

Fast gain calibration in radio astronomy using alternating direction implicit methods: Analysis and applications

Stefano Salvini¹ and Stefan J. Wijnholds²

¹ Oxford e-Research Centre, 7 Keble Road, OX1 3QG, Oxford, UK
e-mail: stef.salvini@oerc.ox.ac.uk

² Netherlands Institute for Radio Astronomy (ASTRON), PO Box 2, 7990 AA Dwingeloo, The Netherlands
e-mail: wijnholds@astron.nl

Received 27 June 2014 / Accepted 18 August 2014

ABSTRACT

Context. Modern radio astronomical arrays have (or will have) more than one order of magnitude more receivers than classical synthesis arrays, such as the VLA and the WSRT. This makes gain calibration a computationally demanding task. Several alternating direction implicit (ADI) approaches have therefore been proposed that reduce numerical complexity for this task from $O(P^3)$ to $O(P^2)$, where P is the number of receive paths to be calibrated

Aims. We present an ADI method, show that it converges to the optimal solution, and assess its numerical, computational and statistical performance. We also discuss its suitability for application in self-calibration and report on its successful application in LOFAR standard pipelines.

Methods. Convergence is proved by rigorous mathematical analysis using a contraction mapping. Its numerical, algorithmic, and statistical performance, as well as its suitability for application in self-calibration, are assessed using simulations.

Results. Our simulations confirm the $O(P^2)$ complexity and excellent numerical and computational properties of the algorithm. They also confirm that the algorithm performs at or close to the Cramer-Rao bound (CRB, lower bound on the variance of estimated parameters). We find that the algorithm is suitable for application in self-calibration and discuss how it can be included. We demonstrate an order-of-magnitude speed improvement in calibration over traditional methods on actual LOFAR data.

Conclusions. In this paper, we demonstrate that ADI methods are a valid and computationally more efficient alternative to traditional gain calibration methods and we report on its successful application in a number of actual data reduction pipelines.

Key words. instrumentation: interferometers – methods: numerical – methods: statistical – techniques: interferometric

1. Introduction

Antenna-based gain calibration is a key step in the data reduction pipeline of any radio telescope. A commonly used method of estimating these antenna-based gains and possible other parameters in a (self-)calibration process is the Levenberg-Marquardt (LM) nonlinear least squares solver. Theoretically, the LM algorithm has at least $O(N^3)$ complexity, where N is the number of free parameters to be estimated. The LM solver has proved its value in self-calibration processes, but it is becoming a limiting factor in (near) real-time pipelines for modern telescopes, such as the Low Frequency Array (LOFAR, de Vos et al. 2009; Van Haarlem et al. 2013) and the Murchison Widefield Array (MWA, Lonsdale et al. 2009; Bowman et al. 2013), owing to its cubic scaling with the number of receivers. The situation will only become worse for the Square Kilometre Array (SKA, Dewdney et al. 2009, 2013).

This has motivated researchers to search for faster solvers with better scalability for antenna based gain calibration. Hamaker (2000) has already noted that solving for the gain of one specific receive path, assuming that all other receive paths are already calibrated while iterating over all antennas, could potentially be a fast way to solve for antenna-based gains in full polarization. This leads to an alternating direction implicit (ADI) method, which is used in the MWA real-time system for tile based calibration (Mitchell et al. 2008). In the MWA pipeline, the gain estimates found for a given timeslice are used as initial

estimates for the next timeslice. This makes a single iteration sufficient for achieving the required calibration accuracy. This cleverly exploits the electronic stability of the MWA system. Mitchell et al. (2008) also proposed to reduce the noise on the estimates by using a weighted average between the current and previous gain estimates.

Salvini et al. (2011) showed that averaging the odd and even iterations not only reduces the noise on the estimates, but also considerably increases the rate of convergence and the robustness of the method. ADI methods have $O(P^2)$ complexity where P is the number of receive paths to be calibrated. Since the number of visibilities also scales with P^2 , these algorithms scale linearly with the number of data points and therefore have the lowest possible computational complexity for algorithms exploiting all available information.

Iterative algorithms, such as the ADI method presented in this paper, can be sensitive to the choice of initial estimates or exhibit slow convergence. In Sect. 4 we therefore provide a rigorous convergence analysis. This gives a clear view of the algorithm's effectiveness and its potential limitations. We also discuss why these limitations are unlikely to hamper proper performance of the algorithm in practical radio astronomical applications, as by its actual use.

Hamaker (2000), Mitchell et al. (2008), and Salvini et al. (2011) have derived the basic ADI iteration from the unweighted least squares cost function. In practice, weighted least squares methods are known to provide more accurate estimates if the

signal-to-noise ratio (S/N) varies widely among data points. In this paper, we therefore start our derivation from the weighted least squares cost function and show that radio astronomical arrays can usually exploit the unweighted LS method.

In Sect. 5, we compare the statistics of the gain estimates produced by the algorithm in Monte Carlo simulations with the Cramer-Rao bound (CRB). The CRB is the theoretical lower bound on the covariance of the estimated parameters obtained by an ideal unbiased estimator. The results indicate that the algorithm performs at the CRB when the covariance matched and unweighted least squares cost functions coincide (as expected) while performing very close to the CRB in most realistic scenarios. In the radio astronomical community, the ADI method presented in this paper is now usually referred to as *StefCal*, a name coined in jest by our colleagues Oleg Smirnov and Ilse van Bemmelen. In view of its (close to) statistically efficient performance and high computational efficiency, we adapted the name to *StEFCal*, an admittedly rather contrived acronym for “statistically efficient and fast calibration”.

StEFCal provides a considerable computational advantage over algorithms derived from the weighted least squares cost function, which usually scale with P^3 (Ng & See 1996; Boonstra & van der Veen 2003; Wijnholds & van der Veen 2009). In Sect. 5 we also consider the computational performance of *StEFCal*, highlighting its low computational complexity as well as its efficiency, its very small memory footprint, and scalability with problem size.

In Sect. 6, we briefly discuss the extension of *StEFCal* to full polarization calibration. This is now used routinely within *MEqTrees* (Noordam & Smirnov 2010) and the LOFAR standard preprocessing pipeline (Salvini & Wijnholds 2014a). Our simulations in Sect. 5 also show that *StEFCal* is suitable for integration in self-calibration approaches that rely on iterative refinement of the source model. In Sect. 7, we show that *StEFCal* can be easily integrated in an actual system by reporting on the successful integration of *StEFCal* in a calibration pipeline for a LOFAR subsystem.

We conclude our paper in Sect. 8 by discussing possible alternative variants of the algorithm and possibilities for integrating *StEFCal* as a building block in other calibration algorithms, including algorithms dealing with direction-dependent gains, such as *SAGECal* (Yatawatta et al. 2009; Kazemi et al. 2011) and the differential gains method proposed by Smirnov (2011), corrupted or missing data values and polarization. For convenience of the reader, Table 1 summarizes the notational conventions and frequently used symbols in this paper.

2. Problem statement

2.1. Measurement equation

The radio astronomical system to be calibrated can have many different architectures. For example, antenna-based gain calibration can be applied to a synthesis array of dishes in interferometers, such as the VLA or the WSRT, but also to a synthesis array of stations in instruments, such as LOFAR or the envisaged Low Frequency Aperture Array (LFAA) system for the SKA (Dewdney et al. 2013). Antenna-based gain calibration is also required within an aperture array station, where it becomes tile-based calibration in systems, such as the LOFAR high band antenna system (Van Haarlem et al. 2013) or the MWA. In this paper, we will therefore use generic terms, such as “receiving element”, “element” or “antenna” to denote an individual element in a (synthesis) array instead of architecture-dependent

Table 1. Notation and frequently used symbols.

a	scalar value
\mathbf{a}	vector \mathbf{a}
\mathbf{A}	matrix \mathbf{A}
$\mathbf{A}_{:,k}$	k th column of the matrix \mathbf{A}
$\text{diag}(\cdot)$	converts a vector into a diagonal matrix
\odot	Hadamard or element-wise product of matrices or vectors
$(\cdot)^*$	conjugation
$(\cdot)^T$	transpose
$(\cdot)^H$	Hermitian transpose
$(\cdot)^{[i]}$	value at the i th iteration
$\mathcal{E}\{\mathbf{A}\}$	expected value of \mathbf{A}
$\widehat{\mathbf{R}}$	array covariance matrix, diagonal set to zero
\mathbf{M}	model cov. matrix of observed scene, diagonal set to zero
\mathbf{g}	vector of complex valued receiver path gains
\mathbf{G}	$\mathbf{G} = \text{diag}(\mathbf{g})$
Δ	$\Delta = \widehat{\mathbf{R}} - \mathbf{G}\mathbf{M}\mathbf{G}^H$

terms such as “dish”, “station” or “tile”. We will also use the word “array” to refer to the system of elements to be calibrated instead of specific terms such as “station array”, “synthesis array” or “tile array”.

In this paper, we consider the scalar measurement equation or data model. The ADI method can be extended to full polarization as shown by Hamaker (2000), Mitchell et al. (2008) and Salvini & Wijnholds (2014a,b), but this complicates the analysis unnecessarily. In our analysis, we assume that the source and noise signals are represented by complex valued samples that are mutually and temporally independent and that can be modeled as identically distributed Gaussian noise. We assume that these signals are spectrally filtered such that the narrowband condition (Zatman 1998) holds, which ensures that time delays can be represented by multiplication by phasors.

Besides allowing this representation of time delays, spectral filtering is a crucial step in (ultra-)wide band systems like modern radio telescopes for two other reasons. Firstly, it ensures that the noise in each channel can be assumed to be white noise regardless of bandpass fluctuations of the instrument or the inherent power spectrum of the observed sources. Secondly, observations using an increasingly larger fractional bandwidth are more likely to be affected by human-generated radio frequency interference (RFI). Most of this RFI can be detected and flagged (Boonstra 2005; Offringa 2012; Offringa et al. 2013). If the channel width matches the bandwidth of the RFI signals (typically a few kHz), the S/N of the RFI in the occupied channel is maximized, thereby facilitating detection. As an additional bonus, the amount of spectrum that is flagged is minimized in this case. RFI that escapes detection can cause outliers in the measured data that do not fit a Gaussian noise model. In such cases, an appropriate weighting of the data samples can help to improve robustness to such outliers (Kazemi & Yatawatta 2013). In Sect. 8 we briefly discuss how such weighting can be incorporated in *StEFCal*, although at the expense of some computational efficiency.

The direction-independent gain of the p th receive path of an array consisting of P elements can be represented by the complex valued scalar g_p . The output signal of the p th element, receiving signals from Q sources, as a function of time t can be described by

$$x_p(t) = g_p \sum_{q=1}^Q a_{p,q} s_q(t) + n_p(t), \quad (1)$$

where $a_{p,q}$ is the p th antenna's response to the q th source, $s_q(t)$ is the source signal, and $n_p(t)$ represents the noise on the measurement. Since we assume that the narrowband condition holds, the factors $a_{p,q}$ are the geometry-dependent phase terms that result in the familiar Fourier-transform relationship between the source structure and the measured visibilities (Thompson et al. 2004).

