The SunPy Package

[0] SunPy
William Steltz - May 18, 2025

[1] SunPy is an open-source Python library designed to support solar physics research by providing tools to access, analyze, and visualize solar data. It simplifies working with data from space observatories, works with the broader scientific Python ecosystem (like NumPy and Astropy), and promotes reproducible research within the solar physics community.

[2] I picked this package because it was space related, but also something different. I haven't seen a type of code that can map the sun and return the X-ray flux. Overall, this package was something that I haven't seen before, and it captured my attention.

[3] I was using the version 6.1.1 of SunPy. SunPy came out in 2011, with work starting in 2010. It was made to replace older tools like SolarSoft, which is written in a different language called IDL. SolarSoft was the main toolkit for solar data for a long time, but SunPy offers a more modern, open, and easier-to-use option in Python.

SunPy is also closely connected to Astropy, another big astronomy Python project. It follows many of the same standards and works well with other Astropy tools, making it easier for scientists to use them together.

[4] Yes, SunPy is still actively maintained. It has a growing community of contributors and is managed by the SunPy Project, not just a single individual. While the original authors helped start the project, development is now handled by a broader team of solar physicists and developers from around the world. SunPy follows open-source best practices and welcomes contributions. It has clear, well-documented instructions on how to contribute, available on its official website.

[5] It was relatively easy to install, I just simply used the command "!pip install SunPy[all]."

[6] Yes, it installs through the standard pip protocol.

[7] Yes, SunPy's source code is open and free to look at. Even though "!pip install SunPy" installs the package for use, you can see and explore the actual code on its GitHub page: github.com/sunpy/sunpy. You can also download it using git clone if you want to study it, make changes, or help improve it.

[8] Yes, SunPy is used by other solar physics Python packages like AiaPy, which processes data from NASA's SDO mission, and sunraster, which handles solar spectroscopic data. These packages rely on SunPy's core tools for working with solar data and coordinates. SunPy is also registered in the ASCL under the ID ascl:1401.010, allowing it to be properly cited in scientific papers through the ADS system.

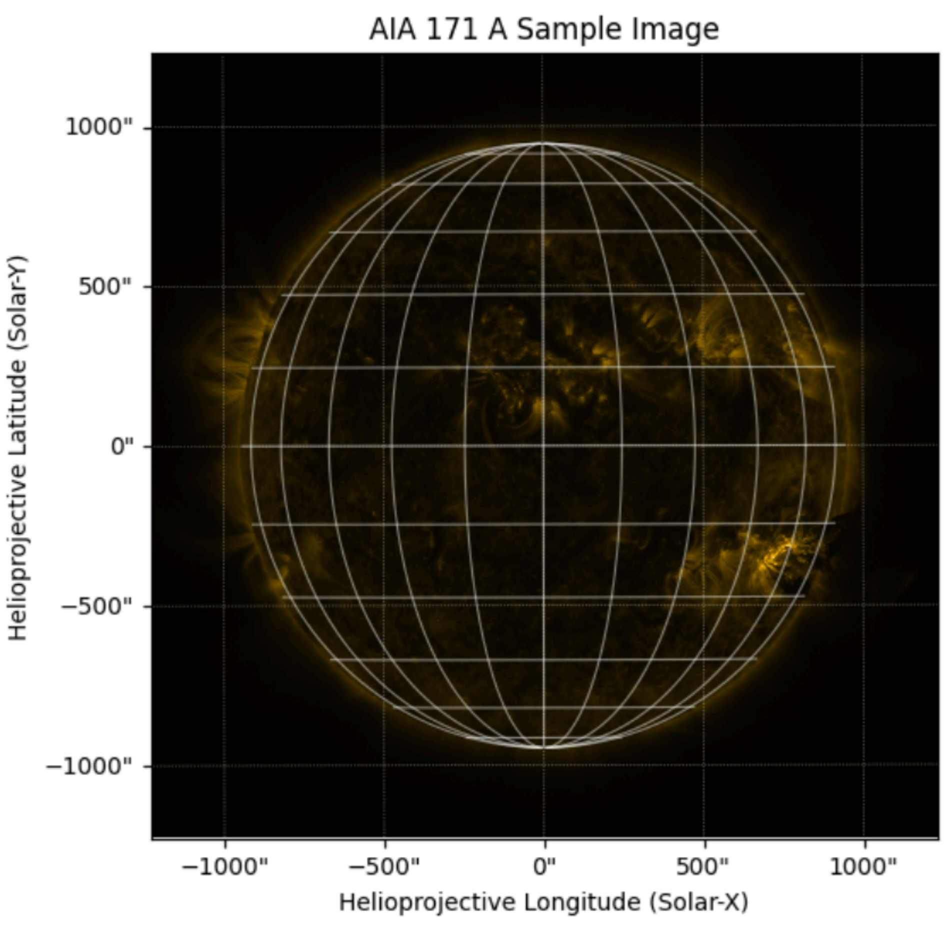
[9] SunPy is mainly used as a Python library that you can run in several ways: within Python scripts, interactive Python sessions, or in Jupyter notebooks. Researchers often use Jupyter notebooks because they allow combining code, data visualization, and notes all in one place, which is great for exploring and analyzing solar data.

