1 Solutions of Complex Antenna Gains - Algorithms and Simulations

1.1 Visibilities and Antenna Gains - Calibration

The visibilities recorded using an interferometer, are obtained from two point correlations of the signals between various pairs of antennas.

If E_i and E_j represent the signals from two antennas i and j, corresponding to the same spectral channel, then the complex correlation between the two, given by the time average of the complex product of the two signals, forms the recorded visibilities.

$$V_{ij} = \left\langle E_i E_j^* \right\rangle \tag{1}$$

Let g_i and g_j be the independent complex gains for the two antennas i and j. Then

$$V_{ij} = \left\langle g_i E_i^o g_j^* E_j^{o*} \right\rangle \tag{2}$$

where E_i^o and E_j^o are the incident radiation on the two antennas.

$$V_{ij} = g_i g_j^* \left\langle E_i^o E_j^{o*} \right\rangle \tag{3}$$

For a point source at phase tracking centre, $\left\langle E_i^o E_j^{o*} \right\rangle = 1$

$$V_{ij} = g_i g_j^* \tag{4}$$

This is a set of equations for all the complex visibilities, as non-linear functions of the individual antenna gains.

1.2 Iterative Method

This method is based on minimizing χ^2. The equations for the individual gains are obtained by minimizing χ^2 and are solved iteratively, beginning with an initial guess for the values of the gains.

Assuming that the noise on the antenna dependent complex gains are independent,with a gaussian probability density function (this implies that the noise on the real and imaginary parts are independently gaussian random processes), one can estimate g_is by minimizing, with respect to g_is, the function χ^2 given by

$$\chi^2 = \sum_{i<j} \left| V_{ij} - g_i \; g_j^\star \right|^2 w_{ij} \tag{5}$$

Expanding, we get

$$\chi^2 = \sum_{i<j} (V_{ij} - g_i g_j^*)(V_{ij}^* - g_i^* g_j) w_{ij} \tag{6}$$

Differentiating w.r.to g_i^R and g_i^I , equating them to zero, and combining the two equations, we get

$$g_k \;=\; \frac{\sum\limits_{\substack{i \\ i \neq k}} V_{ki} g_k w_{ki}}{\sum\limits_{\substack{i \\ i \neq k}} |g_k|^2 w_{ki}} \tag{7}$$

In the above equation,since the antenna dependent complex gains also appear on the right-hand side of Eq. 7, it has to be solved iteratively starting with some initial guess for g_js or initializing them all to (1,0).

Eq. 7 can be written in the iterative form, along with a damping factor α as:

$$g_k^n \;=\; g_k^{n-1} + \alpha \left[\frac{\sum\limits_{\substack{i \\ i \neq k}} V_{ki} g_k^{n-1} w_{ki}}{\sum\limits_{\substack{i \\ i \neq k}} \left| g_k^{n-1} \right|^2 w_{ki}} - g_k^{n-1} \right] \tag{8}$$

where n is the iteration number and $0 < \alpha < 1$.

This method is a form of χ^2 minimization by steepest descent. This can eb seen from the fact that the new value of the gains after each iteration, is calculated as the sum of the previous estimate of the gains and an increment.

The programs currently in use, at the GMRT, *antsol* and *rantsol*, use this algorithm.

`antsol` : Task to compute the antenna gains from a point source visibility in LTA format. The task writes the following in the output file: antenna gains, the residual visibility (Amp,Phs), the input visibility (Amp,Phs), and the model visibility (Amp,Phs) as function of hour angle (HA).

`rantsol` : Task to compute the antenna gains from a point source visibility in LTA format. Two passes per data record are performed. After the first pass, antennas with amplitude gain less than user defined threshold are marked as bad for the second pass. Antennas which are found to be bad in this manner are assigned a complex gain of (0,0). The task writes the following in the output file: antenna gains, the residual gains (Amp,Phs), the input gains (Amp,Phs), and the model gains (Amp,Phs) as function of hour angle (HA).

