HERA Memorandum #94: Comparing Visibility Solutions from Relative Redundant Calibration by Degenerate Translation

Matyas Molnar and Bojan Nikolic

Astrophysics Group, Cavendish Laboratory, University of Cambridge

Received 5th February 2021, updated 7th June 2022

Abstract

We develop routines to make sets of separately redundantly calibrated visibilities directly comparable by solving for degenerate parameter offsets between them. This allows computation of statistics of visibilities across different days (without the need for absolute calibration) and provides a new way of identifying and flagging data corrupted by e.g. radio-frequency interference (RFI). The computations are done using maximum likelihood estimation (MLE) techniques developed previously that allow for robust statistics. The method appears to be a useful quality assessment step and, in initial application, appears to identify plausibly bad data not previously flagged.

This memo builds on HERA Memorandum #84 [1], which presents a generalized approach to redundant calibration, with MLE computations accelerated in JAX. The code for the work described in this memorandum can be found at `https://github.com/bnikolic/simpleredcal`.

1 Introduction

Redundant calibration is a key step in the processing of Hydrogen Epoch of Reionization Array (HERA) data. Like the more widely used "self-calibration", redundant calibration solves for both the visibilities and the gains of the receiving elements of the interferometer. Unlike self-calibration, the visibilities when doing redundant calibration are only estimated in the relatively few sampled cells in the UV plane where the redundant baselines happen to lie. In self-calibration, since the model of the sky is estimated in the image-plane, the technique effectively estimates the visibility values across the whole UV plane. Consequently, self-calibration benefits greatly from combining multiple times and frequencies into a single estimate of the sky, as earth-rotation and bandwidth smearing fill out the coverage of the UV plane.

Since redundant calibration does not gain anything by filling out the UV plane, the incentive to combine multiple times/frequencies of observations is less immediate. Nevertheless, there are some reasons to compare redundant calibration estimates of visibilities across different times and frequencies, and potentially eventually combine the solving so that a single all-encompassing estimate is made based on as much data as possible. The reasons for this are:

1. Independently solving for the visibility values introduces more degrees of freedom than needed – the sky visibilities *must* be exactly equal at the same Local Apparent Sidereal Time (LAST) on different days, and as frequency channels are made narrower, the visibilities of neighbouring redundant calibration solutions must converge too. More degrees of freedom typically means a higher possibility of over-fitting and some loss on signal-to-noise (S/N).

2. RFI introduces a strongly non-Gaussian error into radio observations. Robust estimators of visibilities and antenna gains are far easier to construct as more data are considered simultaneously.

In this memorandum, we present a way of comparing redundant calibration visibility solutions at the same LAST across Julian dates (JDs) to test their stability[1], which provides another angle on the quality of the calibration and the data itself. We further discuss how such results can be used for further flagging.

2 Comparing visibility solutions from redundant calibration

2.1 Degeneracies in redundant calibration

Relative redundant calibration gain solutions are degenerate in the space represented by an overall amplitude $A(\nu)$, an overall phase $\Delta(\nu)$, and two tilts (or phase gradient components) $\Delta_x(\nu)$ and $\Delta_y(\nu)$ [see e.g. 1]. This degeneracy applies to each solution interval separately, i.e. each time, frequency and day slot of data, for which the solver may converge to different parts of this space.

Observed visibilities are affected by $A(\nu)$, $\Delta_x(\nu)$ and $\Delta_y(\nu)$; $\Delta(\nu)$, however, has no influence on the visibilities at all. When true visibilities are expected to be the same or very similar in a set of slots (e.g. same frequency and LAST on different days), then the *observed* visibilities after relative redundant calibration most be consistent up to a transformation in the $A(\nu)$, $\Delta_x(\nu)$ and $\Delta_y(\nu)$ space.

Solving for, or potentially marginalizing, these parameters with an MLE framework enables us to check if these solutions are consistent, as well as allowing for direct comparison of the visibility solutions.

2.2 MLE to translate between degeneracies

To compare datasets that have been redundantly calibrated, we minimize:

Gaussian assumed noise

$$-\ln\left(\mathcal{L}^G_{\rm deg}\right)(\nu) = \frac{1}{2}\sum_{\alpha}\sum_{\{i,j\}_\alpha}\ln\left(2\pi\sigma^2_{ij}(\nu)\right) + \frac{\left|U'_\alpha(\nu) - W_\alpha(\nu)\right|^2}{\sigma^2_{ij}(\nu)} \tag{1}$$

Cauchy assumed noise

$$-\ln\left(\mathcal{L}^C_{\rm deg}\right)(\nu) = \sum_{\alpha}\sum_{\{i,j\}_\alpha}\ln\left(\pi\gamma_{ij}(\nu)\right) + \ln\left(1 + \left(\frac{\left|U'_\alpha(\nu) - W_\alpha(\nu)\right|}{\gamma_{ij}(\nu)}\right)^2\right) \tag{2}$$

where

$$W_\alpha(\nu) = A^2(\nu)e^{i\left[\Delta_x(\nu)x_\alpha + \Delta_y(\nu)y_\alpha\right]}U_\alpha \tag{3}$$

with (x_α, y_α) the baseline coordinates of redundant set α, U_α the redundant calibration visibility solutions for the redundant baseline set indexed by α, and U'_α the set of visibility solutions on a different JD but same LAST.

The degenerate parameters found from minimizing either Eq. (1) or Eq. (2) can then be used to translate the visibility solutions of one dataset into the degenerate space of the other, so that both datasets are degenerately consistent. They can then be directly compared and statistical diagnostics may be run on them.

[1] Comparison over adjacent frequency channels and time integrations can also be done with the presented method and code, which can be used to quantify how visibilities vary over these dimensions.

2.3 Phase gradients

In relative redundant calibration, the degenerate parameters (§2.1) can take any value. This does not affect the convergence of the calibration but it can prevent convergence of the solver, which reconciles visibilities on different days. A particular problem encountered is that the phase gradient terms can take on very high values and vastly differ between adjacent times and frequencies.

To deal with this, when redundantly calibrating the 2$^{\text{nd}}$ dataset (U'_α), we reuse the solutions from the 1$^{\text{st}}$ redundant calibration as initial parameters, and introduce the following penalization term:

$$\mathcal{P}_{\text{tilt}} = \sum_i \left(\phi'_i - \phi_i\right)^2 \tag{4}$$

where ϕ_i are the solved gain phases from the 1$^{\text{st}}$ redundant calibration and ϕ'_i are the gain phases that minimize $-\ln(\mathcal{L}_{\text{rel}})$ [see §1.1 in 1] for the redundant calibration of the 2$^{\text{nd}}$ dataset. This penalty term ensures that the phases, and therefore the tilts, of the 2$^{\text{nd}}$ calibration solutions are as close as possible to those from the 1$^{\text{st}}$.

