Improving the Scalability of Calibration

Stef Salvini
Stef.salvini@oerc.ox.ac.uk

(+ S.Winjholds, O.Smirnov, K.Zarb-Adami, B.Mort, F.Dulwich)

Content

- What is StEfCal
- Application to Antenna Calibration
- Applications to Selfcal
  - WSRT
  - LOFAR

StEfCal in a Nutshell

- StEfCal
  - Oleg Smirnov's nickname (not mine!!!)
  - Statistical Efficient Calibration (thanks to Stefan W!!)
- Stefcal is
  - A fast algorithm for **specific** optimisation problems
  - It relies on the nature of these problems
  - It improves over existing methods for these problems
- Stefcal is **not**
  - A general optimisation method
    - No replacement to LM
  - A general solver for RA outside its range of applicability
  - A substitute for selfcal
    - Though it seems to make it more efficient

OXFORD e-Research CENTRE

Algorithm

- Some of the material already presented in December at AAVP
  - Revised and extended since
  - Maths foundations much elucidated
- $O(N^2)$ floating-point operations throughout
- L_2 (least-squares) minimisation
  - For minimizing $|| M – G D GH ||_F$
  - M: model; D: data; G: diagonal or block-diagonal
  - Distance between model sky and calibrated observation
- Accuracy and robustness
- In particular w.r.t. Incomplete visibilities
  - Missing baselines
  - Partial cross-correlation
- Limited dependency on the model sky complexity

OXFORD e-Research CENTRE

The L_2 step

- Levenberg-Marquardt etc. very expensive
- The new algorithm seeks for the zeros of the norm of the gradient of the χ^2 of the data (*D*) – model sky (*M*)

$$\underline{\nabla} \| D - G^H M G \|_F^2$$

- Where *G* (complex gains) is
  - Diagonal complex when polarisations are not coupled
  - 2x2 block diagonal (one block per antenna) when polarisations are uncoupled
- Same formalism used for both cases
- Number of operations: $O(N_2)$

OXFORD e-Research CENTRE

Two Algorithms in one?

- It can minimise
  - Difference between visibilities
    - General case: many sources
  - difference between dominant eigenspaces of Model and observed visibilities
    - few bright sources: better stability, faster convergence
- Incorporates
  - Good termination criterion (norm of gradient)
  - Stopping criteria
    - Too slow convergence
    - Unable to improve
    - Good termination
  - Better than LM, Interior point, etc in all cases examined

Antenna calibration

- From the measurement equation

$$V^{obs} = \Gamma \cdot V \cdot \Gamma^H$$

- Where
  - Complex gain Γ is diagonal

$$\Gamma = G \cdot \Phi \qquad \Phi = \mathrm{diag}(e^{i\phi_j})$$

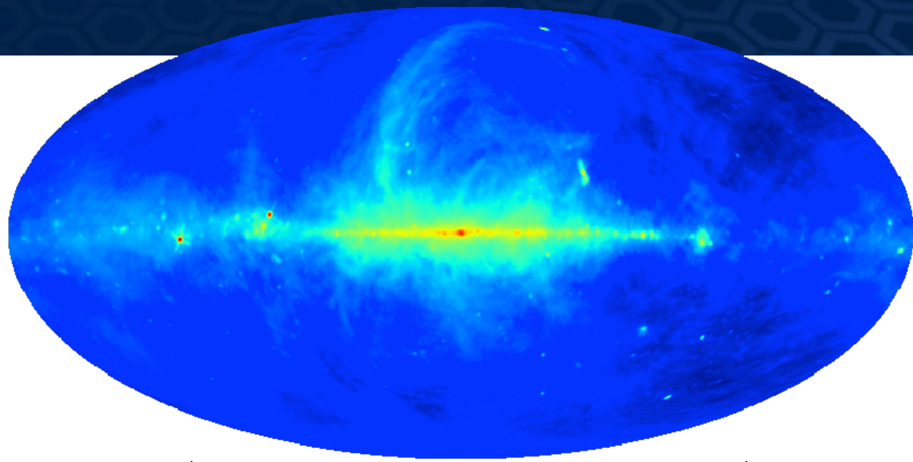
- Hence
  - Error of phases only, unitary transformation: easy problem
  - Error of gains: difficult problem – it “scrambles” the eigenvalues

OXFORD e-Research CENTRE

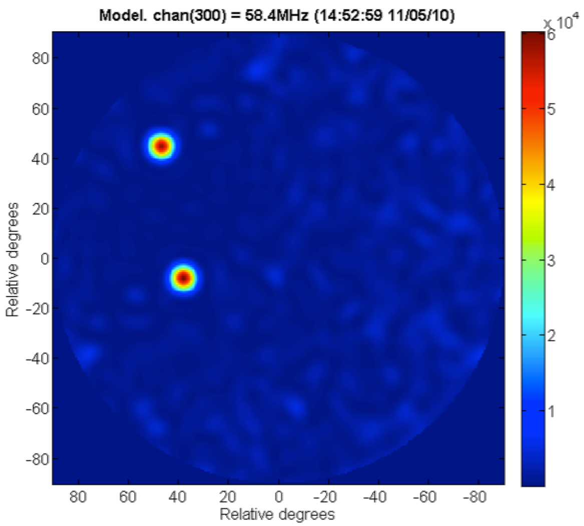
Chilbolton LBA LOFAR Station

- Chilbolton LBA LOFAR station data
  - Thanks to Griffin Foster (OU)!
- Channel 300: 58.4 MHz
  - Other channels also available
  - Sequence of snapshots
  - Observations spaced by ~520 seconds
- Model sky of increasing complexity
  - 2 sources
  - 500 sources
  - 5,000 sources