We can stack the element-indexed quantities in vectors of length P : the output signals of the individual sensors in $\mathbf{x}(t) = [x_1(t), \dots, x_P(t)]^T$; the complex valued gains in $\mathbf{g} = [g_1, \dots, g_P]^T$; the array response vector to the q th source as $\mathbf{a}_q = [a_{1,q}, \dots, a_{P,q}]^T$; finally, the noise vector as $\mathbf{n}(t) = [n_1(t), \dots, n_P(t)]^T$.

With these definitions, we can describe the array signal vector as

$$\mathbf{x}(t) = \mathbf{g} \odot \sum_{q=1}^Q \mathbf{a}_q s_q(t) + \mathbf{n}(t). \quad (2)$$

Defining the $P \times P$ diagonal matrix $\mathbf{G} = \text{diag}(\mathbf{g})$, the $P \times Q$ array response matrix $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_Q]$, and the $Q \times 1$ source signal vector $\mathbf{s}(t) = [s_1(t), \dots, s_Q(t)]^T$, we can reformulate (2) in a convenient matrix form:

$$\mathbf{x}(t) = \mathbf{G}\mathbf{A}\mathbf{s}(t) + \mathbf{n}(t). \quad (3)$$

Defining $\mathbf{X} = [\mathbf{x}(T), \dots, \mathbf{x}(KT)]$, where K is the number of samples and KT defines the overall measurement duration, we can then estimate the array covariance matrix, often referred to as visibility matrix or matrix of visibilities, by

$$\widehat{\mathbf{R}} = \frac{1}{K} \mathbf{X}\mathbf{X}^H. \quad (4)$$

The model for the array covariance matrix follows from

$$\mathbf{R} = \mathcal{E} \left\{ \frac{1}{K} \mathbf{X}\mathbf{X}^H \right\} = \mathbf{G}\mathbf{A}\mathbf{\Sigma}_s\mathbf{A}^H\mathbf{G}^H + \mathbf{\Sigma}_n, \quad (5)$$

where $\mathbf{\Sigma}_s = \mathcal{E} \{ \mathbf{s}(t)\mathbf{s}^H(t) \}$ is the covariance matrix of the source signals, and $\mathbf{\Sigma}_n = \mathcal{E} \{ \mathbf{n}(t)\mathbf{n}^H(t) \}$ is the noise covariance matrix. In (5), we have assumed that the source and noise signals are mutually uncorrelated. We also assume that the noise signals on the individual sensors are uncorrelated, such that the noise covariance matrix is diagonal, i.e., $\mathbf{\Sigma}_n = \text{diag}(\boldsymbol{\sigma}_n)$. In Sect. 8, we indicate how the algorithm can deal with more complicated noise models. For convenience of notation, we introduce $\mathbf{R}_0 = \mathbf{A}\mathbf{\Sigma}_s\mathbf{A}^H$, so that we can write (5) as

$$\mathbf{R} = \mathbf{G}\mathbf{R}_0\mathbf{G}^H + \mathbf{\Sigma}_n. \quad (6)$$

2.2. Optimization problem

The antenna based gains and phases can be calibrated by a measurement in which the source structure is known, so we can predict model visibilities \mathbf{R}_0 . Since the receiver path noise powers $\boldsymbol{\sigma}_n$ are usually not known, the calibration problem is described by

$$\{ \mathbf{g}, \widehat{\boldsymbol{\sigma}}_n \} = \underset{\mathbf{g}, \boldsymbol{\sigma}_n}{\text{argmin}} \left\| \mathbf{W}^H (\widehat{\mathbf{R}} - \mathbf{G}\mathbf{R}_0\mathbf{G}^H - \mathbf{\Sigma}_n) \mathbf{W} \right\|_F^2. \quad (7)$$

This equation describes our problem as a weighted least squares estimation problem. This allows us to apply covariance matched weighting by taking $\mathbf{W} = \mathbf{R}^{-1/2}$, leading to estimates that are asymptotically, for a large number of samples, equivalent to

maximum likelihood estimates (Ottersten et al. 1998). However, in radio astronomy, sources are typically much weaker than the noise, i.e., the S/N per sample is usually very low. Exceptions to this statement are observations of the brightest sources on the sky, such as Cas A, Cyg A, and the Sun, in which self-noise becomes a significant issue (Kulkarni 1989; Wijnholds 2010). Besides such exceptional cases, the model covariance matrix can be approximated by $\mathbf{R} \approx \mathbf{\Sigma}_n$. Since many radio astronomical instruments are arrays of identical elements, whereby $\mathbf{\Sigma}_n \approx \sigma_n \mathbf{I}$, we are justified in using $\mathbf{W} = \mathbf{I}$. In the Monte Carlo simulations presented in Sect. 5.3, we demonstrate that violating these assumptions only leads to small deviations from the CRB, even in extreme situations unlikely to occur in reality.

In many practical cases, we have an incomplete model of the observed field, and we employ the best available model $\mathbf{M} \approx \mathbf{R}_0$ which, for example, only includes the brightest sources. In our simulations, we consider both complete and incomplete information on the observed field. Another practical matter is that the autocorrelations are dominated by the noise power of the array elements. Since accurate modeling of the diagonal of the array covariance matrix involves estimating the noise power of each individual element, we are forced to estimate the antenna-based gains using the crosscorrelations followed by estimation of the noise powers using the diagonal elements. For the gain estimation step, it is therefore convenient to set the diagonal entries of $\widehat{\mathbf{R}}$ and \mathbf{M} to zero. This assumption is made throughout this paper, thus ignoring $\mathbf{\Sigma}_n$. This simplifies the estimation problem described in (7) to

$$\mathbf{g} = \underset{\mathbf{g}}{\text{argmin}} \left\| \widehat{\mathbf{R}} - \mathbf{G}\mathbf{M}\mathbf{G}^H \right\|_F^2 = \underset{\mathbf{g}}{\text{argmin}} \|\Delta\|_F^2 \quad (8)$$

where we have introduced $\Delta = \widehat{\mathbf{R}} - \mathbf{G}\mathbf{M}\mathbf{G}^H$ for brevity of notation.

3. The algorithm

Using an ADI approach, we first solve for \mathbf{G}^H holding \mathbf{G} constant, then for \mathbf{G} holding \mathbf{G}^H constant. Since Δ is Hermitian, the two steps are equivalent and the iteration consists of only the following step:

$$\mathbf{G}^{[i]} = \underset{\mathbf{G}}{\text{argmin}} \left\| \widehat{\mathbf{R}} - \mathbf{G}^{[i-1]}\mathbf{M}\mathbf{G}^H \right\|_F. \quad (9)$$

Since $\|\mathbf{x}\|_F = \|\mathbf{x}\|_2$ for any vector \mathbf{x} , and setting

$$\mathbf{Z}^{[i]} = \mathbf{G}^{[i]}\mathbf{M} \quad (10)$$

we can write

$$\|\Delta\|_F = \left\| \widehat{\mathbf{R}} - \mathbf{Z}\mathbf{G}^H \right\|_F = \sqrt{\sum_{p=1}^P \left\| \widehat{\mathbf{R}}_{:,p} - \mathbf{Z}_{:,p} g_p^* \right\|_2^2}. \quad (11)$$

Equation (11) shows that the complex gains g_p are decoupled and that each iteration consists of solving P independent $P \times 1$ linear least squares problems. Using, for example, the normal equation method we readily obtain:

$$g_p^{[i]} = \left\{ \frac{(\mathbf{Z}_{:,p}^{[i-1]})^H \cdot \widehat{\mathbf{R}}_{:,p}}{(\mathbf{Z}_{:,p}^{[i-1]})^H \cdot \mathbf{Z}_{:,p}^{[i-1]}} \right\}^* = \frac{\widehat{\mathbf{R}}_{:,p}^H \cdot \mathbf{Z}_{:,p}^{[i-1]}}{(\mathbf{Z}_{:,p}^{[i-1]})^H \cdot \mathbf{Z}_{:,p}^{[i-1]}}, \quad (12)$$

which is the basic ADI iteration.

In practice, the basic iteration may converge very slowly. For example, in the case of the sky model used in Sect. 5, it does

not converge at all, bouncing to and fro between two vectors \mathbf{g} . We resolved this issue by replacing the gain solution of each *even* iteration by the average of the current gain solution and the gain solution of the previous *odd* iteration. This simple process makes the iteration both very fast and very robust. This is the basic variant of the StEFCal algorithm, which is described here as Algorithm 1. Its convergence properties are studied in detail in the next section, while its numerical, computational, and statistical performance are discussed in Sect. 5.

Algorithm 1 Algorithm StEFCal

```

Initiate  $\mathbf{G}^{[0]}$ ;  $\mathbf{G}^{[0]} = \mathbf{I}$  is adequate in most cases
for  $i = 1, 2, \dots, i_{\max}$  do
  for  $p = 1, 2, \dots, P$  do
     $\mathbf{z} \leftarrow \mathbf{G}^{[i-1]} \cdot \mathbf{M}_{:,p} \equiv \mathbf{g}^{[i-1]} \odot \mathbf{M}_{:,p}$ 
     $g_p \leftarrow (\widehat{\mathbf{R}}_{:,p}^H \cdot \mathbf{z}) / (\mathbf{z}^H \cdot \mathbf{z})$ 
  end for
  if  $\text{mod}_2(i) = 0$  then
    if  $\|\mathbf{g}^{[i]} - \mathbf{g}^{[i-1]}\|_F / \|\mathbf{g}^{[i]}\|_F \leq \tau$  then
      Convergence reached
    else
       $\mathbf{G}^{[i]} \leftarrow (\mathbf{G}^{[i]} + \mathbf{G}^{[i-1]})/2$ 
    end if
  end if
end for

```

We want to stress a few important points

- There are no parallel dependencies in the inner loop (the dependency on \mathbf{z} is trivially resolved by employing a local vector \mathbf{z} on each computational unit or core, or by using the individual elements of \mathbf{z} at once without needing to store the vector, although at the cost of potentially lower performance).
- All data are accessed through unit strides (contiguous memory locations).
- The memory footprint is very small. Basically, only one extra P -vector is required (for requirements of \mathbf{z} see above) besides the visibility matrices.

Thus, the algorithm can be readily implemented on many-core architectures (such as GPUs, Intel Xeon Phi, etc.), as well as on multiple cores, for example by employing OpenMP. Codes and algorithms are in a very advanced development stage and are available on request.

The gain estimation problem has an inherent phase ambiguity. In this paper, we choose, entirely arbitrarily, to use the first receiver as phase reference. This constraint can be imposed either within each iteration at the cost of $\mathcal{O}(P)$ operations or at the end of the computation. In practical terms, we did not find any difference in rate of convergence and results between these two options.

