LBWG memo 11

Workplan of the overall algorithm

Neal Jackson, Leah Morabito, et al. 28.7.2018

This document expands and supersedes the discussion of the LB pipeline algorithm on the Google doc. It was prompted by the thought that we need to go through each step specifying rather precisely (i) the inputs for each step (for measurement sets, their nomenclature, format and calibration status: for tables, their nature, contents and column numbers; for files, their format; (ii) the outputs, with the same level of documentation; (iii) the system requirements (location of scripts, required libraries and Python packages); (iv) documentation about how and whether they work (at the level of previous memos, for example).

We begin with a set of subbands which have been processed through the `prefactor` pipeline (the `ndppp_prep_target` files). These have also been processed through the current LB generic pipeline to apply these solutions, up to and including the `ndppp_apply_cal` step. (Some of the following algorithm is thus already in the GP, some will require small changes, and the rest is still to be written). These subbands are at the raw average stage of 1s integration and 16 channels per subband.

We define three classes of source:

1. The **primary delay calibrator**. This is the source which has the highest degree of coherence; in practice, this means the source with the highest degree of coherence on the IBs. This is used to calculate a delay and phase which is applied to all data. All subsequent calibrations are incremental calibrations on top of this. (This should make all subsequent operations more efficient in that no very short solution intervals need to be used after this).

2. **Iteration 1 sources.** These are sources which need to be subtracted from the image in order to make maps of everything else. The main criterion for such a source is that it has significant flux density on the shortest baselines that we want to use in imaging (about 30-50km). These sources will also form the basis of the directional-dependent phase solutions in the field.

3. **Iteration 2 sources.** All other sources in the field.

The algorithm is then:

1. **BEGIN "ITERATION 0".** Interrogate the LBCS catalogue to find all candidate delay calibrators. These are all sources with 'P' for all of the stations, or all sources with the highest proportion of 'P' entries. *Scripts exist:* `download_lbcs_lotss.py` on the main branch and is in the plugins directory on the new branch. *Time required:* zero. `Leah/Frits incorporating into final pipeline, will be pushed to main branch.`

2. Extract 20-subband measurement sets for each of these sources, at 2ch/8s resolution and run the new closure phase scripts (see Memo 3). Choose the one source which has the highest closure statistic on the long baselines as the primary delay calibrator.

Scripts exist: `closure_v4.py` (currently used, calculates closure phase scatter) and `closure_v5.py` (calculates SNR rather than closure phase scatter); then standard NDPPP shift-average. *Time required:* (1-2 hours; highly parallel, few subbands, can fit on one CEP3 node). `Under control, in pipeline.`

3. Extract 200-subband measurement set for the primary delay calibrator. *Time required:* 6 hours for current pipeline. Has been done in less. **Needs investigating: is the current parallelization efficient? Does the generic pipeline run significantly slower than home-brew parallelization? Can we use killMS instead?** `Leah working on`

4. Run the EHT imager to find a model. Calculate delay solutions with separate clock and TEC – in practice, we are currently calculating phase-only solutions and then running LoSoTo to do clock/TEC separation. Calculate phase solutions by self-calibration. Calculate amplitude solutions (can use LoTSS information to help). Check these carefully. *Scripts exist but require testing.* EHT imager script `eht_imager.py` exists but has been tested on few datasets. Phase/amplitude selfcal to be done using `loop3.py`, which exists but needs testing. Should produce a parmdb. **Switch needs adding to loop3 to produce either parmdb or h5parm; switch needs adding to optionally do phase and amplitude solutions**. Quality testing for phase solutions is in principle provided by `loop3.py`. Loop needs adding to choose another delay calibrator if this step fails. *Time required:* 12 hours, mainly in the clock/TEC separation. **Can we reduce the time by optimising the LoSoTo setup?** `Joe continuing to test - consult Leah/Neal`