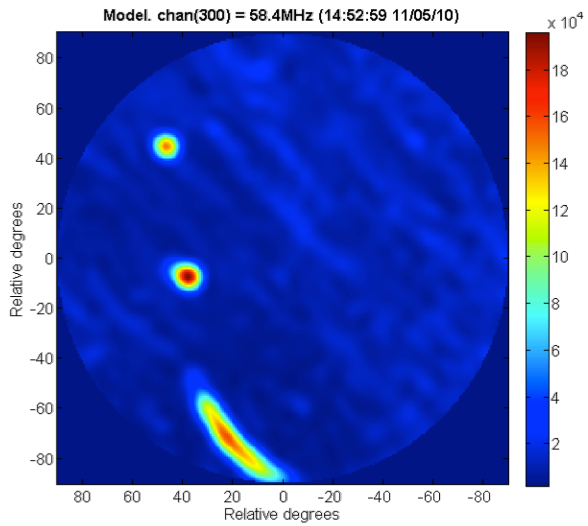
Model Sky



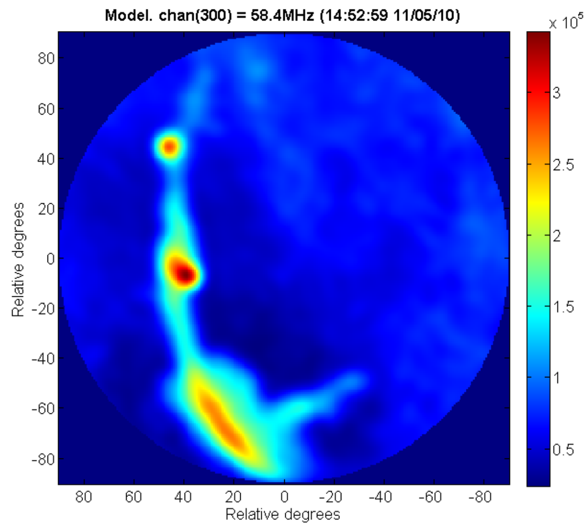
2 sources

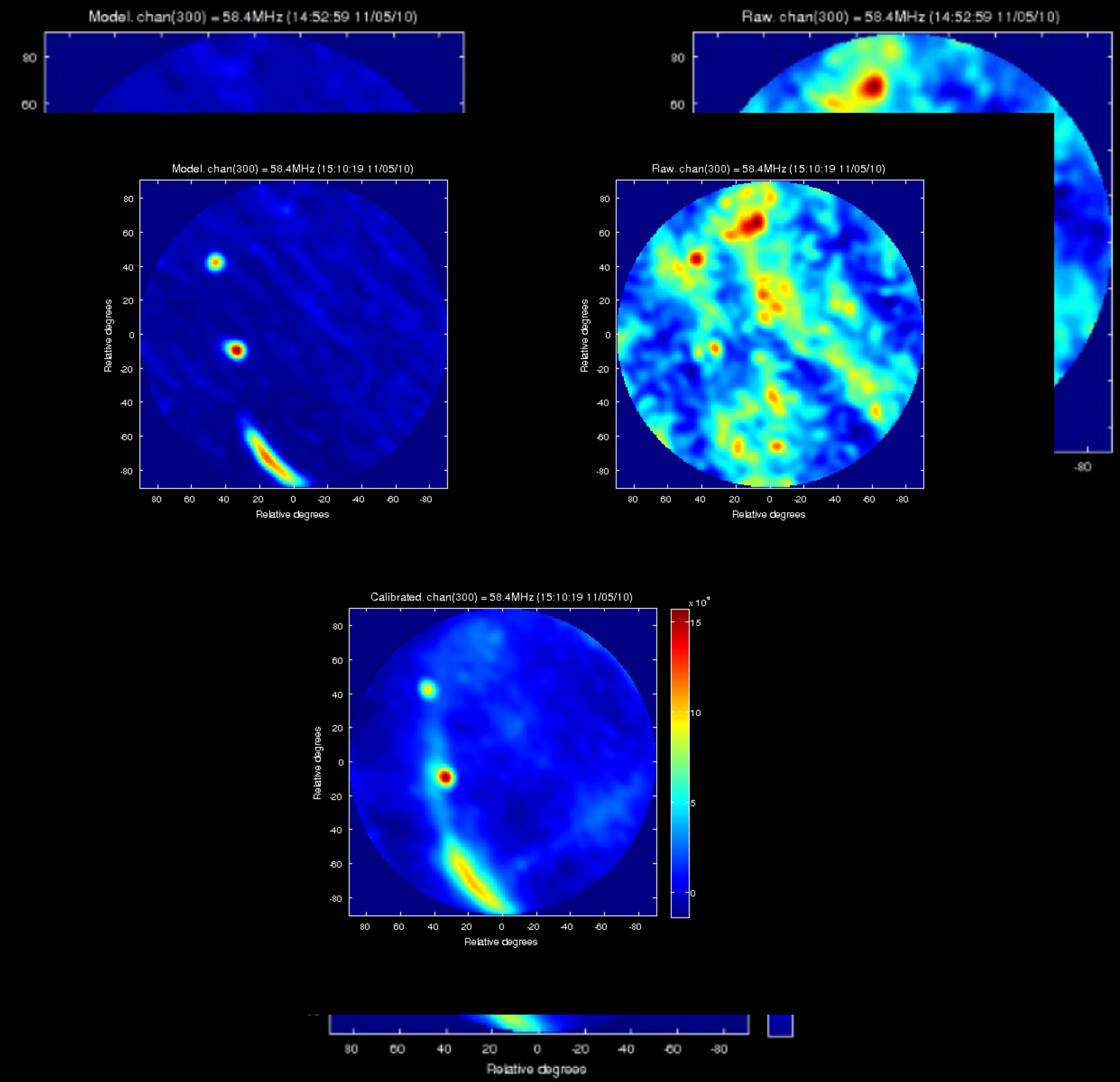


500 sources



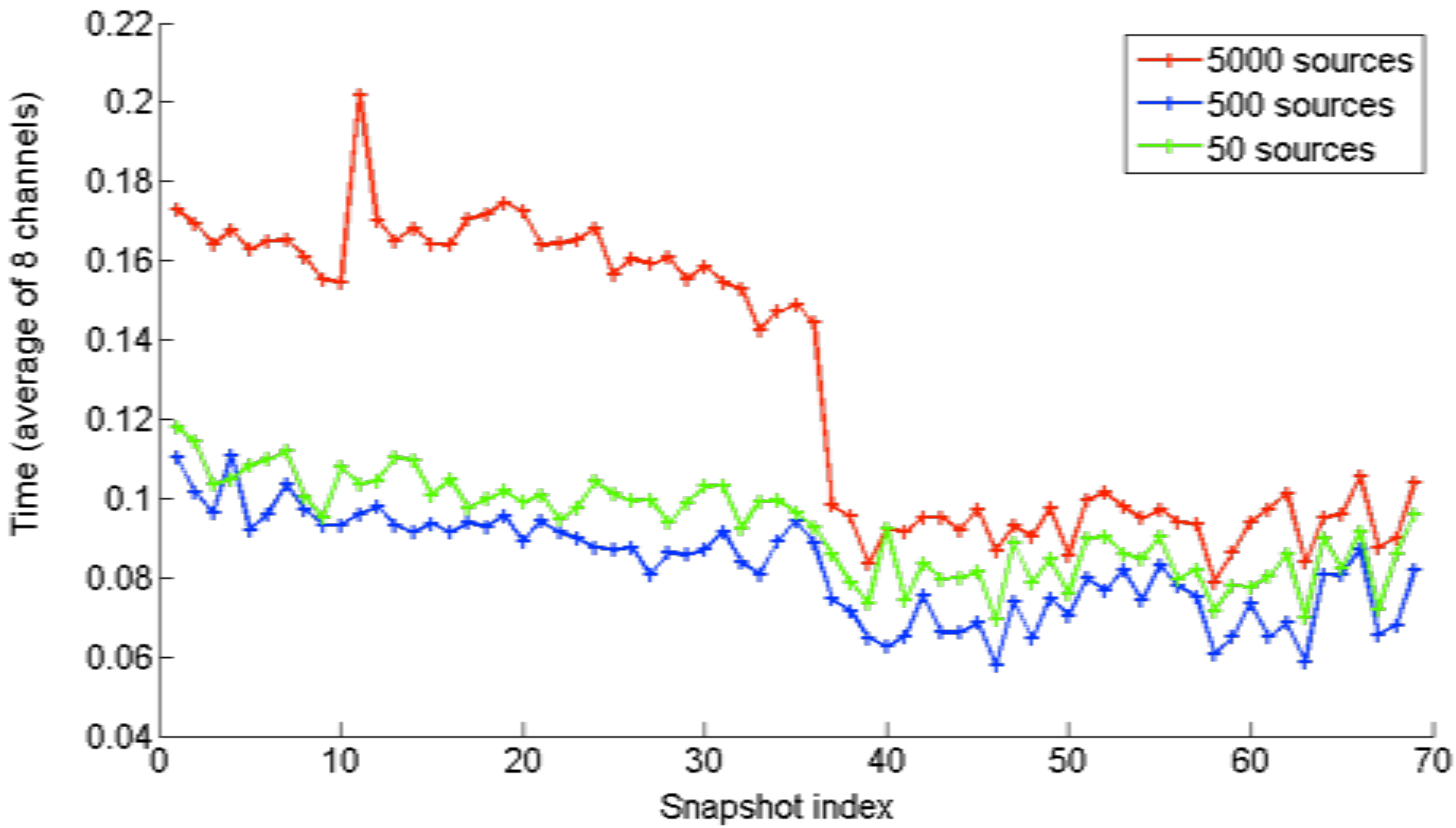
5000 sources





Model. chan(300) = 58.4MHz (14:52:59 11/05/10)
Raw. chan(300) = 58.4MHz (14:52:59 11/05/10)
Model. chan(300) = 58.4MHz (15:10:19 11/05/10)
Raw. chan(300) = 58.4MHz (15:10:19 11/05/10)
Calibrated. chan(300) = 58.4MHz (15:10:19 11/05/10)
Relative degrees
Relative degrees