We now look at the algorithm in terms of the gradient with respect to the real and imaginary parts of the complex gains of the function

$$\mathbf{g} = \underset{\mathbf{g}}{\operatorname{argmin}} \|\Delta\|_F^2 = \underset{\mathbf{g}}{\operatorname{argmin}} \operatorname{Tr}(\Delta\Delta^H). \quad (13)$$

At a minimum, the partial derivatives with respect to the real and imaginary parts of the complex gains must all be zero. Hence,

$$\begin{aligned} \frac{\partial}{\partial \operatorname{Re}(g_p)} \operatorname{Tr}(\Delta\Delta^H) &= \operatorname{Tr} \left[\frac{\partial \Delta\Delta^H}{\partial \operatorname{Re}(g_p)} \right] = \\ &= -\operatorname{Tr} \left[(\mathbf{E}_p \mathbf{M} \mathbf{G}^H + \mathbf{G} \mathbf{M} \mathbf{E}_p) \Delta^H + \text{c.c.} \right] \\ &= -2\operatorname{Re} \left[\operatorname{Tr}(\mathbf{E}_p \mathbf{M} \mathbf{G}^H \Delta^H) + \operatorname{Tr}(\mathbf{G} \mathbf{M} \mathbf{E}_p \Delta^H) \right] \\ &= -4\operatorname{Re} \left\{ \operatorname{Tr}(\Delta \mathbf{E}_p \mathbf{Z}^H) \right\} \\ &= -4\operatorname{Re} \left[\mathbf{Z}_{:,p}^H \Delta_{:,p} \right] \\ &= -4\operatorname{Re} \left[\mathbf{Z}_{:,p}^H \cdot (\widehat{\mathbf{R}}_{:,p} - \mathbf{Z}_{:,p} g_p^*) \right] \\ &= 0 \end{aligned} \quad (14)$$

where c.c. stands for complex conjugate and \mathbf{E}_p denotes the $P \times P$ elementary matrix, which only contains zeros except for the (p, p) -element, which is unity. We used the properties of the trace of a product of matrices; we used the Hermitian properties of Δ , \mathbf{M} , and \mathbf{E}_p ; and finally, we used $\mathbf{Z}_{:,p} = (\mathbf{G}\mathbf{M})_{:,p}$. Likewise, for the imaginary part of g_p we obtain

$$\frac{\partial \operatorname{Tr}(\Delta\Delta^H)}{\partial \operatorname{Im}(g_p)} = -4\operatorname{Im} \left[\mathbf{Z}_{:,p}^H \cdot (\widehat{\mathbf{R}}_{:,p} - \mathbf{Z}_{:,p} g_p^*) \right] = 0. \quad (15)$$

Looking at (11), (14), and (15), and at Algorithm 1, we can see that the termination condition in the algorithm implies zero gradient as a function of the real and imaginary parts of all g_p , for $p = 1, \dots, P$, achieved through a process of local minimization via a linear least squares method. Because the algorithm shows very good convergence in all realistic cases studied, we can infer that StEFCal does indeed produce gains that minimize Δ in the least squares sense.

Moreover, using Eqs. (14) and (15), we can obtain the components of the gradient with respect to the real and imaginary part of \mathbf{g} at the i th iteration by

$$\begin{aligned} \nabla_{\operatorname{Re}(g_p)} \|\Delta\|_F^2 &= -4\operatorname{Re} \left[\mathbf{z}_p^{[i-1]H} \widehat{\mathbf{R}}_{:,p} - (\mathbf{z}_p^{[i-1]H} \mathbf{z}_p^{[i-1]}) g_p^{[i-1]*} \right] \\ \nabla_{\operatorname{Im}(g_p)} \|\Delta\|_F^2 &= -4\operatorname{Im} \left[\mathbf{z}_p^{[i-1]H} \widehat{\mathbf{R}}_{:,p} - (\mathbf{z}_p^{[i-1]H} \mathbf{z}_p^{[i-1]}) g_p^{[i-1]*} \right] \end{aligned}$$

where we used the notation $\mathbf{z}_p^{[i-1]} = \mathbf{Z}_{:,p}^{[i-1]} = \mathbf{g}^{[i-1]} \odot \mathbf{M}_{:,p}$. The dot products have already been computed to generate the new gain estimate $g_p^{[i]}$, so the components of the gradient can be generated at virtually no cost.

4. Analysis of convergence

In this section we first introduce the concept of contraction mapping and then employ this concept to analyze the convergence properties of the proposed algorithm. The special case of calibration on a single point source shows that the algorithm converges for all initial estimates except for initial estimates in the null space of \mathbf{g} . Finally, we study the general case of an arbitrary source distribution, showing that convergence is achieved when certain conditions on the observed scene and the initial estimate are met. We discuss these conditions and argue that they are met in practical situations. The convergence analysis presented below considers convergence in the noise-free case. The effect of measurement noise is studied in detail using Monte Carlo simulations in Sect. 5.3.

4.1. Condition for convergence

A contraction mapping on a complete metric space \mathcal{M} with distance measure d is a function $f : \mathcal{M} \rightarrow \mathcal{M}$ with the property that a real valued number $\mu < 1$ exists such that for all $x, y \in \mathcal{M}$

$$d(f(x), f(y)) \leq \mu d(x, y). \quad (16)$$

The Banach fixed point theorem states that the sequence of values resulting from iterative application of a contraction mapping converges to a fixed point (Palais 2007). This theorem can be understood intuitively. We consider two arbitrary points in a Euclidian space using the induced norm of their difference to measure the distance between them. If a given operation is a contraction mapping, applying that operation to both points separately will produce two new points that are closer together. Repeated application of the operation on the resulting points will make the distance between each pair of new points shorter than the previous pair. If we continue applying the operation long enough, we can make this distance arbitrarily small, thus effectively converging to a single point. If we can show that the full iteration (two basic iteration as described by (12) and the averaging step) is a contraction mapping, we can conclude that the iterative application of the full iteration leads to a converging sequence of values $g^{[i]}$.

Replacing $\tilde{\mathbf{R}}$ with $\mathbf{R} = \mathbf{G}\mathbf{M}\mathbf{G}^H = \mathbf{g}\mathbf{g}^H \odot \mathbf{M}$, the basic iteration for a single element described by (12) reads as

$$g_p^{[i]} = \frac{(\mathbf{g} \odot \mathbf{M}_{:,p})^H (\mathbf{g}^{[i-1]} \odot \mathbf{M}_{:,p})}{(\mathbf{g}^{[i-1]} \odot \mathbf{M}_{:,p})^H (\mathbf{g}^{[i-1]} \odot \mathbf{M}_{:,p})} g_p. \quad (17)$$

Introducing the weight vector $w_p = \mathbf{M}_{:,p}^* \odot \mathbf{M}_{:,p}$, we can read the products in the numerator and denominator as weighted inner products and write the basic iteration as

$$g_p^{[i]} = \frac{\langle \mathbf{g}^{[i-1]}, \mathbf{g} \rangle_{w_p}}{\langle \mathbf{g}^{[i-1]}, \mathbf{g}^{[i-1]} \rangle_{w_p}} g_p. \quad (18)$$

The initial estimate $g^{[i-1]}$ can be written in terms of a scaling α of the true gain values \mathbf{g} and an error vector orthogonal to \mathbf{g} (in the usual Euclidean sense) ϵ , i.e., we may write $g^{[i-1]} = \alpha(\mathbf{g} + \epsilon)$. Substitution in (18) gives

$$g_p^{[i]} = \frac{1}{\alpha^*} \frac{\langle \mathbf{g} + \epsilon, \mathbf{g} \rangle_{w_p}}{\|\mathbf{g} + \epsilon\|_{w_p}^2} g_p = \frac{1}{\alpha^*} \beta_p g_p. \quad (19)$$

This formulation gives interesting insight into the operation of the basic iteration. If the initial estimate is purely a scaling of the true value, $\beta_p = 1$, and the algorithm only tries to adjust the amplitude. If the initial estimate has a component that is orthogonal to the true gain vector, the algorithm tries to remove ϵ by projecting the guessed gain vector on the true gain vector. The impact of ϵ depends on the element being considered, because the scene used for calibration may be such that the calculation of β_p involves geometry in a weighted Euclidean space.

Introducing the vector $\beta = [\beta_1, \dots, \beta_p]^T$, we can write the full iteration for a single element as

$$g_p^{[i]} = \frac{1}{2} \left(\frac{\langle \frac{1}{\alpha^*} \beta \odot \mathbf{g}, \mathbf{g} \rangle_{w_p}}{\|\frac{1}{\alpha^*} \beta \odot \mathbf{g}\|_{w_p}^2} g_p + \frac{1}{\alpha^*} \beta_p g_p \right). \quad (20)$$

For the gain vector, the full iteration is described by

$$\mathbf{g}^{[i+1]} = \frac{1}{2} \left(\alpha \tilde{\beta} \odot \mathbf{g} + \frac{1}{\alpha^*} \beta \odot \mathbf{g} \right) = \frac{1}{2} \left(\frac{|\alpha|^2 \tilde{\beta} + \beta}{\alpha^*} \right) \odot \mathbf{g}, \quad (21)$$

where we defined $\tilde{\beta}_p = \langle \beta \odot \mathbf{g}, \mathbf{g} \rangle_{w_p} / \|\beta \odot \mathbf{g}\|_{w_p}^2$ (note the similarity in form and hence interpretation as β_p) and the vector $\tilde{\beta} = [\tilde{\beta}_1, \dots, \tilde{\beta}_p]^T$ for brevity of notation. Although not recognizable in this equation, the initial estimate $\mathbf{g}^{[i-1]}$ comes in via α , β and $\tilde{\beta}$.