The final expression to minimize in the relative redundant calibration of the 2$^{\text{nd}}$ dataset, to ensure consistent tilts, is therefore:

$$-\ln(\mathcal{L}_{\text{rel}}) + \mathcal{P}_{\text{tilt}} \tag{5}$$

Computationally, it is found that constraining tilts in this manner, as well as using cartesian coordinates for the input parameters ($\mathfrak{Re}$ and $\mathfrak{Im}$ components for both visibilities and gains) in relative redundant calibration, is the fastest, easiest and most reliable method.

3 Implementation

3.1 Fully worked example

The code for comparing a pair of relatively calibrated visibility solutions for a particular frequency and time slice on separate JDs, and finding the degenerate parameters that translate between, is shown below. As in [1], we accelerate our computations with JAX. A full example notebook can be found at SimpleRedCal.ipynb.

Here, we use HERA data observed on JDs 2458098.43869 and 2458099.43124, taken from H1C_IDR2.2. We select frequency channel 605 and the 0$^{\text{th}}$ time integration of the 1$^{\text{st}}$ dataset, as an example. We only look at the EE polarization, assume that the noise follows a Gaussian distribution, and neglect the noise contribution (1$^{\text{st}}$ term in Eq. (1)) in MLE calculations (effectively assuming it is uniform across baselines).

We start by loading the 1$^{\text{st}}$ HERA H1C_IDR2 dataset.

```
from hera_cal.io import HERAData
from jax import numpy as jnp
from simpleredcal.red_likelihood import doDegVisVis, doRelCal, group_data, \
red_ant_sep, relabelAnts, split_rel_results
from simpleredcal.red_utils import find_flag_file, find_nearest, find_zen_file, \
get_bad_ants, match_lst

# Select 1st dataset to relatively calibrate
JD1 = 2458098.43869
chan = 605  # frequency channel
time_int1 = 0  # time integration of 1st dataset
noise_dist = 'gaussian'  # assumed noise distribution
```

```
coords = 'cartesian'  # parameter coordinate system

zen_fn1 = find_zen_file(JD1)  # find path of dataset
bad_ants1 = get_bad_ants(zen_fn1)  # get bad antennas from commissioning
flags_fn1 = find_flag_file(JD1, 'first')  # import flags from firstcal
print('Bad antennas for JD {} are: {}'.format(JD1, bad_ants1))
# Bad antennas for JD 2458098.43869 are: [0 2 11 24 50 53 54 67 69 98 122 136 139]

# Load dataset from uvh5 file to numpy array, with flagging applied
hdraw1, RedG1, cMData1 = group_data(zen_fn1, pol='ee', chans=chan, tints=time_int1,
                                    bad_ants=bad_ants1, flag_path=flags_fn1)
# 0 out of 741 data points flagged for visibility dataset
# zen.2458098.43869.HH.uvh5

cData1 = jnp.squeeze(cMData1.filled())  # filled with nans for flags
ants = jnp.unique(RedG1[:, 1:])
no_ants = ants.size  # number of antennas
no_unq_bls = jnp.unique(RedG1[:, 0]).size  # number of redundant baselines
cRedG1 = relabelAnts(RedG1)  # relabel antennas with consecutive numbering
```

We then load the 2nd dataset on another JD that matches the 1st dataset in LAST

```
# Select 2nd dataset to relatively calibrate, that matches the LAST of the 1st
JD2 = match_lst(JD1, 2458099, tint=time_int1)  # finding the JD_time of the dataset
# that matches the LAST of the dataset used in 1
zen_fn2 = find_zen_file(JD2)
bad_ants2 = get_bad_ants(zen_fn2)
flags_fn2 = find_flag_file(JD2, 'first')

# Find time int in dataset 2 that corresponds to closest LAST to that of dataset 1
hdraw2 = HERAData(zen_fn2)
time_int2 = int(find_nearest(hdraw2.lsts, hdraw1.lsts[time_int1])[1])

# Load dataset from uvh5 file to numpy array, with flagging applied
_, RedG2, cMData2 = group_data(zen_fn2, pol='ee', chans=chan, tints=time_int2,
                               bad_ants=bad_ants2, flag_path=flags_fn2)
# 0 out of 741 data points flagged for visibility dataset
# zen.2458098.43869.HH.uvh5

cData2 = jnp.squeeze(cMData2.filled())  # filled with nans for flags

print('Do the visibilities for JDs {} and {} have:\nthe same bad antennas? {}\n'\
      'the same redundant grouping? {}'.format(JD1, JD2, (bad_ants1 == bad_ants2)\
      .all(), (RedG1==RedG2).all()))

# Do the visibilities for JDs 2458098.43869 and 2458099.43124 have:
# the same bad antennas? True
# the same redundant grouping? True
```

Relative redundant calibration is now performed on both datasets, with Gaussian assumed noise:

```
# Relative redundant calibration of the 1st dataset
res_rel1, initp = doRelCal(cRedG1, cData1, no_unq_bls, no_ants,
```

```
        distribution=noise_dist, coords=coords, norm_gains=True, return_initp=True)
# Optimization terminated successfully.

# Relative redundant calibration of the 2nd dataset
res_rel2 = doRelCal(cRedG1, cData2, no_unq_bls, no_ants,
        distribution=noise_dist, coords=coords, norm_gains=True, initp=initp,
        phase_reg_initp=True)
# Optimization terminated successfully.

# Get the relatively calibrated gain and visibility solutions
res_rel_vis1, res_rel_gains2 = split_rel_results(res_rel1['x'], no_unq_bls,
                                                 coords=coords)
res_rel_vis2, res_rel_gains2 = split_rel_results(res_rel2['x'], no_unq_bls,
                                                 coords=coords)
```

We now compare the redundant visibility solutions from both of these datasets, again, using Gaussian assumed noise:

```
# Translating between relatively calibrated visibility sets
ant_sep = red_ant_sep(RedG1, hdraw1.antpos)
res_deg = doDegVisVis(ant_sep, res_rel_vis1, res_rel_vis2, distribution=noise_dist)
# Optimization terminated successfully.
print('Degenerate parameters are:\nAmplitude = {}\n'\
      'Phase gradient in x = {:e}\nPhase gradient in y = {:e}'.format(*res_deg['x']))
# Degenerate parameters are:
# Amplitude = 0.9920773888258738
# Phase gradient in x = -6.344390e-06
# Phase gradient in y = 8.818280e-05
```

The $\mathfrak{Re}$ and $\mathfrak{Im}$ residuals for the above are plotted in Fig. 1. The results if we had assumed Cauchy noise (done with `noise_dist = 'cauchy'`) are extremely similar.

