# On-the-fly data processing with Scanamorphos: application to ArTéMiS

software initially developed to make maps from *Herschel* PACS & SPIRE scans
now tailored to ground-based/balloon-borne instruments (ArTéMiS, NIKA2, PILOT)

main task: subtraction of the low-frequency noise
implemented principle: maximal use of the redundancy in the data (no filtering)

low-frequency noise: both **correlated drifts** and **flicker noise**

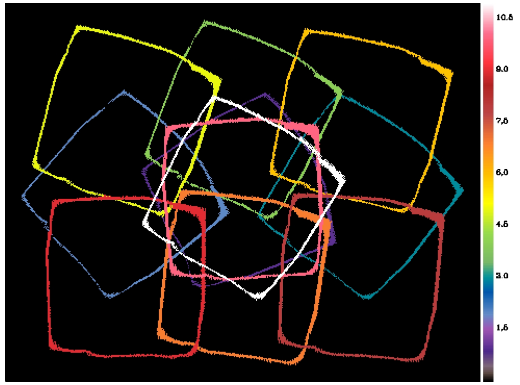
thermal fluctuations for *Herschel*

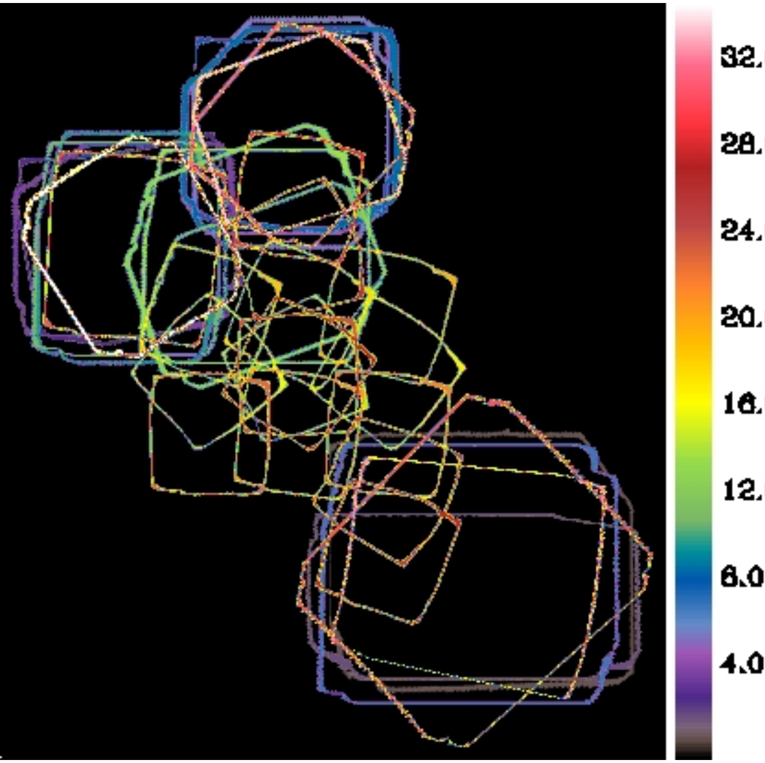
**mainly the atmosphere for ground-based instruments**

algorithm for *Herschel* described in [Roussel 2013, PASP 125, 1126]
updates and user guide for ArTémiS: [https://arxiv.org/abs/1803.04264]

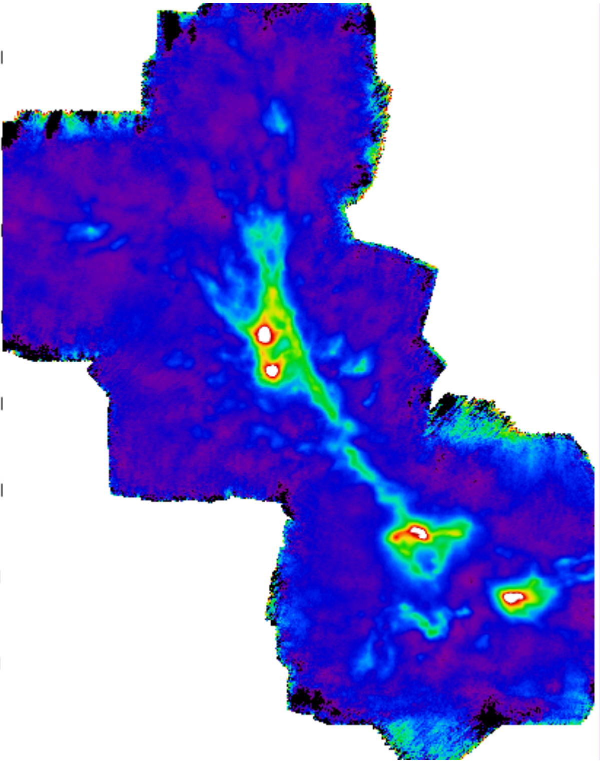
Scan geometry is crucial to obtain optimal results:

– adjust the scan length to the angular scales
on which recovery of extended emission is needed

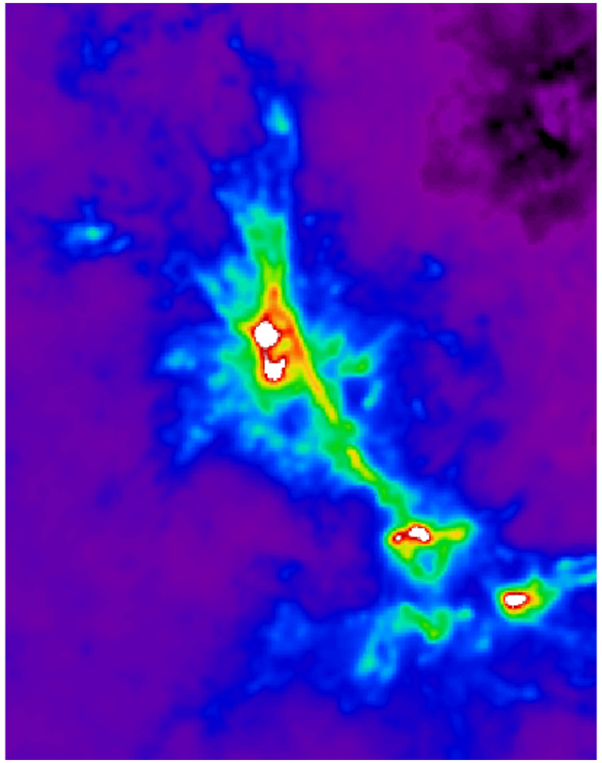




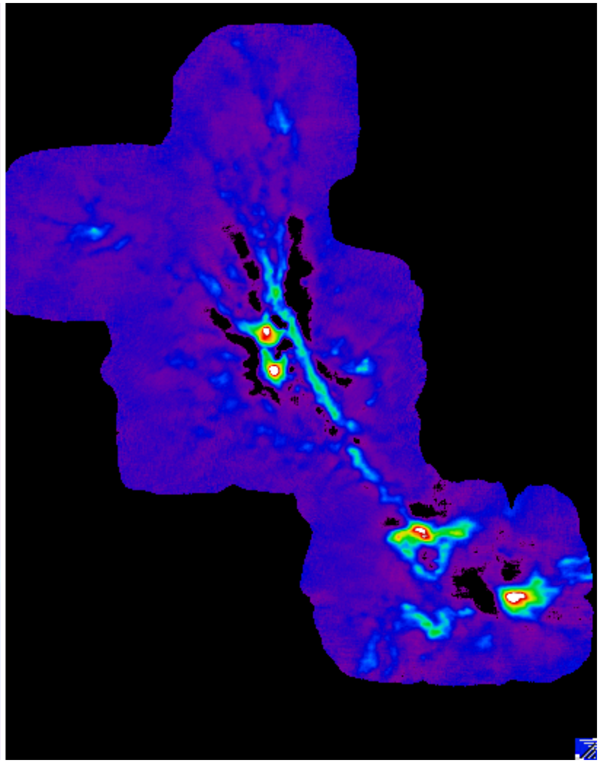
scan outlines
for N6334 350 $\mu$m mosaic



ArTéMiS (Scanamorphos)



