Astronomy & Astrophysics

Faceting for direction-dependent spectral deconvolution

C. Tasse[1,2], B. Hugo[2,3], M. Mirmont[8], O. Smirnov[2,3], M. Atemkeng[2], L. Bester[3], M. J. Hardcastle[4], R. Lakhoo[5,6], S. Perkins[3], and T. Shimwell[7]

[1] GEPI, Observatoire de Paris, Université PSL, CNRS, 5 place Jules Janssen, 92190 Meudon, France
  e-mail: cyril.tasse@obspm.fr
[2] Department of Physics & Electronics, Rhodes University, PO Box 94, Grahamstown 6140, South Africa
[3] SKA South Africa, 3rd Floor, The Park, Park Road, Pinelands 7405, South Africa
[4] Centre for Astrophysics Research, School of Physics, Astronomy and Mathematics, University of Hertfordshire, College Lane, Hatfield AL10 9AB, UK
[5] Oxford e-Research Centre, University of Oxford, 7 Keble Road, Oxford OX1 3QG, UK
[6] Wolfson College, University of Oxford, Linton Road, Oxford OX2 6UD, UK
[7] Leiden Observatory, Leiden University, PO Box 9513, 2300 RA Leiden, The Netherlands
[8] Ampyx Power, Lulofsstraat 55-13, The Hague, The Netherlands

ABSTRACT

The new generation of radio interferometers is characterized by high sensitivity, wide fields of view and large fractional bandwidth. To synthesize the deepest images enabled by the high dynamic range of these instruments requires us to take into account the direction-dependent Jones matrices, while estimating the spectral properties of the sky in the imaging and deconvolution algorithms. In this paper we discuss and implement a wideband wide-field spectral deconvolution framework (DDFacet) based on image plane faceting, that takes into account generic direction-dependent effects. Specifically, we present a wide-field co-planar faceting scheme, and discuss the various effects that need to be taken into account to solve for the deconvolution problem (image plane normalization, position-dependent Point Spread Function, etc). We discuss two wideband spectral deconvolution algorithms based on hybrid matching pursuit and sub-space optimisation respectively. A few interesting technical features incorporated in our imager are discussed, including baseline dependent averaging, which has the effect of improving computing efficiency. The version of DDFacet presented here can account for any externally defined Jones matrices and/or beam patterns.

Key words. instrumentation: adaptive optics – instrumentation: interferometers – methods: data analysis – techniques: interferometric

1. Introduction

The new generation of interferometers is characterized by very wide fields of view, large fractional bandwidth, high sensitivity, and high resolution. The cross-correlation between voltages from pairs of antenna (the visibilities) are often affected by severe baseline-time-frequency direction-dependent effects (DDEs) such as the complex beam patterns (pointing errors, dish deformation, antenna coupling within phased arrays), or by the ionosphere and its associated Faraday rotation. The dynamic range needed to achieve the deepest extragalactic surveys involves calibrating for DDEs (see Noordam & Smirnov 2010; Kazemi et al. 2011; Tasse 2014b,a; Smirnov & Tasse 2015, and references therein) and taking them into account in the imaging and deconvolution algorithms.

The present paper discusses the issues associated with estimating spatial and spectral properties of the sky in the presence of DDEs. Those can be taken into account (i) in the Fourier domain using A-Projection (Bhatnagar et al. 2008; Tasse et al. 2013), or (ii) in the image domain using a facet approach (van Weeren et al. 2016; Williams et al. 2016). Algorithms of type (i) have the advantage of giving a smooth image plane correction, while (ii) can give rise to various discontinuity effects. However, (i) is often impractical in the framework of DDE calibration, since a continuous (analytic) image plane

description of the Jones matrices has to be provided, while most calibration schemes estimate Jones matrices in a discrete set of directions. An additional step would be to spatially interpolate the DDE calibration solutions, but this often proves to be difficult due to the very nature of the Jones matrices (2×2 complex valued), and to the unitary ambiguity (see Yatawatta 2013, for a discussion on estimating beam pattern from sets of direction-dependent Jones matrices). Instead, in this paper we address the issue of imaging and deconvolution in the presence of generic DDEs using the faceting approach.

In Sect. 2, we present a general method of imaging non-coplanar data on a multi-facet single tangential plane using modified W-projection kernels (FW kernels). This is a generalization of the original idea presented by Kogan & Greisen (2009). In Sect. 3, we describe the non-trivial effects that arise when forming a dirty image from a set of visibilities and Jones matrices. Specifically, the vast majority of modern interferometers have large fractional bandwidth and, since the station (or antenna[1]) beams scale with frequency, the effective PSF varies across the field of view. We therefore stress here that even if (i) the effect of decorrelation is minimized, and (ii) DDEs are corrected for, all existing imagers will give biased morphological

---

[1] Throughout the paper, we use the terms stations and antennas interchangably to refer to the elements of an interferometer.

results (unresolved sources will appear to be larger toward the edge of the field, since higher-order spatial frequencies are preferentially attenuated by the beam response). The imaging and deconvolution framework presented here (DDFacet[2]) is the first to take all these combined effects into account.

Another important aspect of the work presented in this paper is wide-band spectral deconvolution: the large fractional bandwidth of modern radio interferometers and the need for high dynamic range images means deconvolution algorithms need to estimate the intrinsic continuum sources spectral properties. This is routinely done by the widely used wide band MTMS-CLEAN deconvolution algorithm (Rau & Cornwell 2011). An alternative approach has been implemented by Offringa et al. (2014) in the WSCLEAN package. However, since the sources are affected by frequency dependent DDEs, one needs to couple wide-band and DDE deconvolution algorithms. To our knowledge only Bhatnagar et al. (2013) have implemented such an algorithm, but it is heavily reliant on the assumption that the antennas are dishes, and most directly applicable to the VLA. Also, it does not deal with baseline-time-frequency dependent DDEs (which give rise to a direction-dependent PSF). We aim to build a flexible framework that can solve the wide-band deconvolution problem in the presence of generic DDEs. Specifically, in Sect. 4, we present two wide-band deconvolution algorithms that natively incorporate and compensate for the DDE effects discussed above. The first uses a variation of a matching pursuit algorithm to which we have added an optimisation step. The second uses joint deconvolution on subsets of pixels, and we refer to it as a subspace deconvolution. We have implemented one such approach using a genetic algorithm.

In Sect. 5, we present an implementation of the ideas presented in this paper. Our implementation includes baseline-dependent averaging (Sect. 5.2), and irregular tessellation (Sect. 5.4). A simulation is discussed in detail in Sect. 6. We outline the main results of this paper in Sect. 7.

2. Faceting for wide field imaging

The purpose of faceting is to approximate a wider field of view with many small narrow-field images. Cornwell & Perley (1992) have proposed a polyhedron-like faceting approach, where each narrow-field facet is tangent to the celestial sphere at its own phase center. One of the biggest drawbacks of the noncoplanar polyhedron faceting approach is that the minor cycle deconvolution becomes complicated. Specifically, one needs to re-project each noncoplanar facet into a single plane after synthesis (i.e. in the image-space). Doing the necessary re-projections and inevitable (and expensive) corrections for the areas where the facets overlap can be done through astronomy mosaicing software packages such as the Montage suite (Jacob et al. 2004).

Alternatively, Kogan & Greisen (2009) have described a fundamentally different technique allowing one to build the facets onto a single, common tangential plane. This algorithm consists in applying w-dependent (u, v) coordinate transformation. However, phase errors due to noncoplanarity quickly become a problem, and a W-projection type (Cornwell et al. 2008) correction needs to be applied. As shown in Sect. 2.2 the kernels we are applying are facet-dependent, and differ from the classical W-projection kernels (the gridding and degridding algorithms are described in detail in Sect. 5.2).

2.1. Formalism for the faceting

In order to model the complex direction-dependent effects (DDE–station beams, ionosphere, Faraday rotation, etc), we use the Radio Interferometry Measurement Equation (RIME) formalism, which provides a model of a generic interferometer (for extensive discussions of the validity and limitations of the measurement equation see Hamaker et al. 1996; Smirnov 2011). Each of the physical phenomena that transform or convert the electric field before the correlation is modeled by linear transformations (2×2 matrices). If $s = [l, m, n = \sqrt{1 - l^2 - m^2}]^T$ is a sky direction, and $\mathbf{M}^H$ stands for the Hermitian transpose operator of matrix $\mathbf{M}$, then the 2×2 correlation matrix $\mathbf{V}_{(pq),tv}$ between antennas p and q at time t and frequency v can be written as:

$$\mathbf{V}_{(pq),tv} = \int_s (\mathbf{G}_{pstv} \mathbf{X}_s \, \mathbf{G}^H_{qstv}) k^s_{(pq),tv} \mathrm{d}s + n_{(pq),tv} \quad (1)$$

with $k^s_{(pq),tv} = \exp\left(-2\mathrm{i}\pi \frac{v}{c}(\boldsymbol{b}^T_{pq,t}(\boldsymbol{s} - \boldsymbol{s}_0))\right) \quad (2)$

and $\boldsymbol{b}_{pq,t} = \begin{bmatrix} u_{pq,t} \\ v_{pq,t} \\ w_{pq,t} \end{bmatrix} = \begin{bmatrix} u_{p,t} \\ v_{p,t} \\ w_{p,t} \end{bmatrix} - \begin{bmatrix} u_{q,t} \\ v_{q,t} \\ w_{q,t} \end{bmatrix} \quad (3)$

and $\boldsymbol{s} = \begin{bmatrix} l \\ m \\ n \end{bmatrix}$ and $\boldsymbol{s}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $\quad (4)$

where $\boldsymbol{b}_{pq,t}$ is the baseline vector between antennas p and q. The scalar term $k^s_{(pq)tv}$ describes the effect of the array geometry and correlator on the observed phase shift of a coherent plane wave between antennas p and q. The 2×2 matrix $\mathbf{G}_{pstv}$ is the product of direction-dependent Jones matrices corresponding to antenna p (e.g., beam, ionosphere phase screen and Faraday rotation), and $\mathbf{X}_s$ is referred as the sky term in the direction $\boldsymbol{s}$, and is the true underlying source coherency matrix. The term $n_{(pq),tv}$ is a random variable modeling the system noise. In the following however, we will assume $\mathrm{E}\{n_{(pq),tv}\} = 0$ and implicitly work on the expected values $\mathrm{E}\{.\}$ rather than on the random variables (except in Sect. 3.4 and 3.5, where we describe the structure of the noise in the image domain). Making the (tv) indices implicit, we can transform Eq. (2) to:

$$\mathbf{V}_{pq} = \sum_\varphi \left[\int_{s \in \Omega_\varphi} (\mathbf{G}_p \mathbf{X}_s \, \mathbf{G}^H_q) k^s_{pq,\varphi} \right] \quad (5)$$

$$k^s_{pq,\varphi} = \exp\left(-2\mathrm{i}\pi \frac{v}{c} \boldsymbol{b}^T_{pq}(\boldsymbol{s}_\varphi - \boldsymbol{s}_0)\right) \quad (6)$$
$$\times \exp\left(-2\mathrm{i}\pi \frac{v}{c} \boldsymbol{b}^T_{pq} \delta\boldsymbol{s}_\varphi\right),$$

where $\boldsymbol{s}_\varphi = [l_\varphi, m_\varphi, n_\varphi]^T$ is the direction of the facet phase center and $\delta\boldsymbol{s}_\varphi = \boldsymbol{s} - \boldsymbol{s}_\varphi = [l - l_\varphi, m - m_\varphi, n - n_\varphi] = [\delta l_\varphi, \delta m_\varphi, \delta n_\varphi]$ are the sky coordinates in the reference frame of facet φ.

Applying the term first exponential in Eq. (6) to the visibilities, one still need to apply the position dependent term (second exponential), which can be decomposed as:

$$\exp\left(-2\mathrm{i}\pi \frac{v}{c} \boldsymbol{b}^T_{pq} \delta\boldsymbol{s}_\varphi\right) = \exp\left(-2\mathrm{i}\pi \frac{v}{c}(u\delta l_\varphi + v\delta m_\varphi)\right) \quad (7)$$
$$\times \exp\left(-2\mathrm{i}\pi \frac{v}{c} w\delta n_\varphi\right).$$

---

[2] DDFacet is an imager, implemented in C and Python. It will be made publicly available shortly after the publication of this paper.

The second exponential term is the analog of the w-term corrected for in the W-projection style algorithms. As pointed out by Kogan & Greisen (2009), the first order Taylor expansion approximation of δn_φ can be written as:

$$\delta n_\varphi \approx -\frac{1}{\sqrt{1-l_\varphi^2-m_\varphi^2}}(l_\varphi \delta l_\varphi + m_\varphi \delta m_\varphi) \quad (8)$$

which, conveniently, can be expressed in terms of a coordinate change in uv:

$$\exp\left(-2i\pi \frac{\nu}{c} \boldsymbol{b}_{pq}^T \delta \boldsymbol{s}_\varphi\right) \approx \exp\left(-2i\pi \frac{\nu}{c}(u' \delta l_\varphi + v' \delta m_\varphi)\right) \quad (9)$$

with $u' = u - w\frac{l_\varphi}{\sqrt{1-l_\varphi^2-m_\varphi^2}}$ and $v' = v - w\frac{m_\varphi}{\sqrt{1-l_\varphi^2-m_\varphi^2}}$. The linear approximation of δn_i (Eq. (8)) is plotted in Fig. 1.

