Radio Astronomy Image Reconstruction in the Big Data Era

Luke Pratley

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Space and Climate Physics

University College London

September 26, 2019

I, Luke Pratley, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Next generation radio interferometric telescopes pave the way for the future of radio astronomy with extremely wide-fields of view and precision polarimetry not possible at other optical wavelengths, with the required cost of image reconstruction. These instruments will be used to map large scale Galactic and extra-galactic structures at higher resolution and fidelity than ever before. However, radio astronomy has entered the era of big data, limiting the expected sensitivity and fidelity of the instruments due to the large amounts of data. New image reconstruction methods are critical to meet the data requirements needed to obtain new scientific discoveries in radio astronomy. To meet this need, this work takes traditional radio astronomical imaging and introduces new of state-of-the-art image reconstruction frameworks of sparse image reconstruction algorithms. The software package PURIFY, developed in this work, uses convex optimization algorithms (i.e. alternating direction method of multipliers) to solve for the reconstructed image. We design, implement, and apply distributed radio interferometric image reconstruction methods for the message passing interface (MPI), showing that PURIFY scales to big data image reconstruction on computing clusters. We design a distributed wide-field imaging algorithm for non-coplanar arrays, while providing new theoretical insights for wide-field imaging. It is shown that PURIFY's methods provide higher dynamic range than traditional image reconstruction methods, providing a more accurate and detailed sky model for real observations. This sets the stage for state-of-the-art image reconstruction methods to be distributed and applied to next generation interferometric telescopes, where

they can be used to meet big data challenges and to make new scientific discoveries in radio astronomy and astrophysics.

Impact Statement

The main theme of this thesis adapts and develops new methods with data science and convex optimization. Then the thesis demonstrates these methods can be applied to scientific analysis using real data sets. Furthermore, it was shown that these methods provide detailed and accurate reconstruction from interferometric telescopes while being distributed across a computing cluster to cope with the big data era. The developments in this thesis will support imaging with radio telescopes which collect a lot of data and have wide-fields of view. The developments in this thesis are directly related to the Square Kilometre Array (SKA) telescope, an international project to build the worlds largest radio telescope that is has major big data challenges. It is clear that this thesis could make an impact on the big data challenges and new scientific discoveries in radio astronomy and astrophysics.

The distributed image reconstruction algorithms used in this work can be used for image reconstruction challenges outside of astronomy. One prime example includes biomedical imaging, where magnetic resonance imaging (MRI) machines can be used to create detailed images of the human body. The mathematics behind radio interferometric imaging and MRI is extremely transferable, and the developments in this thesis could bring new ideas to medical imaging. However, many areas outside of academia use data science and it is clear that applying these methods to real data sets is valuable to understand. The Segmented Planar Imaging Detector for Electro-optical Reconnaissance (SPIDER) is a newly proposed interferometric optical imaging device, imaging methods from this thesis can be directly applied to this device

for both astronomical and reconnaissance imaging.

The work leading to this thesis has resulted in the research articles

- [1] L. Pratley, J. D. McEwen, M. d'Avezac, R. E. Carrillo, A. Onose, and Y. Wiaux. Robust sparse image reconstruction of radio interferometric observations with purify. MNRAS, 473:1038–1058, January 2018
- [2] L. Pratley, M. Johnston-Hollitt, and J. D. McEwen. A Fast and Exact w-stacking and w-projection Hybrid Algorithm for Wide-field Interferometric Imaging. ApJ, 874:174, April 2019
- [3] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Xiaohao Cai, David Perez-Suarez, Ilektra Christidi, and Roland Guichard. Distributed and parallel sparse convex optimization for radio interferometry with PURIFY. *Astronomy and Computing, submitted, arXiv:1903.04502*, 2019
- [4] L. Pratley, M. Johnston-Hollitt, and J. D. McEwen. w-stacking w-projection hybrid algorithm for wide-field interferometric imaging: implementation details and improvements. *PASA, submitted*, Mar 2019
- [5] L. Pratley and J. D. McEwen. Sparse Image Reconstruction for the SPIDER Optical Interferometric Telescope. *MNRAS, submitted*, March 2019
- [6] L. Pratley and J. D. McEwen. Load balancing for distributed interferometric image reconstruction. *MNRAS, submitted*, March 2019

and the software

- [7] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Rafael Carrillo, Ilektra Christidi, Roland Guichard, David Pérez-Suárez, and Yves Wiaux. PURIFY, February 2019
- [8] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Rafael Carrillo, Ilektra Christidi, Roland Guichard, David Pérez-Suárez, and Yves Wiaux. SOPT, February 2019

Acknowledgements

First of all, I would like to thank my supervisor Jason D. McEwen for providing me with resources and advice during my PhD studies. I thank him for collaboration, helping with writing proposals, reading my papers, checking the equations, and to improving the quality of my work. I would like to thank the research software development group (RSDG) at UCL for teaching me about good software practice and their contributions to PURIFY and SOPT software packages and being good company. Specifically Mayeul d'Avezac for his depth of knowledge of C++, MPI, template programming, and build systems. David Perez-Suarez for help with continuous integration, docker, unit test coverage, and knowledge of useful things that make life easier, Ilektra Christidi for her insights into how to design and structure the code, and Roland Guichard for his developments towards the YAML parser and his availability for reviewing pull requests. I thank Xiaohao Cai for helping me to understanding convex optimization through discussions of papers and for him proof reading my mathematics. I also thank Arwa Dabbech for her teaching me about the spread spectrum effect. I thank Chris Wallis, Peter Taylor, Mathew Price for teaching me about weak lensing and spherical image processing.

I would like to thank Samantha Babister and Philippa Elwell for helping me with arranging booking and travel funds. I thank the astronomy group at the Mullard Space Science Laboratory (MSSL) for being supportive. I thank Daisuke Kawata for being supportive and prioritizing any issues I had during my PhD, especially for supporting me moving to London. I would like to thank UCL for providing me with the Graduate Research and Overseas Research

scholarships (ORS and GRS) used to complete this thesis. I would also like to thank the Science and Technology Funding Counsel (STFC) for the funding to visit International Centre for Radio Astronomy Research (ICRAR) in Western Australia that helped towards new developments in wide-field interferometry. I would like to thank the Royal Astronomical Society (RAS) for funding during this thesis. I acknowledge and thank Mr. William Georgetti for starting the William Georgetti scholarship that provided the funds that helped with living costs and travel expenses during this thesis.

I thank the students at the University of Western Australia (UWA) and Curtin University for including me while I was visiting Perth.

I would like to thank the A-team (Monica, Peter, Ahlam) for the trips to the pub, it kept me kind of sane while living in Guildford. Peter's parents for accommodating me in San Fransisco, and the visit to the train museum in Sacramento. I want to thank Monica for going to see (quality) movies like Skyscraper and Jurassic World 2, and someone who appreciates the dogs of London. Aisha and Ahlam for organizing the ski trip to France and Switzerland, as well as Bryan, Tom, Nabil, and Richard for adding to the company. I also thank Nabil, Aisha, and Ahlam for the road trip to the Netherlands. These experiences were vital to the production of this thesis.

I want to thank my lectures from the Victoria University of Wellington (VUW) for encouraging me and teaching broad concepts and skills from mathematics and physics. Specifically, I would like to thank Christopher Atkin, Hung Pham for teaching me the language of pure mathematics. I would like to thank Ulrich Zülicke, Michele Governale, and Eric Le Ru for teaching me the language of theoretical condensed matter physics. I would like to thank Conor, Felix, and Harry for the team work of getting through undergraduate mathematics and physics.

I would like to thank my Family for their support and encouragement during my travel and studies, especially during my time away from New Zealand. I would like to thank my sisters for always being there for me during

this thesis.

Lastly, I would like to thank Melanie Johnston-Hollitt for introducing me to astronomy and astrophysics while in undergraduate and teaching me radio interferometry, extra-galactic magnetism, high energy astrophysics, and how to write research proposals. The chance to observe at the Australia Telescope Compact Array (ATCA) was incredibly valuable, the rare lightning storm that shut the telescope down due to the power outage and non-stow-able antenna was a true Australian experience. The hotel with the collapsing stairs, the food, and the 25 year old smoked pig in Li Jiang was an experience to never forget. Furthermore, for providing the opportunities, advice, insight, and perspective that greatly improved the scientific quality of this thesis. Ignore those who say the line between amateur and professional has blurred.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The principles of aperture synthesis, using multiple radio antenna to act as a larger telescope, date back as far as the work of [10] in 1947. However, [11] in 1960 first described how aperture synthesis could be used to construct a large scale radio interferometric telescope. Thus, the limit in resolution of single dish radio telescopes could be overcome by using radio interferometric telescopes to improve our ability to observe and therefore understand the radio sky – at the cost of performing computation to solve an ill-posed inverse problem.

Since the 1960s, large radio interferometric arrays have been constructed to observe the sky at high resolution and sensitivity. This includes interferometric arrays such as the Westerbork Synthesis Radio Telescope (WSRT), Very Large Array (VLA), and Australia Telescope Compact Array (ATCA) [12]. These telescopes were used to pioneer Galactic and extra-galactic astronomy at low radio frequencies. In many cases this has provided an understanding of astrophysical processes that is simply not possible at non-radio wavelengths (approximately ranging from 1 meter to 1 millimeter). Two specific examples where radio astronomy and interferometry is a critical for astrophysics are neutral hydrogen (21 cm) spectral line observations and the study of cosmic magnetic fields in the Milky Way and galaxy clusters. Simply put, the 21 cm spectral line is only observed at radio wavelengths, and it has been used to measure the rotation rate of nearby spiral galaxies [13]. For astronomical objects in the distant Universe the 21 cm spectral line becomes

cosmologically red-shifted and is currently being employed in detection experiments for baryon acoustic oscillations and the epoch of re-ionization (EoR) at longer wavelengths [14]. Additionally, radio interferometers probe the magnetized Universe by observing effects such as broadband synchrotron emission (electrons with relativistic energies accelerating in a magnetic field), where radio galaxies and supernova remnants being examples of sources of synchrotron emission. With accurate polarimetry not possible at other wavelengths, radio telescopes can use synchrotron emission to more directly probe magnetized mediums such as the inter-stellar medium (ISM) of the Milky Way and the intra-cluster medium (ICM) of galaxy clusters through the use of the Faraday effect [15].

Next generation radio interferometers are currently coming on-line for astronomers to use. It is expected that these telescopes will provide images of the radio sky at higher resolution and sensitivity than ever before. High fidelity images of the radio sky are required for achieving science goals that can greatly improve our understanding of the Universe in areas of cosmology and astrophysics – with 21 cm and cosmic magnetism science goals only possible at radio wavelengths. However, the large volumes of data, wide-fields of view, and instrumental complexity of these telescopes provide an imaging challenge of unprecedented scale – with the Square Kilometre Array (SKA) providing the most computationally intensive challenge [16]. Big data telescopes, such as the SKA, will not reach the expected fidelity if new distributed image reconstruction algorithms are not developed.

It is clear that there are two major challenges with next generation imaging. The first is to create accurate images of the radio sky for both compact sources and medium to large extended structures. The second challenge is to develop methods of image reconstruction that are computationally efficient enough to scale for large data sets and not require excessive computation. In this thesis we propose that the first challenge can be met with algorithms from convex optimization, where an image is found

that is consistent with the measurements while imposing prior knowledge of the radio sky using wavelet transforms (i.e. that the sky can be efficiently modeled using a particular basis/representation). We also propose that second challenge can be met by distributing the computation and memory used by convex optimization and interferometric imaging algorithms. The developments in this work have been made available using the software packages PURIFY[1] and SOPT[2].

In Chapter 2, we start by introducing compressive sensing and sparse regularization, two closely related frameworks that use convex optimization to perform accurate signal reconstruction on multiple spatial scales. We then link this to interferometric imaging, where degridding and gridding algorithms can be used to efficiently approximate Fourier transforms and predict how close a reconstruction is to the observed interferometric measurements. Chapter 3 introduces basic mathematical tools and algorithms from convex optimization, specifically proximal operators and the alternating direction method of multipliers (ADMM) algorithm, and puts them into the context of radio astronomy. We apply these methods to both simulated and real interferometric observations in Chapter 4, and show that sparse image reconstruction can effectively model structures observed in real interferometric data. This practical application of sparse image reconstruction sets the stage for future algorithm development. The motivates the implementation of the distributed ADMM algorithm, where the measurements, wavelet transforms, proximal operators, and degridding algorithms can be distributed efficiently using the Message Parsing Interface (MPI). The implementation of this algorithm is described and demonstrated in Chapter 5. Then in Chapter 6 we layout mathematical theory of wide-field interferometric imaging with the celestial sphere, including new developments in understanding wide-field imaging with non-coplanar arrays that include the out of plane w-term. We show how radial symmetry can reduce the computation required to model the effect of non-

[1]PURIFY can be found at `https://github.com/astro-informatics/purify`.
[2]SOPT can be found at `https://github.com/astro-informatics/sopt`.

coplanar arrays over wide fields of view when using the w-projection algorithm and combine this with the w-stacking algorithm using developments in distributed interferometric imaging. We use the distributed ADMM algorithm to perform wide-field corrections during image reconstruction to observations of the Vela and Puppis A supernova remnants. In Chapter 7, we describe details and improvements to the wide-field imaging algorithm developments in the previous chapter. In Chapter 8 we describe the implementation of a degridding algorithm that allows even distribution of computational load on a computing cluster when performing wide-field image reconstruction, improving performance by removing computational bottlenecks. In Chapter 9 we then review these methods in the application to a proposed optical interferometric imaging telescope that uses photonic integrated circuits. This thesis is concluded in Chapter 10.

This thesis describes the research, development, and application of computationally distributed interferometric image reconstruction algorithms with the motivation of creating a pathway towards solving imaging challenges from big data radio telescopes – creating an accurate image of the radio sky from next generation radio interferometric telescopes. This will open the door to new scientific discoveries with next generation radio interferometric telescopes.

Chapter 2

Aperture Synthesis and Sparsity

In this chapter we review literature from the areas of radio interferometry, compressive sensing and sparse regularization, and interferometric imaging. These concepts are core to the chapters that follow.

Radio interferometry has been critical for imaging the radio universe at higher resolution and sensitivity than possible with a single radio telescope. However, radio interferometers are limited by the number of possible pairs of antennae in an array, which limits the number of possible measurements made during an observation. Consequently, image reconstruction methods are needed to reconstruct the true sky brightness distribution from the raw data acquired by the telescope, which amounts to solving an ill-posed inverse problem. Traditional methods, which are mostly variations of the Högbom CLEAN algorithm [17], do not exploit modern state-of-the-art image reconstruction techniques.

Next-generation radio interferometers, such as the LOw Frequency ARray (LOFAR; 18), the Murchison Widefield Array (MWA; 19), the Australian Square Kilometre Array Pathfinder (ASKAP; 20), and the Square Kilometer Array (SKA; 21), must meet the challenge of processing and imaging extremely large volumes of data. These experiments have ambitious, high-profile science goals, including detecting the Epoch of Re-ionization (EoR; 14), mapping large scale structure [22], and investigating cosmic magnetic fields [15]. If these science goals are to be realized, state of the art methods in image reconstruction

are needed to process big data and to reconstruct images with high fidelity.

Compressive sensing is a robust mathematical framework for signal reconstruction. The theoretical framework of compressive sensing motivates sparse regularization and convex optimization approaches for solving inverse problems, such as those encountered in radio interferometry. The framework of compressive sensing was first applied to radio interferometry in the study of [23], in the synthesis framework, where it was shown that compressive sensing and sparse regularization approaches can produce higher quality reconstructed images than standard interferometric imaging methods. In [24] the analysis framework was considered and the sparsity averaging reweighted analysis (SARA) algorithm was developed and applied to radio interferometric imaging, demonstrating excellent performance [see also 25]. It has also been shown that the compressive sensing framework can be applied to wide-field of view observations [26] and can correct for directional dependent effects, such as non-coplanar baselines [27, 28]. In [29] state-of-the-art convex optimization algorithms that scale to very large data-sets were developed to solve sparse regularization problems, such as the SARA problem. These algorithms were implemented in the first release of the PURIFY software package [29] for solving radio interferometric imaging problems by sparse regularization. Recently, new algorithms for solving these problems were developed by [30], including proximal alternating direction method of multipliers (ADMM) and primal dual algorithms, paving the way to image the large radio interferometric data-sets that will characterize the SKA era. Alternative compressive sensing approaches have also be applied to aperture synthesis [31, 32, 33] and rotation measure synthesis [34, 35].

2.1 Aperture synthesis and radio interferometry

In aperture synthesis, an array of antennae are collectively used to image the sky at higher resolution than possible with a single dish, hence synthesizing a larger aperture [36]. Each pair of antennae measures a phase and amplitude

of a Fourier component of the brightness distribution across the sky. It is through the measurement of these Fourier components that the sky is effectively imaged. However, due to a limited number of antennae, not all Fourier components can be measured in an observation. An ill-posed inverse problem must be solved to reconstruct the true sky brightness distribution. How this ill-posed inverse problem is solved has a significant impact on the fidelity of the reconstructed image.

Each antenna in an array measures an incoming electric field across its field of view. The electric fields are then cross-correlated between antenna pairs, using a device called a correlator, in-order to calculate the visibility

$$y(\boldsymbol{b} = \boldsymbol{a}_2 - \boldsymbol{a}_1) = \langle \mathcal{E}(\boldsymbol{a}_1, t)\mathcal{E}^*(\boldsymbol{a}_2, t)\rangle_{\Delta t}\,, \tag{2.1}$$

where $\mathcal{E}$ is the electric field amplitude (for polarimetric analysis this is the complex valued electric field vector that includes cross-correlations between the vector components), $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are the spatial positions of the two antenna, t is time, and Δt is the time interval over which the expected value, denoted by $\langle\cdot\rangle$, is taken, which is longer than the time scale of the radio wave observed [36, 12]. The vector difference between the positions of the antennae $\boldsymbol{b} = \boldsymbol{a}_2 - \boldsymbol{a}_1$ is called the baseline.

It is well known that a visibility contains spatial information about the brightness distribution across the sky. While there have been more general measurement equations developed for radio interferometry [37, 38, 39, 40], the van Cittert-Zernike theorem [41] states that the visibility y is related to the sky brightness distribution x, at wavelength λ, by

$$y(\boldsymbol{b}) = \int_{S^2} a(\boldsymbol{\sigma})x(\boldsymbol{\sigma})\mathrm{e}^{-2\pi i\lambda \boldsymbol{b}\cdot\boldsymbol{\sigma}}\,\mathrm{d}\Omega\,, \tag{2.2}$$

where a is the primary beam of the telescope, $\boldsymbol{b}$ is the baseline separating the two antennae, and $\boldsymbol{\sigma}$ denotes a location on the celestial sphere S^2 with area element $\mathrm{d}\Omega$. In principle $\boldsymbol{b} = (u, v, w)$ is a vector in 3 dimensions when the

baselines do not lie on a single plane. The measurement equation is a mapping from the sphere to the 3 dimensional Fourier plane. When the baselines in an array are co-planar (i.e. $w = 0$) and the field of view is narrow (i.e. the sky is approximately flat within the field of view), Eq. 2.2 reduces to a Fourier relation:

$$y(u,v) = \int_{\mathbb{R}^2} a(l,m)x(l,m)\mathrm{e}^{-2\pi i(ul+vm)}\,\mathrm{d}l\mathrm{d}m\,, \tag{2.3}$$

where (l,m) are the coordinates of the plane of the sky, typically with a phase centered on the pointing direction of the telescope, and $\boldsymbol{u} = (u,v)$ are the corresponding Fourier coordinates defined by the baseline: $\boldsymbol{u} = \boldsymbol{b}/\lambda$ (where λ is the observed wavelength). In this context, a visibility measures a Fourier component of the sky brightness distribution in the plane of the sky [36, 12].