Timing and performance



OXFORD e-Research CENTRE

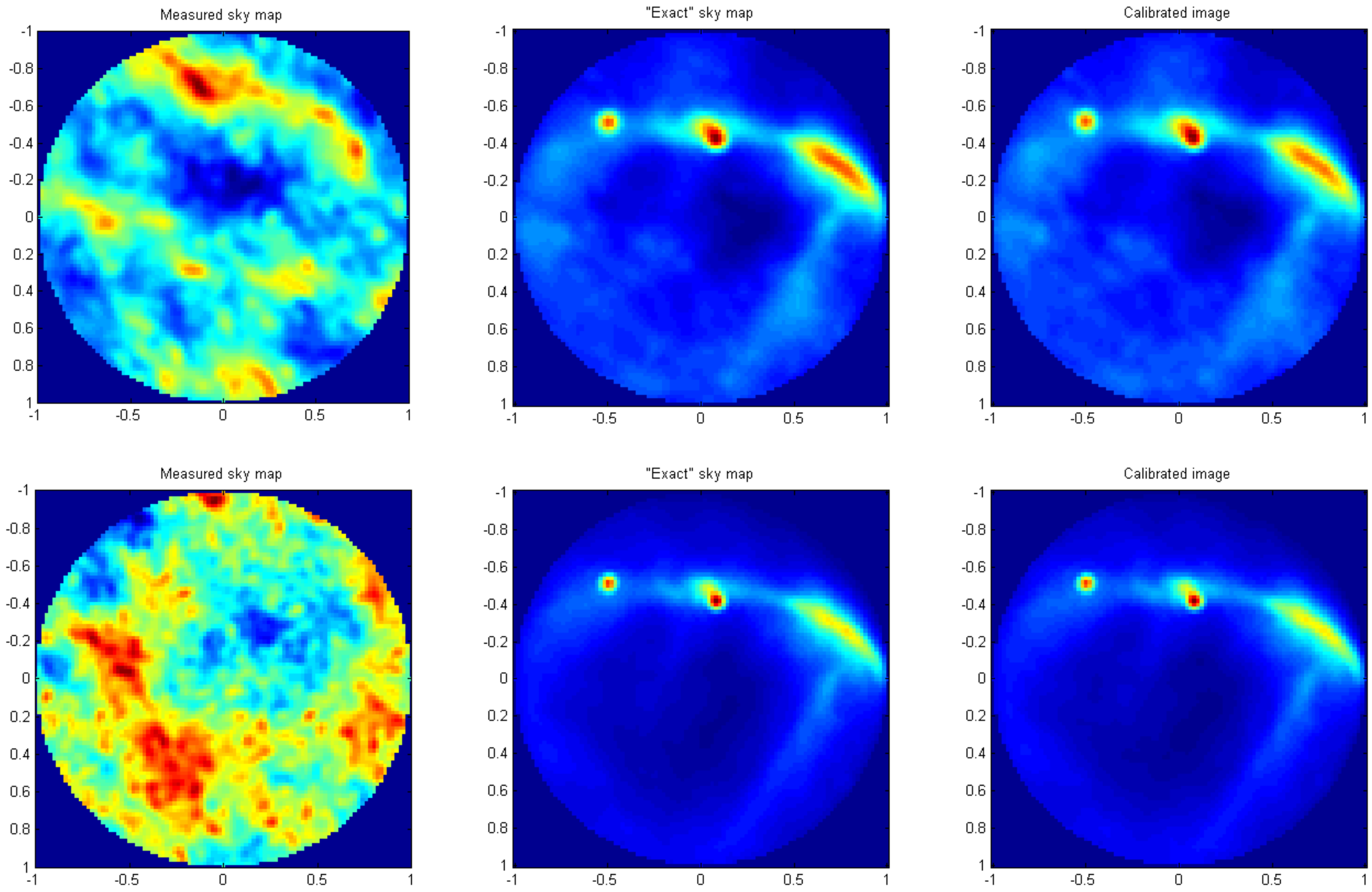
Simulated Sky

- Antennas
  - STD of gains errors ~ 50%
  - STD of phase errors: ~ 2 π rad
- Noise:
  - equivalent to 150 K
    - Diagonal elements of noise: $V_{ii} = kT_i = \sigma_i^2$
    - Off-diagonal elements of noise: $V_{ij} = G\left(0, \sqrt{\frac{\sigma_i \sigma_j}{M}}\right)$
      G is Gaussian random variate, with M the number of integration points
- Number of integration points M = 1,000,000
  - Corresponding to sampling rate 1 GHz, channelised into 1,000 channels, integrated for 1 second

OXFORD e-Research CENTRE

Simulated Sky

	Observed	Exact	Calibrated
96 antennas (LOFAR)			
351 antennas (~ SuperTerp)			



Some performance figures

N. Antennas	Stefan W		Stef S	
	Time (sec)	Normwise error in G	Time (sec)	Normwise error in G
96 (LOFAR)	0.403	0.204	0.015	0.240
351 (~ SuperTerp)	11.58	0.110	0.058	0.103
1,000 (~ SKA1 station)	273.74	0.069	0.381	0.034

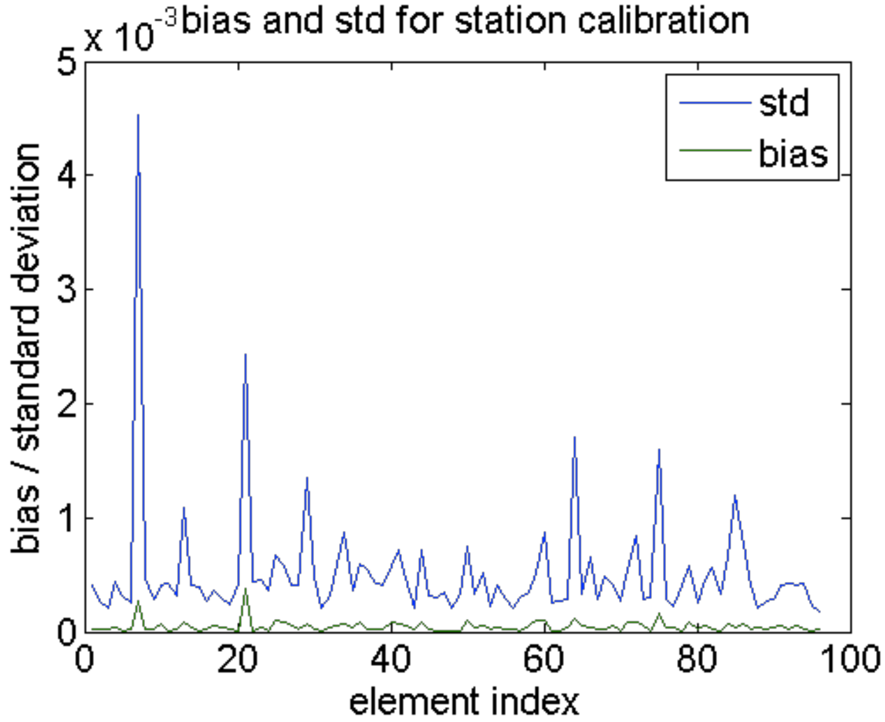
- Simulated sky (GSM – 25,000 sources) + receiver noise
- 200 sources used for calibration
- MATLAB code
- My own laptop (Intel Core 2 i7, 2.0 GHz, Windows