These are the standard programs available and are routinely used.

The following sections describe other algorithms that have been implemented and tried, in this project work, to compute solutions for complex antenna gains. These matrix based techniques have been tried as an attempt to find an algorithm, that may be optimal in terms of accuracy as well as computational load and speed.

1.3 Solution by Non Linear Least Squares

Eqn 4 can be iteratively solved by the method of non-linear least squares to obtain estimates of the complex antenna gains. This is a largely over-determined problem. If there are N complex gains, there are 2*N parameters (real and imaginary parts) and N*N-1 equations (real and imaginary parts of N*(N-1)/2 visibilities). Error estimates on these solutions can be obtained from the covariance values for these 2*N parameters.

1.3.1 Non Linear Least Squares

This is a Gradient based χ^2 minimization with the values of the Hessian estimated from first derivatives. This process is decribed below.

Let the model to be fitted be $y = y(x; a)$ where the vector a defines the list of parameters that have to be solved for.

The χ^2 merit function would be

$$\chi^2(a) = \sum_{i=1}^{N} \left[\frac{y_i - y(x_i; a)}{\sigma_i} \right]^2 \tag{9}$$

Differentiate χ^2 w.r.to all the parameters a, and equate to zero, to minimize χ^2

$$\frac{d\chi^2}{da_k} = -2\sum_{i=1}^{N} \frac{[y_i - y(x_i;a)]}{\sigma_i^2}\frac{dy(x_i;a)}{da_k} \qquad k = 1, 2, ..., M \tag{10}$$

Taking an additional partial derivative gives

$$\frac{d^2x^2}{da_k da_l} = 2\sum_{i=1}^{N}\left[\frac{dy(x_i:a)}{da_k}\frac{dy(x_i;a)}{da_l} - [y_i - y(x_i;a)]\frac{d^2y(x_i;a)}{da_l da_k}\right] \tag{11}$$

Defining

$$\beta_k = \frac{-1}{2}\frac{d\chi^2}{da_k} \tag{12}$$

and

$$\alpha_{kl} = \frac{1}{2}\frac{d^2y(x_i;a)}{da_l da_k} \tag{13}$$

we can write

$$\sum_{i=1}^{M} \alpha_{kl}\delta a_l = \beta_k \tag{14}$$

This set of equations is solved for the increments δa_l which when added to the current approximation, gives the next approximation. In the context of least squares, the matrix $[\alpha]$ is the curvature matrix and is one half times the Hessian matrix.

The components a_{kl} of the Hessian matrix (eqn 11) depend both on the first as well as second derivatives of the basis functions with respect to all their parameters. The term multiplying the second derivative in equation (11) is $[y_i - y(x_i;a)]$. For a successful model, thie term should just be the random measurement error of each point. This error can have either sign, and should in general be uncorrelated with the model. Therefore, the second derivative terms tend to cancel out when summed over i. Inclusion of this term can in fact be destabilizing if the model fits badly or is contaminated by outlier points that are unlikely to be offset by compensating points of the other sign. The second derivative term in equation 11 is hence ignored and the second derivative of χ^2 is approximated by the product of the two first derivatives.

$$\alpha_{kl} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}\left[\frac{dy(x_i;a)}{da_k}\frac{dy(x_i;a)}{da_l}\right] \tag{15}$$

This inverse Hessian method is most suited for searching for the minimum χ^2 solution, when the initial guess for that iteration is already near the minima. Far away from the minima, the method of steepest descent, is the fastest.

The steepest descent method corresponds to

$$\delta a_l = constant * \beta_i \tag{16}$$

The Levenberg Marquardt method consists of iterations that vary smoothly between these two methods, depending on how near to the actual solution, the current parameter set is.

1.3.2 Levenberg Marquardt Algorithm

This method consists of iterations that vary smoothly between the extremes of the inverse-Hessian method (Eqn 14) and the steepest descent method (Eqn 16). The latter method is used far from the minimum, switching continuously to the former as the minimum approaches.

