Multi-frequency image reconstruction for radio-interferometry with self-tuned regularization parameters

Rita Ammanouil, André Ferrari, Rémi Flamary, Chiara Ferrari and David Mary

Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Lab. J.-L. Lagrange, France

Abstract—As the world's largest radio telescope, the Square Kilometer Array (SKA) will provide radio interferometric data with unprecedented detail. Image reconstruction algorithms for radio interferometry are challenged to scale well with TeraByte image sizes never seen before. In this work, we investigate one such 3D image reconstruction algorithm known as MUFFIN (MUlti-Frequency image reconstruction For radio INterferometry). In particular, we focus on the challenging task of automatically finding the optimal regularization parameter values. In practice, finding the regularization parameters using classical grid search is computationally intensive and nontrivial due to the lack of ground-truth. We adopt a greedy strategy where, at each iteration, the optimal parameters are found by minimizing the predicted Stein unbiased risk estimate (PSURE). The proposed self-tuned version of MUFFIN involves parallel and computationally efficient steps, and scales well with large-scale data. Finally, numerical results on a 3D image are presented to showcase the performance of the proposed approach.

I. INTRODUCTION

The Square Kilometer Array (SKA) is an international project intended to build the largest radio telescope in the world. This telescope will produce interferometric data by combining signals from each pair of antennas in an array measuring radio emissions from a given area in the sky [1]. With up to one million receiving elements, the SKA will achieve (sub-) arcsec spatial resolution and provide measurements over hundreds of frequency bands [2], [3]. When operational, it is expected that the SKA will produce unprecedented images of the sky in the lowest-frequency part of the electromagnetic spectrum. However, this will come at the cost of a tremendous volume with data sets sizing up to several hundreds of Pbytes/yr. As a result, exploiting these massive data sets in a manageable amount of time will pose considerable challenges [4].

In radio interferometry, most of the data exploitation and analysis relies on producing a proper image of the sky from incomplete Fourier measurements taken by the telescope. For perfectly calibrated measurements, this reduces to a deconvolution problem in the image plane aimed at recovering the sky image given the dirty image (the convolved image) and the dirty beam (the point spread function). When directional dependent effects and anisoplanetic effects are only partially corrected, the deconvolution takes place in the so-called "minor loop" of the image reconstruction pipeline [5]. The CLEAN method [6] and its multi-frequency extension [7] are among the most popular methods for deconvolution. CLEAN assumes that the sky is mostly empty, and contains a few point sources. The desired "clean" image is obtained by successive detections and subtractions of the brightest sources in the dirty image. However, CLEAN and most of the algorithms that belong to the CLEAN family are not tailored for large-scale data. For this reason, a lot of effort has gone into designing scalable methods based on convex optimization [8]–[14]. These algorithms, which unrestrictedly operate in the image or visibility plane, adopt sparse models in an adequate domain, and

perform deconvolution by minimizing a data fidelity term and one or more regularizations. The scalability of these algorithms comes from the fact that computationally intensive steps are performed in parallel. In the mono-frequency case [8], [9], parallel computations rely on splitting the data into multiple blocks and exploiting randomization over data blocks at each iteration. In the multi-frequency case [12], [13], parallel computations exploit the fact that each node can be assigned computations at a certain wavelength band. In addition to parallelization, scalability is achieved in [12] by avoiding expensive operations like matrix inversions. Furthermore, the authors of [13] reduce the memory load by adopting a fully distributed memory storage architecture. Despite their various advantages, the efficiency of convex optimization techniques is usually governed by the choice of appropriate regularization parameters. The scope of this work is to discuss a scalable method for setting the optimal parameter values.

A typical strategy for parameter tuning is to do a grid search and chose the parameter that gives the best image. However, judging the estimated image quality is a nontrivial and complicated task due to the lack of ground truth. In addition to this, grid search is very time consuming when dealing with large-scale data. The literature on automatic parameter selection provides various quantitative measures for evaluating the quality of an estimated image. These measures can be broadly classified as those based on the discrepancy principle [15], [16], the L-curve method [17], [18], generalized cross-validation (GCV) [19], and the Stein Unbiased Risk Estimator (SURE) [20], [21]. Among these measures, the SURE is the most attractive one in image processing since it provides an unbiased estimate of the Mean Square Error (MSE). Furthermore, the principles underlying the SURE have been successfully adopted for image deconvolution [22], [23]. This work applies a weighted version of SURE, Predicted SURE (PSURE), proposed in [22] to the MUFFIN (MUlti-Frequency image reconstruction For radio INterferometry) algorithm [12] in a greedy way similarly to [23]. This new version of MUFFIN can be seen as a self-tuned extension of the original algorithm. We show that the steps required for self-tuning can be performed in parallel and in a memory efficient way. To the best of our knowledge, [24] is the only work to propose a self-tuning approach. Their approach was restricted to the case of mono-frequency radio-interferometric image deconvolution, and consists in setting the regularization parameter proportionally to estimated noise variances. However, this strategy is confined to the particular case of an inverse problem with a sparse synthesis prior and solved using a primal algorithm. In contrast with [24], the application of PSURE is not limited to MUFFIN and can be extended to most deconvolution algorithms based on convex optimization.

