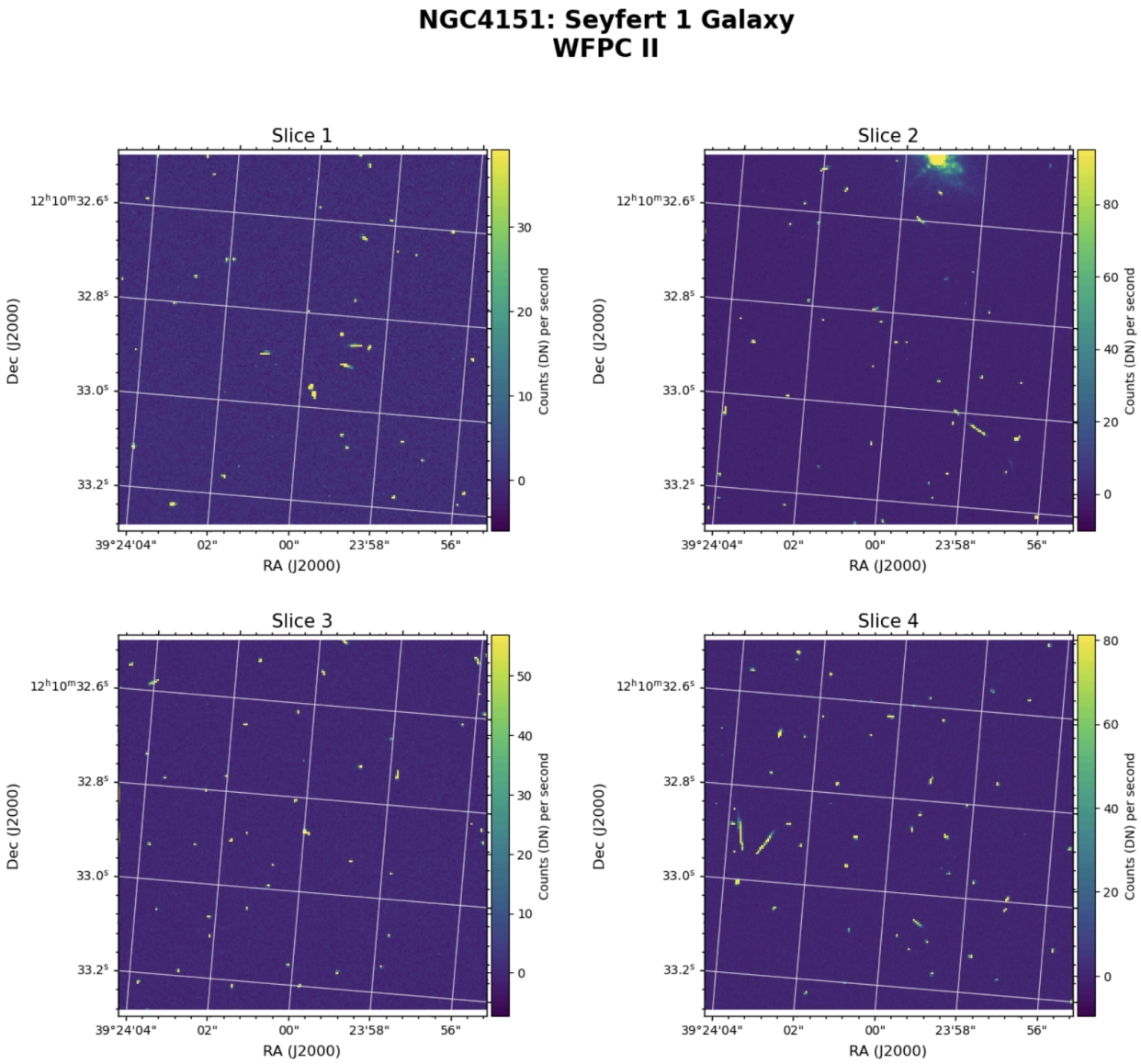
[10] I used APLpy to slice and display a FITS data cube, as illustrated in section 12**.** The FITS cube was sliced using APLpy's FITSfigure function with a dimension argument of [0, 1], which displays a slice with right ascension on the x axis and declination on the y axis. I displayed slices 1-4 by changing the value of the "slices" keyword argument within the FITSfigure function. APLpy's plotting capabilities were used in tandem with Matplotlib in order to format the figure, using Matplotlib's subplots function in order to plot several subplots within the figure.