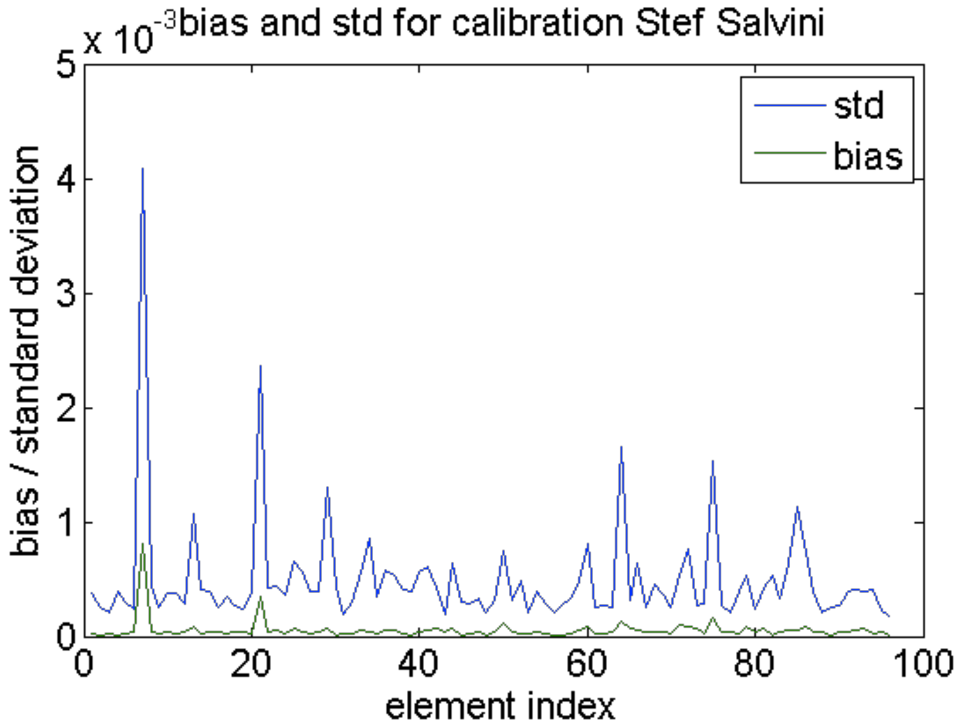
Some more performance figures

N. Antennas	Stefan W		Stef S	
	Time (sec)	Normwise error in G	Time (sec)	Normwise error in G
96 (LOFAR)	0.243	0.169	0.026	0.197
351 (~ SuperTerp)	10.13	0.076	0.054	0.094
1,000 (~ SKA1 station)	239.85	0.048	0.400	0.033

- Same as previous table
- Reduced baselines: (>= 35% of maximum baselines)

Bias & STD compared to Stefan W





OXFORD e-Research CENTRE

Stefcal: Adapting To Selfcal

- The same math should (in principle) work for the interferometer calibration case
  - i.e. use a prior sky model (LSM), and solve for per-station gains
- Main difference is, everything is a function of frequency and time
  - as opposed to single snapshot
- Quick-and-dirty Python implementation now available in MeqTrees

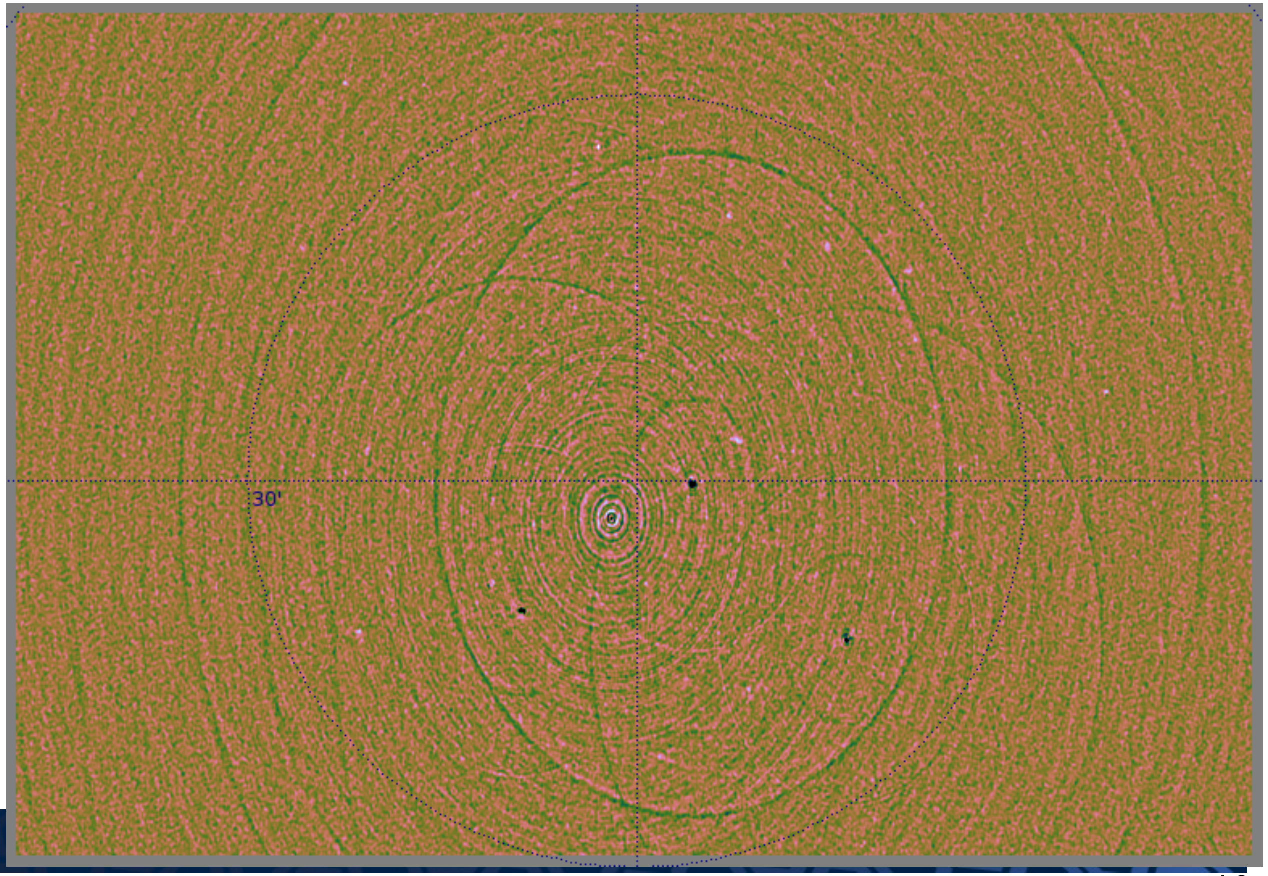
Stefcal with WSRT

- Using 3C147 as a test case
- ~1500 timeslots, 28 channels, 74 baselines
- Runtime **~1m40s** (of which ~half in the solver)
- Compare to MeqTrees+LSQ selfcal: **~10m**
  - ...or just to regenerate the residuals (or corrected data) using prior LSQ solutions: ~**3m**
- Faster to recalibrate than to load solutions!