The Fourier transform relation of Eq. 2.3 cannot be inverted directly to obtain an accurate estimate of $x(l,m)$ since $y(u,v)$ cannot be measured for all Fourier coordinates. The missing samples of $y(u,v)$ leave Eq. 2.3 as an ill-posed inverse problem, which has an infinite number of possible solutions. To recover a suitable, unique solution, regularization is used to inject prior information regarding the underlying signal.

The most common techniques used to solve for the true sky brightness distribution are CLEAN [*e.g.* 17] and the maximum entropy method (MEM) [*e.g.* 42]. The basic CLEAN algorithm was developed in the 1970's [17]. CLEAN implicitly imposes a sparse prior in a point source (Dirac) basis [43], and is essentially a matching pursuit algorithm [44]. Variations of CLEAN have also been developed for resolved and extended structures, multi-frequency synthesis, and polarized sources [45, 46, 47, 48, 49, 50, 51]. The MEM algorithm regularizes the ill-posed radio interferometric inverse problem through an entropic prior, maximizing an objective function comprised of an entropy term and a data fidelity term (in practice an additional flux constraint is typically imposed in radio interferometric applications of MEM; [42]). In practice, CLEAN often struggles to image diffuse structure, while MEM struggles to resolve point sources. CLEAN, and its variants, are of

widespread use in radio interferometric imaging today, while MEM has not experienced such widespread adoption.

2.2 Sparse regularization for radio interferometric imaging

In its fundamental form, compressive sensing provides a framework for recovering signals from small numbers of measurements and considers the efficient design of the signal measurement process [52, 53, 54, 55]. In radio interferometry, there is little control over the measurement process since the baseline configurations are typically limited by the interferometer (nevertheless, there may be scope for telescope optimization; [27, 28]). The compressive sensing framework, however, motivates a robust method of reconstructing images from the visibilities measured by a telescope through sparse regularization. Sparse regularization exploits the fact that many natural signals—such as astronomical images—are sparse or compressible, *i.e.* for a suitable representation (*e.g.* wavelet basis) most of the coefficients for the ground truth image are zero or close to zero, respectively. In this section we review sparse regularization and how it is applied to radio interferometric imaging.

2.2.1 Sparse regularization

Consider the ill-posed inverse problem of estimating the image $\boldsymbol{x} \in \mathbb{R}^N$ from measurements $\boldsymbol{y} \in \mathbb{C}^M$, where the measurements are acquired by the process $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n}$, where the operator $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$ models the acquisition system and $\boldsymbol{n} \in \mathbb{C}^M$ represents noise. This problem accurately models interferometric imaging, as discussed in more detail in the subsequent sections. For now, we consider sparse regularization approaches to solve this general problem.

Sparse regularization techniques promote sparse solutions when solving ill-posed inverse problems. Typically, natural signals are sparse in a suitable basis (*e.g.* a Dirac, Fourier, or wavelet basis) or, more generally, in a sparsifying

dictionary. The atoms (*cf.* basis functions) of the dictionary [56] can be represented by columns of the operator $\boldsymbol{\Psi} \in \mathbb{C}^{N \times D}$, where N is the number of pixels in the image and D is the number of coefficients of the sparse representation, *i.e.* $\boldsymbol{\alpha} \in \mathbb{C}^D$. The image can then be decomposed into its sparse representation by $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$.

A sparse solution to the inverse problem described above can be promoted by imposing a penalty on the number of non-zero coefficients of the sparse representation $\boldsymbol{\alpha}$ through the ℓ_0-norm, where the ℓ_0-norm $\|\boldsymbol{\alpha}\|_{\ell_0}$ is defined as the number of non-zero coefficients of $\boldsymbol{\alpha}$. In principle, the inverse problem can then be solved by minimising the ℓ_0-norm of the sparse coefficients, subject to a data fidelity constraint:

$$\min_{\boldsymbol{\alpha} \in \mathbb{C}^D} \|\boldsymbol{\alpha}\|_{\ell_0} \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\alpha}\|_{\ell_2} \leq \varepsilon \,. \tag{2.4}$$

Given the solution to this problem, denoted $\boldsymbol{\alpha}^\star$, a recovered image can be synthesised by $\boldsymbol{x}^\star = \boldsymbol{\Psi}\boldsymbol{\alpha}^\star$. The solution to this minimization problem is given by a model that matches the measurements, within error $\varepsilon \in \mathbb{R}^+$, while being constructed from a minimal number of coefficients in the sparse representation. However, this problem cannot be solved in a high dimensional setting because the ℓ_0-norm is non-differentiable and the minimization problem is non-convex: it is considered an NP hard problem [52].

The closest convex relaxation of the ℓ_0 problem is the ℓ_1 problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{C}^D} \|\boldsymbol{\alpha}\|_{\ell_1} \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\alpha}\|_{\ell_2} \leq \varepsilon \,, \tag{2.5}$$

where the ℓ_p-norm is defined by $\|\boldsymbol{r}\|_{\ell_p} = (\sum_i |r_i|^p)^{\frac{1}{p}}$ (hence the ℓ_1-norm is the sum of the absolute value of the components of a vector. The ℓ_2-norm is the usual Euclidean norm). This ℓ_1 minimization problem also promotes sparsity and in some cases exhibits the same solution as the ℓ_0 problem [52, 54]. Furthermore, since the ℓ_1 minimization problem is a convex problem it can be solved using efficient convex optimization algorithms [*e.g.* 57].

The problem defined by Eq. 2.5 is proposed in the standard synthesis setting, where one recovers the coefficients $\boldsymbol{\alpha}$ and synthesises the recovered image by $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$. Alternatively, we can propose the problem in the analysis setting using the adjoint wavelet transform $\boldsymbol{\Psi}^\dagger$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \left\| \boldsymbol{\Psi}^\dagger \boldsymbol{x} \right\|_{\ell_1} \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \varepsilon\,, \tag{2.6}$$

where one recovers the image $\boldsymbol{x}$ directly, while still imposing sparsity in some sparse representation. When the sparsifying operator $\boldsymbol{\Psi}$ is an orthogonal basis the solutions of the synthesis and analysis problems are identical. However, for an overcomplete dictionary the solutions are very different and the analysis setting has been shown to perform very well in practice [*e.g.* 24, 25]. Moreover, reweighted schemes to better approximate the solution of the ℓ_0 problem by solving a sequence of ℓ_1 problems can also be considered [58, 24, 25]. While these approaches can further improve the quality of the reconstructed image we do not consider them further here.

Additionally, sparse regularization problems allow extra constraints to be imposed, such as a real and positive valued image, which is the case for total intensity (Stokes I) radio interferometric observations. However, the positivity and real valued image constraints may be removed for polarimetric imaging, such as linear polarization or the Stokes parameters. Complex valued linear polarization reconstructions of $P = Q + iU$ can also be performed in principle and will be rotationally invariant for rotations in P [51].

2.2.2 Radio interferometric measurement operator

In solving sparse regularization problems, the measurement operator is required to compare how close the reconstructed model matches the measured data. How close the measurement operator matches the true measurement process will have an impact on reconstruction quality.

In the context of radio astronomy, the measurement process is given by Eq. 2.3. We assume co-planar baselines and a small field-field of view here;

we do not consider direction-dependent effects in the measurement operator, although they can nevertheless be modelled in the framework presented [27, 28]. In the compressive sensing setting, the measurements $\boldsymbol{y} \in \mathbb{C}^M$ denote the visibilities $\boldsymbol{y}_i = y(u_i, v_i)$ and the image $\boldsymbol{x} \in \mathbb{R}^N$ denotes the sky brightness distribution $x_p = x(l_p, m_p)$ (for $i = 1, \ldots, M$ and $p = 1, \ldots, N$). The measurement operator $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$ specifies a discrete representation of Eq. 2.3. Ideally, $\boldsymbol{\Phi}$ would represent a direct Fourier transform from the N pixels of the image to the M non-uniformly spaced visibilities. However, this would require $\mathcal{O}(MN)$ computations. Consequently, a direct Fourier transform of the visibilities is not possible for the settings experienced in practice, where a single observation may be comprised of very large numbers of visibilities and high-resolution reconstructed images are required.

Alternatively, it is possible to approximate a direct Fourier transform. One can first interpolate the visibilities onto a regularly spaced grid, which requires order $\mathcal{O}(M)$ operations. Then, it is possible to take advantage of the Fast Fourier Transform (FFT), which requires order $\mathcal{O}(N \log N)$ operations. This approach requires considerably fewer computations than the direct Fourier transform [59], rendering a non-uniform Fourier transform computationally feasible for very large observational data-sets, but it is an approximation. This approximation is the standard approach considered in radio astronomy.

The standard radio interferometric measurement operator $\boldsymbol{\Phi}$ can be written as a series of linear operators:

$$\boldsymbol{\Phi} = \boldsymbol{WGFZSB}, \tag{2.7}$$

where $\boldsymbol{B} \in \mathbb{C}^{N \times N}$ is the primary beam of telescope, $\boldsymbol{S} \in \mathbb{C}^{N \times N}$ is a gridding correction operator that scales the image to correct for the interpolation convolution kernel, $\boldsymbol{Z} \in \mathbb{C}^{\alpha^2 N \times N}$ is a zero-padding operator that provides oversampling by factor α in each dimension of the Fourier domain, $\bar{\boldsymbol{F}} \in \mathbb{C}^{\alpha^2 N \times \alpha^2 N}$ is a FFT operator, $\boldsymbol{G} \in \mathbb{C}^{M \times \alpha^2 N}$ is a convolutional interpolation operator that uses a convolution kernel to interpolate visibilities from Fourier

coefficients on a regular grid to Fourier components in the continuous Fourier plane, and $\boldsymbol{W} \in \mathbb{C}^{M \times M}$ weights the measurements according to their error. Alternatively, it is possible to whiten the measurements by applying weighting $\boldsymbol{W}$ in the ℓ_2-norm directly. A diagram of the process of applying the measurement operator $\boldsymbol{\Phi}$ and its adjoint $\boldsymbol{\Phi}^\dagger$ is shown in Figure 2.1. Since the weights are applied in the measurement operator, it is necessary to also weight the measurements, *i.e.* $\boldsymbol{y} \rightarrow \boldsymbol{W}\boldsymbol{y}$.

2.3 Convolutional gridding and degridding

The fidelity of reconstructed radio interferometric images depends not only on the technique used to solve the resulting inverse problem but also on the accuracy with which the measurement operator models the measurement process. Ideally, the measurement operator would match the measurement process exactly. However, this is not possible due to the computational time required for a direct Fourier transform. We are forced to use a measurement operator that interpolates the visibilities onto and off of a regular grid through the operator $\boldsymbol{G}$, so that we may apply an FFT $\boldsymbol{F}$ to regularly spaced data. Interpolation is typically performed by convolution with a suitable kernel, which then determines the convolutional degridding operator $\boldsymbol{G}$. Several interpolating convolutional kernels have been suggested in the literature; we introduce a subset of these kernels in this section. The choice of convolution kernel affects the quality of the image, through aliasing error, and total computation time, through the support size of the kernel. Ideally, a convolution kernel will have minimal support while maximally suppressing aliasing error, allowing high quality images to be reconstructed in minimal computation time.

2.3.1 Degridding

To replicate the measurement process, Fourier coefficients need to be interpolated off of the FFT grid, *i.e.* they need to be *degridded*. An ideal interpolation that does not change the content of an image is the well-

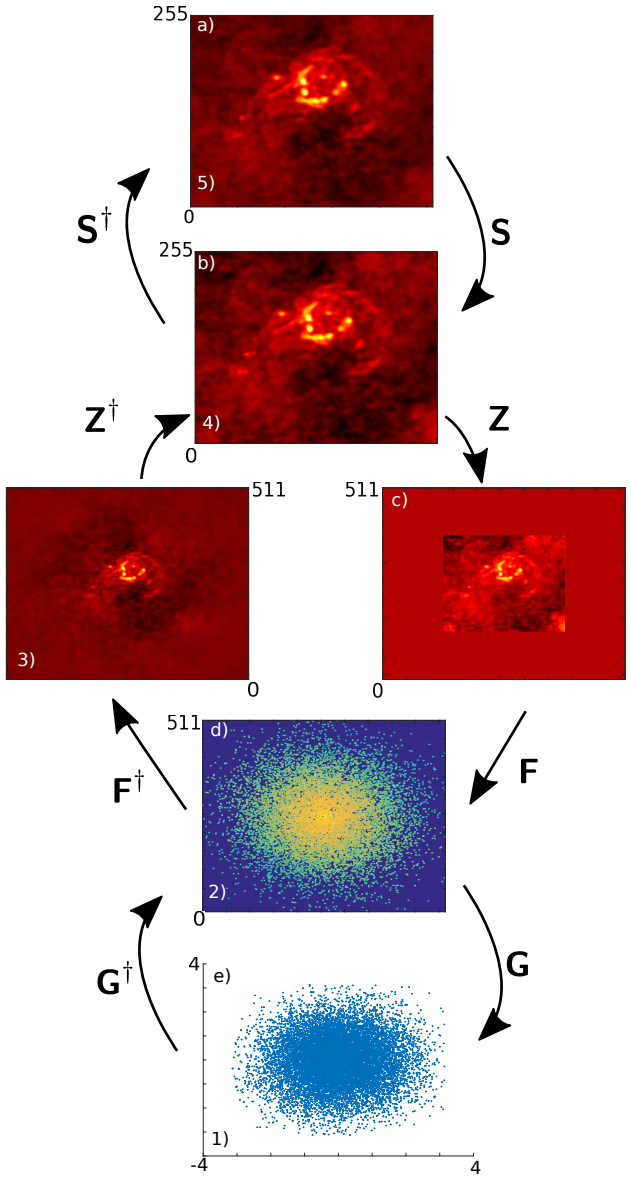

Figure 2.1: Representation of the application of the forward and adjoint measurement operator. The labels a) to e) represent the process of the forward measurement operator, while numbers 1) to 5) represent the process of the adjoint operator. The measurement operator consists of the following steps: a) observed image; b) image is corrected for degridding; c) image is zero-padded to twice the field of view; d) Image is Fourier transformed; e) Fourier coefficients are convolved to continuous points off of the grid. The adjoint measurement operator consists of the following steps: 1) Fourier coefficients in a continuous plane; 2) Fourier coefficients are gridded onto an oversampled grid; 3) image from the transformed Fourier coefficients; 4) image cutout; 5) image corrected for the gridding.

known (Shannon) Sinc interpolation [60, 61], where a continuous band-limited image can be exactly reconstructed from the discrete Nyquist sampled signal. Sinc interpolation can also be considered in the context of interpolating the Fourier domain, which is exact for a space-limited image. In practice, Sinc interpolation in this context can be performed by zero-padding the image domain, which up-samples the Fourier domain via Sinc interpolation.

In the context of degridding, a Sinc interpolation kernel preserves the image and frequency content of the signal when the image has a limited field of view. However, Sinc interpolation is computationally expensive because the Sinc kernel does not have finite local support in harmonic space. A computationally inexpensive method, due to its small support, is to interpolate in the Fourier domain using the nearest neighbour grid point. Nearest neighbour interpolation in the Fourier domain corresponds to convolving with a Box kernel, which corresponds to multiplying with a Sinc function in the image domain. Since the Sinc function has infinite support in the image domain, this introduces artefacts known as aliasing error. The Sinc and nearest-neighbour approaches to interpolating visibilities represent the two extreme cases.

We require kernels with small support in harmonic space (so they are computationally efficient) and small support in image space (to suppress aliasing error). However, the uncertainty principle means there is a fundamental limit on how localised a function can be in both harmonic space and image space. In practice, we seek a trade-off between the two extremes, so that the support of the kernel in harmonic space is not so large as to be computationally expensive, while the support in image space is also well-localised to suppress aliasing error.

Since the interpolation is performed by a convolution, it is necessary to correct for this operation, which can be achieved by multiplication in the image domain with an appropriate window. Furthermore, interpolation accuracy can be increased by zero-padding in the image domain to up-sample the Fourier domain. The process of degridding therefore starts by scaling the image by

the diagonal operator $\boldsymbol{S}$, which preemptively corrects for the interpolation kernel of $\boldsymbol{G}$. This correction is calculated from the reciprocal of the inverse Fourier transform of the interpolation kernel. The image is then zero-padded using the zero-padding operator $\boldsymbol{Z}$ which up-samples harmonic space. An FFT is applied to obtain an up-sampled Fourier grid using the operator $\boldsymbol{F}$. The model measurements are then interpolated off of the grid using the circulant convolution operator $\boldsymbol{G}$. The explicit construction of $\boldsymbol{G}$ is discussed in Section 2.3.4.

2.3.2 Gridding

Most image reconstruction algorithms in radio astronomy require going both backward and forward between the image and measurement domain. Typically, mapping from the measurement domain to the image domain is performed by the adjoint of the measurement operator, since the measurement operator does not have a defined inverse, given by

$$\boldsymbol{\Phi}^\dagger = \boldsymbol{B}^\dagger \boldsymbol{S}^\dagger \boldsymbol{Z}^\dagger \boldsymbol{F}^\dagger \boldsymbol{G}^\dagger \boldsymbol{W}^\dagger . \tag{2.8}$$

Gridding can be considered the reverse process of degridding. Mathematically, the gridding operator is the adjoint of the degridding operator and is performed by application of $\boldsymbol{G}^\dagger$. The full adjoint measurement operator consists of the following operations. First the weighting $\boldsymbol{W}^\dagger = \boldsymbol{W}$ is applied, before the visibilities are interpolated onto an up-sampled Fourier grid using $\boldsymbol{G}^\dagger$. Then an inverse FFT is performed by $\boldsymbol{F}^\dagger$ to produce an image. The image is cropped to the desired field of view using $\boldsymbol{Z}^\dagger$, and the convolution is corrected by $\boldsymbol{S}^\dagger$. Lastly, the adjoint of the primary beam $\boldsymbol{B}^\dagger$ is applied.

A consequence of interpolating the visibilities onto a grid is that the signal is now represented via a Fourier series rather than a Fourier transform. This means the imaged region has periodic boundary conditions. In the case of a radio interferometer, the visibilities can contain information over the entire sky, and the signal may not end at the boundaries of the imaged region. In this

case, the interpolation kernel is used to apodize aliasing error, where structure from outside the boundaries of the imaged region is folded back in [59].

2.3.3 Aliasing error

In the case where the convolution kernel does not sufficiently attenuate the image outside the imaged region, gridding and degridding the signal in the Fourier domain will cause features from outside the imaged region to fold into the image. This effect is known as aliasing error. Two ways to minimise aliasing error are to either image a wider field of view, so that the primary beam of the telescope naturally attenuates structures outside the field of view, or to choose a convolution kernel that attenuates the aliasing error sufficiently.

An ideal convolution kernel would set the image to zero outside the imaged field of view, which would eliminate aliasing error. This can be done with a Sinc convolution kernel, which is computationally expensive. An inexpensive kernel, like a Box kernel, is highly delocalised in the image domain, so does not suppress structure outside the imaged field of view from being folded back in.

To increase image quality and computational performance, a convolution kernel needs a minimal support in harmonic space while attenuating the image outside the field of view. Any attenuation within the imaged field of view is corrected for by $\boldsymbol{S}$, calculated from the Fourier transform of the gridding kernel.

If the gridding kernel apodizes the image domain strongly within the gridded field of view, correcting by $\boldsymbol{S}$ will induce numerical errors [62]. This means that while the suppression due to the gridding kernel can reduce aliasing error, correcting for it has the potential to cause numerical error.

2.3.4 Interpolation kernels

Next, we introduce the convolution kernels used in this work. The width (support) of the gridding kernel J is given in units of grid cells. The oversampling ratio in each dimension is denoted by α.

The degridding matrix is a circulant convolution matrix that interpolates

the measurements off of the discrete Fourier grid onto the continuous Fourier plane. The convolution can be seen as a weighted average of the nearest neighbour grid points. The interpolation kernel determines the weighting of each grid point. Weighting is maximum at the location of the measurement and typically decreases in value when the grid points are further from the measurement location.