2.2. Accurate noncoplanar faceting

As shown in Fig. 1, a more accurate approximation of the $\delta n_\varphi = f\left\{\delta l_\varphi, \delta m_\varphi\right\}$ term may be obtained by a fitting a low-order 2D polynomial

$$\delta n_\varphi \approx (\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \mathbf{P}_{k,\varphi}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}}) \quad (10)$$

where $\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}} = \left[1, \delta l_\varphi, \delta l_\varphi^2, \dots, \delta l_\varphi^k\right]^T$ is a basis function vector for the kth-order 2D polynomial, and $\mathbf{P}_{k,\varphi}$ is the matrix containing the polynomial coefficients. We can then write

$$\boldsymbol{b}_{pq}^T \delta \boldsymbol{s}_\varphi = u \delta l_\varphi + v \delta m_\varphi + w(\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \mathbf{P}_{k,\varphi}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}}) \quad (11)$$

$$\overset{\text{def}\overline{P_k}}{=\!=\!=} (u + w\mathrm{P}_{k,\varphi}^{[10]})\delta l_\varphi + (v + w\mathrm{P}_{k,\varphi}^{[01]})\delta m_\varphi$$
$$+ w(\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \overline{\mathbf{P}_{k,\varphi}}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}}) \quad (12)$$

$$\overset{\text{def}(u',v')}{=\!=\!=} u' \delta l_\varphi + v' \delta m_\varphi + w(\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \overline{\mathbf{P}_{k,\varphi}}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}}) \quad (13)$$

$$\overset{\text{def}\mathbf{C}_{s_\varphi}}{=\!=\!=} (\mathbf{C}_{s_\varphi} \boldsymbol{b}_{pq})^T \delta \boldsymbol{s}_\varphi + w(\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \overline{\mathbf{P}_{k,\varphi}}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}}) \quad (14)$$

where $\overline{\mathbf{P}_{k,\varphi}}$ is equal to $\mathbf{P}_{k,\varphi}$ with the coefficients $\mathrm{P}_k^{[01]}$ and $\mathrm{P}_k^{[10]}$ in cells $[0, 1]$ and $[1, 0]$ zeroed. Based on this polynomial fit, we compute a series of convolutional kernels which we term FW-kernels (for "fitted w kernels"), and apply them by exact analogy with W-projection. As in Kogan & Greisen (2009), we can see here that the first order coefficient of the polynomial fit $\mathbf{P}_{k,\varphi}$ is equivalent to a w-dependent (u, v) coordinate scaling. This has the effect of taking off the main component of the w-related phase gradient, and thereby reducing the FW-kernels' support size (step from Eq. (11) to Eq. (12)) that depends on (i) the w-coordinate, (ii) the facet diameter and (iii) its location. In practice, if the facets are small enough (as it is the case when applying DDEs–see Sect. 3), the support of the FW-kernels is only marginally larger than the spheroidal-only kernel. The FW-kernels are computed per facet, per a given w-plane, as

$$\mathcal{W}_\mathcal{F}\left\{\delta l_\varphi, \delta m_\varphi\right\} = \exp\left(-2i\pi \frac{\nu}{c} w(\delta \boldsymbol{l}_\varphi^{\boldsymbol{k}})^T \overline{\mathbf{P}_{k,\varphi}}(\delta \boldsymbol{m}_\varphi^{\boldsymbol{k}})\right). \quad (15)$$

In practice, a 3rd to 5th-order polynomial is sufficient to accurately represent the w-related phase variation across a given facet.
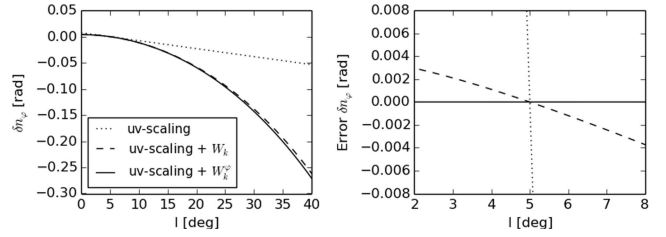
Fig. 1. Comparison between the true δn_i term, and the first-order approximation (*right*), and residuals (*left*). The Kogan & Greisen (2009) approximation breaks down away from the facet center (labeled as *uv*-scaling, dotted line). Applying classical W-projection together with a Kogan & Greisen (2009) style coordinate transformation works better (dashed line), but a blind 3rd–5th-order polynomial works best (solid line).

3. Imaging in the presence of direction dependent effects

In this section, we describe, in terms of linear algebra, how the DDEs are taken into account in the forward (gridding) and backward (gridding) imaging steps.

Specifically, we describe how the dirty images and associated PSFs are constructed from the set of direction-time-frequency dependent Jones matrices. We show in Sect. 3.1 that, in general, in the presence of baseline-time-frequency dependent effects, the linear mapping $\mathcal{T}_\nu$ (Eq. (18) below) between the sky and the dirty image is not a convolution operator. However, in Sect. 3.5, we describe a first-order image correction that modifies this function into a direction-dependent convolution operator, under the condition that the Mueller matrices are approximately baseline-time-frequency independent. As shown in Sect. 3.6, this correction is not sufficient, and the effective PSF retains a directional variation. "Local" PSFs then have to be estimated per facet. In this way, the normalized linear mapping $\widetilde{\mathcal{T}_\nu}$ (Eq. (30) below) can be understood as a local convolution operator.

3.1. Forward and backward mappings

In order to study the properties of the linear function, it is convenient to describe this mapping from image to visibility space and back performed by the algorithm using linear algebra. For a given sample of 4 visibility products, Eq. (1) can be written in terms of a series of linear transformations:

$$\mathbf{v}_{b_\nu} = \mathcal{S}_{b_\nu} \mathcal{F} \mathcal{M}_{b_\nu} \boldsymbol{x}_\nu \overset{\text{def}\mathcal{A}_{b_\nu}}{=\!=\!=} \mathcal{A}_{b_\nu} \boldsymbol{x}_\nu \quad (16)$$

where $\mathbf{v}_{b_\nu}$ is the visibility 4-vector sampled by baseline $\boldsymbol{b} = (p, q, t)$ at frequency ν (for most of this section, we assume a narrow-band scenario). If n_x is the number of pixels in the sky model, the true sky image vector $\boldsymbol{x}_\nu$ has a size of $4n_x$, and for each sky pixel $i = 1 \dots n_x$, the four corresponding correlations[3] (XX, XY, YX, YY)$_i$ or (RR, RL, LR, LL)$_i$ are packed into $\boldsymbol{x}_\nu$ starting from index $4i$. Then, $\mathcal{M}_{b_\nu}$ represents the DDEs, and is a $(4n_x) \times (4n_x)$ block diagonal matrix. For any given image pixel i, the corresponding 4×4 block of $\mathcal{M}_{b_\nu}$ is the Mueller-like[4] matrix associated with the pixel direction $\boldsymbol{s}_i$:

---

[3] In practice the gridder and degridder of DDFacet works on Stokes quantities that can be constructed from measured visibilities.

[4] A Mueller matrix proper operates on Stokes vectors rather than visibility vectors. The Mueller matrix in this case would be given by $\mathbf{S}^{-1}\mathbf{M}_{s_i}^{b_\nu}\mathbf{S}$, where $\mathbf{S}$ is the 4×4 conversion matrix mapping Stokes vectors

Table 1. Overview of the mathematical notations used throughout this paper.

n_x ..	number of pixels in the image domain.
n_v ..	number of visibilities (per single frequency).
n_p ..	number of spatio-spectral parameters to model the sky.
$\mathbf{I}$ ..	The unit matrix.
$\langle w_i f_i \rangle_\Omega$	The mean weighted sum of f_i over a set of indices Ω. Weights are assumed to be normalized throughout.
$\boldsymbol{b}$ ..	Index representing a baseline-time, i.e. a pair of antennas pq at time t.
$\boldsymbol{b}_v$ ..	Index representing a baseline-time-frequency (BTF) point.
Ω_v	The set of all BTFs $\boldsymbol{b}_v$ (i.e. n_v elements) for a given frequency v.
$\mathbf{v}_{\boldsymbol{b}_v}$.	The 4-visibilities vector corresponding to $\boldsymbol{b}_v$.
$\boldsymbol{x}_v$ ..	The true sky vector (at frequency v) of size $(4n_x)$.
$\boldsymbol{y}_v$ ..	The dirty image vector (at frequency v) of size $(4n_x)$.
$\mathcal{F}$ ..	The Fourier transform matrix of size $(n_v \times n_x)$. Each column is the Fourier kernel mapping a given sky coordinate to all visibilities.
$\boldsymbol{S}_{\boldsymbol{b}_v}$.	The matrix of size $(4 \times 4n_v)$ representing the operator that selects 4 out of $4n_v$ visibilities.
$\mathbf{M}_s^{\boldsymbol{b}_v}$.	The 4×4 Mueller-like matrix of a given BTF point $\boldsymbol{b}_v$, for direction s.
$\mathcal{M}_{\boldsymbol{b}_v}$.	A block diagonal matrix of size $(4n_x \times 4n_x)$, representing the DDE effect over the image domain for BTF point $\boldsymbol{b}_v$. Each diagonal block i is a 4×4 $\mathbf{M}_{s_i}^{\boldsymbol{b}_v}$ matrix.
$\mathcal{T}_v$.	The direction-dependent transfer function at frequency v, mapping the sky vector $\boldsymbol{x}_v$ onto the dirty image vector $\boldsymbol{y}_v$
$\widetilde{\mathcal{M}}_v$.	The image plane normalisation used to tranform $\mathcal{T}_v$ to a local convolution matrix.
$\widetilde{(.)}$ ..	An object affected by the image plane normalisation $\widetilde{\mathcal{M}}_v$.
$\mathcal{W}_v$.	Diagonal matrix of size $(4n_v \times 4n_v)$ representing the data weighting for frequency v.
$C_{\boldsymbol{b}_v}$.	The PSF corresponding to a single BTF $\boldsymbol{b}_v$.
Θ ..	A transfer function mapping the spatio-spectral sky model vector to the spectral dirty cube.
Π_v^i .	A $(n_x \times n_p)$ matrix describing the transfer function between the spatio-spectral sky model vector around pixel i to the spectral dirty cube at frequency v.

$\mathbf{M}_{s_i}^{\boldsymbol{b}_v} = \mathbf{G}_q^*(t, v, s_i) \otimes \mathbf{G}_p(t, v, s_i)$. $\mathcal{F}$ is the Fourier transform operator of size $(4n_v) \times (4n_x)$. Each of its (4×4) blocks is a scalar matrix, the scalar being the kernel of the Fourier basis $k_{(pq),tv}^{s_i}$ (see Eq. (2)). The matrix $\boldsymbol{S}_{\boldsymbol{b}_v}$ is the sampling matrix, size $4 \times (4n_v)$, which selects the 4 visibilities corresponding to $\boldsymbol{b}_v$

For the full set of n_v 4-visibilities associated with channel v, which we designate as Ω_v, ($\boldsymbol{b}_v \in \Omega_v$ then means that the $\boldsymbol{b}_v$ index can be taken to represent a visibility index from 1 to n_v), we can

to visibility vectors. See for example Hamaker et al. (1996); Smirnov (2011).

stack n_v instances of Eq. (16) to write the forward (image-to-visibility) mapping as:

$$\mathbf{v}_v = \begin{bmatrix} \vdots \\ \mathcal{A}_{\boldsymbol{b}_v} \\ \vdots \end{bmatrix} \boldsymbol{x}_v \stackrel{\mathrm{def}\,\mathcal{A}_v}{=\!=\!=} \mathcal{A}_v \boldsymbol{x}_v. \tag{17}$$

Note that $\mathcal{A}_v$ represents the "ideal" mapping from images to visibilities, in the sense that a unique DDE is applied at every pixel (ignoring the approximation inherent to pixelizing the sky, we can say that $\mathcal{A}_v$ represents the true instrumental response). Implementing $\mathcal{A}_v$ directly in the forward (modeling) step of an imager would be computationally prohibitive: it is essentially a DFT (Direct Fourier Transform) with pixel-by-pixel application of DDEs. Existing approaches therefore construct some FFT-based (Fast Fourier Transform) approximation to $\mathcal{A}_v$. The convolutional function approach, i.e. AW-projection, approximates $\mathcal{A}_v$ by a single FFT followed by convolutions in the uv-plane during degridding. The facet-based approach of the present work segments the sky $\boldsymbol{x}_v$ into facets, then does an FFT per-facet, while applying a constant DDE $\mathbf{M}_{s_\varphi}^{\boldsymbol{b}_v}$ (where s_φ is the direction of facet φ). The resulting approximate forward operator, $\widehat{\mathcal{A}_v}$, becomes exactly equal to $\mathcal{A}_v$ in the limit of single-pixel facets (see Sect. 3.7 for a further discussion).

3.2. Forming the dirty image

Since $\mathcal{A}_v$ is generally noninvertible, imaging algorithms tend to construct the adjoint operator $\mathcal{A}_v^H$, or some approximation thereof $\widehat{\mathcal{A}_v^H}$, to go back from the visibility domain to the image domain[5]. This amounts to forming the so-called dirty image.

In the framework of facets and DDE calibration, we obtain what is at best an estimate $\widehat{\mathcal{A}_v}$ (due to finite facet sizes, and also calibration errors), and therefore the adjoint operator being applied is also an approximation. The same applies to convolutional gridding approaches. For the purposes of this section, however, let us assume that the approximation is perfect. We then have the following for the dirty image vector $\boldsymbol{y}_v$:

$$\boldsymbol{y}_v = \mathcal{A}_v^H \mathcal{W}_v \mathbf{v}_v \tag{18}$$

$$= \mathcal{A}_v^H \mathcal{W}_v \mathcal{A}_v \, \boldsymbol{x}_v \stackrel{\mathrm{def}\,\mathcal{T}}{=\!=\!=} \mathcal{T} \, \boldsymbol{x}_v \tag{19}$$

$$= \left\langle w_{\boldsymbol{b}_v} \mathcal{M}_{\boldsymbol{b}_v}^H \mathcal{F}^H \boldsymbol{S}_{\boldsymbol{b}_v}^H \boldsymbol{S}_{\boldsymbol{b}_v} \mathcal{F} \mathcal{M}_{\boldsymbol{b}_v} \right\rangle_{\Omega_v} \boldsymbol{x}_v \tag{20}$$

$$\stackrel{\mathrm{def}\,C_{\mathrm{b}_v}}{=\!=\!=} \left\langle w_{\boldsymbol{b}_v} \mathcal{M}_{\boldsymbol{b}_v}^H C_{\boldsymbol{b}_v} \mathcal{M}_{\boldsymbol{b}_v} \right\rangle_{\Omega_v} \boldsymbol{x}_v \tag{21}$$

where $\mathcal{W}_v$ is a diagonal matrix containing the set of weights $w_{\boldsymbol{b}_v}$ at frequency v. Note that the weighted sum comes about due to the block-column of Eq. (17) being left-multiplied by its conjugate, a block-row.