3.2 Batch processing

The full relative calibration the datasets can be done with the rel_cal.py script:

```
python rel_cal.py '2458099.43124' --pol 'ee' --flag_type 'first' --dist 'gaussian'

python rel_cal.py '2458099.43869' --pol 'ee' --flag_type 'first' --dist 'gaussian'

python rel_cal.py '2458098.43869' --pol 'ee' --flag_type 'first' --dist 'gaussian'
--initp_jd 2458099
```

In the last command, we impose the tilt constraint from Eq. (4) to ensure that the solutions found have similar tilts as for JD 2458099. Note that in order to reuse the solutions for JD 2458099, the first two executions must have ended.

Comparison of two datasets by fitting for the degenerate parameters that translate between their relative redundant calibration visibility solutions can be done with deg_cal.py:

```
python deg_cal.py '2458098.43869' --deg_dim 'jd' --pol 'ee' --dist 'gaussian'
--tgt_jd 2458099
```

Note that adjacent times and frequencies can be compared by specifying `'tint'` and `'freq'` after the

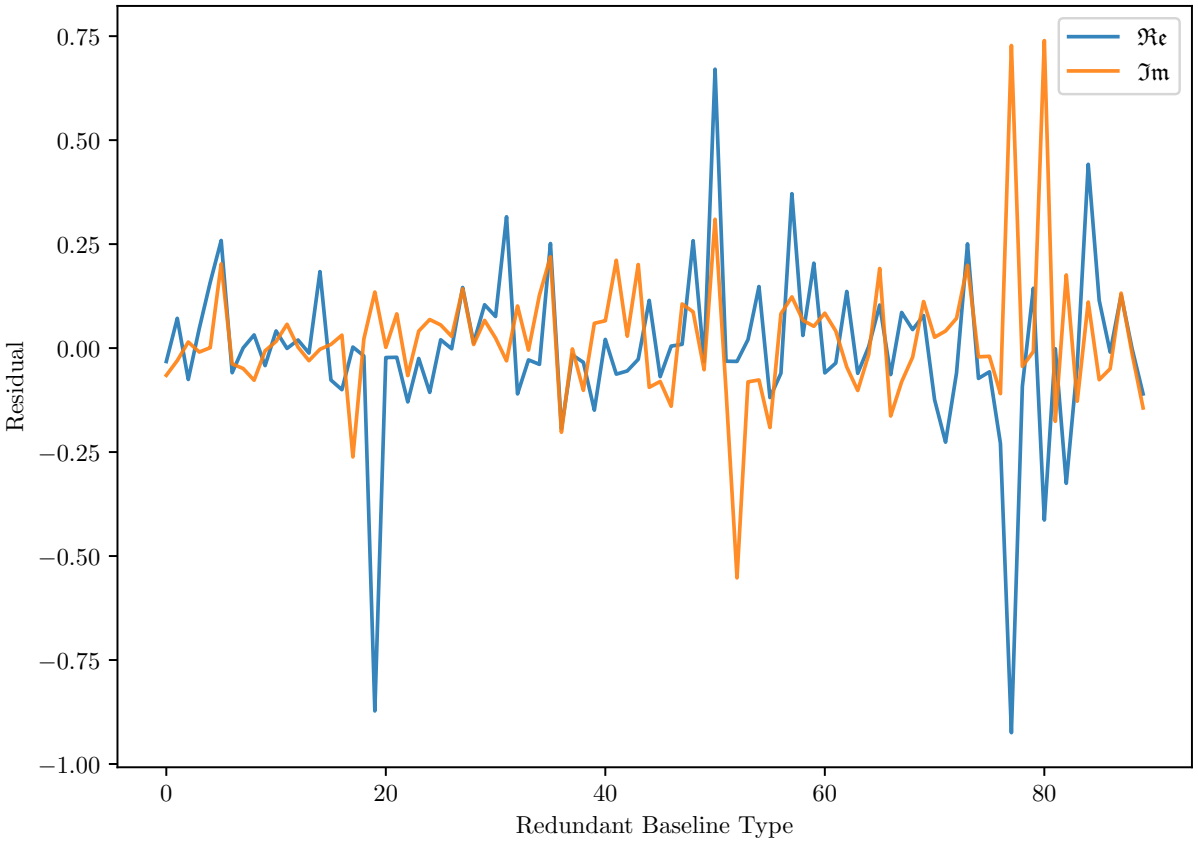


Figure 1: Normalized residuals, by visibility amplitude, between visibility solutions on JDs 2458098 and 2458099 at LAST 1.3826 for frequency channel 605. The length of the baseline is decreased and the number of baseline of each type is increased as the baseline type ID is increased. The 14 m and 28 m East-West baselines are represented by baseline types 2 and 6, respectively. Note that the normalization is done by dividing the residuals by the geometric mean of the redundant visibility amplitude solutions for the two days being compared.

`deg_dim` argument, respectively.

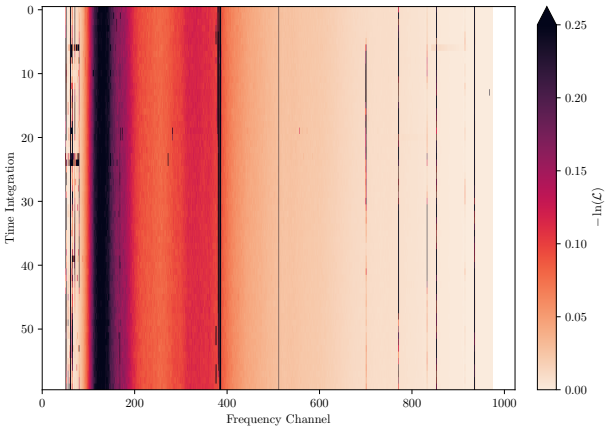
4 Results

4.1 Relative redundant calibration

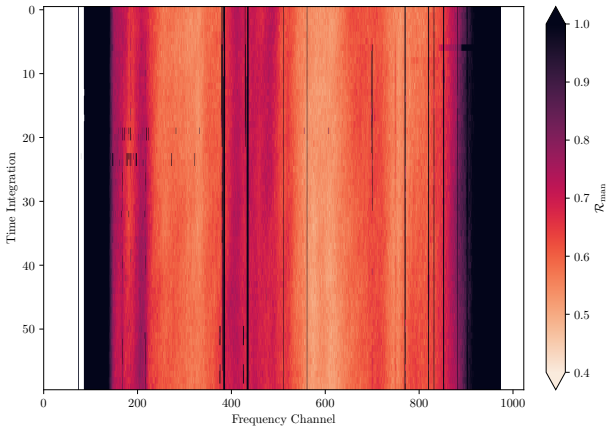
The minimum negative log-likelihoods $-\ln(\mathcal{L}_{\rm rel})$ for the relative redundant calibration of datasets labelled by JDs 2458098.43869, 2458099.43124 and 2458099.43869, with Gaussian assumed noise, are plotted in the left column of Fig. 2. We also use the median (over baselines) absolute normalized residual for these solutions, given by

$$\mathcal{R}_{man} = \underset{\rm bls}{med}\left(\frac{|V_{\rm meas} - V_{\rm pred}|}{\sqrt{|V_{\rm meas}|\,|V_{\rm pred}|}}\right) \tag{6}$$