In 1-D Fourier space, the degridding matrix $\boldsymbol{G}$ is constructed from a kernel $d(u)$ by [63]

$$\boldsymbol{G}_{i,\{k_i+j\}_K} = d(u_i - (k_i + j)), \tag{2.9}$$

where i is the index of the measurement $\boldsymbol{y}_i$, k_i is the closest integer to visibility coordinate $u_i - J/2$ (in units of pixels), and $j = 1 \dots J$ are the possible non-zero entries of the kernel. The modulo-K function is denoted by $\{\cdot\}_K$, where $K = \alpha\sqrt{N}$ is the dimension of the Fourier grid in 1-D (for notational sake, the 2-D Fourier grid is comprised of $N = \sqrt{N} \times \sqrt{N}$ samples).

The diagonal convolution correction operator $\boldsymbol{S}$ can be calculated in a similar way:

$$\boldsymbol{S}_{i,i} = s\left(\frac{i}{K} - \frac{1}{2}\right), \tag{2.10}$$

where $s(x)$ is the reciprocal of the inverse Fourier transform of $d(u)$. In practice, $\boldsymbol{S}$ can be computed numerically from $\boldsymbol{G}$ or analytically if the inverse Fourier transform of the convolution kernel is tractable.

2.3.4.1 Sinc

The Sinc convolution kernel is ideal when its infinite support is considered. This convolution kernel can be written as [64, 65]

$$d(u) = \left(\frac{u\pi}{N}\right)^{-1} \sin\left(\frac{u\pi}{N}\right). \tag{2.11}$$

The convolution correction is

$$s(x) = \begin{cases} \frac{1}{N}, & \text{if } |x| \leq \frac{N}{2} \\ 0, & \text{otherwise} \end{cases} . \tag{2.12}$$

The advantage of the Sinc convolution kernel is that it corresponds to multiplication by a Box function in the image domain, which bounds the signal at the edges of the imaged region. Consequently, there is close to no aliasing error.

2.3.4.2 Box

The Box function is fast to compute since it is localised in harmonic space, but it does not suppress aliasing error effectively. This kernel has the form [64, 65]:

$$d(u) = \begin{cases} \frac{1}{J}, & \text{if } |u| \leq \frac{J}{2} \\ 0, & \text{otherwise} \end{cases} . \tag{2.13}$$

The Fourier transform of the Box function is the Sinc function, so the convolution correction reads

$$s(x) = \left[\frac{\sin(xJ\pi)}{xJ\pi} \right]^{-1} . \tag{2.14}$$

The Sinc function is not bounded by the edges of the image, and the sidelobes of the Sinc function can cause large aliasing error. This is why the Box function is far from ideal, even if it is fast to compute.

2.3.4.3 Gaussian

The Gaussian kernel is moderately well-localised in both image and Fourier space and takes the form:

$$d(u) = \mathrm{e}^{-\frac{u^2}{2\sigma^2}} . \tag{2.15}$$

The gridding correction is calculated by the Fourier transform and also takes the form of a Gaussian:

$$s(x) = \left[\frac{\pi}{2\sigma^2}\right]^{-1/2} \mathrm{e}^{2x^2\pi^2\sigma^2}. \tag{2.16}$$

An optimal choice for σ as a function of the support size J was found in the work of [63], where it was shown that $\sigma = 0.31J^{0.52}$ works better than using the typical value $\sigma = 1$. In the early years of radio astronomy, in the 1970's, the Gaussian kernel was used for convolutional gridding [12].

2.3.4.4 Prolate spheroidal wavefunction

Prolate spheroidal wavefunctions (PSWFs) do not have an explicit analytic form but there are several ways of characterising them [66, 67, 68, 69]. The most useful way to characterise PSWFs is in terms of energy concentration. PSWFs are bandlimited functions that maximise the energy concentration in a given interval, by finding the function f that maximises the ratio

$$\frac{\int_{-\tau}^{\tau} |f(t)|^2 \mathrm{d}t}{\int_{-\infty}^{\infty} |f(t)|^2 \mathrm{d}t}, \tag{2.17}$$

for an interval $[-\tau, \tau]$. For a convolution kernel, this is an ideal property since we want the convolution kernel to have minimal support in the Fourier domain and to have a maximal amount of energy concentrated over the imaged region in the image domain. This allows one to have minimal support in the Fourier domain while maximally suppressing aliasing error in the image domain.

The standard choice of PSWFs in radio astronomy are a modified version, where more energy is weighted towards the centre of the image, since typically this is the scientific region of interest. The standard choice of weighted PSWFs are described in the work of [70, 62]. The convolution kernel is given by

$$d(u) = |1 - \eta^2(u)|^{\kappa} \psi_{\kappa}(\pi J/2, \eta(u)), \tag{2.18}$$

where $\eta(u) = 2u/J$, κ is a parameter that varies the weighting, and ψ_{κ} is a

zero order PSWF that can be calculated using a rational approximation:

$$\psi_\kappa(\pi J/2, \eta) = \frac{\sum_{k=0}^{n} p_k(\eta^2 - \eta_2^2)^k}{\sum_{k=0}^{d} q_k(\eta^2 - \eta_2^2)^k}, \tag{2.19}$$

where the p_k and q_k polynomial coefficients are specified in [62, 70]. The case of $\kappa = 0$ reduces to an unweighted PSWF. In this work, we use the polynomial coefficients for a support of $J = 6$ and $\kappa = 1$, the standard used in the radio interferometric imaging packages MIRIAD[1] [71] and Astronomical Image Processing System (AIPS; [72])[2]. The correction is provided by [70]:

$$s(x) \approx \frac{1}{\psi_0(\pi J/2, 2x)}. \tag{2.20}$$

2.3.4.5 Kaiser-Bessel

Kaiser-Bessel functions are another useful form of convolution kernel. The zeroth order Kaiser-Bessel function can be expressed as

$$d(u) = \frac{I_0\left(\beta\sqrt{1 - \left(\frac{2u}{J}\right)^2}\right)}{I_0(\beta)}, \tag{2.21}$$

where J is the support, I_0 is the zeroth order modified Bessel function of the first kind, and β determines the spread of the Kaiser-Bessel function [73, 63]. The gridding correction is calculated from the Fourier transform, yielding [73, 63]:

$$s(x) = \left[\frac{\sin\left(\sqrt{\pi^2 x^2 J^2 - \beta^2}\right)}{\sqrt{\pi^2 x^2 J^2 - \beta^2}}\right]^{-1}. \tag{2.22}$$

An optimal choice for β as a function of the support size J was found in the work of [63], where it was shown that for $\beta = 2.34J$ the Kaiser-Bessel kernel performs similarly to the optimal min-max kernel considered in [63]. In [64], it is suggested that the zeroth order Kaiser-Bessel functions perform

[1] `http://www.atnf.csiro.au/computing/software/miriad/`
[2] `http://www.aips.nrao.edu/index.shtml`

similarly to the zeroth-order PSWFs, which is consistent with the results of [73]. Kaiser-Bessel functions, however, have the advantage that they have an analytic expression that can be evaluated easily and accurately. Note that Kaiser-Bessel functions are the standard choice of interpolation kernel in the interferometric imaging package WSCLEAN[3] [50].

2.4 Wide-field Imaging

In the past where the field of view of instruments was relatively small, it was common practice to assume curvature was negligible and proceed with a two dimensional Fourier transform over the uv-plane (using cartesian coordinates). With the arrival of next generation telescopes, such as the LOw Frequency ARray (LOFAR; [74]), Murchison Widefield Array (MWA; [19]), and Hydrogen Epoch of Reionization Array (HERA; [75]), telescopes became non-coplanar arrays with extremely large fields of view. Such instruments are precursors to the low frequency component of the Square Kilometre Array (SKA-LOW), and are already encountering 'big data' challenges. Imaging and correcting for DDEs (with wide-field of view DDEs being the most basic) are among the most computationally intensive and critical challenges that needs to be solved if the SKA is to meet its scientific goals, in areas such as the Epoch of Reionization (EoR) [14] and Cosmic Magnetism [15].

2.4.1 w-stacking, w-projection, and Faceting

Until now, the approach to account for the third Fourier dimension, w, has been to use mathematical approximations to correct for this term and the associated wide-field effects in the measurement equation, reducing the problem back to a two dimensional Fourier transform via the so-called 'w-projection algorithm' [76, 77, 50] and 'w-stacking' algorithm [50]. However, other recent developments have been made that use 'Faceting', where more general wide-field and instrumental DDEs can be approximately modeled by splitting the field of view into smaller regions known as facets [78]. But still,

[3] `https://sourceforge.net/projects/wsclean/`

accurately correcting wide-field effects for non-coplanar baselines remains a computational challenge.

2.4.2 Wide-field measurement equation

The interferometric measurement equation for a wide field radio telescope can be represented by the following integral

$$y(u,v,w') = \int x(l,m)a(l,m)\frac{\mathrm{e}^{-2\pi i w'(\sqrt{1-l^2-m^2}-1)}}{\sqrt{1-l^2-m^2}}\mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,, \quad (2.23)$$

(u,v,w') are the baseline coordinates and (l,m,n) are directional cosines restricted to the unit sphere. In this work, we define $w' = w + \bar{w}$, where $\bar{w}$ is the average value of w-terms, and w is the effective w-component (with zero mean). x is the sky brightness, $n(\boldsymbol{l}) = \sqrt{1-l^2-m^2}$ is a parametrization of the upper hemisphere, and a includes direction dependent effects such as the primary beam and Field of View (FoV). The measurement equation is a mathematical model of the measurement operation that allows one to calculate model measurements y when provided with a sky model x. Having such a measurement equation allows one to find a best fit model of the sky brightness, for a given set of (incomplete) measurements. Many techniques are available for inverting a measurement equation in an attempt to find a best fit model. This includes traditional methods such as CLEAN [17] and Maximum Entropy [79, 42], and state of the art deconvolution methods such as Sparse Regularization algorithms [30, 1, 80]. There are many other variations of the measurement equation, that can include general direction dependent effects and polarization [37, 39, 40]. But, all interferometric measurement equations can be derived from the van Cittert-Zernike theorem [41].

This measurement equation is typically approximated by a non-uniform fast Fourier transform, since it reduces the computational complexity from $\mathcal{O}(MN)$ to $\mathcal{O}(MJ^2 + N\log N)$, where N is the number of pixels M is the number of visibilities, and J is the number of weights to interpolate off the fast Fourier transform (FFT) grid for each axis [63, 12]. This process is traditionally

known as degridding. The version of the measurement equation relevant in this work is represented by the following linear operations

$$\boldsymbol{y} = \boldsymbol{WGCFZSx} \tag{2.24}$$

$\boldsymbol{S}$ represents a gridding correction and correction of baseline independent effects such as $\bar{w}$, $\boldsymbol{Z}$ represents zero padding of the image, $\boldsymbol{F}$ is an FFT, $\boldsymbol{G}$ represents a sparse circulant convolution matrix that interpolates measurements off the grid and the combined $\boldsymbol{GC}$ includes baseline dependent effects such as variations in the primary beam and w-component in the interpolation, and $\boldsymbol{W}$ are weights applied to the measurements. This linear operator represents the application of the measurement equation, so is typically called a measurement operator $\boldsymbol{\Phi} = \boldsymbol{WGCFZS}$ with $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$.

In this case, $\boldsymbol{x}_i = x(\boldsymbol{l}_i)$ and $\boldsymbol{y}_i = y(\boldsymbol{u}_i)$ are discrete vectors in $\mathbb{C}^{N \times 1}$ and $\mathbb{C}^{M \times 1}$ of the sky brightness and visibilities, respectively.

Since the measurement operator is linear it has an adjoint operator $\boldsymbol{\Phi}^\dagger$, which essentially, consists of applying these operators in reverse. Additionally, it is possible to represent these operators in matrix form, however, this is not always efficient or practical.

The dirty map can be calculated by $\boldsymbol{\Phi}^\dagger \boldsymbol{y}$, and the residuals by $\boldsymbol{\Phi}^\dagger \boldsymbol{\Phi x} - \boldsymbol{\Phi}^\dagger \boldsymbol{y}$.

Chapter 3

Convex Optimization Algorithms

In this chapter, we review the mathematical tools and algorithms from convex optimization that are can be used to perform signal reconstruction, through solving least squares minimization problems that contain a penalty regularization term, i.e. sparse regularization. In this thesis, these tools are used to reconstruct images using observations from radio interferometric telescopes.

3.1 Sparse Regularization

Sparse regularization is a method that can estimate the radio sky brightness and isolate a single likely solution. In radio astronomy, the measurements have Gaussian uncertainty, leading to least squares minimization. To impose a penalty against over fitting of the radio sky, we can add a regularization term that penalizes models that over fit the measurements, i.e. a penalty that encourages the model to be sparse in parameters while fitting the radio sky. The Bayesian statistical inference framework can be used to construct the sparse regularization problem, as shown in [166]. From Bayes' theorem, the posterior can then be expressed as

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto \exp\Big[-\|\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2}^2/2\sigma^2\Big]\exp\Big[-\gamma g(\boldsymbol{x})\Big], \tag{3.1}$$

where g is a penalty that imposes structures on $\boldsymbol{x}$ and $\gamma \geq 0$ determines the strength of the penalty. *Maximum a posteriori* (MAP) estimation is found by choosing the estimate of $\boldsymbol{x}$ that will maximize the posterior, which is equivalent to minimizing the negative log posterior, i.e.

$$\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2}^2 / 2\sigma^2 + \gamma g(\boldsymbol{x}) \right\}. \tag{3.2}$$

This minimization problem is known as regularized least squares, with the regularization term being $g(\boldsymbol{x})$. In many cases $g(\boldsymbol{x})$ is chosen to penalize the number of parameters that determine $\boldsymbol{x}$ and reduce over fitting; moreover, it can also be used to enforce other properties for $\boldsymbol{x}$ like smoothness. Furthermore, it is possible to add indicator functions as a prior that can restrict our solution to be real or positive valued, as is done in the constrained problem below. MAP estimation can be solved efficiently using the Forward Backward Splitting algorithm [e.g. 165].

An issue of using MAP estimation to perform sparse regularization is choosing a proper regularization parameter γ (although there are ways to address this; 81). The choice of γ, however, can be avoided after moving from the unconstrained problem in MAP estimation to the constrained problem

$$\arg\min_{\boldsymbol{x}} g(\boldsymbol{x}) + \iota_{\mathcal{B}^{\varepsilon}(\boldsymbol{y})}(\boldsymbol{\Phi}\boldsymbol{x}) + \iota_{\mathbb{R}_+^N}(\boldsymbol{x}), \tag{3.3}$$

where ι is the indicator function that restricts $\boldsymbol{\Phi}\boldsymbol{x}$ to the set

$$\mathcal{B}^{\varepsilon}(\boldsymbol{y}) = \{\boldsymbol{q} : \|\boldsymbol{y} - \boldsymbol{q}\|_{\ell_2} \leq \varepsilon\}, \tag{3.4}$$

ε is the error tolerance, and $\iota_{\mathbb{R}_+^N}$ restricts the solution to be positive.

One main advantage of the constrained objective function, compared to the unconstrained form (3.2), is that the parameter ε can be estimated from $\boldsymbol{y}$ [1], and therefore could be easier to set than assign a pertinent value for γ in (3.2). Note, in practice, that the weights in $\boldsymbol{y}$ might be relative with no

flux scale attached, or are not reliable, which will cause a difficulty for the constrained problem. On the other hand, progress is being made on methods that can estimate values of γ for the unconstrained problem. It is also worth noticing that these two forms, (3.2) and (3.3), have close relationship and, in some sense, are equivalent to each other after assigning proper values for ε and γ. The majority of this work is focused on the constrained problem (3.3) and we assume ε can be estimated.

3.1.1 Analysis and Synthesis

In the following we focus on using the ℓ_1-norm for the function g and require our solution to have positive real values, where the ℓ_p-norm is defined by $\|\boldsymbol{x}\|_{\ell_p} = (\sum_i x_i^p)^{1/p}$ for $p > 0$. Additionally, we need to choose the representation of our signal to efficiently model the sky. This is done using a linear transform $\boldsymbol{\Psi}$, with the convention that $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ represents the coefficients of $\boldsymbol{x}$ under the basis or dictionary $\boldsymbol{\Psi}$. A wavelet transform is convenient because it can efficiently represent structures as a function of scale and position. Moreover, $\boldsymbol{\Psi}$ is not restricted to be a basis, but can be an over-complete frame containing a collection of transforms. In this work, we use a collection of wavelet transforms to model the radio sky, as done in [24, 30, 1, 2].

The synthesis forms of the objective function for the unconstrained and constrained problems are respectively

$$\boldsymbol{x}^* = \boldsymbol{\Psi} \times \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\alpha}\|_{\ell_2}^2 / 2\sigma^2 + \gamma \|\boldsymbol{\alpha}\|_{\ell_1}, \quad \text{s.t.} \quad \boldsymbol{\Psi}\boldsymbol{\alpha} \in \mathbb{R}_+ \right\}, \tag{3.5}$$

$$\boldsymbol{x}^* = \boldsymbol{\Psi} \times \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left\{ \|\boldsymbol{\alpha}\|_{\ell_1}, \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\alpha}\|_{\ell_2} \leq \varepsilon \quad \& \quad \boldsymbol{\Psi}\boldsymbol{\alpha} \in \mathbb{R}_+ \right\}. \tag{3.6}$$

The analysis forms of the objective function for the unconstrained and constrained problems are respectively

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2}^2 / 2\sigma^2 + \gamma \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_{\ell_1}, \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}_+ \right\}, \tag{3.7}$$

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_{\ell_1}, \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \varepsilon \quad \& \quad \boldsymbol{x} \in \mathbb{R}_+ \right\}. \tag{3.8}$$

In the synthesis form we solve for the wavelet coefficients $\boldsymbol{\alpha}$ directly and in the analysis form we solve for the pixel coefficients $\boldsymbol{x}$ directly. In practice they provide different results depending on the problem to be solved [82]. We follow the work of [24], which uses an over-complete frame in the analysis setting and is typically found to provide better reconstruction quality than the synthesis setting. The objective function can be solved multiple-times after reweighting the ℓ_1-norm in the analysis setting with an over-complete frame, using what is called Sparsity Averaging Reweighted Analysis (SARA) [24].

Recent works have considered polarimetric [83, 84, 85] and spectral sparse image reconstruction [86, 87]. The works of [84, 85] show that where polarimetric images are reconstructed as a four component vector of Stokes parameters I (total intensity), Q and U (linear polarizations), and V (circular polarization), it is possible to enforce the physical constraint that $I \geq \sqrt{Q^2+U^2+V^2}$. Such a constraint enforces physical structures on both total intensity and polarized intensity, increasing the physicality of the reconstructions. Additionally, it is possible to impose non-parametric structures on spectra, such as spectral smoothness or sparsity, increasing the fidelity across the spectrum.

The challenge in finding the global solution of these objective functions, (3.5)–(3.8), is that they are non-differentiable (because of the non-differentiability of the ℓ_1 regularization term) and are not always continuous (because they contain constraints). However, these objective functions have the property that they are convex and lower semi-continuous (l.s.c.). In the following sections, we introduce proximal operators, which provide tools and algorithms that can be used to find solutions to the above convex minimization problems.

3.2 Proximal Operators

In the previous section we introduced the convex objective functions (3.2) and (3.3), which need to be minimized to obtain a likely solution of the radio

sky. When the problem is poised as minimization of a convex cost function, there are many convex optimization tools – proximal operators and proximal algorithms among them – on hand to solve it and find a global minimizer. In the following, we briefly recall some concepts and operators of convex functions and convex sets, which are useful when discussing solutions to convex inverse problems. A more detailed introduction to these concepts can be found in [88, 57, 89], and have been discussed in the context of radio interferometric imaging previously [29, 24, 90, 30, 1, 2]. In this section, we review the basic mathematics of proximal operators, and introduce the closed-form solution of proximal operators used in this work.