The remainder of this article is organized as follows. Section II describes the deconvolution problem solved using MUFFIN. Section III reviews the PSURE and derives its expression for MUFFIN. Section IV explains how this measure can be used to find the optimal regularization parameters and the number of iterations. Section V presents the experimental results.

II. Problem description

When all direction dependent effects (ionosphere, antenna gains, ...) and the non-coplanar baseline effects are correctly compensated, the collection of measured visibilities reduces to a sampled Fourier transform of the sky. Various algorithms [25], [26] transform the sampled visibilities from the Fourier to the image domain resulting in the dirty image. The image reconstruction pipeline generally alternates between major and minor loops, that estimate the measurement operator and deconvolve the dirty image respectively [5]. The deconvolution step recovers the sky image knowing the dirty image and the convolution operator. Let $\boldsymbol{x}_\ell^\star$ be the column vector collecting the sky intensity image at wavelength λ_ℓ, with $\ell = 1, \ldots, L$ and L the number of spectral channels. The dirty image $\boldsymbol{y}_\ell$ at wavelength λ_ℓ is related to the sky image by:

$$\boldsymbol{y}_\ell = \boldsymbol{H}_\ell \boldsymbol{x}_\ell^\star + \boldsymbol{n}_\ell, \tag{1}$$

where $\boldsymbol{H}_\ell$ represents a convolution by the point spread function at λ_ℓ, and $\boldsymbol{n}_\ell$ is a perturbation vector accounting for noise and modeling error. Eq. (1) defines an ill-posed deconvolution problem owing to the partial coverage of the Fourier plane. Various convex optimization techniques have been proposed the solve this problem. In what follows, we briefly review the approach adopted in [12] where the authors solved this problem in a cost minimization framework, by adding to the data fidelity term a regularization term f_{reg} representing prior on $\boldsymbol{x}_1^\star, \ldots, \boldsymbol{x}_L^\star$. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ denote the corresponding optimization variables at each wavelength, and $\boldsymbol{X}$ denote the matrix $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L]$. With these notations, the cost function writes:

$$\min_{\boldsymbol{X}} \sum_{\ell=1}^{L} \frac{1}{2} \|\boldsymbol{y}_l - \boldsymbol{H}_\ell \boldsymbol{x}_\ell\|^2 + f_{\text{reg}}(\boldsymbol{X}). \tag{2}$$

The last term in eq. (2) incorporates the positivity constraint, a spatial-spectral sparse analysis prior and has the following form:

$$f_{\text{reg}}(\boldsymbol{X}) := \mathbf{1}_{\mathbb{R}+}(\boldsymbol{X}) + \mu_s \sum_{l=1}^{L} \|\mathbf{W}_s \boldsymbol{x}_\ell\|_1 + \mu_\lambda \sum_{n=1}^{N} \|\mathbf{W}_\lambda \boldsymbol{x}^n\|_1, \tag{3}$$

where $\boldsymbol{x}^n$ (the nth row of $\boldsymbol{X}$) is the spectrum associated to pixel n, $\mathbf{W}_s$ and $\mathbf{W}_\lambda$ are the operators associated with, respectively, the spatial and spectral decomposition, μ_s and μ_λ are the corresponding regularization parameters. Without loss of generality, we considered in the experiments a union of orthogonal bases for the spatial regularization and a cosine decomposition for the spectral model widely used in the literature [8], [10], [24]. The authors of [12] solve the optimization problem defined in (2) and (3) using the MUFFIN algorithm based on the primal-dual algorithm proposed in [27], [28]. The steps required for each iteration of MUFFIN are described in Algorithm 1 where $(\cdot)_+$ is the projection on the positive orthant, and $\text{sat}(u)$ is the saturation function defined as follows:

$$\text{sat}(u) := \begin{cases} -1 & \text{if } u < -1 \\ 1 & \text{if } u > 1 \\ u & \text{if } |u| \le 1, \end{cases} \tag{4}$$

and the parameters σ and τ are fixed according to [27] in order to guarantee convergence. A major advantage of the MUFFIN Algorithm is that the most computationally demanding steps are separable w.r.t. the wavelengths, leading to a parallel and distributed implementation. More precisely, each wavelength is associated to a compute node where the time consuming steps are computed in parallel which drastically reduces the execution time. This is in contrast with the work of [14] where the most expensive step requiring a singular value decomposition is not parallelized.