5. Apply all solutions to the `ndppp_prep_target` measurements sets in situ by overwriting the corrected data column with the data column with all previous parmdbs applied, plus the clock, TEC, phase and amplitude solutions. *Scripts exist:* standard NDPPP. *Time required:* less than 1 hour (Leah estimate from experience with applying calibrations). `In pipeline, needs testing with the LoSoTo-processed h5parm.`

6. **BEGIN ITERATION 1.** Select sources from LBCS and LoTSS which require subtraction. These include all sources with large flux densities on 50 km baselines (see Memo 8). These are in principle predictable without the closure script, using LoTSS fluxes only. *Scripts exist:* standard `download_cats` producing list of sources and positions. **Needs doing: determine what LoTSS flux corresponds to the cutoff criteria in Memo 8.** `Marco to continue/add comments to memo on how many sources need subtracting.`

7. Split all Iteration 1 sources, 200 subbands[1] to an averaging level of 1ch/16s (not further since we are going to do incremental delay calibration; see memo 2). *Scripts exist:* standard NDPPP. *Time required:* in current generic pipeline, about 20-30 sources and 200 subbands will take 5-6 days. This is too long: see comments above. **Needs benchmarking and consideration of other parallelization arrangements or killMS.** `Aug 2018 busy week: Leah has tested generic pipeline vs homebrew, implementing optimal scaling.`

8. Do the following steps in parallel for each source:

[1] The maximum usable subband is likely to be 230 in absolute subband numbers

- Interrogate FIRST (or LOTSS) to find the required image size. (NB the `eht_imager` contains routines for downloading FIRST images).
- Run the EHT imager to produce a model; check its quality. `See previous comments about EHT imager testing`
- Run the closure script to work out what range of baseline length have coherent signal (see memo 3). In emergency when EHT fails to converge, perform the following steps using only a limited u-v range; an initial point-source model for the delays; and a FIRST or LoTSS model for the initial run of loop 3. (Note that the `eht_imager` script contains routines for downloading FIRST images to order).
- Calculate the differential delay (TEC only) using NDPPP and producing `h5parm`.
- Run loop 3 to generate phase solutions in `h5parm` format.
- Use `wsclean` with previous h5parm to make image. Track coherent flux in image. **Can we apply h5parm on the fly or does this step need to modify the data? LKM: I'm pretty sure we have to apply first.**
- Repeat loop 3 and `wsclean` until coherent flux stops increasing. `Neal will continue testing loop 3. L1 solutions? Neal/Joe/Sean to continue thinking about treatment of partial solutions.`
- If an EHT model has been used, the position will be wrong; in this case, update the model to the correct position using centroiding from LoTSS. (Failure to do this will result in large residuals in the subsequent subtraction step).
- Check solution quality and update global `h5parm` structure (see notes below on HDF5 subroutines).

If these steps fail (bad solution quality throughout), put the source into a supplementary list to be processed in Iteration 2. *Scripts exist:* FIRST image size estimator needs writing. EHT imager exists but needs testing. NDPPP TEC calculator exists but is in experimental phase. Loop 3 exists but needs testing. `wsclean`/loop3 control loop needs writing. `h5parm` administration routines need writing (see notes below). *Time required*: should be very quick considering the embarrassingly-parallel nature and the ×256 averaging. 2-3 hours? **This step needs testing by hand on a number of sources to check that good maps can be produced automatically in this way.**

9. Global subtract all Iteration 1 sources using `sagecal` including delay. To avoid making new measurement sets, begin with the write `corrected_data` column of the `ndppp_prep_target` files and do the subtraction into the `data` column. *Scripts exist:* `sagecal` interface written by Marco and is being tested. Needs modifying (does not need to do wsclean to generate image to search for sources, as these have been mapped in iteration 1). Should examine sagecal phase solutions and check that these are very close to 1. **Need to understand exactly how sagecal solutions work in terms of amplitude/phase correction for each correlation**. `New NDPPP can do the subtraction in principle, given skymodel and phase solutions. TBD which is actually going to be used. Will need to specify a uvrange in sagecal. Marco/Leah to try a subtraction with existing models.`