Let X be a vector space and $\Gamma_0(X)$ be the class of proper, l.s.c. convex functions that map from X to $(-\infty, +\infty]$. A function f is defined as l.s.c if $f(\boldsymbol{x}) \leq \liminf_{\boldsymbol{a}\to\boldsymbol{x}} f(\boldsymbol{a})$ [91]. Intuitively, this means that $f(\boldsymbol{x})$ is bounded to be below the limit point at $\boldsymbol{x}$. For example, the ceiling function is l.s.c. A function h is convex when

$$h(\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2) \leq \alpha h(\boldsymbol{x}_1) + (1-\alpha)h(\boldsymbol{x}_2), \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in X, \forall \alpha \in [0,1], \tag{3.9}$$

which is then true for $\forall h \in \Gamma_0(X)$.

The subdifferential of h at $\boldsymbol{x} \in X$, denoted by $\partial h(\boldsymbol{x})$, is defined as

$$\partial h(\boldsymbol{x}) := \{\boldsymbol{u} \in X : h(\boldsymbol{z}) \geq h(\boldsymbol{x}) + \boldsymbol{u}^\top(\boldsymbol{z} - \boldsymbol{x}), \forall \boldsymbol{z} \in X\}. \tag{3.10}$$

When h is differentiable, the subdifferential is a singleton containing the gradient ∇h. If $\boldsymbol{0} \in \partial h(\boldsymbol{x})$ then $\boldsymbol{x}$ belongs to the set of global minimizers of h [57]. The convex conjugate of $h \in \Gamma_0(X)$, denoted by $h^* \in \Gamma_0(X)$, is defined as

$$h^*(\boldsymbol{m}) := \sup_{\boldsymbol{x}\in X} \left(\boldsymbol{m}^\top \boldsymbol{x} - h(\boldsymbol{x})\right). \tag{3.11}$$

It follows that $\boldsymbol{m}$ will lie in the sub-differential of h at all points $\boldsymbol{x}$ that attain the supremum, as described in [89]. The (convex) conjugate can be used to map a convex objective function from the primal representation to the dual

representation, which is useful if both representations have the same optimal values when strong duality holds [88, 57, 92, 89].

For $\forall h \in \Gamma_0(X)$ and any constant $\lambda > 0$, the proximity operator of function λh at $\boldsymbol{v} \in X$, which is denoted by $\text{prox}_{\lambda h}(\boldsymbol{v})$ and maps between $X \to X$, is defined as the solution of the minimization problem

$$\text{prox}_{\lambda h}(\boldsymbol{v}) = \underset{\boldsymbol{x} \in X}{\text{argmin}} \left(\lambda h(\boldsymbol{x}) + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{v}\|_{\ell_2}^2 \right). \tag{3.12}$$

We see that $\text{prox}_{\lambda h}(\boldsymbol{v})$ is a point that is chosen in X by compromising between minimizing h and being close to $\boldsymbol{v}$, where this compromise is weighted by λ. For large λ more movement is taken towards minimizing h, and for small λ less movement is taken from $\boldsymbol{v}$. The proximal operator in (3.12) involves solving a minimization problem, which sometimes has a simple analytic form and sometimes not. When there is no analytic form it needs to be solved or estimated iteratively. It can be shown that the proximal operator is closely related to the subdifferential (3.10), being equivalent to the inverse operation $(I + \lambda \partial h)^{-1}(\boldsymbol{v})$ [57].

When applied to a convex function, the proximal operator can be used to find a global minimizer through the recursive iteration. This is because the proximal operator is what is known as firmly non-expansive. More importantly it is a contraction, meaning repeated application of the proximal operator

$$\boldsymbol{x}^{k+1} = \text{prox}_{\lambda h}(\boldsymbol{x}^k) \tag{3.13}$$

will converge to a fixed point that minimizes λh and therefore also minimizes h; that is, $\boldsymbol{x} = \text{prox}_{\lambda h}(\boldsymbol{x})$ if and only if $\boldsymbol{x}$ minimizes h [88, 57].

The proximal operator has plenty of useful properties. For example, the proximal operator for the translation, the semi-orthogonal linear transform and the convex conjugation are

$$\text{prox}_{\lambda h(\cdot + \boldsymbol{a})}(\boldsymbol{x}) = \text{prox}_{\lambda h}(\boldsymbol{x} + \boldsymbol{a}) - \boldsymbol{a}, \quad \forall \boldsymbol{a} \in X, \tag{3.14}$$

$$\text{prox}_{\lambda h(\boldsymbol{L}(\cdot))}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{L}^\dagger \left(\text{prox}_{\lambda h}(\boldsymbol{L}\boldsymbol{x}) - \boldsymbol{L}\boldsymbol{x}\right), \quad \boldsymbol{L}\boldsymbol{L}^\dagger = \boldsymbol{I} \tag{3.15}$$

and

$$\text{prox}_{\lambda h^*}(\boldsymbol{x}) = \boldsymbol{x} - \lambda \text{prox}_{\lambda^{-1} h}(\boldsymbol{x}/\lambda), \tag{3.16}$$

respectively. The property for convex conjugation is also known as Moreau decomposition. Refer to [88, 57] and references therein for other properties and more details. Typically, it is difficult to obtain a closed form of the proximal operator for two functions $f + g$. The algorithms in the following section split the algorithm into solving for $f + g$ given the proximal operator of f and g separately, and are typically called proximal splitting algorithms. First, we introduce closed forms of proximal operators that are used in radio interferometric imaging (but more examples are listed in [88, 57]).

In this work, we focus on ℓ_1 regularized least squares, i.e., using the ℓ_1 prior for g in the constrained problem (3.3). We need to minimize an ℓ_1-norm with the condition that the solution lies within an ℓ_2-ball with the size of our error ε, while being real or positive valued. This can be mathematically stated as

$$\boldsymbol{x}^\star = \underset{\boldsymbol{x}}{\text{argmin}} \left\{ \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_{\ell_1} + \iota_{\mathcal{C}}(\boldsymbol{x}) + \iota_{\mathcal{B}^\varepsilon_{\ell_2}(\boldsymbol{y})}(\boldsymbol{\Phi}\boldsymbol{x}) \right\}, \tag{3.17}$$

where we normally take $\mathcal{C} = \mathbb{R}^N_+$, and the ℓ_2-ball $\mathcal{B}^\varepsilon_{\ell_2}$ to be the closed ball of radius ε, and $\iota_{\mathcal{C}}(\boldsymbol{x})$ is the indicator function for $\boldsymbol{x}$ being in $\mathcal{C}$ which will be detailed below. We now present the proximal operators needed to minimize this objective function.

3.2.1 Indicator Function

Fix any nonempty closed convex set $\mathcal{C}$, on which we define the indicator function as

$$\iota_{\mathcal{C}}(\boldsymbol{x}) := \begin{cases} 0, & \boldsymbol{x} \in \mathcal{C}, \\ +\infty, & \boldsymbol{x} \notin \mathcal{C}. \end{cases} \tag{3.18}$$

We recall the projection operator $\mathcal{P}_{\mathcal{C}}$, i.e.

$$\mathcal{P}_{\mathcal{C}}(\boldsymbol{x}) := \underset{\boldsymbol{v}\in\mathcal{C}}{\operatorname{argmin}} \|\boldsymbol{x}-\boldsymbol{v}\|_{\ell_2}^2. \tag{3.19}$$

If $\mathcal{C} \subseteq X$, then we have $\iota_{\mathcal{C}} \in \Gamma_0(X)$ and

$$\mathcal{P}_{\mathcal{C}}(\boldsymbol{x}) = \underset{\boldsymbol{v}\in\mathcal{C}}{\operatorname{argmin}} \|\boldsymbol{x}-\boldsymbol{v}\|_{\ell_2}^2 = \underset{\boldsymbol{v}\in X}{\operatorname{argmin}} \left\{\iota_{\mathcal{C}}(\boldsymbol{v}) + \|\boldsymbol{x}-\boldsymbol{v}\|_{\ell_2}^2\right\} = \operatorname{prox}_{\iota_{\mathcal{C}}}(\boldsymbol{x}), \tag{3.20}$$

Therefore, the proximal operator can be regarded as an extension of the projection operator [57]. The indicator function is useful for e.g. restricting a cost function to a set of solutions, or enforcing real or positive values on the solutions as assumptions for an image of the radio sky.

3.2.2 Fidelity Constraint

Let the closed ℓ_2-ball $\mathcal{B}_{\ell_2}^{\varepsilon}$ centered at $\boldsymbol{z} \in X$ with radius ε be the set

$$\mathcal{B}_{\ell_2}^{\varepsilon}(\boldsymbol{z}) := \{\boldsymbol{v} \in X : \|\boldsymbol{z}-\boldsymbol{v}\|_{\ell_2} \leq \varepsilon\}. \tag{3.21}$$

Then the proximal operator of an ℓ_2-ball centered at zero reads

$$\begin{aligned}\operatorname{prox}_{\mathcal{B}_{\ell_2}^{\varepsilon}(0)}(\boldsymbol{x}) &= \underset{\boldsymbol{v}\in X}{\operatorname{argmin}} \left\{\iota_{\mathcal{B}_{\ell_2}^{\varepsilon}(0)}(\boldsymbol{v}) + \frac{1}{2}\|\boldsymbol{v}-\boldsymbol{x}\|_{\ell_2}^2\right\} \\ &= \begin{cases} \boldsymbol{x}, & \boldsymbol{x} \in \mathcal{B}_{\ell_2}^{\varepsilon}(0), \\ \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\varepsilon, & \boldsymbol{x} \notin \mathcal{B}_{\ell_2}^{\varepsilon}(0). \end{cases}\end{aligned} \tag{3.22}$$

In detail, when $\boldsymbol{x} \in \mathcal{B}_{\ell_2}^{\varepsilon}(0)$, we have $\operatorname{prox}_{\mathcal{B}_{\ell_2}^{\varepsilon}(0)}(\boldsymbol{x}) = \boldsymbol{x}$ straightforwardly; when $\boldsymbol{x} \notin \mathcal{B}_{\ell_2}^{\varepsilon}(0)$, computing $\operatorname{prox}_{\mathcal{B}_{\ell_2}^{\varepsilon}(0)}(\boldsymbol{x})$ is to find a $\boldsymbol{v} \in \mathcal{B}_{\ell_2}^{\varepsilon}(\boldsymbol{0})$ such that it minimizes $\|\boldsymbol{v}-\boldsymbol{x}\|_{\ell_2}^2$. From the triangle inequality, we require that $\boldsymbol{v}$ is parallel to $\boldsymbol{x}$ for it to be a minimizer. It follows that we can scale $\boldsymbol{x}$ into $\mathcal{B}_{\ell_2}^{\varepsilon}(0)$ to obtain the explicit representation of $\operatorname{prox}_{\mathcal{B}_{\ell_2}^{\varepsilon}(0)}(\boldsymbol{x})$ shown in (3.22). Using the translation property of the proximal operator in (3.14), we can find the

proximal operator of an ℓ_2-ball centered at $\boldsymbol{z}$, i.e.,

$$\mathcal{P}_{\mathcal{B}}^{\varepsilon}(\boldsymbol{z}) := \text{prox}_{\mathcal{B}_{\ell_2}^{\varepsilon}(\boldsymbol{z})}(\boldsymbol{x}) = \begin{cases} \boldsymbol{x}, & \boldsymbol{x}-\boldsymbol{z} \in \mathcal{B}_{\ell_2}^{\varepsilon}(\boldsymbol{0}), \\ \frac{\boldsymbol{x}-\boldsymbol{z}}{\|\boldsymbol{x}-\boldsymbol{z}\|}\varepsilon + \boldsymbol{z}, & \boldsymbol{x}-\boldsymbol{z} \notin \mathcal{B}_{\ell_2}^{\varepsilon}(\boldsymbol{0}). \end{cases} \tag{3.23}$$

3.2.3 Promoting Sparsity

The ℓ_1-norm is the sum of the absolute values of all components of a vector. Since it is convex and can promote sparsity when serving as a prior distribution or regularization, it is widely used in signal/image processing and has been shown highly effective in radio astronomy.

The proximal operator of the ℓ_1-norm reads

$$\begin{aligned} \text{prox}_{\lambda\|\cdot\|_{\ell_1}}(\boldsymbol{x}) &= \underset{\boldsymbol{v}\in X}{\text{argmin}} \left\{ \lambda\|\boldsymbol{v}\|_{\ell_1} + \frac{1}{2}\|\boldsymbol{v}-\boldsymbol{x}\|_{\ell_2}^2 \right\} \\ &= \mathcal{S}_{\lambda}(\boldsymbol{x}). \end{aligned} \tag{3.24}$$

Here $\mathcal{S}_{\lambda}(\boldsymbol{x})$ is the soft thresholding of vector $\boldsymbol{x} = (x_1, \cdots, x_i, \cdots)$, which is defined as

$$\mathcal{S}_{\lambda}(\boldsymbol{x}) = (\mathcal{S}_{\lambda}(x_1), \cdots, \mathcal{S}_{\lambda}(x_i), \cdots), \tag{3.25}$$

where

$$\mathcal{S}_{\lambda}(x_i) = \begin{cases} 0, & |x_i| \leq \lambda, \\ \frac{x_i(|x_i|-\lambda)}{|x_i|}, & |x_i| > \lambda. \end{cases} \tag{3.26}$$

An intuitive explanation can be found analyzing the proximal operator minimization problem in v for positive and negative $\pm|x|$ separately.

We start with $x = |x|$ and aim minimize the polynomial $\lambda|v| + \frac{1}{2}(v-|x|)^2 = v^2/2 + (\lambda|v| - |x|v) + x^2/2$. We have $x^2 + v^2$ is positive, it follows the polynomial will reach a minimum when $|x|v \geq \lambda|v|$ otherwise we obtain the solution $v = 0$. With $\lambda > 0$, the inequality $|x|v \geq \lambda|v|$ is only true if $v = |v|$ or $v = 0$. This inequality simplifies to $|x| \geq \lambda$. We can then find the solution to minimize the polynomial $v^2/2 + (\lambda - |x|)v + x^2/2$ through differentiation. We are left with the solution $v = |x| - \lambda$ when $|x| \geq \lambda$ and $v = 0$ otherwise.

We repeat the same argument for negative x and choose $x = -|x|$ to read $\lambda|v| + \frac{1}{2}(v + |x|)^2 = v^2/2 + (\lambda|v| + |x|v) + x^2/2$. It follows the polynomial will reach a minimum when $-|x|v \geq \lambda|v|$, this is only true if $v = -|v|$ or $v = 0$. Again, this inequality simplifies to $|x| \geq \lambda$. We can then find the solution to minimize the polynomial $v^2/2 - (-\lambda + |x|)v + x^2/2$, which is $v = -(|x| - \lambda)$ when $|x| \geq \lambda$ and $v = 0$ otherwise. Combining both solutions for $x = \pm|x|$, we obtain the soft thresholding formula.

3.2.4 Summary

This section has provided an introduction to proximal operators and examples of their closed-form solutions that are commonly used for interferometric imaging of real observations [30, 1, 80]. Proximal operators are especially powerful when the objective function is non-smooth, which is often required to enforce physicality on the solution. One important example in polarimetric imaging is to use a proximal operator that will project onto the set of solutions that contains $I \geq \sqrt{U^2 + Q^2 + V^2}$ [84, 85].

We have provided proximal operators for a function f, but we often need to minimize an addition of functions, e.g. $f + g$. In the next section, we show how to solve for the minimizer of $f + g$ when the proximal operators of f and g are known separately.

3.3 Proximal Algorithms

Let $X = \mathbb{R}^N, f \in \Gamma_0(X)$, $g \in \Gamma_0(X)$, using the tools from the previous section, we can solve the convex optimization problem with the general form

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x}) \,. \tag{3.27}$$

Here, for simplicity, we assume each of the minimization problems, like (3.27) considered in this work has a global minimizer. If the proximal operator of $f + g$ was known or could be computed easily, we could recursively iterate the proximal operator to find a solution to (3.27). However, we often only

know the proximal operator for f and g separately. In the following, we briefly introduce a few algorithms among the proximal algorithm category which can address this kind of minimization problem. Moreover, these algorithms can be adapted to be distributed across computing clusters [57, 93, 89, 30]; as done in later chapters of this work.

3.3.1 Forward-Backward Splitting

In the case that f is differentiable, problem (3.27) can be solved using the Forward-Backward splitting algorithm. Starting with a proper initialization, $\forall \lambda \in (0, +\infty)$, the iterative scheme can be represented as

$$\boldsymbol{x}^{(k+1)} = \text{prox}_{\lambda g}(\boldsymbol{x}^{(k)} - \lambda \nabla f(\boldsymbol{x}^{(k)})), \tag{3.28}$$

which includes a forward gradient step (explicit) regarding function f and a backward step (implicit and involves solving a proximal operator) with respect to g. Refer to [94, 95, 90, 33, 89, 30] and references therein for more details and the variants of the Forward-Backward splitting algorithm.

As an example, we see that formula (3.28) can be directly used to solve the unconstrained problem (3.2) so as to obtain an MAP estimator of the sky in radio astronomy. When g is the ℓ_1-norm, this algorithm becomes the Iterative Shrinkage-Thresholding Algorithm (ISTA), where it is possible to obtain accelerated convergence by using Fast ISTA (FISTA) [95], which is detailed in Algorithm 1.

Algorithm 1 FISTA

1: **given** $\boldsymbol{x}^{(0)} \in \mathbb{R}^N, \lambda > 0, \theta_0 = 1, \hat{\boldsymbol{x}}^{(0)} = \boldsymbol{x}^{(0)}$
2: **repeat for** $k = 0, \ldots$
3: $\quad \boldsymbol{x}^{(k+1)} = \text{prox}_{\lambda g}(\hat{\boldsymbol{x}}^{(k)} - \lambda \nabla f(\hat{\boldsymbol{x}}^{(k)}))$
4: $\quad \theta_{k+1} = \frac{1+\sqrt{1+4\theta_k^2}}{2}$
5: $\quad \hat{\boldsymbol{x}}^{(k+1)} = \boldsymbol{x}^{(k+1)} + \frac{\theta_k - 1}{\theta_{k+1}}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})$
6: **until convergence**

The Forward-Backward algorithm is often simpler to compute than the algorithms that follow, which is an advantage of solving the unconstrained

problem over the constrained problem. However, there are many cases where f is not differentiable; for example when it represents an indicator function. Note that the Forward-Backward algorithm cannot be used to solve the constrained problem (3.17) directly, due to the non-differentiable indicator function.

3.3.2 Douglas-Rachford Splitting

When both f and g in (3.27) are non-differentiable, the Douglas-Rachford splitting algorithm can be applied; see [96, 97] for more details on the Douglas-Rachford splitting algorithm. Its iterative formula, $\forall \lambda \in (0, +\infty)$, reads

$$\begin{cases} \boldsymbol{x}^{(k)} = \text{prox}_{\lambda g}(\boldsymbol{v}^{(k)}), \\ \boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)} + \gamma^{(k)}(\text{prox}_{\lambda f}(2\boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k)}) - \boldsymbol{x}^{(k)}), \end{cases} \tag{3.29}$$

where $\gamma^{(k)} \in (\alpha, 2-\alpha)$, $\alpha \in (0,1)$. This iterative scheme needs the proximal operator for f and g individually. Therefore, the Douglas-Rachford splitting algorithm is restricted by the degree of difficulty of computing the proximal operators of f and g. The algorithm is summarized in Algorithm 2.

As an example, the Douglas-Rachford splitting algorithm can theoretically be used to solve the constrained problem (3.3) after moving its constraint into the objective functional by using the indicator function on an ℓ_2-ball. However, if $\boldsymbol{\Phi}$ is not an identity operator, as in radio interferometry, solving the proximal operator of this kind of indicator function is not easy computationally.

