

Sunpy - Python para física solar: una implementación para seguimiento de correlaciones locales

José Iván Campos Rozo ^{a *}, Santiago Vargas Domínguez ^{a **}

^aObservatorio Astronómico Nacional, Universidad Nacional de Colombia.

Aceptado Diciembre de 2014; Publicado en línea Marzo 2015.

ISSN 2256-3830.

Resumen

Python, un lenguaje de programación de alto nivel, ha experimentado un rápido crecimiento y una gran aceptación en la comunidad científica en los últimos años. En especial este lenguaje tiene actualmente un impacto directo y fuerte en la investigación en física solar. Python ha madurado considerablemente y un gran número de aplicaciones fundamentales que se quieran usar, están implementada en alguna de las librerías o módulos. Sunpy, es un esfuerzo en común de la comunidad solar internacional de programadores avanzados en Python, para el desarrollo de herramientas que se aplican en el procesamiento y análisis de datos satelitales solares. En este trabajo se presenta un desarrollo basado en Python, para el cálculo y análisis de movimientos propios en series temporales de imágenes a través de un algoritmo de seguimiento de correlaciones locales. Una interfaz gráfica nos permite seleccionar diferentes parámetros para la computación, visualización y análisis de los campos de flujo.

Palabras Claves: Programación, física solar, análisis de datos.

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.

Keywords: Programming, Solar Physics, Data Analysis.

* jicamposr@unal.edu.co

**svargasd@unal.edu.co

1. Introducción

1.1. ¿Qué es Python?

Python es un poderoso lenguaje de programación de fácil aprendizaje[3]. Es muy eficiente para la lectura de estructuras de datos de alto nivel y es una muy sencilla pero efectiva aproximación programación orientada a objetos. Python es un lenguaje ideal para desarrollar códigos y aplicaciones de forma fácil y rápida en muchas áreas y la mayoría de las plataformas. El interpretador de Python y la amplia existencia de librerías estandar están disponibles de forma libre en forma fuente o binarios para todas las plataformas en la página web de Python y es de libre distribución. El interprete de Python es fácilmente extendible implementando funciones de C o C++. Python también es muy favorable para realizar desarrollo de aplicaciones. Varias de las características atractivas de Python lo han convertido en un lenguaje propicio para la ciencia y el análisis de datos en los años recientes. Estas caraterísticas pueden ser resumidas a continuación:

Aprendizaje: Python es de fácil aprendizaje (IDL también lo es, pero Python es de proposito general y multi-paradigma). Tiene multipropósitos como: Scikit-image (paquete para procesamiento de imágenes), Scikit-learn (red de trabajo neuronal), Mayavi (visualización 3-D), Sympy (matemática simbólica), Django (desarrollo web) y muchos más. Para física solar los paquetes científicos mas importantes que usamos para el desarrollo de librerías son:

- Scipy: Una colección abierta para computación científica en Python [5].
- Numpy: Una poderosa herramienta de manejo de arreglos N-dimensionales, herramientas de integración y algebra lineal y más.[4]
- Pandas: Librería para fácil uso de estructuras de bases de datos y análisis de datos.
- Matplotlib: Librería de visualización 2-D.[6]
- Astropy: El proyecto Astropy es un esfuerzo de la comunidad pythonera de desarrollar librerías para astronomía.

1.2. SunPy

SunPy[2] es una librería libre y de acceso abierto para física solar basada en Python. SunPy intenta ser una alternativa libre y de acceso abierto como contra parte a SolarSoft, el ambiente de análisis de datos basado en IDL, otro lenguaje de programación científico. El propósito de Sunpy es brindar a los usuarios herramientas para la visualización y análisis de datos solares. Sunpy es construido usando Python y las librerías mencionadas arriba como resultado del desarrollo de Astropy. Gracias a contribuidores de ESA code of Summer y el Google Summer of Code, Sunpy es hoy en día un proyecto mundial y no está asociado a ninguna insitución individual.