Equation 16 has a 'constant' about whose magnitude, there is no information, in the gradient. The gradient/Hessian matrix just gives information about the slope along whose direction, the χ^2 minimum lies.

An estimate of the 'distance' to move along the direction defined by the gradient, can be obtained from the value of χ^2 which gives an estimate of how far off from the best solution, the current solution is. The size of the 'constant' can then be varied accordingly, giving it larger values when the value of the χ^2 is higher and iterating with smaller steps, when near the minimum.

The quantity χ^2 is non-dimensional, but β_k has the dimensions of $1/a_k$, which may be dimensional. The constant of proportionality betweenβ_k and δa_k must therefore have the dimensions of a_k^2. Therefore the quantity that appropriately sets the scale of the constant, is $1/a_{kk}$, the reciprocal of the diagonal element. This scale can still be too big to be taken directly as the step size or the 'constant' in eqn 16, and hence a factor λ is introduced, to control the size of the step. $\lambda >> 1$ will correspond to cut down the step size.

$$\delta a_l = \frac{1}{\lambda \alpha_{ll}} \beta_l \tag{17}$$

or

$$\lambda \alpha_{ll} \delta a_l = \beta_l \tag{18}$$

Equations 14 and 18 can be combined be defining a new matrix α' as follows.

$$\alpha'_{jj} = \alpha_{jj}(1 + \lambda) \tag{19}$$

$$\alpha'_{jk} = \alpha_{jk} \tag{20}$$

Combining Eqns 14 and 18 we get

$$\sum_{i=1}^{M} \alpha'_{kl} \delta a_l = \beta_k \tag{21}$$

When λ' is very large, the matrix α' is forced into being diagonally dominant, and the steepest descent method takes over. Eqn 21 goes to 18. On the other hand, if λ approaches zero,Eqn 21 goes to 14.

This LM method begins with an initial guess, and proceeds according to the following steps.

1. Compute $\chi^2(a)$
2. Pick a modest value for λ, say $\lambda = 0.001$.
3. Solve the linear equations 21 for δa and evaluate $\chi^2(a + \delta a)$
4. If $\chi^2(a + \delta a) \geq \chi^2(a)$, increase λ by a factor of 10 and go back to (3).
5. If $\chi^2(a + \delta a) \leq \chi^2(a)$, decrease λ by a factor of 10, update the trial solution $a < - - a + \delta a$ and go back to (3).

Next is the choice of a suitable stopping criterion. It is usually a waste to iterate till machine precision, since, the parameters can tend to wander around near the χ^2 minimum. The iterations can be stopped once χ^2 stops changing by more than say 0.01 absolutely, or some fractional amount like 10^{-3} of itself (if there are of the order of 1000 data points, contributing to the value of χ^2). Once an acceptable minimum has been reached, λ is set to zero and the covariance matrix $[C]$ of the standard errors in the fitted parameters a is computed as $[C] = [\alpha]^{-1}$

1.3.3 Framework and Implementation

The Numerical Recipes routine 'mrqmin' implements the Levenberg Marquardt method.

This routine was used and simulations were carried out. Once the simulations were validated, the algorithm was applied to real data and the resulting values for the antenna gains, were compared to the values obtained via the standard 'rantsol' results and validated.

This algorithm treats the complex visibilities and gains as pairs of real numbers, corresponding to the real and imaginary parts. Hence for N antennas, and $N * (N-1)/2$ baselines, there are $2 * N$ parameters to be fitted, and $N * (N-1)$ equations. This is a highly over-determined problem, which requires this kind of non-linear least square technique to arrive at the minimum χ^2 solution. The choice of working with real and imaginary parts as separate parameters and measurements, is because the actual measurements of the visibilities are indeed made as separate real and imaginary parts.

