sunpy 1.0 The community-developed, free and open-source solar data analysis environment for Python.

SH41C-3309

Poster Presenter: Steven D. Christe (NASA GSFC) steven.christe@nasa.gov
Primary Paper Contributors (alphabetical order): The SunPy Community, Will T. Barnes[1,2] , Monica G. Bobra[3,29] , Steven D. Christe[4,28] , Nabil Freij[28,30], Laura A. Hayes[4] , Jack Ireland[4,31,29] ,Stuart Mumford[5,6,29,32], David Perez-Suarez[7,33,29], Daniel F. Ryan[4,8] , Albert Y. Shih[4]
Sunpy Contributors (alphabetical order): Prateek Chanda[9], Kolja Glogowski[10,11], Russell Hewett[12,29] , V. Keith Hughitt[13] , Andrew Hill[14], Kaustubh Hiware[15], Andrew Inglis[4], Michael S. F. Kirk[4,8] , Sudarshan Konge[16], James Paul Mason[4] , Shane Anthony Maloney[17,18], Sophie A. Murray[17,18], Asish Panda[19], Jongyeob Park[20], Tiago M. D. Pereira[21,22] , Kevin Reardon[23,29], Sabrina Savage[24,29], Brigitta M. Sipőcz[25] , David Stansby[26] , Yash Jain[15], Garrison Taylor[27], Tannmay Yadav[15], and Rajul[15]

1 Lockheed Martin Solar and Astrophysics Laboratory, Palo Alto, CA 94304, USA; 2 Bay Area Environmental Research Institute, Moffett Field, CA 94952, USA; 3 W.W. Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA 94305, USA; 4 NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA; 5 SP2RC, School of Mathematics and Statistics, The University of Sheffield, UK; 6 Aperio Software, Leeds, LS6 3HN, UK; 7 University College London, Gower Street, London, UK, 8 The Catholic University of America, Washington, DC 20664, USA, 9 Indian Institute of Engineering Science and Technology Shibpur, India, 10 Kiepenheuer-Institut für Sonnenphysik, Freiburg, Germany, 11 eScience Department, Computing Center, University of Freiburg, Freiburg, Germany, 12 Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0123, USA, 13 Center for Cancer Research, National Cancer Institute, Bethesda, MD 20892-9760, USA, 14 Oklahoma Baptist University, Shawnee, OK 74804, USA, 15 Indian Institute of Technology Kharagpur, India, 16 Microsoft India Development Center, India, 17 Astrophysics & Space Physics, School of Physics, Trinity College Dublin, Dublin, Ireland, 18 Astronomy and Astrophysics, Cosmic Physics, Dublin Institute for Advanced Studies, Dublin, Ireland, 19 Google Summer of Code 2014; 20 Space Science Division, Korea Astronomy and Space Science Institute, Daejeon 34055, Republic of Korea, 21 Institute of Theoretical Astrophysics, University of Oslo, Oslo, Norway; 22 Rosseland Centre for Solar Physics, University of Oslo, Oslo, Norway; 23 National Solar Observatory, Boulder, CO 80303, USA, 24 NASA Marshall Space Flight Center, Huntsville, AL 35812, USA, 25 DIRAC Institute, Department of Astronomy, University of Washington, Seattle, WA 98195, USA, 26 Mullard Space Science Laboratory, University of England, Holmbury Hill Rd, Dorking RH5 6NT, UK, 27 Harvard-Smithsonian Center for Astrophysics, Cambridge, MA 02138, USA, 28 Institute for Environmental Analytics, University of Reading, Reading RG6 6BX, 29 SunPy Board Member, 30 SunPy Deputy Lead Developer, 31 SunPy Communications Officer, 32 SunPy Lead Developer, 33 SunPy Summer of Code Administrator

python™ NUMFOCUS OPEN CODE = BETTER SCIENCE

Abstract

The goal of the SunPy project is to facilitate and promote the use and development of community-led, free, and open source data analysis software for solar physics based on the scientific Python environment. The project achieves this goal by developing and maintaining the sunpy core package and supporting an ecosystem of affiliated packages. This poster describes the first official stable release (version 1.0) of the core package and concludes with a discussion of the future of the SunPy project.