Algorithm 2 Douglas-Rachford Splitting Algorithm

1: **given** $\boldsymbol{v}^{(0)} \in \mathbb{R}^N, \alpha \in (0,1), \lambda > 0$
2: **repeat for** $k = 0, \ldots$
3: $\quad \boldsymbol{x}^{(k)} = \text{prox}_{\lambda g}(\boldsymbol{v}^{(k)})$
4: $\quad \gamma^{(k)} \in (\alpha, 2-\alpha)$
5: $\quad \boldsymbol{v}^{(k+1)} = \boldsymbol{v}^{(k)} + \gamma^{(k)}(\text{prox}_{\lambda f}(2\boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k)}) - \boldsymbol{x}^{(k)})$
6: **until convergence**

3.3.3 Alternating Direction Method of Multipliers

The Forward-Backward and Douglas-Rachford splitting algorithms presented above require the proximal operators f and g to be easy to compute. In

practice, this is sometimes not the case. For example, when function f involves explicitly a linear transformation $\boldsymbol{L} \in \mathbb{R}^{K \times N}$ (e.g. a measurement operator), we must consider the problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{L}\boldsymbol{x}) + g(\boldsymbol{x}), \tag{3.30}$$

where the proximal operator of $f(\boldsymbol{L}\boldsymbol{x})$ has no explicit expression.

Problem (3.30) can be addressed by the alternating direction method of multipliers (ADMM) [93, 98, 30]. After setting $\boldsymbol{v} = \boldsymbol{L}\boldsymbol{x}$, problem (3.30) becomes

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{v}) + g(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{v} = \boldsymbol{L}\boldsymbol{x}. \tag{3.31}$$

This problem has the following augmented Lagrangian with index $\lambda \in (0, +\infty)$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{z}) := f(\boldsymbol{v}) + g(\boldsymbol{x}) + \frac{1}{\lambda} \boldsymbol{z}^\dagger (\boldsymbol{L}\boldsymbol{x} - \boldsymbol{v}) + \frac{1}{2\lambda} \|\boldsymbol{L}\boldsymbol{x} - \boldsymbol{v}\|_{\ell_2}^2, \tag{3.32}$$

which can be solved alternatively corresponding to $\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{z}$. The variable $\boldsymbol{z}$ is typically known as a Lagrange multiplier. More precisely, $\mathcal{L}$ is minimized with respect to variables $\boldsymbol{x}$ and $\boldsymbol{v}$ alternatively while updating the dual variable $\boldsymbol{z}$ (using the dual ascent method [93]) to ensure that the constraint $\boldsymbol{v} = \boldsymbol{L}\boldsymbol{x}$ is met in the final solution, i.e.,

$$\boldsymbol{x}^{(k)} = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^N} \mathcal{L}(\boldsymbol{x}, \boldsymbol{v}^{(k)}, \boldsymbol{z}^{(k)}), \tag{3.33}$$

$$\boldsymbol{v}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{v} \in \mathbb{R}^K} \mathcal{L}(\boldsymbol{x}^{(k)}, \boldsymbol{v}, \boldsymbol{z}^{(k)}), \tag{3.34}$$

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} + (\boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k+1)}), \tag{3.35}$$

which can be rewritten as

$$\boldsymbol{x}^{(k)} = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \left(g(\boldsymbol{x}) + \frac{1}{2\lambda}\|\boldsymbol{L}\boldsymbol{x} - (\boldsymbol{v}^{(k)} - \boldsymbol{z}^{(k)})\|_{\ell_2}^2 \right), \tag{3.36}$$

$$\boldsymbol{v}^{(k+1)} = \underset{\boldsymbol{v}\in\mathbb{R}^K}{\operatorname{argmin}} \left(f(\boldsymbol{v}) + \frac{1}{2\lambda}\|\boldsymbol{v} - (\boldsymbol{L}\boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)})\|_{\ell_2}^2 \right), \tag{3.37}$$

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} + (\boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k+1)}). \tag{3.38}$$

Note, importantly, that the above problem (3.37) is actually computing the proximal operator of function f without involving the operator $\boldsymbol{L}$, which circumvents computing the proximal operator of $f(\boldsymbol{L}\boldsymbol{x})$ directly and generally has an explicit expression. We comment that ADMM has a close relationship to the Douglas-Rachford algorithm and Primal-Dual splitting (see [57, 89] for more details). The procedures of ADMM are briefly summarized in Algorithm 3, where we define

$$\operatorname{prox}_{\lambda g}^{\boldsymbol{L}}(\boldsymbol{u}) = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \left(g(\boldsymbol{x}) + \frac{1}{2\lambda}\|\boldsymbol{L}\boldsymbol{x} - \boldsymbol{u}\|_{\ell_2}^2 \right), \tag{3.39}$$

which may have a simple closed-form solution, or can be solved iteratively using a Forward-Backward method since its second term is differentiable.

Algorithm 3 Alternating Direction Method of Multipliers (ADMM)

1: **given** $\boldsymbol{z}^{(0)}, \boldsymbol{v}^{(0)} \in \mathbb{R}^K, \lambda > 0$
2: **repeat for** $k = 0, \ldots$
3: $\quad \boldsymbol{x}^{(k)} = \operatorname{prox}_{\lambda g}^{\boldsymbol{L}}(\boldsymbol{v}^{(k)} - \boldsymbol{z}^{(k)})$
4: $\quad \boldsymbol{v}^{(k+1)} = \operatorname{prox}_{\lambda f}(\boldsymbol{L}\boldsymbol{x}^{(k)} + \boldsymbol{z}^{(k)})$
5: $\quad \boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} + \boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{v}^{(k+1)}$
6: **until convergence**

A generalization of ADMM is simultaneous direction method of multipliers (SDMM), which can be applied to an objective function of more than two functions [99, 29]. However, this method often requires operator inversion which can be expensive [30].

In this work, after setting $f(\boldsymbol{L}\boldsymbol{x})$ to be the indicator function on an ℓ_2-ball, we can use ADMM to solve the constrained problem (3.3) with the positivity

constraint, i.e. the problem (3.17). We approximately solve $\text{prox}_{\lambda g}^{\boldsymbol{L}}(\boldsymbol{u})$ using an iteration of the Forward-Backward splitting method, and then use the Dual Forward-Backward splitting algorithm (which will be presented in Section 3.3.5) to solve $\text{prox}_{\lambda g}(\boldsymbol{u})$ iteratively, where g contains the ℓ_1-norm and the positivity constraint (see [30] and Section 5.1 for more detail).

3.3.4 Primal-Dual Splitting

In addition to ADMM, problem (3.30) can also be solved by the Primal-Dual splitting algorithm; an algorithm that like ADMM can be adapted to be distributed and performed in parallel [89, 30]. Firstly, the primal problem (3.30) can be rewritten as the following Primal-Dual formulation, i.e.,

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{z}} g(\boldsymbol{x}) + \langle \boldsymbol{L}\boldsymbol{x}, \boldsymbol{z} \rangle - f^*(\boldsymbol{z}), \tag{3.40}$$

which is a saddle point problem, where $\langle \boldsymbol{L}\boldsymbol{x}, \boldsymbol{z} \rangle = \boldsymbol{z}^\dagger \boldsymbol{L}\boldsymbol{x}$. It can be solved from minimizing and maximizing with respect to $\boldsymbol{x}$ and $\boldsymbol{z}$ alternatively, where for each subproblem the Forward-Backward ideas presented in Section 3.3.1 can be applied if needed. The Primal-Dual algorithm is summarized in Algorithm 4. Furthermore, Moreau decomposition in equation (3.16) can be used to calculate the proximal operator of f^* given the proximal operator of f, i.e.

$$\text{prox}_{\sigma f^*}(\boldsymbol{z}) = \boldsymbol{z} - \sigma \text{prox}_{\sigma^{-1} f}(\boldsymbol{z}/\sigma)\,.$$

Like ADMM, the Primal-Dual algorithm splits the objective function into two minimization problems, one is a Primal problem and the other is a Dual problem. In particular, ADMM can be considered to be in the family of Primal-Dual algorithms [89].

The Primal-Dual and ADMM algorithms are both very efficient algorithms to solve problems like (3.30). The Primal-Dual algorithm generally can achieve better convergence rates than ADMM. However, since ADMM needs to compute the proximal operators $\text{prox}_{\lambda f}$ and $\text{prox}_{\lambda g}^{\boldsymbol{L}}$ and the Primal-Dual algorithm needs to compute $\text{prox}_{\sigma f^*}$ and $\text{prox}_{\tau g}$, which method is more

appropriate often depends on the overall problem itself. In addition, there has been plenty of work to optimize them further, which makes their performance more comparable and in some cases equivalent to each other (see [89] for an overview of Primal-Dual methods).

Algorithm 4 Primal-Dual Algorithm

1: **given** $\boldsymbol{x}^{(0)} \in \mathbb{R}^N, \boldsymbol{z}^{(0)} \in \mathbb{R}^N, \tau, \sigma > 0, \theta \in [0,1]$
2: **repeat for** $k = 0, \ldots$
3: $\quad \boldsymbol{z}^{(k+1)} = \text{prox}_{\sigma f^*}(\boldsymbol{z}^{(k)} + \sigma \boldsymbol{L}\hat{\boldsymbol{x}}^{(k)})$
4: $\quad \boldsymbol{x}^{(k+1)} = \text{prox}_{\tau g}(\boldsymbol{x}^{(k)} - \tau \boldsymbol{L}^\dagger \boldsymbol{z}^{(k+1)})$
5: $\quad \hat{\boldsymbol{x}}^{(k+1)} = \boldsymbol{x}^{(k+1)} + \theta(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})$
6: **until convergence**

The derivation to this algorithm is almost identical to Dual Forward-Backward Splitting, which is mentioned in the next section and also considered a Primal-Dual algorithm.

3.3.5 Dual Forward-Backward Splitting

An algorithm closely related to the Primal-Dual algorithm is known as the Dual Forward-Backward splitting algorithm [100, 89]. To obtain the dual problem of (3.30), using Lagrangian multiplier $\boldsymbol{z}$, formulate the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{z}) := f(\boldsymbol{v}) + g(\boldsymbol{x}) + \langle \boldsymbol{L}\boldsymbol{x} - \boldsymbol{v}, \boldsymbol{z} \rangle . \tag{3.41}$$

By minimizing the Lagrangian over $\boldsymbol{x}$ and $\boldsymbol{v}$, we have

$$\begin{aligned} \inf_{\boldsymbol{x},\boldsymbol{v}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{z}) &= -\sup_{\boldsymbol{v}} \left(\langle \boldsymbol{z}, \boldsymbol{v} \rangle - f(\boldsymbol{v}) \right) - \sup_{\boldsymbol{x}} \left(\langle -\boldsymbol{L}^\dagger \boldsymbol{z}, \boldsymbol{x} \rangle - g(\boldsymbol{x}) \right) \\ &= -f^*(\boldsymbol{z}) - g^*(-\boldsymbol{L}^\dagger \boldsymbol{z}). \end{aligned} \tag{3.42}$$

Then we have the dual problem of problem (3.30), i.e.

$$\min_{\boldsymbol{z}} f^*(\boldsymbol{z}) + g^*(-\boldsymbol{L}^\dagger \boldsymbol{z}). \tag{3.43}$$

Note that term $g^*(-\boldsymbol{L}^\dagger \boldsymbol{z})$ is differentiable and it is shown in [92] that

$$\partial_{\boldsymbol{z}} g^* = -\boldsymbol{L}\left(\underset{\boldsymbol{v}\in X}{\operatorname{argmin}}\left\{\langle \boldsymbol{L}^\dagger \boldsymbol{z}, \boldsymbol{v}\rangle + g(\boldsymbol{v})\right\}\right). \tag{3.44}$$

Let $\bar{g}(\boldsymbol{z}) = g^*(-\boldsymbol{L}^\dagger \boldsymbol{z})$, applying the Forward-Backward splitting iterative scheme (3.28) with the relaxation on (3.43) $f^* + g^* \to \sigma f^* + \sigma g^*$ with $\sigma \in (0, \infty)$ and combing with the dual ascent method [93], we have

$$\boldsymbol{z}^{(k+1)} = \operatorname{prox}_{\sigma f^*}(\boldsymbol{z}^{(k)} - \sigma \nabla \bar{g}(\hat{\boldsymbol{z}}^{(k)})), \tag{3.45}$$

$$\hat{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k+1)} + \theta(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}). \tag{3.46}$$

which is the so-call Dual Forward-Backward splitting algorithm.

In particular, applying the Forward-Backward splitting iterative scheme (3.28) on the minimization problem in (3.44), we have

$$\nabla \bar{g}(\boldsymbol{z}) = -\boldsymbol{L}\operatorname{prox}_{\tau g}(\boldsymbol{x} - \tau \boldsymbol{L}^\dagger \boldsymbol{z}). \tag{3.47}$$

Let $\boldsymbol{x}^{(k+1)} = \operatorname{prox}_{\tau g}(\boldsymbol{x}^{(k)} - \tau \boldsymbol{L}^\dagger \hat{\boldsymbol{z}}^{(k)})$ and substituting (3.47) into (3.45), we have the following iteration scheme

$$\boldsymbol{x}^{(k+1)} = \operatorname{prox}_{\tau g}(\boldsymbol{x}^{(k)} - \tau \boldsymbol{L}^\dagger \hat{\boldsymbol{z}}^{(k)}) \tag{3.48}$$

$$\boldsymbol{z}^{(k+1)} = \operatorname{prox}_{\sigma f^*}(\boldsymbol{z}^{(k)} + \sigma \boldsymbol{L}\boldsymbol{x}^{(k+1)}) \tag{3.49}$$

$$\hat{\boldsymbol{z}}^{(k+1)} = \boldsymbol{z}^{(k+1)} + \theta(\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}) \tag{3.50}$$

After rearranging the order of the variables and replacing the relaxation strategy for $\boldsymbol{z}$ by $\boldsymbol{x}$, the above Dual Forward-Backward splitting algorithm turns into the Primal-Dual algorithm (see Algorithm 4). See [89] for more discussions about the relation between the Dual Forward-Backward splitting algorithm and the Primal-Dual algorithm.

Chapter 4

Sparse Image Reconstruction of Interferometric Observations

In this chapter we apply the alternating direction method of multipliers (ADMM) algorithm developed by [30] in the PURIFY software package, which has been entirely redesigned and re-implemented in C++, and apply it to observational data from the Very Large Array (VLA) and the Australia Telescope Compact Array (ATCA). In addition, we discuss conceptual differences between the restored CLEAN image and the reconstructed PURIFY model. The previous version of PURIFY supported only simple models of the measurement operator modelling the telescope. In this work we extend PURIFY to support a wider range of more accurate measurement operator models, including a number of different convolutional interpolation kernels (for gridding and degridding). Moreover, we study how the choice of kernel can affect the quality of sparse image reconstruction, and use an accurate kernel for image reconstruction of real interferometric observations.

The remaining sections of the chapter are structured as follows. We first discuss the development of the PURIFY software package used in this work in Section 4.1. Then, the interpolation kernels mentioned in Section 2.3 are tested and compared with PURIFY using simulations in Section 4.2. Section 4.3 discusses the similarities and differences between images recovered by CLEAN and PURIFY and also considerations in applying PURIFY to real

observational data. The reconstruction of images from observations made by the VLA and ATCA are presented in Section 4.4. Section 4.5 states the final conclusions.

4.1 PURIFY

To apply compressive sensing techniques to radio interferometry, one needs to pose the sparse regularization problems in Section 2.2.1 and then solve them using the measurement operator of Section 2.2.2. The software package PURIFY has been designed and written for this purpose.

The first public version of PURIFY was written in C and solved the problems described in [29], where it was shown on simulations to produce higher fidelity reconstructed images than standard radio interferometric imaging methods. To solve ℓ_1 minimization problems, PURIFY calls the Sparse OPTimization (SOPT) software package [24, 25]. This first version of PURIFY used the simultaneous-direction method of multipliers (SDMM) algorithm [29]. Recently, new algorithms have been developed for radio interferometry imaging by [30], including the proximal ADMM and primal dual algorithms, which have numerous advantages for the analysis of very large data-sets (see [30] for further discussion).

PURIFY (2.0.0) and SOPT (2.0.0) have been developed and used in this chapter. Both PURIFY and SOPT have been completely redesigned and rewritten in C++11 to work on Linux and Mac operating systems. The Eigen[1] library is used for matrix and array manipulation [101] and casacore[2] is used to read observational data in the form of measurement sets [102]. SOPT is not only useful for interferometric imaging: it is a general purpose code for solving sparse regularization problems and can be used to solve a variety of problems. The first version of PURIFY was limited to measurement operators based on Gaussian kernels for convolutional gridding. The new version of PURIFY, however, supports numerous kernels, including the state-of-the-art kernels

[1] `http://eigen.tuxfamily.org`
[2] `http://casacore.github.io/casacore`

discussed in the literature [*e.g.* 63], as described in Section 2.3. Additionally, the ADMM algorithm of [30] has been implemented in PURIFY and SOPT. Implementation of the primal dual algorithm of [30] into PURIFY and SOPT is available in a later release. The primal dual algorithm achieves greater flexibility, in terms of memory requirements and computational burden per iteration, by using full splitting and randomized updates. All results presented in this article are obtained with the ADMM algorithm, solving the analysis problem of Eq. 2.6, with an additional positivity constraint (however, it is possible to remove the positivity or reality constraints). While the development of fully distributed implementations of the algorithms supported by PURIFY and SOPT is can be found in later chapters, 2.0.0 versions are parallelised with OpenMP so that the gridding, degridding, and FFT calculations can be performed efficiently. The versions of PURIFY 2.0.0 [3] and SOPT 2.0.0 [4] used in this chapter are publicly available. However, other versions of PURIFY have been developed in this thesis, the differences can be seen in Table 4.1.

4.2 Simulations

In the previous chapter we described how the measurement operator $\mathbf{\Phi}$ approximates a direct Fourier transform. If this approximation is inaccurate, it will introduce error when recovering interferometric images. The choice of the interpolation kernel will therefore have an impact on reconstruction quality. In this section we perform simulations to assess the performance of different convolution kernels, using the ADMM algorithm [30] implemented with PURIFY 2.0.0 to recover images in the analysis framework, with an additional positivity constraint.

4.2.1 Simulations

To assess the impact that the interpolation kernel has on image reconstruction with PURIFY, we perform quality tests using simulated measurements. We

[3] `http://basp-group.github.io/purify`
[4] `http://basp-group.github.io/sopt`

Table 4.1: A listing of each PURIFY version and how each version has been modified throughout this thesis.

Version	Changes
PURIFY 1.0.0	This version used SDMM and the SARA algorithm and was developed in C and used SOPT 1.0. However, the SDMM algorithm is not included in future versions. Gaussian convolution kernels were used for degridding. More details can be found at [29].
PURIFY 2.0.0	PURIFY and SOPT were redeveloped in C++11, as described in this Chapter. This version includes more convolution kernels, uses the ADMM algorithm, and includes multi-threading.
PURIFY 3.0.1	The distributed measurement operators, wavelet operators and proximal operators are added as discussed in Chapter 5, allowing for distributed image reconstruction with ADMM. The new w-stacking w-projection hybrid algorithm described in Chapters 6 and 7 is also included.
PURIFY 3.0.1+	A new distributed measurement operator is developed in Chapter 8. This measurement operator allows kernels to be evenly distributed across compute nodes for construction and application during w-stacking. This load balances the computation of gridding kernels across a computing cluster.

compare the signal to noise ratio (SNR) of the reconstructed image with the ground truth image, reconstructing with different uv-coverages and different interpolation kernels. Note that we cannot replicate all of the complexities of the real observational setting with simple simulations. For example our simulated observations do not include contributes from sources outside the field of view. Nevertheless, simulations where the ground truth image is known are useful for a partial assessment of the performance of different convolution kernels.

