1 Introduction

I, with the help of Devojyoti, wrote these codes to investigate the effect of linear and non-linear polarisation calibration schemes, under various conditions. In particular, I have tested the cases, where the instrumental leakage is low and high. Under each possibility, I have also investigated the effect when the source has high and low polarisation fractions. For the linear calibration scheme, I have used CASA. For the non-linear calibration scheme, I have used quartical as well as a code I wrote. While the codes were developed by me, I have discussed the details of the procedure with Devojyoti and we are in complete agreement about the results and the methodology. So, this is really a joint effort and should be treated as such. In the spirit of reproducibility, we thought that it is better to upload all codes used in this work on Github, along with a detailed description of the steps which is needed to run this code as well as the understanding we get from it.

2 Simulating visibilities

Let us denote the jones matrix corresponding to antenna i, which describes the leakage matrix, as

$$J_i^{leakage} = \begin{pmatrix} 1 & d_{xi}e^{i\phi_{xi}} \\ d_{yi}e^{i\phi_{yi}} & 1 \end{pmatrix} \tag{1}$$

Both d_{xi}s and d_{yi}s are chosen from randomly from a uniform distribution. For the low leakage case, the uniform distribution corresponding to $(0.01, 0.07)$. For the high leakage case, the limits are $(0.15, 0.25)$. ϕ_{xi} and ϕ_{yi} are drawn from a gaussian distribution with mean of 50° and sigma of 10°. We also use a crosshand phase (θ) to be 50°. Thus the combined Jones matrix for each antenna i is given by

$$J_i = \begin{pmatrix} 1 & d_{xi}e^{i\phi_{xi}} \\ d_{yi}e^{i\phi_{yi}} & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{2}$$

The visibilities for baseline corresponding to the antenna pair (p, q) for a source with Stokes parameters (I, Q, U, V) are predicted as

$$\begin{pmatrix} V_{pq,RR} & V_{pq,RL} \\ V_{pq,LR} & V_{pq,LL} \end{pmatrix} = 0.5 F J_p \begin{pmatrix} I+V & Q+iU \\ Q-iU & I-V \end{pmatrix} J_q^{\dagger}, \quad F \text{ is the Fourier Transform operator} \tag{3}$$

We simulate the visibilities for 5 point sources, the properties of which are given in Table 1

Source name	I	Q	U	V
S_1	10	0	0	0
S_2	50	1	5	0
S_3	50	5	1	0
S_4	50	4	15	0
S_5	50	15	-5	0

Table 1: Model properties of the sources simulated

The goal of this experiment is to investigate how each of the algorithms fare in recovering these input source properties after calibration.

The simulated MS can be generated in the following way.

1. If low leakage case, first run *simulate_leakages.py* in the low_leakages folder. Similarly,for different cases, we run the codes with same name, but located in its own corresponding folder.

2. Create a folder called CASA, quartical and self_crosshand inside the folder. If you have the github code, then the folders already exist.

3. Copy a GMRT MS (band2band3band4, just to ensure that the polarization is stated as RRLL in the MS header), such that it has only 1 channel and 1 time. Name the MS es should be as follows: 'CASA/unpolarized_source_I10.ms', 'CASA/polarized_source_I50_Q1_U5_V0.ms', 'CASA/polarized_source_I50_Q5_U1_V0.ms', 'CASA/polarized_source_I50_Q4_U 'CASA/polarized_source_I50_Q15_neg_U5_V0.ms'

