Build your own Planck LFI maps

Elina Keihänen, Department of Physics, University of Helsinki

on behalf of the **Planck Collaboration**

What is Plot_4Dmap

Plot_4Dmap is a software package that allows the user to quickly construct and plot maps of Planck LFI data, for an arbitrary pointing period range and combination of radiometers. The software can be run on a normal laptop.

Required inputs are the so called 4D maps, which contain the LFI data in compressed form. Both the software and the input data objects will be distributed via Planck Legacy Archive.

The code produces a TGA image of the map and opens it on screen. The map can also be stored as a FITSIO file in standard Healpix format for further processing.

4D maps

Despite the name, 4D maps have nothing to do with relativity. The name refers to the 4 coordinates of the data object: pointing angles θ, ϕ, ψ, and pointing period.

One 4D map object contains the calibrated data for one LFI horn in compressed form. Before constructing the 4D object, the time-ordered data is destriped to clean it of $1/f$ noise. The data is then binned to cells defined by a Healpix pixel at resolution Nside=1024, and the beam orientation. The binned data, arranged according to pointing period (PID), constitutes the 4D map object.

One 4D map object at resolution Nside=1024 takes roughly 8 GB of disk space. The full four-year LFI data set takes up 90 GB.

The 4D map objects for all LFI horns will be available through PLA.

Requirements

To install and use the software, one needs a C++ compiler, and a basic image viewer tool that can display a TGA image, such as 'open' on a Mac. All required libraries, including the FITSIO library, are included in the code package. The code has been tested on Mac OS X and linux environments.

To store the input 4D map objects, 90 GB of disk space is needed.

Examples of practical usage

Example 1: Temperature map

The following command constructs and plots a temperature map for pointing periods 3-1000 (roughly one month), using data from 70 GHz horn LFI-18.

```
plot_4Dmap bin1=LFI18 startPID=3 endPID=1000
```

'LFI18' is a symbolic link to the actual 4Dmap file. Resolution of the output map is Nside=1024 by default. The command produces the following plot.
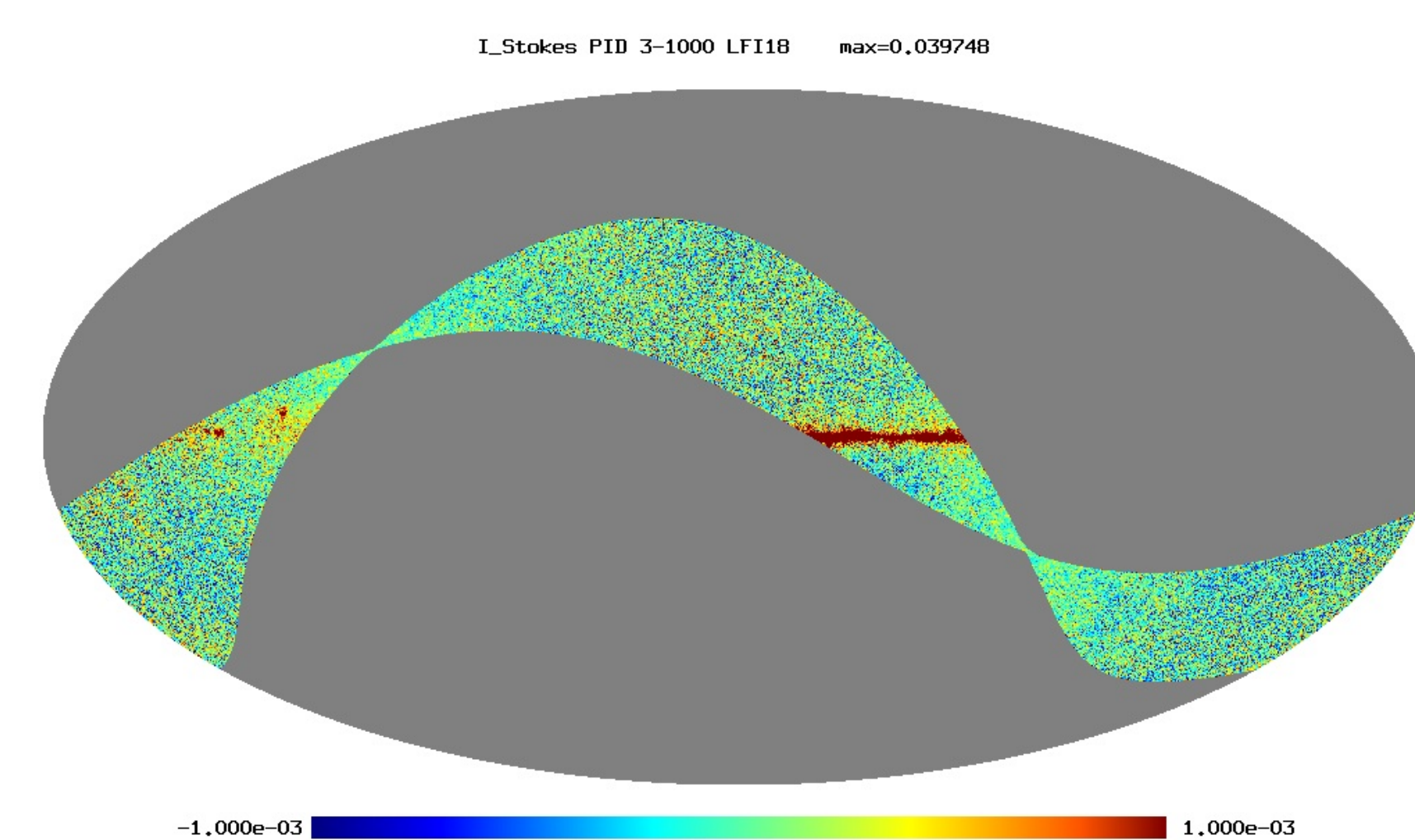