For each $\boldsymbol{b}_v$, the matrix $C_{\boldsymbol{b}_v} = \mathcal{F}^H \boldsymbol{S}_{\boldsymbol{b}_v}^H \boldsymbol{S}_{\boldsymbol{b}_v} \mathcal{F}$ is a convolution, as a direct consequence of the Fourier convolution theorem. This matrix represents the convolution of the sky by the PSF corresponding to a single uv-point (i.e. a single fringe). In the absence of DDEs ($\mathcal{M}_{\boldsymbol{b}_v} \equiv \mathbf{I}$), the linear mapping $\mathcal{T}_v$ can be written as a a weighted sum of such matrices, and is therefore also a convolution:

$$\boldsymbol{y}_v = \left\langle w_{\boldsymbol{b}_v} C_{\boldsymbol{b}_v} \right\rangle_{\Omega_v} \boldsymbol{x}_v \stackrel{\mathrm{def}\,C}{=\!=\!=} C \boldsymbol{x}_v. \tag{22}$$

[5] This can be motivated as follows: for any given matrix A, the null spaces $\ker\{A\}$ and $\ker\{A^H A\}$ are identical. Therefore, applying the adjoint operator $\mathcal{A}^H$ to go back to images from visibilities preserves all information not destroyed by $\mathcal{A}$.

This is just the familiar result that in the absence of DDEs, the dirty image is a convolution of the apparent sky by a PSF.

Below, we show that in the presence of DDEs (even corrected-for DDEs), this relationship generally ceases to be a true convolution. We will also show that, under certain conditions, the mapping can be modified (at least approximately) into a local convolution, i.e. one where the PSF varies only slowly with direction. This distinction is important: most minor-loop deconvolution algorithms such as CLEAN either assume the problem is a true (global) convolution, or can be trivially modified (at least in the faceted approach) to deal with a local convolution problem, i.e. a position-dependent (per facet) PSF.

3.3. Toeplitz matrices and convolution

Any matrix $\mathbf{C}$ representing a one-dimensional convolution is Toeplitz, and vice versa. A Toeplitz matrix is a matrix in which each descending diagonal is constant, i.e. $C_{ij} = C_{i+1,j+1} \equiv c_{i-j}$. We now show that a similar property, which we'll call Toeplitzian, can be defined for convolution of 2D, 4-polarization images. We can then discuss how DDEs break the convolution relationship by making the matrix representing the tranfer function less Toeplitzian.

First, consider matrices that represent 2D convolution of scalar (unpolarized) images. The pixel ordering, i.e. the order in which we stack the pixels of a 2D image into the image vectors $\mathbf{x}$ and $\mathbf{y}$, induces a mapping from vector index i to pixel coordinates (l_i, m_i). Given a fixed pixel ordering, consider a matrix $\mathbf{C}$ whose its elements are constant with respect to a translation of pixel coordinates, i.e. $C_{ij} = C_{i'j'}$ for all ij and $i'j'$ such that $l_i - l_j = l_{i'} - l_{j'}$ and $m_i - m_j = m_{i'} - m_{j'}$. There then exists a function of pixel coodinates $c(l, m)$ such that for all ij

$$C_{ij} = c(l_i - l_j, m_i - m_j) = c(\Delta l_{ij}, \Delta m_{ij}), \tag{23}$$

and it is then easy to see that applying the $\mathbf{C}$ matrix to the image vector $\mathbf{x}$ corresponds to a 2D convolution of the corresponding image by c, and vice versa. For an $n \times n$ image, assuming the conventional pixel ordering of stacked columns (or rows), the matrix $\mathbf{C}$ is composed of $n \times n$ blocks, each block being an $n \times n$ Toeplitz matrix. Each constant descending diagonal in each such block represents a constant pixel separation $\Delta l, \Delta m$. In other words, C_{ij} is constant for any pair of pixels having the same pixel separation $\Delta l, \Delta m$.

To generalize this to 4-polarization images, we simply replace C_{ij} in Eq. (23) by a 4×4 scalar matrix. Our general Toeplitzian matrix is then composed of $n \times n$ blocks, each block being a $4n \times 4n$ Toeplitz matrix composed of 4×4 scalar matrices. Each column of such matrix represents the convolution kernel (or PSF), shifted to the position of the appropriate image pixel.

The linear function defined by the PSF $\mathbf{C}_{b_\nu}$ or $\mathbf{C}$ is Toeplitzian, with 1 on the main diagonal (corresponding to the peak of the PSF). We focus on two regimes in which a matrix becomes non-Toeplitzian. The first one is simple, when C_{ij} in Eq. (23) is constant to within a 4×4 per-column scaling factor M_j. This correponds to an attenuation of the image by $\mathbf{M}_\nu$, followed by a convolution:

$$\mathbf{y}_\nu = \mathbf{C}\mathbf{M}_\nu\mathbf{x}_\nu \stackrel{\mathrm{def}\,\widetilde{\mathbf{x}}_\nu}{=\!=\!=} \mathbf{C}\widetilde{\mathbf{x}}_\nu. \tag{24}$$

This regime arises when trivial (i.e. non time-baseline dependent) DDEs are present and not accounted for when forming the dirty image. $\mathbf{M}_\nu$ can be factored out of the sum in Eq. (21) and

absorbed into the apparent sky $\widetilde{\mathbf{x}}_\nu$. In this case we can still talk of the PSF shape being constant across the image.

The more complex regime arises when the mapping is non-Toeplitzian in the sense that the shape of the PSF changes across the image. This naturally arises when nontrival DDEs are present and not accounted for, and the dirty image is the weighted sum of the sky affected by baseline-dependent DDE

$$\mathbf{y}_\nu = \langle w_{b_\nu} \mathbf{C}_{b_\nu} \mathbf{M}_{b_\nu} \rangle_{\Omega_\nu} \mathbf{x}_\nu. \tag{25}$$

More subtly, even if DDEs are perfectly known and accounted for in $\mathcal{A}^H$, the resulting function is, generally, not a convolution, in the sense that the shape of the PSF becomes direction-depedent. This is obvious in the case of nonunitary $\mathbf{M}_{b_\nu}$ (since its amplitude essentially appears twice in Eq. (21), and the resulting dirty image requires renormalization – we will return to this again below). Less obvious is that this holds, generally, even for unitary $\mathbf{M}_{b_\nu}$. Consider the simple case of a scalar, unitary DDE (i.e. a phase term affecting both polarizations equally). This corresponds to a diagonal $\mathbf{M}_{b_\nu}$ with $M_i = \mathrm{e}^{\iota\psi_i}$ on the diagonal. If the matrix elements of $\mathbf{C}_{b_\nu}$ are given by C_{ij}, then each element of $\mathbf{M}_{b_\nu}^H \mathbf{C}_{b_\nu} \mathbf{M}_{b_\nu}$, i.e. the response at dirty image pixel j to a source at pixel i (i.e. the PSF sidelobe response), is given by

$$C'_{ij} = M_i^* C_{ij} M_j = C_{ij}\mathrm{e}^{\iota(\psi_j - \psi_i)}. \tag{26}$$

It is easy to see that the (Toeplitzian) condition of Eq. (23) is only satisfied if $\psi_j - \psi_i$ is constant for any pair of pixels having the same pixel separation $\Delta l, \Delta m$. This condition is only true for a linear phase slope over the image.

We have shown that here all nontrivial DDEs, including unitary ones, with the exception of linear phase slopes, generally result in a direction-dependent PSF even when perfectly known and accounted for via Eq. (18). Note that this equation (or some approximation thereof) is applied by all existing imagers. If we consider the w-term as a DDE (see, e.g., Smirnov 2011), we can see that W-projection and W-stacking also represent approximations of Eq. (18), and therefore still yield a direction-dependent PSF.

3.4. Loss of local convolution property and nonuniform noise reponse

Equations (23) and (26) give us a framework in which we can reason about the degree of direction-dependence in the PSF. The pixel separation $\Delta l, \Delta m$ corresponds to the PSF sidelobe at C_{ij}. Thus, the direction-dependence of a particular PSF sidelobe $\Delta l, \Delta m$ is determined by the variation of the Mueller matrix across the image on a length scale of $\Delta l, \Delta m$. For direction-dependent effects that are locally approximately linear (i.e. close to the form of Eq. (26)), the problem is locally a convolution. As long as this is true, and assuming $\mathbf{M}_{b_\nu}$ is known, one could in principle incorporate knowledge of a local direction-dependent PSF into the minor cycle deconvolution algorithm, using the linear function defined above to form up the dirty images. In the context of facet imaging this seems straightfoward, as we can simply compute a PSF per facet (see below). However, if the Mueller matrices are nonunitary, $\mathcal{T}_\nu$ has two very undesirable properties.

Firstly, as is clear from Eq. (26), the PSF sidelobe response C'_{ij} is coupled to $\mathbf{M}_{b_\nu}$ at both positions i and j. Ideally, we would like to decouple the PSF sidelobe response from the DDE at position i.

Secondly, consider the thermal noise response in the dirty image given by $\mathcal{T}_\nu$. Thermal noise can be assumed to be independent and identically distributed Gaussian in the visibilities $\boldsymbol{v}_{\boldsymbol{b}_\nu}$. If $\boldsymbol{a}$ is a vector of random variables and $\boldsymbol{b} = \boldsymbol{B}\boldsymbol{a}$, then the covariance matrices of the two vectors are related by $\mathrm{Cov}\{\boldsymbol{b}\} = \boldsymbol{B}\mathrm{Cov}\{\boldsymbol{a}\}\boldsymbol{B}^H$. Applying this to Eq. (18), and using $\sigma^2_{\boldsymbol{b}_\nu}$ for the variance of the real and imaginary parts of each visibility, we get

$$\mathrm{Cov}\{\boldsymbol{y}_\nu\} = \left\langle w^2_{\boldsymbol{b}_\nu} \sigma^2_{\boldsymbol{b}_\nu} \mathcal{M}^H_{\boldsymbol{b}_\nu} \boldsymbol{C}_{\boldsymbol{b}_\nu} \mathcal{M}_{\boldsymbol{b}_\nu} \right\rangle_{\Omega_\nu}, \tag{27}$$

spatially uniform. In particular, the variance (of the four polarization products) at each pixel i is given by $\left\langle w^2_{\boldsymbol{b}_\nu} \sigma^2_{\boldsymbol{b}_\nu} \mathcal{M}^H_{\boldsymbol{b}_\nu i} \mathcal{M}_{\boldsymbol{b}_\nu i} \right\rangle_{\Omega_\nu}$.

3.5. Image-plane corrections

We show in this section that when $\mathcal{M}_{\boldsymbol{b}_\nu}$ is approximately baseline-time independent (at a given frequency), we can find a dirty image correction that brings $\mathcal{T}$ back to a direction-dependent convolution operator. This is a reasonable assumption in general if the fractional bandwith of the data chunk is small enough[6], and in this case we can write

$$\mathcal{T} \overset{\mathrm{def}\widetilde{\mathcal{M}_\nu}}{=\joinrel=\joinrel=} \widetilde{\mathcal{M}_\nu}^H \langle w_b \boldsymbol{C}_{\boldsymbol{b}_\nu} \rangle_{\Omega_\nu} \widetilde{\mathcal{M}_\nu} \tag{28}$$

$$\overset{\mathrm{def}\boldsymbol{C}_S}{=\joinrel=\joinrel=} \widetilde{\mathcal{M}_\nu}^H \boldsymbol{C}_S \widetilde{\mathcal{M}_\nu}. \tag{29}$$

We can see from Eq. (28) that we can construct a modified normalized image

$$\widetilde{\boldsymbol{y}_\nu} = \left(\widetilde{\mathcal{M}_\nu}^H\right)^{-1} \mathcal{T}_\nu \boldsymbol{x}_\nu \overset{\mathrm{def}\widetilde{\mathcal{T}_\nu}}{=\joinrel=\joinrel=} \widetilde{\mathcal{T}_\nu} \boldsymbol{x}_\nu \tag{30}$$

mapping. The columns i of $\widetilde{\mathcal{T}_\nu}$ are then the PSF of a source centered on pixel i, and they only differ one to the other by a matrix product. In other words, the PSF centered on pixel i is the same as the PSF centered on pixel j to within a constant. Strictly speaking $\widetilde{\mathcal{T}_\nu}$ is not a convolution matrix, but we will refer to it as a direction-dependent convolution matrix. An alternative way to look at this is to write $\widetilde{\boldsymbol{y}_\nu} = \boldsymbol{C}_S \widetilde{\boldsymbol{x}_\nu}$ where $\widetilde{\boldsymbol{x}_\nu} = \widetilde{\mathcal{M}_\nu} \boldsymbol{x}_\nu$ is the apparent beam-attenuated sky.

In order to obtain $\widetilde{\mathcal{M}_\nu}$, we can see from Eq. (28) that although $\mathcal{T}_\nu$ is not block diagonal, each ith (4×4) block on its diagonal is

$$\mathcal{T}_\nu[i,i] = \left\langle w_b \mathcal{M}^H_{b,i} \mathcal{M}_{b,i} \right\rangle_{\Omega_\nu}. \tag{31}$$

Assuming approximate baseline-time independence at ν of the direction-dependent local convolution function (i.e. $\mathcal{T}_\nu[i,i] \approx \widetilde{\mathcal{M}_i}^H \widetilde{\mathcal{M}_i}$), we get

$$\widetilde{\mathcal{M}_\nu} \overset{\frown}{=} \sqrt{\left\langle w_b \mathcal{M}_{\boldsymbol{b}_\nu}^H \mathcal{M}_{\boldsymbol{b}_\nu} \right\rangle_{\Omega_\nu}}. \tag{32}$$

If the assumption in Eq. (28) holds (definition of $\widetilde{\mathcal{M}_\nu}$), then the image plane correction exists, and it is given by Eq. (32). Furthermore we take into account the deviation from this approximation by using the local PSF (Sect. 3.6) in our deconvolution

---

[6] In DDFacet the data is internally imaged into frequency-chunked spectral cubes, and the correction described here (Eq. (30)) is done on a user-defined but ideally small bandwidth.

algorithms (Sect. 4). Applying the $\widetilde{\mathcal{M}_\nu}$ correction in Eq. (27), the normalized image-plane pixel covariance $\mathrm{Cov}\{\widetilde{\boldsymbol{y}_\nu}\}$ becomes

$$\mathrm{Cov}\{\widetilde{\boldsymbol{y}_\nu}\} = \left\langle w^2_{\boldsymbol{b}_\nu} \sigma^2_{\boldsymbol{b}_\nu} \boldsymbol{C}_{\boldsymbol{b}_\nu} \right\rangle_{\Omega_\nu}, \tag{33}$$

and $\mathrm{Cov}\{\widetilde{\boldsymbol{y}_\nu}\}$ is spatially uniform.