For convergence, we like to show that, if we have two distinct initial estimates $\mathbf{g}_1^{[0]}$ and $\mathbf{g}_2^{[0]}$, we have

$$\|\mathbf{g}_1^{[2]} - \mathbf{g}_2^{[2]}\| \leq \|\mathbf{g}_1^{[0]} - \mathbf{g}_2^{[0]}\|, \quad (22)$$

where we used the Euclidian norm (induced norm) as distance measure in the linear space of potential gain vectors. Since we are considering the change in Euclidian distance between these two initial estimates, we can attribute the differential error vector to one of the gains without loss of generality, and model the initial estimates as $\mathbf{g}_1^{[0]} = \alpha_1(\mathbf{g} + \epsilon)$ and $\mathbf{g}_2^{[0]} = \alpha_2\mathbf{g}$, leading to

$$\left\| \frac{1}{2} \left(\frac{|\alpha_1|^2 \tilde{\beta}_1 + \beta_1}{\alpha_1^*} - \frac{|\alpha_2|^2 \tilde{\beta}_2 + \beta_2}{\alpha_2^*} \right) \odot \mathbf{g} \right\| \leq \|\alpha_1(\mathbf{g} + \epsilon) - \alpha_2\mathbf{g}\|. \quad (23)$$

Since $\epsilon = 0$ for $\mathbf{g}_2^{[0]}$, we have $\beta_2 = \tilde{\beta}_2 = \mathbf{1}$. This allows us to simplify the lefthand side of (23) to

$$\begin{aligned} & \left\| \frac{1}{2} \left(\frac{\alpha_2^* |\alpha_1|^2 \tilde{\beta}_1 + \alpha_2^* \beta_1 - \alpha_1^* |\alpha_2|^2 - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right) \odot \mathbf{g} \right\| = \\ & \frac{1}{2} \left\| \text{diag} \left(\alpha_1 \tilde{\beta}_1 - \alpha_2 + \frac{\alpha_2^* \beta_1 - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right) \mathbf{g} \right\| \\ & \leq \frac{1}{2} \sigma_{\max} \left(\text{diag} \left(\alpha_1 \tilde{\beta}_1 - \alpha_2 + \frac{\alpha_2^* \beta_1 - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right) \right) \|\mathbf{g}\|, \end{aligned} \quad (24)$$

where $\sigma_{\max}(\cdot)$ denotes the largest singular value of a matrix. For a diagonal matrix \mathbf{D} , such as that in (24), we have:

$$\sigma_{\max}^2(\mathbf{D}) = \lambda_{\max}(\mathbf{D}^H \mathbf{D}) = \max_n \{|d_n|^2\}, \quad (25)$$

where d_n denotes the n th element on the main diagonal of \mathbf{D} .

Squaring the left- and righthand sides of (22) and exploiting the fact that $\epsilon \perp \mathbf{g}$, we require

$$\begin{aligned} & \frac{1}{4} \left| \alpha_1 \tilde{\beta}_{1,p_{\max}} - \alpha_2 + \frac{\alpha_2^* \beta_{1,p_{\max}} - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right|^2 \\ & \leq |\alpha_1 - \alpha_2|^2 + |\alpha_1|^2 \frac{\|\epsilon\|_{w_p}^2}{\|\mathbf{g}\|_{w_p}^2} \end{aligned} \quad (26)$$

for convergence, where p_{\max} is the value of p that maximizes the lefthand side.

4.2. Convergence for single point source calibration

When the observed scene consists of a single point source, we can assume $\mathbf{M} = \mathbf{R}_0 = \mathbf{1}\mathbf{1}^H$ without loss of generality. The weighted inner products in β and $\tilde{\beta}$ therefore reduce to the standard Euclidean inner product since $w_p = \mathbf{1}$ for all p . Since the

error vector ϵ is assumed to be perpendicular to \mathbf{g} , (19) shows that ϵ will be projected out in the first basic iteration. As a result, we have $\beta_p = 1$ for all p after the first iteration, which reduces the condition for convergence in (26) to

$$\frac{1}{4} \left| \alpha_1 - \alpha_2 + \frac{\alpha_2^* - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right|^2 \leq |\alpha_1 - \alpha_2|^2. \quad (27)$$

Substitution of

$$\begin{aligned} \left| \alpha_1 - \alpha_2 + \frac{\alpha_2^* - \alpha_1^*}{\alpha_1^* \alpha_2^*} \right|^2 &\leq \left(|\alpha_1 - \alpha_2| + \frac{|\alpha_2^* - \alpha_1^*|}{|\alpha_1^*| |\alpha_2^*|} \right)^2 \\ &= |\alpha_1 - \alpha_2|^2 + \frac{(2|\alpha_1| |\alpha_2| + 1) |\alpha_1 - \alpha_2|^2}{|\alpha_1|^2 |\alpha_2|^2} \end{aligned} \quad (28)$$

in (27), followed by some algebraic manipulation, gives

$$2|\alpha_1| |\alpha_2| + 1 \leq 3|\alpha_1|^2 |\alpha_2|^2, \quad (29)$$

which holds if $|\alpha_1| |\alpha_2| \geq 1$. This may not be true for the initial estimate provided to the algorithm, but as long as the initial estimate does not lie in the null space of \mathbf{g} , this condition will be met after the first full iteration, since $\frac{1}{2} |1/\bar{\alpha} + \alpha| \geq 1$ for all values α satisfying $|\alpha| \geq 0$. Equation (29) also shows that if $|\alpha_1|$ and $|\alpha_2|$ are close to unity, i.e., if the estimates are close to the true value, the rate of convergence becomes quite slow. Slow convergence close to the true solution has indeed been observed in our simulations.

Another interesting insight from this analysis lies in the distribution of consecutive estimates around the true value. Each full iteration involves a gain estimate that scales the true gain vector with $|\alpha|$ and a gain estimate scaling $1/|\alpha|$ with $|\alpha|$ moving closer to unity in each full iteration. The algorithm thus generates two sequences of points that converge to the true value, one from above and one from below.

4.3. Convergence in general

To assess the convergence in the general case of calibration on an arbitrary scene, we first note that the condition for convergence in (26) is tightest if either one of the terms on the righthand side equals zero. We therefore analyze those two extreme cases. In the first case, $\epsilon = \mathbf{0}$, we have $\beta_1 = \tilde{\beta}_1 = 1$. This leads to the condition expressed in (27), for which convergence was discussed in the previous subsection.

In the second case $\alpha_1 = \alpha_2 = \alpha$, making the first term of the righthand side of (26) zero, such that the condition for convergence holds if

$$\frac{1}{4} \left| \alpha (\tilde{\beta}_{1,p_{\max}} - 1) + \frac{1}{\alpha} (\beta_{1,p_{\max}} - 1) \right|^2 \leq |\alpha|^2 \frac{\|\epsilon\|^2}{\|\mathbf{g}\|^2}. \quad (30)$$

Since $\beta_{p_{\max}}$ is based on a weighted inner product instead of the usual Euclidean inner product (as in the single point source case), we cannot show that (30) holds in general. The significance of this is that it is possible to construct cases for which the algorithm fails. For example, when $\mathbf{M}_{:,p} = \mathbf{0}$ for some value of p (this holds for all p in the case of an empty scene) or if, for specific weights \mathbf{w}_p , $\|\mathbf{g} + \epsilon\|_{\mathbf{w}_p} \rightarrow 0$, $\beta_{p_{\max}}$ becomes very large.

In summary, we conclude that, to ensure convergence, the following conditions should be met:

1. The inner product between the initial estimate and the true value should be nonzero; i.e., the initial estimate should not lie in the space orthogonal to the true value or be close to the zero vector.

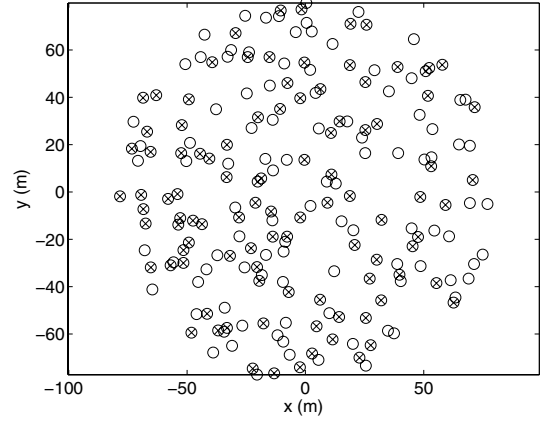


Fig. 1. Positions of the first 200 (circles) and 100 (crosses) antennas in the random configuration of 4000 antennas generated for the simulations.

2. The observed scene should be such that γ in $\langle \epsilon, \mathbf{g} \rangle_{\mathbf{w}_p} = \gamma \|\epsilon\|_{\mathbf{w}_p} \|\mathbf{g}\|_{\mathbf{w}_p}$ is small for all values of p . This ensures that $\beta_{p_{\max}}$ remains small. Physically, this means that the observed scene should be suitable for a calibration measurement. For example, it is not possible to do gain amplitude and phase calibration on a homogeneously filled or empty scene.

The first criterion can usually be ensured by available knowledge of the measurement system, either from precalibration or design. The second criterion is a requirement for any calibration measurement. We therefore conclude that the algorithm will work in most practical situations. This conclusion was confirmed by extensive testing in simulation, of which some examples will be presented in the next section.

5. Simulations

In this section we show how the algorithm performs in terms of numerical, computational, and statistical performance.

5.1. Numerical performance and practical convergence

We tested StEFCal with an extensive number of cases that cover a wide range of simulated sky models, with varying numbers and locations of receivers and levels of corruption and noise. These simulations all supported the conclusions drawn from the specific cases reported here.

The results shown here, unless otherwise indicated, employ a simulated sky consisting of 1000 point sources with powers exponentially distributed between 10^0 and 10^{-4} Jy (1 Jy equals 10^{-26} W/m²/Hz) randomly positioned in the sky. The array configuration consists of up to 4000 antennas randomly distributed over a circular range with a diameter of 160 m and minimum separation of 1.5 m, corresponding to a Nyquist frequency of 100 MHz. Where a smaller number of antennas was required, the upper left portion of the visibility matrix (associated with the first P antennas) was used, as illustrated by the antenna layouts in Fig. 1. For convenience of presentation, uncorrelated receiver noise was not included in the example illustrated in this section. The effect of noise is studied in more detail in Sect. 5.3.

The model visibility matrix was built for two cases:

- Case 1: Incomplete sky model: only the 18 brightest sources (all sources brighter than 1% of the brightest source) were included in the model visibilities.

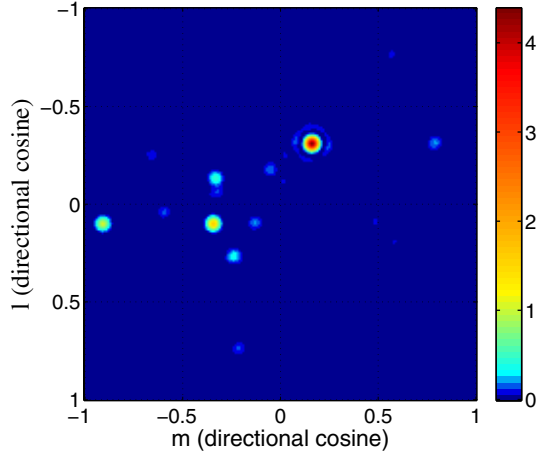


Fig. 2. Scene as observed by a perfectly calibrated array. The sky model contains 1000 sources but only the brightest are visible with this color scale. The weakest sources are even drowned in the sidelobe response of the brightest sources.

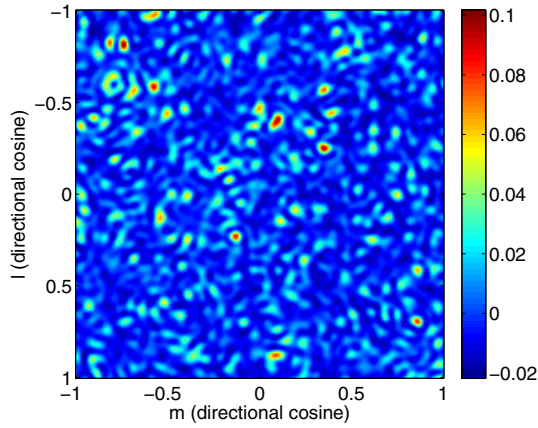


Fig. 3. Observed sky: sky image as seen by the instrument prior to calibration.