4. Next run simulate_vis_polarized_all.py after changing the file names appropriately.

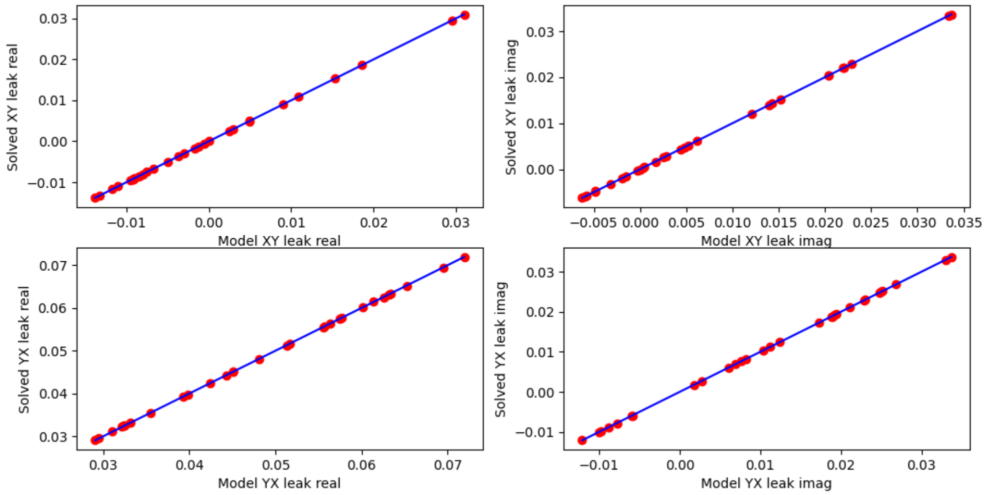


Figure 1: XX

3 Low leakage

3.1 Linear analysis

Linear analysis under each test scenario can be performed by running the script named polcalibrate_CASA.py located inside the CASA folder in each test case.

In the linear regime, we can solve for relative leakages in the circular basis. Hence, for accurate comparison, we have used a reference antenna to solve for the leakages. When comparing with the input leakages, we have appropriately computed the relative leakage of each antenna with respect to the reference antenna. In Figure 1, we show the comparison between the relative input and solved leakages. The script named compare_leakage_model.py inside the CASA folder is run to produce this figure. It is evident that the relative leakages solved matches exacty with the relative input leakages. Next, we have solved for the crosshand phase using S_1. Then we apply all the solved parameters to all sources. In Figure 2, we compare the model and recovered source linear polarization properties for sources $S_2 \cdots S_5$. We observe a very tiny deviation in the recovered polarisation angle compared to the model. The script, compare_model_corrected.py, is run after using the appropriate figure names to generate this figure.

3.2 Nonlinear analysis

Nonlinear analysis, using quartucal, under each test scenario can be performed by running the bash script named analyse_quartical.sh located inside the quartical folder in each test case.

Comparing the determined leakages using quartical with the input has another issue. We note here that quartical solves a matrix of the following form when use the *leakage* form.

$$\begin{pmatrix} 1 & d_{xi}e^{i\phi_{xi}} \\ d_{yi}e^{i\phi_{yi}} & 1 \end{pmatrix}$$

As mentioned in Kansabanik et al. 2024, the optimizing equation, which uses the Frobenius norm, is degenerate with respect to a unitary matrix. This implies that the leakage matrix determined using quartical has an unknown unitary matrix[1]. multiplied with it, which makes it not comparable directly with the input leakages. To do proper comparison, we take the following approach. The approach has been mentioned in Kansabanik et al. 2024. Here, we mention the procedure for completeness purposes.

Let us assume that the leakage matrix determined by quartical for the reference antenna and a general antenna i are denoted by J_{ref} and J_i respectively. The normalized Jones matrix for each antenna is then given by

$$J_{ref} = H_{ref}U_{ref} \tag{4}$$
$$J_{norm,i} = J_i J_{ref}^{-1} H_{ref} \tag{5}$$

[1] the effect of this multiplication during calibration is discussed in detail later in the high leakage case