In practice, the Mueller blocks in $\mathcal{M}_{\boldsymbol{b}_\nu}$ are assumed to be diagonally dominant and are reduced to scalar matrices when computing $\widetilde{\mathcal{M}_\nu}$.

3.6. Direction-dependent PSFs

As shown above, the combined effects of (i) baseline-time-frequency dependence of the DDEs, and (ii) decorrelation cause the linear mappings $\mathcal{T}_\nu$ and $\widetilde{\mathcal{T}_\nu}$ not to be exact convolution matrices. Specifically, the large fractional bandwidth makes the beam pattern vary significantly toward the edge of the field, and the effective PSF is also direction-dependent. All modern imagers are indeed affected by problem (i) in the minor cycle, and problem (ii) in both the major and minor cycles, and so will produce morphologically biased results away from the pointing center. In this section we describe how DDFacet takes into account and compensates for these effects.

3.6.1. Effect of decorrelation

It follows from Eq. (1) that any source in the sky corresponds to a complex vector rotating in the *uv*-domain and any visibility measurement is an averaged value over that domain. This fact causes the amplitude of the averaged vector to decrease (in the extreme case in which the phase of the complex vector ranges over $[-\pi, \pi]$ in the domain of averaging, the average vector amplitude can be zero). This effect is known as decorrelation and is described in much detail by Bridle & Schwab (1999), Thompson et al. (2001), Smirnov (2011), Atemkeng et al. (in prep., and references therein). One can see from Eq. (1) that the magnitude of decorrelation depends on (i) the baseline coordinates and (ii) the distance of the source to the phase center, causing the effective PSF to be direction-dependent. This effect is a direct image-domain consequence of baseline and direction-dependent decorrelation, and is known in the literature as smearing.
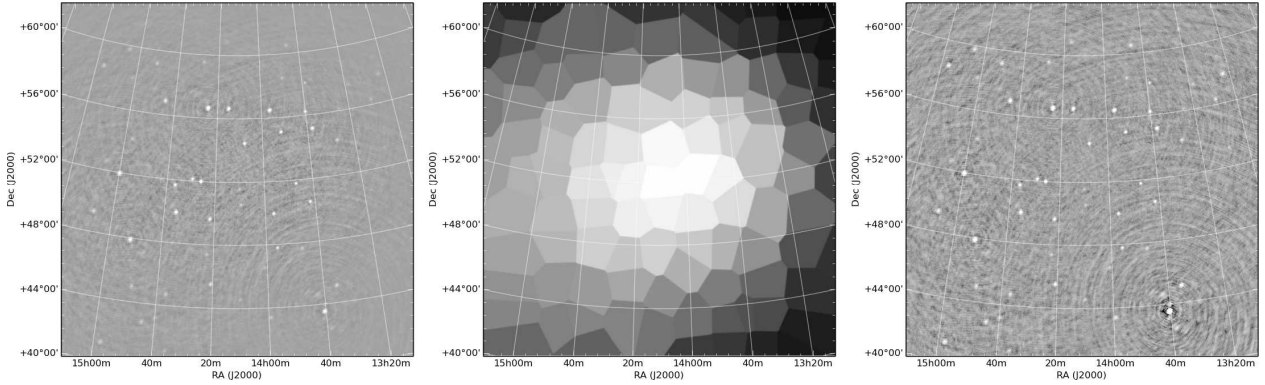
The effective mapping is therefore direction-dependent, and no imaging and deconvolution can take this effect into account. This has the direct effect of incorrectly estimating the source's morphology, and the error gets worse as the source gets further away from the phase center. Since the longest baselines are most affected, decorrelation is minimized by accepting a small decorrelation (e.g. a few per cent decrease in the ratio to peak to integrated flux density) at the edge of the field.

A major strength of a facet-based imaging and deconvolution framework is that we can take decorrelation into account in quite an easy way by computing a PSF per facet. While computing the PSF, each unit visibility is multiplied by the factor $\gamma_{pq,t}$, defined as

$$\gamma_{pq,t\nu} = \frac{\sin\left(\phi_{pq,t}\right)}{\phi_{pq,t}} \frac{\sin\left(\psi_{pq,t}\right)}{\psi_{pq,t}} \tag{34}$$

with $\phi_{pq,t\nu} = \pi \dfrac{\Delta_\nu}{c} \boldsymbol{b}^T_{pq,t} (\boldsymbol{s} - \boldsymbol{s}_0)$ \hfill (35)

and $\psi_{pq,t\nu} = \pi \dfrac{\nu}{c} \dfrac{\mathrm{d}\boldsymbol{b}_{pq,t}}{\mathrm{dt}}^T (\boldsymbol{s} - \boldsymbol{s}_0) \Delta_t,$ \hfill (36)

Fig. 2. The nonunitary Mueller matrices in $\mathcal{A}$ cause the mapping in Eq. (19) to not be a convolution operator. The *left panels* shows the dirty image $\boldsymbol{y}_\nu$ obtained after applying $\mathcal{A}^H$ to the visibilities. The image plane correction $\widetilde{\mathcal{M}_\nu}$ is displayed in the *central panel* and the corrected image $\widetilde{\boldsymbol{y}_\nu}$ is shown in the *right panel*. As explained in Sect. 3.5 the modified mapping is approximately a direction-dependent local convolution function.
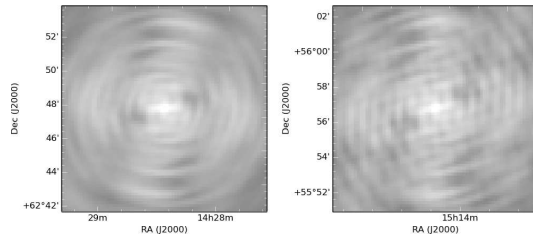
Fig. 3. PSF estimated at various locations of the image plane even after the transformation described in Eq. (30) is applied. The net local convolution function significantly varies, and this effect is taken into account by computing a PSF per facet.

where Δ_t and Δ_ν are the time and frequency size for the domain over which the given visibility has been averaged, and $\frac{d\boldsymbol{b}_{pq,t}}{dt}$ is the speed of the baseline in the *uv* domain. Conversely, in the forward step of major cycle, $\gamma_{pq,tv}$ can be applied to the model visibilities on a per-facet basis. This allows decorrelation to be properly accounted for both in the minor and major cycles.

3.6.2. Per-facet PSFs

In the facet approach, it is staightforward to compute a per-facet PSF that takes all of the above effects into account during deconvolution. We compute a PSF per facet ϕ for a point source following

$$\widetilde{\boldsymbol{y}_{\nu 1}} = \widetilde{\mathcal{T}_\nu^\phi} \boldsymbol{0}_1 \tag{37}$$

where $\widetilde{\mathcal{T}_\nu^\phi}$ is the local convolution function function in facet ϕ, and $\boldsymbol{0}_1$ is a vector containing zeros everywhere except the central pixel, which is set to the value $\{I, Q, U, V\} = \{1, 0, 0, 0\}$ Jy. Figure 3 shows the PSF evaluated for a source in two different facets.

Note that in the full-polarization case, i.e. given DDEs with a nontrivial polarization response (nondiagonal, or at least non-scalar Mueller matrices), it is in principle incorrect to speak of one PSF. All four Stokes components are, in general, convolved with different PSFs, and there are also "leakage PSFs" that transfer power between components. A fully accurate description of the local convolution relationship therefore requires that 16 independent PSFs be computed, with all the consequent expense (i.e. 16 separate gridding operations for the PSF computation). In practice, we limit ourselves to computing the Stokes I PSF, and,

during the minor cycle of deconvolutin, assume that the other Stokes component PSFs are the same, and treat leakages as negligible, trusting in the major cycle to correct the effect. The impact of this approximation on polarization deconvolution is a topic for future study.

3.7. In-facet errors

As explained in Sect. 1, the facet-based imaging and deconvolution framework presented here has the disadvantage of taking DDEs into account in a discontinuous manner in the image domain. Indeed, within a direction-dependent facet, DDEs are assumed to be constant while they continuously vary. This is typically the case for beam effects that vary very quickly, especially around the half power point. We show in this section that this effect can be partially accounted for by applying a spatially smooth term to the image $\widetilde{\boldsymbol{y}_\nu}$.

In this section we estimate the flux density error across a given facet φ that arises due to the fact that the Jones matrix has been assumed to be spatially constant. Following Eq. (16), the residual visibility on a given baseline $\boldsymbol{b}$ can be written as

$$\mathbf{r}_{\boldsymbol{b}_\nu} = \mathcal{S}_{\boldsymbol{b}_\nu} \mathcal{F}(\mathcal{M}_{\boldsymbol{b}_\nu} \boldsymbol{x}_\nu - \mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi \widehat{\boldsymbol{x}_\varphi}) \tag{38}$$

where $\mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi$ is a $(4n_x) \times (4n_x)$ block diagonal matrix which represents the direction-independent Jones matrix that has been assumed for that facet, and $\widehat{\boldsymbol{x}_\varphi}$ is the sky that has been estimated. We assume the deconvolution algorithm is subject to an $\mathcal{L}_2$-norm constraint, and

$$\chi^2 = \sum_{\boldsymbol{b}} (\mathcal{M}_{\boldsymbol{b}_\nu} \boldsymbol{x}_\nu - \mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi \widehat{\boldsymbol{x}_\varphi})^H C_{\boldsymbol{b}_\nu} (\mathcal{M}_{\boldsymbol{b}_\nu} \boldsymbol{x}_\nu - \mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi \widehat{\boldsymbol{x}_\varphi}) \tag{39}$$

is minimized, giving

$$\frac{\partial \chi^2}{\partial \widehat{\boldsymbol{x}_\varphi}} = \boldsymbol{0} = \sum_{\boldsymbol{b}} 2\mathcal{M}_{\boldsymbol{b}_\nu}{}^{\varphi H} C_{\boldsymbol{b}_\nu} \mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi \widehat{\boldsymbol{x}_\varphi} - 2\mathcal{M}_{\boldsymbol{b}_\nu}{}^{\varphi H} C_{\boldsymbol{b}_\nu} \mathcal{M}_{\boldsymbol{b}_\nu} \boldsymbol{x}_\nu \tag{40}$$

and therefore

$$\boldsymbol{x}_\nu = \left(\sum_{\boldsymbol{b}} \mathcal{M}_{\boldsymbol{b}_\nu}{}^{\varphi H} C_{\boldsymbol{b}_\nu} \mathcal{M}_{\boldsymbol{b}_\nu} \right)^{-1} \left(\sum_{\boldsymbol{b}} \mathcal{M}_{\boldsymbol{b}_\nu}{}^{\varphi H} C_{\boldsymbol{b}_\nu} \mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi \right) \widehat{\boldsymbol{x}_\varphi} \tag{41}$$

As in Sect. 3.5, assuming the $\mathcal{M}_{\boldsymbol{b}_\nu}$ and $\mathcal{M}_{\boldsymbol{b}_\nu}{}^\varphi$ are baseline-time independent at ν we get

$$\boldsymbol{x}_\nu = \mathcal{M}_\nu{}^{-1} \mathcal{M}_\nu{}^\varphi \widehat{\boldsymbol{x}_\varphi}. \tag{42}$$

Data: $\widetilde{y}$, t, α, $\boldsymbol{Q}$, $\mathbf{v}$
Result: The estimated model $\widehat{\boldsymbol{\pi}}$ of the true sky $\boldsymbol{\pi}$ in the basis function $\mathcal{P}$.

1   initialization: $\widehat{\boldsymbol{\pi}} = \mathbf{0}$;
   /* Start n_{Cycle} major cycles   */
2   **for** i_{Cycle} *in range(n_{Cycle})* **do**
     /* Start minor cycles   */
3   **while** max $\{\widetilde{\delta y}\} > t$ **do**
4   Find location of brightest pixel:
       $i = \mathrm{argmax}\left(\widetilde{\delta y}\right)$;
5   Find locally best sky model centered on pixel i:
       $\widehat{\boldsymbol{\pi}}_i = \underset{\boldsymbol{\pi}_i}{\mathrm{argmin}}\left(C_0\{\widetilde{\Theta_i}d\boldsymbol{\pi}_i | \widetilde{\delta y}, \boldsymbol{Q}\}\right)$ [s.t. $C_1\{\boldsymbol{\pi}_i\}$];
6   Update sky model:
       $\widehat{\boldsymbol{\pi}} \leftarrow \widehat{\boldsymbol{\pi}} + \alpha\, \widehat{\boldsymbol{\pi}}_i$;
7   Update residual image:
       $\widetilde{\delta y} \leftarrow \widetilde{\delta y} - \alpha\widetilde{\Theta_i}\widehat{\boldsymbol{\pi}}_i$;
8   **end**
9   Update residual image:
     $\widetilde{\delta y} = \widetilde{\boldsymbol{\mathcal{M}}_\nu}^{-H}\mathcal{A}^H\boldsymbol{W}(\mathbf{v} - \mathcal{A}\Pi\widehat{\boldsymbol{\pi}})$;
10   **end**
11   Compute model image $\widehat{\boldsymbol{x}_\nu} = \boldsymbol{\Pi}_\nu\widehat{\boldsymbol{\pi}}$;
12   Compute restored image $\widehat{\boldsymbol{x}} \leftarrow \widetilde{\delta y} + C\widehat{\boldsymbol{x}_\nu}$;

Algorithm 1: HMP deconvolution algorithm. Here t is a user defined flux density threshold, α is a minor cycle loop gain. Other symbols are defined in Table 1 and/or in the main text.