OXFORD e-Research CENTRE

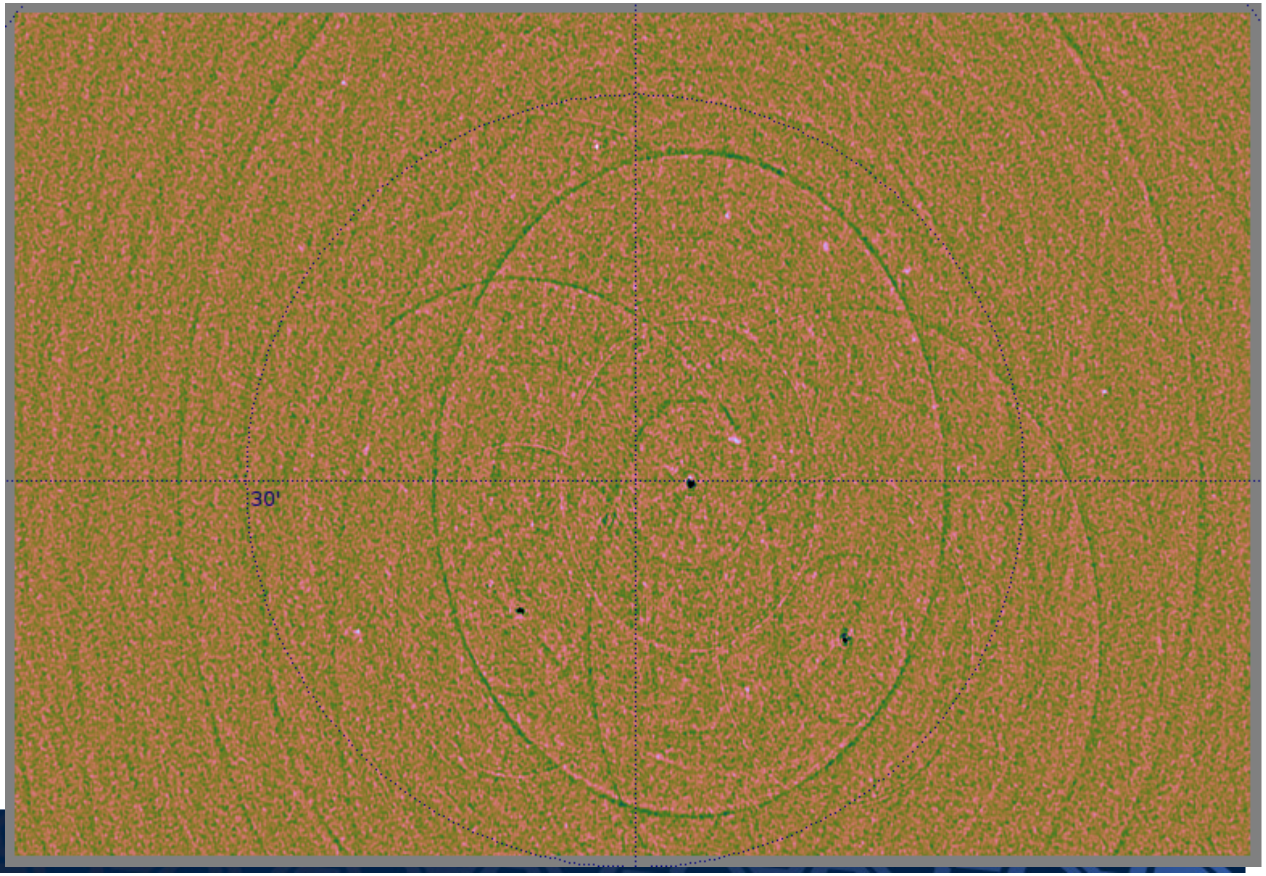
But are the results identical?

- In a nutshell: YES
- Same residuals (to all intents and purposes)
- Remaining artefacts (both selfcal and stefcal):
  - (DDEs)
  - interferometer errors
- MeqTrees can solve for the latter in a separate pass, in **3-4 m**



Per-interferometer errors

- Extended stefcal to solve for these too
- Runtime: **~0s**
- Integrated into the regular solve loop at virtually zero cost
- (strictly speaking, need a second stefcal pass to apply)



OXFORD e-Research CENTRE

Full-pol Stefcal

- Original algorithm formulated for a “scalar” case

$$D_{pq} = G_p M_{pq} G_q^H$$

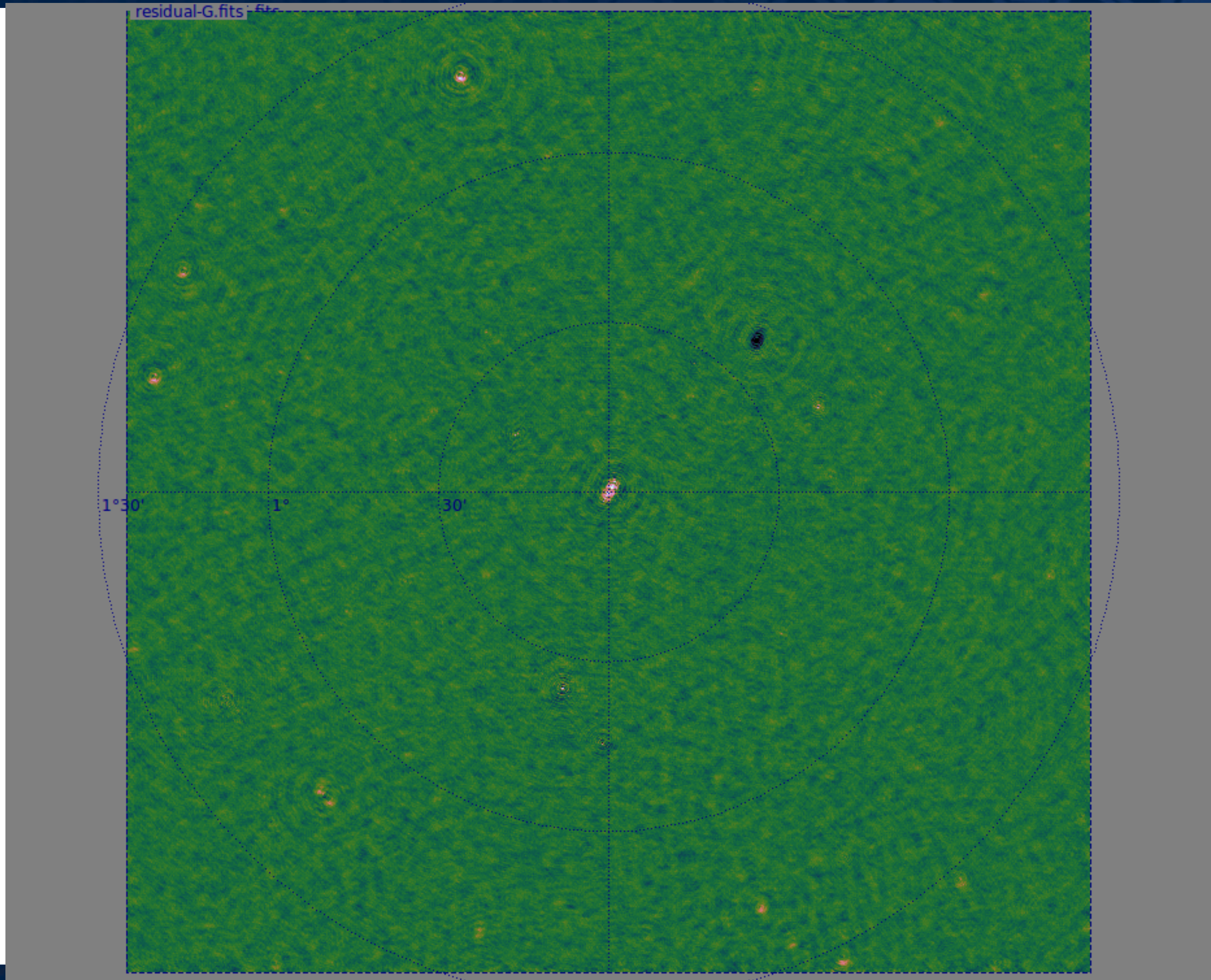
  - where **G**'s are diagonal
  - (and this was used for the 3C147 reduction)
- LOFAR needs 2x2 right off the bat
- Reformulated the algorithm to use full 2x2 Jones matrices instead

LOFAR Double-Double

- D-D observation
  - Flagged, demixed and averaged in time
  - 1240 timeslots (7h), 1 channel, 990 baselines
  - ~40 sources in the LSM
- Doing full 2x2 G-Jones solution *w/o the beam*
- MeqTrees+LSQ runtime: **~15m**
  - (BBS ~30m)
- MeqTrees+Stefcal: **~2m**
  - of which ~half in the solver