10. **BEGIN ITERATION 2.** Split all remaining sources from the corrected data to 1ch/16s files. *Scripts exist:* standard NDPPP. *Time:* see comments above, likely to be ∼1 week with current software timings. `See previous comments about parallelization.`

11. Do the following steps in parallel for each source:

    - Interrogate FIRST (or LOTSS) to find the required image size. (NB the `eht_imager` contains routines for downloading FIRST images).
    - Run the EHT imager to produce a model; check its quality.
    - Find the best `h5parm` solution from Iteration 1 sources.
    - Run the closure script to work out what range of baseline length have coherent signal (see memo 3). In emergency when EHT fails to converge, perform the following steps using only a limited u-v range; an initial point-source model for the delays; and a FIRST or LoTSS model for the initial run of loop 3. (Note that the `eht_imager` script contains routines for downloading FIRST images to order).
    - Run `wsclean` over the appropriate range of baselines.
    - If the EHT imager has been used, update the map centroid using LoTSS.

Notes on the h5parm routines

In Iterations 1 and 2, phase solutions are calculated, stored and propagated as `h5parms`. The structures needed for the administration of these solutions are as follows:

- A master text file (MTF). The first row should begin with # and contain a list of antennas, in order. Each subsequent row contains: the name of the corresponding `h5parm` file; the RA and Dec; and one column for each antenna which has an entry which is 1 or 0 according to whether the phase solution is good or not good.

- A routine `dir2phasesol`. This routine is provided with an RA and Dec. It uses the MTF and the extant phase tables to generate an `h5parm` which combines the nearest usable signal for each station in the data.

- A routine `updatelist`. This is done after a loop-3 solve. The loop-3 solve has begun by applying a phase table, and has put out an incremental phase table together with an indication of which stations have provided a good solution. The routine is given both the initial and incremental phase tables, together with the goodness of the solution per antenna (the former is read from the MTF and the latter provided by loop 3). It combines these to make a phase solution at the position of the current source, and writes another line into the MTF containing this solution.

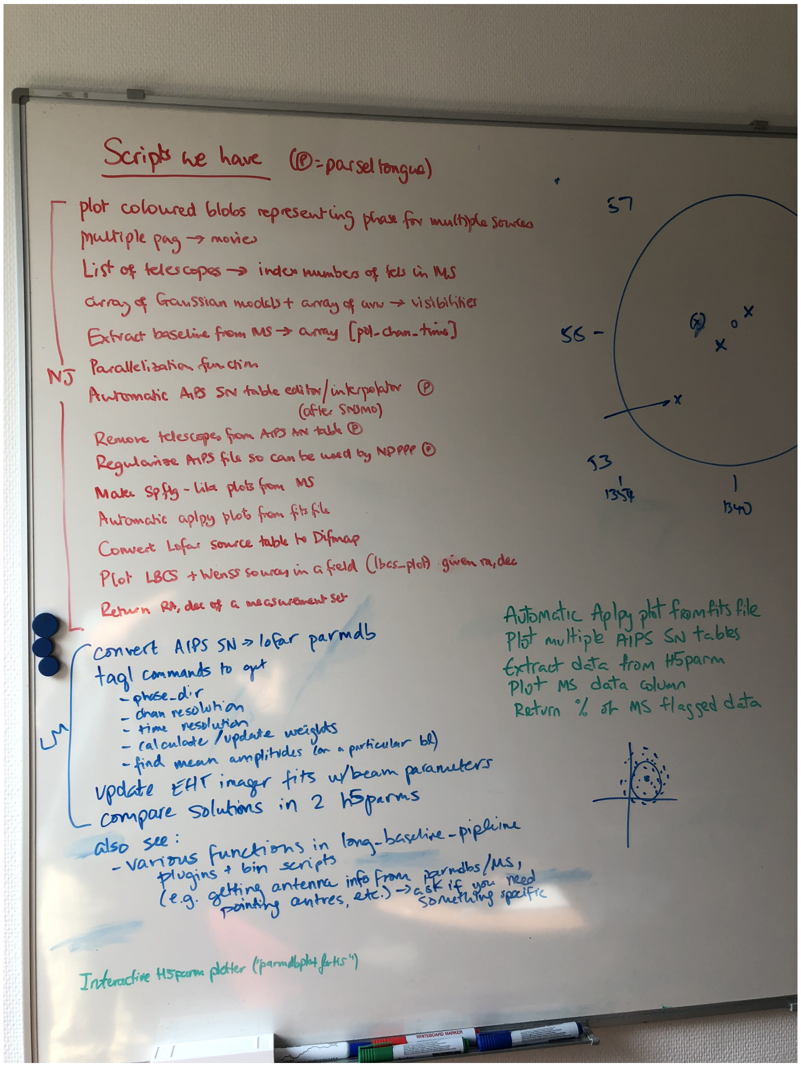
- A routine `applyh5parm`. This routine takes in an input MS and applies an `h5parm` to produce an output MS.