SPIRE



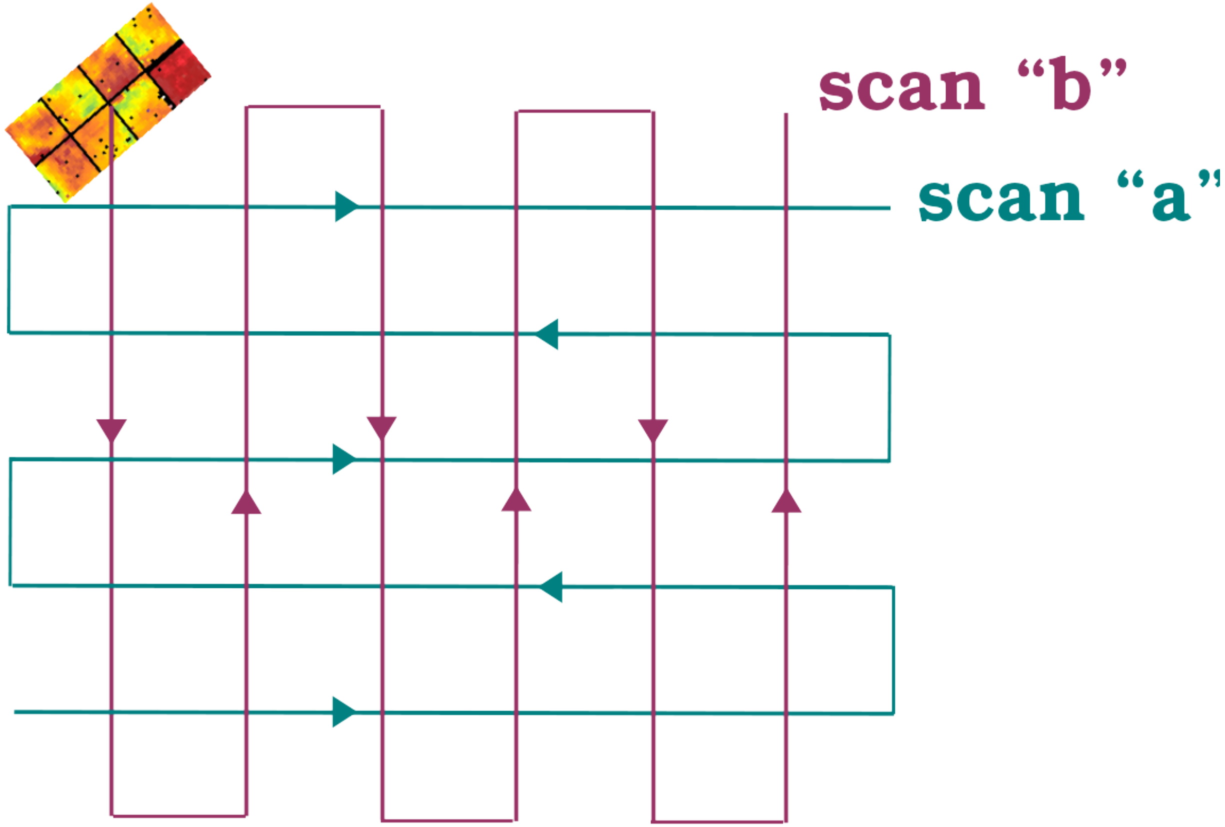
ArTéMiS (pipeline default)

Scan geometry is crucial to obtain optimal results:

– adjust the scan length to the angular scales on which recovery of extended emission is needed

– cover each region of interest with 2 well distinct scan directions (ideally orthogonal scans)

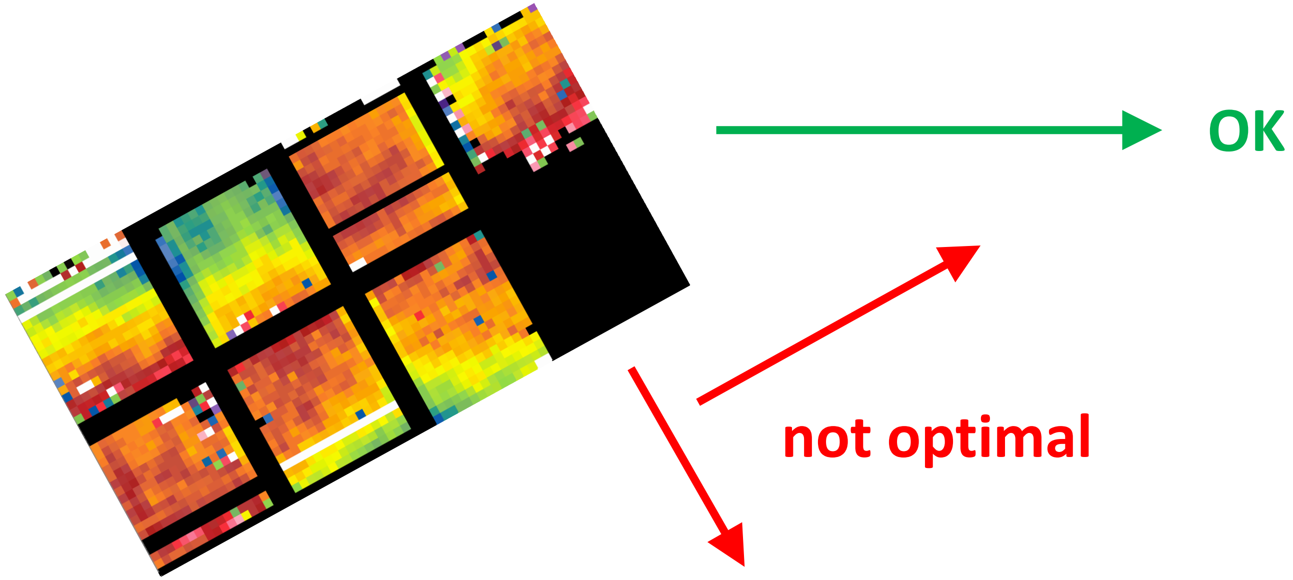
otherwise: not enough information to disentangle drifts from signal



Scan geometry is crucial to obtain optimal results:

– adjust the scan length to the angular scales on which recovery of extended emission is needed
– cover each region of interest with 2 well distinct scan directions (ideally orthogonal scans)

– avoid scanning parallel to the array axes



OK

not optimal

Drifts are additive $\Rightarrow$ all multiplicative effects (flatfield, opacity correction) must be corrected beforehand.

recorded signal R = time-invariant sky emission S
+ atmosphere + instrumental drifts D (low-f noise)
+ white noise + glitches HF (high-f noise)

$$\mathbf{R(t, b) = S(p) + D_{aver}(t) + D_{indiv}(t, b) + HF(t, b)}$$

variables: time t, bolometer b, sky pixel p

definition of a **stability length $l_s$**

within $l_s$ , S is considered uniform and D stable (rejection of compact sources / glitches)

chosen to contain ~ 7 samples per crossing for simple statistics

for ArTéMiS: on the order of 0.5 FWHM (depends on scan speed and sampling rate)

iterative process to subtract the drifts
exploitation of all the available redundancy ⇒ { large memory requirement
$t_{proc} \sim 2 \times t_{obs}$ (on-target)

- first step: **baseline subtraction** (linear fits to signal on whole scan legs)
  ⇒ removal of drifts and sky gradients on scales larger than scan legs
  uses a **fully-automatic source mask** if the `/galactic` option is set

- second step: subtraction of the average drift on small timescales

  $$\boldsymbol{\Delta(t_1, t_2) = R(t_1, b_i) - R(t_2, b_j)}$$
  $$= S(p) - S(p) + D_{aver}(t_1) - D_{aver}(t_2) + (D_{indiv}+HF)(t_1, b_i) - (D_{indiv}+HF)(t_2, b_j)$$

  **coaddition over $(p, b_i, b_j)$** → $\boldsymbol{D_{aver}(t_1) - D_{aver}(t_2)}$ + mean of uncorrelated terms

- third step: subtraction of the **individual drifts** on successively smaller timescales
  (timescale decreased by a factor 3 each time)

usage for ArTéMiS:

1) within the pipeline (APIS): apply flux calibration and opacity correction
2) format the data for input to `scanam_artemis` (interface provided with the code)
3) process the data
4) optionally make maps within the pipeline

output:
- maps assembled in a cube (signal, error, weight, subtracted drifts)
- processed data reinjected into pipeline structures

`scanam_artemis` available on: [www2.iap.fr/users/roussel/artemis](www2.iap.fr/users/roussel/artemis)
as well as the user guide with illustrations (N6334 mosaic)

THANKS TO THE ORGANIZERS !