Example 2: Polarisation map

Data from several horns can be combined to one map. To make a polarisation map, one must combine at least two horns with complementing polarisation angles.

The following command constructs a U polarisation map ($sig=2$) at resolution Nside=256 for pointing periods 3-21482 (first two years of the mission), using data from 30 GHz horns LFI-27 and LFI28. 1 deg (FWHM) gaussian smoothing is applied to the plot to suppress noise.

```
plot_4Dmap bin1=LFI27 bin2=LFI28 \
        startPID=3 endPID=21482 \
        fwhm=60 nside=256 sig=2
```
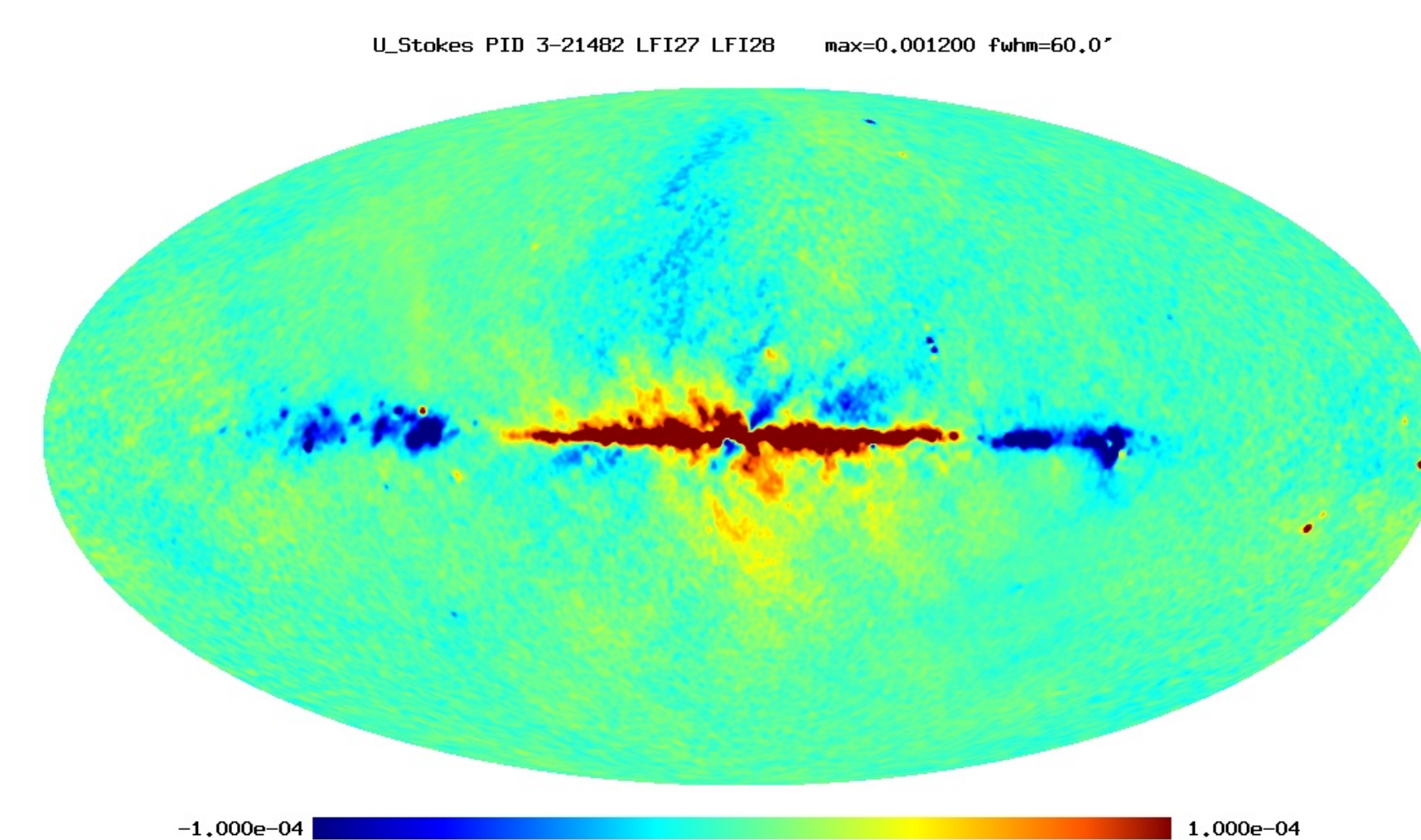