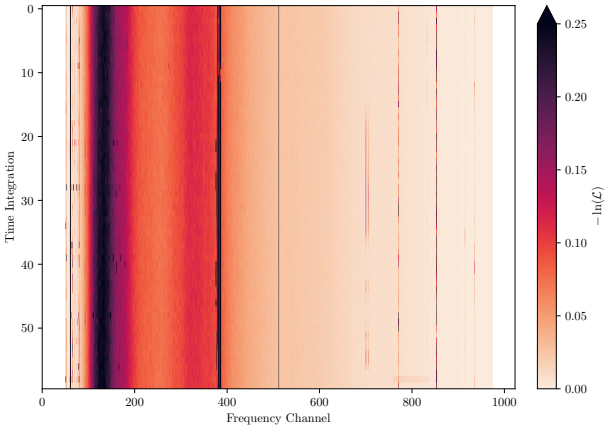
which can be thought of as a goodness of fit measure for this minimization (i.e. the median of the absolute values of Fig. 1, but for each frequency/time integration slice). These are also shown in the right column of Fig. 2.



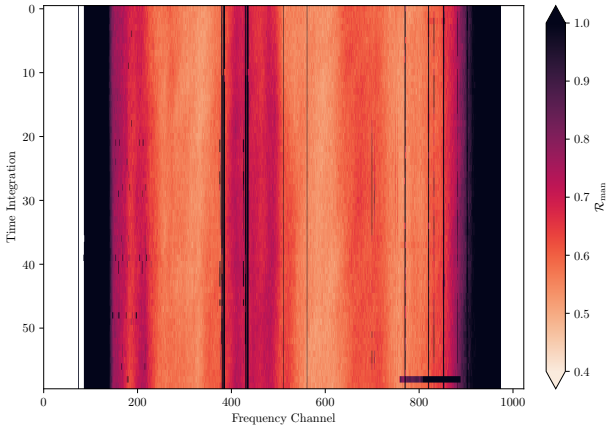
(a) Minimum $-\ln(\mathcal{L}_{\rm rel})$ for the relative calibration of HERA dataset labelled by JD 2458098.43869.



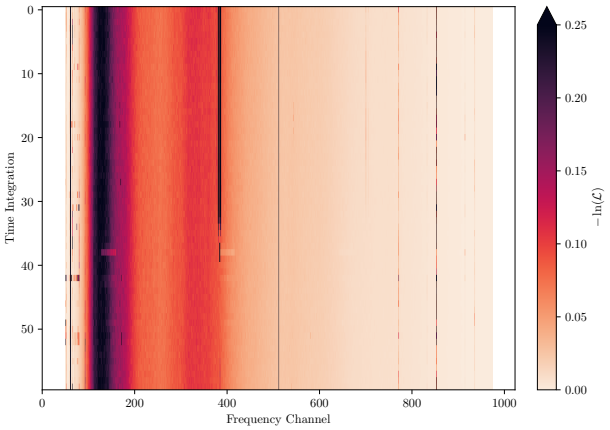
(b) $\mathcal{R}_{man}$ for the relative calibration of HERA dataset labelled by JD 2458098.43869.



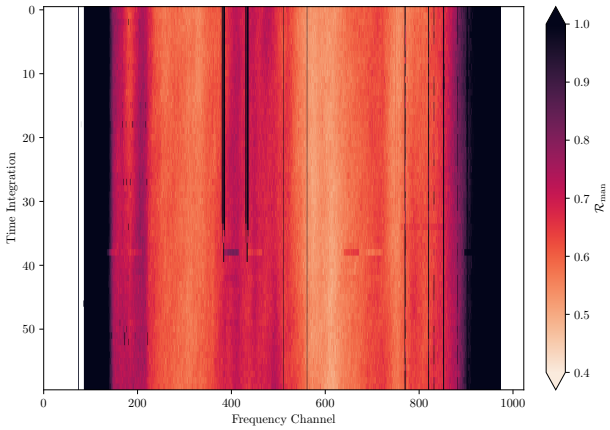
(c) Minimum $-\ln(\mathcal{L}_{\rm rel})$ for the relative calibration of HERA dataset labelled by JD 2458099.43123.



(d) $\mathcal{R}_{man}$ for the relative calibration of HERA dataset labelled by JD 2458099.43124.



(e) Minimum $-\ln(\mathcal{L}_{\rm rel})$ for the relative calibration of HERA dataset labelled by JD 2458099.43869.



(f) $\mathcal{R}_{man}$ for the relative calibration of HERA dataset labelled by JD 2458099.43869.

Figure 2: Minimum $-\ln(\mathcal{L}_{\rm rel})$ (left) and $\mathcal{R}_{man}$ (right) for the relative redundant calibration of visibilities on JDs 2458098 (between LAST 1.3826 and 1.4288; Figs. 2a and 2b) and 2458099 (between LAST 1.3528 and 1.4460; Figs. 2c to 2f), [using Eq. (2) in 1]. Note that the penalty term $\mathcal{P}_{\rm tilt}$ has been added in Fig. 2a (see Eq. (5)) so that tilts are the same as for the visibilities on JD 2458099 at the same LAST. Extreme values have been clipped.

4.2 Comparing visibility solutions

The minimum negative log-likelihoods $-\ln(\mathcal{L}_{\text{deg}})$ found by degenerate comparison of the redundant visibility solutions from the HERA dataset labelled by JD 2458098.43869 and the corresponding visibilities that match in LAST on JD 2458099 (labelled by 2458099.43124 and 2458099.43869, due to a slight offset in time), with Gaussian assumed noise, are shown in Fig. 3. We further show $\mathcal{R}_{man}$ for this minimization in Fig. 4.

In Figs. 2 to 4, the dark vertical lines correspond to RFI from known bad frequency channels. The negative log-likelihoods in these plots also decreases as the frequency channel increases; this is due to the natural shape of visibility solutions whose amplitudes decrease for higher frequencies.

We note a few dark horizontal bars at the upper end of the frequency channels, which indicate that the solver struggled to reconcile the visibility solutions. These can be traced back to Fig. 2, where these dark horizontal bars also appear in the redundant calibration of these datasets. The cause of this is most likely intermittent RFI.

It is important to remember that the negative log-likelihoods in Fig. 3 compares a pair of redundant calibration visibility solutions: any dark region may be from a fault in either of the JDs being compared. We note that if this computation is extended to additional JDs, the anchor day is being compared to all other days, therefore any RFI present in the anchor day will be present in the degenerate comparison negative log-likelihoods. To avoid having to rely on a *good* anchor day and computing all possible pairs of comparisons, one can transform all visibility solutions from all JDs to the same degenerate space, and run statistics on the entire dataset to find outliers. This is work in progress.

