ISSN 1845–8319

SUNPY: PYTHON FOR SOLAR PHYSICS. AN IMPLEMENTATION FOR LOCAL CORRELATION TRACKING

J. I. CAMPOS ROZO[1] and S. VARGAS DOMINGUEZ[2]

[1] *Universidad Nacional de Colombia, Bogotá, Colombia*
[2]*Big Bear Solar Observatory, NJIT, 40386 North Shore Lane Big Bear City, CA 92314-9672, U.S.A.*

Abstract. Python programming language has experienced a great progress and growing use in the scientific community in the last years as well as a direct impact on solar physics. Python is a very mature language and almost any fundamental feature you might want to do is already implemented in a library or module. SunPy is a common effort of, using the advantages of Python, developing tools to be applied for processing and analysis of solar data. In this work we present a particular development, based on Python, for the analysis of proper motions in time series of images through the local correlation tracking algorithm. A graphic user interface allows to select different parameters for the computations, visualization and analysis of flow fields.

Key words: Programming - Solar Physics - Data Analysis

1. What is Python?

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python is an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, http://www.python.org, and may be freely distributed. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. Some of the attractive characteristics of Python have made the widespread use of this language in recent years. They can be summarized in the following. Learning: Python is easy to deal with (IDL too, but Python is a widely used general-purpose,

high-level, object-oriented, interpreted and functional programming language). Multi-purpose: packages for image processing (Scikit-image), neuronal network (Scikit-learn), 3D-display (Mayavi), symbolic math (Sympy), web-development (e.g. Django, Pylons) and a lot more.

Some of the already developed scientific core packages are :

- Scipy: a collection of open source software for scientific computing in Python.
- Numpy: a powerful N-dimensional array object, useful linear algebra and other tools for integrating functions.
- Matplotlib: a Python 2D plotting library.
- Pandas: easy-to-use data structures and data analysis tools for the Python programming language.
- Astropy: the Astropy Project is a community effort to develop a single core package for Astronomy in Python.

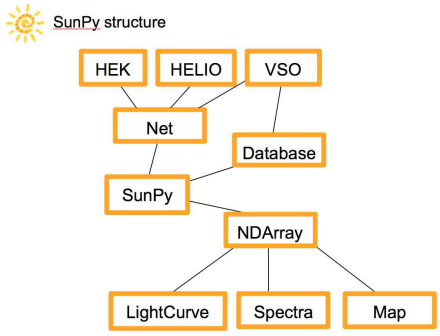

Figure 1: Structure of SunPy.

2. SunPy

SunPy is a free and open-source software library for solar physics based on Python. It is a community-developed free and open-source software package for solar physics research. SunPy is meant to be a free alternative to the SolarSoft data analysis environment which is based on the IDL scientific programming language. The aim of the SunPy project is to provide the software tools necessary so that anyone can analyze solar data. SunPy is written using the Python programming language and is built upon the scientific Python environment which includes the core packages listed above. The development of SunPy is associated with that of Astropy. SunPy was founded on March 28, 2011 by a small group of scientists and developers at the NASA Goddard Space Flight Center. Thanks to the generous support of the ESA Summer of Code and the Google Summer of Code as well as contributors from around the world, SunPy is now a global project and is not associated with any individual institution.
One of the helpful characteristics of SunPy is the capability to make tests and version controls. The tests are done with Pytest (a library written in Python) and the version control is done with Git. These tools allow us to make fewer errors and avoid repeating the same code from different developers. There are several ways to install SunPy in all different operation systems (Windows, MacOs, Linux, Solaris, etc). Figure 1 shows the global building structure of SunPy. SunPy has three mean SuperClasses or packages (Net, Database, NDArray). With Net you can obtain data from Heliophysics Event Knowledgebase (HEK), Heliophysics Integrated Observatory (HELIO) and Virtual Solar Observatory (VSO) databases. With the Database package we can save and organize our data into a local or remote database and this function is connected with VSO network. The last package, NDArray, has three sub-packages: (1) Lightcurves (still under development) which already support SDO/EVE, GOES XRS and PROBA2/LYRA. (2) Spectra is the newest and the least developed but there are some functions to work with Callisto data. (3) Map, to make maps of SDO, LASCO, SOHO, STEREO, RHESSI, YOHKOH, IRIS, TRACE. At the moment, J. I. Campos Rozo is working on Hinode/SOT Map (together with Dr. David Perez-Suarez from SANSA, South African National Space Agency).

3. Examples

Let us develop an example for the use of the *Map* function as shown in Fig. 2. To start up using some of the SunPy functions we need to import the libraries, such as Numpy, Scipy, Matplotlib. One important thing is that when you import any library or class you can change its name if wanted. Next step is to build a *Map*. If the *Map* you want to create does not belong to the list of different satellites and instruments allowed, that function won't be built. The *peek* function plots the map with title and axis. We can also compose maps, for example, we can combine SDO/HMI with RHESSI images by using the *compositeMap* function, that function use two objects obtained via *Map* function. We can create a data cube by using *Map* function and adding the keyword *cube=True*. The created cube can be saved in video format, e.g. mp4.