Example 3: Storing the output map in file

The output map can be written into a FITSIO file in standard Healpix format.

The following command constructs a map of for 44 GHz horns LFI-25 and LFI26, for combined odd surveys (1,3,5,7). The shortcut keyword *survey* is used to define the pointing period range. The map (I,Q,U) is written into FITSIO file for further processing. As auxiliary information, also a white noise covariance matrix for each pixel and hit count map are stored.

```
plot_4Dmap bin1=LFI25 bin2=LFI26 \
        survey=1357 \
        out_map=map2526_odd.fits \
        out_wncov=cov2526_odd.fits \
        out_hitmap=hits2526_odd.fits
```
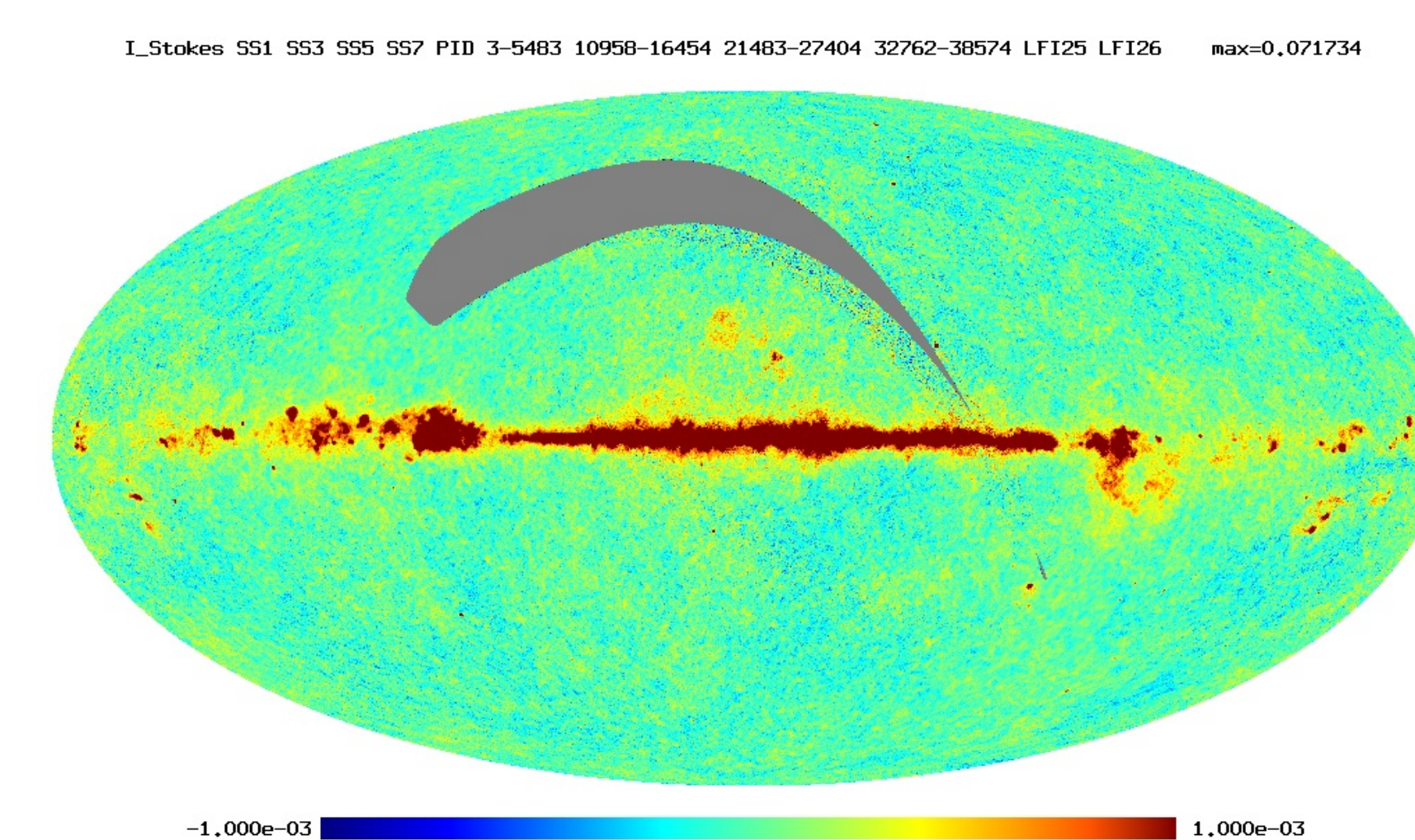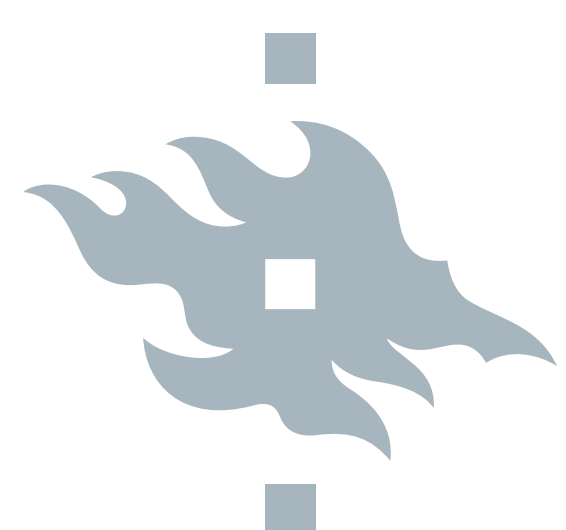

Example 4: Point source

Beside mollview plots, plot_4Dmap can be used to produce gnomonic plots. The image data can be stored in file for further processing.