To ensure the simulated measurements do not limit the reconstruction quality, a high quality 'ground truth' measurement operator is applied to 256×256 pixel test images of HII emission of M31 and of 30 Doradus (30Dor). In principle, we would generate the measurements using a direct Fourier

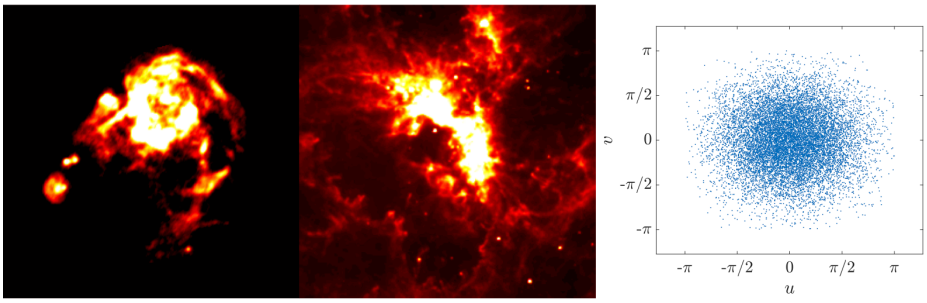

Figure 4.1: Ground truth images of M31 (left) and 30Dor (middle) used in simulations (of size 256 × 256). An example of a variable density visibility coverage in the Fourier plane, normalised to a domain of $\pm\pi$ (right). To generate a simulated observation, the measurement operator was applied to a ground truth image. Each simulation has added thermal noise and a random variable density coverage in the Fourier plane. The reconstruction quality was evaluated as a function of the number of Fourier components measured. The SNR was averaged over ten random coverages, with error bar given by the standard deviation (see Figure 4.2).

transform, however this is not practical due to the required computational resources. Instead, we increase the accuracy of the measurement operator. The Kaiser-Bessel kernel with a support of 8×8 pixels and an oversampling ratio of $\alpha = 2$ is used for the ground truth measurement operator. The Kaiser-Bessel kernel typically requires only a small support, so choosing a support of 8×8 provides an accurate measurement model [63].

We calculate the average SNR for reconstructing M31 and 30Dor from M visibilities, in a way that does not depend on a specific uv-coverage. The uv-coverages are randomly generated to follow a Gaussian variable sampling density with a standard deviation of $\pm\pi/3$ in the uv-plane, where the uv-plane has been normalised to a maximum height and width of $\pm\pi$. Ten sample uv-coverages were generated using M visibilities. The average SNR of a reconstruction from M visibilities was calculated using the ten sample uv-coverages. The standard deviation is used to estimate the spread of the SNRs of the reconstructed images. The test images of M31 and 30Dor and a sample uv-coverage are shown in Figure 4.1.

Gaussian noise was added to the simulated visibilities. The input SNR (ISNR) of the simulated visibilities was chosen to be 30 dB. The ISNR can be

used to calculate the standard deviation of the Gaussian distribution of noise [29]:

$$\sigma_n = \frac{\|\boldsymbol{y}_0\|_{\ell_2}}{\sqrt{M}} \times 10^{-\frac{\text{ISNR}}{20}}, \tag{4.1}$$

where $\boldsymbol{y}_0$ are the ground truth visibilities, M is the number of visibilities, and ISNR is measured in dB. This formula shows that the Gaussian noise is proportional to the root mean squared of the input visibilities, as expected form the definition of a signal to noise ratio. For example if the RMS value of the input visibilities is 1 Jy, then the noise distribution will have a spread of $10^{-\frac{\text{ISNR}}{20}}$ Jy.

The noise is assumed to be Gaussian and independently and identically distributed, which allows the use of the χ^2 distribution to estimate the bound ε for the ℓ_2-norm [29]:

$$\varepsilon^2 = (2M + 2\sqrt{4M})\frac{\sigma_n^2}{2}, \tag{4.2}$$

where for these tests we set ε^2 to two standard deviations above the mean of the χ^2 distribution. Following the work of [29], we calculate the SNR from the relation

$$\text{SNR} = 20\log_{10}\left[\frac{\|\boldsymbol{x}\|_{\ell_2}}{\|\boldsymbol{x} - \boldsymbol{x}^\star\|_{\ell_2}}\right], \tag{4.3}$$

where $\boldsymbol{x}$ is the ground truth image and $\boldsymbol{x}^\star$ is the reconstructed image.

We solve the ℓ_1 problem in the analysis setting (Eq. 2.6), using ADMM. For the ADMM step size γ, we use the fixed value of

$$\gamma = \beta\|\boldsymbol{\Psi}^\dagger\boldsymbol{\Phi}^\dagger\boldsymbol{y}_0\|_{\ell_\infty}, \tag{4.4}$$

with $\beta = 10^{-3}$, as recommended in [29] and [30], where $\|\boldsymbol{\Psi}^\dagger\boldsymbol{\Phi}^\dagger\boldsymbol{y}_0\|_{\ell_\infty}$ returns the maximum coefficient of the measurements in the wavelet representation. The reconstructions were solved by assuming sparsity in the SARA wavelet dictionary, which includes a Dirac (*i.e.* point source) basis and Daubechies wavelets 1 to 8 [24, 25]. Note that re-weighting is not considered. In these simulations, ADMM is stopped when the data fidelity constraint is satisfied

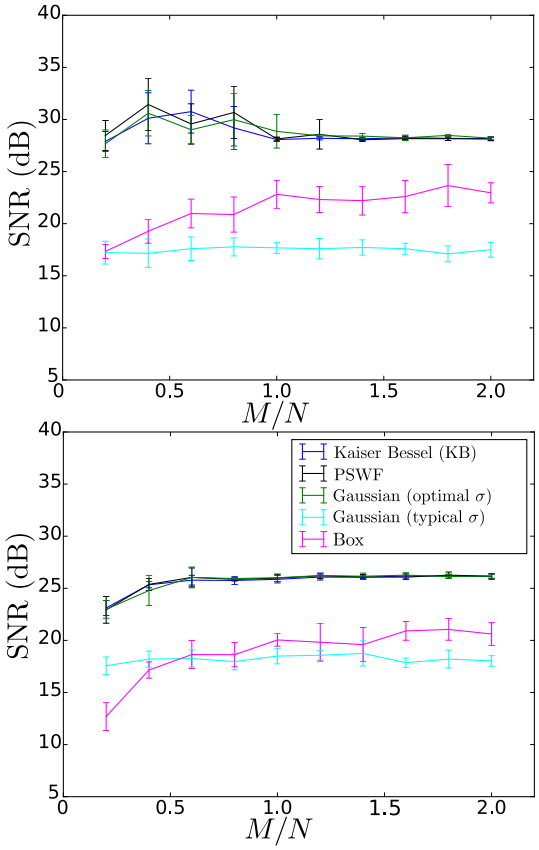

Figure 4.2: The top and bottom plots of the SNR of the reconstructions of M31 and 30Dor respectively, with an input SNR of 30dB. M/N is the ratio of measurements to pixels. Kaiser-Bessel and optimised Gaussian kernels can perform as well as the PSWF. Furthermore, choosing a bad choice of kernel, like a Box function or a Gaussian kernel with a typical σ, limits the possible quality of the reconstruction.

and the relative difference in the model image between iterations is less than 10^{-3}. Each reconstruction was run for a maximum of 100 iterations.

4.2.2 Results

The SNR of the reconstructed images as a function of number of visibilities M/N is shown in Figure 4.2 for both M31 and 30Dor. Simulations were performed using five of the different interpolation kernels described in Section 2.3, including: Kaiser-Bessel ($J = 4$, $\beta = 2.34J$), PSWF ($J = 6$, $\kappa = 1$), Box function ($J = 1$), Gaussian with a typical σ ($J = 4$, $\sigma = 1$) and optimised σ ($J = 4$, $\sigma = 0.31J^{0.52}$). An oversampling ratio of $\alpha = 2$ was used for all cases.

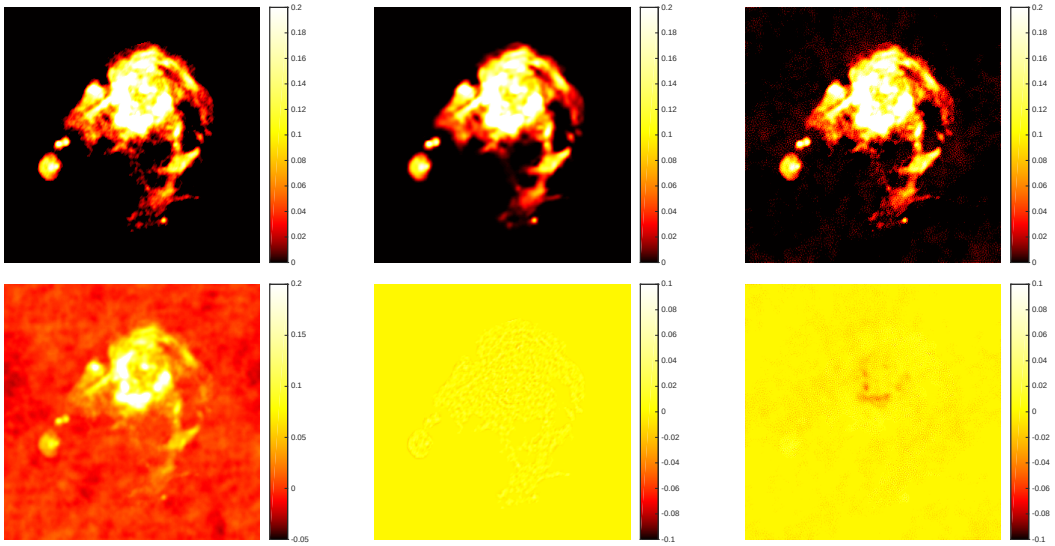

Figure 4.3: (M31) Left column shows ground truth (top) and dirty image (bottom). Middle column shows reconstructed image (top) and error (bottom) with Kaiser-Bessel kernel. Right column shows reconstructed image (top) and error (bottom) with Box kernel. For these simulations $M = 2N$ visibilities were used, with an input SNR of 30 dB. The error image shows that the Box kernel reconstruction has artefacts, which explains why the SNR is lower than the Kaiser-Bessel reconstruction. The Box kernel reconstruction did not converge within 100 iterations (based on the convergence criteria described in the text), while the Kaiser-Bessel kernel reconstruction did.

Similar SNR results were found for reconstructions using the SARA dictionary for both the M31 and 30Dor images. The Kaiser-Bessel, PSWF, and Gaussian kernels with an optimised σ were found to provide reconstructions of the same level of quality. The tests for these kernels converged within 100 iterations.

However, the Gaussian kernel with a typical σ and the Box function provide reconstructions that have an SNR that is 5 to 10 dB below the other kernels in these tests. Furthermore, for the Box kernel, the reconstructions had often not converged within 100 iterations, while for the Gaussian with a typical σ the majority of tests converged.

To illustrate the difference between reconstructions using the Kaiser-Bessel and Box interpolation kernels, Figure 4.3 and Figure 4.4 show example reconstructions for $M = 2N$. Error images are also shown, defined as the

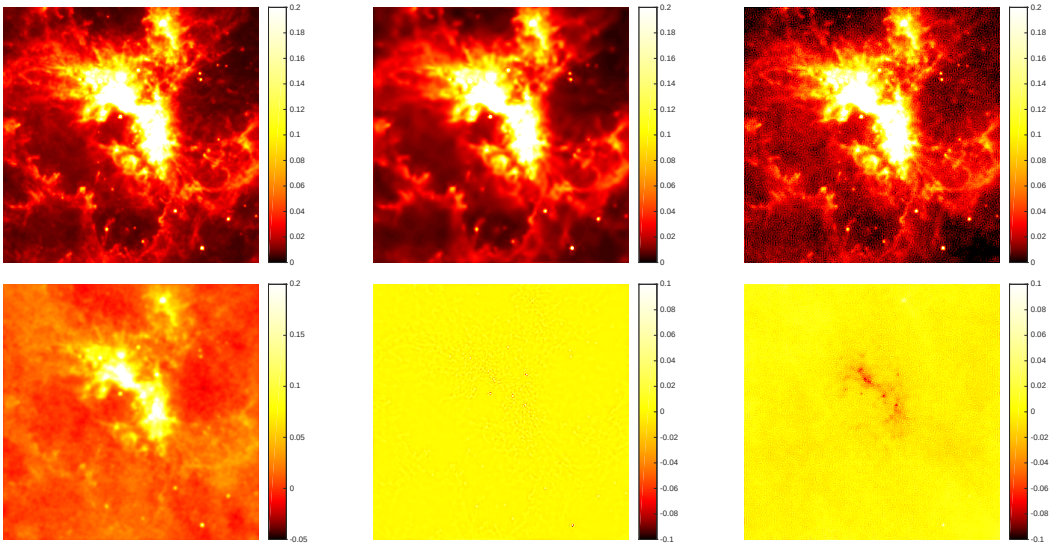

Figure 4.4: (30Dor) Left column shows ground truth (top) and dirty image (bottom). Middle column shows reconstructed image (top) and error (bottom) with Kaiser-Bessel kernel. Right column shows reconstructed image (top) and error (bottom) with Box kernel. For these simulations $M = 2N$ visibilities were used, with an input SNR of 30 dB. The error image shows that the Box kernel reconstruction has artefacts, which explains why the SNR is lower than the Kaiser-Bessel reconstruction. The Box kernel reconstruction did not converge within 100 iterations (based on the convergence criteria described in the text), while the Kaiser-Bessel kernel reconstruction did.

difference between the reconstructed and ground truth image. The structure in the Kaiser-Bessel kernel error images looks close to Gaussian error. The structure in the Box kernel error images shows artefacts, which spread throughout the reconstructed image, explaining the lower SNR.

Tests were also performed using only a Dirac basis as the sparsifying dictionary, which provides a proxy for the CLEAN algorithm. The results obtained were consistent with those found with the SARA wavelet dictionary. This suggests that these results found here are likely to apply also to CLEAN and other similar algorithms.

Additional tests were performed at an ISNR of 10 dB, where it was found that there was minimal difference between the reconstructed SNR with different interpolation kernels. This suggests that the choice of interpolation kernel will limit the reconstruction SNR when the level of artefacts is

comparable or greater than the noise level. Consequently, for high dynamic range imaging the choice of kernel is important.

4.2.3 Discussion

Many calibration and imaging techniques depend on gridding and degridding methods to approximate the Fourier transform. While it has been understood that gridding methods in radio astronomy can impact image quality [65, 64, 62, 59], the current study confirms that gridding with poor kernels reduces the quality of images that can be recovered by sparse regularization approaches, such as those implemented in PURIFY, and also those that can be recovered by CLEAN. The magnitude of the impact depends on the quality of the measurements. For high quality measurements with high ISNR, the use of poor interpolation kernels will limit the SNR of the reconstruction. At low measurement ISNR, noise dominates the limit imposed by the interpolation kernel.

In particular, we have found that the Gaussian kernel with an optimal σ and Kaiser-Bessel kernel perform as well as the PSWF, while using a smaller support. Moreover, both of the former have analytic forms that can be easily evaluated, which is not the case for the PSWF, where approximations are typically made and look-up-tables used. This suggests that the Kaiser-Bessel kernel is just as good as the PSWF for sparse image reconstruction, and computationally less expensive with a smaller support. These finding are consistent with previous works, suggesting that the Kaiser-Bessel kernel is on par with optimal kernels [64, 73, 63].

4.3 Applying PURIFY to observations

The application of compressive sensing to radio interferometry is a relatively new development and to date most of the exploration of compressive sensing has been via simulated observations. Simulations are useful for testing the performance of reconstructions because the ground truth and noise level is known, and appropriate algorithm parameters can be estimated

accurately. However, this is not the case when reconstructing images from real observations.

In the next section (Section 4.4) we demonstrate that PURIFY can perform high quality image reconstruction on real observations and compare reconstructed images with those recovered by the CLEAN algorithm. However, to compare PURIFY and CLEAN reconstructions, we need to make clear the fundamental differences between the final outputs produced by each approach. In this section we discuss CLEAN in the context of sparse image reconstruction and clarify where the differences lie. In addition we describe how to apply PURIFY to real observations, including how to set the pixel size, weighting, and other parameters of the algorithm.

4.3.1 CLEAN comparison

Variations of CLEAN, such as Clark and Cotton-Schwab CLEAN [45, 46], work by iteratively building a model of the sky in major and minor cycles. This can be expressed in terms of iterations [30]

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \mathcal{T}\Big(\boldsymbol{\Phi}^{\dagger}\Big(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}^{(t-1)}\Big) \Big), \tag{4.5}$$

where $\boldsymbol{x}^{(t)}$ represents the solution after t iterations, and $\mathcal{T}$ represents the process of deconvolving the brightest sources in the residuals $\boldsymbol{\Phi}^{\dagger}\Big(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}^{(t-1)}\Big)$.

CLEAN operates in minor and major cycles, the minor cycles $\mathcal{T}$ are performed after the calculation of a major cycle $\boldsymbol{\Phi}^{\dagger}\Big(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}^{(t-1)}\Big)$. The minor cycles iteratively subtract the brightest sources from the image using an approximate point-spread function (PSF), which allows the location of the peaks of multiple sources to be found quickly. The major cycle performs an accurate subtraction of sources located in the minor cycle to generate the residuals for the next round of minor cycles.

CLEAN is essentially a matching pursuit algorithm [43], with a threshold constraint as suggested by [17], where the algorithm stops when the peak pixel of the residual image is below $\varepsilon_{\text{threshold}}$, $\| \boldsymbol{\Phi}^{\dagger}\left(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\right) \|_{\ell_\infty} \leq \varepsilon_{\text{threshold}}$.

Most variations of CLEAN impose the prior that the sky is sparse in a Dirac representation (CLEAN components/point sources), while multi-scale and adaptive scale pixel decomposition (ASP) CLEAN consider atoms with wider support to better model a sky containing extended sources [103, 49, 104]. The solution obtained by the CLEAN algorithm $\boldsymbol{x}$ is typically called a CLEAN *component* image.

4.3.1.1 CLEAN restoration

In the case that the CLEAN components $\boldsymbol{x}$ could accurately model the entire sky, there would be nothing but noise remaining in the residuals. However, often it is not possible for CLEAN components to model diffuse structures that cannot be represented efficiently by point sources. For this reason, a final *restored* image is constructed to include structures not deconvolved by CLEAN. The final restored image is found by convolving the CLEAN components with a Gaussian and then adding the residual image:

$$\boldsymbol{x}^{\text{restored}} = \boldsymbol{P}\boldsymbol{x} + \boldsymbol{\Phi}^{\dagger}\left(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\right), \tag{4.6}$$

where $\boldsymbol{P}$ is a post-processing operator that convolves $\boldsymbol{x}$ with a Gaussian of the same full width at half maximum as the dirty beam. The final restored image is expressed in units of Jy/Beam. These modifications mean the process of constructing a final restored image is not consistent with finding a solution that best fits the data for a given prior, even if the motivations are pragmatic. The restoration process hides the poor modeling of the CLEAN component model, making the overall image more aesthetically pleasing.

The CLEAN residuals are therefore not a true representation of how well the restored image models the true sky. Rather, the residuals $\boldsymbol{\Phi}^{\dagger}\left(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\right)$ of a reconstructed CLEAN image are due to the CLEAN components $\boldsymbol{x}$, not the final restored image $\boldsymbol{x}^{\text{restored}}$.

An additional systematic that can occur with the CLEAN method is that the dirty beam may not be well approximated by a Gaussian, which

is assumed in constructing the restored image [105]. This could impact studies that require accurate characterization of point sources, such as weak lensing [106]. Additionally, in low frequency imaging the ionospheric distortion on short timescales can produce a non-Gaussian dirty beam. For low frequency radio astronomy this is a serious issue, as discussed in [107].