The complex visibility is given as

$$V_{ij} = g_i g_j^* \tag{22}$$

Writing V_{ij} and g_i in terms of real and imaginary parts

$$V_{ij} = V'_{ij} + \iota V''_{ij} \tag{23}$$

$$V^*_{ij} = V'_{ij} - \iota V''_{ij} \tag{24}$$

and

$$g_i = g'_i + \iota g''_i \tag{25}$$

$$g^*_i = g'_i - \iota g''_i \tag{26}$$

Equation 22 becomes

$$V'_{ij} = g'_i g'_j + g''_i g''_j \tag{27}$$

$$V''_{ij} = g''_i g'_j - g'_i g''_j \tag{28}$$

The first derivatives that will be required for the calculation of the elements of the Hessian matrix, are obtained as follows.

Differentiating Equation 27 w.r.to g'_k

$$\frac{dV'_{ij}}{dg'_k} = \delta_{ik}g'_j + g'_i\delta_{jk} \tag{29}$$

If $k == i$, $\frac{dV'_{ij}}{dg'_k} = g'_j$ and if $k == j$, $\frac{dV'_{ij}}{dg'_k} = g'_i$.

Similarly,

$$\frac{dV'_{ij}}{dg''_k} = \delta_{ik}g''_j + g''_i\delta_{jk} \tag{30}$$

$$\frac{dV''_{ij}}{dg'_k} = -\delta_{ik}g''_j + g''_i\delta_{jk} \tag{31}$$

$$\frac{dV''_{ij}}{dg''_k} = \delta_{ik}g'_j - g'_i\delta_{jk} \tag{32}$$

The values of the visibility values after each iteration, as well as the values of the first derivatives w.r.to all the parameters, are all calculated using these equations.

The simulations were carried out with the following kinds of data sets.

1. *Actual Gains* - The actual gains were taken as $(1, 0)$, randomly modified to have amplitudes ranging from $(1 - 0.5 * R) < Amp(g_i) < (1 + 0.5 * R)$ where R is a fraction $0.00001 < R < 1$. In addition, the phases of the actual gains were distributed uniform randomly around the unit circle. The random numbers to modify the amplitudes were obtianed from a gaussian distribution, and scaled by R. The random numbers used to scramble the phases, were obtained from a uniform distribution, and multiplied by $2 * \pi$ to get a random phase.

2. *Visibilities* - The visibilities were calculated according to eqns 27 and 28. Additive gaussian noise, scaled by R was added to the computed visibilities.

3. *Initial Guess* - In some runs, the initial guess was fixed at $(1, 0)$. In order to remove any phase bias from the initial guess, these initial guesses were also subject to an amplitude and phase scrambling, as was done to generate the actual gains.

4. $Fixing Parameters$ - The initial guess for the gain of the first antenna was made real, to fix it's phase as a reference.

5. $Masking$ - A mask of zeros and ones was used to suppress some visibilities and/or antennas. This was done to test the stability of the solutions, with varying amounts of information per antenna.

The mrqmin routine of the Numerical Recipes Library is described below.

It requires the following

1. an array $'x'$ which in this case is a list of indices for visibility values.

2. an array $'y'$ with data values - in this case - the visibility values.

3. an array of weights - that represent the rms on each measurement of the elements of $'y'$.

   The value of the uncertainty on each measured value is given by σ_i. If known, this can be given in and the χ^2 value will be computed accordingly and should be of the order of the total number of data points, or slightly less This is because of the high redundancy of the measurements here.(Higher the redundancy, closer the χ^2 will be to N and for no redundancy, $\chi^2 = 0$.) This is the $true$ χ^2 value. ($2N - 1$ parameters are to be solved for, from $N * (N - 1)$ measured values - a highly over-determined problem.) If the σ_i are not known a priori, as can be the case for real data, it can be given in as $'1'$ and this will give a χ^2 value given by the total number of data points multiplied by the square of the average noise in the visibilities.

   If required, by a two pass process - one can obtain a good estimate of the noise in the measured visibilities. In the first pass, σ_i's are set to $'1'$ and the 'noise' is estimated from the obtained minimum χ^2 value. Then this 'noise' is put back into this input array, and a true chi-square estimate is obtained, which helps in formalizing the exit criterion and errors on parameters.