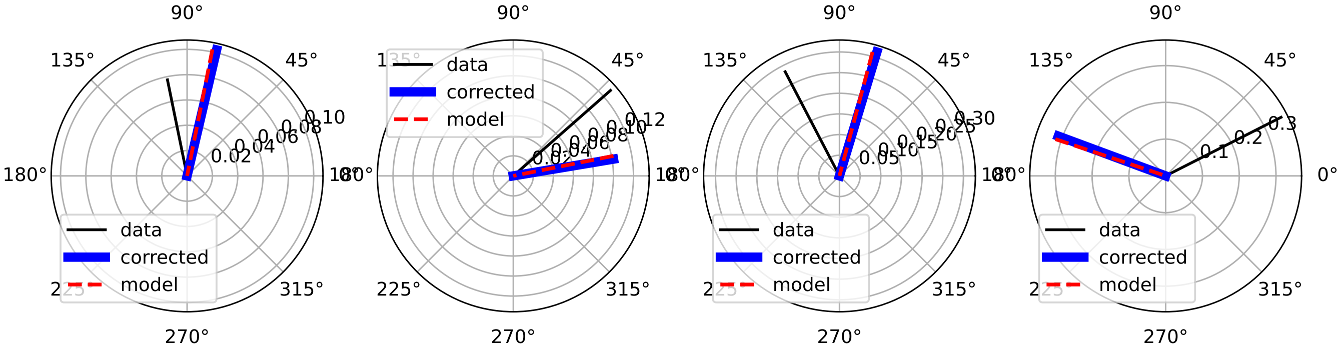


Figure 2: XX

The input $J_i^{leakage}$ is also normalised in the same manner, using the same reference antenna. Then they are compared, and the comparison is shown in Figure 3. This figure can be generated by running the script, compare_quartical_model_leak1.py, located inside the quartical folder. In Figure 4 we show the comparison between the input and recovered source parameters. The agreement is slightly better than what we find with CASA, but not sufficiently to say that CASA has performed poorly.

4 High leakage

The most general case is implemented in a folder named high_leakage_case_general.

4.1 Linear analysis

The CASA analysis is exactly similar and the codes which are run are also named same. In Figure **??**, we show the comparison between the relative input and solved leakages. We find that the solved leakages match exactly with the input leakage, after appropriate normalisation. This is not surprising, considering that the cross-hand visibilities are exactly same for unpolarized source both in the linear and non-linear regime. In Figure 2, we compare the model and recovered source polarization properties for sources $S_2 \cdots S_5$. We observe large deviations in the recovered polarisation properties compared to the model. In fact, the Stokes I is different for the recovered source, in comparison to the input source model. We have verified that the I and V after calibration is same as that in the data, after corruption by the antenna Jones matrices. This demonstrates that CASA does not apply leakage corrections and crosshand corrections to the co-polar visibilities.

4.2 Nonlinear analysis: Quartical

Here, we also follow the strategy described in the linear case. We solve for the leakage matrices, and then use them to determine the crosshand phase matrix. All of the Jones matrices are determined using quartical. The analysis code is again named analyse_quartical.sh, located in a folder named quartical_leakage. The comparison between the leakages after appropriate normalization is shown in Figure 7. This figure can be generated by running the script, compare_quartical_model_leak1.py, located inside the quartical folder. In Figure 8 we show the comparison between the input and recovered source parameters. The agreement is slightly better than what we find with CASA, in the sense that Stokes I is well matched now. However, Stokes V is roughly of the same order as that determined in the CASA calibration. The question is hence, why even when we are using Quartical, which is a non-linear solver, we can not recovering the input parameters.

The answer to this question is quite non-trivial, and has to do the "unitary matrix" degeneracy we mentioned earlier. Remember that we had created a Jones matrix $J_i^{leakage}$ for each antenna. We thought that this matrix will only cause mixing between the Stokes pairs (I,Q), (I,U) and (I,V). However, as we show below, this is not correct.