Case 2: Complete sky model: all 1000 sources were included.

Case 1 represents a typical situation in practical radio astronomical calibration scenarios. For both cases, convergence to machine accuracy (10^{-15}) and to a much looser tolerance (10^{-5}) was studied and compared to give an indication of the convergence requirements in realistic cases.

Figure 2 shows the “exact” sky, i.e. the sky image as viewed by the perfectly calibrated array. “Corrupted” skies were obtained by perturbing the antenna-based gains with amplitudes randomly distributed between 0.5 and 1.5 with unit mean and phases randomly distributed between 0 and 2π radians. The results for Case 1 are shown in Figs. 3 through 5, using $P = 500$ antennas and at a frequency of about 35.5 MHz. Figure 3 shows the sky as imaged by the instrument prior to calibration. We ignored the autocorrelations of the measured visibility matrix and the matrix of model visibilities. As discussed in Sect. 8, it is straightforward to use StEFCal in scenarios with more entries of the covariance matrix set to zero to flag specific data values. This was studied, but not reported here for reasons of brevity, and supports the conclusions drawn.

The images obtained after calibration are indistinguishable from Fig. 2 at the resolution of the array, so they are not shown here. The difference between the image after calibration and the incomplete sky model (which includes just 18 sources for

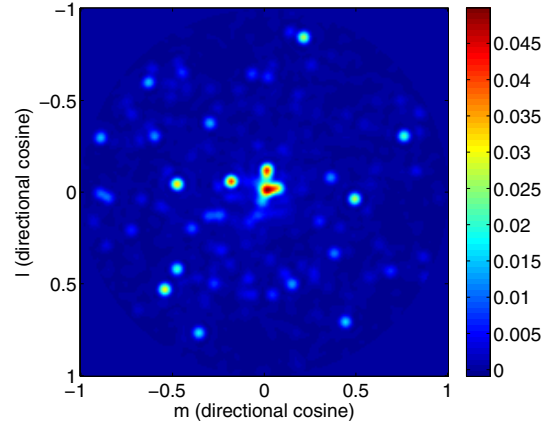


Fig. 4. Difference between the image after calibration and the model sky containing only the 18 brightest sources.

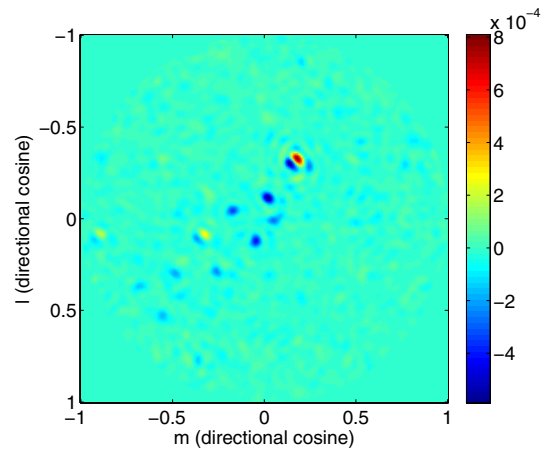


Fig. 5. Difference between the image after calibration on only the 18 brightest sources and the exact sky.

Table 2. Maximum difference between exact sky and image after calibration to different tolerances.

$\max(\text{sky}_{\text{exact}} - \text{sky}_{10^{-5}})$	3.2×10^{-8}
$\max(\text{sky}_{\text{exact}} - \text{sky}_{10^{-15}})$	1.3×10^{-15}
$\max(\text{sky}_{10^{-5}} - \text{sky}_{10^{-15}})$	3.2×10^{-8}

Case 1) is more interesting and is shown in Fig. 4. This clearly shows that the next brightest sources can be identified correctly after calibration and subtraction of the 18 brightest sources. Figure 5 shows the difference between the image after calibration and the exact sky showing that the errors due to ignoring the weak sources in the model are two orders of magnitude smaller than the next brightest sources. This shows that StEFCal can be used as an algorithmic component in self-calibration procedures based on iterative source model refinement.

We also carried out simulations with different settings for the tolerance for convergence. The results for Case 2 (complete sky model) are reported here. Table 2 summarizes the results for tolerances of 10^{-5} and 10^{-15} . In the table, we show the maximum difference between the absolute value of the image pixels between the exact sky and the image after calibration for both convergence tolerances. The maximum difference we found is on the order of 10^{-8} , which is well below the noise level. This led us to conclude that effective convergence can be achieved with limited effort and reasonably “loose” convergence requirements.

Table 3. Algorithm measured performance.

P	N_{it} fixed		Tolerance 10^{-5}		
	N_{it}	time	N_{it}	time	convergence
50	40	0.0006	12	0.0002	3.5E-06
100	40	0.002	14	0.001	4.9E-06
200	40	0.008	16	0.003	1.9E-06
300	40	0.017	16	0.007	3.6E-07
400	40	0.036	16	0.014	2.6E-06
500	40	0.062	18	0.028	3.0E-08
600	40	0.093	18	0.042	1.7E-08
800	40	0.220	18	0.099	5.5E-07
1000	40	0.258	18	0.116	3.8E-08
1500	40	0.579	18	0.260	1.9E-07
2000	40	1.029	20	0.515	4.5E-09
3000	40	2.321	20	1.169	1.6E-08
4000	40	4.108	20	2.059	1.6E-07

Notes. In the table, N_{it} is the number of iterations and all times are in seconds.

The table also reports the difference between two images obtained after calibration for the two tolerances.

As already mentioned, we have run a long series of tests with the number of antennas ranging from 20 to 4000, a variety of source models, with and without adding antenna-based noise. All cases showed similar outcomes to those reported here. This supports the suitability of StEFCal for practical use in terms of numerical performance and speed of convergence. The number of iterations required appears to depend only very weakly on the number of receivers. Typically, in the cases reported here, irrespective of the number of antennas, 10^{-5} or better convergence could be achieved in 20 iterations or less and 10^{-15} convergence in 40 iterations or less. Benchmarks are reported in Sect. 5.2.

5.2. Computational performance

The incomplete source model introduced in Sect. 5.1 (Case 1) was used in the computational benchmarks reported here, although far more extensive tests were carried out. In all cases, StEFCal has shown very good performance characteristics despite its iterative nature. This requires a refresh of the data for each iteration for problems too large to fit in cache, as is the case for any other iterative approach.

A number of factors underpin StEFCal's performance:

- All computations are carried out through vector operations.
- Data are accessed by unit-stride, contiguous memory patterns. This ensures maximum utilization of data loaded, ensuring full use of cache lines, as well as emphasizing the role of prefetching.
- StEFCal requires only a modest memory footprint (as discussed below).

We coded the algorithm in Fortran 90, compiled using the Intel MKL Fortran compiler version 11.00. We used either Intel MKL BLAS or handwritten code. The difference between these two versions was 3% at most, so we only report on the results using the Intel MKL library. We used a desktop system with an Intel dual core Core 2 running at 3.0 GHz, single threaded (only one core active), with single-threaded MKL BLAS.

We would like to mention that multithreaded parallelism over frequency channels and/or time slices have also been developed and tested to very good effect. Table 3 and Fig. 6 report the performance obtained for arrays with 50 to 4000 antennas. The

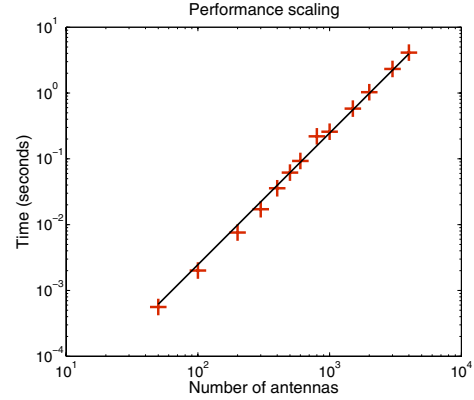


Fig. 6. Algorithm scaling on a logarithmic scale. The red crosses denote the measured computing times; the black line shows the expected computing times for quadratic scaling, normalized to $P = 500$: $T_P = (\frac{P}{500})^2 \cdot T_{500}$.

figure compares the measured computing times against those expected for perfect quadratic scaling with the number of antennas, normalized to $P = 500$. Both the table and figure highlight the efficiency of the algorithms, as well as its quadratic scalability over the number of elements.

One full iteration requires the element-wise product of two complex vectors and two complex dot products per gain parameter. In terms of *real valued* floating point operations, this is equivalent to $24P^2$ flop, where each complex multiply-add requires 8 flop. One dot product corresponds to computing the square of the 2-norm (or F-norm) and the cost of that dot product can be halved, leading to a grand total per iteration of $20P^2$ flop. The algorithm does indeed require $O(P^2)$ operations as in our initial claim.

The system used for benchmarking had a peak speed of 12 Gflops, and the peak speed observed for MKL DGEMM was ~ 11 Gflops. StEFCal showed performance figures of over 3 Gflops. Given the nature of the computation, both the speed and the scalability observed are very good (over 25% peak speed). Tests have shown that even better performance can be obtained on more modern CPUs.

The memory footprint of StEFCAL is modest:

- The measured visibilities and the model visibilities are complex valued $P \times P$ matrices, requiring $16P^2$ bytes per matrix for storage in double-precision floating-point format. Thanks to their Hermiticity, one could store these matrices in compressed triangular storage format. However, this would require accessing their elements with non-unit variable strides, thus considerably lowering computational performance: given the memory available on current systems, performance issues are overriding.
- One complex valued vector of length P is returned as output.
- One internally allocated complex valued vector of length P is used.
- Depending on code internals, other complex vectors of length P may be required.

The total amount of input and output data ($32P^2 + 16P$ bytes assuming double precision floating point format) results in the low computational intensity of $24P^2 N_{it} / (32P^2 + 16P)$ flop per byte or approximately $3N_{it}/4$ flop per byte for high values of P . Thus StEFCal is memory bound, as observed in practice. As already mentioned, this is greatly ameliorated by the memory access pattern. While traditional $O(P^3)$ methods may increase the number

of operations per memory transfer, they also increase the number of operations by the same amount, thus resulting in much lower performance overall.

5.3. Statistical performance

We performed a series of Monte Carlo simulations to assess the statistical performance and robustness of the proposed algorithm. We have defined three scenarios:

1. Calibration on two weak (S/N of -10 dB per time sample) point sources.
2. Calibration on a realistic sky model.
3. Calibration on two strong (S/N of 20 dB per time sample) point sources.

By choosing these scenarios, we try to explore the sensitivity of StEFCal to violation of the assumption that $\mathbf{R} \approx \sigma_n \mathbf{I}$, which was used to derive the algorithm from the weighted least squares cost function and hence get a feel for the range of applicability of the algorithm. The antenna configuration used in these simulations is the 200-element configuration shown in Fig. 1.