For reference, we show in Fig. 5 the final flagging that is applied on the very same visibility datasets if calibration is done through the HERA analysis pipeline, which employs hera_cal for its calibration routines.

The results from the degenerate comparison, notably $-\ln(\mathcal{L}_{\text{deg}})$ and $\mathcal{R}_{man}$, can be used for further flagging: we can plot histograms for both of these quantities, separated by the flagging from Fig. 5; in doing so, we get a feel for the distribution of the data, which consequently indicates a reasonable cut-off value, after which frequency/time slices should be cast as outliers and flagged (if they haven't been already). There is, however, some spectral structure in $-\ln(\mathcal{L}_{\text{deg}})$ (see Fig. 3) due to the shape of the bandpass. We therefore divide $-\ln(\mathcal{L}_{\text{deg}})$ by the smoothed variance for the 14 m baselines $\sigma^2_{14\,\text{m}}$ calculated from auto-correlations to obtain a dimensionless quantity that is comparable across frequencies. The histograms for $-\ln(\mathcal{L}_{\text{deg}})$ and $\mathcal{R}_{man}$ are plotted in Fig. 6.

We compute $\sigma^2_{14\,\text{m}}$ by taking the mean of the noise variance for the 14 m baselines (the noise is calculated using the `.std.uvh5` files; see e.g. [2]), sigma-clipping about the median at the 4σ level, linearly interpolating the sigma-clipped values, and applying a Savitzky-Golay filter with window length of 17 time integrations and fitting polynomial of order 3 (with negative values set to 1×10^{-8}), all along the time dimension. This smoothing is done so that any intermittent high noise do not cancel out high $-\ln(\mathcal{L}_{\text{deg}})$ values. No smoothing is done on the frequency dimension: smoothing over frequency is harder, so for the purpose of this study we do not smooth over it. In addition, bad frequency channels more easily identifiable. The histograms for $-\ln(\mathcal{L}_{\text{deg}})/\sigma^2_{14\,\text{m}}$ and $\mathcal{R}_{man}$ are plotted in Fig. 6.

From Figs. 5 and 6, we find several frequency/time slices that lie above the indicative outlier cut-offs, but that are not flagged through the HERA pipeline (cf. Fig. 5). We present these slices in Table 1.

We further show the visibility amplitudes and phases for the slices in Table 1 for the redundantly solved 14 m baseline, as well as for the $(12, 13)$ baseline of the raw data, in Figs. 7 and 8.

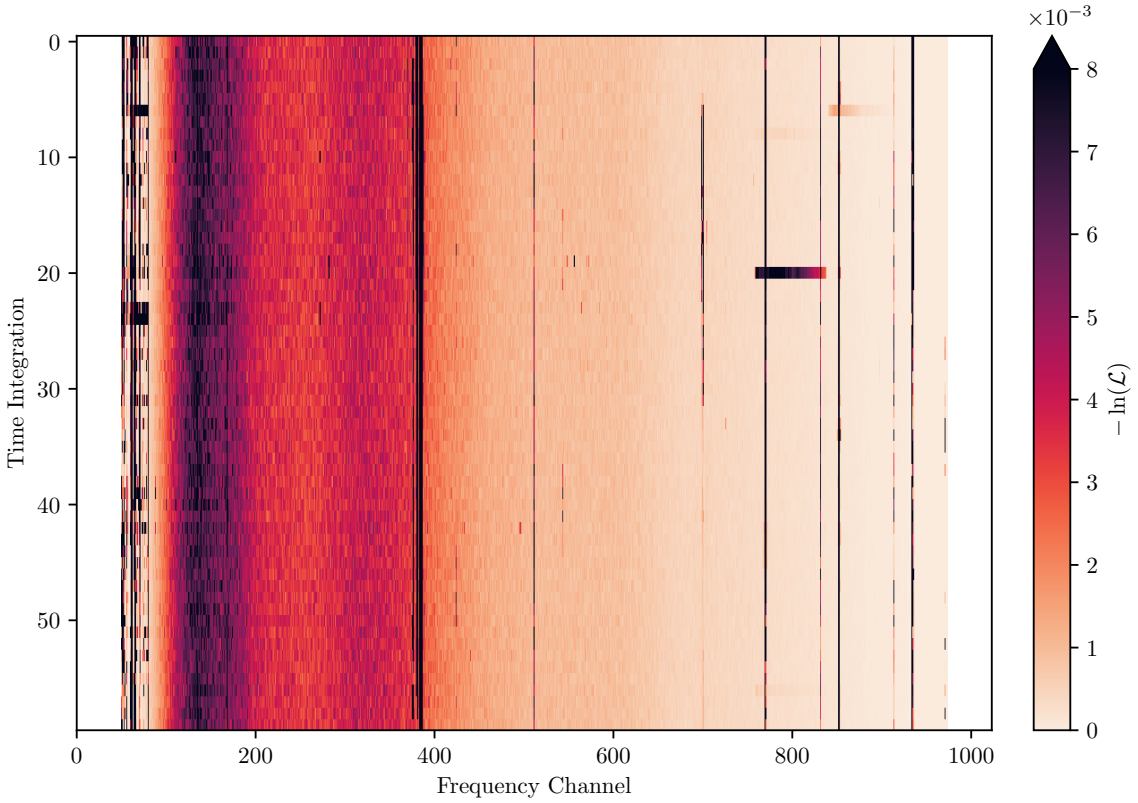


Figure 3: Minimum $-\ln\left(\mathcal{L}_{\rm deg}\right)$ for the comparison of visibilities on JDs 2458098 and 2458099 between LAST 1.3826 and 1.4288, using Eq. (1). Extreme values have been clipped.