Development

As of version 1.0, sunpy consists of 48,427 lines of code contributed by 123 unique contributors with over 11,659 commits. The total number of contributors is large for a package of this size in the heliophysics community (Ware et al. 2019). On average, a new contributor is added every month. In order to maintain a high-quality code base, every contribution is reviewed and must satisfy a set of strict and documented requirements. Converting new contributors into core developers is crucial to the long-term health of the community.

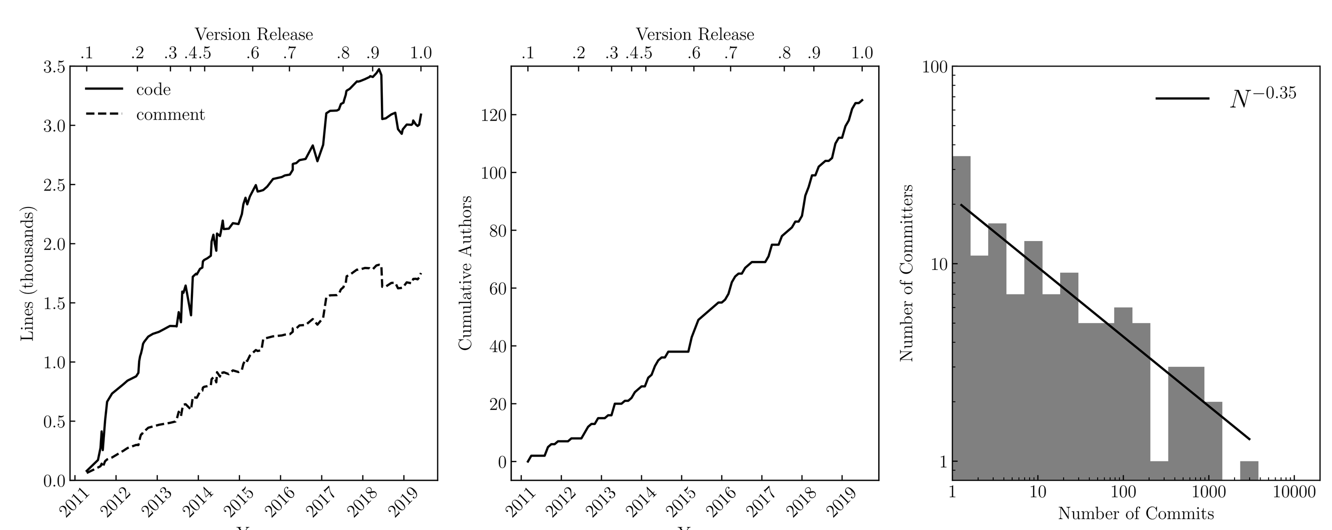
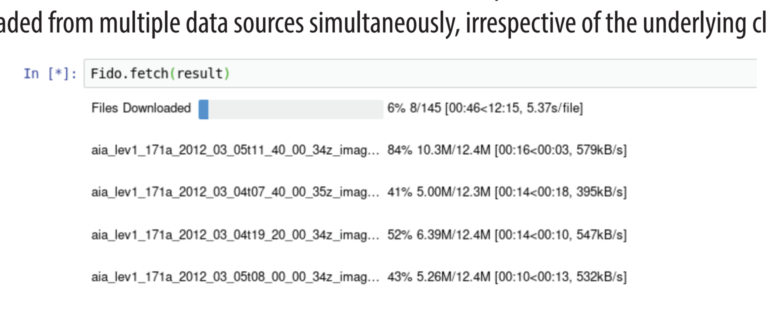


Figure. Left panel: a plot of the steady increase in the total number of lines of code (solid line) and lines of comments/documentation (dotted line) as a function of time. Major version releases are indicated along the top axis. A striking reduction in the code base occurred after version 0.9. This period saw a major code reorganization and deletion of obsolete features along with removing support for Python 2. Middle panel: the cumulative number of contributors to sunpy as a function of time shows a steady increase in the number of people involved in the development team. Right panel: a plot of the distribution of the number of commits per contributor. This distribution indicates that the majority of commits are undertaken by a small group of contributors. The average number of commits per contributor is less than 10 commits.