TRAVEL GRANT FROM RADIONET

# Scanamorphos for ArTéMiS:
# step by step on an example

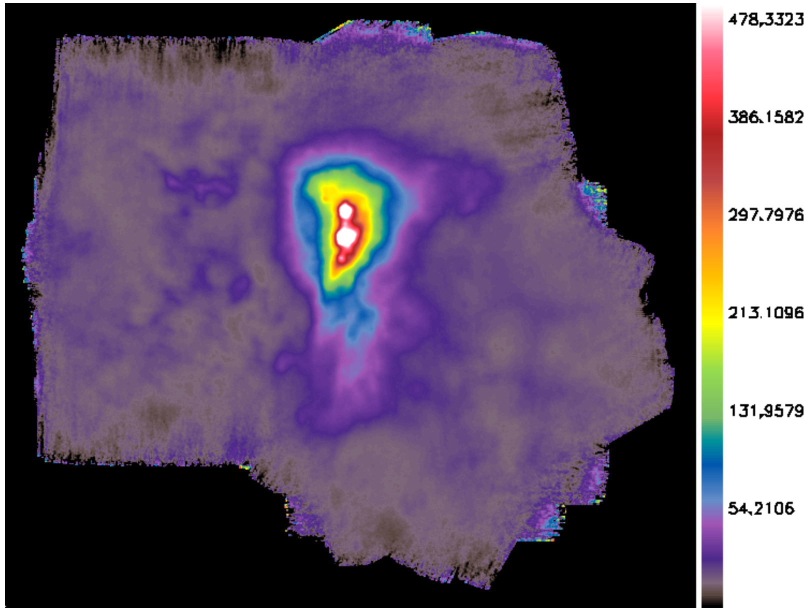
to speed up the processing:

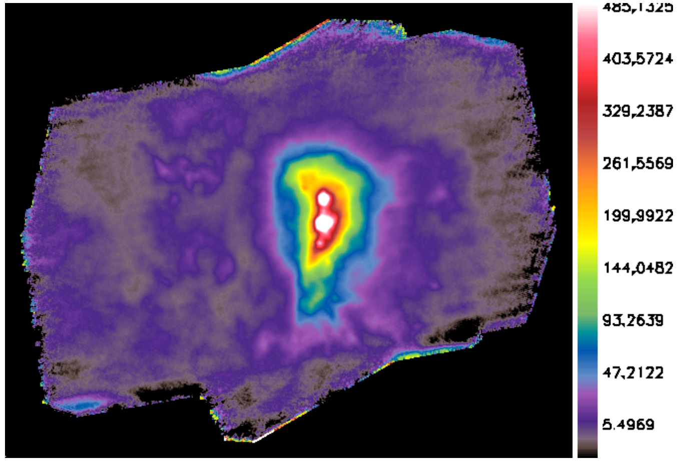
demo on a set of 2 central scans on a very bright target: Sgr B2

for demo purposes only !

data already calibrated (cf tutorials by F. Schuller and P. André)

map
composed
of 7 scans





map you will
obtain
(hopefully !)

scans 23904 and 32910 taken in 2016

## interface pipeline → Scanamorphos and formatting of input data

make sure that obs1_artemis_config.pro contains the relevant info:

```
   project_name = ’E-097.C-0184-2016’
   calibration_table = ’calibration_table_350_2016’
```

```
IDL > dir_out = …
```

directory where input structures, temporary files and output cube will be written

```
IDL > list = [23904, 32910]
```

```
IDL > format_input_scanam_artemis, dir_out=dir_out, list_scannum=list
```

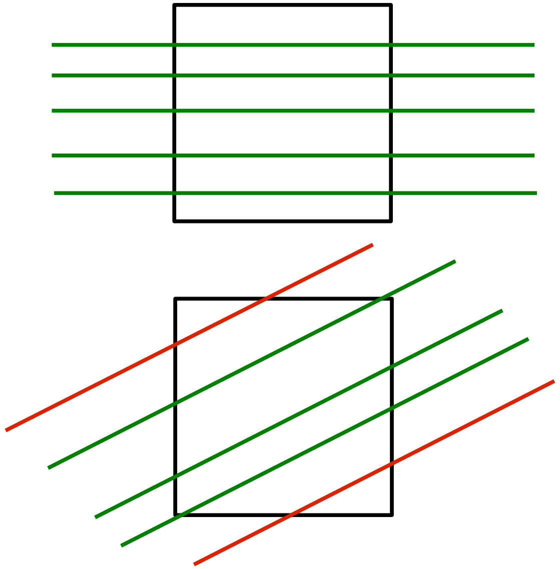
( `array=350` not necessary, since this is the default)

→ creation of the `scanlist_artemis` ascii file and the input structures in dir_out

# interactive processing

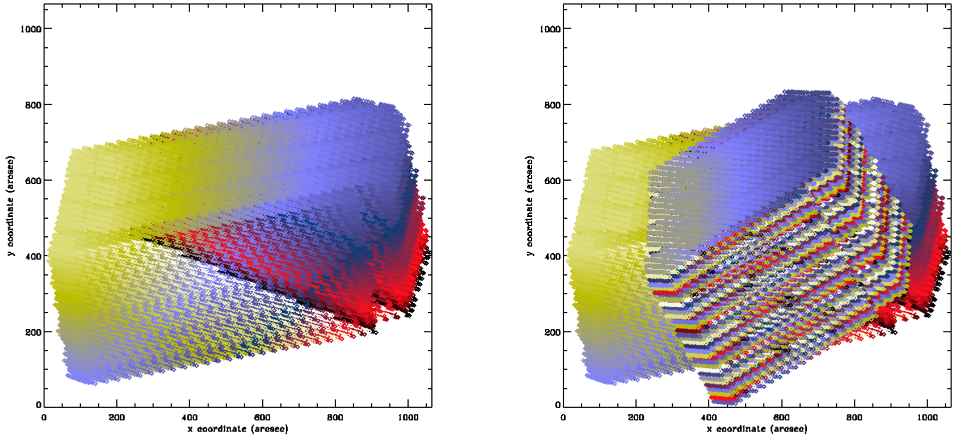
```
IDL > scanam_artemis, /galactic, /visu, /vis_traject, dir_scanlist=dir_out
```

1) data ingestion and determination of astrometric frame

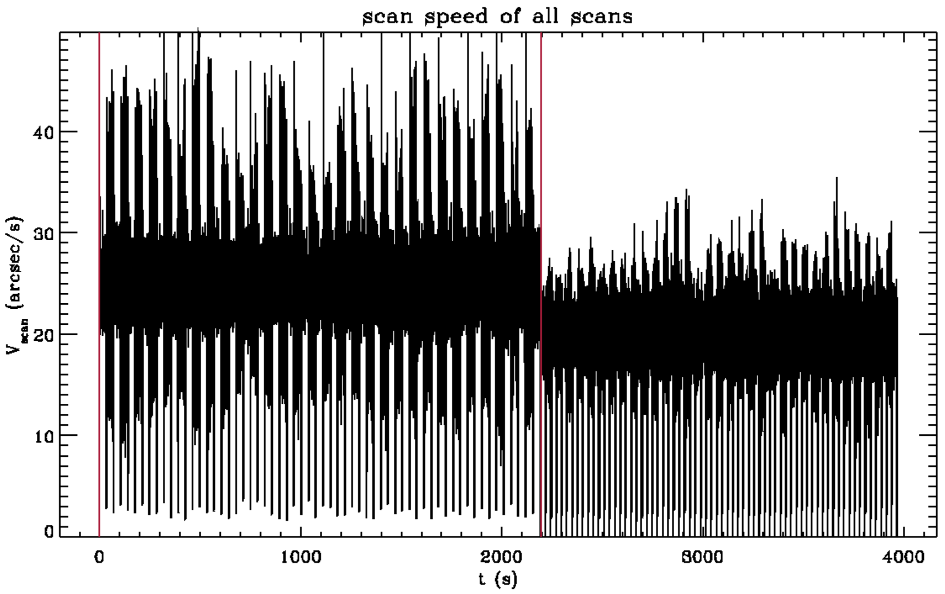


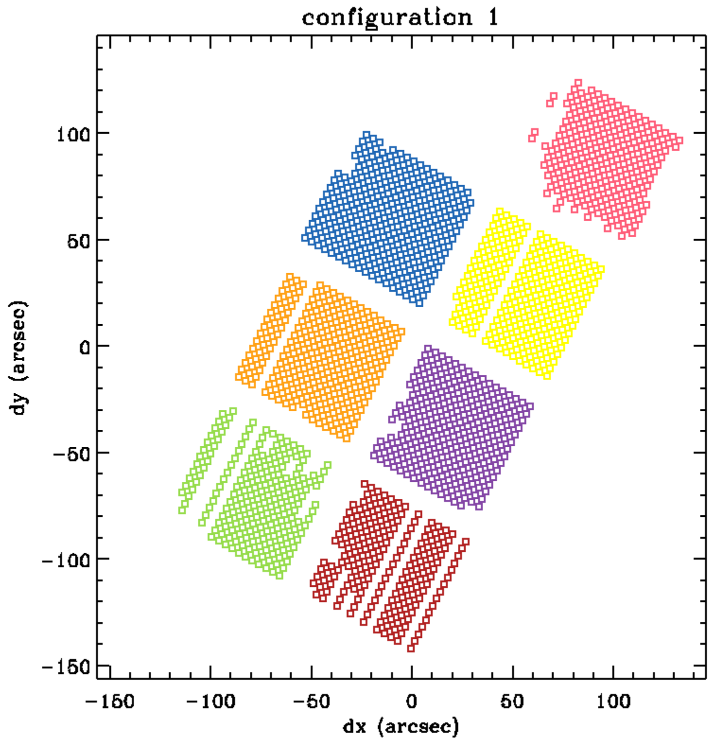
if possible: for the processing, use the orientation that maximizes the number of samples within a pixel of size $l_s$ for most scan legs