22

LOFAR DD



OXFORD e-Research CENTRE

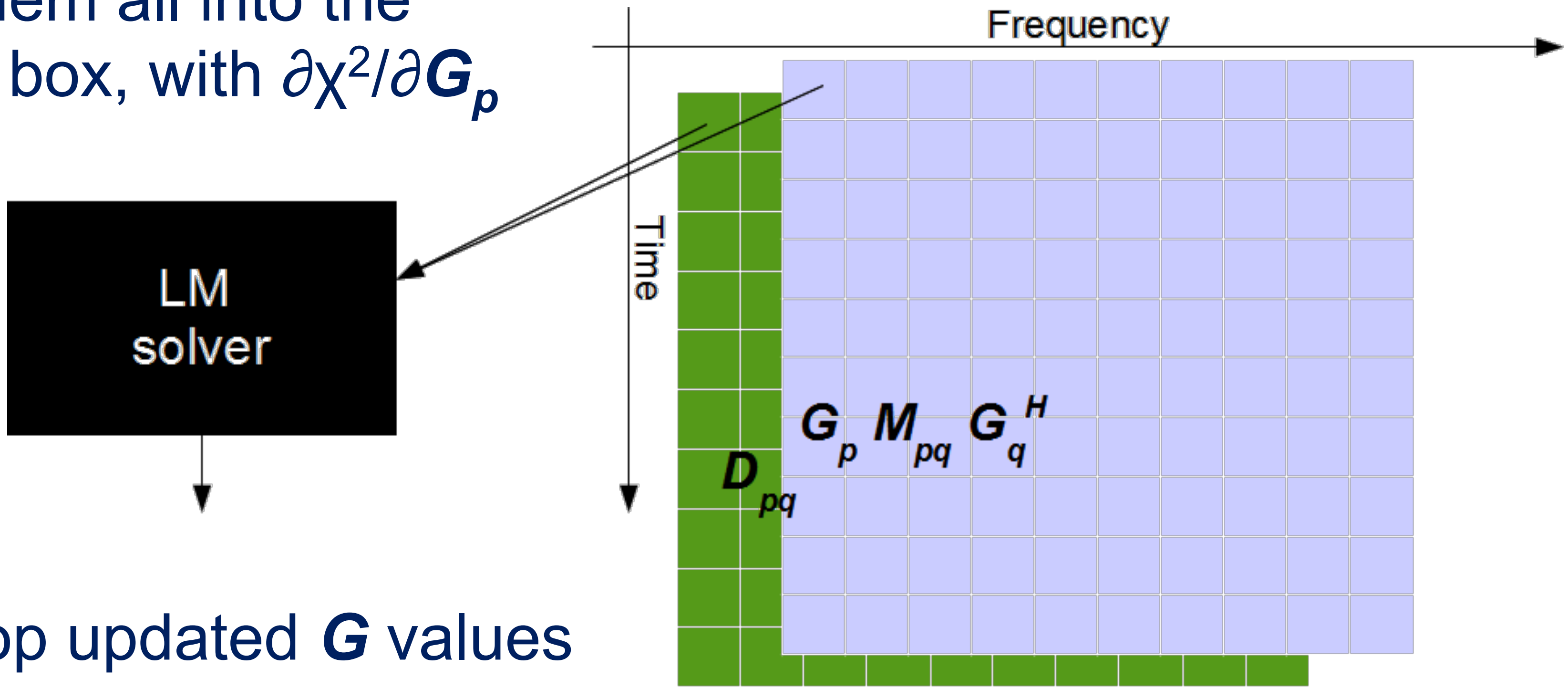
Scaling

	WSRT (14) 74 baselines 28 ch 1437 t/s 300 sources	LOFAR (45) 990 baselines 1 ch x 1280 t/s 42 sources full 2x2 solution	MeerKAT (64) 2016 baselines 8 ch x 480 t/s 884 sources	SKA1 (256)
MeqTrees stefcal time	1m40s	2m20s	3m	
MeqTrees selfcal (LSQ) time	10m	15m	24m	
"pure" stefcal time (minus I/O and model predict)	1m	1m15s	1m20s The Stefcal time …	
"pure" selfcal time	~9m	~13m	~22m	
speedup	**x9**	**x11**	**x16**	**xSilly**

OXFORD e-Research CENTRE

What's going on here?

- Selfcal: find ***G*** to minimize $D_{pq} - G_p M_{pq} G_q^H$
- Each t/f point is a set of χ^2-equations (per baseline)
- toss them all into the solver box, with $\partial\chi^2/\partial G_p$



- Out pop updated ***G*** values
- Rinse and repeat, until it converges

The Stefcal Version: Linearizing the RIME

- Find ***G*** to minimize $D_{pq} - G_p M_{pq} G(0)_q^H$
  - Where ***G(0)*** is the value from the previous iteration
- But this is just a linear equation!
- Can just write out the ***G*** updates directly:

$$G_p = \sum_p D_q Y_{pq}^H \left(\sum_q Y_q Y_q^H\right)^{-1} \text{, where } Y_p = M_{pq} G_{(0)pq}^H$$

- This gives us one approximate update step
- Rinse & repeat until it converges
  - (With some clever averaging – essential to achieve convergence)

The Stefcal advantage

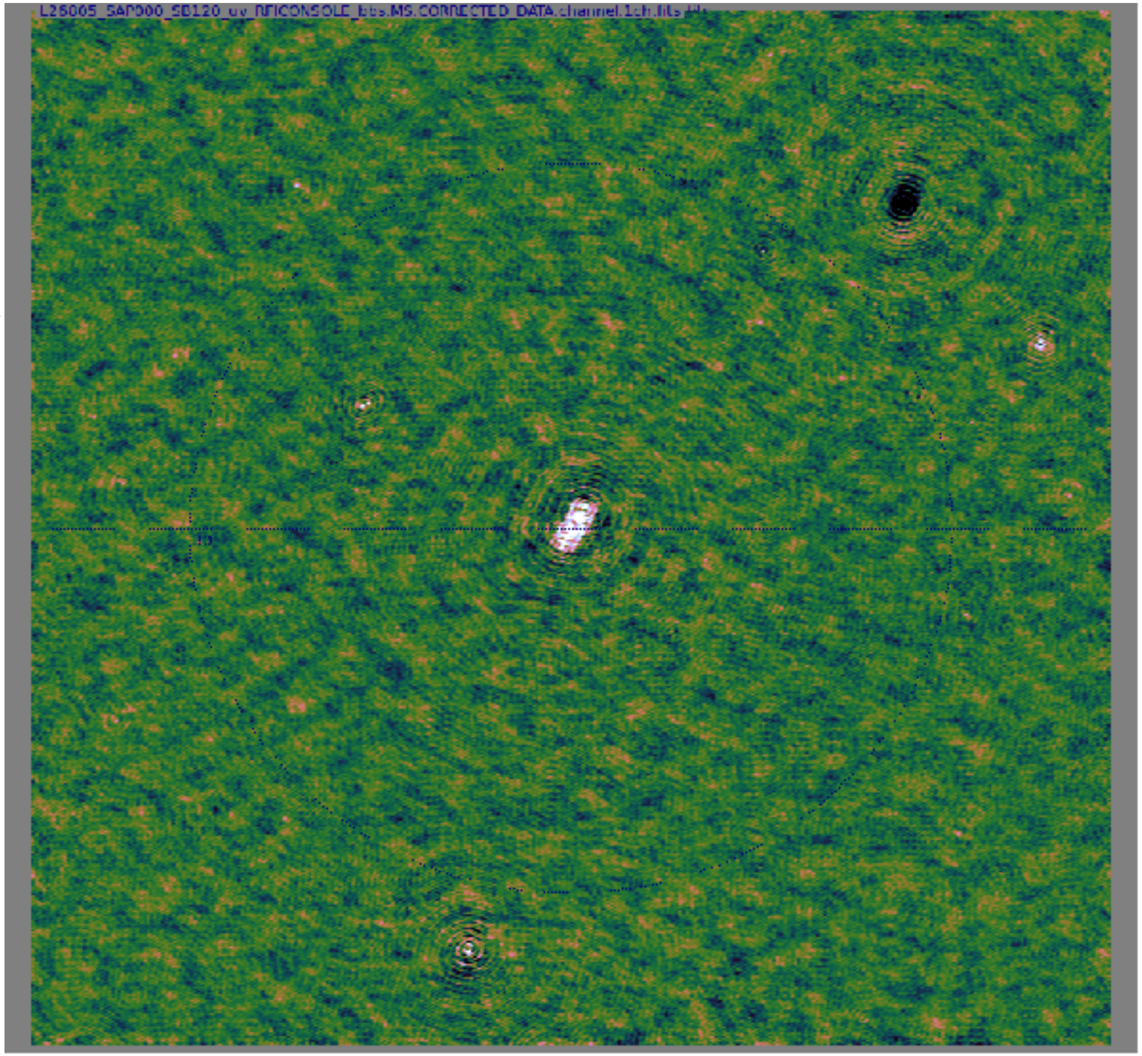
- For N antennas, model predict scales as N^2
  - There are N^2 baselines after all...
- LM (or any least-squares) scales as N^3 due to matrix inversion
- Stefcal update step scales as N^2
  - ...and is very cheap to compute
  - ...so we can do many fast iterations
- <u>No need for derivatives!</u>
  - Cheap on RAM
  - Can process entire WSRT/LOFAR MS in one gulp