Algorithm 1: MUFFIN update

1 The master computes $\boldsymbol{T} = \mu_\lambda \boldsymbol{V} \boldsymbol{W}_\lambda$. Equivalently, it computes:

$$\boldsymbol{t}^n = \mu_\lambda \boldsymbol{W}_\lambda^\dagger \boldsymbol{v}^n, \quad \text{for } n = 1 \ldots N \tag{5}$$

and sends the col. ℓ of $\boldsymbol{T}$, denoted as $\boldsymbol{t}_\ell$, to node ℓ.

2 Each compute node $\ell = 1 \ldots L$ computes sequentially:

$$\boldsymbol{x}_\ell = \tilde{\boldsymbol{x}}_\ell, \tag{6}$$
$$\boldsymbol{\nabla}_\ell = \boldsymbol{H}_\ell^\dagger (\boldsymbol{H}_\ell \boldsymbol{x}_\ell - \boldsymbol{y}_\ell) \tag{7}$$
$$\boldsymbol{s}_\ell = \mu_s \boldsymbol{W}_s^\dagger \boldsymbol{u}_\ell \tag{8}$$
$$\boldsymbol{m}_\ell = \boldsymbol{x}_\ell - \tau(\boldsymbol{\nabla}_\ell + \boldsymbol{s}_\ell + \boldsymbol{t}_\ell) \tag{9}$$
$$\tilde{\boldsymbol{x}}_\ell = (\boldsymbol{m}_\ell)_+ \tag{10}$$
$$\boldsymbol{p}_\ell = \boldsymbol{u}_\ell + \sigma \mu_s \boldsymbol{W}_s (2\tilde{\boldsymbol{x}}_\ell - \boldsymbol{x}_\ell) \tag{11}$$
$$\boldsymbol{u}_\ell = \text{sat}(\boldsymbol{p}_\ell), \tag{12}$$

and sends $\boldsymbol{x}_\ell$ and $\tilde{\boldsymbol{x}}_\ell$ to the master node.

3 The master computes sequentially, for $n = 1 \ldots N$:

$$\tilde{\boldsymbol{v}}^n = \boldsymbol{v}^n + \sigma \mu_\lambda \boldsymbol{W}_\lambda (2\tilde{\boldsymbol{x}}^n - \boldsymbol{x}^n) \tag{13}$$
$$\boldsymbol{v}^n = \text{sat}(\tilde{\boldsymbol{v}}^n) \tag{14}$$

Return : $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L]$

III. Predicted-SURE measure

The SURE provides an unbiased estimate of the MSE. However the SURE measure is only applicable in the case of image denoising, which is not the case in the present study. The concepts underlying the SURE have been extended to the case of image reconstruction using weighted versions of the SURE measure [22], [23]. In what follows, we briefly review the essential concepts behind one such measure, the PSURE measure, proposed in [22] and how it can be used to automatically select the optimal parameter values. Similarly to the SURE, PSURE provides an unbiased estimate of the weighted MSE that is only a function of the measurements. More precisely:

$$\sum_{\ell=1}^{L} \mathbb{E}\left[\|\boldsymbol{H}_\ell(\boldsymbol{x}_\ell^\star - \boldsymbol{x}_\ell)\|^2\right] = \sum_{\ell=1}^{L} \mathbb{E}\left[\text{PSURE}_\ell\right], \tag{15}$$

with

$$\text{PSURE}_\ell := \|\boldsymbol{y}_\ell - \boldsymbol{H}_\ell \boldsymbol{x}_\ell\|^2 + 2\sigma_\ell^2 \text{tr}\{\boldsymbol{H}_\ell \boldsymbol{\mathcal{J}}(\boldsymbol{x}_\ell)\} - N\sigma_\ell^2, \tag{16}$$

where σ_l is the noise variance at λ_ℓ, and $\boldsymbol{\mathcal{J}}(\boldsymbol{x}_\ell)$ is the Jacobian matrix of $\boldsymbol{x}_\ell$ with respect to the measurements $\boldsymbol{y}_\ell$ such that the (i, j)-th element of the resulting matrix is defined as

$$[\boldsymbol{\mathcal{J}}(\boldsymbol{x}_\ell)]_{i,j} = \frac{\partial x_{i,\ell}}{\partial y_j}. \tag{17}$$