`/vis_traject` option: visualization of OTF array trajectories

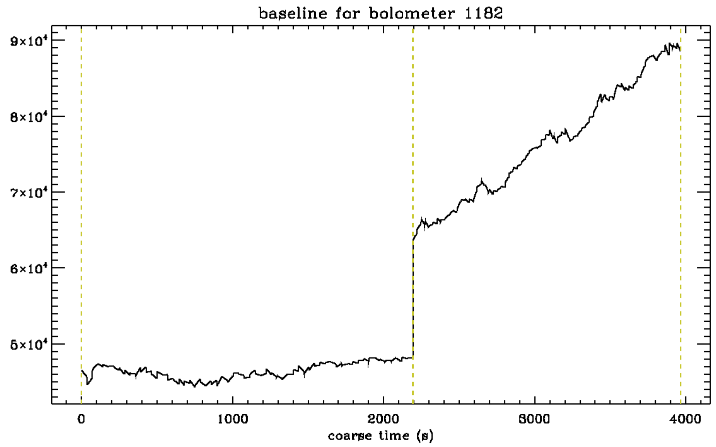


# information about scan speed and array geometry:



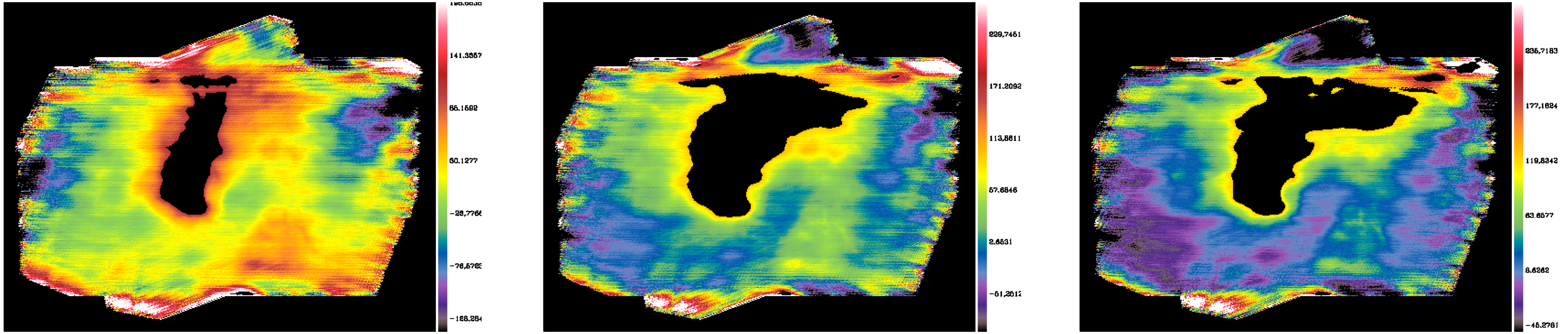


2) computation of high-frequency noise (white noise)
for weighting and noise thresholds
(updated several times during the processing)

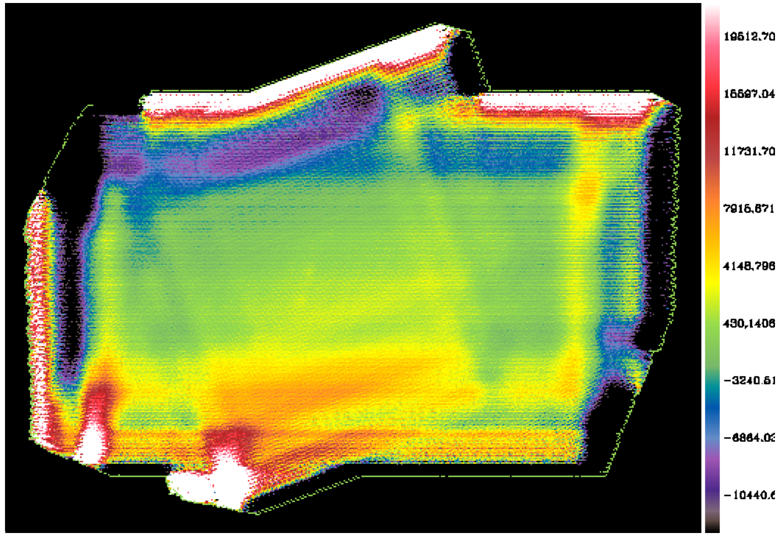


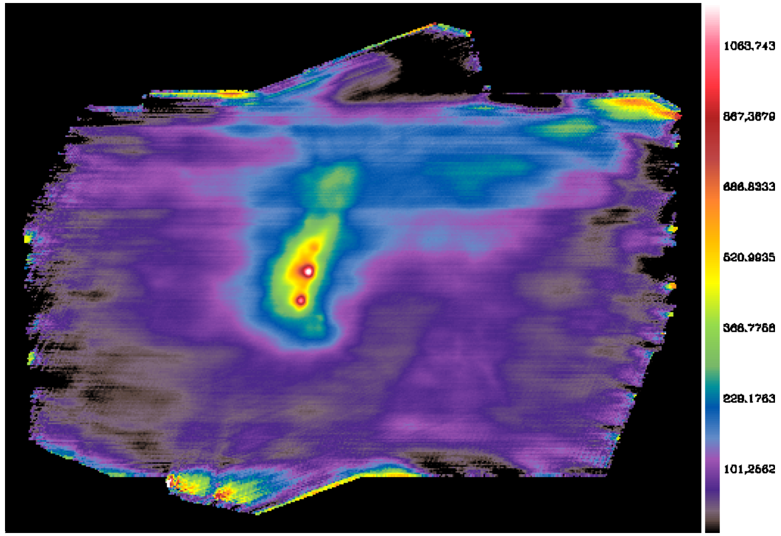
3) initial baseline subtraction: several iterations
linear fits to the average signal
first on whole scans, then on segments of 4 scan legs,
then on individual scan legs

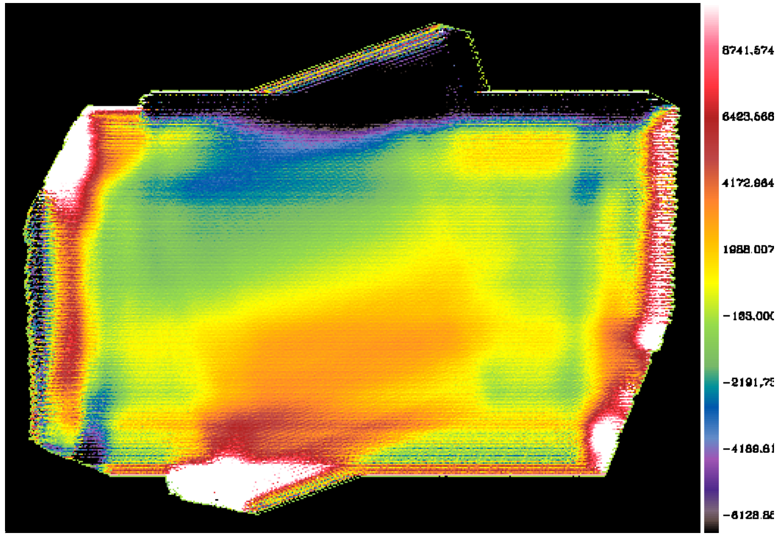
meanwhile: construction of an automatic and iterative source mask
(if the `/galactic` option is set ; do not use for diffuse sources !)