Additionally, I explored APLpy's two dimensional FITS file plotting capabilities by using its show_rgb function to generate an RGB image of the galactic center from the Two Micron All-Sky Survey. I used APLpy's show_contour and show_markers capabilities to overlay a 24 micron MIPS contour onto the image and place markers at the locations of young stellar objects. I was able to easily save this plot as a png image via APLpy's save function. The files containing the FITS data for the image and the contours, as well as the young stellar object coordinates, were found through the beginner tutorial on APLpy's website, at https://aplpy.readthedocs.io/en/stable/fitsfigure/quickstart.html. These files are accessible in my github repository, and are available for download via http://aplpy.github.io/downloads/tutorial.tar.gz. Both examples of coding applications of APLpy highlight the ease and efficiency of plotting with this package, as each case took relatively few lines of code and ran pretty seamlessly.

[11] APLpy's main purpose is to produce figures. It uses Astropy's WCSAxes to draw coordinate axes and then renders the plots using Matplotlib. Previous versions of APLpy were able to directly plot coordinate axes, but the new version is reliant upon Astropy's WCSAxes functionality. It has always been reliant upon Matplotlib.

[12] Among APLpy's chief functionalities is the ability to display FITS cube slices, which allows astronomers to observe higher dimensional FITS file data in two dimensional "slices". The following figure displays four slices of a 800 x 800 x 4 primary array data cube (trimmed to 200 x 200 x 4 pixels) containing four CCD images taken by Hubble's Wide Field Planetary Camera II. This particular FITS data is imaging of NGC4151, the Seyfert 1 Galaxy

and was one of several sample FITS files from the MAST data archive. This file can be found at https://fits.gsfc.nasa.gov/samples/WFPC2u5780205r_c0fx.fits.



[13] APLpy is pure python and does not require any additional accompanying code.

[14] APLpy takes FITS files as an input.

[15] APLpy outputs plots in the form of EPS, PDF, PS, PNG, or SVG files.

[16] APLpy has unit/regression tests that can be run with "python setup.py test."

[17] Because the package has both unit and regression tests, we can be confident that it is checking that its internal functions are producing sound results and interacting with one another reliably. However, I did encounter a number of minor deprecation warning messages in my plotting process that indicate that some additional maintenance of the code is required.

[18] APLpy mainly relies on Matplotlib, numpy, and Astropy's fits and WCS.

[19] APLpy's documentation is very easily accessible. All of the documentation is available on the APLpy website, https://aplpy.github.io/. This was sufficient for the purposes of this project.

[20] There is a preferred citation method when using APLpy. Their website requests citation in the following format: "This research made use of APLpy, an open-source plotting package for Python (Robitaille and Bressert, 2012; Robitaille, 2019)." This citation refers to https://doi.org/10.5281/zenodo.2567476.

[21] I referenced the following websites and online publications in writing this report:

- https://aplpy.readthedocs.io/en/stable/
- https://aplpy.readthedocs.io/_/downloads/en/stable/pdf/
- https://github.com/aplpy/aplpy
- https://aplpy.github.io/
- https://aplpy.readthedocs.io/en/stable/fitsfigure/slicing.html#example
- https://github.com/aplpy/aplpy/blob/main/CITATION
- https://python4astronomers.github.io/plotting/aplpy.html#:~:text=APLpy%20(the%20Astronomical%20Plotting%20Library,plots%20interactively%20or%20using%20scripts
- https://astropy4cambridge.readthedocs.io/en/latest/_static/Affiliated%20Package%20-%20APLpy%20and%20WCSAxes.html
- https://github.com/aplpy/aplpy/blob/main/CHANGES.md

In plotting my figure in section 12, I used the following additional resources:

- https://fits.gsfc.nasa.gov/samples/
- https://www.stsci.edu/files/live/sites/www/files/home/hst/instrumentation/legacy/wfpc2/_documents/wfpc2_dhb.pdf

[22] APLpy is used for plotting in the publication, "Weak lensing analysis of A115, A2219 and A2261: Detection of galaxy groups and filaments around clusters" by Dutta et al. The package

is also used as a plotting tool in the publication "SIGNALS on the mixing of oxygen and nitrogen in the spiral galaxy NGC 6946" by Bresolin et al.

[23] I did not need to learn any new python methods to use this package.

[24] I did not have any previous experience with APLpy before this project. However, I collaborated with Isaac Sherwood and Sophiya Mehra in exploring the package's capabilities.