First attempt at combining 2007 and 2008 data for L1527

May 14, 2009

Giles

Results from 2007:

As described in previous memos, 35 good data files were collected on L1527 during Nov. 2007. The tau was around 0.05. Nominally, we achieved a handful of 3-sigma detections, but the χ_r^2 was of order 1.6 (for Q and U) so we were not confident enough to publish these data without further analysis.

Processing the data from 2008:

In Sept. 2008, we collected 50 good data files in good-to-incredible tau. Unfortunately, there was intermittent noise affecting these data, complicating the analysis. It seems that this was correlated bolometer noise, as it often affects entire rows. The tau gets better over the course of the run, dropping from 0.05 to 0.03, but at the same time the bolometer noise gets steadily worse. This noise is only occasionally apparent in the "I" maps, but is evident in many or even most of the *sharpinteg* Q and U maps.

I introduced a new feature into *sharpinteg* which allows one to set a threshold for the allowable Q/U errors. For any polarimetry pixel where the outputted Q (or U) error exceeds this user-defined noise threshold, *sharpinteg* will mask the output so that no polarization data are outputted. This allows one to remove most of the correlated noise before it gets into the Q and U values calculated by *sharpcombine*. But the result is that for many files the majority of the pixels are flagged as unusable for polarimetry. Initially, I set this threshold to 200 millionths (about twice the usual Q/U noise level).

I used the RGM file that Tristan posted (but see below) and I used my usual favorite flags (note my new sharpinteg flag):

```
sharpinteg: -f 1 -em -w -sil -m 200

sharpcombine: -hwp 91 -l 51 51 -sm 2 -ma 5 -ps 9.5 -pm 12.0

-bg 10 0 -ip 0.0034 0.00017 0.0036 0.0
```

I also developed pointing corrections, and used the posted smoothed tau.

After masking the noisiest pixels as described above I examined all of the *sharpinteg* Q/U maps by eye again. For a minority of the files I could still see evidence of correlated noise. For these files (only) I developed my own "custom RGM files" designed to exclude obviously noisy rows or pixels. I wrote a version of the *dointeg* macro that automatically knows which RGMs to apply to which files.

I also explored the effect that varying the threshold had on the Stokes χ_r^2, which I calculated by applying *chi2* to three equal-weight bins. (In this memo, when reporting χ_r^2 results, I average map-wide results for Q and U since the two Stokes parameters give similar χ_r^2 .) By reducing the noise threshold from 200 to 150, I was able to reduce the Stokes χ_r^2 by 16%, which can be thought of as an improvement in signal-to-noise of 8%. The *sharpcombine* output Q/U errors at the peak go up by 6%, however, so the improvement in the overall signal-to-noise may be modest. However, I decided to go with this more restrictive threshold.

Dependence of reduced chi-squared on time scale:

The results for the map-wide Q-U average χ_r^2 were as follows:

2007 (three bins as described in earlier memos): Stokes χ_r^2 = 1.57

2008 (three bins as described above; 150 threshhold): Stokes χ_r^2 = 1.47

Thus the level of systematic error internal to these runs can be estimated as Stokes χ_r^2 = 1.52, which is the average of the above two values.

2007+2008 (six bins total): Stokes χ_r^2 = 1.73

Most (52 parts out of 73) of the systematic error seems to be internal to the runs, not due to differences between the runs. Accordingly I treat the systematic error as random excess noise and simply inflate the nominal errors by the square root of the Stokes χ_r^2 .

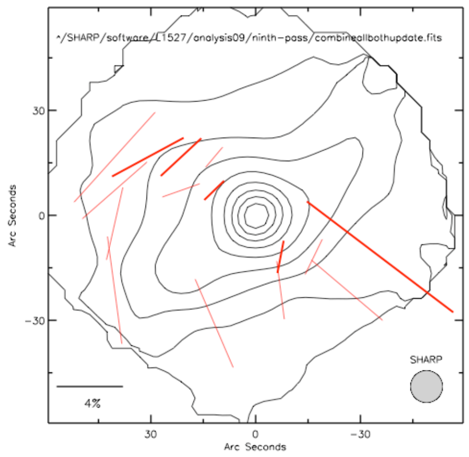
The appearance of the Stokes χ_r^2 maps outputted by *chi2* is fairly random. In particular the peaks in χ_r^2 are not clearly correlated with flux levels.

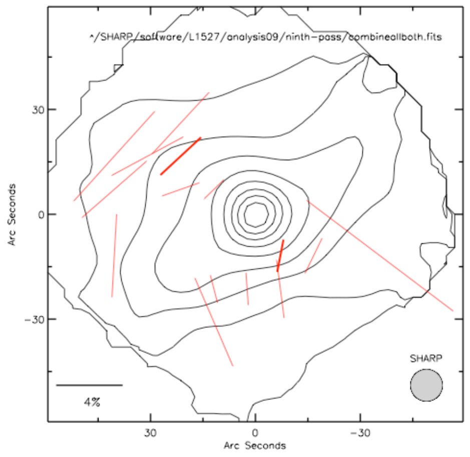
Methods for error-inflation and results:

I use two different methods for inflating the nominal errors. The first method is the *update* method, in which the *chi2* program is asked to alter the errors in the *sharpcombine* output file so as to inflate them in a point-by-point fashion (-update flag). The second method is called the *map-wide inflation factor* method. In this case we simply modify the thresholds used when invoking *polsharp5*, so as to account for the extra errors. For example, for the 2007+2008 map, the 3-sigma threshold is set to $(3)(1.73)^{0.5}$ = 3.95-sigma in the *polsharp5* invocation.