One can see that when no DDEs are being applied during deconvolution ($\boldsymbol{\mathcal{M}}_\nu^\varphi = \mathbf{I}$, as is traditionally done in radio astronomy), one can correct the fluxes by applying a smooth beam correction in the image domain.

4. Wideband deconvolution

In this section, we describe how we solve for the sky in the local deconvolution problem as well as the global inverse problem[7] $\widetilde{y} = \widetilde{\mathcal{T}_\nu}\boldsymbol{x}$. We present two multiscale wideband deconvolution algorithms that iteratively estimate the underlying true sky. In contrast to the calibration problem, the deconvolution problem is linear, but is strongly ill-conditioned. A wide variety of algorithms have been developed to tackle the conditioning issue.

The first and largest family of deconvolution algorithms in radio interferometry is based on compressive sampling theory (or compressive sensing), and assumes the sky can be fully described by a small number of coefficients in a given dictionary[8] of functions (a sparse representation). The dictionary of functions can be, but is not necessarily, a basis function from deltas to shapelets. In practice and for a given dataset, a specific convex solver is used to estimate the coeffiscient associated to the functions of the dictionary. The cost function is often an $\mathcal{L}_1$-norm subject to an $\mathcal{L}_2$ constraint. The widely used CLEAN algorithm is one of those[9] solvers, but we can also mention

---

[7] In an abuse of language we can also call inverting $\widetilde{\mathcal{T}}_\nu$ a deconvolution problem, although strictly speaking it is not a convolution operator.
[8] In contrast to a basis, the decomposition of a vector in these dictionaries is not necessarily unique.
[9] Although CLEAN was written before compressive sampling theory had been described.

MORESANE (Dabbech et al. 2015), or SASIR (Garsden et al. 2015). Each one of these methods uses a specific solver to estimate the coefficients associated with a given dictionary. The second family of algorithms deals with ill-conditioning using Bayesian inference.

Only a few existing algorithms are able to accurately estimate flux densities as well as intrinsic spectral properties (while taking Jones matrices into account). The most efficient and widely used of these is the MTMS-CLEAN algorithm (for multi-term multi-scale, see Rau & Cornwell 2011, and references therein). Bhatnagar et al. (2013) have extended this algorithm in order to take time-frequency dependent DDEs into account. The drawbacks of this algorithm combination are that (i) since each Taylor coefficient image stacks information from potentially large fractional bandwidth, $\widetilde{\mathcal{T}}_\nu$ (Eq. (30), Sect. 3) will tend not to be a convolution operator, (ii) it decomposes the signal in terms of Taylor basis functions, and the signal needs to be gridded n_t-times if n_t is the number of Taylor terms, and (iii) baseline-dependent averaging cannot be used with A-Projection (see Sect. 5.2).

Instead, we produce a ($n_{ch} \times n_{pix}$) spectral cube, the dirty images of size (n_{pix}) being formed into the corresponding n_{ch} frequency chunks. The spectral cube then contains information about the sky's spectral properties. We present in this section two wideband deconvolution algorithms that estimate flux densities as well as the intrinsic spectral properties (taking into account Jones matrices such as primary beam direction-time-frequency behavior). The first uses a variation of the matching pursuit CLEAN algorithm, while the second uses a genetic algorithm.

4.1. HMP deconvolution

In this section we present the HMP deconvolution algorithm (Hybrid Matching Pursuit). The idea is quite simple and consists of decomposing the signal around the brightest pixel i in the spectral cube $\widetilde{y}$ into a sum of components with different spatial and spectral properties. The basis function is similar to MTMS-CLEAN (Rau & Cornwell 2011), but the idea differs in that (i) we grid the data only once (we do not create dirty images at different resolutions and for different Taylor terms), (ii) the optimisation step is done on a set of pixels (and not only on the brightest pixel), and (iii) at each iteration all coefficients are estimated in the chosen basis function (as opposed to the maximum coefficient only). This last point is illustrated by the example of a faint extended signal containing a brighter point source. While Cornwell (2008) have to introduce an ad hoc "small-scale bias" to reconstruct the compact emission, we aim at finding nonzero coefficients for the point source and the extended emission, at each iteration (the same applies to the spectral axis). The following algorithm is implemented in DDFacet, natively taking direction-dependent residual images and associated PSFs into account (see Sect. 3.5 for a discussion of the normalization).

We first choose a set $\mathcal{P}$ of functions into which we want to decompose the spectral cube. For example, it can be made of Gaussians with various sizes and spectral indices. The sky image $\boldsymbol{x}_\nu^i$ of models centered on pixel i at a frequency ν is then written as

$$\boldsymbol{x}_\nu = \sum_i \Pi_\nu^i\boldsymbol{\pi}_i \overset{\mathrm{def}\Pi_\nu}{=\!=\!=} \boldsymbol{\Pi}_\nu\boldsymbol{\pi} \tag{43}$$

and

$$\boldsymbol{x} = \begin{bmatrix} \vdots \\ \boldsymbol{x}_\nu \\ \vdots \end{bmatrix} \overset{\mathrm{def}\Pi}{=\!=\!=} \boldsymbol{\Pi}\boldsymbol{\pi} \tag{44}$$

where Π_ν^i is the $(n_x \times n_p)$ matrix containing the spectro-spatial dictionary estimated at frequency ν, and $\boldsymbol{\pi}_i$ is the spectro-spatial sky model of pixel i, containing the n_p parameter values of the spectro-spatial dictionary. We can then write the contribution $\widetilde{\boldsymbol{y}}_i$ of pixel i to the spectral cube as

$$\widetilde{\boldsymbol{y}}_i = \begin{bmatrix} \vdots \\ \widetilde{\boldsymbol{y}}_{i,\nu} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \widetilde{\mathcal{T}}_\nu \Pi_\nu^i \\ \vdots \end{bmatrix} \boldsymbol{\pi}_i \tag{45}$$

$$\stackrel{\text{def}\widetilde{\Theta}_i}{=\!=\!=} \widetilde{\Theta}_i \boldsymbol{\pi}_i \tag{46}$$

$$\text{and } \widetilde{\boldsymbol{y}} = \sum_i \widetilde{\Theta}_i \boldsymbol{\pi}_i \tag{47}$$

$$\stackrel{\text{def}\widetilde{\Theta}}{=\!=\!=} \widetilde{\Theta} \boldsymbol{\pi} \tag{48}$$

where ν is the frequency chunk and $\widetilde{\mathcal{T}}_\nu$ is the normalized spectral PSF. In short, $\widetilde{\Theta}$ maps the vector of spatio-spectral coefiscients $\boldsymbol{\pi}$ for all pixels to the spectral cube $\widetilde{\boldsymbol{y}}$, taking into account the local spectral PSFs.

The algorithm is described in detail in Alg. 1. Particular attention needs to be given to step 5 where we estimate the best local model by minimizing a cost function. Different cost functions give different variations of HMP. Relaxing the constraint C_1, we can for instance set $C_0\{\widetilde{\Theta}_I \boldsymbol{d\pi}_I | \widetilde{\boldsymbol{\delta y}}, \boldsymbol{Q}\}$ as

$$C_0\{\widetilde{\Theta}_I \boldsymbol{d\pi}_I | \widetilde{\boldsymbol{\delta y}}, \boldsymbol{Q}\} \stackrel{\text{def}}{=} \left\| \widetilde{\boldsymbol{y}} - \widetilde{\Theta}_i \boldsymbol{\pi}_i \right\|_Q \tag{49}$$

where the $\mathcal{L}_2$ norm $\|\boldsymbol{x}\|_Q = \boldsymbol{x}^T \boldsymbol{Q}^{-1} \boldsymbol{x}$ of $\boldsymbol{x}$ is computed for the metric $\boldsymbol{Q}$, with $\boldsymbol{Q}$ being in practice a tapering function. The least-squares solution is then given by the pseudo-inverse

$$\widehat{\boldsymbol{\pi}}_i = \left[\widetilde{\Theta}_i^T \boldsymbol{Q} \, \widetilde{\Theta}_i \right]^{-1} \widetilde{\Theta}_i^T \boldsymbol{Q} \, \widetilde{\boldsymbol{y}}. \tag{50}$$

Alternatively, we can use a Non-Negative Least Squares (NNLS) optimisation in step 5 (Alg. 2) by setting $C_0\{\widetilde{\Theta}_I \boldsymbol{d\pi}_I | \widetilde{\boldsymbol{\delta y}}, \boldsymbol{Q}\}$ as in Eq. (49) while constraining the solution using $C_1\{\boldsymbol{x}\} \stackrel{\text{def}}{=} (\boldsymbol{x} > 0)$. In our experience the HMP-NNLS gives the best results in reconstructing extended emission.

4.2. Wide-band joint subspace deconvolution

In this section, we describe the SSD (SubSpace Deconvolution algorithm). It is a generic hybrid joint deconvolution algorithm that uses subspace optimisation. We present in Sect. 4.2.1 the generic scheme for subspace optimisation in the framework of deconvolution, and in Sect. 4.2.2 we present one such algorithm that uses a genetic algorithm in the optimisation step (SSDGA).

4.2.1. Subspace optimisation for deconvolution

It is well known that deconvolution algorithms based on Matching-Pursuit solvers (specifically CLEAN) are not robust in the deconvolution of extended emission. Joint deconvolution algorithms are more robust, as demonstrated by Garsden et al. (2015), but are not useful with large images since their sizes can exceed $\left(10^{4-4.5} \times 10^{4-4.5}\right)$ pixels. Indeed, Eq. (48) is costly to invert because $\widetilde{\Theta}$ is expensive to apply[10] the data. Therefore in order to make joint deconvolution practical with real life

---

[10] $\widetilde{\Theta}$ is done computationally by degridding.

data-sets, we aim at incorporating it in a matching pursuit-type scheme. As for HMP (Sect. 4.1), the idea is to decompose the signal into a basis function but here the parameter space at each iteration is not a set of coefficients for one pixel only, but for a subset $\mathcal{I}$ of pixels in the spectral cube (an island).

To illustrate the idea of SSD, consider the global transfer function in Eq. (48). Since the convolution matrix is diagonally dominant (the PSF goes to zero far from the center), the main idea is that distant regions can be deconvolved separately. This amounts to building an operator $\widetilde{\widetilde{\Theta}}$ with zeros where $\widetilde{\Theta}$ is considered to be negligible such that $\widetilde{\boldsymbol{y}} \approx \widetilde{\widetilde{\Theta}} \boldsymbol{\pi}$, and the deconvolution is done jointly within each subspace $\{\boldsymbol{\pi}_\mathcal{I}\}$ of the global $\{\boldsymbol{\pi}\}$ parameter space. This approximation will however lead to biases in the estimate $\widehat{\boldsymbol{\pi}}$ of $\boldsymbol{\pi}$, because the contribution of the sky in island $\mathcal{I}$ to the observed flux in island $\mathcal{I}'$ has been neglected.

This will happen for example when a bright (a) source in an $\mathcal{I}_a$ island has a faint (b) source ($\mathcal{I}_b$ island) in its side-lobe, and when the two islands are deconvolved independently. The faint source flux can be over- (or under-) estimated in the first iteration since the cross-contamination term is ignored. However if one computes the global residual map in a second iteration, most of the side-lobe of source (a) has been properly removed at $\mathcal{I}_b$. If the islands are jointly deconvolved again, the sky model estimate will be better than in the previous iteration. In our experience, this algorithm has remarkable convergence properties.

The SSD algorithm is described in detail in Alg. 2. Given a residual image, in a first step the brightest regions are isolated and joint deconvolution is performed independently on groups of pixels (here called islands) using the local convolution operator $\widetilde{\Theta}_\mathcal{I}$ with $\widetilde{\Theta}_\mathcal{I} = \widetilde{\mathcal{T}}_\nu^\phi \boldsymbol{S}_\mathcal{I} \boldsymbol{\Pi}_\mathcal{I}$, where $\boldsymbol{S}_\mathcal{I}$ is an $(n_{pix} \times n_{pix}^\mathcal{I})$ matrix that maps the $n_{pix}^\mathcal{I}$ pixels of island $\mathcal{I}$ onto the full set of n_{pix} pixels. For example we can minimize the cost function by setting

$$C_0 \stackrel{\text{def}}{=} \left\| \widetilde{\boldsymbol{\delta y}} - \widetilde{\Theta}_\mathcal{I} \boldsymbol{d\pi}_\mathcal{I} \right\|_Q \tag{51}$$

where $\boldsymbol{d\pi}_\mathcal{I}$ are the differential values of the spatio-spectral coefficients in a given basic function (see Sect. 4.1).

In a second step, the union of the sky models are subtracted from the visibilities, and the visibilities are re-imaged (corresponding to the step $\widetilde{\boldsymbol{\delta y}} = \widetilde{\boldsymbol{y}} - \widetilde{\widetilde{\Theta}} \widehat{\boldsymbol{\pi}}$).