4. an array of initial guesses for all parameters, here - initial guesses for all the gains.

5. arrays to compute and return the covariance and conjugate matrices of the least square parameter fit. The conjugate matrix is the hessian matrix, scaled by a factor of 0.5.

6. a variable to return back the value of χ^2, and another, to hold the step size in the χ^2 minimization iterations.

The mrqmin routine, evaluates a χ^2 and incrementally moves the initial guesses towards the minimum χ^2 solution , based on the Levenberg Marquardt algorithm. The step size - changes adaptively, depending on whether χ^2 increases or decreases.

This whole routine runs in a loop, within which χ^2 is evaluated and compared to the value obtained in the previous iteration.

If χ^2 decreases, it means that the solution is approaching convergence, and hence the step size parameter λ is decreased by a factor of 10. This means that the algorithm is approaching the inverse Hessian method.

If χ^2 increases, it means that the solution is diverging away from the optimal solution, and the step size λ is increased. Also, in this case, the new solution is discarded, and another try is made from the previous solution, with this new increased step size, λ. This corresponds to the algorithm approaching the steepest descent method.

The χ^2 value, when the sigmas have been set to $'1'$, is a relative χ^2 value, and the exit criterion is set to when the χ^2 decreases by less than say 0.1% of itself.

There is an inherent ambiguity in the phases of the gain solutions, since information about the absolute position of the source in the sky is lost once a correlation is performed. Therefore certain referencing methods must be used.

One is to take one antenna as a reference antenna and make the gain of that antenna purely real. This ensures that the gains of all other antennas will have phases corresponding to the phase differences of each of the gains with that of the reference antenna. This is what is done in practice, with real data.

Another method is to obtain the solutions, and then rotate all the gains in the complex plane, such that the phase of a reference antenna matches with that of the corresponding *actual* gain. This is possible only in a simulation.

In the case of a simulation, the method outlined above, is an absolute referencing, based on making equal, the actual and computed gains of ONE antenna. The resulting rotated phases may or may not be the best choice for the other antennas. A third method can be used, which finds an angle by which to rotate all the gains, that minimizes the sum of the differences between each of the actual and computed gains. This is an average based method, and does NOT require the choice of a reference antenna.

This can be done by the following method.

Consider two sets of complex numbers $Y = \{y_i\}$ and $Z = \{z_i\}$, each of whose elements are known to be almost identical in amplitude to the corresponding element of the other set, but differ by an almost constant phase difference θ.

Let Y represent the *Actual gains* and Z represent the computed gains, that have been subject to fixed, arbitrary clockwise rotation by $theta$.

Let each z_i of Z be written in terms of real and imaginary parts as $Z_i = z_i' + \iota Z_i''$

$$\begin{pmatrix} z_i' \\ z_i'' \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} y_i' \\ y_i'' \end{pmatrix} \tag{33}$$

This is assuming that Z has been obtained from Y by a clockwise rotation by an angle θ. Hence once θ is known, Z will have to be rotated anticlockwise by θ.

A χ^2 measure can be written down as follows.

$$\chi^2 = \sum_i \left[(z_i' - (y_i' cos\theta + y_i'' sin\theta))^2 + (z_i'' - (-y_i' sin\theta + y_i'' cos\theta))^2 \right] \tag{34}$$

Expanding, and differentiating χ^2 w.r.to θ , and equating $\frac{\delta\chi^2}{\delta\theta}$ to zero,

$$\theta = tan^{-1} \left(\frac{z_i' y_i'' - z_i'' y_i'}{z_i' y_i' + z_i'' y_i''} \right) \tag{35}$$

Each element of the complex set Z must now be rotated anticlockwise by θ, to align it with Y.

This method might not align all the values of z_i onto y_i, but collectively, they will be the nearest they can get, in a least squares manner. Almost always this is seen to produce a more accurate agreement between the actual and computed complex gains than the absolute referencing described earlier.