`All the h5 routines exist (written by Sean). h5parms to be given to Sean (by Leah, Neal, .....) for testing. Joe/Neal/Sean to discuss nature of loop3 solutions.`

Future telecons by Zoom

Action item: individuals to comment on and tick off items below!!

1. Benchmarking and speed increase of the production of shifted/averaged datasets. Investigating if killMS does this faster. `Implemented in pipeline - scales as sqrt n. Leah/Etienne discussion.`
2. Determine a relation between LoTSS flux and other cutoff criteria for selecting Iteration 1 sources. `Currently 5Jy LoTSS and 500mJy/LBCS.`
3. Answering the question about the 600-800km baselines at the end of Memo 4. `See later`
4. We should join all catalogue queries in one step. This is already done for LBCS and LoTSS; we should also include FIRST. This may not save much time, but there's no reason not to set everything up at the same time. `Same as question 2. LBCS to be archived (including fits files) and made accessible. NJ/MI/JC.`
5. Joint most important Writing and testing of loop 2 (`h5parm` routines and the interface to loop 3) `Sean leading. 4 routines; 2/3 available for testing. Needs comparing with AIPS. NJ testing.`
6. (Joint most important) Testing by hand of the eht/imaging/selfcal loop in Iteration 1 on a number of sources. Checking parameters and comparing against the pipeline (and memo 9). `Joe - Testing is continuing. Progress made with installation and bug reports to the EHT team`
7. Testing the subtraction step - in particular the NDPPP delay application.
8. Modifying the subtraction step to apply the delay solutions.
9. Providing quality control and visualisation steps to be run after important pipeline steps. This is particularly important for steps within the generic pipeline, which can otherwise fail bafflingly.
10. Figure out relation of sagecal Jones matrices to amplitude/phase corrections for diagnostic purposes
11. Test whether we are getting the signal-to-noise we should be getting by comparing with LBCS raw data (very preliminary indications: no [NJ]) and investigating why (e.g. dependence from field centre)
12. Test how vulnerable the phase calibration is to poor signal-to-noise using simulated data (basically get a dataset at a position away from bright sources, inject models into it and try to recover them using the pipeline/hand calibration)
13. Test the degree to which we can make maps of very faint sources, by making large-as-possible images with a central well-calibrated bright source subtracted in regions where a number of faint sources are present in LoTSS
14. Convert pipeline to use h5parms. Some steps will need to be adjusted, and at least some scripts – copyST_gains.py, for example. `LM/AD`



Scripts we have (Ⓟ=parsel tongue)

NJ

plot coloured blobs representing phase for multiple sources
multiple png → movies
List of telescopes → index numbers of tels in MS
array of Gaussian models + array of uvw → visibilities
Extract baseline from MS → array [pol-chan-time]
Parallelization function
Automatic AIPS SN table editor/interpolator Ⓟ
(after SNSMO)
Remove telescopes from AIPS AN table Ⓟ
Regularize AIPS file so can be used by NDPPP Ⓟ
Make Spflg-like plots from MS
Automatic aplpy plots from fits file
Convert Lofar source table to Difmap
Plot LBCS + Wenss sources in a field (lbcs_plot) given ra, dec
Return RA, dec of a measurement set