Since we can only estimate the phase difference between the antennas, we assume that the first antenna will be used as phase reference. We can therefore define the $(3P - 1) \times 1$ vector of free parameters $\boldsymbol{\theta} = [\gamma_1, \dots, \gamma_P, \varphi_2, \dots, \varphi_P, \sigma_{n,1}, \dots, \sigma_{n,P}]^T$, where γ_p , φ_p , and $\sigma_{n,p}$ are the gain amplitude, gain phase, and the noise power of the p th element, respectively. Expressions for the Cramer-Rao bound (CRB), the minimum achievable variance for an unbiased estimator (Kay 1993; Moon & Stirling 2000), for this scenario is derived by Wijnholds & van der Veen (2009).

5.3.1. Calibration on two weak point sources

In this scenario, two sources with source power $\sigma_q = 1$ for $q = 1, 2$ were located at $(l_1, m_1) = (0, 0)$ (field center or zenith) and $(l_2, m_2) = (0.4, 0.3)$, where l and m are direction cosines. We set the measurement frequency to 60 MHz ($\lambda = 5.0$ m) and defined $\sigma_n = 10$ for all antennas, such that both sources have an S/N of -10 dB per time sample. This ensures that the assumption $\mathbf{R} \approx \sigma_n \mathbf{I}$, which we used to derive the algorithm from the weighted LS cost function, holds. In this Monte Carlo simulation, we calibrated the data using StEFCal, as well as the multisource calibration algorithm proposed in Wijnholds & van der Veen (2009), to compare the two approaches in terms of statistical and computational efficiency. Simulations were done for $K = \{10^3, 3 \times 10^3, 10^4, 3 \times 10^4, 10^5, 3 \times 10^5, 10^6\}$ time samples and each simulation was repeated 100 times. For Nyquist-sampled time series, the number of samples is equal to the product of bandwidth and integration time, i.e., $K = B\tau$. The chosen range of values for K thus covers the most commonly used range of values for bandwidth and integration time in radio astronomical calibration problems with high spectral and temporal resolution.

The biases found in the simulations are considerably smaller than the standard deviation for this scenario based on the CRB. This indicates that our algorithm is unbiased, hence that a comparison with the CRB is meaningful to assess the statistical performance of the algorithm. Figure 7 shows the variance of the estimated gain amplitude and phase parameters for $K = 10^6$, clearly showing that both algorithms achieve the CRB for large K . Figure 8 shows the variance for a representative complex valued gain estimate for all simulated values of the number of samples K . The result indicates that the CRB is already

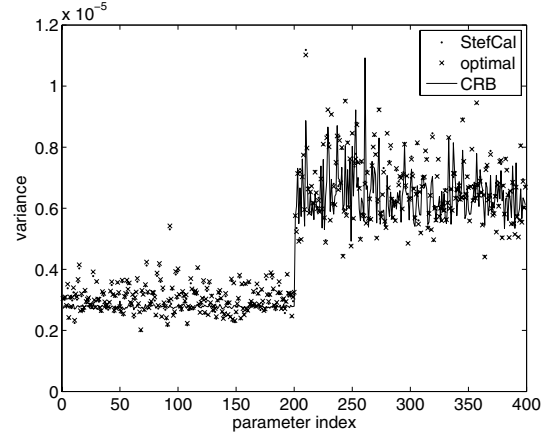


Fig. 7. Variance on the estimated gain amplitudes (indices 1 through 200, dimensionless) and phases (indices 201 through 399, in units of rad^2) parameters for calibration on two point sources with a S/N of -10 dB. The solid line indicates the CRB.

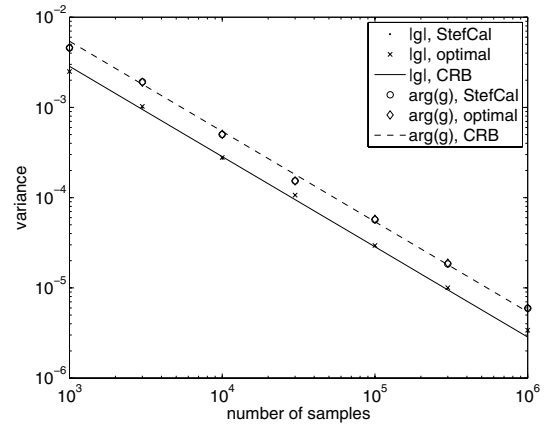


Fig. 8. Variance (dimensionless for amplitude parameters, in units of rad^2 for phases) of a representative complex valued gain estimate ($p = 10$) as function of the number of time samples. The lines mark the CRB for the two parameters involved.

achieved for very low values of K . Based on the theory of random matrices, matrix-wise convergence of the covariance matrix estimate starts when the number of samples is about ten times bigger than the number of elements (Couillet & Debbah 2011), which, in the case of a 200-element array, would be at $K \approx 2 \times 10^3$. The proposed algorithm does not rely on mathematical operations that depend on matrix-wise convergence to work properly, and this may provide an intuitive explanation for this attractive feature of StEFCal.

The simulation results indicate that StEFCal achieves statistically optimal performance when $\mathbf{R} \approx \sigma_n \mathbf{I}$ and thus has statistical performance similar to statistically efficient methods, such as the algorithm described by Wijnholds & van der Veen (2009) or optimization of the cost function using the Levenberg-Marquardt solver. However, the proposed algorithm has only $O(P^2)$ complexity instead of the $O(P^3)$ complexity of many commonly used methods. This should give a significant reduction in computational cost of calibration, especially for large arrays. The Monte Carlo simulations described here were done in Matlab on a single core of a 2.66 GHz Intel Core i7 CPU. Gain calibration for a single realization took, on average, 2.24 s when using the method described by Wijnholds & van der Veen (2009) while taking only 0.12 s when using StEFCal.

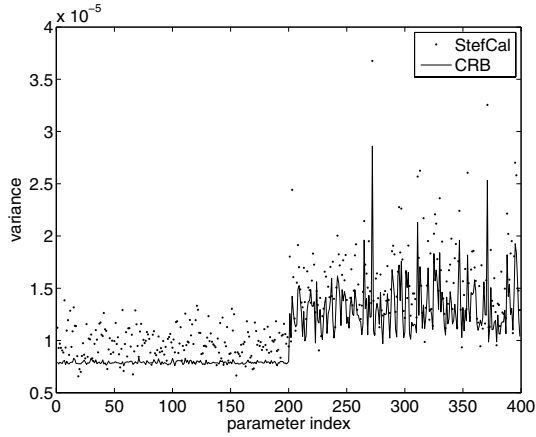


Fig. 9. Variance on the estimated gain amplitudes (indices 1 through 200, dimensionless) and phases (indices 201 through 399, in units of rad^2) parameters for calibration on the scene shown in Fig. 2 with noise power equal to the integrated power of all sources.

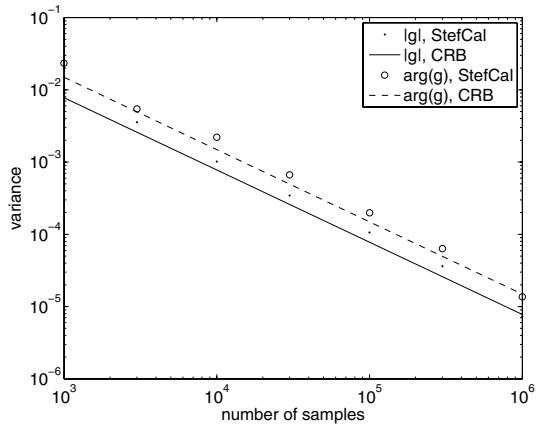


Fig. 10. Variance (dimensionless for gain amplitude, in units of rad^2 for gain phase) of a representative complex valued gain estimate ($p = 20$) as function of the number of time samples. The lines mark the CRB of the two corresponding parameters.

5.3.2. Calibration on a realistic scene

In many array applications, the scene on which the array needs to be calibrated in the field is considerably more complicated than one or just a few point sources. To see how the algorithm performs in a more realistic scenario, we used the scene shown in Fig. 2 for calibration. We are still in the low-S/N regime, with the noise power in each antenna being about ten times the total power in the scene and the strongest sources having an S/N per sample of -13.3 dB, so $\mathbf{R} \approx \sigma_n \mathbf{I}$ still holds.

We set up our Monte Carlo simulations in the same way as for the first scenario. After checking that the algorithm produced unbiased results, we compared the variance on the estimated parameters with the CRB. The results are shown in Figs. 9 and 10. They indicate that the performance of StEFCal is still very close to statistically optimal.

5.3.3. Calibration on two strong point sources

For our last scenario, we defined a simulation with two point sources located at $(l_1, m_1) = (0, 0)$ and $(l_2, m_2) = (0.4, 0.3)$ with an S/N of 20 dB per time sample. This is a scenario that clearly

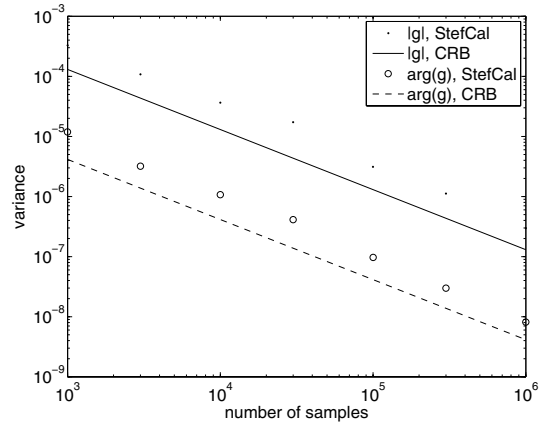


Fig. 11. Variance (dimensionless for gain amplitude, in units of rad^2 for gain phase) of a representative complex valued gain estimate ($p = 5$) as function of the number for time samples. The lines mark the CRB of the two corresponding parameters.

violates the assumption that $\mathbf{R} \approx \sigma_n \mathbf{I}$. We performed the Monte Carlo simulation in the same way as in the previous cases.

Figure 11 shows the variance of the gain and phase associated with a representative element as function of the number of samples. The gain estimates, while not as close to the CRB as in the previous cases, are still quite close to the bound. The average gain amplitude error is still only 55% higher than the CRB, while the average phase error is only 26% higher than the CRB. This is acceptable given the high accuracy achieved in such a high-S/N regime. We conclude that StEFCal provides a performance that is close to optimal, even in scenarios designed to break the underlying assumptions made to use the LS rather than the WLS cost function. This shows that the algorithm is fairly robust in terms of its statistical performance and will provide statistically efficient estimates in scenarios typical of radio astronomy.

6. Extension to full polarization calibration

It is straightforward to apply the ADI approach to the full polarization case as demonstrated by Hamaker (2000) and Mitchell et al. (2008). Initial results for StEFCal have been presented by Salvini & Wijnholds (2014a) and Salvini & Wijnholds (2014b) and confirm the validity of StEFCal for full polarization calibration, still retaining $\mathcal{O}(P^2)$ computational complexity. In this section, we sketch the StEFCal algorithm for the full polarization case. A full analysis will be provided in a future paper.