Data Search and Retrieval

One of the most important tasks that occurs before any analysis can take place is to search for and retrieve data. A particular science goal may require data from multiple data providers, each of which may have different methods for data search and retrieval. This heterogeneity increases the effort required by scientists to get the data they need. In order to address this issue, the sunpy.net subpackage provides interfaces to many commonly used data providers and catalogs in solar physics. The most powerful component of sunpy.net is the Fido interface for data search and retrieval. Fido provides a unified interface that simplifies and homogenizes search and retrieval by allowing data to be queried and downloaded from multiple data sources simultaneously, irrespective of the underlying client.



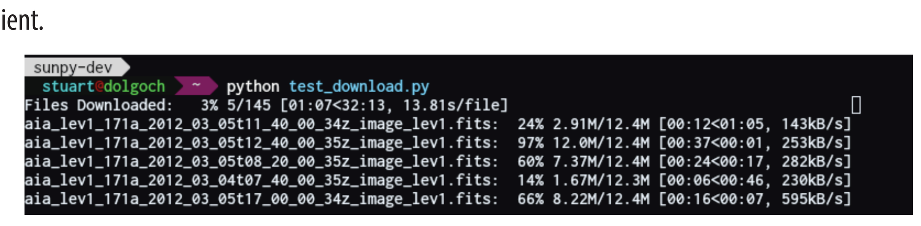


Figure. A Fido fetch command downloading files in parallel with overall and individual progress bars as seen in a Jupyter notebook (left) and in a terminal (above). This functionality is powered by the parfive package.

Currently, Fido supports the Virtual Solar Observatory (VSO; Hill et al. 2009), the Joint Science Operations Center (JSOC; see Affiliated Packages), and a number of individual data providers that make their data available via web-accessible resources such as HTTP(S) websites (RHESSI, SDO EVE, GOES XRS, PROBA2 LYRA, and NOAA sunspot number prediction) and FTP servers (NOAA sunspot number, NoRH). A Fido search accesses multiple instruments and all available data providers in a single query. Search queries optionally include a variety of attributes, such as instrument, time range, and wavelength. The attributes can be joined using Boolean operators to enable complex queries. The result of a Fido query can be inspected and edited before retrieval and is then downloaded via asynchronous and parallel download streams. Fido also recognizes failed data downloads and allows for re-requesting files that were not retrieved. The downloaded files can then be read into custom data classes.

Data Types

The sunpy package provides core data types that are designed to provide a general, standard, and consistent interface for loading and representing solar data across different instruments and missions. The two core data types currently supported in sunpy are handled by TimeSeries and Map, objects that support 1D temporal data and 2D image data, respectively. The purpose of these core classes is to standardize data structures regardless of the data source (e.g., observational data from independent instruments). The classes maintain a consistent interface for accessing data attributes such as the data array itself as well as the metadata and relevant units. These core classes also include functionality for data manipulation and data visualization.