Una de las herramientas más útiles de Sunpy es la herramienta de *test* y el de control de versiones. La herramienta de *test* está hecha a través de la librería Pytest (librería escrita en Python) y el control de versiones usando Git, y GitHub (una red social para el control de versiones). Estas herramientas nos permiten cometer un mínimo de errores y controlar que no estemos escribiendo funciones u objetos con las mismas finalidades repetidas sin darnos cuenta. Hay varias maneras de instalar Sunpy dependiendo del sistema operativo (Window, MacOS, Linux, Solaris, etc.). Sunpy está dividido en super clases o paquetes divididos en clases para manipulación de arreglos, paquetes para manipulación de datos online y de bases de datos. Para los arreglos tenemos, mapas, curvas de luz y espectros en radio. También podemos descargar datos de las bases VSO[8], HEK y HELIO.

1.3. Ejemplos

Mostraremos un ejemplo del uso de los mapas usando Sunpy como se muestra en la Figura 1.3. Lo principal es importar las librerías núcleo de Sunpy. Cuando la imagen solicitada no pertenece a un satélite aceptado por Sunpy, sus ejes pueden tener algunos errores. Otra de las herramientas usadas en Sunpy es la composición de mapas. A continuación mostraremos una composición entre un mapa de SDO/HMI y un mapa de RHESSI usando la función *compositeMap*.

2. Seguimiento de correlaciones locales

Se desarrolló un proyecto semillero de investigación para la Universidad Nacional de Colombia, basado en una herramienta con interfaz gráfica para el cálculo de movimientos de flujos fotosféricos en el Sol usando series temporales de imágenes satelitales solares. También se aportó con el desarrollo de lectura de mapas generados por el instrumento SOT en el telescopio espacial Hinode. La técnica de seguimiento de correlación local es un código robusto usado para el estudio de dinámicas de estructuras en imágenes de series temporales. Se usa una ventana de correlación que se mueve a través de los píxeles de la imagen. La técnica de correlación local permite determinar movimientos propios y generar mapas de flujos de velocidades horizontales[1].

Este procedimiento se puede usar para el estudio de la dinámica del plasma fotosférico en diferentes escalas espaciales, por ejemplo, análisis granular y celdas convectivas supergranulares, flujos meridionales, etc.

3. Implementación de una interfaz gráfica

Como parte de este proyecto, se implementó una interfaz gráfica (*widget*) para facilitar el acceso, manejo y al análisis de las series temporales de imágenes solares, desarrollada en Python. Por medio de la interfaz se pueden controlar, de manera sencilla, varios parámetros para el procesamiento y el cálculo de los mapas de flujos de las secuencias de filtrogramas (cubos de datos). La Figura 3 muestra un ejemplo de los resultados obtenidos y de como seleccionar los parámetros de procesamiento. Los resultados de la aplicación del código desarrollado permiten evidenciar el comportamiento de los mapas de flujo fotosférico, al igual que el campo de divergencias o campos de vórtices en las series de imágenes. El análisis de la dinámica fotosférica es importante para determinar las condiciones del plasma solar, y su evolución temporal. Recientes estudios usan la técnica de correlaciones locales para determinar por ejemplo vórtices a pequeña escala en la superficie solar [9].

References

- [1] November, L. J. and Simon, G. W.: 1988, *Astrophys. J.* 333, 427.
- [2] <http://www.sunpy.org/>
- [3] <https://www.python.org/>
- [4] <http://www.numpy.org/>
- [5] <http://www.scipy.org/>
- [6] <http://matplotlib.org/>
- [7] <http://pyqt.sourceforge.net/Docs/PyQt4/index.html>
- [8] <http://sdac.virtualsolar.org/>
- [9] Vargas Domínguez, S., Palacios, J., Balmaceda, L., Cabello, I., Domingo, V. *Monthly Notices of the Royal Astronomical Society*, Volume 416, Issue 1, pp. 148-154.