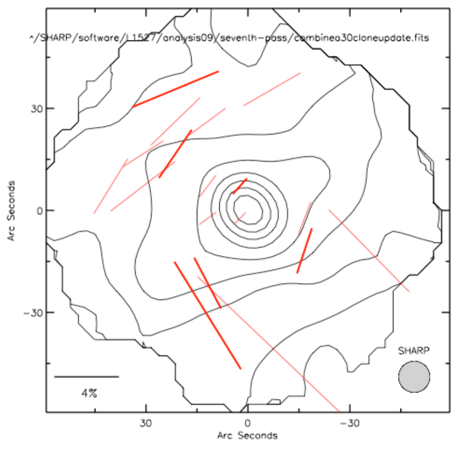
The *update* method is inaccurate when the bin count is low (6 bins is marginal) but the *map-wide inflation factor* method ignores the fact that some parts of the map could have higher errors, depending on the cause of the systematic error that is affecting us. But since the two methods suffer from different defects, for cases where they give similar results we may be on safe ground.

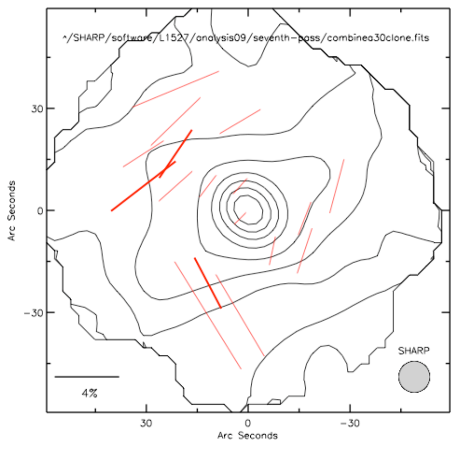
Below I show the results for the combined (2007+2008) data set. On the left is the *update* method and on the right is the map-wide inflation method. Thick bars are 3-sigma, thin bars are 2-sigma.

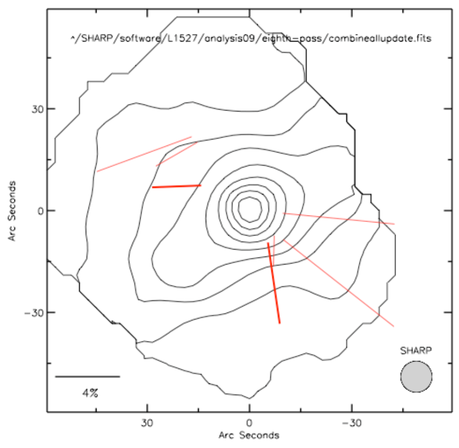

^/SHARP/software/L1527/analysis09/ninth-pass/combineallbothupdate.fits
30
0
-30
Arc Seconds
4%
SHARP
30
0
-30
Arc Seconds

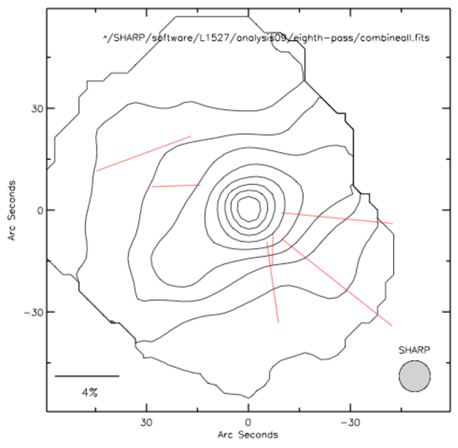

^/SHARP/software/L1527/analysis09/ninth-pass/combineallboth.fits
30
0
-30
Arc Seconds
4%
SHARP
30
0
-30
Arc Seconds

On the next page I show the same analysis applied individually to the two separate runs. 2007 is on top and 2008 on the bottom. Note that the use of the *update* method is quite inaccurate when there are only 3 bins (as is the case for these run-specific maps), but I show the maps for comparison purposes.


~/SHARP/software/L1527/analysis09/seventh-pass/combinea30cloneupdate.fits
30
0
−30
Arc Seconds
30
0
−30
Arc Seconds
SHARP
4%


~/SHARP/software/L1527/analysis09/seventh-pass/combinea30clone.fits
30
0
−30
Arc Seconds
30
0
−30
Arc Seconds
SHARP
4%


~/SHARP/software/L1527/analysis09/eighth-pass/combineallupdate.fits
30
0
−30
Arc Seconds
30
0
−30
Arc Seconds
SHARP
4%


~/SHARP/software/L1527/analysis09/eighth-pass/combineall.fits
30
0
−30
Arc Seconds
30
0
−30
Arc Seconds
SHARP
4%