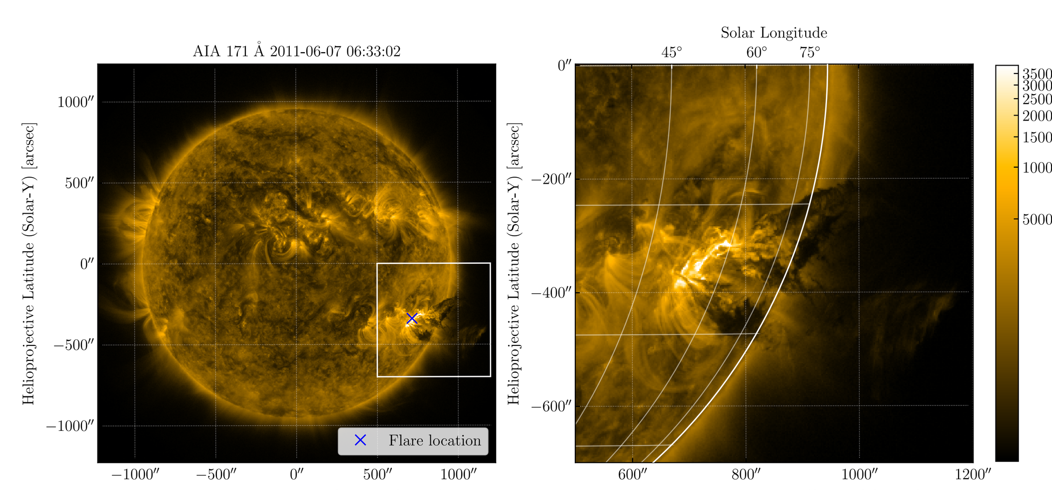


Figure. Example of a Map visualization from observations of the 171 Å wavelength channel of AIA on board SDO. The left panel shows an image of the entire Sun with the flare position as given by the HEK. The right panel shows the cropped view in the white box of the left-hand panel, focusing on an erupting flare (the same event is shown in the TimeSeries Figure below).

Map

A majority of solar data is in the form of images of the Sun. Images of the Sun are taken in multiple wavelengths from a wide range of both space- and ground-based instruments. Images also require precise coordinate information in order to compare solar features observed across multiple wavelengths with different instruments. The Map class in sunpy provides a framework to contain and analyze image data. A Map can be created from local files or using a URL to a remote data file. The Map class will automatically detect supported instruments and parse the metadata to infer the coordinate system. Other source-specific metadata is used to determine the appropriate color table and image normalization for visualization. It is also possible to create a custom Map by providing a 2D data array and metadata.

Supported by TimeSeries
Geostationary Operational Environmental Satellite (GOES) X-Ray Sensor (XRS)
Fermi Gamma-ray Burst Monitor (GBM)
Nobeyama Radioheliograph (NoRH)
PRoject for Onboard Autonomy (PROBA2) Large Yield Radiometer (LYRA)
Solar Dynamics Observatory (SDO) EUV Variability Experiment (EVE)
Reuven Ramaty High Energy Solar Spectroscopic Imager (RHESSI)

Supported by Map
COronal Solar Magnetism Observatory (COSMO) K-coronagraph (K-Cor)
Hinode X-Ray Telescope (XRT)
Interface Region Imaging Spectrograph (IRIS) Slit Jaw Imager (SJI)
PROBA2 Sun Watcher using Active Pixel System detector and Image Processing (SWAP)
RHESSI
Solar and Heliospheric Observatory (SOHO) Extreme ultraviolet Imaging Telescope (EIT)
SOHO Large Angle Spectroscopic COronagraph (LASCO)
SOHO Michelson Doppler Imager (MDI)
SDO Atmospheric Imaging Assembly (AIA)
SDO Helioseismic and Magnetic Imager (HMI)
Solar TErrestrial RElations Observatory (STEREO) Extreme Ultraviolet Imager (EUVI), COronagraph 1 and 2 (COR1/2) for both STEREO A and B
Transition Region and Coronal Explorer (TRACE)
Yohkoh Soft X-ray Telescope (SXT)

TimeSeries

Many observations in the field of solar physics consist of spatially integrated measurements as a function of time. For example, the X-Ray Sensor (XRS) instrument on board the Geostationary Operational Environmental Satellite (GOES), which is used as the classification standard for solar flares, continuously measures the disk-integrated X-ray flux as a function of time in two broadband channels. The TimeSeries class aims to accommodate such solar time series data. TimeSeries allows users to load time series data from a variety of solar instruments with appropriate units and timescales. A user can create a TimeSeries object either from data files stored locally (e.g., observational data sets acquired through Fido, see Data Aquisition), or manually from custom time series data. The data array, metadata, and units data are all stored as attributes in the TimeSeries class.