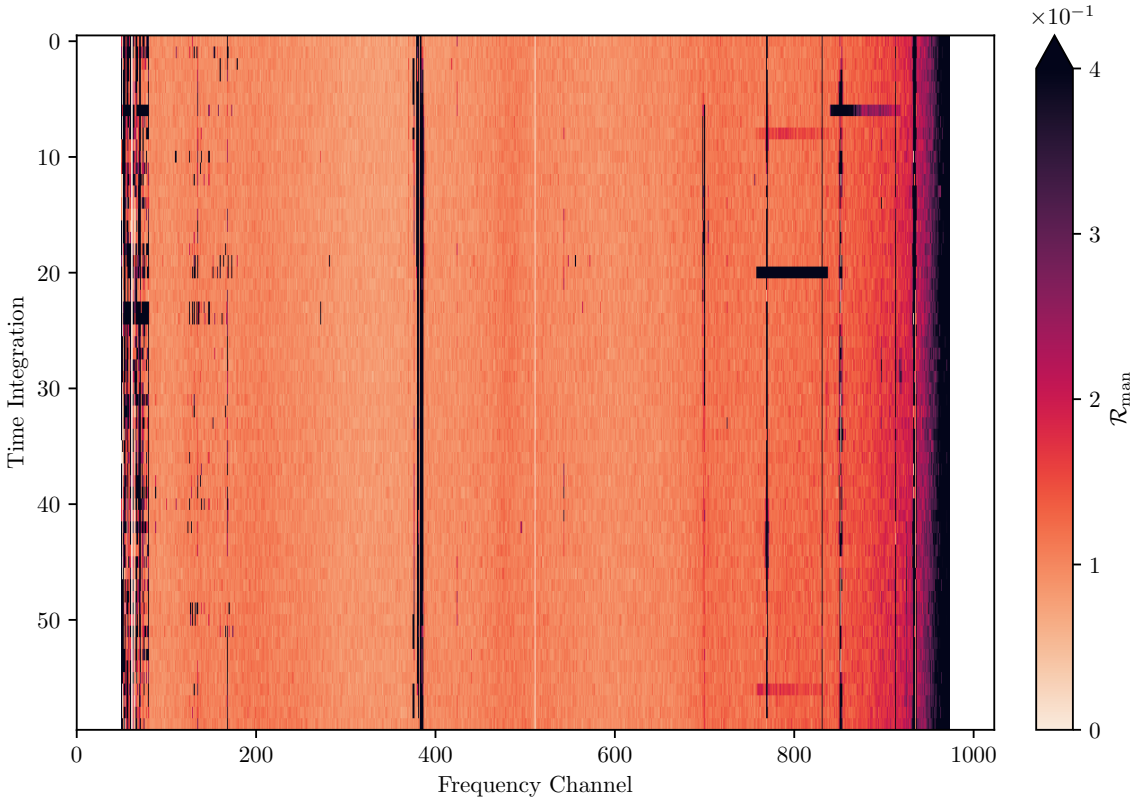


Figure 4: $\mathcal{R}_{man}$ when comparing between redundant visibility solutions on JDs 2458098 and 2458099 between LAST 1.3826 and 1.4288, post degenerate translation. Extreme values have been clipped.

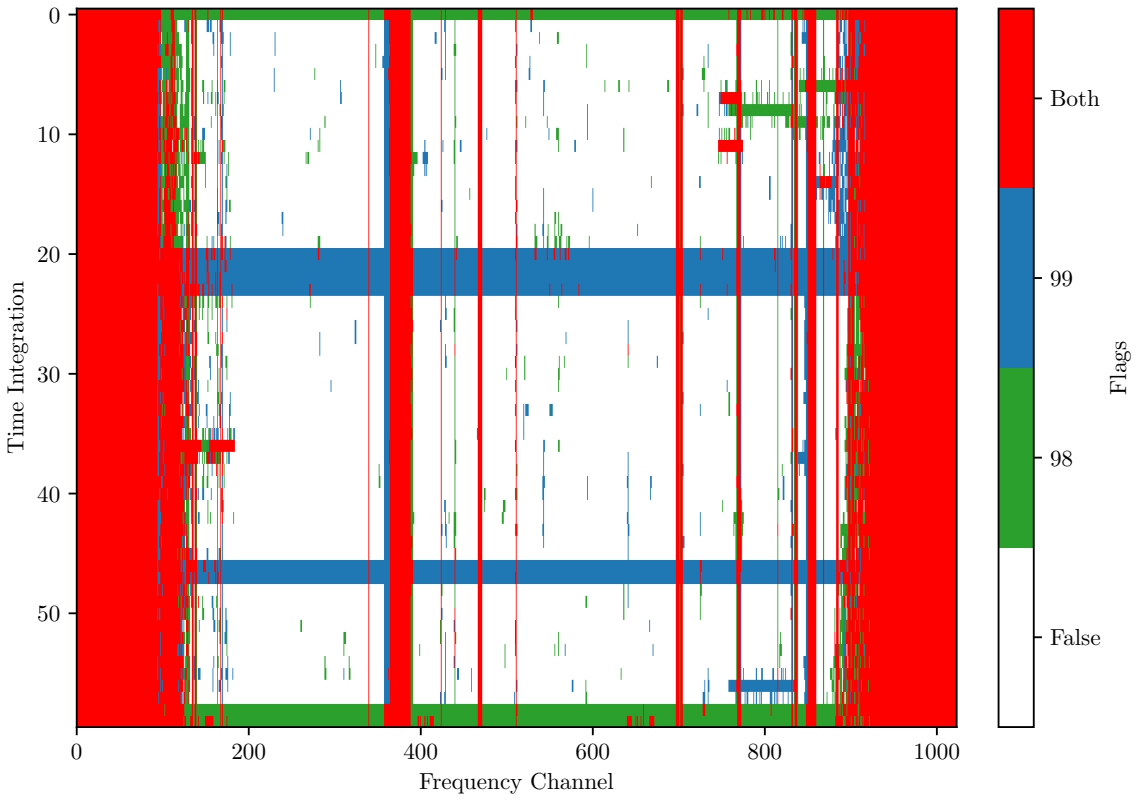
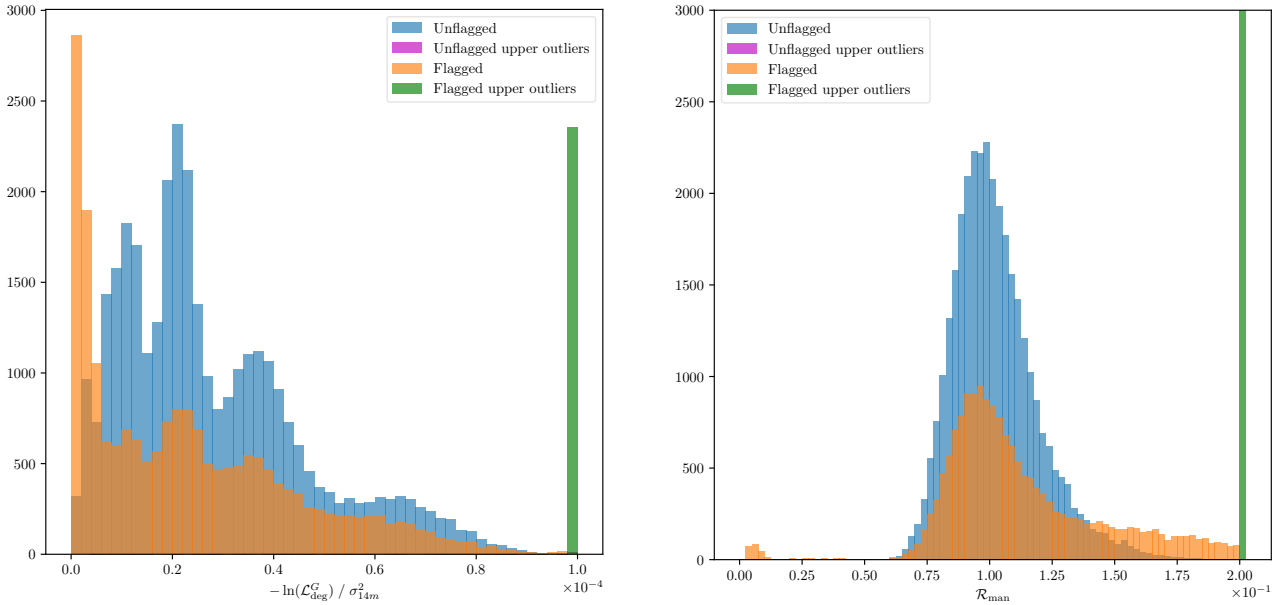


