StefCal vs. Classical antsol: A critique

S. Bhatnagar

January 17, 2013

Abstract

At more than one occasion recently, it was brought to my attention that a fundamentally new and faster algorithm to solve for antenna gains, called StefCal, has been invented. Since, if true, this is good and important advancement in radio astronomical post processing, I decided to look into the StefCal algorithm and the differences, if any, between it and the classical algorithm used for over three decades called "antsol". This report documents that investigation. My conclusion is that StefCal is not a new algorithm or approach and is in fact identical to the algorithm used in most widely used software packages for radio astronomical calibration and imaging.

1 StefCal

The formulation of the problem is given on Slide 26 of this presentation[1].

Find G to minimize

$$D_{pq} - G_p M_{pq} G_q^H \tag{1}$$

From the structure of this equation, I gather that p and q refers to the antenna IDs and pq corresponds to the baseline index between antenna p and q. G then is the antenna based complex gain (a 2×2 matrix). D_{pq} and M_{pq} are the observed data and the model data. As given in the slides for the presentation, minimizing the χ^2 gives the update direction as

$$G_p = \sum_p D_q Y_{pq}^H \left(\sum_q Y_q Y_q^H \right)^{-1} \tag{2}$$

where $Y_p = M_{pq} G_{pq}^H(0)$.

There are several inconsistencies in this expression (as written on the slide):

1. Definition of Y_p involves only one sub-script, but is used in the equation with 1 and 2 subscripts

[1]http://www.astron.nl/AACal2012/documents/08_Salvini_StefCal.pdf

2. The first summation over p is also probably incorrect. Should be over q (can be also concluded from a physical understanding of the update equation).

3. G_{pq} is undefined

4. D with a single sub-script does not make sense.

I did a quick-n-dirty derivation and my feeling is that the expression should be written as

$$G_p = \sum_q D_{pq}(M_{pq}G_q^H(0))^H \left[\sum_q \left(M_{pq}G_q^H(0)\right)\left(M_{pq}G_q^H(0)\right)^H\right]^{-1} \tag{3}$$

Also, since $G_q(0)$ refers to the "value from the previous solution", the equation can be more clearly written as

$$G_p^i = \sum_{q,q\neq p} D_{pq}(M_{pq}G_q^{i-1^H})^H \left[\sum_{q,q\neq p} \left(M_{pq}G_q^{i-1^H}\right)\left(M_{pq}G_q^{i-1^H}\right)^H\right]^{-1} \tag{4}$$

where i represents the iteration number.

The "Rinse & repeat until it converges", IMO, is non-scientific speak for "Eq.4 is the update-direction which should be used in an iterative loop to update G_p".

2 Classical "antsol"

The classical antsol solves for direction-independent gain G_p. Just to cut-short possible discussions, in case StefCal is thought to be somehow solving for *direction-dependent* gains, lets me say briefly that that is not true (can be argued from first principles).

Eq.1 is written

$$X_{pq} - G_q G_q^* \tag{5}$$

where $X_{pq} = D_{pq}M_{pq}^{-1}$. These are the "nominal point source visibilities". It's use is fundamental and a crucial step to allow solutions at longer solution intervals rather than solve for each time-stamp and then average the solutions (and also the reason for calling the basic "antsol" algorithm with a more generic term "SelfCal").

Setting up the χ^2 minimization problem, and also using the data weights W_{pq}, one gets the update direction, for the diagonal element of G as

$$G_p^i = \frac{\sum_{q,q\neq p} X_{pq}G_q^{i-1}W_{pq}}{\sum_{q,q\neq p} G_q^{i-1}G_q^{i-1^*}W_{pq}} \tag{6}$$

Detailed derivation (and some physical interpretation of the equations) is in here (Bhatnagar (1998)). The above equation is Eq. 8 in this note[2]. Eq. 9 gives the full iterative

[2] I was very proud of myself when I derived these expression from first principle, only later to realize

algorithm. A robust version of this algorithm also exists (one which is less sensitive to bad data than the original algorithm, has been in use at GMRT for more than 15 years at this point. And now is also the default algorithm used in AIPS since 2003! A vector/matrix version of this is what is in use in CASA since 2002).

Statements on Slide 34 of their presentation is due to the fact that the value of α is hard to compute (involves the Jacobian and second derivatives) and is typically put-in by hand (a good physical interpretation and the heuristics that works is also given, IMO quite nicely in the original paper on MEM by Cornwell!).

This expression is fundamentally not any different from Eq. 4 when written for the 2×2 matrix G. This is also the formulation used in CASA (and, I think, effectively the same expression in AIPS).

3 Difference between StefCal and Classical antsol?

I think there is really no fundamental difference between the two. It comes as a surprise that the algorithm used in BBS and MeqTrees all along was the highly non-optimal one, while the more optimal algorithm was described in the 1980s and a robust version of it in the 90s.

A careful look suggests that pre-StefCal relied on blind application of the LM minimization algorithm – the one implemented in CASACore. That implementation is indeed horrible for problems where the first derivative can be analytically written. I hit this problem when wanting to use that for Pointing SelfCal, and tried to suggest improvements – without success of course. In the end, I wrote a less generic version of iterations myself. The update direction computation there may involve matrix inversion (and in CASA implementation probably also involves AutoDerivatives – which can be expensive for no added advantage, except that you don't have to do even the simple math.). That's an overkill if simple Steepest Descent algorithm would suffice since the update direction can be expressed as an analytical expression. That Steepest Descent is sufficient is obvious. This is probably also the conclusion by the authors of StefCal (Slide 27).

"In the stefcal update step calculation, larger solution intervals are just an extra summation" on Slide 32 probably is a realization that you can coherently average $D_{pq}M^H_{pq}$ prior to solving on the solution interval time scales (but can't average D_{pq} alone if there is significant flux away from the phase center). Since these intervals are quite different for the diagonal and off-diagonal elements, in practice, the scalar version of the expression will be probably required anyway.

A joint solver (simultaneously solving for all the elements of G) is possible - but the original formulation of the problem only needs to be re-cast in vector/matrix form

that Cornwell's paper on SelfCal (Cornwell & Wilkinson 1981) and Thompson & Daddario (1982) papers had the exact same expression – derived based on, what I still feel, were rather elegant **physical** arguments. I did not then push this as a re-invention and just put it up as a technical report. The version of the algorithm now used at GMRT and since 2003 in AIPS as well, is a robust form it, which in fact has something original :)

and does not require invention of a new algorithm (though one has to be careful - some the statements in these slides suggests that proper understanding of those finer points is probably still missing).

Finally, the timing comparison on various slides and on Slide 24 – they appear to be done with MeqTrees only. All this may mean is that MeqTrees had an incorrect (from the performance point of view) implementation all along! In which case a comparison using AIPS/CASA/ASKAPSoft implementation will be probably illuminating.

References

Bhatnagar, S. 1998, Computation Of Antenna Dependent Complex Gains, Tech. Rep. No. R00l72, National Centre for Radio Astrophysics, Pune

Cornwell, T. J. & Wilkinson, P. N. 1981, MNRAS, 196, 1067

Thompson, A. R. & Daddario, L. R. 1982, Radio Science, 17, 357