The conditions for the convergence of SSD are hard to find, but depend on the structure of $\widetilde{\widetilde{\Theta}}$ compared to $\widetilde{\Theta}$. We can estimate at step k the contribution to the observed flux $\boldsymbol{\delta y}_{\overline{\mathcal{I}},k}$ in $\mathcal{I}$ of all islands $\mathcal{I}' \neq \mathcal{I}$. If $\boldsymbol{\delta\widehat{x}}_k = \boldsymbol{x} - \widehat{\boldsymbol{x}}_k$ is the error in the estimate $\widehat{\boldsymbol{x}}$ of $\boldsymbol{x}$, we can write

$$\boldsymbol{\delta y}_{\overline{\mathcal{I}},k} = \boldsymbol{S}_\mathcal{I}^T \left(\sum_{\mathcal{I}' \neq \mathcal{I}} \widetilde{\mathcal{T}}_\nu \boldsymbol{S}_{\mathcal{I}'} \boldsymbol{S}_{\mathcal{I}'}^T \right) \boldsymbol{\delta\widehat{x}}_k. \tag{52}$$

Since each island $\mathcal{I}$ is deconvolved in its own subspace (independently of other islands), the level of the flux density bias at iteration $k + 1$ is

$$\|\boldsymbol{\delta\widehat{x}}_{\mathcal{I},k+1}\| \sim \|\boldsymbol{\delta y}_{\overline{\mathcal{I}},k}\| \tag{53}$$

$$= \sqrt{\boldsymbol{\delta y}_{\overline{\mathcal{I}},k}^T \boldsymbol{\delta y}_{\overline{\mathcal{I}},k}}. \tag{54}$$

Assuming the structures of the side-lobes of the different $\mathcal{I}'$ in $\mathcal{I}$ are uncorrelated, the power in the cross-island terms averages out in the quadratic sum, and we get

$$\|\boldsymbol{\delta\widehat{x}}_{\mathcal{I},k+1}\| \sim \sqrt{\boldsymbol{\delta\widehat{x}}_k^T \sum_{\mathcal{I}' \neq \mathcal{I}} (\boldsymbol{S}_{\mathcal{I}'}^T \widetilde{\mathcal{T}}_\nu^T \boldsymbol{S}_\mathcal{I} \boldsymbol{S}_\mathcal{I}^T \widetilde{\mathcal{T}}_\nu \boldsymbol{S}_{\mathcal{I}'}) \boldsymbol{\delta\widehat{x}}_k} \tag{55}$$

Data: $\widetilde{y}$, t, α, Q
Result: The estimate $\widehat{\pi}$ of π in the basis function $\mathcal{P}$.
1 initialization: $\widehat{\pi} \leftarrow 0$, $\widetilde{\delta y} \leftarrow \widetilde{y}$;
   /* Start n_{Cycle} major cycles   */
2 **while** max $\{\overline{y}\} > t$ **do**
    /* Deconvolve each island $\mathcal{I}$
       independently   */
3   **for** $\mathcal{I}$ in $\{\mathcal{I}\}$ **do**
4      $\widehat{d\pi_\mathcal{I}} = \underset{d\pi_\mathcal{I}}{\arg\min} \left(C_0\{\widetilde{\Theta_\mathcal{I}} d\pi_\mathcal{I} | \widetilde{\delta y}, Q\} \right) [\text{s.t. } C_1\{d\pi_\mathcal{I}\}]$;
5   **end**
6   Update sky model:
     $\widehat{\pi} \leftarrow \widehat{\pi} + \sum_\mathcal{I} \mathcal{S}_\mathcal{I} \widehat{d\pi_\mathcal{I}}$;
7   Compute the residual data:
     $\widetilde{\delta y} = \widetilde{\mathcal{M}_\nu}^{-H} \mathcal{A}^H \mathcal{W} (\mathbf{v} - \mathcal{A}\Pi\widehat{\pi}) \left(= \widetilde{y} - \widetilde{\Theta}\widehat{\pi}\right)$;
8 **end**

Algorithm 2: SSD deconvolution algorithm. Here t is a user defined flux density threshold.

$$\mathcal{G}_\mathcal{I} \overset{\text{def}}{=\!=} \sqrt{\widehat{\delta x_k^T} \mathcal{G}_\mathcal{I} \widehat{\delta x_k}}. \tag{56}$$

Here $\mathcal{G}_\mathcal{I}$ is the power in the side-lobes of all islands $\mathcal{I}'$ to islands $\mathcal{I}$. If the cross-contamination power is small enough SSD converges. For example, in the trivial case of two single pixel islands with equal flux s, and cross-contamination term p (the PSF of $\mathcal{I}$ onto $\mathcal{I}'$ and conversely), at iteration k we have

$$\|\widehat{\delta x_{\mathcal{I},k+1}}\| \sim \|\widehat{\delta x_{\mathcal{I}',k}}\| \sqrt{p^2} \tag{57}$$

$$\sim s\sqrt{g^{2^{k+1}}} \tag{58}$$

and SSD always converges.

4.2.2. An example of SSD using genetic algorithm

We have presented in Sect. 4.2.1 the SSD algorithm, which carries out joint deconvolution over a set of sub-spaces in an independent manner. In this section we detail how the genetic algorithm in SSDGA implements step 4 in Alg. 2. Specifically, we discuss an example of an SSD algorithm, where we perform step 4 (Alg. 2) using a genetic algorithm (SSDGA). Genetic algorithms are very different from convex solvers in the sense that they are (i) combinatorial and (ii) nondeterministic. While genetic algorithms are rather simple to use and very flexible, SSDGA is in principle good for the deconvolution of extended signal. We can for instance optimize the $\mathcal{L}_0$ norm which is a nonconvex problem.

   This step corresponds to fitting the residual dirty image by a spectral sky-model for each island $\mathcal{I}$, convolved by the local spectral PSF.

   Our current implementation is based on the DEAP package (Fortin et al. 2012). Each individual 'sourcekin' consists of a set of fluxes together with a spectral index. Each sourcekin is a spectro-spatial model of the sky in $\mathcal{I}$. It could also include minor axis, major axis, and position angle of a Gaussian for example. The idea consists of building and evolving the population of sourcekin, and the fitness function is set to be $\mathcal{L}_2$ in our case. An example of spectral deconvolution using SSDGA is presented in Sect. 6.
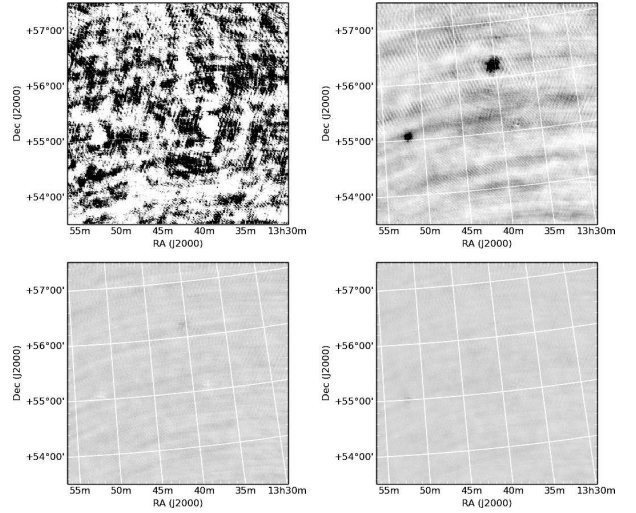
Fig. 4. Residual data for a fraction of the field of view of the simulation described in Sect. 6. The *left to right and top to bottom panels* show the iterations $\{0, 1, 2, 3\}$. As explained in Sect. 4.2.1, the SSD algorithm works differently from a matching pursuit in that it does joint deconvolution on subsets of pixels, and the estimated flux is fully removed at each iteration. The SSD has remarkable convergence properties.

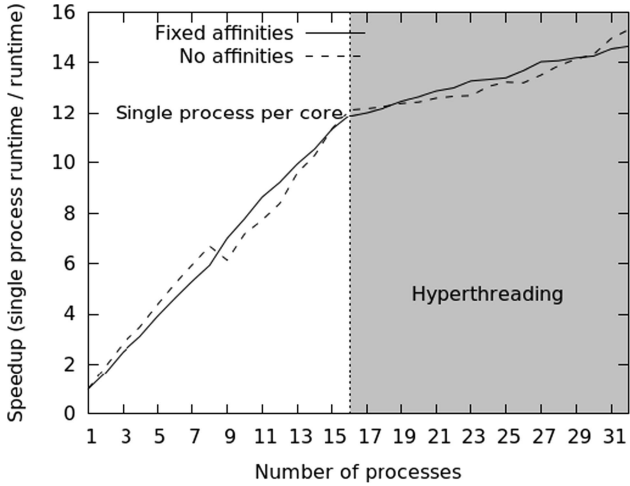## 5. Implementation, performance and features

The bulk of DDFacet is implemented in Python 2.7, with a small performance-critical core module (gridding and degridding) written in C. In this section we discuss some important aspects of the implementation. In Sect. 5.1, we describe aspects of parallelization. In Sect. 5.2, we describe how we use a baseline-dependent averaging scheme in the context of wide-field wideband spectral deconvolution, and we explain how we handle the nonregular spatial domains of Jones matrices in Sect. 5.4. In Sect. 6 we demonstrate our imaging and deconvolution framework on a single simulation.

5.1. Parallelization

The gridding, degridding and FFT operations of faceted imaging are embarrassingly parallel, as every facet can be processed completely independently. The DDFacet implementation is parallelized at the single-node level, using the Python `multiprocessing` package for process-level parallelism, and a custom-developed process manager called `AsyncProcessPool` that implements asynchronous, on-demand, multiprocessing akin to the concurrent futures[11] module found in Python 3. The bulk of the data (visibilities, *uv*-grids and and images) is stored in shared memory using the SharedArray[12] module, and a custom extension called `shared_dict`. This significantly reduces the overhead of inter-process communication. This also allows us to perform I/O and computation concurrently: a successive data chunk is read in while gridding of the previous chunk proceeds. In the minor cycle, we employ the same technique to parallelize the SSD algorithm. For HMP deconvolution (and other CLEAN-style minor loops), the minor loop is inherently serial, but a reasonable speedup is achieved with minimum effort

---

[11] https://docs.python.org/3.4/library/concurrent.futures.html
[12] https://pypi.python.org/pypi/SharedArray

Fig. 5. Speedup factor (in terms of overall wall time) obtained by running DDFacet on multiple cores. The solid line corresponds to fixed affinities (each worker process was pinned to a single CPU core), while the dashed line to no affinities (the OS scheduler was allowed to migrate processes across cores). In the former case, processes 0–7 were pinned to the first physical CPU, and 8–15 to the second CPU. This explains the slightly better performance in the no-affinities regime with ≤8 processes, as the OS scheduler was allowed to make use of the second CPU. The graph also shows significantly worse scaling in the hyperthreaded regime.

by employing the `numexpr` package[13] to vectorize large array operations.

All this allows DDFacet to make very good use of multiple cores in a NUMA architecture, maintaining high core occupancy throughout any given imaging run. We are conducting detailed performance studies and these will be the subject of a separate paper. Here we present the summary results of a simple parallelisation scaling experiment.

We perform an imaging run using 14 h of VLA (C+D configuration) data for the field around the source 3C147, in *L*-band. This totals 2 350 127 time-baseline samples, with 64 channels each, for a total bandwidth of 256 MHz. We make 5100×5100 pixel images of a $2.8° \times 2.8°$ field tiled by $23 \times 23 = 529$ square facets, in two frequency bands of 128 MHz each. A (rotating) primary beam model is applied on a per-facet basis. We run 5 major cycles of HMP CLEAN, down to an absolute flux threshold of 0.4 mJy.

Our test machine has two Intel Xeon E5, Sandy Bridge class CPUs, each with 8 physical cores and 16 virtual cores (hyperthreading enabled). In serial mode, i.e. with all operations running on a single core, we measure a total "wall time" for this imaging run of about 12 h. Ninety four% of this time is spent in the gridding. We then increase the number of parallel processes, and plot the resulting speedup factor (in terms of wall time, thus including all overheads) in Fig. 5. We see exemplary linear scaling of performance up to 16 processes (i.e. to the point where each physical core is occupied by a single process). Beyond this point, the scaling relation declines, as processes running on virtual cores start competing for resources of a single physical core. Note that a speedup factor of 12 from 16 cores is excellent efficiency: a quick calculation shows that this corresponds to 98% of the computation being parallelized.

From this we can conclude that our parallel implementation scales linearly with available physical cores, while the benefits of hyperthreading are marginal in comparison. We also find that the computational cost of the gridding step dominates overall processing. DDFacet therefore implements two strategies for reducing the overall cost of gridding: baseline-dependent averaging (BDA) and sparsification.

5.2. Baseline-dependent averaging

Averaging visibilities has the effect of reducing data volumes and increasing computing efficiency. However, information is unavoidably lost in the process, and therefore inverting the Measurement Equation from the averaged (and therefore smaller) set of visibility measurements is, numerically, subject to poorer conditioning.

The metric we use to limit the loss of information is based on decorrelation effects[14], and those will indeed constrain the maximum time and frequency domain over which visibilities can be averaged. It can be seen from the RIME (Eq. (1)) that decorrelation can be caused by the variation over time, frequency and direction of (i) the Jones matrices or (ii) the sky, and most importantly (iii) the geometric phase term (the kernel *k* term in Eq. (1)). Decorrelation due (i) and (ii) largely depends on the target, the instrument and the observing frequency. For example, low-frequency $\nu \lesssim 300$ MHz data (such as that taken by the LOFAR telescope) is affected by ionospheric phase, which varies on the timescale of 10–30 s (and is also direction-dependent due to the large FoVs). At higher frequencies, tropospheric phase begins to have a similar (although effectively direction-independent) effect. The decorrelation due to (iii) is well understood and predictable. For a given direction *s*, if the phase varies linearly across the time or frequency domain $\mathcal{D}$, and one can write

$$\chi_{\mathcal{D}} = \text{sinc} \{\Delta\psi_{\mathcal{D}}\} \tag{59}$$

$$\text{with } \Delta\psi_{\mathcal{D}} = \pi\delta\boldsymbol{b}_{\mathcal{D}}^T \boldsymbol{s} \tag{60}$$

$$\text{and } \delta\boldsymbol{b}_{\mathcal{D}} = \boldsymbol{b}_{\mathcal{D}_1} - \boldsymbol{b}_{\mathcal{D}_0}, \tag{61}$$

where $\boldsymbol{b}_{\mathcal{D}_0}$ and $\boldsymbol{b}_{\mathcal{D}_1}$ are the baseline vectors (in units of wavelength) at the edges of the domain $\mathcal{D}$.

Interferometric data is typically conservatively averaged at best, using a common time-frequency bin across all baselines that corresponds to no more than a few percent amplitude loss on the longest baseline for a source on the edge of the field of view. Several authors have come to the conclusion that this is sub-optimal, and that one could use BDA instead (see Cotton 1989, 1999; Atemkeng et al., in prep.), with more aggressive averaging on the shorter baselines, since for a given direction, time and frequency domain, they decorrelate less than the longer baselines. With core-heavy arrays such as MeerKAT and SKA1, the potential storage savings of BDA can be substantial, since the data sets are dominated by short spacings.