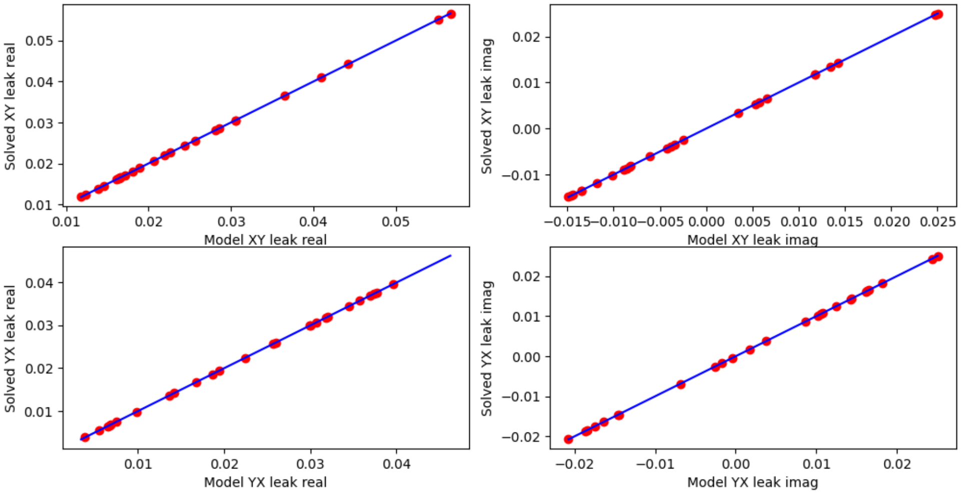


Figure 3: XX

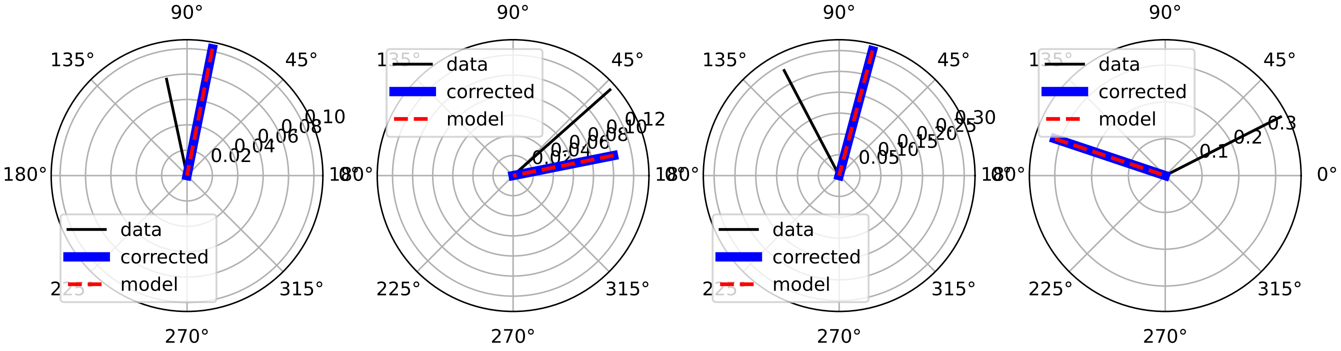


Figure 4: XX

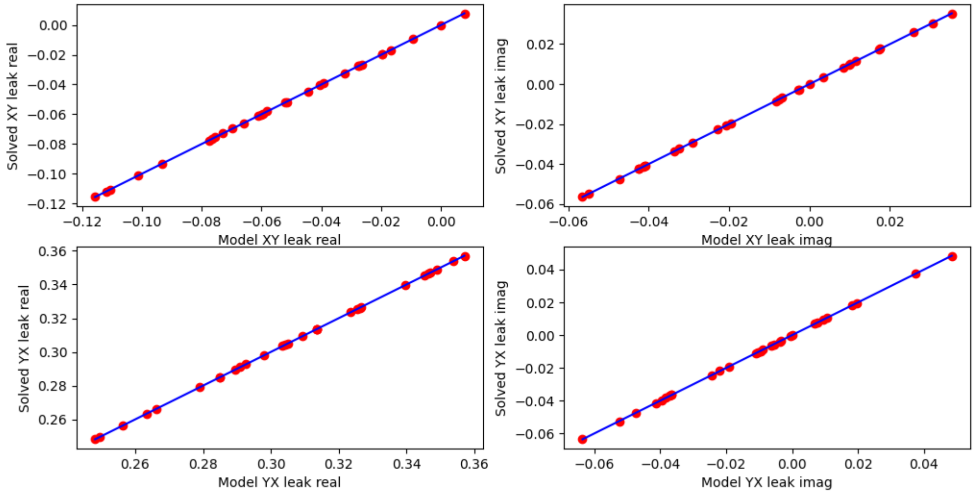
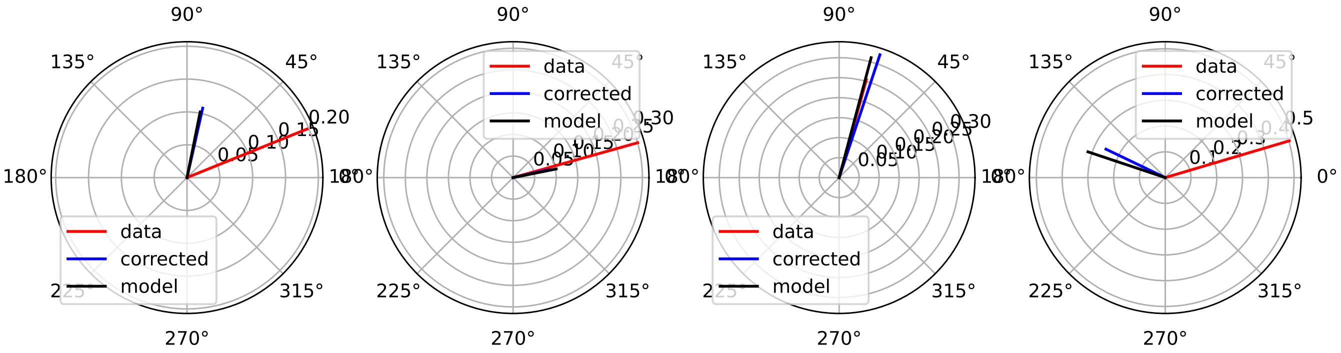
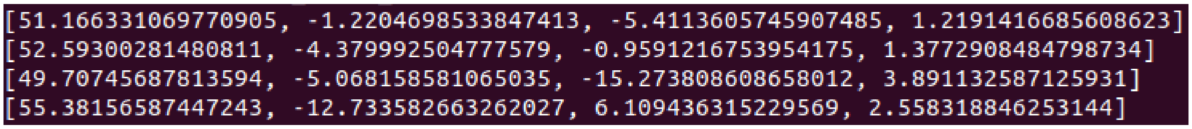


Figure 5: XX





```
[51.166331069770905, -1.2204698533847413, -5.4113605745907485, 1.2191416685608623]
[52.59300281480811, -4.379992504777579, -0.9591216753954175, 1.3772908484798734]
[49.70745687813594, -5.068158581065035, -15.273808608658012, 3.891132587125931]
[55.38156587447243, -12.733582663262027, 6.109436315229569, 2.558318846253144]
```