Note that all jacobians used in the rest of the article are computed with respect to the measurements, similarly to $\boldsymbol{\mathcal{J}}(\boldsymbol{x}_\ell)$. It is important to note that while the PSURE estimates a weighted MSE, being a function of $\boldsymbol{x}$, it is also dependent on the set of parameters used to estimate $\boldsymbol{x}$ i.e. μ_s and μ_λ. As a result, one may search for the values of μ_s and μ_λ that minimize the PSURE, and in that way aim at minimizing the true weighted MSE. The estimation of the PSURE in (16), requires the knowledge of the noise variance and the estimation of the Jacobian matrices $\boldsymbol{\mathcal{J}}(\boldsymbol{x}_\ell)$ at each band. Note that, $\boldsymbol{x}_\ell$ does not have a closed form expression as a function of $\boldsymbol{y}_\ell$, as a result, it is not

possible to have a closed form expression for the Jacobian matrices. One alternative is to estimate these matrices iteratively according to the estimation of x_l in algorithm 1. We derived the Jacobian matrix expression of MUFFIN's reconstruction operator with respect to the measurements. This is done using straightforward applications of the product and the chain rules for derivations. The derivatives are interpreted in the weak sense of distributions whenever the function is not differentiable. Nevertheless, one concern with the Jacobians estimation is that these matrices have enormous sizes. According to the definition in (17), the number of columns of $\mathcal{J}(x_\ell)$ equals the size of the data cube. For a typical reconstruction setting in radio-interferomery, it would be impossible to store or manipulate this matrix. In order to avoid the storage of Jacobian matrices, we use the following stochastic approximation [29] that provides an estimate of the trace required in equation (16):

$$\text{tr}\{\boldsymbol{H}_\ell \mathcal{J}(\boldsymbol{x}_\ell)\} \approx \boldsymbol{e}^t \boldsymbol{H}_\ell \mathcal{J}(\boldsymbol{x}_\ell)\boldsymbol{e} \qquad (18)$$

where $\boldsymbol{e}$ an i.i.d. zero-mean random vector with unit variance. As a result, in order to estimate the last term in equation (16), we can iterate over

$$\mathcal{J}_e(\boldsymbol{x}_\ell) := \mathcal{J}(\boldsymbol{x}_\ell)\boldsymbol{e} \qquad (19)$$

rather than $\mathcal{J}(\boldsymbol{x}_\ell)$, which is the size of $\boldsymbol{x}$. Using this approximation, the PSURE estimation for the MUFFIN algorithm can be updated at each iteration of algorithm 1 according to the steps described in algorithm 2. In the algorithm, we used the notation $\text{Diag}(\boldsymbol{u})$ to refer to the diagonal function which returns a square diagonal matrix with the elements of its input vector $\boldsymbol{u}$ on the main diagonal. The functions $\mathcal{U}(\cdot)$ and $\Pi(\cdot)$ are applied component-wised and defined as:

$$\mathcal{U}(u) := \begin{cases} 0 & \text{if} \quad u \leq 0 \\ 1 & \text{if} \quad u > 0 \end{cases} \qquad (20)$$

and

$$\Pi(u) := \begin{cases} 1 & \text{if} \quad -1 \leq u \leq 1 \\ 0 & \text{if} \quad \text{otherwise.} \end{cases} \qquad (21)$$

IV. Automated parameter selection

The global method for setting one of the regularization parameters is to find the minimizer of the PSURE at the convergence of algorithm 1, using for example a golden section search. We adopt the greedy approach proposed in [23] where the parameter is updated at each iteration. More precisely, a golden section search is performed at each iteration of the algorithm. In this case, the algorithm can be stopped when the value of the estimated weighted MSE, i.e. the PSURE, is smaller than a predefined threshold. This way the number of iterations is automatically set together with the regularization parameter.

In MUFFIN, two regularization parameters have to be set, namely μ_s and μ_λ. We adopt a greedy strategy described in algorithm 3 which decouples the calibration in two steps, and finally runs MUFFIN with the fixed optimal parameters. In the first set of iterations, μ_λ is set to 0 and PSURE is used to automatically tune μ_s. Note that in this case the problem is separable w.r.t. the wavelengths, and each node independently iterates Eqs. (6)-(12) and $\boldsymbol{t}_\ell = \boldsymbol{0}$. This setting reduces the computational load on the master node. However, the master node still receives the PSURE estimate at each band and centralizes the computation of the overall PSURE estimate in order to choose the best regularization parameter. In the second set of iterations, μ_s is fixed to its optimal value (from previous step) and the automatic tuning of μ_λ is triggered using the same scheme as before. In the last step MUFFIN is ran with the optimal fixed parameters allowing to ensure and speed-up the convergence.

Algorithm 2: PSURE update

1 The master computes $\mathcal{J}(\boldsymbol{T})$. Equivalently, it computes:

$$\mathcal{J}_e(\boldsymbol{t}^n) = \mu_\lambda \boldsymbol{W}_\lambda^\dagger \mathcal{J}_e(\boldsymbol{v}^n), \quad \text{for } n = 1 \dots N \qquad (22)$$

and sends the col. ℓ of $\mathcal{J}_e(\boldsymbol{T})$, denoted as $\mathcal{J}_e(\boldsymbol{t}_\ell)$, to node ℓ.