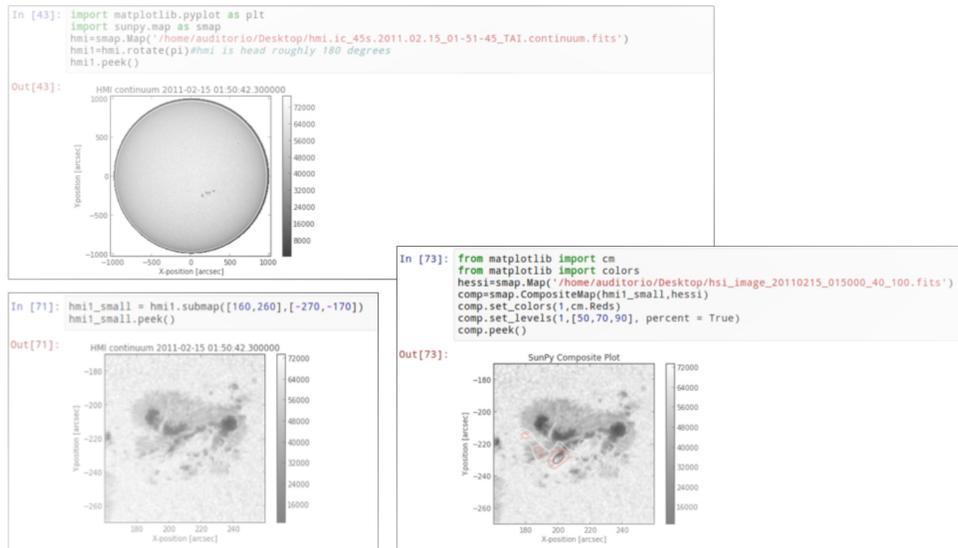


Fig. 1. Ejemplos del uso de la función *Map* en Sunpy.

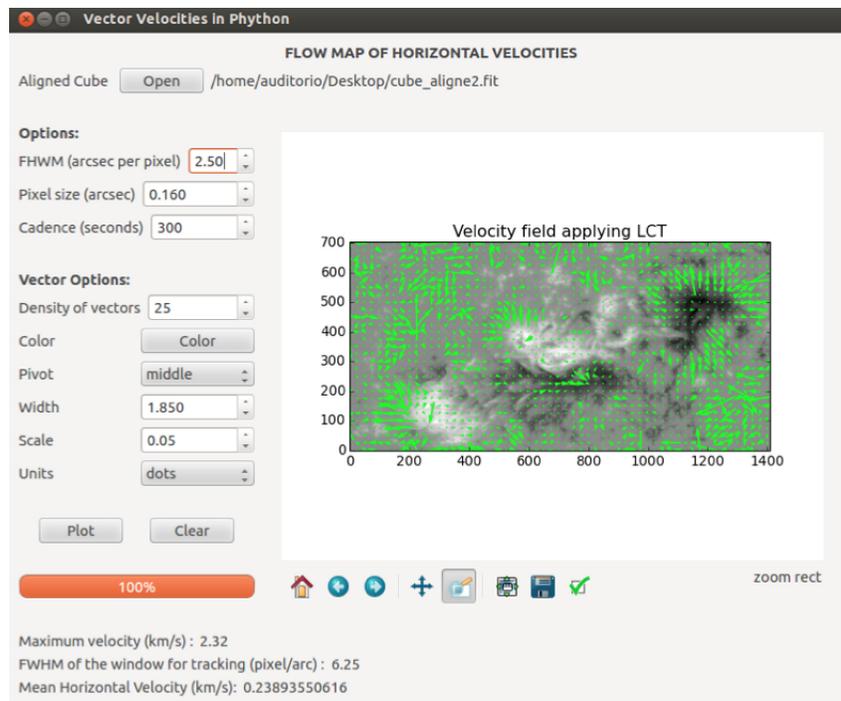


Fig. 2. Python GUI para controlar parámetros en la aplicación del método de seguimiento de correlación local.