1.3.4 Methods of Testing

The algorithm described above was implemented and applied to simulated as well as real data. On simulated data , several checks were made to ensure that the implementation was correct and that the algorithm performed the way it is supposed to.

1. The mrqmin routine returns the value of χ^2 after every iteration. Once χ^2 converges, the value of the standard deviation of the noise that went into the visibilities, can be estimated by dividing the minimum χ^2 by the total number of data points and then taking the square root. This number must match the value of R that represented the fractional gaussian noise that was added to the visibilities. This is a test for simulated data. For real data, to run this check, the mrqmin routine will have to be run in two passes - the first time with all the σ_i values equal to one, and then the resulting estimate of a σ_i from the minimum χ^2 is used in the second pass, and the true χ^2 value is calculated. This should be less than or equal to the total number of data points. In this case of a highly overdetermined system, the true χ^2 should come out to be slightly less than the total number of data points.

2. The diagonal elements of the covariance matrix, give an estimate of the mean squared difference between the actual and computed gains. The average value of the diagonal elements of the covariance matrix should be of the same order as the mean square difference between the actual and recomputed gains.

3. The MS differences between the original and recomputed visibilities, should be of the order of R^2, since the 'noise' that was introduced into the visibillities was independent of the gains or visibilities, and had a variance of R^2.

4. A check on the method of phase referencing for simulated data was performed. The angle by which to rotate the gains of all the antennas was computed by making equal the phases of the actual and computed gains, of ONE antenna. The RMS difference between the two sets of gains was computed. By the second method the angle by which to rotate the gains of all the antennas, was computed by the method described above. The RMS difference between the two sets of gains, was found to be less than the the RMS obtained by the first method of phase referencing.

5. Visual confirmation was also done by looking at polar plots of the actual gains, the initial guesses, and the final converged solutions.

6. To test the stability of these gain solutions, the simulation was run with a few antennas and/or baselines masked.

7. When the algorithm ran on real data, the solutions obtained were compared to

the solutions obtained with the standard tool 'rantsol' . χ^2 values were also compared and cross-checked.

8. On real data the effect of the data mask was checked. It was found that the solutions it produced either way were almost the same, and these were equal to the corresponding solutions and χ^2 values given by 'rantsol'.

1.3.5 Advantages of this approach

The main advantages of this method are the following.

1. This method was found to be reasonably immune to outliers in the visibilities, and was not affected by 'dead' antennas. Those antennas just gave abnormally low gains. This robustness eliminates the need to prune the data values, in between iterations.

2. Since the algorithm takes larger steps far away from the solution, and smaller steps near the minimum χ^2 solution, the total number of iterations is brought down. Hence the algorithm works fast. It escapes the danger of the inverse Hessian method diverging because of a bad initial guess, as well as the tendency of the steepest descent method to take small steps when it would be more optimal to take larger steps.

3. The values of the covariances, have been thoroughly validated and there is no doubt from the simulations, that the values of the covariance matrix can be trusted. This will help when analysing real data, where the stability and accuracy of the gain solution can be easily determined.

4. For real data, in cases where the χ^2 converges properly, a rough estimate of the noise on the visibilities can be obtained. A histogram formed from the differences of the original visibilities and those computed from the new estimates of the gains, can give an idea of the statistical nature of the visibilities. Discrepant visibility points in the data set will appear as outliers, and these represent those visibilities which are corrupted and cannot be fit directly by the used model.

1.4 Solution by Matrix Decomposition

The visibility values for a point source at phase tracking centre, are given as follows.

$$V_{ij} = g_i g_j^* \tag{36}$$

The process of obtaining separate values of g_i for all antennas, can be considered to be a factorization of each complex visibility into two complex gains.