2 Each compute node $\ell = 1 \dots L$ computes sequentially:

$$\mathcal{J}_e(\boldsymbol{x}_\ell) = \mathcal{J}_e(\tilde{\boldsymbol{x}}_\ell) \qquad (23)$$
$$\mathcal{J}_e(\boldsymbol{\nabla}_\ell) = \boldsymbol{H}_\ell^\dagger(\boldsymbol{H}_\ell \mathcal{J}_e(\boldsymbol{x}_\ell) - \boldsymbol{y}_\ell) \qquad (24)$$
$$\mathcal{J}_e(\boldsymbol{s}_\ell) = \mu_s \boldsymbol{W}_s^\dagger \mathcal{J}_e(\boldsymbol{u}_\ell) \qquad (25)$$
$$\mathcal{J}_e(\boldsymbol{m}_\ell) = \boldsymbol{x}_\ell - \tau(\mathcal{J}_e(\boldsymbol{\nabla}_\ell) + \mathcal{J}_e(\boldsymbol{s}_\ell) + \mathcal{J}_e(\boldsymbol{t}_\ell)) \qquad (26)$$
$$\mathcal{J}_e(\tilde{\boldsymbol{x}}_\ell) = \text{Diag}\{\mathcal{U}(\boldsymbol{m}_\ell)\} \mathcal{J}_e(\boldsymbol{m}_\ell) \qquad (27)$$
$$\mathcal{J}_e(\boldsymbol{p}_\ell) = \boldsymbol{u}_\ell + \sigma \mu_s \boldsymbol{W}_s(2\mathcal{J}_e(\tilde{\boldsymbol{x}}_\ell) - \mathcal{J}_e(\boldsymbol{x}_\ell)) \qquad (28)$$
$$\mathcal{J}_e(\boldsymbol{u}_\ell) = \text{Diag}\{\Pi(\boldsymbol{p}_\ell)\} \mathcal{J}_e(\boldsymbol{p}_\ell), \qquad (29)$$

and sends $\mathcal{J}_e(\boldsymbol{x}_\ell)$ and $\mathcal{J}_e(\tilde{\boldsymbol{x}}_\ell)$ to the master.

3 The master computes sequentially, for $n = 1 \dots N$:

$$\mathcal{J}_e(\tilde{\boldsymbol{v}}^n) = \mathcal{J}_e(\boldsymbol{v}^n) + \sigma \mu_\lambda \boldsymbol{W}_\lambda(2\mathcal{J}(\tilde{\boldsymbol{x}}^n) - \mathcal{J}(\boldsymbol{x}^n))$$
$$\mathcal{J}_e(\boldsymbol{v}^n) = \text{Diag}\{\Pi(\tilde{\boldsymbol{v}}^n)\} \mathcal{J}_e(\tilde{\boldsymbol{v}}^n), \qquad (30)$$

and estimates PSURE$= \sum_{\ell=1}^{L} \text{PSURE}_\ell$ eq. (15)–(18)
Return : PSURE

Algorithm 3: Self-tuned MUFFIN

1 $\mu_\lambda = 0$
2 **repeat**
3 $\quad$ MUFFIN update (alg. 1)
4 $\quad$ $\mu_s = \text{argmin}_{\mu_s}$PSURE (alg. 2 + golden section)
5 **until** *stopping criteria* 1;
6 **repeat**
7 $\quad$ MUFFIN update (alg. 1)
8 $\quad$ $\mu_\lambda = \text{argmin}_{\mu_\lambda}$PSURE (alg. 2 + golden section)
9 **until** *stopping criteria* 2;
10 **repeat**
11 $\quad$ MUFFIN update (alg. 1)
12 **until** *stopping criteria* 3;
$\quad$ **Return** : $\boldsymbol{X}$

V. Simulations

We simulate the PSF of MeerKAT[1] observations using the HI-inator package [2] based on the MeqTrees software [30]. We simulated a cube with 100 frequency bands and a size of 256×256 pixels. Figure 1 shows the PSF at the 100-th wavelength band which corresponds to a Fourier coverage produced by a total observation time of 8 hours. The sky corresponds to the radio emission of an HII region in the M31 galaxy. The sky cube for this real sky image is computed using first order power-law spectrum models. The 256×256 map of spectral indices is constructed following the procedure detailed in [31]: for each pixel, the spectral index is a linear combination of an homogeneous Gaussian field and the reference sky image. For $\boldsymbol{W}_s$ and $\boldsymbol{W}_\lambda$, we considered a union of 8 Daubechies wavelet bases for the spatial regularization and a cosine

---

[1]MeerKAT is the South African SKA pathfinder instrument.
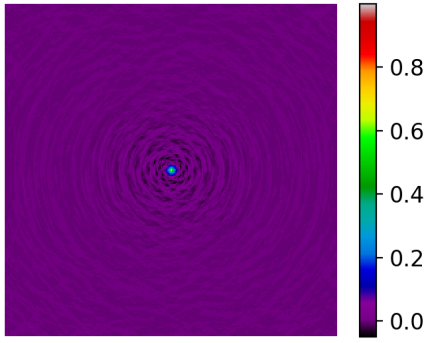[2]https://github.com/SpheMakh/HI-Inator