Stefcal & differential gains

- Stefcal adapted for **differential gains**
- Use Stefcal iteratively
- So far, preliminary studies (O.Smirnov, S.Salvini) proved fast and accurate

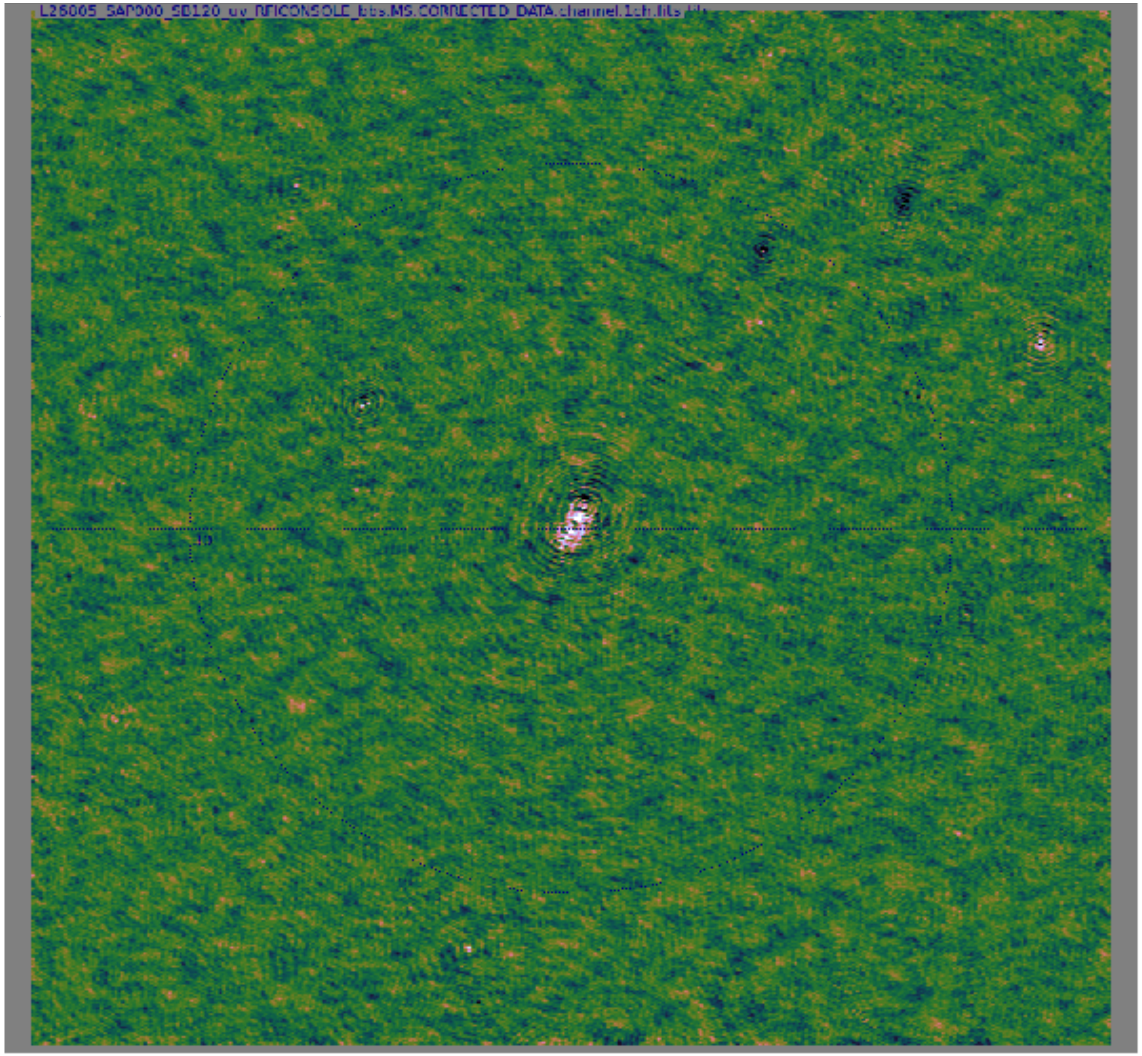
Does it work for LOFAR?

- Yes, though more testing needed
- LOFAR DD field, 2 directions: 7m
- Scales linearly with # of directions 9 directions: ~20m
- Bonus: improves ***G*** at the same time



Does it work for LOFAR?

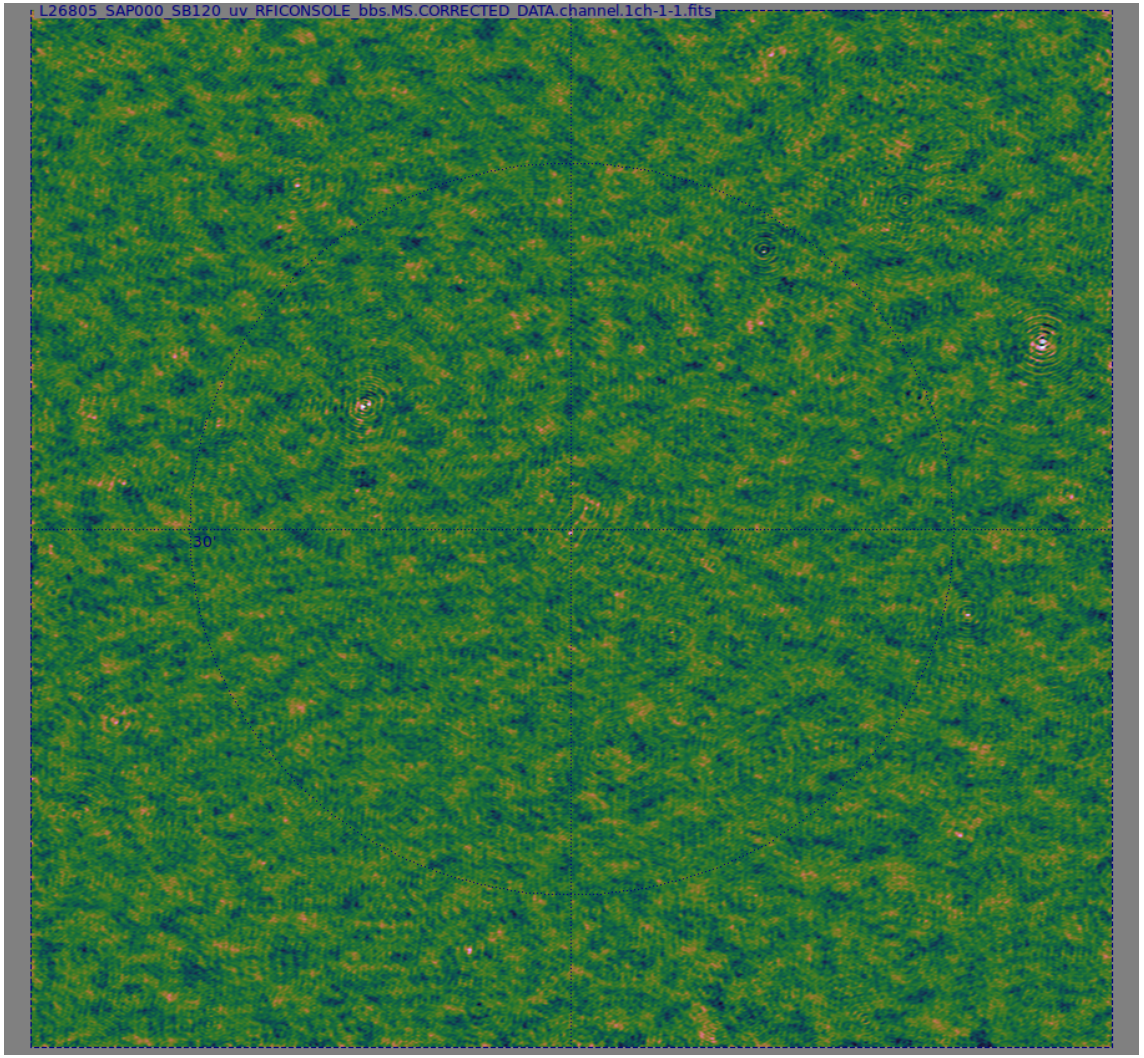
- Yes, though more testing needed
- LOFAR DD field, 2 directions: 7m
- Scales linearly with # of directions 9 directions: ~20m
- Bonus: improves ***G*** at the same time