Figure 6: XX

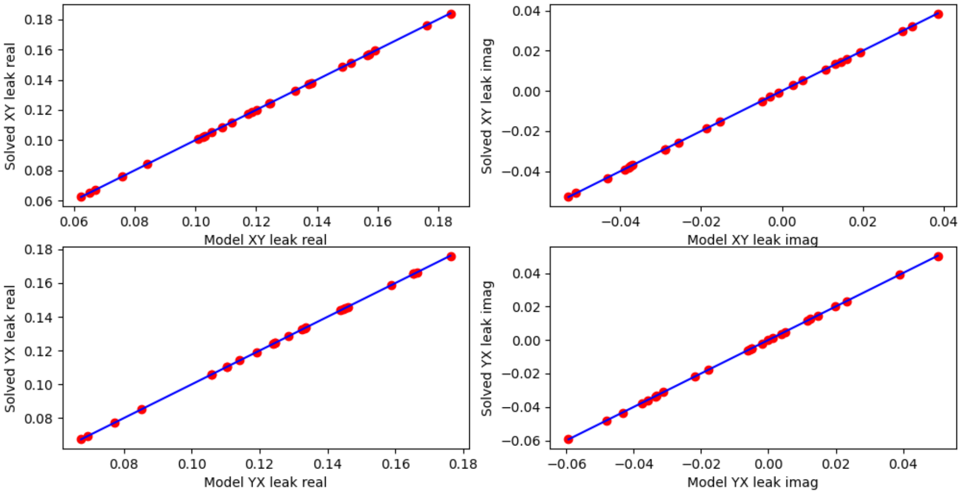
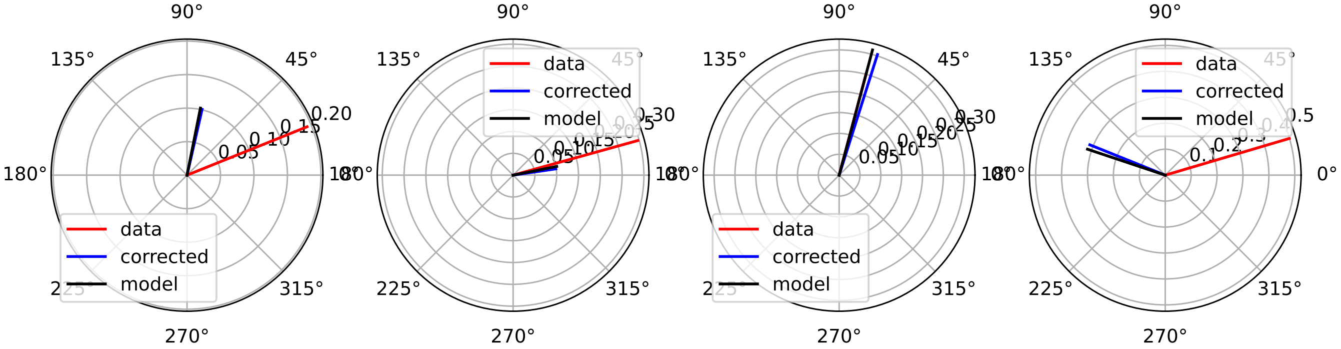
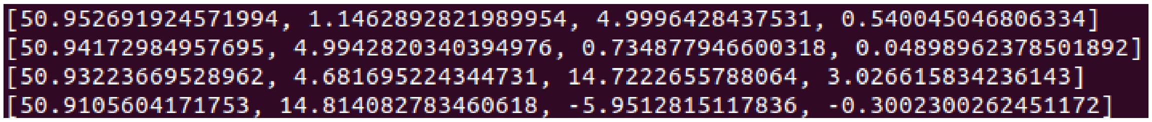