Fig. 1. Point spread function (PSF) at the 100-th wavelength band showing the central lobe and the ringed side-lobes at larger angular distances.
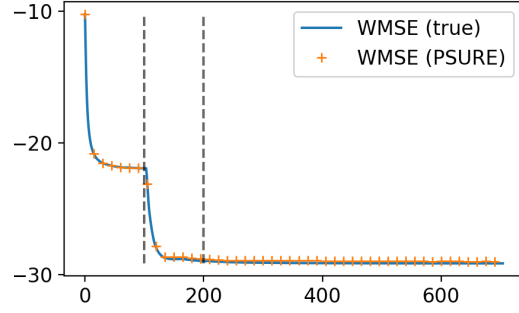

Fig. 2. Variation of the true WMSE and $\widehat{\text{WMSE}}$ (in dB) with the number of iteration obtained with self-tuned MUFFIN. Step 1 (from iter. 0–100): $\mu_\lambda = 0$ and μ_s is updated by the algorithm. Step 2 (from iter. 101–200): $\mu_s = 0.27$ and μ_λ is updated by the algorithm. Step 3 (from iter. 201–700): $\mu_s = 0.27$ and $\mu_\lambda = 1.93$.
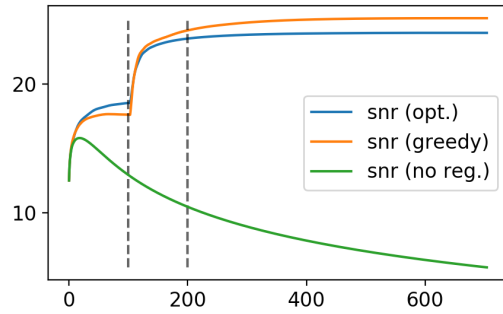

Fig. 3. Variation of the SNR (in dB) as a function of the iterations obtained with self-tuned MUFFIN using the parameters found by the greedy golden section search (greedy) $\mu_s = 0.27$ and $\mu_\lambda = 1.93$, with MUFFIN using the optimal parameters found by the golden section search $\mu_s = 0.43$ and $\mu_\lambda = 2.20$, and with MUFFIN using $\mu_s = 0$ and $\mu_\lambda = 0$.

decomposition for the spectral model. The dirty image was created by convolving the sky image with the PSF, and Gaussian noise was finally added to the dirty image such as the resulting signal to noise (SNR) is equal to 10 dB. The parameters of the algorithm are set to $\sigma = 10$ and $\tau = 10^{-3}$. The simulations were performed using 20 compute nodes[3] which corresponds to 5 spectral bands per node. The algorithm is written using Python, and the Message Passing Interface (MPI) is used to deal with the parallel computing architecture.

We tested self-tuned MUFFIN on the simulated data set. The golden section search intervals for μ_s and μ_λ were set to $[0, \ 2]$ and $[0, \ 3]$ respectively. In the first step, μ_λ is set to 0, and self-tuned MUFFIN automatically found that the optimal μ_s is equal to 0.27. In the second step, μ_s is set to its optimal value 0.27, self-tuned MUFFIN automatically found that the optimal μ_λ is equal to 1.93. Figure 2 shows the variation of the true weighted mean square error (WMSE):

$$\text{WMSE}(\boldsymbol{X}, \boldsymbol{X}^\star) = \frac{1}{LN} \sum_{\ell=1}^{L} \|\boldsymbol{H}_\ell(\boldsymbol{x}_\ell - \boldsymbol{x}_\ell^\star)\|_{\text{F}}^2, \qquad (31)$$

and the WMSE estimated using PSURE, which is the one actually used by self-tuned MUFFIN, and defined as

$$\widehat{\text{WMSE}}(\boldsymbol{X}, \boldsymbol{X}^\star) = \frac{1}{LN} \sum_{\ell=1}^{L} \text{PSURE}_\ell. \qquad (32)$$

Note that the true WMSE and the WMSE estimated using PSURE are reported in dB. The dashed gray lines in the Figure delimit step 1 (from iter. 1–100), steps 2 (from iter. 101–200), and step 3 (from iter. 201–700). More precisely, in the first step $\mu_\lambda = 0$ and μ_s is updated, while in the second step $\mu_s = 0.27$ and μ_λ is updated by the algorithm, and finally in the third step $\mu_s = 0.27$ and $\mu_\lambda = 1.93$. Note that we could have stopped after the first 200 iterations, nevertheless step 3 allows to verify that the algorithm has definitely converged. Furthermore each iteration in step 3 is relatively fast since it does not involve a golden section search. Figure 2 shows that the PSURE estimate is a very good estimate of the WMSE. In addition to this, it shows that the WMSE decreased throughout the iterations, and it had an abrupt decrease when the spectral regularizer was triggered.