The mathematical problem is structured in terms of matrices whose elements are two-by-two complex blocks, rather than by individual complex values. In particular, the gain matrix \mathbf{G} is a block diagonal matrix whose 2×2 blocks on the main diagonal describe the response of the two feeds of each receiver:

$$\mathbf{G}_p = \begin{bmatrix} \mathbf{G}_{2p-1,2p-1} & \mathbf{G}_{2p-1,2p} \\ \mathbf{G}_{2p,2p-1} & \mathbf{G}_{2p,2p} \end{bmatrix}. \quad (31)$$

Taking this structure into account, the full polarization calibration problem can still be formulated as

$$\widehat{\mathbf{G}} = \underset{\mathbf{G}}{\operatorname{argmin}} \left\| \widehat{\mathbf{R}} - \mathbf{G} \mathbf{M} \mathbf{G}^H \right\|_F^2. \quad (32)$$

It naturally follows that the basic step of full polarization StEFCal consists of solving $P \times 2 \times 2$ linear least squares problems for each iteration, within the same StEFCal iteration framework as for the scalar case described in this paper.

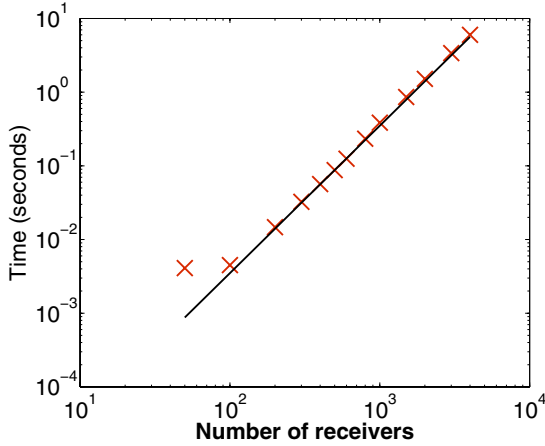


Fig. 12. Full polarization algorithm scaling on a logarithmic scale. The red crosses denote the measured computing times; the black line shows the expected computing times for quadratic scaling, normalized to $P = 500$: $T_P = \left(\frac{P}{500}\right)^2 \cdot T_{500}$.

Table 4. Full polarization StEFCal performance (see text). Times are in seconds.

N. cores	Time	GFlops/s	% CGEMM
1	16.63	17.9	41.8
2	8.34	35.8	41.9
3	5.57	53.5	42.2
4	4.19	71.2	41.8
8	2.90	102.9	42.0
10	2.35	127.0	42.2
12	1.92	155.3	41.3
16	1.72	173.3	38.3

Notes. In the table, all times are in seconds.

In general, the simple StEFCal algorithm, which has proved very successful for the scalar case, exhibits slow or difficult convergence. As shown by Salvini & Wijnholds (2014b), this can be corrected by employing a multistep approach, whereby the two previous solutions at the even steps are also included in the averaging process. Moreover, some heuristics can be employed to regularize the convergence rate.

Performance again proved very good, since the same considerations as for the scalar algorithm apply. Since the density of operations increases by a factor two per data item, a marginally better speed has been obtained, in terms of Gflop per second. An example of performance results is shown in Table 4. This involved a realistic scenario of full polarization calibration of the proposed SKA LFAA station, comprising of 256 antennas (512 dipoles) for 1024 frequencies. The code was parallelized over frequencies using OpenMP, whereby each core grabs the first available frequency still to be calibrated (dynamic load balancing). All computations were carried out using single precision to a tolerance of 10^{-5} , delivering better than 1% accuracy in the complex gains, as required. Performance figures are compared to the performance of MKL CGEMM (complex matrix–complex matrix product), which virtually runs at peak speed and gives a good indication of maximum speeds achievable.

Figure 12 shows that scalability with problem size has very similar characteristics to those for the scalar problem. In this simulation, we fixed the number of iterations to 100 and compared the measured data against exact P^2 scalability, normalized to $P = 500$. It should be noted that the number of operations per iteration now reads as $48P^2$. We also like to point out that

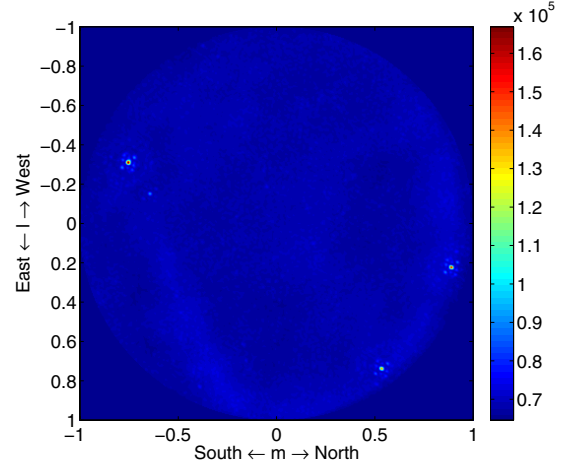


Fig. 13. Calibrated all-sky image at 59.67 MHz made with the 288-antenna AARTFAAC system.

the convergence rate, i.e. the number of iterations required for a given accuracy, exhibits the same very weak dependence on problem size, i.e. the number of receivers, as in the case of scalar StEFCal.

7. Applications

Figure 13 shows a calibrated all-sky map produced by the Amsterdam-ASTRON Radio Transient Facility and Analysis Centre (AARTFAAC¹, Prasad & Wijnholds 2013). This system is a transient monitoring facility installed on the six innermost stations of the LOFAR, which this facility combines as a single station with 288 antennas spread over an area with a diameter of about 350 m. In this section, we describe the integration of StEFCal in the AARTFAAC calibration pipeline to demonstrate the ease of integration of the proposed algorithm in an existing pipeline and the computational benefits.

The calibration of the AARTFAAC system involves estimating the direction-independent complex valued gains of the receiving elements, the apparent source powers of the four bright point sources seen in Fig. 13, and a non-diagonal noise covariance matrix to model the diffuse emission seen in the image and the system noise. The non-diagonal noise covariance matrix was modeled with one complex valued parameter for every entry associated with a pair of antennas that were less than ten wavelengths apart and a real valued parameter for every entry on the main diagonal. This calibration challenge was solved using the weighted alternating least squares (WALS) algorithm described by Wijnholds (2010). In each main iteration of the WALS method, the direction-independent gains are first estimated assuming that the other parameters are known, then the source powers are updated, and finally the noise covariance matrix is updated. To calibrate the AARTFAAC data set used to produce Fig. 13, six main iterations were required. The middle column of Table 5 shows the average time estimation of each group of parameters took in Matlab on an Intel Core i7 CPU on a machine with 4 GB RAM.

StEFCal was easily integrated in the WALS algorithm by simply replacing the gain estimation step (which used an algorithm of $O(P^3)$ complexity) with the StEFCal algorithm. StEFCal was configured to iterate to convergence in each main loop of the WALS method. Table 5 reports the computational

¹ www.aartfaac.org

Table 5. Processing time in seconds per main iteration of array calibration using the original WALs as described by Wijnholds (2010) and WALs with StEFCal.

Parameter	Original	With StEFCal
Gain	6.528	0.184
Source power	0.023	0.030
Noise covariance	0.003	0.003
Total	6.554	0.217

times, showing that StEFCal resulted in an increase in performance of over a factor 30 when “cold starting”, i.e. without any prior information for any timeslice. As gains are expected to vary smoothly over time, a further eight-fold increase in performance was obtained by using the results from the previous timeslice as initial guess in the calibration of each snapshot (a factor 250+ overall). This underscores the capability of StEFCal to make good use of initial gain estimates.

The full polarization version of StEFCal is currently employed in MEqTrees (Noordam & Smirnov 2010). It is being implemented in the standard LOFAR pre-processing pipeline and studied for the SKA. As an example, the LOFAR central processor requires a number of steps including direction-independent gain calibration (station-based) to provide initial corrections for clock drift and propagation effects. The latter are mainly caused by the ionosphere and may require full polarization corrections on baselines to stations outside the core area. Recently, Dijkema (2014) has implemented the basic version of full polarization StEFCal for the standard processing pipeline of LOFAR. This implementation was used to run the same pipeline on several data sets from actual LOFAR observations twice, once with the standard Levenberg-Marquardt (LM) solver and once with the LM solver replaced by StEFCal. In all cases, the results obtained were practically identical, but the pipeline with StEFCal was typically a factor 10 to 20 faster than the pipeline running the LM solver. Based on the material presented in this paper, we expect that we can improve performance significantly by optimizing the implementation of StEFCal used.

8. Discussion and future work

8.1. Other variants of StEFCal

We also studied a variant of StEFCal with relaxation, in which the complex gains are used as soon as they become available, rather than using the full set of complex gains from the previous iteration; i.e. the gain vector gets updated while looping over all receivers and is then applied immediately. This variant is listed in Algorithm 2. In general, this variant needs fewer iterations. However, the receiver loop (the p -loop) for each iteration has parallel dependencies, which makes this variant much less portable to multicore and many-core platforms, such as GPUs, although it could be valuable for more traditional CPUs.

Numerical performance appears very close to the standard StEFCal, shown as Algorithm 1, but we have not attempted to obtain a formal proof of convergence. Figure 14 shows the faster convergence of Algorithm 2, in particular at the beginning of the iteration.

Another variant of Algorithm 2, whose details are not reported here, aims to decrease the parallel dependencies by block-wise updates of the estimated gain vector (thus the latest values of the previous block are used, while the old values of the current

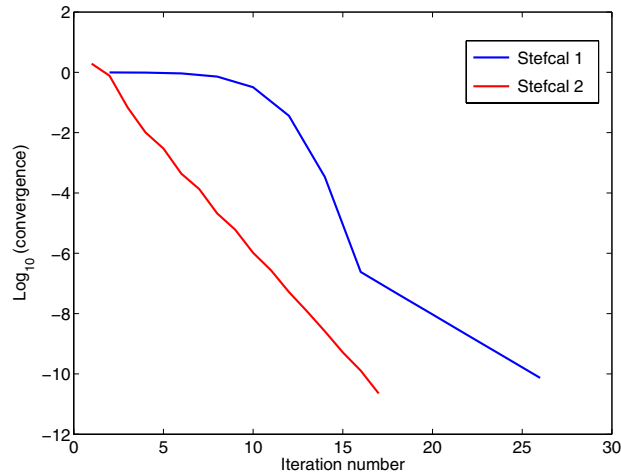


Fig. 14. Comparing algorithm performance for $P = 500$.