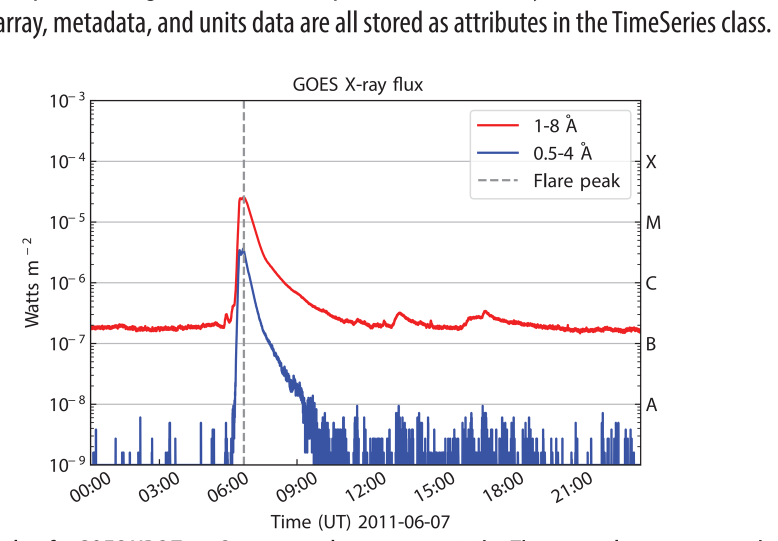


Figure. Example of a GOES XRS TimeSeries visualization over 24 hr. The two colors represent the two broadband channels; 1–8 Å (red) and 0.5–4 Å (blue). The sharp increase in flux at 06:33 UT is a solar flare and the gray dashed line indicates the time of the peak provided by the HEK.

Coordinates

The sunpy.coordinates subpackage provides support for representing and transforming coordinates used in solar physics. These coordinates may represent events (e.g., flares), features on or above the Sun (e.g., magnetic loops), or the position of structures traveling throughout the heliosphere (e.g., coronal mass ejections). The package currently implements many of the most widely used Sun-centered coordinate frames including Helioprojective Cartesian (HPC), Heliographic Carrington (HGC), and Heliographic Stonyhurst (HGS), as well as Heliocentric Aries Ecliptic (HAE), Heliocentric Cartesian (HCC), and Heliocentric Earth Equatorial (HEEQ). Additional coordinate frames will be available in future releases. The functionality provided in this package is built on top of and integrates with the astropy coordinates framework. A few example applications of the sunpy.coordinates subpackage are shown below.

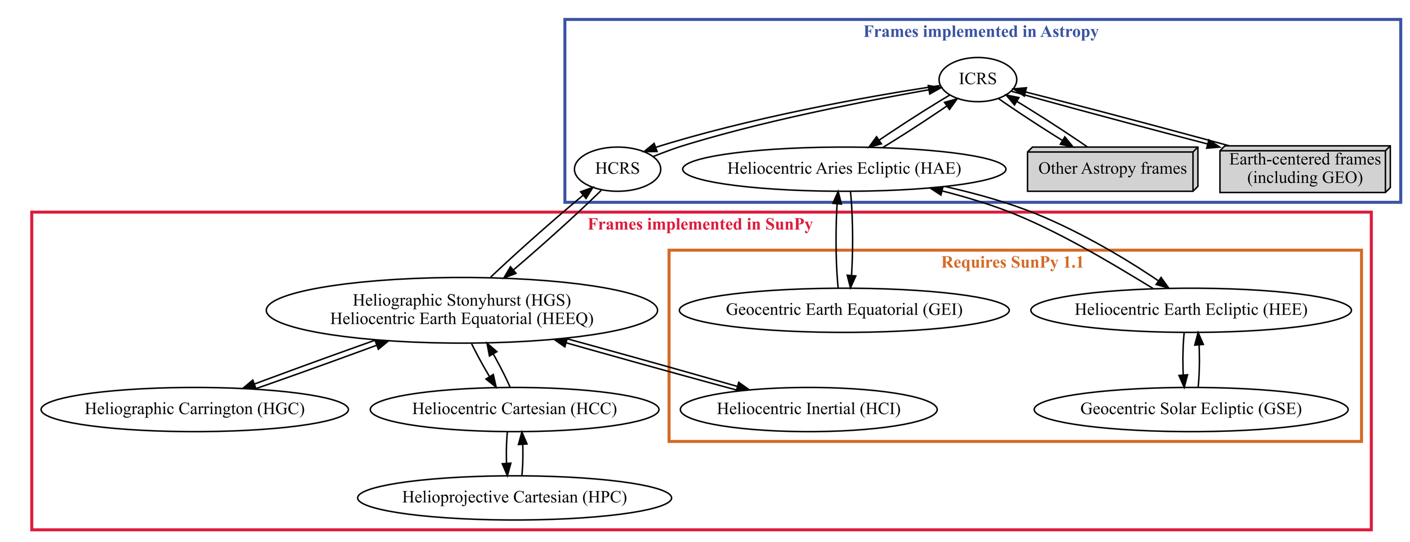


Figure. Diagram of the coordinate frames accessible through sunpy.coordinates, and how they transform between each other. The frames within the blue box are implemented in astropy.coordinates, but in the shared framework, any frame can be transformed to any other frame in this diagram.