These results were compared to the results obtained using optimal fixed parameters. More precisely, we ran each time MUFFIN using a pre-fixed value of μ_s in the first 100 iterations (μ_λ being set to zero), and then by setting μ_λ to some pre-fixed value in the next 600

---

[3]This work was granted access to the HPC and visualization resources of "Centre de Calcul Interactif" hosted by "Université Nice Sophia Antipolis"

iterations. The number of iterations was set to 100 in the first step and 600 in the second step in order to be comparable to the setting used with the self-tuned MUFFIN in the previous simulation. We used the golden section search to find the optimal μ_s that would yield the lowest WMSE at iteration 100. Then the golden section was used to find the optimal μ_λ that would yield the lowest WMSE at iteration 700 given that the first 100 iterations were run with the previously found optimal value for μ_s and $\mu_\lambda = 0$. In the golden section search, μ_s and μ_λ were constrained to the same intervals used previously, $[0, \ 2]$ and $[0, \ 3]$ respectively. This time, we used the real sky to estimate the weighted mean-square error at convergence. The golden section search with MUFFIN found $\mu_s = 0.43$ and $\mu_\lambda = 2.20$. These values are slightly higher than the optimal values found using the greedy search in self-tuned MUFFIN. Figure 3 shows the variation of the SNR defined as:

$$\text{SNR}(\boldsymbol{X}, \boldsymbol{X}^\star) := 10 \log_{10} \left(\frac{\|\boldsymbol{X}^\star\|_2^2}{\|\boldsymbol{X} - \boldsymbol{X}^\star\|_2^2} \right), \qquad (33)$$

obtained with the self-tuned MUFFIN, MUFFIN using the optimal parameters $\mu_s = 0.43$ and $\mu_\lambda = 2.20$, and MUFFIN using $\mu_s = 0$ and $\mu_\lambda = 0$. Figure 3 shows that both self-tuned MUFFIN and MUFFIN with the optimal parameters outperformed MUFFIN when the regularization parameters where both set to zero. It also shows that in the first step, MUFFIN with the optimal μ_s outperformed self-tuned MUFFIN. However, beyond the first step, self-tuned MUFFIN
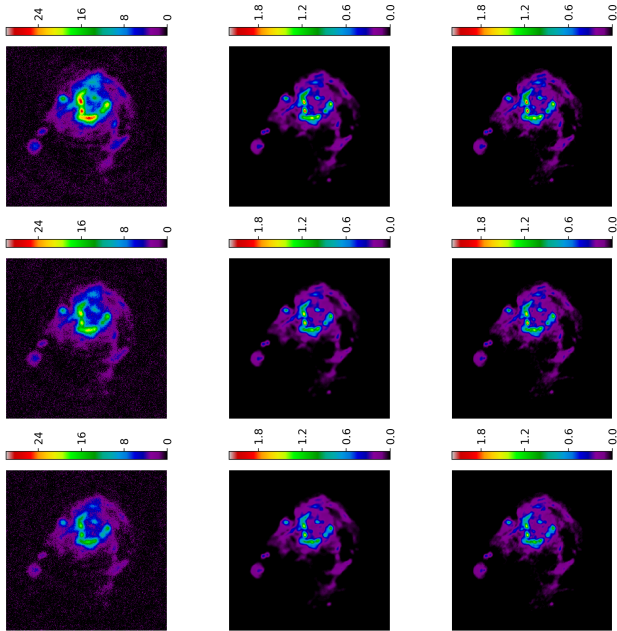
Fig. 4. Left column: M31 Dirty image. Central column: M31 reconstructed images. Right column: M31 sky images. First raw, central raw and bottom raw correspond to the initial, central and last wavelength respectively.

was able to catch up very quickly with the optimally tuned MUFFIN and eventually outperformed it. This is due to the adopted two-step tuning strategy which is computationally efficient but sub-optimal compared to a simultaneous optimization of both parameters. As a result, the optimally tuned MUFFIN was over-regularized w.r.t. μ_s compared to self-tuned MUFFIN. Finally, note that the self-tuned MUFFIN code will be available on the authors Github pages.

VI. CONCLUSION

In conclusion, we proposed an extended version of MUFFIN denoted as self-tuned MUFFIN. Self-tuned MUFFIN is endowed with a parallel and computationally efficient implementation, and it allows to automatically find the optimal regularization parameters.

REFERENCES

[1] A. R. Thompson, J. M. Moran, and G. W. S. Jr, *Interferometry and synthesis in radio astronomy*. John Wiley & Sons, 2008.

[2] P. Dewdney, W. Turner, R. Millenaar, R. McCool, J. Lazio, and T. Cornwell, "Ska1 system baseline design," *Document number SKA-TEL-SKO-DD-001 Revision*, vol. 1, no. 1, 2013.