4.3.2 PURIFY

PURIFY adopts the prior that the sky has a sparse representation. This can include a representation as a collection of point sources and/or single or multiple wavelet dictionaries. This allows more flexibility when modelling both point sources and extended sources simultaneously, providing more accurate deconvolution of complex structure. As a result diffuse structures are not expected in the residual image, hence, there is no need to combine the model with the residuals as is done with the CLEAN algorithm. PURIFY provides a final image that is reconstructed accurately enough to eliminate the need to convolve the model with a Gaussian beam.

PURIFY therefore provides several advantages over CLEAN. First, it means the residuals correspond to the final image used for scientific analysis, such that the final image is the model that minimizes the error (this is not true for the CLEAN restored image). Second, no restoration process provides an advantage when computing statistics on an image and for general scientific interpretation, because there is no need to include Gaussian and dirty beam dependence when analyzing the flux values of pixels. This simplifies analyzing spectral index or rotation measure values across diffuse sources [108].

4.3.3 Choice of pixel size

The final image recovered by PURIFY is sampled at discrete pixel values, hence there is a choice in the size of a pixel of the discrete image representing the sky brightness. The size and number of pixels can be determined by the resolution and field of view of the telescope. The size of the pixel can be estimated from the resolving power of the longest baseline and number of pixels determined

by the field of view imaged (by the Nyquist relation).

Radio astronomy packages such as Common Astronomy Software Applications (CASA) or MIRIAD typically assume between 3 and 5 pixels across the FWHM of the synthesised beam, found by least squares fitting a Gaussian to the main lobe of the synthesised beam [102, 71, 50]. The synthesized beam can be affected by choice of weighting and the synthesised beam may not be Gaussian, which can affect the Gaussian fit of the FWHM. In particular, the fitted FWHM will not always match the resolution of the longest baseline.[5]

Ideally, the size of the image should include all of the bright sources within the telescope's field of view. When bright sources are outside the imaged field of view they cannot be modelled but will be aliased into the imaged region, which can limit image fidelity. Bright point sources outside the imaged region can also limit image fidelity since their synthesized beam sidelobes are not modelled and removed during image reconstruction.

PURIFY is flexible with regard to the pixel sampling rate and size and these parameters can be set by the user. However, the default approach to setting the pixel size is to adopt Nyquist sampling where the resolution of the model is fundamentally limited by the uv-sampling pattern. However, for bright sources with a large SNR it is possible to accurately reconstruct super resolved structures well past the measured uv coverage. This can only be done by reconstructing a higher resolution image. Sparse image reconstruction algorithms since this work have been used to accurately super resolve the high SNR radio core of Cygnus A [80].

4.3.4 Weighting

In radio interferometry it is standard practice to weight the measurements according to natural, uniform, or robust weighting schemes, which are described in detail in [109]. The visibilities are weighted by the natural

[5]The author has found that fitting the synthesized beam for pixel size can produce much larger pixels when using natural weighting compared to uniform weighting.

weighting scheme to optimize the sensitivity of an observation. However, for observations containing extended emission, the sidelobes in the image domain due to natural weighting can dominate the synthesised beam. In this case, CLEAN can perform badly, so the visibilities are uniformly weighted to minimize sidelobes. Typically, there are more short baselines than long baselines, which lowers the average resolution of the naturally weighted dirty and restored maps. We concisely review different weighting schemes, including the standard natural, uniform and robust weighting schemes used in radio interferometry. PURIFY 2.0 supports all of these schemes.

4.3.4.1 Natural

Natural weighting maximises the sensitivity of the observation, with weights set to $\boldsymbol{W}_{ii}^{\text{natural}} = \sigma_i^{-1}$, where σ_i is the standard deviation of the error for visibility $\boldsymbol{y}_i$. Note that here we consider the weighting operator as a component of the measurement operator following Eq. 2.7, hence its entries are given by σ_i^{-1}, rather than a scaling of the visibilities only, in which case the weights are given by σ_i^{-2}. Natural weighting is also known as whitening: each measurement has the same (unit) variance after weighting [29]. Whitening is a standard weighting approach in statistical data analysis and image processing. Using natural weighting for interferometric imaging allows one to use a χ^2 distribution when comparing how well the model visibilities fit the data, which can be used for a statistical interpretation of the bound on the ℓ_2-norm.

4.3.4.2 Uniform

Uniform weighting minimises the amplitude of sidelobes over a given field of view, which is achieved by calculating an average weighting from the nearest neighbours of a visibility. Explicitly, an average weight is calculated by

$$\boldsymbol{W}_{ii}^{\text{gridded}} = \sqrt{\frac{1}{|\mathcal{S}_i|} \sum_{k \in \mathcal{S}_i} \left(\boldsymbol{W}_{k,k}^{\text{natural}} \right)^2}, \tag{4.7}$$

where $\mathcal{S}_i$ denotes the set of visibility indices that are included in the grid cell corresponding to visibility i, and $|\mathcal{S}_i|$ denotes the number of elements in $\mathcal{S}_i$.

The uniform weights are then calculated by normalising the natural weights:

$$\boldsymbol{W}_{ii}^{\text{uniform}} = \frac{\boldsymbol{W}_{ii}^{\text{natural}}}{\boldsymbol{W}_{ii}^{\text{gridded}}}\,. \tag{4.8}$$

It is possible to control the field of view at which the synthesised beam sidelobe suppression due to weighting occurs by changing the resolution of the grid cells. As the grid resolution increases, the field of view for dirty beam sidelobe suppression increases, although the suppression level is reduced. As the field of view for suppression increases, the weighting tends to natural weighting.

4.3.4.3 Robust

Robust weighting allows one to vary a robustness parameter R to continuously move between natural and uniform weighting:

$$\boldsymbol{W}_{ii}^{\text{robust}} = \frac{\boldsymbol{W}_{ii}^{\text{natural}}}{\sqrt{1+\rho\left(\boldsymbol{W}_{ii}^{\text{gridded}}\right)^2}} \tag{4.9}$$

where

$$\rho = \frac{\sum_k \left(\boldsymbol{W}_{kk}^{\text{natural}}\right)^2}{\sum_k \left(\boldsymbol{W}_{kk}^{\text{gridded}}\right)^4} \times 10^{-2R+\log_{10}(25)}\,. \tag{4.10}$$

4.3.5 Parameter choice

The parameters of PURIFY are set automatically, following the recommendations of [29] and [30]. We also consider some minor modifications of these schemes that can be useful when analysing real observations, where, for example, the errors on the visibilities that are provided (*i.e.* weights) may not be accurate. Two parameters that need to be set carefully are the bound on the data fidelity error bound ε and the step size of the algorithm γ. We adopt a method to estimate ε using the Stokes V visibilities and to adaptively estimate the step size γ during the first steps of the algorithm.

4.3.5.1 Choosing the error bound ε

The parameter ε determines the error on how closely the model visibilities are required to match the measured visibilities. If ε is too small the model will start to fit to noise and if ε is too large the model will not model structures accurately.

In the case of natural weighting, ε can be estimated by [29]

$$\varepsilon^2 = (2M + q\sqrt{4M})\frac{\sigma_n^2}{2}, \tag{4.11}$$

where ε^2 is set to q standard deviations above the mean of the χ^2 distribution. However, for typical observations $2M \gg \sqrt{4M}$, so this interpretation is less useful (due to the concentration of measure in high dimensions). For real observations with large M we simply estimate ε from the mean of the χ^2 distribution and allow a scaling:

$$\varepsilon_\eta = \eta\sqrt{M}\sigma_n, \tag{4.12}$$

where η allows one to vary ε to include non-thermal noise contributions, such as instrumental errors and radio frequency interference (RFI). When using this latter approach to set ε we explicitly denote the η dependence by ε_η.

In principle, standard calibration and self-calibration methods can be applied with PURIFY but to date these have not yet been tested. Such an approach may be considered by choosing a high error bound for ε to generate a sky model of the brightest sources, applying a calibration algorithm to recover calibration parameters, before iterating.

In the case that the source of noise in the visibilities is thermal, the weights should be accurate. However, if the weights are not accurate we adopt a method where Stokes V can be used to estimate the noise level and thus ε. This is because Stokes V rarely contains astrophysical sources and so is dominated by thermal noise. To estimate the noise on a measurement, we use

the median absolute deviation (MAD) method [110, 111]

$$\sigma_n = \sqrt{\left[\frac{\text{Median}(\text{Real}(\boldsymbol{W}\mathsf{y}_\text{V}))}{0.67449}\right]^2 + \left[\frac{\text{Median}(\text{Imag}(\boldsymbol{W}\mathsf{y}_\text{V}))}{0.67449}\right]^2}, \tag{4.13}$$

where $\boldsymbol{W}\mathsf{y}_\text{V}$ is the weighted Stokes V visibilities. The MAD method provides a robust way to estimate σ_n given Gaussian noise, and should be reliable when Stokes V is dominated by thermal noise.

Furthermore, if the weights are only proportional to the standard deviation of noise, they will be incorrect by a scaling factor. The MAD method can be used to determine the standard deviation of the noise from a sample distribution. While using the MAD method to estimate σ_n is intended to work for thermal noise contributions, it might not be accurate when there are polarimetric, amplitude, and phase calibration errors or RFI.

4.3.5.2 Adapting the step size γ

In [29], it is suggested that the algorithm step size γ can be set by

$$\gamma = \beta \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}^{(0)}\|_{\ell_\infty}, \tag{4.14}$$

$\boldsymbol{x}^{(0)}$ is an initial estimate of the image. Typically, the initial estimate is chosen as $\boldsymbol{x}^{(0)} = \boldsymbol{\Phi}^\dagger \boldsymbol{y}$ (*i.e.* the dirty image). While the choice of γ should not affect the final result of the algorithm, it does affect the rate of convergence.

We adapt this approach and allow γ to be re-estimated as the algorithm progresses, before settling on a fixed value of γ to guarantee convergence. In this case, a candidate adaptive step size for the i-th iteration can be calculated $\tilde{\gamma}_i = \beta \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}^{(i)}\|_{\ell_\infty}$. If the current candidate for the step size changes by a small amount only, there is no need to change the step size used. In this case, a

general rule for adapting the step size can be set:

$$\gamma_i = \begin{cases} \tilde{\gamma}_i, & \text{if } \frac{\tilde{\gamma}_i - \gamma_{i-1}}{\gamma_{i-1}} > \delta_{\text{adapt}} \\ \gamma_{i-1}, & \text{if } \frac{\tilde{\gamma}_i - \gamma_{i-1}}{\gamma_{i-1}} \leq \delta_{\text{adapt}} \\ \gamma_{i-1}, & \text{if } i \geq i_{\text{adapt}} \end{cases}, \tag{4.15}$$

where δ_{adapt} is the minimum relative difference needed for adapting the step size and i_{adapt} is the number of iterations after which the step size will not be adapted and will remain fixed.

4.3.6 Input parameters of PURIFY

As described already, the parameters of PURIFY are set automatically and so PURIFY can be run simply be providing the filename of an input measurement set and the output filename of the image to be recovered. The user does not need to set any parameters. However, the default settings can be overridden.

The main parameters of interest that a user may want to overwrite are specified in Table 4.2. These include the η value in setting ε_η, the β parameter in setting γ, the δ_{adapt} and i_{adapt} parameters that control adapting γ, the relative variation of the solution criteria δ, the residual norm convergence criteria ξ, and the maximum number of iterations i_{max}.

In analysing the observations considered in the next section, the value of η varies from 1.4 to 7, and depends on the quality of the data set, such as how free it is from calibration error and RFI. The i_{adapt} parameter is set to a fraction of the maximum number of iterations. It is important to set i_{adapt} such that the step size γ stops adapting before convergence. The relative variation criteria of the objective function was chosen to be $\delta = 5 \times 10^{-3}$. The choice of residual norm convergence criteria ξ also depends on the quality of the data set.

Table 4.2: Description of main user parameters for using PURIFY to reconstruct an observation. All parameters are set automatically but can be overwritten.

Parameter	PURIFY option	Description	Value
η	`--l2_bound`	Parameterization of the fidelity constraint: $\varepsilon_\eta = \eta\sqrt{M}\sigma_n$.	$\eta = 1.4$ (default); $\eta \in [1,10]$ (typical).
β	`--beta`	Parameterization of the step size of the algorithm: $\tilde{\gamma}_i = \beta\|\boldsymbol{\Psi}^\dagger \boldsymbol{x}^{(i)}\|_{\ell_\infty}$ (default). One can also fix $\gamma = \beta\|\boldsymbol{\Psi}^\dagger \boldsymbol{x}^{(0)}\|_{\ell_\infty}$.	$\beta = 10^{-3}$ (default)
δ_{adapt}	`--relative_gamma_adapt`	Relative difference criteria for adapting γ_i.	$\delta_{\text{adapt}} = 0.01$ (default).
i_{adapt}	`--adapt_iter`	Number of iterations to consider adapting the step size γ_i (should be before convergence).	$i_{\text{adapt}} = 100$ (default).
δ	`--relative_variation`	Relative difference convergence criteria on the ℓ_2-norm of the solution: $\frac{\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)}\|_{\ell_2}}{\|\boldsymbol{x}^{(i)}\|_{\ell_2}} \leq \delta$.	$\delta = 5 \times 10^{-3}$ (default).
ξ	`--residual_convergence`	Convergence criteria on the ℓ_2 residual norm: $\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \xi\varepsilon_\eta$	$\xi = 1$ (default); require $\xi \geq 1$.
$i_{\max}$	`--niters`	Maximum number of iterations.	$i_{\max} = \infty$ (default).

4.4 PURIFY reconstruction of observations

In this section we compare the use of PURIFY and Cotton-Schwab CLEAN for reconstructing total intensity (Stokes I) observations made by the Very Large Array (VLA) and the Australia Telescope Compact Array (ATCA). In particular, we consider observations of the radio galaxies 3C129, Cygnus A, PKS J0334-39, and PKS J0116-473. To perform the Cotton-Schwab CLEAN algorithm, we use WSCLEAN [50]. WSCLEAN is a standard choice for imaging in several MWA [19] science pipelines including continuum, transients, EoR and polarization modes [112, 113, 114, 115, 116]. For PURIFY, we present results using the ADMM algorithm [30], in the analysis setting, with a positivity constraint and the SARA wavelet dictionary [24], without reweighting.

4.4.1 Observations

In this section we discuss the details of the observations considered. The sampling patterns in the uv-plane for each observation are shown in Figure 4.5.

4.4.1.1 3C129

The observation of the bent tailed radio galaxy 3C129 has a phase center of RA = 04h 45m 31.695s, DEC = $+44^\circ$ $55'$ $19.95''$ (J2000), and was obtained from the NRAO archive. It was performed using the VLA with the project code AT0166, with two 50 MHz channels centered at 4.59 and 4.89 GHz. The observations were performed on the 25^{th} of July 1994 in configuration B and 3^{rd} of November 1994 in configuration C respectively. The total integration time was 79.7 minutes in configuration B and 15.8 minutes in configuration C. The calibration and flagging of radio frequency interference was performed using CASA, following the standard procedure found in the CASA manual. The gains were calibrated using sources 0420+417, 0518+165, and 0134+329, to solve for the instrumental and source polarization. Source 0420+417 was observed alternately to solve the polarimetric calibration solutions with

paralactic angle coverage.

4.4.1.2 Cygnus A

The VLA observation and reduction of Cygnus A in the X band (central frequency of 8.953 GHz, and 92 MHz bandwidth) was performed by Rick Perley[6] (PI:Perley, project code 14B-336 (legacy: AP658)). Cygnus A was observed in 2014 between the 3rd of November (18:39:44.0 UTC) to 4th November (04:28:12.0 UTC), using configuration C. The pointing centre was located at RA = 19h 59m 28.356s, DEC = $+40^\circ$ $44'$ $02.075''$ (J2000). The data was reduced and calibrated using AIPS, following standard procedure that can be found in the AIPS Cookbook[7].

4.4.1.3 PKS J0334-39

The observation of PKS J0334-39 was first presented in the work of [108], where the tailed radio galaxy's polametric structure was used to probe the environment of the galaxy cluster Abell 3135. The observation was also reprocessed using self calibration in [51], where it was used as an example of applying Generalised Complex CLEAN [51] to a observation. The observation was performed using the ATCA (with the pre-CABB correlator) in 2001 is centered on RA = 03h 34m 07.18s DEC = -39°00′03.19″ (J2000), at a central frequency of 1.384 GHz. There are 12 channels, each with a width of 8 MHz. The observation was performed in configuration 6A for 59 minutes, 1.5A for 76 minutes, 750A for 79.7 minutes, 375 for 75.4 minutes. A detailed description of the calibration procedure, performed using MIRIAD, can be found in [108].

4.4.1.4 PKS J0116-473

The observation of PKS J0116-473 used in this work was first presented in [117]. The total intensity, polametric structure, and morphology of PKS J0116-473 have been studied in detail at 12 and 22 cm wavelengths. The ATCA observations of PKS J0116-473 used in this work were extracted from the archive (PI:Shankar, project code C770), then calibrated and flagged following

[6]Private communication.

[7]`http://www.aips.nrao.edu/cook.html`

a standard ATCA data reduction procedure found in the MIRIAD manual[8]. The phase center is located at RA = 14h 59m, 15.75s DEC =-36° 55′ 47.87″ (J2000), and the central observation frequency is 1.384 GHz. After flagging and removing channels due to cross-channel interference, there are 12 channels each with 8 MHz channel width. The observations were performed in 1999, on the 10^{th} and 12^{th} of January (configuration 375, 1115 minutes integration), on the 7^{th} (750C, 1088.3 minutes) and 20^{th} (6C, 1109.3 minutes) of February, and on the 24^{th} and 25^{th} of April (1.5C, 1112 minutes). Sources PKS B1934-638 and PKS B0823-500 were used to set the flux density scale at 1.384 GHz. The time variations in complex antenna gains and bandpass were calibrated using alternating observations of the unresolved source PKS B0153-410.

4.4.2 Reconstructions

In this section we present the reconstructions from real observations. We show the reconstructed model image, alongside the residuals. For the CLEAN reconstructions we show the post-processed restored image (see Section 4.3.1.1), while for PURIFY there is no need for post-processing so there is no restored image but only a reconstructed model image (see Section 4.3.1.1). For PURIFY reconstructions we use natural weighting, and for CLEAN we use both natural and uniform weightings.[9]

The CLEAN thresholds and FWHM of the restoring beams can be found in Table 4.3. The CLEAN components are restricted to be positive valued. Allowing negativity might improve the fit for both CLEAN and PURIFY, but we choose positivity for this comparison. CLEAN has not been restricted to regions around the source. CLEAN was run until the residual peak reached the cutoff flux value. The cutoff flux value was measured after trialling CLEAN with different values and estimating 3 times the RMS noise of the residual map. We make it clear that there are many different CLEAN configurations

[8] `http://www.atnf.csiro.au/computing/software/miriad/userguide/userhtml.html`

[9] Rather than using measurement sets for the ATCA data sets, the tables were read with PURIFY from uvfits files. In all other cases, the observations were read from measurement sets.

and CLEAN algorithms that might produce better looking images, but testing when CLEAN works best is out of scope for this work. We are careful to make the distinction between the *restored* image and the *reconstructed* image for CLEAN (see Section 4.3.1.1), since the restored image is not used to generate the residuals. When we refer to the reconstructed image, we are referring to the CLEAN component image.

For PURIFY, the error constraint in the model is set using ε_η. The ADMM step size was set adaptively as described in Section 4.3.5.2. PURIFY images have a resolution set by the longest baseline in the observation.