Application of Coordinates

This section shows a number of different applications of coordinates.

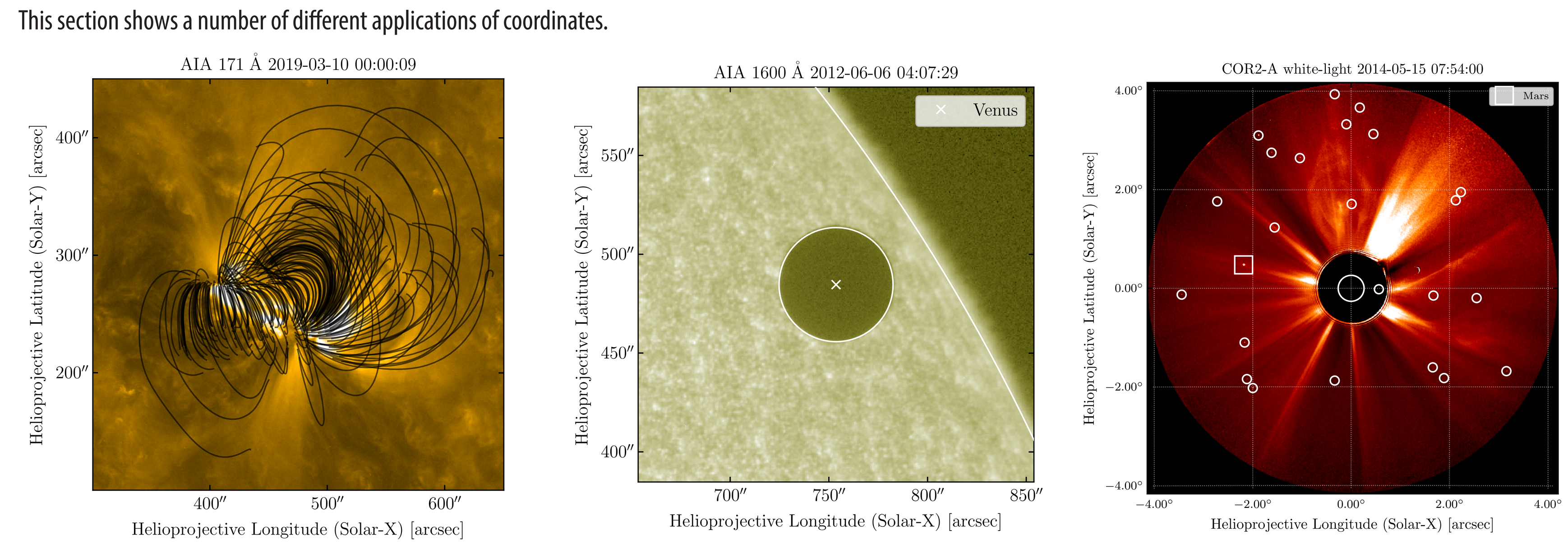


Figure: Several examples of using the coordinate machinery provided by the sunpy.coordinates subpackage. (a) Magnetic field lines traced from a potential field extrapolation overlaid on an SDO AIA 171 Å observation of an active region from 2019 March 10 00:00:09 UTC. The field extrapolation was computed with pfsspy (Stansby 2019). (b) The Venus transit as viewed by SDO AIA in 1600 Å. The predicted position of Venus is overplotted in the coordinate frame of the AIA image. (c) A coronagraph image of the solar corona as observed by STEREO-A COR-2. The predicted positions of stars from the Gaia (Gaia Collaboration et al. 2016) Data Release 2 catalog (Gaia Collaboration et al. 2018), marked by circles, as well as Mars, marked by a box, are overplotted in the coordinate frame of the image.

