A FASTER WAY TO ACCOUNT FOR PIXEL FOOTPRINT PROJECTION ON PLANETARY SURFACES WITH PYTHON: THE SPiP MODULE.

A. Stcherbinine[1], C. S. Edwards[1], K. Saboi[1], N. M. Smith[1], C. Haberle[1].
[1]Department of Astronomy and Planetary Science, Northern Arizona University, Flagstaff, AZ, USA (aurelien.stcherbinine@nau.edu),

Introduction: *SPiP* (*Spacecraft Pixel footprint Projection*) [1] is a Python 3 module dedicated to the projection and handling of the spatial footprint of the pixels of an orbital instrument on a planetary surface.

Here, we will focus on the processing of data from the Emirates Mars Infrared Spectrometer (EMIRS) instrument onboard the Emirates Mars Mission (EMM) “Hope” probe [2, 3], as it has been the main purpose for the realization of this module. But this may apply to other instruments and/or targets.

Rationale: Considering the spatial extent of the pixel footprints is important, especially when their size is larger than the intended spatial resolution. EMIRS pixels have an instantaneous field of view (IFOV) of 5.4 mrad [3], which leads to pixel sizes typically between 100 and 300 km, or up to tens of degrees in longitude/latitude for high emission angles (see figure 22 of [3] and figure 1a), which is significantly larger than the resolution of 0.5° per pixel of the derived global maps (L3 gridded products) [4].

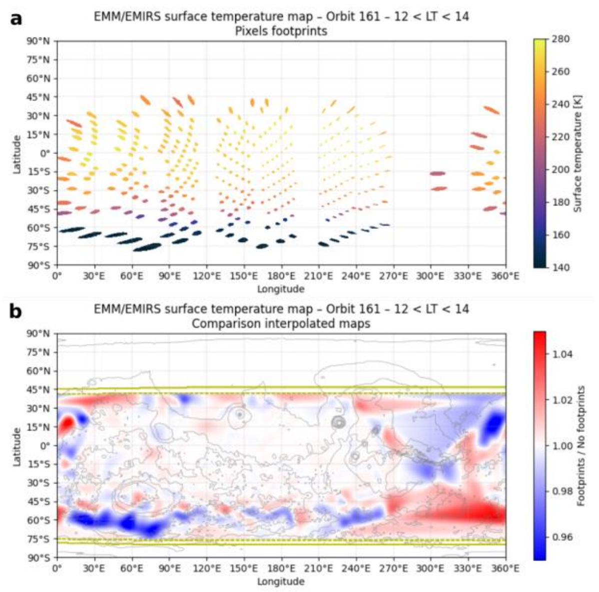
a
EMM/EMIRS surface temperature map - Orbit 161 - 12 < LT < 14
Pixels footprints
Latitude
Longitude
Surface temperature [K]
b
EMM/EMIRS surface temperature map - Orbit 161 - 12 < LT < 14
Comparison interpolated maps
Footprints / No footprints

Figure 1 *– Illustration of the impact of the pixel footprint extension.* ***a****. Pixels footprints.* ***b****. Ratio between a map of the surface temperature generated by considering the pixel footprint, and one obtain using only the center point of the pixels.*

Impact of the pixel spatial extension on retrievals. As we can see in figure 1, if considering the spatial extension of the pixels footprints does not significantly affect the results at low emission angles (small circular footprints), it results in changes of a few percent for high emission angles (elongated elliptical footprints), which corresponds here to variations up to +/- 10 K for the surface temperature. In addition, considering the actual footprints of the pixels also extends the northern and southern latitudes of a few degrees (dotted vs solid yellow lines in figure 1b).

Why this module? Along with all the geometry information, the shapes of the EMIRS pixels footprints are computed using the SPICE kernels [5, 6] and provided to users as 20-points polygons. However, the use of these polygons to generate composite gridded maps is not straightforward and is significantly time-consuming. Thus, we decided to implement a faster way to determine which pixels of a longitude/latitude grid are within an EMIRS pixel footprint using 3D trigonometry.

How it works. For one pixel, all we need is:

- The center longitude & latitude of the pixel
- The sub-spacecraft longitude and latitude
- The distance between the instrument and the surface
- The radius of the planet
- The IFOV of the instrument for one pixel

Then, assuming a perfectly spherical planet, the pixel footprint projected on the surface is an ellipse, entirely characterized by its equation in spherical coordinates. Thus, determining if a point (defined by its lon/lat coordinates) is within the pixel footprint is reduced to solving the equation to test if a point is within an ellipse in spherical coordinates.

By using *numpy arrays* that contain the longitude/latitude grid on which we want to project the map (i.e., 0°E – 360°E / -90°N – 90°N), and knowing the parameters of the projected ellipse, we can identify the pixels of the map covered by the EMIRS pixel footprint in one single operation. It will result in a boolean array flagging the entries of the lon/lat grid that are covered by the instrument footprint.

Using SPiP. If you want to use SPiP to work with EMIRS data, the easiest way is to use the `emirs_ifov_px_projection()` function of the module that takes as arguments the lon/lat grid and the pixel geometry information mentioned above, and returns a boolean mask array with the same shape as the lon/lat grid (or `emirs_ifov_multi_px_projection()` if dealing with multiple pixels). Otherwise, one can also use the `params_ellipse_fov()` function to compute the projected ellipse parameters and pass them to `in_ellipse_spherical()` to generate the boolean array.

Conclusion and perspectives: The SPiP module provides a new and easy way to account for the spatial extent of pixel footprints of orbital instruments on a planetary surface. It is already used within the EMM/EMIRS team, especially to generate the L3 gridded products that are released by the instrument team. Even though it currently assumes a perfectly spherical planet, new versions should improve the representation of the planetary body using an ellipsoid.

Acknowledgments: The source code of the SPiP module is freely available on GitHub at https://github.com/NAU-PIXEL/spip.

References: [1] Stcherbinine, A (2023) *Zenodo*, https://doi.org/10.5281/zenodo.7714205. [2] Amiri, H. et al. (2022) *SSR*, *218*, 4. [3] Edwards, C. et al. (2021) *SSR*, *217*, 77. [4] Smith, N. et al. (2023) This conference. [5] Acton, C. et al. (1996) *PSS*, *44*, 65-70. [6] Acton, C. et al. (2018) *PSS*, *150*, 9-12.