It is important to keep in mind that, for purposes of data storage, the largest time/frequency domain to which any given baseline may be averaged is given by

$$\mathcal{D} = \min \{\mathcal{D}_J, \mathcal{D}_s, \mathcal{D}_k\}, \tag{62}$$

where $\mathcal{D}_J$, $\mathcal{D}_s$, and $\mathcal{D}_k$ are the domains corresponding to an acceptable decorrelation for (i) the Jones matrices, (ii) the sky,

---

[13] https://pypi.python.org/pypi/numexpr

[14] Decorrelation is produced by averaging of complex-valued Jones matrices. It is easy to see, for instance, that averaging a rotating complex scalar will result in a loss of amplitude.
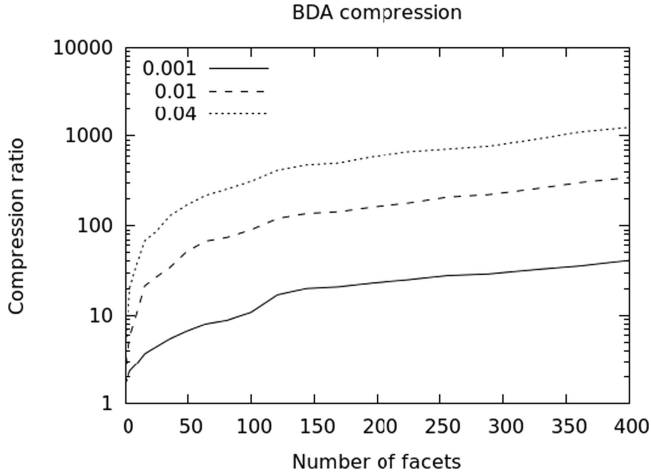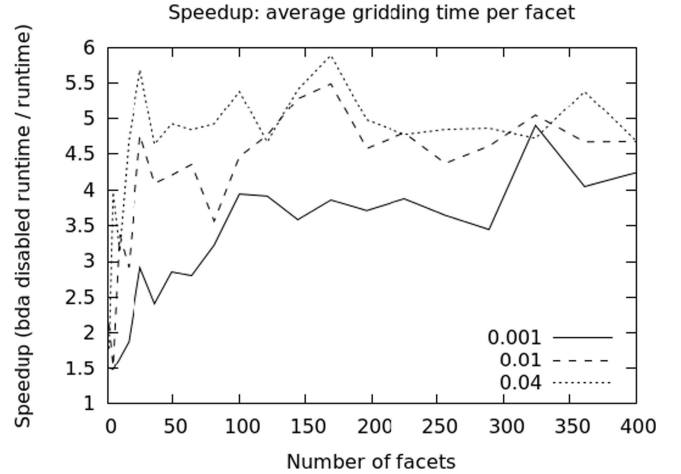
Fig. 6. Compression ratios achieved with baseline-dependent averaging, as a function of facet number (thus facet size), for several different decorrelation levels, for a VLA B-configuration observation.

Fig. 7. Gridding speedup factors achieved with baseline-dependent averaging, as a function of facet number (thus facet size), for several different decorrelation levels. Observation configuration as per previous figure.

and (iii) the geometric phase term respectively. In the presence of DDEs such as the ionosphere, $\mathcal{D}_J$ is the term that typically constrains $\mathcal{D}$, because there is no way to correct the stored visibilities for the DDEs; rather, we need to apply Jones matrices to the data during imaging, as described in Sect. 3 (see Eqs. (17) and (18)). For example, consider a single baseline at low frequency, having a decorrelation time scale of the geometric phase term of the order of a few minutes. As the Jones matrices corresponding to the ionosphere are direction dependent, and vary on an approximately ten second time scale, one cannot average the stored data on time scales larger than seconds without substantially degrading the imaging response.

Even if the storage economies of BDA are not realized (and are in any case limited by $\mathcal{D}_J$), DDFacet derives computational economies from this technique. If one assumes that the sky term is constant across $\mathcal{D}_k$, one can average the 2×2 visibilities over $\mathcal{D}_k$, while applying the per-facet phase rotations, and the (direction-dependent) Jones matrices, to each individual visibility. This is done on-the-fly in the gridder and degridder code. The actual gridding (or degridding) is then done once per averaging domain, rather than once per visibility. The resulting savings can be substantial, since averaging visibilities involves fewer FLOPS than applying the convolution kernel inherent to gridding or degridding. The BDA gridding algorithm implemented in DDFacet is presented in detail in Alg. 3. We note that a similar approach has been implemented in recent versions of WSCLEAN (Offringa et al. 2014), without the direction-dependent Jones correction.

On-the-fly BDA in the context of faceting offers an interesting performance trade-off. Note that $\mathcal{D}_k$ is determined by facet size, rather than the full FoV size. While imaging smaller facets, more agressive BDA may be applied, since more visibilities can be averaged before a given decorrelation level is reached. Note that at the limit of single-pixel facets, per-facet BDA reduces to averaging the entire (phase-shifted) dataset, which is effectively the same as doing a DFT. Figure 6 shows the compression ratio achieved for a few fixed decorrelation levels, as a function of number of facets (across the same FoV), for a VLA B-configuration obervation. Note that more core-heavy configurations such as MeerKAT and SKA1-MID should be able to achieve even higher compression ratios. Figure 7 shows the resulting speedup in gridding time per facet. Note that the

speedup flattens out at around a factor of 4. Presumably, at this point the gridder performance becomes dominated by memory access. Thus, the computational cost of using numerous smaller facets (resulting in more gridding/FFT operations) is partially offset by the computational savings of increased BDA within each facet.

5.3. Sparsification

Recent developments in compressive sensing (CS) theory (see, e.g., Carrillo et al. 2012, and references therein) have provided new mathematical insights into imaging and deconvolution. In particular, CS shows that sparse signals, i.e. signals with limited support in some dictionary, can be successfully recovered from a much smaller number of measurements than that required by e.g. Nyquist sampling, provided a few mathematical criteria (in particular, the Restricted Isometry Property, or RIP) are met. Traditional CLEAN has been shown to be a variation of a CS algorithm.

When multiple major cycles are performed by a CLEAN-like algorithm (as is necessary for even modestly high dynamic range imaging), the models recovered during early major cycles tend to be extremely sparse. This is simply due to the inherent structure of the radio sky: bright sources are few and far between, so it is quite typical that early major cycles of CLEAN affect only a relatively small number of components. Under these conditions, it directly follows from CS theory that we can discard a random subset of uv-samples in the early major cycles (the randomness of the selection ensures that the RIP is maintained), and still recover the same model components, as long as the image signal-to-noise ratio (S/N) remains sufficiently high.

DDFacet implements this idea as the sparsification feature. The implementation naturally interfaces with the BDA gridder. At each major cycle, a user-specified percentage of visibility blocks (i.e. BDA blocks) is tagged as "unlucky" and omitted from the gridding process entirely. Effectively, the uv-coverage is randomly sparsified by a given factor, without changing the relative sampling density. The gridding loop is therefore accelerated by the same factor. The sparsification factor can be set to decrease in successive major cycles, since, at the very least, the final major cycle should be done on the full data.

Data: $\mathbf{v}, \{\mathcal{D}\}$
Result: The dirty cube $\boldsymbol{y}, \widetilde{\boldsymbol{y}}$;

1 $\boldsymbol{y} = \mathbf{0}$;
2 **for** φ *in* $\{\varphi\}$ **do**
3 $\quad \boldsymbol{g}_{s_\varphi} = \mathbf{0}; w_t = 0$;
4 $\quad \widetilde{\mathbf{M}^{b_\nu}} = \mathbf{0}$;
5 $\quad$ **for** $\mathcal{D}$ *in* $\{\mathcal{D}\}$ **do**
6 $\quad\quad \mathbf{V}_{\mathcal{D}} = \mathbf{0}; \boldsymbol{b}_m = (u_m, v_m, w_m) = \mathbf{0}$;
7 $\quad\quad$ **for** $\left(\boldsymbol{b}, w, \mathbf{G}_p, \mathbf{G}_q, \mathbf{v}_{pq} \right)$ *in* $\mathcal{D}$ **do**
8 $\quad\quad\quad \boldsymbol{b}_m \leftarrow \boldsymbol{b}_m + \mathbf{C}_{s_\varphi} \boldsymbol{b}$ (see Eq. (14));
9 $\quad\quad\quad w_m \leftarrow w_m + w$;
10 $\quad\quad\quad \mathbf{M}^{b_\nu} = \mathbf{G}_q^* \otimes \mathbf{G}_p$;
11 $\quad\quad\quad \widetilde{\mathbf{M}^{b_\nu}}^2 \leftarrow \widetilde{\mathbf{M}^{b_\nu}}^2 + w\mathbf{M}^{b_\nu H}\mathbf{M}^{b_\nu}$;
12 $\quad\quad\quad \mathbf{v}_{\mathcal{D}} \leftarrow \mathbf{v}_{\mathcal{D}} + \mathbf{M}^{b_\nu}\mathbf{v}_{pq} \exp\left(-2\pi i \boldsymbol{b}^T \boldsymbol{s}_\varphi\right)$;
13 $\quad\quad$ **end**
14 $\quad\quad \boldsymbol{b}_m \leftarrow \boldsymbol{b}_m / w_m$;
15 $\quad\quad \boldsymbol{g}_{s_\varphi} \leftarrow \boldsymbol{g}_{s_\varphi} + \boldsymbol{C}_{s_\varphi, b_m} \mathrm{Vec}\{\mathbf{V}_{\mathcal{D}}\}$;
16 $\quad\quad w_t \leftarrow w_t + w_m$;
17 $\quad$ **end**
18 $\quad \boldsymbol{g}_{s_\varphi} \leftarrow \boldsymbol{g}_{s_\varphi} / w_t$;
19 $\quad \widetilde{\boldsymbol{y}} \leftarrow \widetilde{\boldsymbol{y}} + \boldsymbol{S}_{s_0} \widetilde{\mathcal{M}_{\nu s_\varphi}}^{-H} \mathcal{F}^H \boldsymbol{g}_{s_\varphi} / w_t$;
20 **end**

Algorithm 3: BDA gridder. Here $\boldsymbol{g}_{s_\varphi}$ is the grid used for facet φ, $\boldsymbol{C}_{s_\varphi, b_m}$ is the convolution function for the w-coordinate of $\boldsymbol{b}_m$.

What is a reasonable sparsification factor? Consider a typical VLA or MeerKAT L-band observation of several hours' duration. This will contain 10^6–10^7 time-baseline samples (per each frequency channel), and offer an image dynamic range of, conservatively, 10^4 (in the presence of very bright sources, this can go to 10^6 and above). Assuming a major cycle gain of 0.1, the first major cycle will clean down to 1000σ, and the second major cycle to 100σ, where σ is the noise rms. A sparsification factor of 100 will reduce S/N by a factor of 10, with a negligible effect on overall uv-coverage, and with the second major cycle remaining well above the noise floor. (In practice, we find that sparsifying above a factor of 100 is not really useful, since at that point the runtime becomes dominated by I/O rather than gridding.) For 5 major cycles, a typical list of sparsification factors would be 100, 100, 30, 10, 1. This results in very substantial acceleration of the first four major cycles.

5.4. Nonregular tessellation

Since astrophysical flux density is not uniformly distributed over the sky, a reasonable DDE calibration strategy consists of clustering sets of sources together, and estimating time–frequency-antenna dependent Jones matrices in each of those directions. As shown by Williams et al. (2016) and van Weeren et al. (2016), it is necessary in that context to image the residual data in the nonregular spatial domains within which the Jones matrices have been estimated.

DDFacet has the ability to image irregularly tessellated images while using 2D FFTs. This is done by providing it with an arbitrary set of nodes (i.e. facet centers). A Voronoi tessellation is then computed internally to generate a polygon file, where each facet is associated with a unique polygon. An image mask is constructed for each polygon. In order to form the residual

image, each facet image is multiplied by its corresponding mask, before being stacked onto the combined residual image. We do the converse operation for the degridding.

5.5. Primary beam models

The present implementation of DDFacet can correct for two classes of DDEs: direction-dependent gain solutions derived from a calibration procedure, and/or a priori known DDEs imposed by the primary beam pattern.

The first class of DDEs is specified as an input list of directions and (frequency-dependent) per-antenna Jones terms associated with that direction. Facets are then determined by the nonregular tesselation procedure described above, and imaging proceeds on a per-facet basis, by applying the correct DDE solution (and primary beam E-Jones, computed at facet center) per facet, as described in Sect. 3. In the absence of DD gain solutions, imaging can be done with primary beam corrections alone. In this case the field of view is split into a user-supplied number of square facets, and the primary beam E-Jones is again computed at each facet's center.

Two primary beam modes are currently implemented. In FITS mode, the user specifies the primary beam pattern as a set of 8 FITS cubes[15] giving the real and imaginary components of the 2×2 E-Jones elements, as a function of frequency and direction in the antenna frame. DDFacet then computes parallactic angle rotation, and interpolates the E-Jones value appropriate to each facet's center, as a function of time and frequency. FITS mode is suitable to describing dish arrays. In LOFAR mode, the primary beam corresponding to a LOFAR station is computed using the LofarStationResponse class.

6. Simulations

In this section we discuss in detail a test case for the framework developed throughout this paper. We simulate a small, semi-realistic, LOFAR LBA data-set. The data consists of an interferometric data-set with 630 baselines (36 LOFAR stations), 2.8 h long, with 5 frequency channels spanning the range 30 to 70 MHz. The station are configured in LBA_INNER mode (van Haarlem et al. 2013) in order to provide us with the widest field of view. The HPBW is on the order 5.77 degrees at 60 MHz.

The simulated sky consists of ~100 point sources with 50-MHz intrinsic flux densities $S_{50} = s^{1.5}$ with $s \sim \mathcal{U}\{0, 100\}$ Jy, and spectral indices $\alpha_{50} \sim \mathcal{U}\{-1, 1\}$, where $\mathcal{U}$ is the uniform distribution. To make the case more difficult yet, we add a bright 10^4 Jy off-axis source.