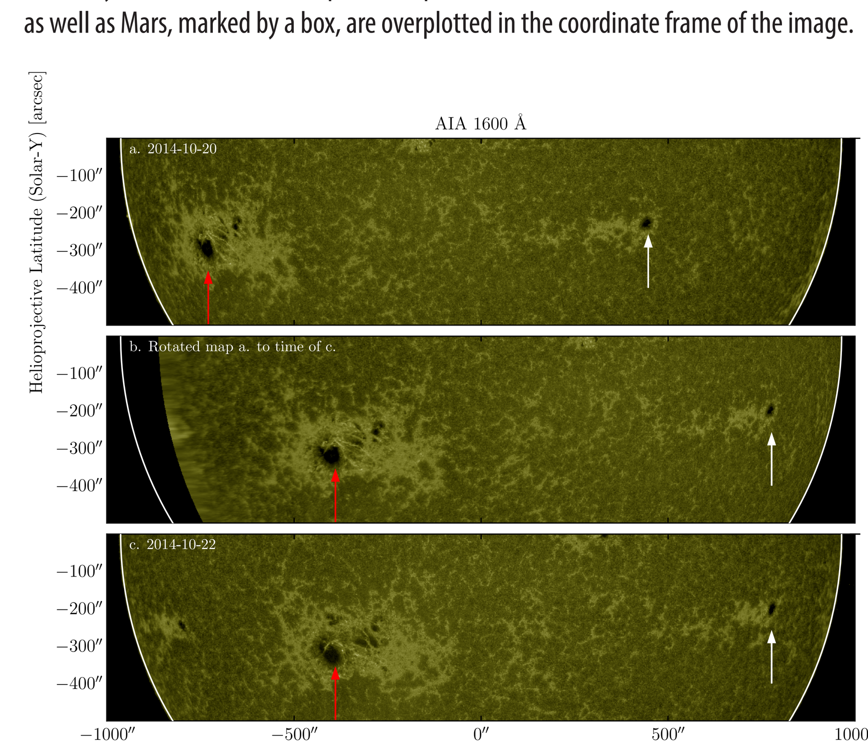


Figure. Example of transforming a Map to a different time, while accounting for differential rotation of the Sun. Panel (a) shows the Sun as observed by SDO AIA in1600 Å on a particular day. A large and small sunspot group are highlighted by a red arrow and a white arrow, respectively. Panel (b) shows the observation after transforming forward in time by two days. Panel (c) shows the real observation at the time of panel (b), which compares well to the transformed image, disregarding the magnetic evolution of the sunspot groups.

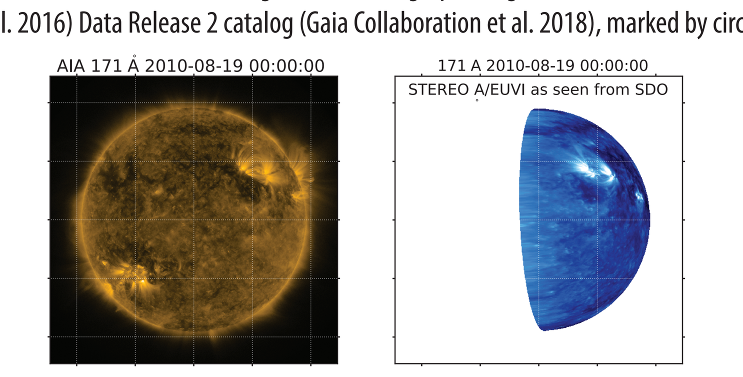


Figure. An example demonstrating how you can reproject an image to the view of a different observers in this case reprojecting a STEREO A EUVI image as seen by SDO (right). The SDO image in the same wavelength band is shown on the left for comparison.

