



SunPy: New Scientific Analysis Capabilities for GOES Observations



ROB/STCE

Daniel Ryan¹, Stuart Mumford², Steven Christe³, David Perez-Suarez⁴, Andrew Inglis³, Marie Dominique¹

¹Royal Observatory of Belgium, ²University of Sheffield, ³NASA Goddard Space Flight Center, ⁴South African National Space Agency

Abstract

The SunPy project is a new open-source software library for solar physics using the Python programming language and is becoming increasingly useful for scientific analysis. SunPy can now be used to derive temperature and emission measure from GOES/XRS observations. The GOES/XRS series has been consistently observing the Sun in soft X-rays since the 1970's and has become the most popular instrument with which to analyse the thermal coronal plasma in solar flares. This new capability represents a significant step forward in making python and SunPy a viable alternative for all aspects of solar physics data analysis.

Why Python?

- free and open source
- inherent readability encourages easier collaboration.
- thriving ecosystem of users and support
- Highly transferable skill both within and beyond science.
- Good mixture of high and low level programming for easy use or fast code, e.g. by calling C.

Theory

- Temperature and emission measure are calculated using White et al. (2005), the same as in Solar SoftWare (SSW).
- The same CHIANTI-generated tables relating temperature, emission measure are GOES flux ratio are used. Results same as SSW.

Deriving Temperature and EM with SunPy

There are two ways calculate GOES temperature and emission measure in SunPy. The required functions are located in the sunpy.instr.goes module. The resulting units of temperature are MK and emission measure cm^{-3} .

1) temp_em() function and GOESLightCurve object

```
>>> from sunpy.instr.goes import temp_em
>>> import sunpy.lightcurve as lc # Import the modules we need
# Create GOESLightCurve object
>>> glc = lc.GOESLightCurve.create('2011-06-07 06:00', '2011-06-07 08:00')
# Create new object with temperature and EM using the temp_em() function.
>>> glc_new = temp_em(glc)
# new values in glc_new.data.temperature and glc_new.data.em
```

2) goes_chianti_tem() function and arrays of GOES flux

```
>>> from sunpy.instr.goes import goes_chianti_tem.
# Let flux1_8 and flux05_4 are numpy arrays of GOES fluxes.
# Create numpy arrays for temperature and EM.
>>> temp, em = goes_chianti_tem(flux1_8, flux05_4)
```

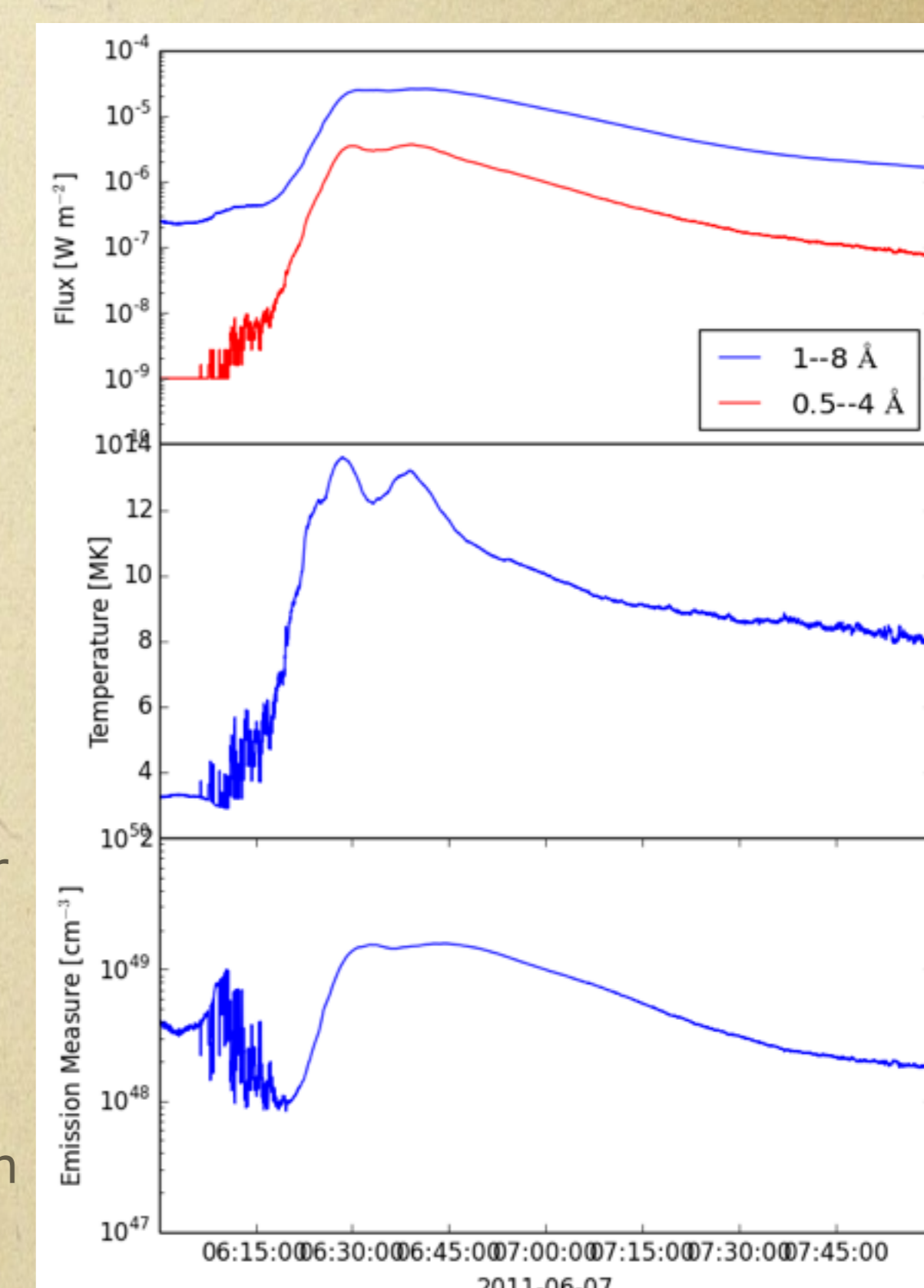


Figure 1. GOES flux for the 2011 June 7 flare (top) and resulting temperature (middle) and emission measure (bottom) calculated with SunPy.

Coming Soon: Calculate GOES Radiative Loss Rates and Integrated and Cumulative Radiated Losses.

See sunpy.org to install, use, and contribute.