Apart from the LBA_INNER station beam being applied, scalar time-frequency-direction dependent Jones matrices are applied to the predicted visibilities. In particular, the Jones matrices have a random phase and amplitude term that varies sinusoidally with time, and this random realization is repeated for each antenna, frequency and direction. The typical amplitude variation is on the level of 0.1, while the phase variation is on the level of 0.1π. This scheme guarantees (i) nonunitarity of the Jones matrices (and Mueller matrices), and (ii) baseline-dependency of the resulting Mueller matrices (see Sect. 3.5).

The Jones matrices together with the LOFAR LBA_INNER beam are provided to DDFacet and the sky is tessellated as described in Sect. 5.4. The image is 8019×8019 pixels with a

---

[15] In principle it should also be possible to specify a different pattern per antenna, although this option is currently not exposed in the interface.
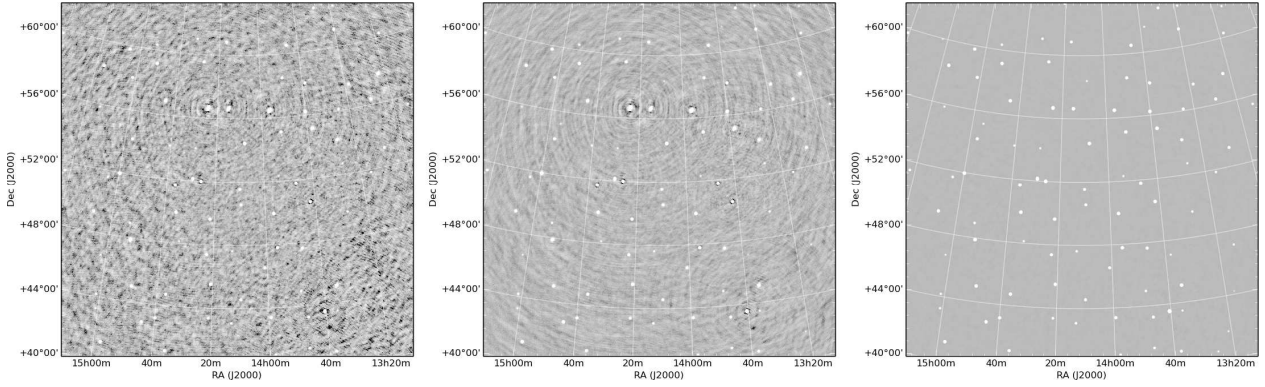
Fig. 8. Deconvolved images for the simulated data-set described in Sect. 6. The *left panel* shows the restored image not taking the beam into account and not using wideband deconvolution. *Central panel* shows the result of the deconvolution taking DDE into account, but still not enabling wideband deconvolution. *right panel* shows the deconvolution result using SSDGA.
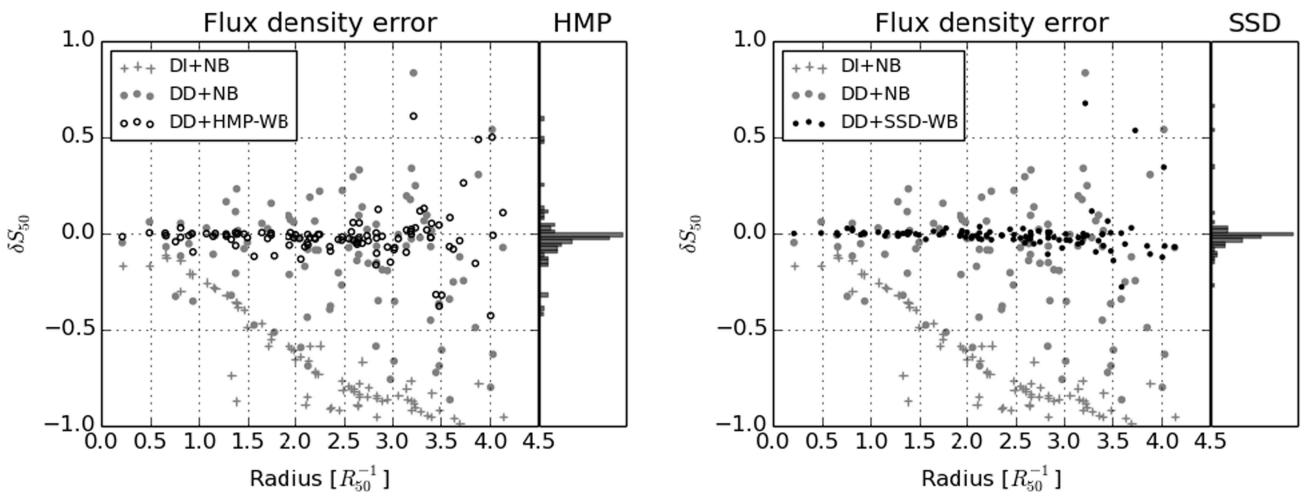
Fig. 9. Relative flux density error $\delta S_{50} = \left(\widehat{S_{50}} - S_{50}\right) S_{50}^{-1}$ between the 50 MHz flux density S_{50} and its estimate $\widehat{S_{50}}$ as a function of radius from the beam center (normalized by the half power radius at 50 MHz). The *left* and *right panels* show the results for the HMP and SSD wide band deconvolutions (empty and filled black circles respectively). The gray cross corresponds to the flux densities errors taking neither DDE correction nor wideband effects into account. The gray dots show the results when taking DDE effects, but not wideband effects into account.

pixel size of $10 \times 10''$, and when the wideband mode is enabled the residual image cube has 3 channels.

To measure the estimated sky produced by the different deconvolution algorithms, we run DDFacet in various modes:

[•]DI/DD: Direction-independent faceting or direction-dependent faceting.

NB/WB: Narrow-band and wideband observations. In the latter case the intrinsic spectral variability is taken into account either by the HMP (HMP-WB) or by the SSDGA (SSD-WB).

The results are presented in Figs. 8–10. Figure 8 shows that, as expected, the residuals are much lower when deconvolved with DD+NB and DI+WB. In Fig. 9, we plot the relative flux error $\delta S_{50} = \left(\widehat{S_{50}} - S_{50}\right) S_{50}^{-1}$ as a function of the distance to the beam center. It is quite clear from the plot that, as expected, the DI+NB gives biased flux density estimates and higher residual values (see Fig. 8). The main component of the bias is corrected using DD+NB, but the flux density scatter is high. With HMP or SSD deconvolution, the dynamic range increases, and the error on both flux densities and spectral indices decreases. In Fig. 10, we plot the error $\delta\alpha_{50}$ in spectral index estimate as a function of the
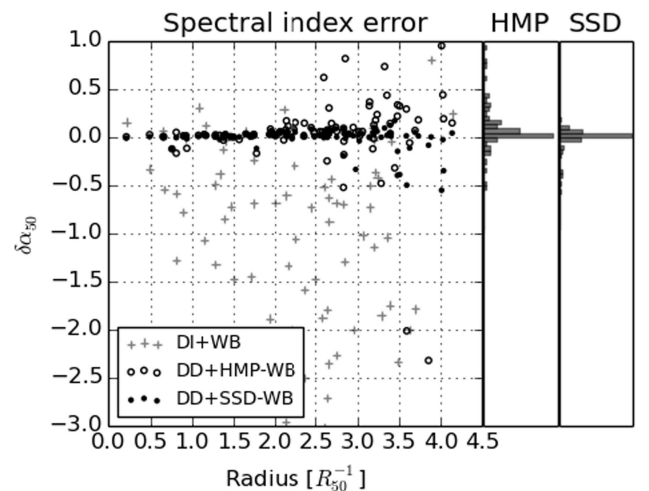
Fig. 10. Error $\delta\alpha_{50} = \widehat{\alpha_{50}} - \alpha_{50}$ between the spectral index α_{50} and its estimate $\widehat{\alpha_{50}}$, as a function of the distance from the beam center. The gray crosses show the spectral index estimates when the DDEs are not taken into account.

distance from the beam center. While DI+WB produces biased spectral index estimates, the HMP or SSD algorithms properly recover them (with SSD doing slightly better).

7. Overview and future work

In this paper we have presented a mathematical framework for wideband wide-field spectral deconvolution that can take generic DDEs into account, as well as an implementation of this framework called the DDFacet imager. This has a number of unique features not available in other imaging and deconvolution frameworks:

- A wide-field coplanar faceting scheme is implemented. This is a generalization of the Kogan & Greisen (2009) scheme. Nontrivial facet-dependent w-kernels are used to correct for noncoplanarity within the facets.
- Generic, spatially discrete, time-frequency-baseline-direction-dependent full polarisation Jones matrices can be taken into account in the imaging and deconvolution steps[16].
- As shown in Sect. 3.5 and 3.6, to account for time-frequency-baseline dependent Mueller matrices, we compute a direction dependent PSF for use in the minor cycle of deconvolution.
- The above also allows for the effects of time and bandwidth averaging to be explicitly incorporated into deconvolution.

In order to accurately deal with large fractional bandwidth of modern interferometers, one has to estimate the spectral properties of the sky term in the Measurement Equation. We have implemented two new wideband deconvolution algorithms:

- A hybrid matching pursuit algorithm (HMP, see Sect. 4.1), with similarities to MTMS-CLEAN (Rau & Cornwell 2011).
- A hybrid joint deconvolution algorithm that we call Subspace Deconvolution (SSD, see Sect. 4.2). This does joint deconvolution on subsets of pixels (islands), while treating the islands independently. For the former, we have implemented a genetic algorithm to perform the nonlinear optimisation step (Sect. 4.2.2)

Finally, a few interesting and innovative technical features are incorporated:

- A general tessellation scheme that supports both regular (square) and nonregular (Voronoi tesselation) facets.
- On-the-fly baseline-dependent averaging within the gridder and degridder, resulting in a factor of several speedup of these operations.
- A sparsification scheme that dramatically accelerates the initial (shallow) major cycles, when using a CLEAN-style deconvolution loop.
- Completely asynchronous multiprocessing, which does I/O and computation concurrently, and achieves excellent scaling properties on NUMA architectures.

The current version of DDFacet can deal with full polarisation deconvolution, and can take into account externally defined Jones matrices and/or beam patterns. It has been successfully tested with data diverse telescopes such as LOFAR, VLA, MeerKAT AR1 and ATCA.

Further development of DDFacet is proceeding in a number of directions. These include (i) extending the basis set of

SSDGA and HMP with Gaussian components for better deconvolution of extended emission, (ii) implementing other types of SSD algorithms, using convex solvers for the optimisation steps, (iii) implementing CS-derived minor cycle algorithms, (iv) back-porting existing multiscale CLEAN algorithms into the framework. On the more technical side, we are working on incorporating (v) GPU-based gridders and degridders, and (vi) the accelerated Direct Fourier Transform implemented in MONT-BLANC (Perkins et al. 2015) for high dynamic range imaging. Finally, a distributed implementation of the framework is being designed.

References

Bhatnagar, S., Cornwell, T. J., Golap, K., & Uson, J. M. 2008, A&A, 487, 419
Bhatnagar, S., Rau, U., & Golap, K. 2013, ApJ, 770, 91
Bridle, A. H., & Schwab, F. R. 1999, in Synthesis Imaging in Radio Astronomy II, eds. G. B. Taylor, C. L. Carilli, & R. A. Perley, ASP Conf. Ser., 180, 371
Carrillo, R. E., McEwen, J. D., & Wiaux, Y. 2012, MNRAS, 426, 1223
Cornwell, T. J. 2008, IEEE J. Sel. Top. Signal Process., 2, 793
Cornwell, T. J. & Perley, R. A. 1992, A&A, 261, 353
Cornwell, T. J., Golap, K., & Bhatnagar, S. 2008, IEEE J. Sel. Top. Signal Process., 2, 647
Cotton, W. D. 1989, in Synthesis Imaging in Radio Astronomy, 6, 233
Cotton, W. 1999, in Synthesis Imaging in Radio Astronomy II, 180, 357
Dabbech, A., Ferrari, C., Mary, D., et al. 2015, A&A, 576, A7
Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. 2012, J. Mach. Learn. Res., 13, 2171
Garsden, H., Girard, J. N., Starck, J. L., et al. 2015, A&A, 575, A90
Hamaker, J. P., Bregman, J. D., & Sault, R. J. 1996, A&AS, 117, 137
Jacob, J. C., Katz, D. S., Prince, T., et al. 2004, The montage architecture for grid-enabled science processing of large, distributed datasets (Pasadena, CA: Jet Propulsion Laboratory, NASA)
Kazemi, S., Yatawatta, S., Zaroubi, S., et al. 2011, MNRAS, 414, 1656
Kogan, L., & Greisen, E. W. 2009, Faceted imaging in AIPS, Tech. Rep. 113, (National Radio Astronomy Observatory. Associated Universities, Inc.)
Noordam, J. E., & Smirnov, O. M. 2010, A&A, 524, A61
Offringa, A. R., McKinley, B., Hurley-Walker, N., et al. 2014, MNRAS, 444, 606
Perkins, S. J., Marais, P. C., Zwart, J. T. L., et al. 2015, Astron. Comput., 12, 73
Rau, U., & Cornwell, T. J. 2011, A&A, 532, A71
Smirnov, O. M. 2011, A&A, 527, A106
Smirnov, O. M., & Tasse, C. 2015, MNRAS, 449, 2668
Tasse, C. 2014a, ArXiv e-prints [arXiv:1410.8706]
Tasse, C. 2014b, A&A, 566, A127
Tasse, C., van der Tol, S., van Zwieten, J., van Diepen, G., & Bhatnagar, S. 2013, A&A, 553, A105
Thompson, A. R., Moran, J. M., & Swenson, Jr., G. W. 2001, Interferometry and Synthesis in Radio Astronomy, 2nd edn. (New York: Wiley)
van Haarlem, M. P., Wise, M. W., Gunst, A. W., et al. 2013, A&A, 556, A2
van Weeren, R. J., Williams, W. L., Hardcastle, M. J., et al. 2016, ApJS, 223, 2
Williams, W. L., van Weeren, R. J., Röttgering, H. J. A., et al. 2016, MNRAS, 460, 2385
Yatawatta, S. 2013, Exp. Astron., 35, 469

---

[16] This is not the case for Bhatnagar et al. (2008); Tasse et al. (2013) that are using A-projection and that need (i) spatially smooth DDE effects, and (ii) time-frequency-baseline independent Jones matrices for the PSF to be direction-independent.