A matrix decomposition like the Singular Value Decomposition, can be used to 'factorize' the visibilities. The SVD transforms a matrix into two column orthonormal matrices, and one diagonal matrix of singular values.

$$SVD([A]) = [U][S][V^T] \tag{37}$$

where U and V are column orthonormal, and S is diagonal. This means that the visibilities can be reconstructed from elements of each column of U and the corresponding column of V, scaled by the corresponding singular value from S. For a hermitian matrix A, the matrices U and V are identical, and the method reduces to a simple matrix eigen value/vector decomposition.

If the complex visibility matrix is diagonalized, the eigen vector corresponding to the largest eigen value, gives the values of the complex gains, after scaling with the largest eigen value.

The matrix of visibilities is Hermitian. The upper triangle consists of the visibilities and the lower triangle, their complex conjugates. The diagonal elements are currently equal to zero since only the cross visibilities are used in the computation of the antenna gains. The values of the diagonal elements can be calculated explicitly in a simulation, or can be estimated from the visibility values themselves. Also, a crude solution to the gains, obtained from one stage of an iterative process, can be used to estimate the diagonal elements of the visibility matrix.

These diagonal elements can be estimated as follows.

$$S_i = \sum_{i \neq j} |V_{ij}|^2 \tag{38}$$

$$k^2 = \sum_i S_i = 2 \sum_{i,j,i<j} |V_{ij}|^2 \tag{39}$$

$$\left(\sum_i g_i^2\right)^2 = k^2 + \sum_i g_i^4 \tag{40}$$

Let $\sum_i g_i^2 = k_1^2$ and $\sum_i g_i^4 = 1/Nk^2$

Then

$$S_i = k_1 g_i^2 - g_i^4 \tag{41}$$

or

$$g_i^4 - k_1 g_i^2 + S_i = 0 \tag{42}$$

This can be solved for g_i^2 and then g_i can be obtained. The diagonal elements of the visibility matrix can then be filled up as $V_{ii} = g_i g_i^*$

This algorithm was implemented and the results obtained were found to be comparable to the non-linear least squares.

1.5 Non Iterative method

This is a non-iterative procedure that works with the complex visibilities and gains in terms of amplitude and phase. It is based on the following formulation.

The complex gains g_i can be written as

$$g = e^{\iota\alpha+\beta} = e^{\beta} e^{\iota\alpha} \tag{43}$$

$$V_{ij} = g_i g_j^* = e^{\beta_i+\beta_j} e^{\iota(\alpha_i-\alpha_j)} \tag{44}$$

$$lnV_{ij} = ln(g_i g_j^*) = \beta_i + \beta_j + \iota(\alpha_i - \alpha_j) \tag{45}$$

From all the real parts of the natural logarithms of the visibilities, we can obtain estimates of the amplitudes of the gains, and from the imaginary parts of the logarithms of the visibilities, we can get values of phase differences.

The real parts of all the terms of Equation 45 are the sums of successive pairs of values of β.

$$\gamma_{ij} = Re(ln|V_{ij}|) = \beta_i + \beta_j \tag{46}$$

These values of γ form an upper triangular matrix of sums of successive pairs of β. The complete matrix is symmetric here, since the γ_i are sums of β_i. Differences of successive rows will give $N-2$ estimates of the differences between each pair of succesive β values. The mean of these $N-2$ estimates of the differences, is computed for each of the $N-1$ differences, and this is what is used in further calculations.

For example,

$$\gamma_{i,j} - \gamma_{i+1,j} = \beta_i - \beta_{i+1} \tag{47}$$

If there are N antennas, there will be $N-1$ such differences. Once these differences are known, if any one value of β is fixed, the other values of β can be evaluated, from where the values of the amplitudes of the gains can be obtained as $Amp(g_i) = e^{\beta}$. There are N unknowns (β)s, and $N-1$ equations in the form of differences. One extra equation, can be obtained from the total sum of the real parts of the logarithm of the visibilities ($\sum_{i<j}\gamma_{ij}$).