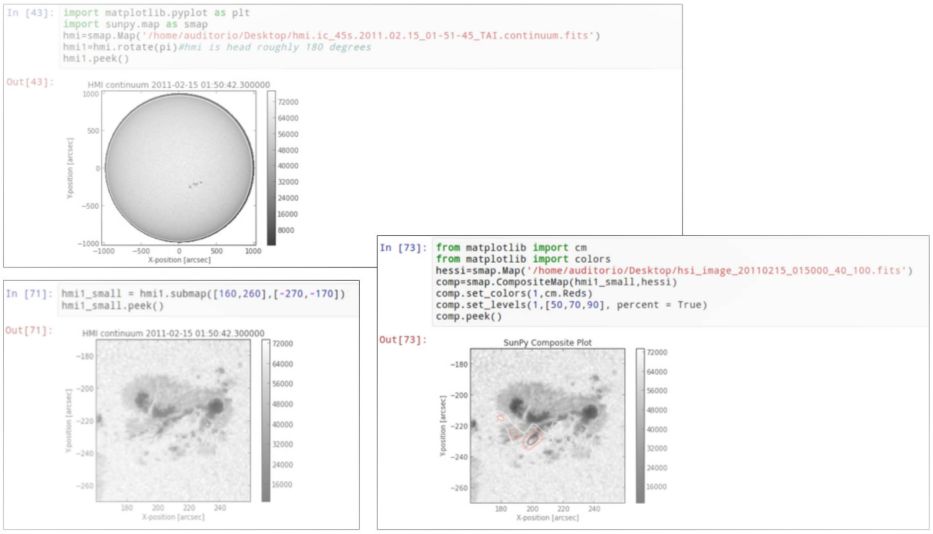

Figure 2: Example of the use of the *Map* function in SunPy.

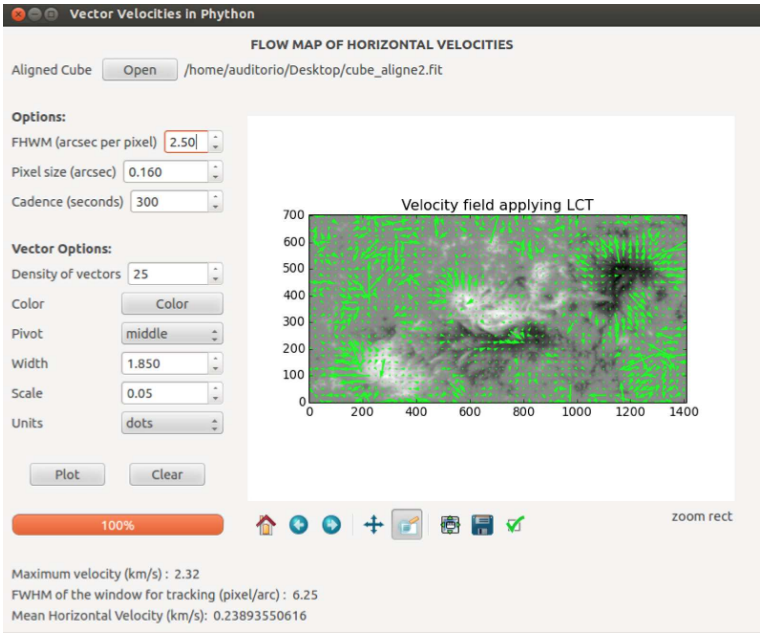

Figure 3: Python GUI to control the application of local correlation tracking algorithm to a time series of images. Different parameters can be selected to apply the algorithm and for the visualization of the velocity output.

4. Implementation for Local Correlation Tracking

We are developing a solar physics hotbed research project at Universidad Nacional de Colombia. Besides from developing the routine SOTMap for SunPy (available in the repository https://github.com/Hypnus1803), Python routines for computing proper motions of structures in a time series of images have been implemented. The Local Correlation Tracking (LCT, November and Simon, 1988) technique is a robust method used to study the dynamics of structures in a time series of images. By tracking pixel displacements, using a correlation window, LCT can determine proper motions and generate flow maps of horizontal velocities. This procedure is used to study the dynamics of plasma in the solar photosphere at different spatial scales, e.g the analysis of granular and supergranular convective cells, meridional flows, etc. As part of this project, a widget implemented in Python was

developed. It generates a user-friendly graphical user interface (GUI) to control various parameters for the process of calculating flow maps from a series of filtergrams (data cube). Figure 3 shows the GUI where the user can select a data cube to be processed and determine the size of the structures wanted to be tracked, among other options.

At the moment additional features are being implemented for supplementary analysis of the data. In particular, the possibility to select different time windows and create multiple flow maps in time in order to study the evolution of flow patterns from one map to the other. Further analysis of the velocity outputs is also under development, i.e to obtain statistics for selected sub-regions in the field of view and the distribution of velocities, and the computation of the divergence field and vorticity.

Acknowledgements

This research has made use of SunPy, an open-source and free community-developed solar data analysis package written in Python. The authors want to thank Dr. Dominik Utz and Dr. Arnold Hanslmeier for the invitation to the 4th International Workshop on Small-Scale Solar Magnetic Fields in Austria.

References

November, L. J. and Simon, G. W.: 1988, *Astrophys. J.* **333**, 427.
http://www.sunpy.org/
https://www.python.org/
http://www.numpy.org/
http://www.scipy.org/
http://matplotlib.org/
http://pyqt.sourceforge.net/Docs/PyQt4/index.html
http://sdac.virtualsolar.org/
http://www.lmsal.com/hek/
http://helioviewer.org/