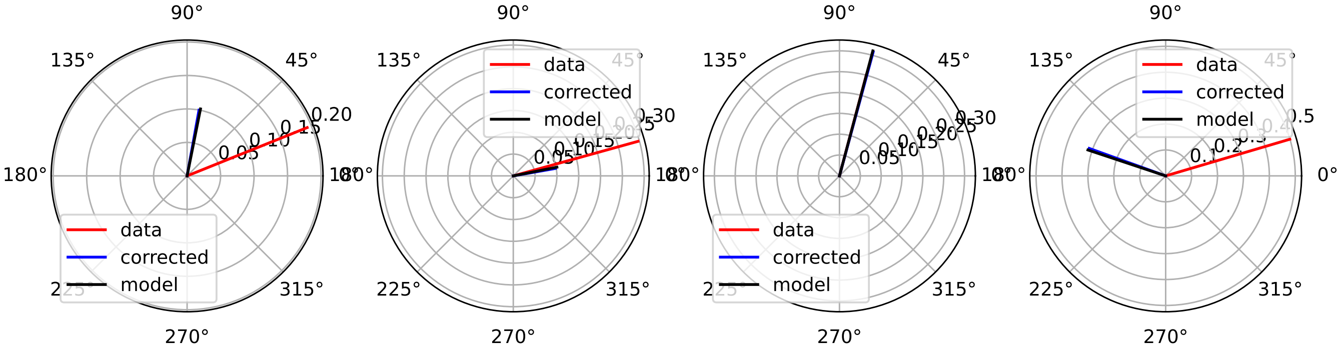
Figure 7: XX

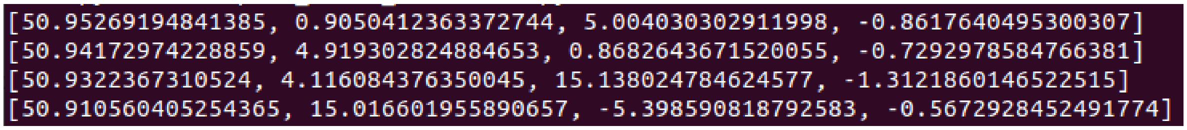




```
[50.952691924571994, 1.1462892821989954, 4.9996428437531, 0.540045046806334]
[50.94172984957695, 4.9942820340394976, 0.734877946600318, 0.04898962378501892]
[50.93223669528962, 4.681695224344731, 14.7222655788064, 3.026615834236143]
[50.9105604171753, 14.814082783460618, -5.9512815117836, -0.3002300262451172]
```

Figure 8: XX





```
[50.95269194841385, 0.9050412363372744, 5.004030302911998, -0.8617640495300307]
[50.94172974228859, 4.919302824884653, 0.8682643671520055, -0.7292978584766381]
[50.9322367310524, 4.116084376350045, 15.138024784624577, -1.3121860146522515]
[50.910560405254365, 15.016601955890657, -5.398590818792583, -0.5672928452491774]
```

Figure 9: XX

The $J_i^{leakage}$ of each antenna can be written as

$$J_i^{leakage} = J_i^{leakage} \left(J_{ref}^{leakage}\right)^{-1} H_{ref} U_{ref} \tag{6}$$

$$J_{ref}^{leakage} = H_{ref} U_{ref} \tag{7}$$

H_{ref}, U_{ref} are Hermitian and unitary matrices respectively. This suggests that $J_i^{lekaage}$ of each antenna i has a common unitary matrix multiplied with it, and the unitary matrix multiplied depends on the jones matrix of the reference antenna. This unitary matrix is will cause rotation of the Stokes vector in the Q, U, V space, even in absence of any crosshand phase term. Hence, solving for just the crosshand phase will not correct for the entire polrotation term. The reason why Stokes I was ok, was because, the non-linear calibration procedure, has uniuqely determined the polconversion matrix and hence Stokes I is ok when we use Quartical. The solution to this issue is to solve for the full general polrotation matrix. This is done in the next subsection. However, we use the antenna leakages determined using Quartical. We first completely remove the effect of the unitary matrix of the reference antenna and store those in a file. The file is named quartical_ant_leaks.txt and can be generated by running get_quartical_leak.py . The created txt file is copied from the quartical folder to the folder named self_crosshand.

4.3 Nonlinear analysis: Determining the full polrotation matrix

For solving the full polrotation matrix, we need to use at least 2 polarised sources. For this purpose, we use S_2 and S_3. The code to solve the 3 angles is named solve_crosshand_phase.py . The phases are applied to the data using the script named generate_corrected_vis.py . by running the script, compare_quartical_model_leak1.py, located inside the quartical folder. In Figure 9 we show the comparison between the input and recovered source parameters. It is evident that all Stokes parameters have been recovered with a much better accuracy. However, in spite of this, we find that Stokes V is still non-zero. This we think is due to numerical errors and convergence parameters chosen in Quartical and our solver used for determining the polrotation matrix.