From Equation 46 , and for $N*(N-1)/2$ values of γ this sum comes out to be

$$Sum = \sum_{i<j}\gamma_{ij} = (N-1)\sum_i \beta_i \tag{48}$$

Hence the last equation that would be required is given by

$$\sum_i \beta_i = Sum/(N-1) \tag{49}$$

The value of β_1 can now be calculated as follows.

$$\beta_1 = \frac{\sum_i \beta_i + \sum_{i=1}^{N-1}[(\beta_{N-i} - \beta_{N-i+1}) * i]}{N} \tag{50}$$

From this value of β_1 all the other β_is can be calculated, and the corresponding amplitudes of the gains can be obtained.

For the phases, equation 45 shows that the imaginary parts of the logarithm of the visibilities, are directly the phase differences, of the gains forming each V_{ij}. Successive row differences of the alphas are now similarly calculated, and making $\alpha_1 = 0$, all the other values of α_i can be determined, and these are the phases of all the gains, referenced to the first antenna.

This method can also be supplemented by a Linear Least Squares method of solution, on the logarithms of the visibilities and gains. In the logarithm of the visibilities, the gains add, and can be separately be solved linearly, by minimizing a measure of χ^2 as the difference between the logarithm of the visibility and the sums of the logarithms of the two gains contributing to that visibility. Values of β as previously defined, can be determined from the values of γ_{ij}.

$$lnV_{ij} = ln(g_i g_j^*) = \beta_i + \beta_j + \iota(\alpha_i - \alpha_j) \tag{51}$$

Let $\gamma_{ij} = Re(ln|V_{ij}|) = \beta_i + \beta_j$
and $\epsilon_{ij} = Im(ln|V_{ij}|) = \alpha_i - \alpha_j$

χ^2 for the γ_{ij} can be given as

$$\chi^2_\beta = \sum_{i<j}[w_{ij}](\gamma_{ij} - \beta_i - \beta_j)^2 \tag{52}$$

where w_{ij} is the weight and is given by the amplitude of the visibility V_{ij}. Lower visibility values will be given less weight.

Minimizing this measure of χ^2, the system of linear equations to be solved for the values of β_l are as follows.

$$\beta_l \sum_{j\neq l} |V_{lj}|^2 + \sum_{j\neq l} \beta_j |V_{lj}|^2 = \sum_j \gamma_{lj} |V_{lj}|^2 \tag{53}$$

With the l index representing rows of a matrix, this can be written in matrix form (normal equations) and solved by any linear least squares technique. In this study it has been implemented using the LU decomposition and back-substitution routines provided in the Numerical Recipes library.

This can similarly be done for the values of α_i. The system of linear equations here, comes out as follows.

$$\alpha_l \sum_{j\neq l} |V_{lj}|^2 - \sum_{j\neq l} \alpha_j |V_{lj}|^2 = \sum_j \epsilon_{lj} |V_{lj}|^2 \tag{54}$$

This as such will be a singular matrix, and can be made non singular by dropping out the first row and column of the matrix. This is equivalent to setting the phase of the first antenna to zero and fixing it.

There will be a 2π ambigiuty in the phases of the solutions at any stage. Hence, a set of phase angles α_i^t can be chosen and subtracted from the actual phases, to bring all the phases near zero. These values of α_i^t can be obtained by the first method described above, to obtain solutions for the phases. These phases are then subtracted out from the phases in the data. This is only to bring all the phases between $-\pi/2$ and $+\pi/2$, to ensure that there is no 2π ambiguity in the phase differences.
This is done by supplying the visibilities with a phase difference corresponding to that of two α_i^ts. The solutions for alphas are then found from the solution of Eqn 54, and the phases α_i^t are added to the solution to take them back to their original positions.

This algorithm was implemented and tested on simulated as well as real data. Its performance was compared to that of the Non Linear Least Squares methodas well as the Matrix Decomposition method.

The value of χ^2 obtained by this method, for simulated data agrees with the other two methods. However, for real data, or for simulated data with large amounts of noise, and where the gains of the antennas have a large variance, the performance lags behind the other two methods.