LM

convert AIPS SN → lofar parmdb
taql commands to get
- phase_dir
- chan resolution
- time resolution
- calculate/update weights
- find mean amplitudes (on a particular bl)

update EHT imager fits w/beam parameters
compare solutions in 2 h5parms

also see:
- various functions in long-baseline-pipeline plugins + bin scripts
(e.g. getting antenna info from parmdbs/MS, pointing centres, etc.) → ask if you need something specific

Automatic Aplpy plot from fits file
Plot multiple AIPS SN tables
Extract data from H5parm
Plot MS data column
Return % of MS flagged data

Interactive H5parm plotter ("parmdbplot for H5")

Figure 1: Scripts we would like from busy week at ASTRON 31 Aug 2018

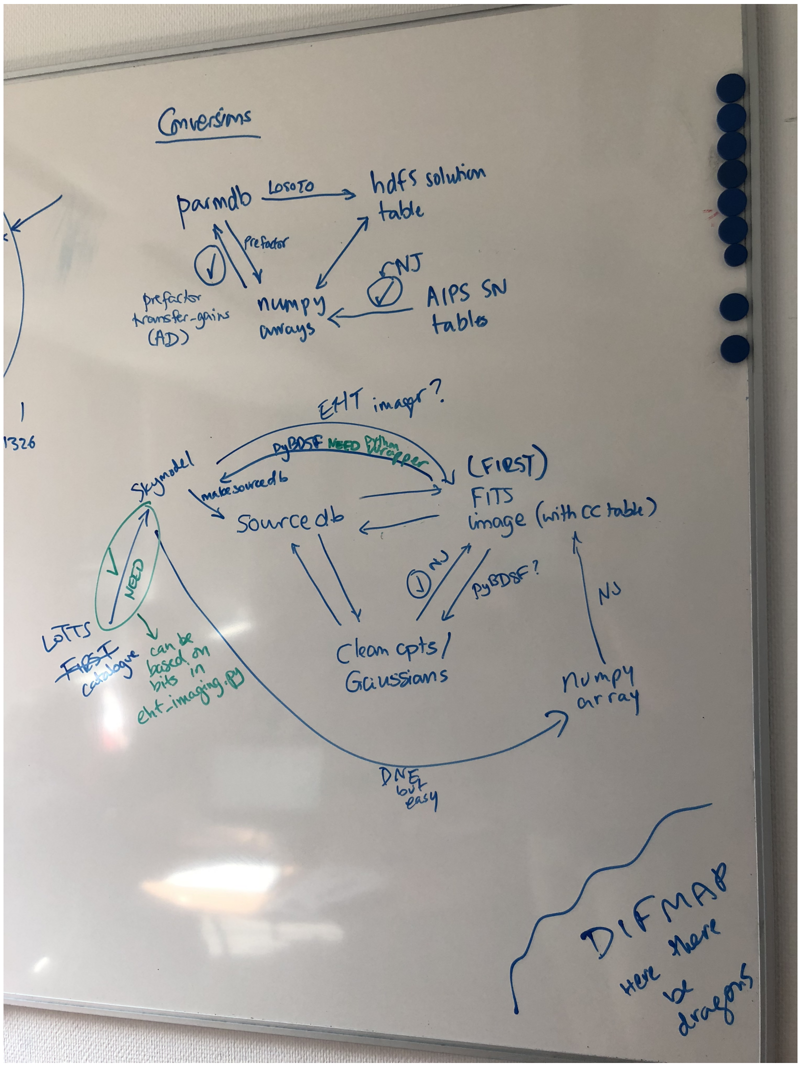


Figure 2: Scripts we would like from busy week at ASTRON 31 Aug 2018