Algorithm 2 Algorithm StEFCal2

```

Initiate  $\mathbf{G}^{[0]}$ ;  $\mathbf{G}^{[0]} = \mathbf{I}$  is adequate in most cases
for  $i = 1, 2, \dots, i_{\max}$  do
   $\mathbf{G}^{[i]} = \mathbf{G}^{[i-1]}$ 
  for  $p = 1, 2, \dots, P$  do
     $\mathbf{z} \leftarrow \mathbf{G}^{[i]} \cdot \mathbf{M}_{:,p} \equiv \mathbf{g}^{[i]} \odot \mathbf{M}_{:,p}$ 
     $\mathbf{g}_p^{[i]} \leftarrow (\widehat{\mathbf{R}}_{:,p}^H \cdot \mathbf{z}) / (\mathbf{z}^H \cdot \mathbf{z})$ 
  end for
  if  $\|\mathbf{g}^{[i]} - \mathbf{g}^{[i-1]}\|_F / \|\mathbf{g}^{[i]}\|_F \leq \tau$  then
    Convergence reached
  end if
end for

```

block are used). As expected, performance falls in between the full relaxation and the original algorithm. It is felt by the authors that loss of parallelism is more important than a rather modest gain in performance.

The first version of StEFCal included an initial stage (again with operation count $O(P^2)$), which purified the largest eigenvalues and vectors of the observed visibilities (including estimation of the autocorrelation terms), and then matched these against the corresponding ones in the model sky. This worked very well and was very fast, when the number of bright sources present in the field of view was much smaller than the array size P . This variant was presented at various meetings but was then dropped because of the simplicity and power of the current version of StEFCal.

8.2. Extension to iterative reweighted least squares

In Sect. 2.1, it was pointed out that appropriate weighting of each data point may be required to reduce the effects of outliers. Typically, this is done by assigning weights to the data values based on their reliability or use a different norm, for example the 1-norm, that is less sensitive to outliers. To handle such cases, we have developed a variant of StEFCal that follows an iterative reweighted least squares (IRLS) approach (Moon & Stirling 2000). In an IRLS algorithm, the data values are weighted so that the 2-norm minimization becomes equivalent to minimization using another norm. In our example below, we minimize the 1-norm of the residuals. The resulting algorithm still has $O(P^2)$ complexity, but the individual iterations require more operations to calculate and apply the weights.

We aim to minimize the 1-norm of the residuals, which is not the matrix 1-norm but the sum of the absolute values of all data points

$$\|\Delta\|_1 = \sum_q \sum_p |\widehat{\mathbf{R}}_{q,p} - g_q \mathbf{M}_{q,p} g_p^*|, \quad (33)$$

where the indices p and q run over the receiving elements. The kernel of the algorithm is modified by using the weights

$$\mathbf{W}_{q,p} = \frac{1}{|\widehat{\mathbf{R}}_{q,p} - g_q \mathbf{M}_{q,p} g_p^*|} \quad (34)$$

with appropriate checks and actions for very small (or zero) entries in the denominator.

We can apply these weights in the basic StEFCal iteration by replacing (12) by

$$g_p^{[i]} = \frac{\widehat{\mathbf{R}}_{:p}^H \cdot (\mathbf{W}_{:p} \odot \mathbf{Z}_{:p}^{[i-1]})}{(\mathbf{Z}_{:p}^{[i-1]})^H (\mathbf{W}_{:p} \odot \mathbf{Z}_{:p}^{[i-1]})} \quad (35)$$

and applying corresponding changes to Algorithm 1. At the end of each iteration, the appropriate weights would need to be computed using (34).

As an example, we consider the calibration in the 2- and 1-norm of test data for 200 receivers with a tolerance of 10^{-7} . The 2-norm calibration required 28 iterations, while the 1-norm calibration required 52 iterations. Computational costs per iteration were higher by a factor 1.5. This factor can be explained by the increased number of operations required for relatively expensive calculations like taking the absolute value of a complex number. An interesting question, which is beyond the scope of this paper, is whether it is necessary to do 1-norm optimization until convergence or whether the weights can be fixed after a limited number of iterations when the solution is sufficiently close to the optimum. Such an approach would reduce computational costs by avoiding the recalculation of weights from a certain point in the iteration process.

8.3. Integration of StEFCal in other algorithms

In Sect. 7, we saw an example of how StEFCal was integrated in an existing calibration pipeline. This particular example involved a non-diagonal noise covariance matrix that was modeled by introducing an additional noise parameter for each off-diagonal entry of the noise covariance matrix that was assumed to be non-zero. In this paper, we set the diagonal entries of the array covariance matrix $\widehat{\mathbf{R}}$ and the model covariance matrix \mathbf{M} to zero. We could easily accommodate the estimation of the non-diagonal noise covariance matrix by setting not only the entries associated with the autocorrelations to zero, but also the entries associated with the non-zero off-diagonal entries of the noise covariance matrix. We can use the same procedure to account for corrupted or missing data or for when short baselines should be excluded. Of course, this should not be done unnecessarily, because exclusion of potentially useful information from the gain estimation process will degrade the gain estimation performance.

Estimation of direction-dependent gains is currently a hot topic in radio astronomy (Wijnholds et al. 2010). Apart from brute force approaches using the Levenberg-Marquardt solver, two iterative approaches, the differential gains method proposed by Smirnov (2011) and calibration using space alternating generalized expectation maximization (SAGECal) proposed

by Yatawatta et al. (2009) and Kazemi et al. (2011), have become quite popular. Both methods iterate over the directions for which antenna-based gains need to be estimated, assuming that the other directions are already calibrated. In doing so, both methods reduce the estimation problem for each specific direction to the problem of estimating direction independent gains. Currently, the Levenberg-Marquardt solver is used for each of these subproblems, but given the nature of the problem, those estimation steps could be replaced by StEFCal, which is a solver specific to the problem. Introducing StEFCal in such an algorithm can potentially reduce their computational requirements significantly as demonstrated by Salvini & Smirnov (2013) and Salvini & Wijnholds (2014a).

8.4. Summary of main results

In this paper we have analyzed the performance of ADI methods for solving antenna-based complex valued gains with $O(P^2)$ complexity. We have

- done a rigorous analysis of convergence showing that the algorithm converges to the optimal value except in a number of special cases unlikely to occur in any practical scenario;
- reported on its numerical and computational performance; in particular, we highlighted its raw speed, as well as its scalability;
- assessed the statistical performance and shown that it performs very close to the Cramer Rao bound (CRB) in most realistic scenarios;
- commented on variations in the basic ADI method, extension to full polarization cases, and inclusion in more complex calibration scenarios.

Acknowledgements. We would like to thank Oleg Smirnov, Marzia Rivi, Ronald Nijboer, Alle-Jan van der Veen, Jaap Bregman, Johan Hamaker, and Tammo Jan Dijkema for their useful contributions during discussions, and their constructive feedback on earlier versions of this paper. The research leading to this paper has received funding from the European Commission Seventh Framework Programme (FP/2007–2013) under grant agreement No. 283393 (RadioNet3).

References

- Boonstra, A. J. 2005, Ph.D. Thesis, Delft University of Technology, The Netherlands
- Boonstra, A.-J., & van der Veen, A.-J. 2003, IEEE Transactions on Signal Processing, 51, 25
- Bowman, J. D., Cairns, I., Kaplan, D. L., et al. 2013, Publ. Astron. Soc. Aust., 30, 28
- Couillet, R., & Debbah, M. 2011, in IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP) [arXiv:1105.0060]
- de Vos, M., Gunst, A. W., & Nijboer, R. 2009, Proc. IEEE, 97, 1431
- Dewdney, P. E., Hall, P. J., Schilizzi, R. T., & Lazio, T. J. L. W. 2009, Proc. of the IEEE, 97, 1482
- Dewdney, P. E., et al. 2013, SKA1 System Baseline Design, Tech. Rep. SKA-TEL-SKO-DD-001, SKA Project Office, Manchester (UK)
- Dijkema, T. J. 2014, in CalIm 2014 Workshop, Kiama (Australia)
- Hamaker, J. P. 2000, A&AS, 143, 515
- Kay, S. M. 1993, Fundamentals of Statistical Signal Processing – Volume I: Estimation Theory, Prentice Hall Signal Processing Series (Prentice Hall)
- Kazemi, S., & Yatawatta, S. 2013, MNRAS, 435, 597
- Kazemi, S., Yatawatta, S., Zaroubi, S., et al. 2011, MNRAS, 414, 1656
- Kulkarni, S. R. 1989, AJ, 98, 1112
- Lonsdale, C. J., Cappallo, R. J., Morales, M. F., et al. 2009, Proc. IEEE, 97, 1497
- Mitchell, D. A., Greenhill, L. J., Wayth, R. B., et al. 2008, IEEE J. Selected Topics in Signal Processing, 2, 707

- Moon, T. K., & Stirling, W. C. 2000, *Mathematical Methods and Algorithms for Signal Processing* (Upper Saddle River (New Jersey): Prentice Hall)
- Ng, B. C., & See, C. M. S. 1996, *IEEE Trans. Antennas Propag.*, 44, 827
- Noordam, J. E., & Smirnov, O. M. 2010, *A&A*, 524, A1
- Offringa, A. R. 2012, Ph.D. Thesis, University of Groningen, Groningen, The Netherlands
- Offringa, A. R., de Bruyn, A. G., Zaroubi, S., et al. 2013, *A&A*, 549, A11
- Ottersten, B., Stoica, P., & Roy, R. 1998, *Digital Signal Processing, A Review Journal*, 8, 185
- Palais, R. S. 2007, *J. Fixed Point Theory and Applications*, 2, 221
- Prasad, P., & Wijnholds, S. J. 2013, *Phil. Trans. Roy. Soc. A*
- Salvini, S., & Smirnov, O. 2013, in *3rd Workshop on 3rd Generation Calibration (3GC3)*
- Salvini, S., & Wijnholds, S. J. 2014a, in *CALIM 2014 Workshop*, Kiama (Australia)
- Salvini, S., & Wijnholds, S. J. 2014b, in *31st URSI General Assembly and Scientific Symposium*, Beijing (China)
- Salvini, S., Dulwich, F., Mort, B., & Zarb-Adami, K. 2011, in *Aperture Array Verification Programme (AAVP) Workshop*, Dwingeloo (The Netherlands)
- Smirnov, O. M. 2011, *A&A*, 527, A106
- Thompson, A. R., Moran, J. M., & Swenson, G. W. 2004, *Interferometry and Synthesis in Radio Astronomy*, 2nd edn. (Weinheim, Germany: Wiley-VCH Verlag GmbH)
- Van Haarlem, M. P., Wise, M. W., Gunst, A. W., et al. 2013, *A&A*, 556, A2
- Wijnholds, S. J. 2010, Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands
- Wijnholds, S. J., van der Tol, S., Nijboer, R., & van der Veen, A.-J. 2010, *IEEE Signal Process. Mag.*, 27, 30
- Wijnholds, S. J., & van der Veen, A.-J. 2009, *IEEE Transactions on Signal Processing*, 57, 3512
- Yatawatta, S., Zaroubi, S., de Bruyn, G., Koopmans, L., & Noordam, J. 2009, in *13th IEEE DSP Workshop*, Marco Island (Florida), 150
- Zatman, M. 1998, *IEE Proc. Radar, Sonar and Navig.*, 145, 85