# before and after the initial baseline subtraction:

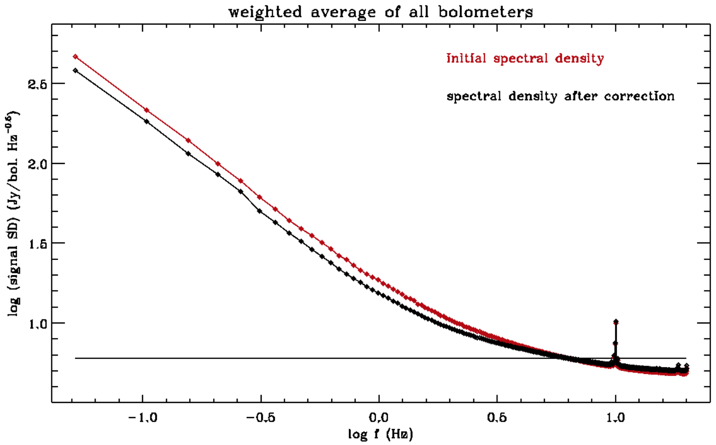






← what's been subtracted (with offsets)

power spectral densities before and after →

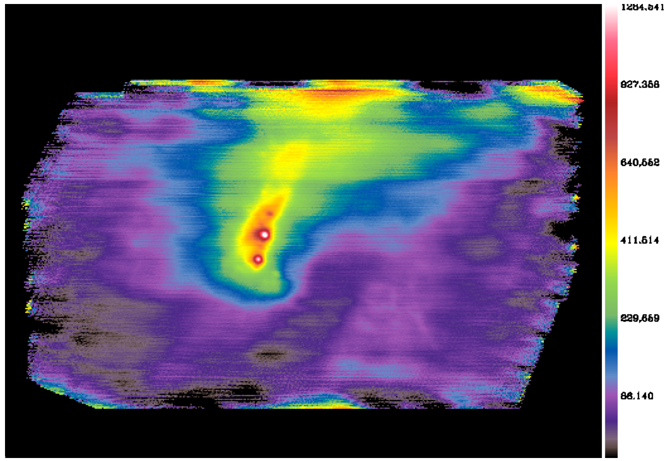


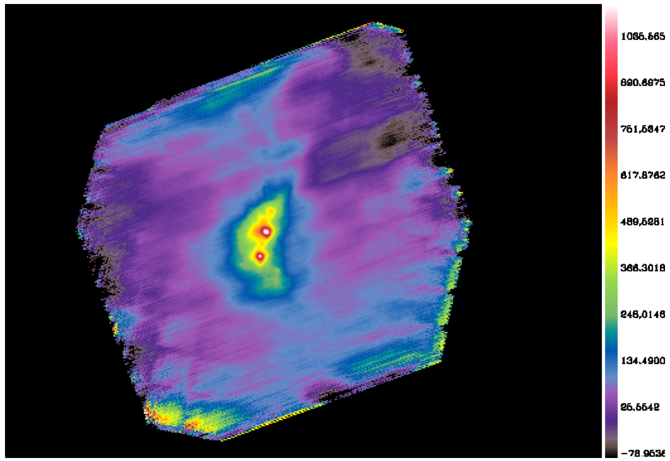
to project and display maps of individual scans at any point during the processing:

```
IDL > do_map_scans_nearest_artemis, map_scans, weight_scans, $
      file_scalars=file_scalars, ind_scans=ind_scans, ind_subscans=ind_subscans, $
      maxnoise=maxnoise
```

(copy/paste this command from the header of `do_map_scans_nearest_artemis.pro`)

```
IDL > for i = 0, nscans - 1 do disp_ima, win=20+i, $
      map_scans(*,*,i), weightmap=weight_scans(*,*,i), $
      [min_map= ..., max_map= ..., title='scan '+chain(i)]
```





4) subtraction of the average drift on small scales (smaller than scan legs): will take a while....

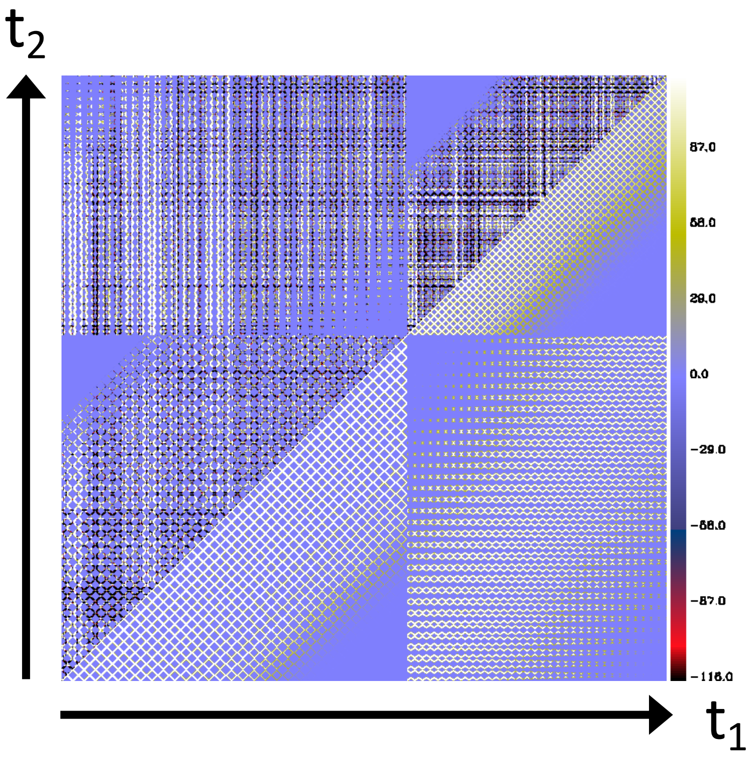
$$\Delta(t_1, t_2) = R(t_1, b_i) - R(t_2, b_j)$$

$$= S(p) – S(p) + D_{aver}(t_1) - D_{aver}(t_2) + (D_{indiv}+HF)(t_1, b_i) – (D_{indiv}+HF)(t_2, b_j)$$

coaddition over $(p, b_i, b_j)$ → $D_{aver}(t_1) - D_{aver}(t_2)$ + weighted mean of uncorrelated terms

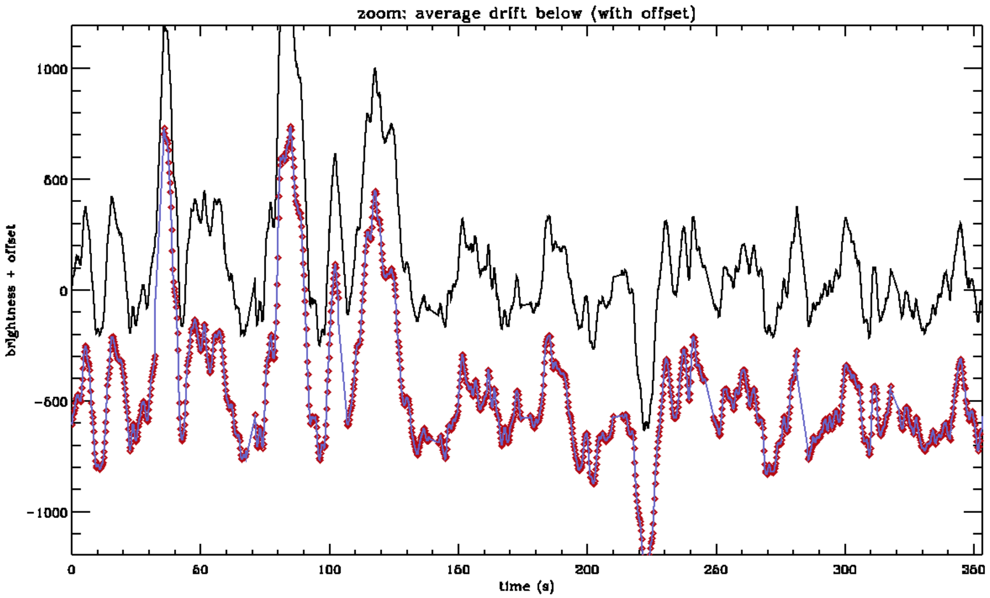
incremental population of $D_{aver}(t_1) - D_{aver}(t_2)$ matrix

(below diagonal: associated weights)



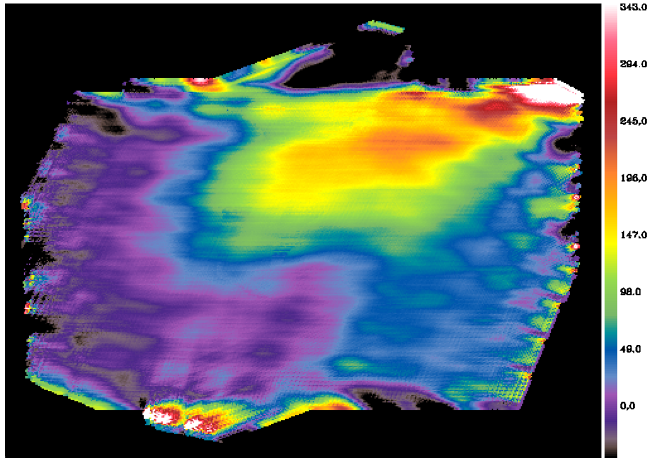
iterative scanning of $D_{aver}(t_1)$ - $D_{aver}(t_2)$ matrix