While SunPy itself doesn't provide a dedicated command-line tool or web interface, it can be integrated into larger applications that do. But primarily, users write Python code to access SunPy's functions and classes directly.

[10] My accompanying JupyterLab notebook has my code. I used different tools like "sunpy.map.Map" to map the sun on a plot.

[11] I was able to create my figures using matplotlib.pyplot and object oriented plotting.

[12] Here is a figure showing a photo taken by the AIA instrument on NASA's Solar Dynamics Observatory (SDO) at the 171 Ångström wavelength, which shows the Sun's corona.



[13] SunPy is purely a Python package. It does not use C/C++ or Fortran.

[14] SunPy takes real solar physics data as input, often in the form of FITS files from missions like SDO or SOHO. These files can be loaded into a notebook if stored locally or downloaded using SunPy's Fido interface by specifying parameters such as time range, wavelength, or instrument. In addition to datasets, many SunPy functions accept parameters like time intervals, coordinates, or data types to search for, filter, or manipulate solar data.

[15] The output depends on how you use SunPy. It mainly returns data like maps, that can be plotted, or time intervals.

[16] Yes, SunPy includes a set of unit tests and regression tests to ensure the code's correctness and stability. The project follows modern software development practices, using continuous integration (CI) systems like GitHub Actions to automatically run tests on every code change.

[17] First, it has the aforementioned unit and regression testing.. Second, it follows strict coding standards and review processes. Third, it builds on trusted scientific libraries like Astropy, NumPy, and SciPy, which have their own well-tested codebases. Finally, being open-source, its methods are fully transparent, allowing anyone to inspect, test, and verify the code and its results.

[18] SunPy relies on several key Python packages to perform its functions. The most important dependencies include NumPy, Astropy, and matplotlib. I found this out by following the guide to SunPy that is available on the website.

[19] SunPy provides documentation such as user guides, API references, tutorials, and example notebooks. The documentation covers concepts like using Map and TimeSeries objects, downloading data with Fido, and plotting solar observations. The documentation is hosted at "docs.sunpy.org." Yes, it is generally sufficient and beginner-friendly, especially if you are already familiar with Python and scientific computing.

[20] Yes, the preferred citation is

The SunPy Community et al. (2015), *The SunPy Project: Open Source Development and Status of the Version 0.5 Release*, Computational Science & Discovery, 8(1), 014009. DOI: 10.1088/1749-4699/8/1/014009

[21]
SunPy Community. (2025). *SunPy Documentation* (Version 6.1.1). Retrieved May 17, 2025, from https://docs.sunpy.org/

[22] Mumford, S. J., et al. (2015).
SunPy: Python for Solar Physics.
Computational Science & Discovery, 8(1), 014009.
https://doi.org/10.1088/1749-4699/8/1/014009

The SunPy Community et al. (2023).
The SunPy Project: An interoperable ecosystem for solar data analysis.
Astronomy and Computing, 45, 101717.
https://doi.org/10.1016/j.ascom.2023.101717

[23] I learned everything I needed either from class, or from the published guides.

[24] This was my first time using the package. I worked alone.