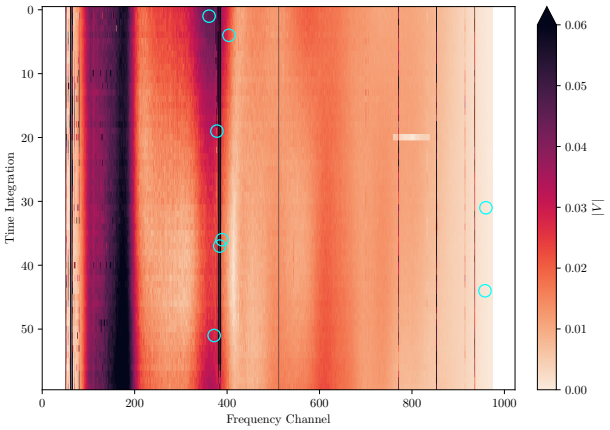
Figure 5: Flags from the `zen.2458098.43869.HH.smooth_abs.calfits`, `zen.2458099.43124.HH.smooth_abs.calfits` and `zen.2458099.43869.HH.smooth_abs.calfits` calibration files, which contain the final flags from the HERA analysis pipeline, aligned in LAST.



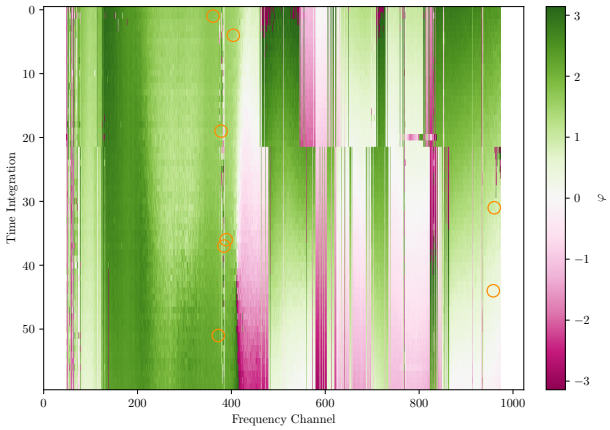
(a) Histogram of $-\ln(\mathcal{L}_{\rm deg})/\sigma^2_{14\,\rm m}$.

(b) Histogram of $\mathcal{R}_{man}$.

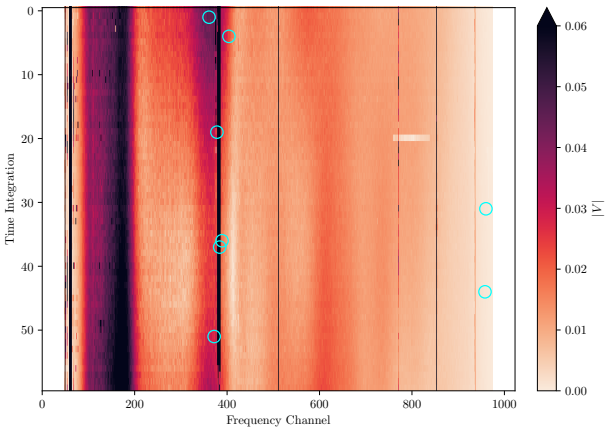
Figure 6: Histograms of $-\ln(\mathcal{L}_{\rm deg})/\sigma^2_{14\,\rm m}$ and $\mathcal{R}_{man}$, separated by the final set of flags from the HERA pipeline (see Fig. 5), for the comparison of redundant visibility solutions on JDs 2458098 and 2458099 between LAST 1.3826 and 1.4288, through degenerate translation. Outliers, $-\ln(\mathcal{L}_{\rm deg})/\sigma^2_{14\,\rm m} > 9.8 \times 10^{-5}$ and $\mathcal{R}_{man} > 0.2$, are grouped and represented as single columns (n.b. the y-axis range of Fig. 6b has been limited; the count for the flagged upper outliers is 4885).



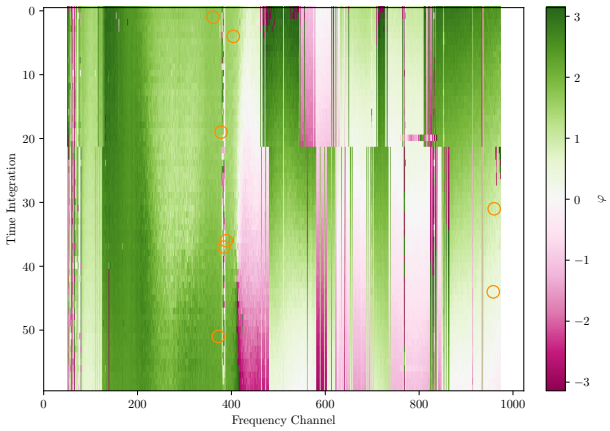
(a) Redundantly calibrated visibility amplitude for the 14 m East-West baseline on JD 2458098.



(b) Redundantly calibrated visibility phase for the 14 m East-West baseline on JD 2458098.

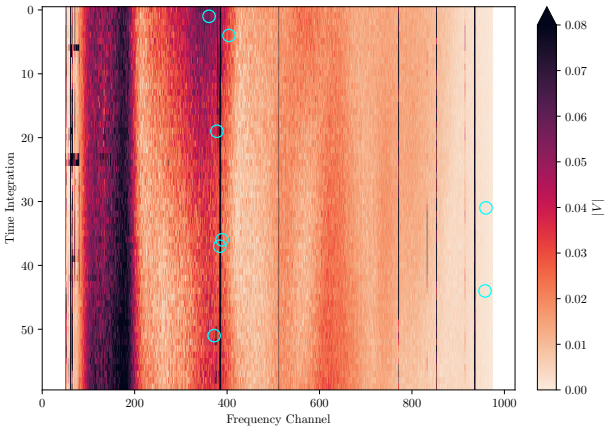


(c) Redundantly calibrated visibility amplitude for the 14 m East-West baseline on JD 2458099.