This example produces a gnomonic plot (*pro=2*) centered around point lat=-5.8 deg, lon=184.6 deg in galactic coordinates (Crab nebula). The plot uses data from horn LFI-28 and survey 2. The size of one image pixel is chosen to be 0.5 arc min, making the size of the whole image 1024*0.5 arcmin = 8.53 degrees. The image is written into an ascii file.

```
plot_4Dmap bin1=LFI28 pro=2 \
        lat=-5.8 lon=184.6 \
        res=0.5 survey=2 \
        out_gnom=gnom_crab.dat
```
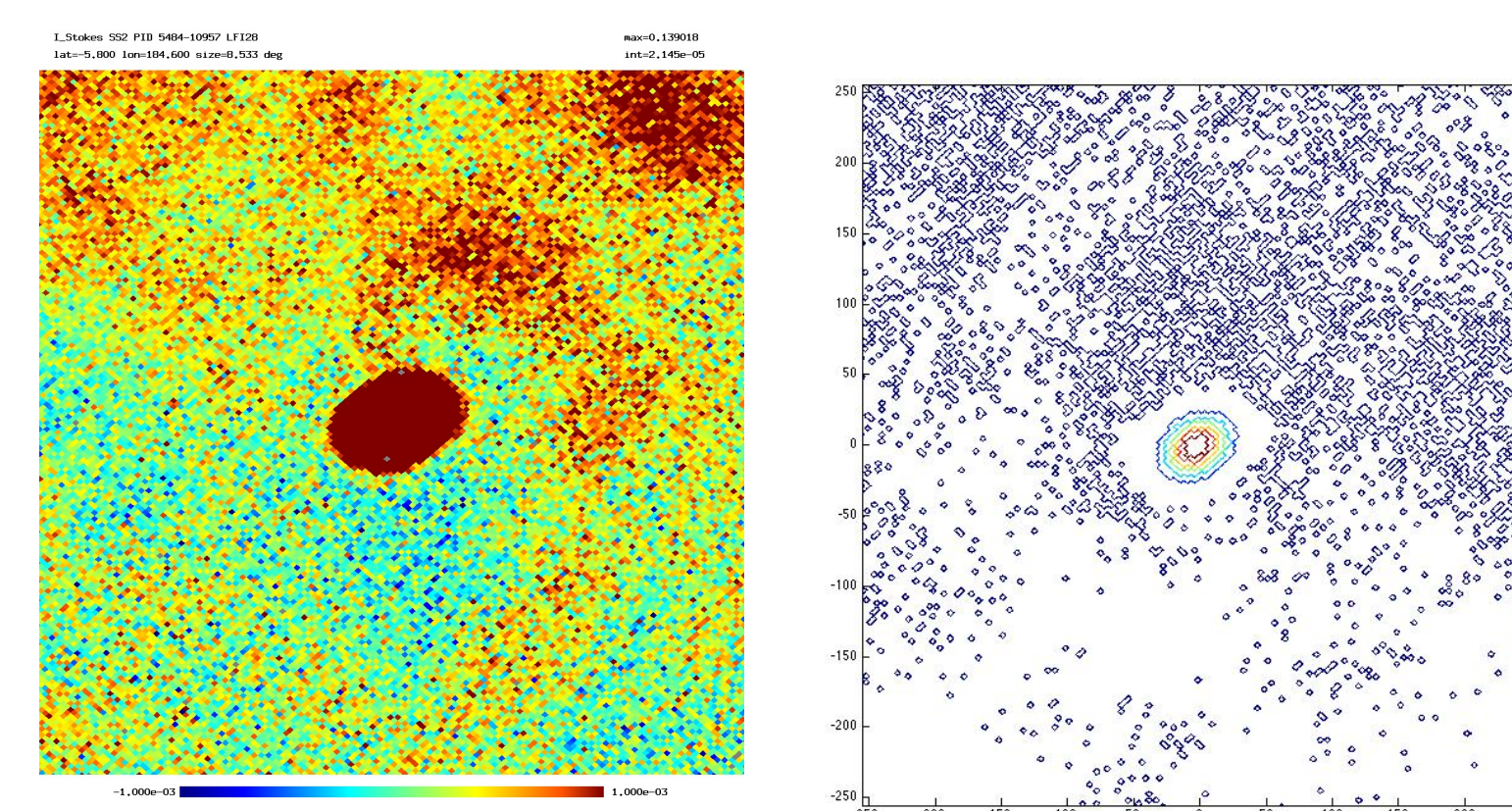

The image on the left shows the TGA output from plot_4dmap. The image data (gnom_crab.dat) was read into Matlab to produce a contour plot on the right.