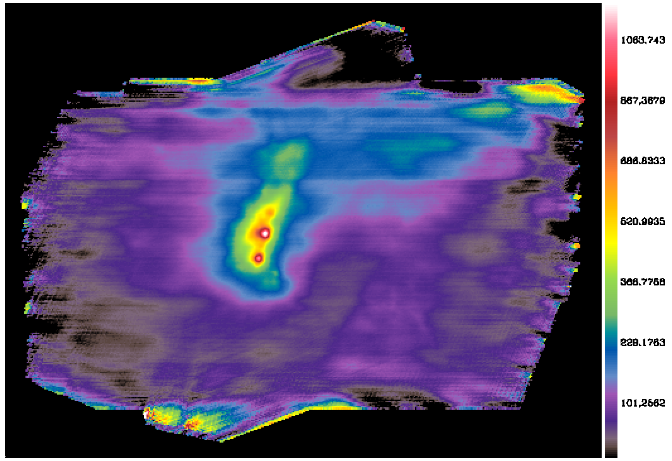
→ $D_{aver}(t)$ time series



projection of $D_{aver}(t)$

compared with map at previous step

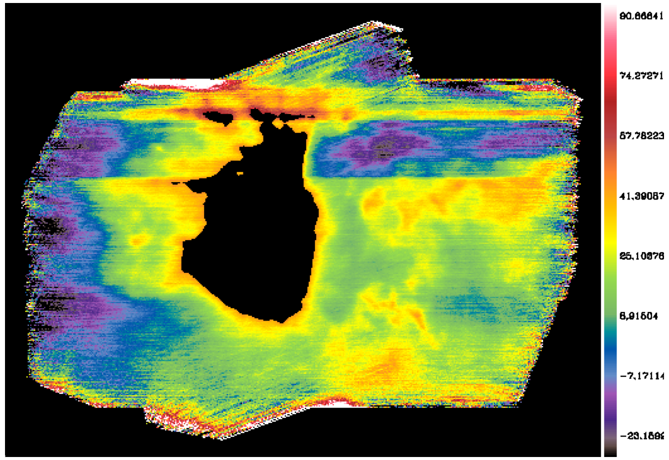


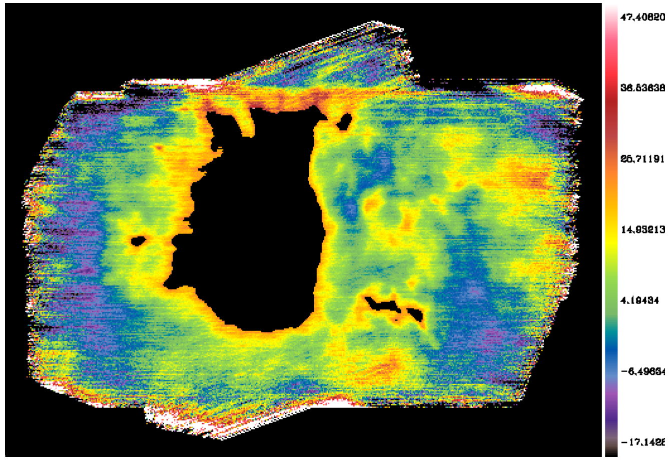


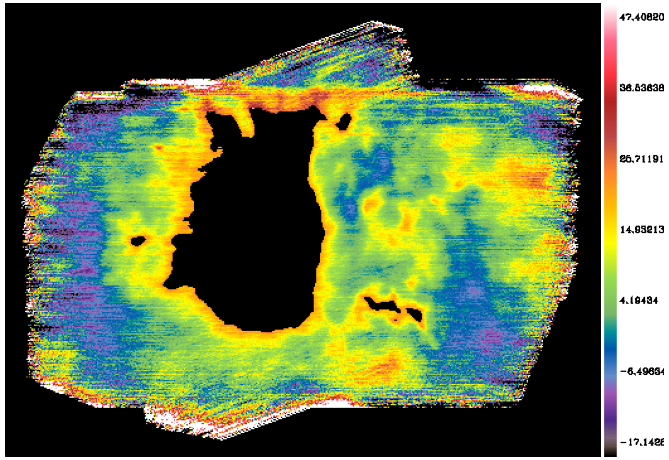
solution of $D_{aver}(t_1) - D_{aver}(t_2)$ matrix not unique
(true drift + spurious component with the same periodicity as the scans)

$\Rightarrow$ baseline subtraction repeated to remove the spurious component
fits on individual scan legs refined: for each subarray separately

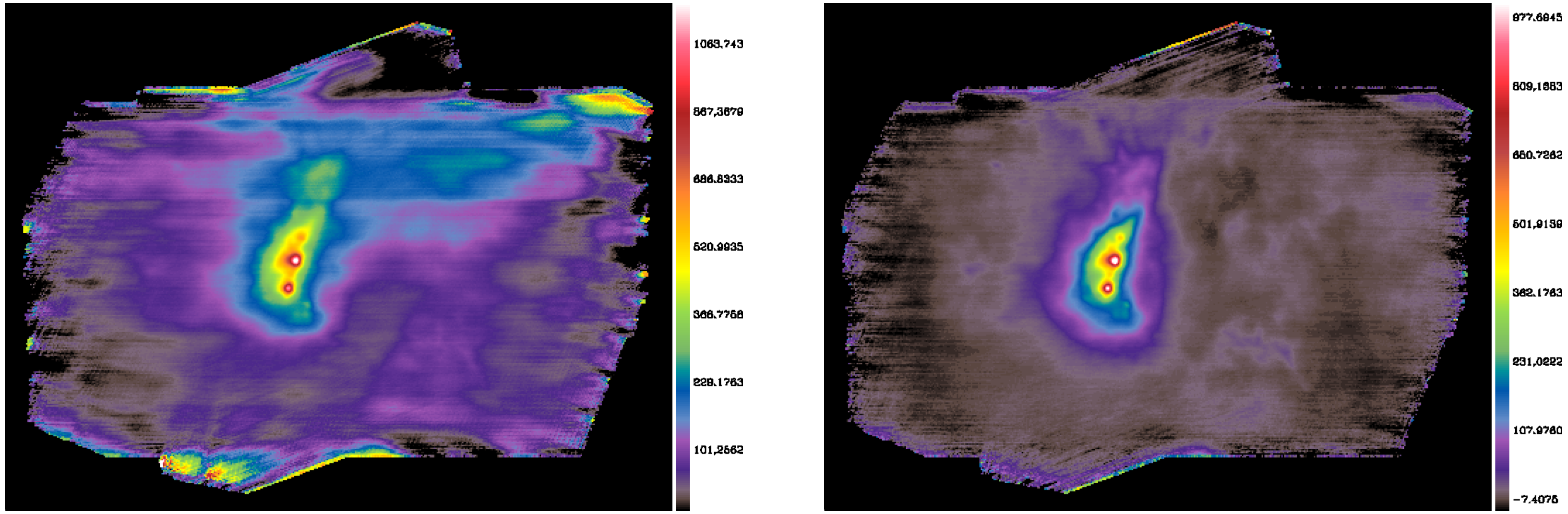
iterative source mask:

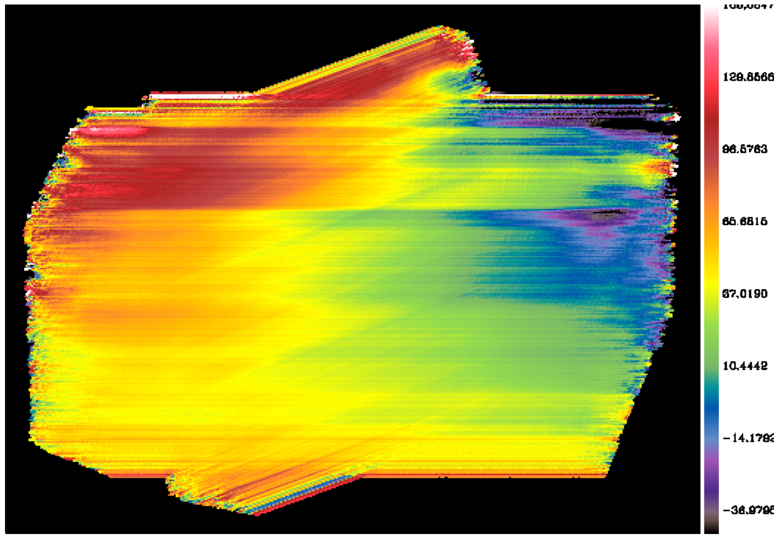






# before and after average drift + second baseline subtraction:



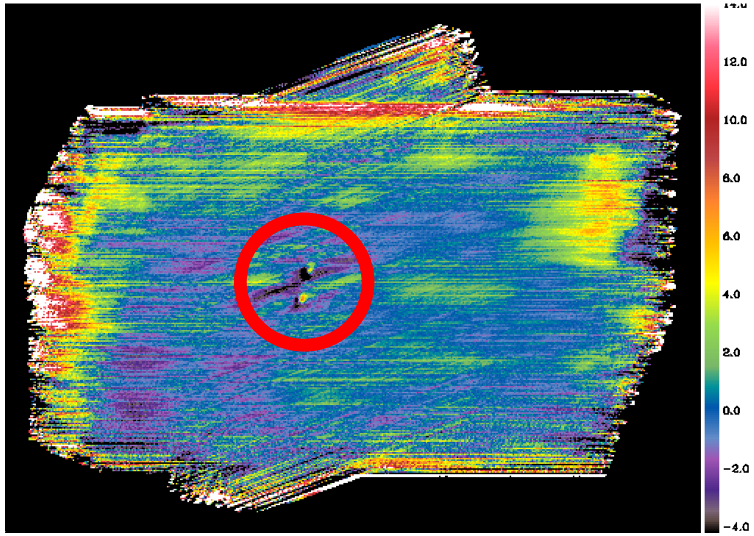


← subtracted baselines (with offsets)

5) subtraction of the individual drifts (flicker noise) on timescales of ~ 1/4 the minimum scan leg duration, followed by baseline subtraction again

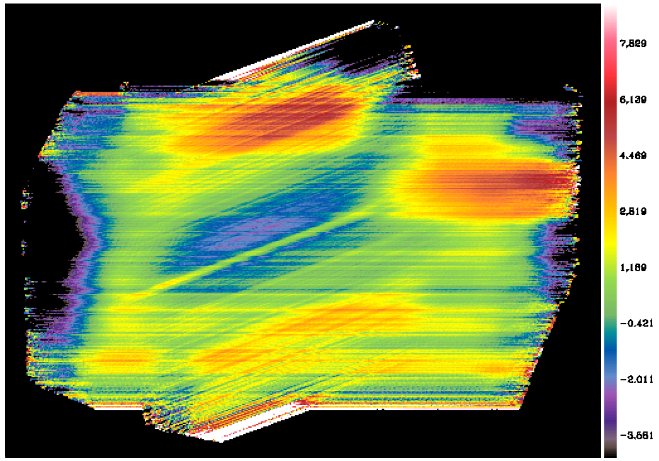
iterated on successively smaller timescales (decreased by a factor 3 each time) until reaching the stability length crossing time $t_c = l_s / v_{scan}$

projected drifts on 5.4 s timescale:

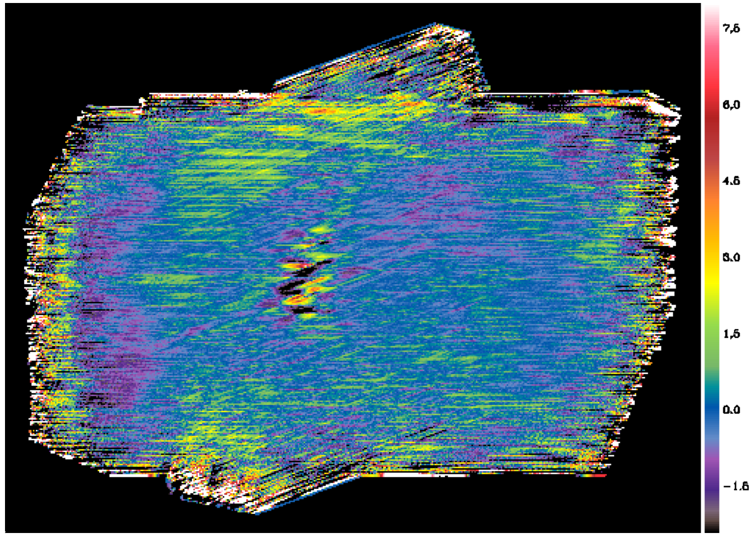


← typical artefacts caused by small pointing errors

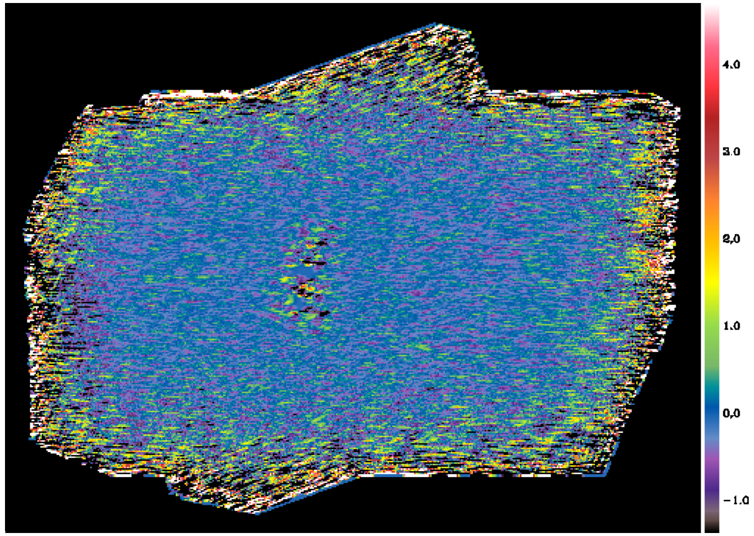
iterated baselines:



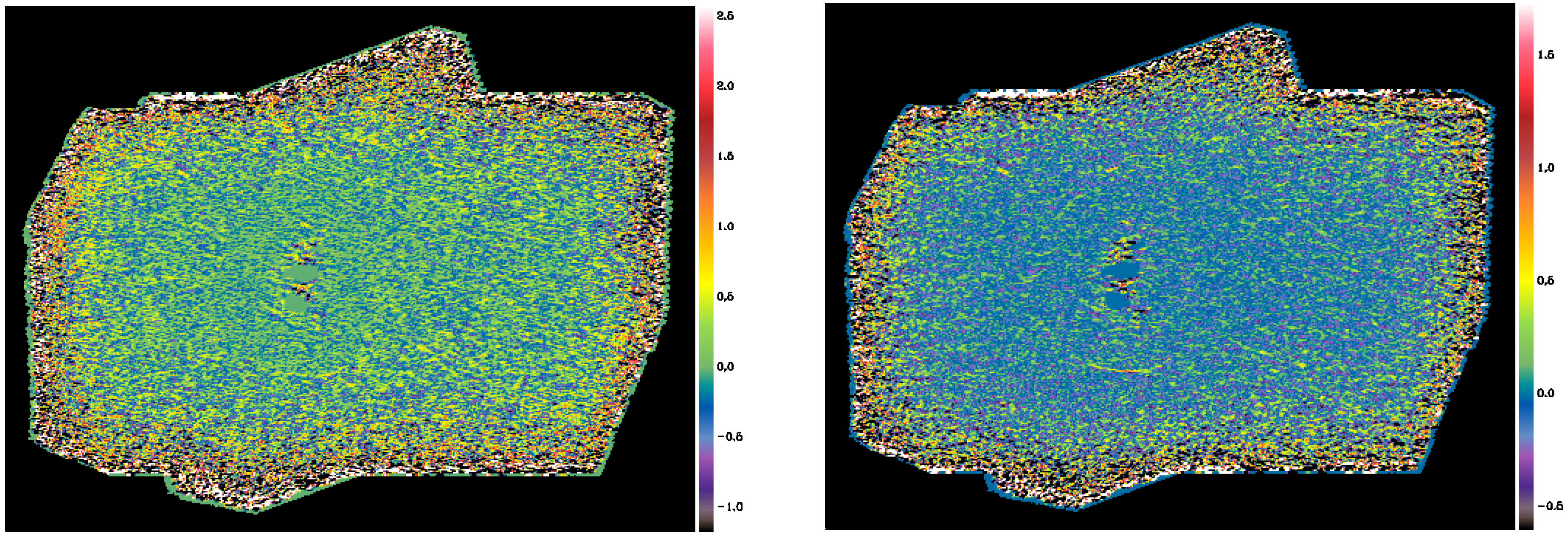
projected drifts on 1.8 s timescale:



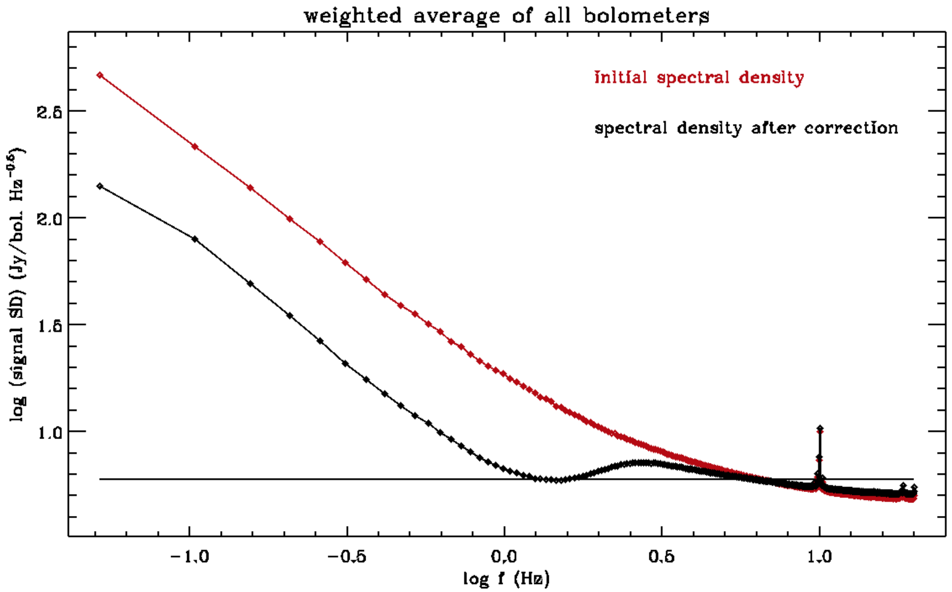
on 0.6 s timescale:

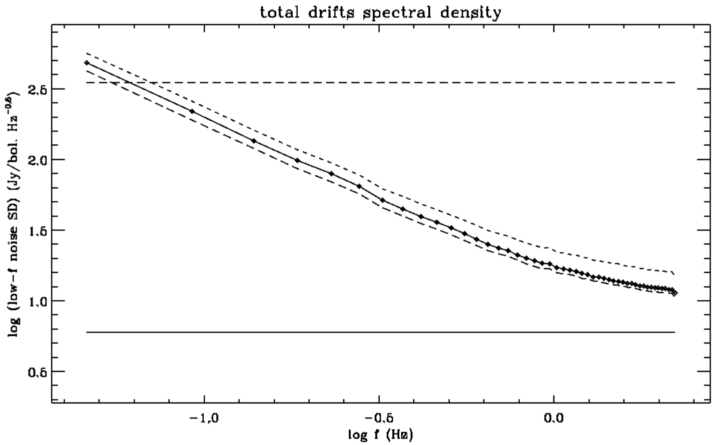


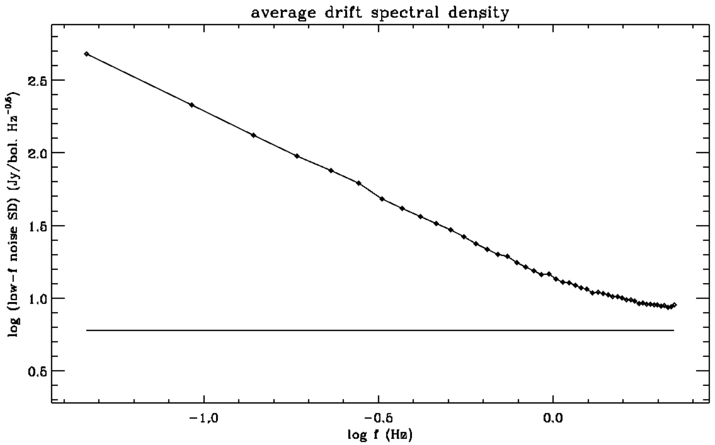
on 0.2 s timescale:

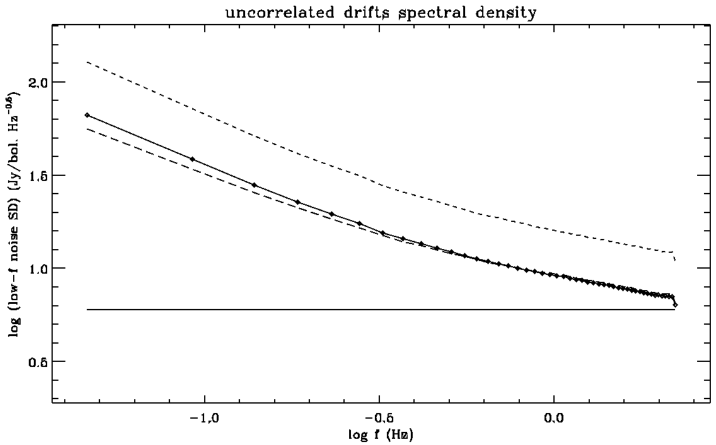


power spectral densities:

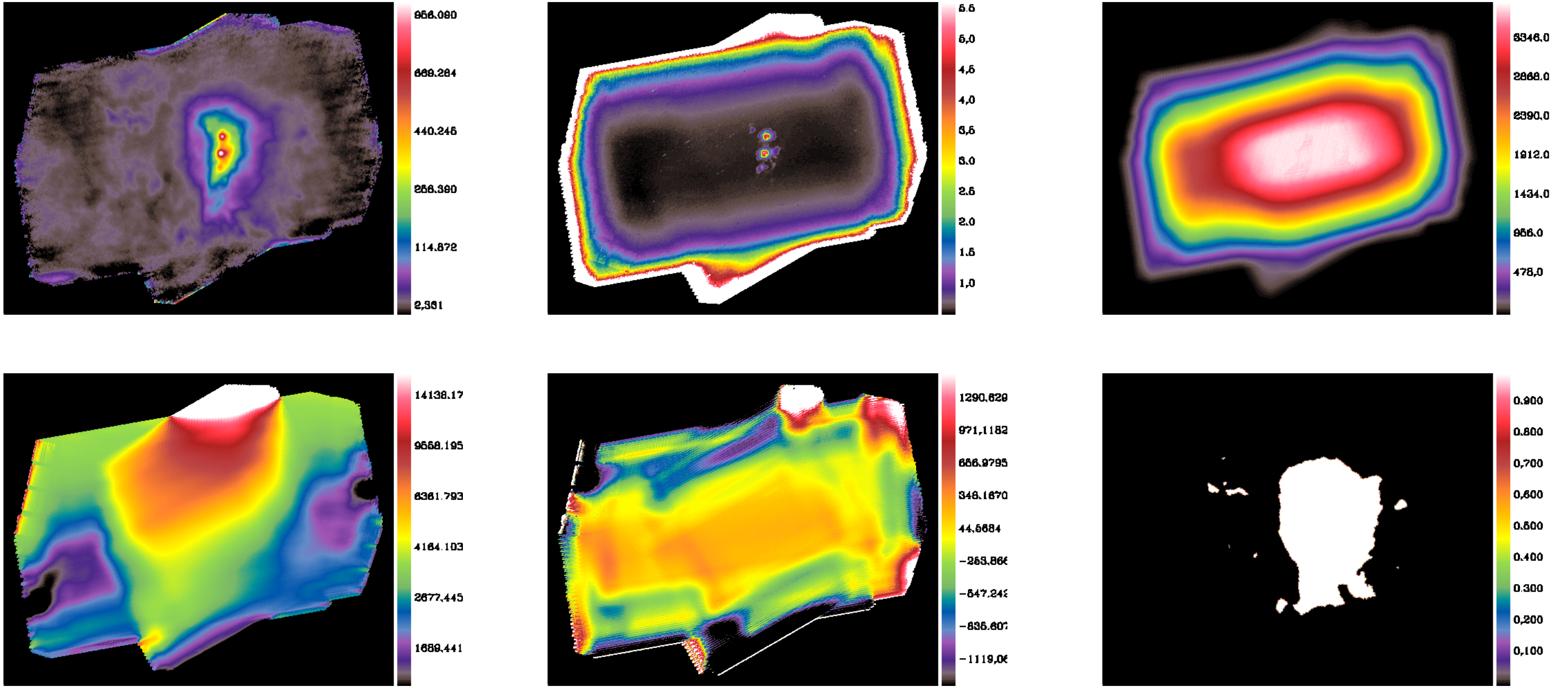








output maps (assembled in a fits cube):


956.090
669.284
440.245
256.390
114.872
2.381
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
3346.0
2868.0
2390.0
1912.0
1434.0
956.0
478.0
14138.17
9558.195
6361.793
4164.103
2677.445
1689.441
1290.629
971.1182
656.9795
348.1670
44.5684
-255.866
-547.242
-835.607
-1119.06
0.900
0.800
0.700
0.600
0.500
0.400
0.300
0.200
0.100