30

Does it work for LOFAR?

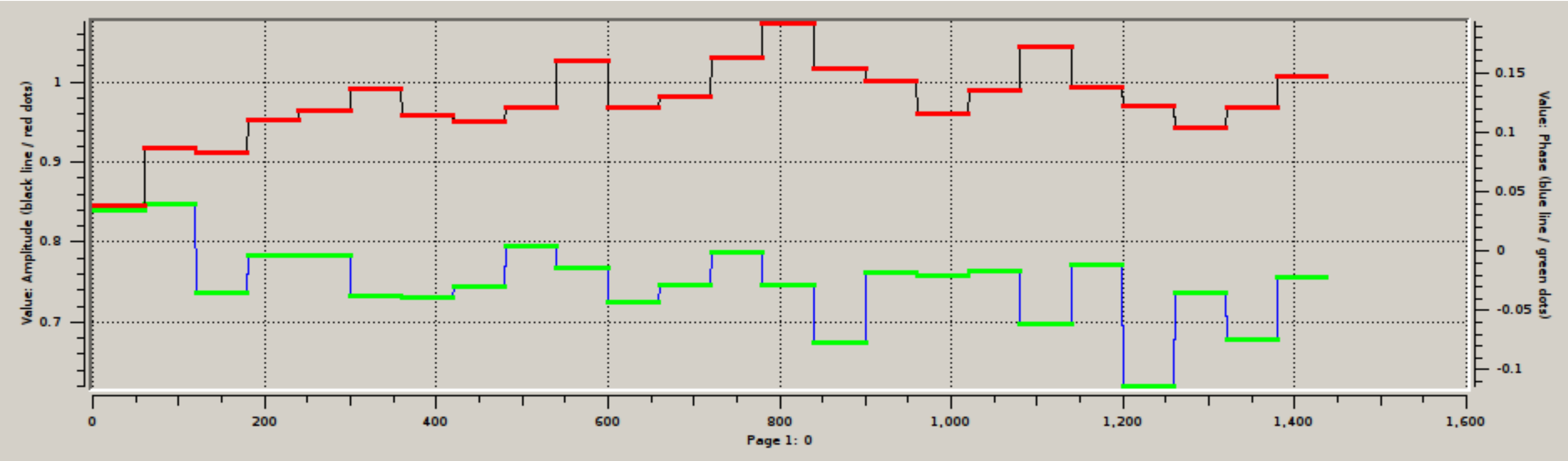
- Yes, though more testing needed
- LOFAR DD field, 2 directions: 7m
- Scales linearly with # of directions 9 directions: ~20m
- Bonus: improves ***G*** at the same time

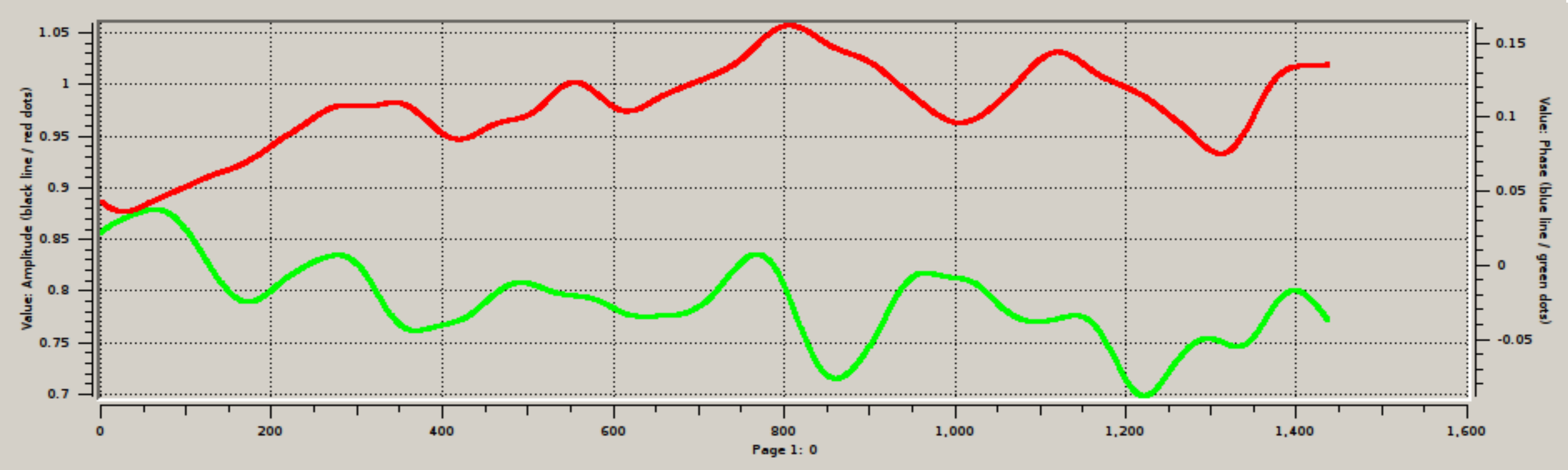


Solution Intervals

- Direction-dependent solutions need to be solved for on <u>longer</u> time intervals than ***G***
- Up till now, we achieved this by solving for e.g. one ***dE*** value per block of *M* timeslots
- Very difficult to mix-and-match intervals in LSQ
- Hence, first do G ($\Delta t=1$),
  then dE ($\Delta t=120$)
- In the stefcal update step calculation, larger solution intervals are just an extra summation.
  - Much cheaper than standard approach
  - Smoothing possible
- Low extra computational cost (~ 20 %) when using a *sliding* average
  - Same results as sliding selfcal

Piecewise vs. smooth *dE*s





In Conclusion: What's The Catch?

- Classic selfcal (and Levenberg-Marquardt) is not necessarily optimal, so why does there have to be one?
- Convergence heuristics needs further improvement
  - ...but then, we don't understand selfcal either really
  - so stefcal we don't understand too, just x10 speed improvemnt
- More testing needed, especially the 2x2 case
  - And especially peeling...
- Quick-and-dirty Python implementation can be rewritten
- The “fast” version (SVD) can be adapted to stefcal
- More gains (factor of several) readily available