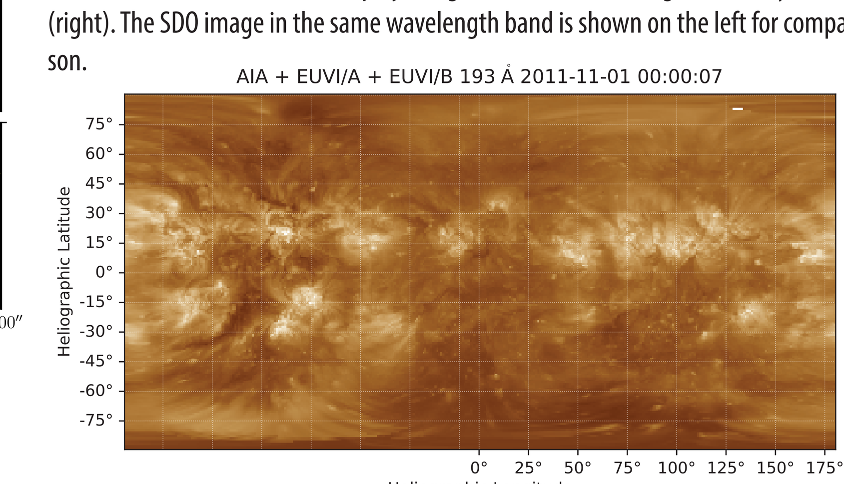


Figure. With SDO/AIA and STEREO/A and STEREO/B, it is possible (given specific dates) to combine combine three EUV images from these satellites to produce a full latitude/longitude map of the Sun. This example shows how images in helioprojective coordinates can be transformed into heliographic coordinates and overlaid. See the example gallery for the full code.

Affiliated Packages

drms

Provides access to HMI, AIA and MDI data hosted on the JSOC. The drms package enables querying the image metadata in the JSOC DRMS. It can also be used to submit tailored data export requests (e.g., movies and images in various formats) and download data files. Used by Fido.

Maintainer: Kolja Glogowski

Ndcube

This package provides functionality for manipulating N-dimensional coordinate-aware data. Support is provided for any combination of axis types such as images, images over time, spectrograms, slit spectragraphs. The package extends Astropy's NDData data container.

Maintainer: Daniel Ryan

radiospectra

This package supports reading and analyzing dynamic radio spectra, as a function of time, primarily from e-Callisto. It provides tools for downloading and reading data, handling metadata, homogenizing data, and subtracting background. This package is planned to undergo major changes to use astropy/specutils.

Maintainer: David Pérez-Suárez

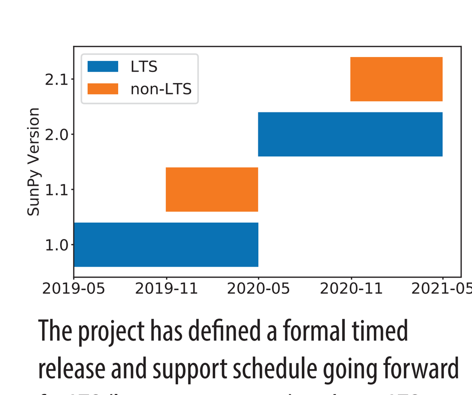
IRISpy

The IRISPy package provides tools to read, manipulate, and visualize data from the NASA Small Explorer mission, IRIS. This package provides data classes which hold data from SJI and the slit spectrograph. Built on top of the functionality provided by ndcube.

Maintainer: Daniel Ryan

Conclusions

Development of the sunpy core package has been ongoing for 8 yr, with the adoption of a formal project structure 5 yr ago. The core package has grown to provide significant and now mature functionality for a growing number of users. The release of version 1.0 is a significant milestone, and comes with a commitment to stability in future releases. Significant additional features are either being actively developed or are planned for future development. The SunPy project is a member of the Python in Heliophysics Community and its development is consistent with the PyHC standards. SunPy is a NumFocus-sponsored project. We thank the members of the community that have contributed to the SunPy project, that have opened issues and provided feedback, and that have supported the project in a number of other ways.



The project has defined a formal timed release and support schedule going forward for LTS (long term support) and non-LTS releases.

What's new in version 1.1?

- Remote data manager
- Support for GOES/SUVI
- New coordinate frames (HEE, GSE, HCI, GEI)
- Initial support for PSP/WISPR files
- Import time for sunpy.map reduction by 40%.

SunPy is a volunteer-based organization. If you make frequent use of sunpy or want to support the project consider making a donation!

The paper this poster is based on is available free online and includes the code to generate most figures. Scan the qrcode to the right.