4.4 Does determing the diagonal gains help

In this section, we try to understand, if it is possible to solve the Stokes V issue, if we were general observers, who had no understanding of how the data was generated. Observers typically does not assume that the diagonal gain is 0. This implies that for quartical, we shal solve for a general matrix using S_1. We also run a gaincal before polcal with S_1. For the purpose of this, scripts with similar names are present in folders named CASA_gaincal and quartical In both cases, we find that the estimated amplitude and phase gains are very similar. We show the determined antenna

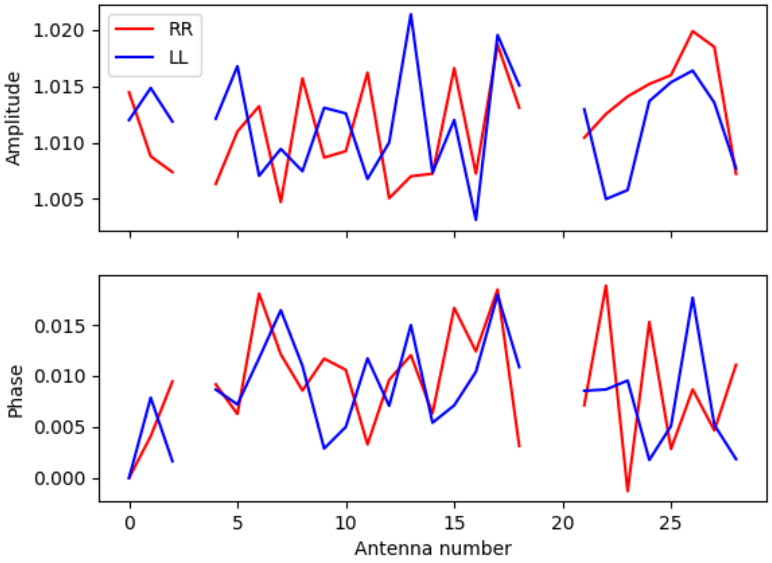
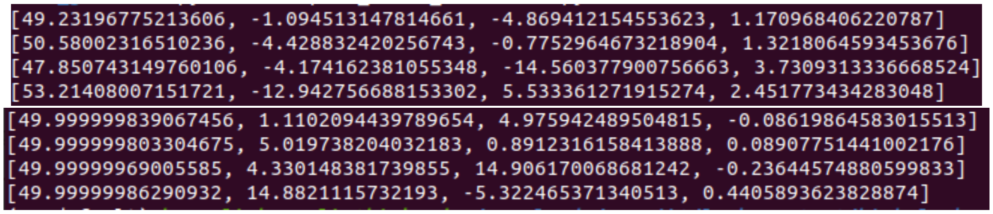


Figure 10: XX



```
[49.23196775213606, -1.094513147814661, -4.869412154553623, 1.170968406220787]
[50.58002316510236, -4.428832420256743, -0.7752964673218904, 1.3218064593453676]
[47.850743149760106, -4.174162381055348, -14.560377900756663, 3.7309313336668524]
[53.21408007151721, -12.942756688153302, 5.533361271915274, 2.451773434283048]
[49.999999839067456, 1.1102094439789654, 4.975942489504815, -0.08619864583015513]
[49.999999803304675, 5.019738204032183, 0.8912316158413888, 0.08907751441002176]
[49.99999969005585, 4.330148381739855, 14.906170068681242, -0.23644574880599833]
[49.99999986290932, 14.8821115732193, -5.322465371340513, 0.4405893623828874]
```

Figure 11: XX

gain terms in Figure 10. The amplitude gains deviation from 1 are about 1–2%. The phase deviations are also of same order. We otain similar numbers when we use CASA gaincal task as well. We repeat the analysis by applying these parameters, and we find that while CASA results have not changed appreciably, for the non-linear case, Stokes V has improved significantly. In Figure 11, we show the recovered polarization properties from CASA analysis, and the procedure we mentioned in Section 4.3. We find that while the non-linear analysis has now recovered all Stokes parameters with an accuracy better than 1%, there are quite large errors in the CASA analysis. Observers can solve the spurious Stokes V issue, by running CASA gaincal for each of the calibrator, and then assuming that the gains are varying with time. The determined gain terms are in reality a function of the source polarization properties and the antenna leakages and hence interpolation schemes based on time-variability will not work.