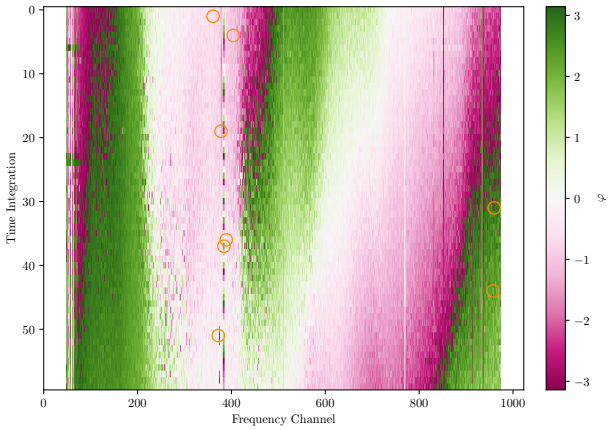


(d) Redundantly calibrated visibility phase for the 14 m East-West baseline on JD 2458099.

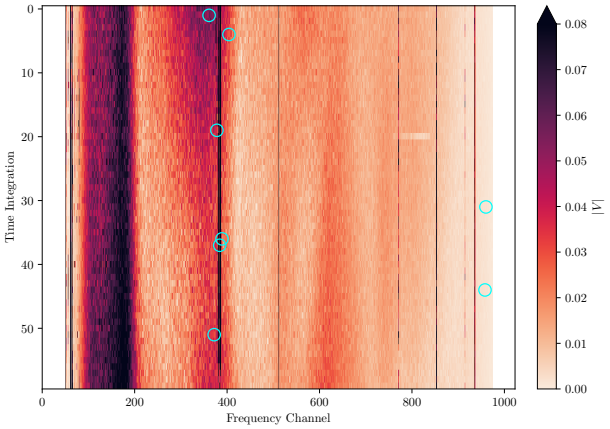
Figure 7: Redundantly calibrated visibilities for the 14 m East-West baselines for JDs 2458098 and 2458099 between LAST 1.3826 and 1.4288. The outliers from Table 1 are circled. Note that the jump in visibility phases in Figs. 7b and 7d at time integration 22 corresponds to the point at which redundantly calibrated visibilities from datasets labelled by 2458099.43124 and 2458099.43869 have been concatenated. Extreme values for the visibility amplitudes have been clipped.



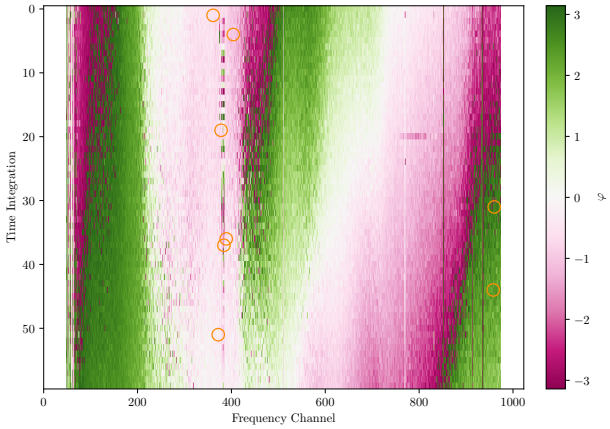
(a) Raw visibility amplitude for baseline $(12, 13)$ (14 m East-West) on JD 2458098.



(b) Raw visibility phase for baseline $(12, 13)$ (14 m East-West) on JD 2458098.



(c) Raw visibility amplitude for baseline $(12, 13)$ (14 m East-West) on JD 2458099.



(d) Raw visibility phase for baseline $(12, 13)$ (14 m East-West) on JD 2458099.

Figure 8: Raw visibilities for the 14 m East-West baseline between antennas 12 and 13 for JDs 2458098 and 2458099 between LAST 1.3826 and 1.4288. The outliers from Table 1 are circled. Extreme values have been clipped.

Channel	Time integration	$-\ln(\mathcal{L}_{\text{deg}})/\sigma^2_{14\,\text{m}}$ $/10^{-6}$	$\mathcal{R}_{man}$ $/10^{-2}$	$-\ln(\mathcal{L}_{\text{deg}})$ $/10^{-3}$	$\sigma^2_{14\,\text{m}}$
311	1	9.823	7.557	5.919	60.258
322	51	9.933	9.661	5.807	58.455
328	19	10.663	8.481	6.165	57.814
334	37	9.870	10.154	5.630	57.036
339	36	9.963	10.111	5.595	56.157
354	4	9.988	9.100	5.527	55.340
908	31	1.244	20.055	0.049	39.733
910	44	1.407	20.712	0.054	38.447

Table 1: Potentially bad frequency/time slices found from Fig. 6. The outlier cut-offs in this study were chosen ad hoc to be $-\ln(\mathcal{L}_{\text{deg}})/\sigma^2_{14\,\text{m}} > 9.8 \times 10^{-5}$ and $\mathcal{R}_{man} > 0.2$. We colour in red the values that are beyond the set cut-offs. The time integration is that for `zen.2458098.43869.HH.uvh5`. We note that these slices do not exceed the cut-offs by significant amounts. Further work is required to see if flagging these slices improve the calibration and/or power spectrum (PS) estimation.

5 Conclusion

In this memorandum, we showed how a pair of redundant calibration visibility solution datasets, solved through a generalized MLE framework, can be compared by translating between their degenerate parameters. Through this comparison, the stability of redundant calibration visibility solutions can be probed, and the negative log-likelihoods for this comparison (Fig. 3) can be used as a metric for further flagging: certain effects that may appear as correct calibration solutions may not be present day-to-day, and should be flagged.

We are in the processing of extending this method to compare multiple JDs, although this requires an anchor JD that sets the tilts, and is computationally expensive: the relative calibration of a single dataset takes ≈ 40 hours using 4 cores and the degenerate comparison a further ≈ 6 hours. We are also looking to speed up this computation by further optimizing the code and running it on GPUs.

After having run this analysis on the whole of the H1C_IDR2.2, we plan on applying the flagging found through this method (with both Gaussian and Cauchy assumed noise), on top of the flagging from the analysis pipeline that uses hera_cal, to then run through PS estimation.

Further down the line, we plan on building a redundant generalized solver to take data across all JDs to find a single set of visibility solutions for any LAST, which will give the best location and scale estimates for visibilities. This will be computationally expensive, but will be faster than running redundant calibration on all the individual datasets. Machine learning (ML) software and accelerated hardware will also speed this up.

References

[1] M. Molnar and B. Nikolic, "HERA Memorandum #84: A Generalized Approach to Redundant Calibration with JAX", November 2020.

[2] J. S. Dillon, "HERA Memorandum #69: H1C IDR 2.2", July 2019.