[3] P. Dewdney, P. Hall, R. Schilizzi, T. Lazio, and L. W. Joseph, "The square kilometre array," *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1482–1496, 2009.

[4] A. J. van der Veen and S. J. Wijnholds, "Signal processing tools for radio astronomy," in *Handbook of Signal Processing Systems*. Springer, 2013, pp. 421–463.

[5] R. Jongerius, S. Wijnholds, R. Nijboer, and H. Corporaal, "An end-to-end computing model for the square kilometre array," *Computer*, vol. 47, no. 9, pp. 48–54, Sept 2014.

[6] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplement Series*, vol. 15, p. 417, 1974.

[7] J. Rich, W. De Blok, T. Cornwell, E. Brinks, F. Walter, I. Bagetakos, and R. Kennicutt Jr, "Multi-scale clean: A comparison of its performance against classical clean on galaxies using things," *The Astronomical Journal*, vol. 136, no. 6, p. 2897, 2008.

[8] A. Onose, R. E. Carrillo, A. Repetti *et al.*, "Scalable splitting algorithms for big-data interferometric imaging in the SKA era," *Monthly Notices of the Royal Astronomical Society*, vol. 462, no. 4, pp. 4314–4335, 2016.

[9] A. Onose, R. E. Carrillo, J. D. McEwen, and Y. Wiaux, "A randomised primal-dual algorithm for distributed radio-interferometric imaging," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1448–1452.

[10] R. E. Carrillo, J. D. McEwen, and Y. Wiaux, "Purify: a new approach to radio-interferometric imaging," *Monthly Notices of the Royal Astronomical Society*, vol. 439, no. 4, pp. 3591–3604, 2014.

[11] A. Ferrari, D. Mary, R. Flamary, and C. Richard, "Distributed image reconstruction for very large arrays in radio astronomy," in *8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE, 2014, pp. 389–392.

[12] J. Deguignet, A. Ferrari, D. Mary, and C. Ferrari, "Distributed multi-frequency image reconstruction for radio-interferometry," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1483–1487.

[13] C. Meillier, P. Bianchi, and W. Hachem, "Two distributed algorithms for the deconvolution of large radio-interferometric multispectral images," in *24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 728–732.

[14] A. Abdulaziz, A. Dabbech, A. Onose, and Y. Wiaux, "A low-rank and joint-sparsity model for hyper-spectral radio-interferometric imaging," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 388–392.

[15] W. C. Karl, "Regularization in image restoration and reconstruction," *Handbook of image video processing*, pp. 183–202, 2005.

[16] V. A. Morozov, "On the solution of functional equations by the method of regularization," in *Soviet Math. Dokl*, vol. 7, no. 1, 1966, pp. 414–417.

[17] T. Regińska, "A regularization parameter in discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 17, no. 3, pp. 740–749, 1996.

[18] P. C. Hansen and D. P. O'Leary, "The use of the l-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1487–1503, 1993.

[19] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[20] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[21] Y. C. Eldar, "Generalized sure for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[22] S. Ramani, Z. Liu, J. Rosen, J.-F. Nielsen, and J. A. Fessler, "Regularization parameter selection for nonlinear iterative image restoration and mri reconstruction using gcv and sure-based methods," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3659–3672, 2012.

[23] R. Giryes, M. Elad, and Y. C. Eldar, "The projected gsure for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2011.

[24] H. Garsden, J. JN. Girard *et al.*, "Lofar sparse image reconstruction," *Astronomy & Astrophysics*, vol. 575, p. A90, 2015.

[25] T. J. Cornwell, K. Golap, and S. Bhatnagar, "The noncoplanar baselines effect in radio interferometry: The w-projection algorithm," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 647–657, 2008.

[26] A. R. Offringa, B. McKinley, N. Hurley-Walker, and F. H. B. et al., "Wsclean: an implementation of a fast, generic wide-field imager for radio astronomy," *Monthly Notices of the Royal Astronomical Society*, vol. 444, no. 1, pp. 606–619, 2014.

[27] L. Condat, "A generic proximal algorithm for convex optimization—application to total variation minimization," *IEEE Signal Processing Letters*, vol. 21, no. 8, pp. 985–989, 2014.

[28] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, Nov. 2011.

[29] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines," *Communications in Statistics-Simulation and Computation*, vol. 19, no. 2, pp. 433–450, 1990.

[30] J. E. Noordam and O. M. Smirnov, "The meqtrees software system and its use for third-generation calibration of radio interferometers," *Astronomy & Astrophysics*, vol. 524, p. A61, 2010.

[31] H. Junklewitz, M. Bell, and T. Enßlin, "A new approach to multi-frequency synthesis in radio interferometry," *Astronomy & Astrophysics*, vol. 581, p. A59, 2015.