Images recovered by CLEAN and PURIFY, and auxiliary plots, are shown in Figures 4.6, 4.7, 4.8, and 4.9. Reconstructions of the source 3C129 are shown in Figure 4.6, for a pixel width and height of 0.4 arcseconds. The PURIFY reconstruction was performed using a value of $\eta = 0.9$ and $\xi = 1$, and ran for 75 iterations. The step size was adapted for the first $i_{\text{adapt}} = 20$ iterations. Figure 4.7 contains the reconstructions of Cygnus A, for a pixel width and height of 0.5 arcseconds. The PURIFY reconstruction was performed using $\eta = 2.14$ and $\xi = 7.07$, and ran for 183 iterations. The step size was adapted for the first $i_{\text{adapt}} = 100$ iterations. Reconstructions of the source PKS J0334-39 are shown in Figure 4.8, for a pixel width and height of 2 arcseconds. The PURIFY reconstruction was performed using $\eta = 1$ and $\xi = 2$, and ran for 372 iterations. The step size was adapted for the first $i_{\text{adapt}} = 200$ iterations. Reconstructions of the source PKS J0116-473 are shown in Figure 4.9, for a pixel width and height of 2.4 arcseconds. The PURIFY reconstruction was performed using $\eta = 1$ and $\xi = 2.3$, and ran for 707 iterations. The step size was adapted for the first $i_{\text{adapt}} = 500$ iterations.

The run times for these reconstructions range from an hour to several hours using a high-performance desktop computer, to produces images of sizes 1024×1024 and 2048×2048 pixels. Currently, a large factor in the computational cost and run time for PURIFY is computing wavelet transforms for a number of dictionaries. In the case that only a Dirac basis

is used and no wavelet transforms are performed, the run time is reduced considerably for large image sizes. However, this greatly reduces the quality of the reconstructed image, because a Dirac basis is not an efficient representation of extended structures. Highly distributed and parallelised algorithms will be implemented in the following chapters to reduce the run-time significantly [30]. While CLEAN methods appear computationally efficient, this comes at a significant cost to reconstruction quality and with additional restrictions on the ability for distribution.

In all cases PURIFY provides more complete reconstructions than CLEAN. When comparing with the CLEAN component images, the CLEAN component images are not smooth and do not reconstruct the diffuse emission well (due to the point source model of CLEAN), while the PURIFY recovered images model diffuse emission. After post-processing the CLEAN component image to yield the CLEAN restored image and comparing with PURIFY, it is also clear that PURIFY provides higher quality reconstructions.

The dirty and residual images of PURIFY are shown in Jy/Beam for comparison. To convert from units of Jy/Pixel to Jy/Beam, the image is divided by the peak of the point spread function $\boldsymbol{\Phi}^{\dagger}\boldsymbol{W}\mathbf{1}$, where $\mathbf{1}$ denotes a vector of ones. This allows direct comparisons of the residual images between CLEAN and PURIFY, since they will have the same units without arbitrary scaling. To compare the residuals the scale of the colour axis has been set to a common scale, using 3 times the median root-mean-squared (RMS) values between the residual images in Table 4.4. The histograms show the full range of pixel values in the residuals, determined by the peak of the absolute residuals, to allow one to inspect outliers.

For all observations, PURIFY can model faint extended structure while also modeling the bright compact sources. Additionally, the PURIFY model has left little structure in the residuals. This is also clear from the histogram of the residual pixel brightness, which shows the residuals are dominated by Gaussian noise. The CLEAN reconstruction leaves visible diffuse structure in

the residuals. The histogram of the residual images show large peaks below the clean cutoff.

The primary difference that natural and uniform weightings have on CLEAN is that uniform weighting suppresses the synthesised beam sidelobes. While this lowers the sensitivity of the observation, CLEAN then performs better at modelling fine structure with CLEAN components, with diffuse structure left in the residuals, which are then added back in the CLEAN restored image.

The RMS of the residuals around the scientific region of interest (see Table 4.4) show that PURIFY consistently fits the measurements better than CLEAN.

Table 4.4 compares the RMS of the residual images with in the regions shown in Figures 4.6, 4.7, 4.8, 4.9. Other than 3C129, PURIFY shows a consistent order of magnitude improvement in the RMS of the residuals.

With this comparison of PURIFY on real data, we have used a standard CLEAN algorithm. We make it clear this is not necessarily the best performance of the CLEAN algorithm and its variants. Choosing a different CLEAN cutoff, masking around sources, and including multi-scale components can produce better quality CLEAN component models. These configurations can improve the quality of the component map, leave less structures in the residual map, and improve the aesthetics of the restored map.

4.4.3 Discussion

From a scientific standpoint, the PURIFY models show more structure than those recovered by CLEAN. This is clear when looking at the surface brightness variation of the lobes of 3C129 and Cygnus A. For 3C129 and Cygnus A, unlike the CLEAN restored images, the surface brightness structure is well resolved in the images recovered by PURIFY.

The CLEAN restored images of PKS J0334-39 and PKS J0116-473 with uniform weighting show an improvement over natural weighting for deconvolving the fine structure, as well as containing diffuse structure.

However, uniform weighting is known to suppress large scale structure, and lowers the sensitivity of the observation (as discussed in [118]). However, PURIFY has the ability to reconstruct the fine details of PKS J0334-39 and PKS J0116-473 without uniform weighting. This demonstrates that PURIFY has the potential to reconstruct observations that can be used to perform a more detailed analysis of morphology and structure of diffuse sources. The reconstruction of Cygnus A shows that it is possible to accurately reconstruct diffuse bright structures in the presence of compact bright sources.

Modeling extended structure accurately is particularly important for understanding the underlying physics of radio sources and their environment. Bent tailed radio galaxies, such as 3C129, are a example of where this is important [119]. The morphology of bent tailed radio galaxies can be used as a probe of their local cluster environment [120, 121, 122, 123, 108, 124].

Additionally, an important class of diffuse, low surface brightness radio sources are cluster relics and halos (*e.g.* [125, 118, 126, 127]), which are believed to be caused by shocks and turbulence in the outskirts of galaxy clusters [128, 129]. Radio halos and relics are not well understood, and they are prime examples of sources with diffuse low surface brightness structure that relates to the physics within the intra-cluster medium and merging galaxy clusters. However, galaxy clusters often contain bright compact sources, providing a challenge in deconvolving low surface brightness sources.

4.5 Conclusion

In this chapter we have further developed the PURIFY software package so that it can be easily applied to observational data from radio interferometric telescopes. PURIFY has been completely redesigned and reimplemented in C++ and now supports the ADMM algorithm developed recently by [30]. Furthermore, the capabilities of convolutional degridding in the measurement operator have been expanded.

Table 4.3: Table listing details of settings used to recover CLEAN images.

Observation	Weighting	Beam Size	Cutoff Jy/Beam	Peak Value Jy/Beam
3C129	Natural	$2.07'' \times 1.88''$, 158°	0.0025	0.050
	Uniform	$1.30'' \times 1.06''$, 33°		0.055
Cygnus A	Natural	$3.48'' \times 2.81''$, 105°	0.1	21.9
	Uniform	$2.25'' \times 1.99''$, 97.4°		16
PKS J0334-39	Natural	$45.6'' \times 36.8''$, 171°	0.001	0.2
	Uniform	$8.6'' \times 4.3''$, 17°		0.09
PKS J0116-473	Natural	$40.0'' \times 24.6''$, 158°	0.001	0.13
	Uniform	$6.33'' \times 4.72''$, 3°		0.086

Table 4.4: Table listing the root-mean-squared of each reconstruction (units are in mJy/Beam).

Observation	PURIFY	CLEAN (natural)	CLEAN (uniform)
3C129	0.10	0.23	0.11
Cygnus A	6.1	59	36
PKS J0334-39	0.052	1.00	0.37
PKS J0116-473	0.054	0.88	0.24

Using simulations we studied the impact of a number of different interpolation kernels on the quality of images recovered by sparse reconstruction approaches to interferometric imaging. The Kaiser-Bessel kernel was found to perform very well—as well as other optimal kernels—while requiring a smaller support size, thereby reducing computation cost, and having an analytic expression that can be evaluated easily and efficiently.

PURIFY was applied to observational data from the VLA and ATCA telescopes, recovering high-quality interferometric images superior to those recovered by CLEAN. Firstly, the PURIFY residuals contain less extended structure and are more Gaussian with a lower RMS. Secondly, the model images recovered by PURIFY are of sufficient quality that there is no need to perform any post-processing as is done for CLEAN (such as restoring the image). On visual inspection, the images recovered by PURIFY reveal extended structure in greater detail. For example, in reconstructed images

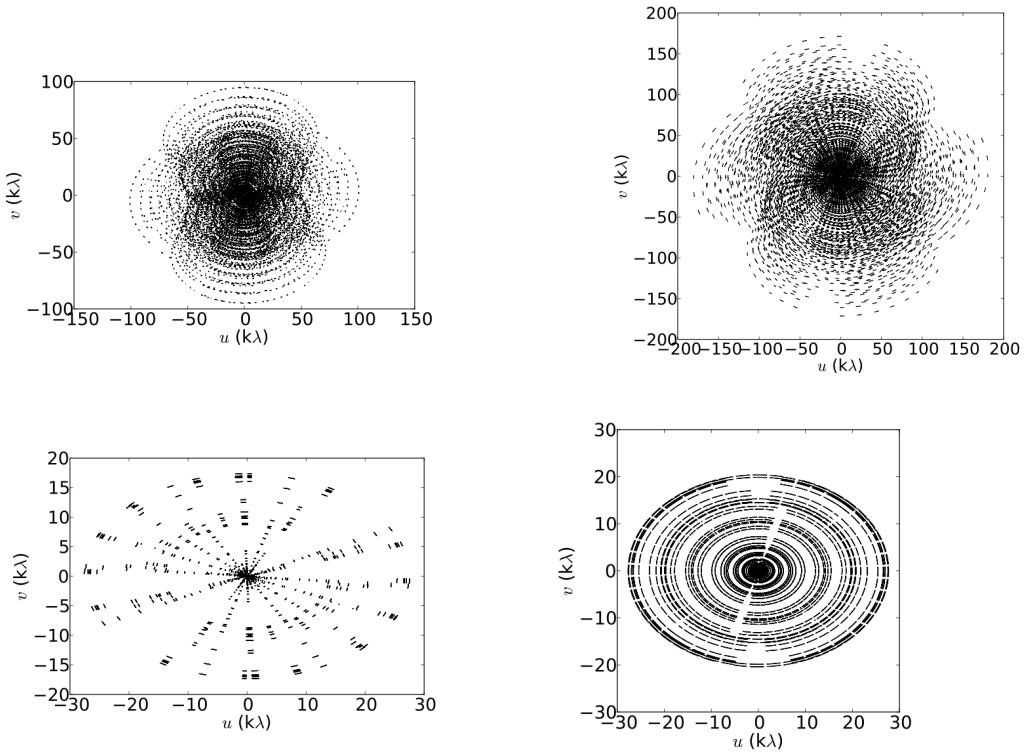

Figure 4.5: Plots showing the uv-coverage of the observations of 3C129 (top left), Cygnus A (top right), PKS J0334-39 (bottom left), and PKS J0116-473 (bottom right). Units of u and v are kilo-wavelengths (kilo-λ).

of 3C129 the internal structure of the radio jets is much more apparent (Figure 4.6). However, we have kept to using a traditional bare bones CLEAN algorithm, using multi-scale CLEAN and other features would provide a better comparison against restored maps. Also comparing both methods on data with calibration errors will change the performance of the PURIFY and CLEAN methods. But it is clear that the reconstruction quality from PURIFY does not need image restoration which is important for facilitating a better scientific understanding of astrophysical processes.

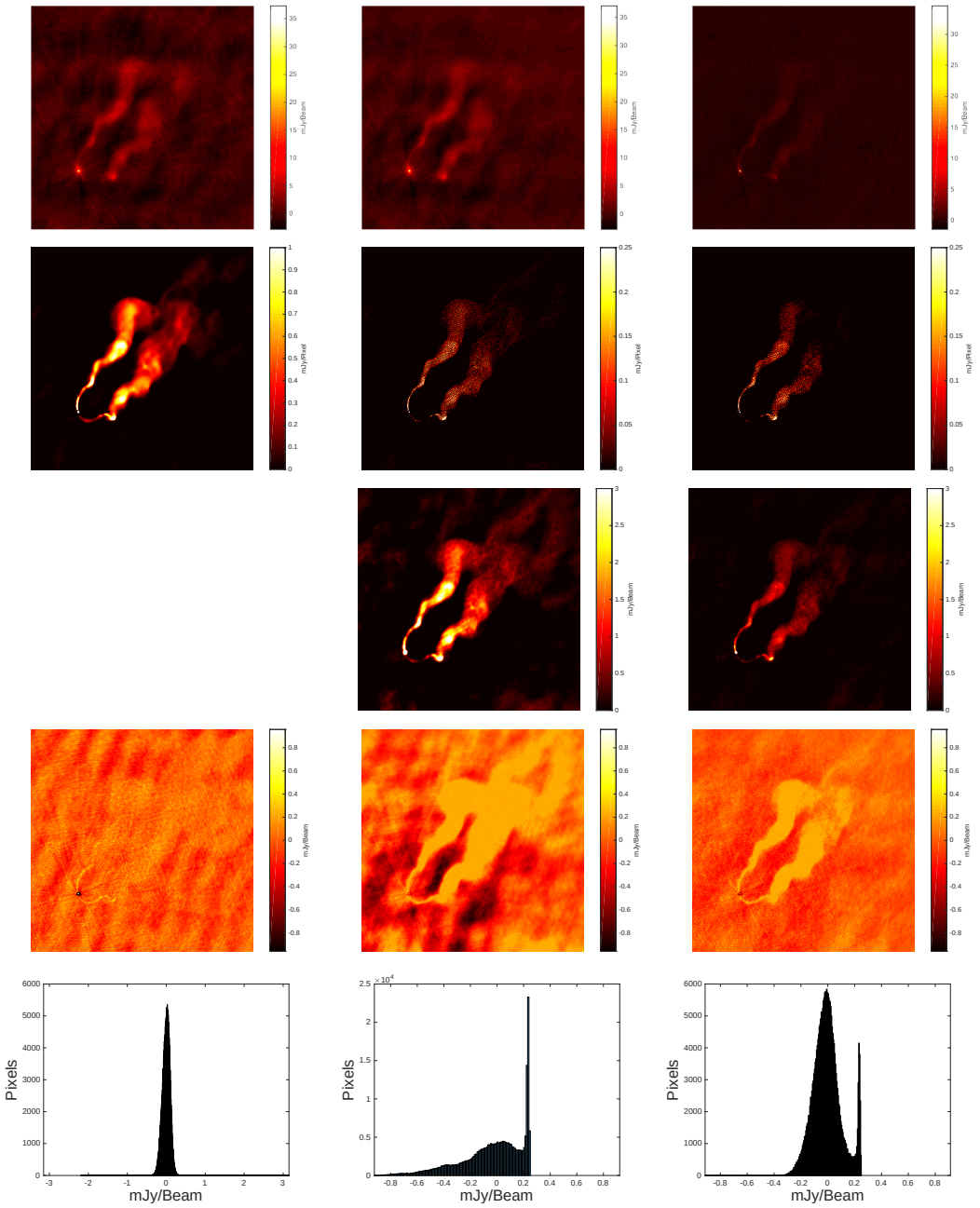

Figure 4.6: PURIFY and CLEAN reconstructions of 3C129. Each pixel is 0.4 arcseconds, and the images are 1024×1024 pixels. The pixels within $[400, 900] \times [400, 900]$ are shown in the images and histogram of this figure. Left column shows a PURIFY reconstruction with natural weighting. Middle and right columns show CLEAN reconstructions with natural and uniform weightings, respectively. From the top to bottom row: synthesised (*i.e.* dirty) image, model image, restored image, residual image, and a histogram of residual image. PURIFY does not require any post-processing and so does not produce a restored image.

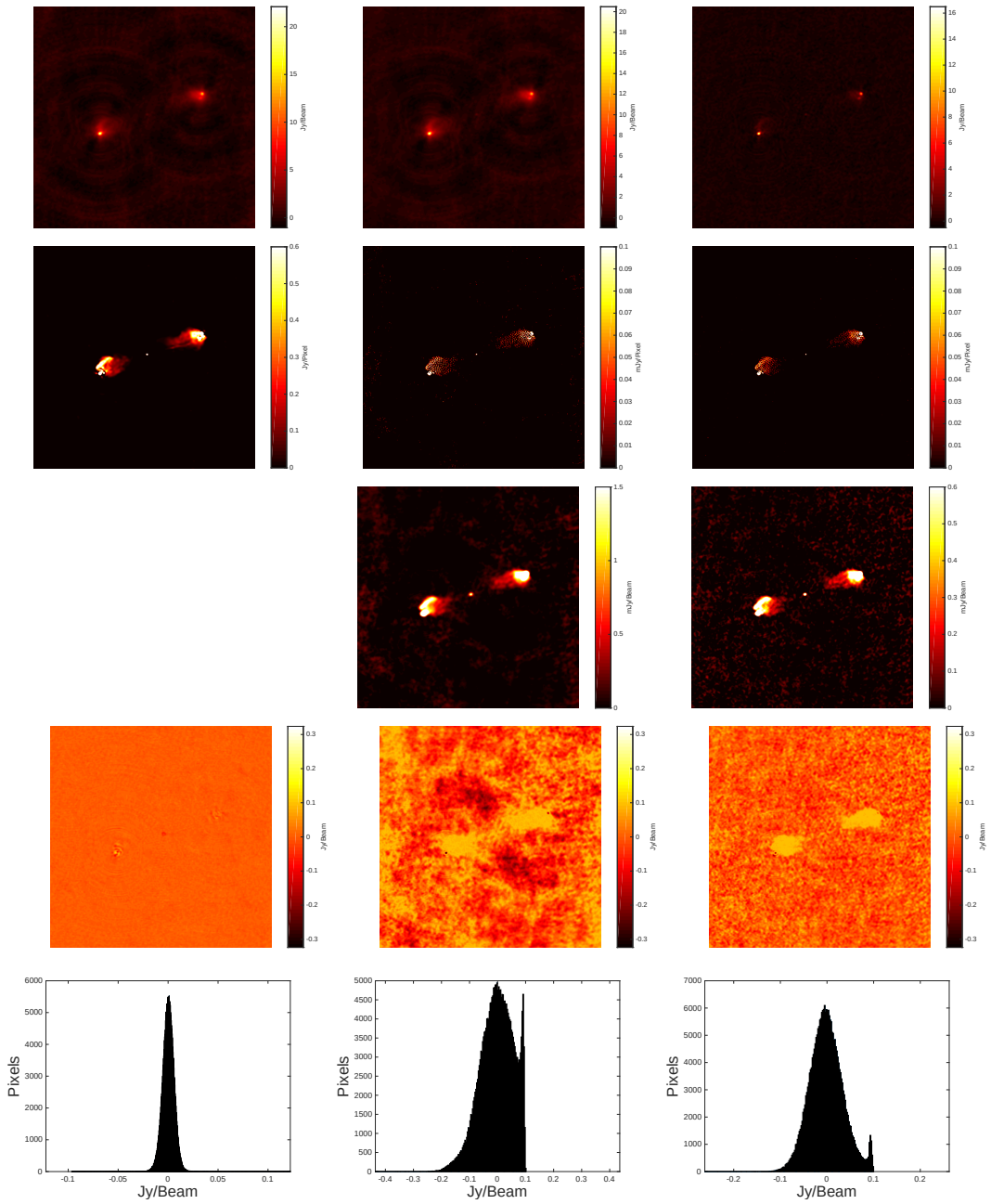

Figure 4.7: PURIFY and CLEAN reconstructions of Cygnus A. Each pixel is 0.5 arcseconds, and the images are 1024×1024 pixels. The pixels within $[256, 756] \times [256, 756]$ are shown in the images and histogram of this figure. Left column shows a PURIFY reconstruction with natural weighting. Middle and right columns show CLEAN reconstructions with natural and uniform weightings, respectively. From the top to bottom row: synthesised (*i.e.* dirty) image, model image, restored image, residual image, and a histogram of residual image. PURIFY does not require any post-processing and so does not produce a restored image.

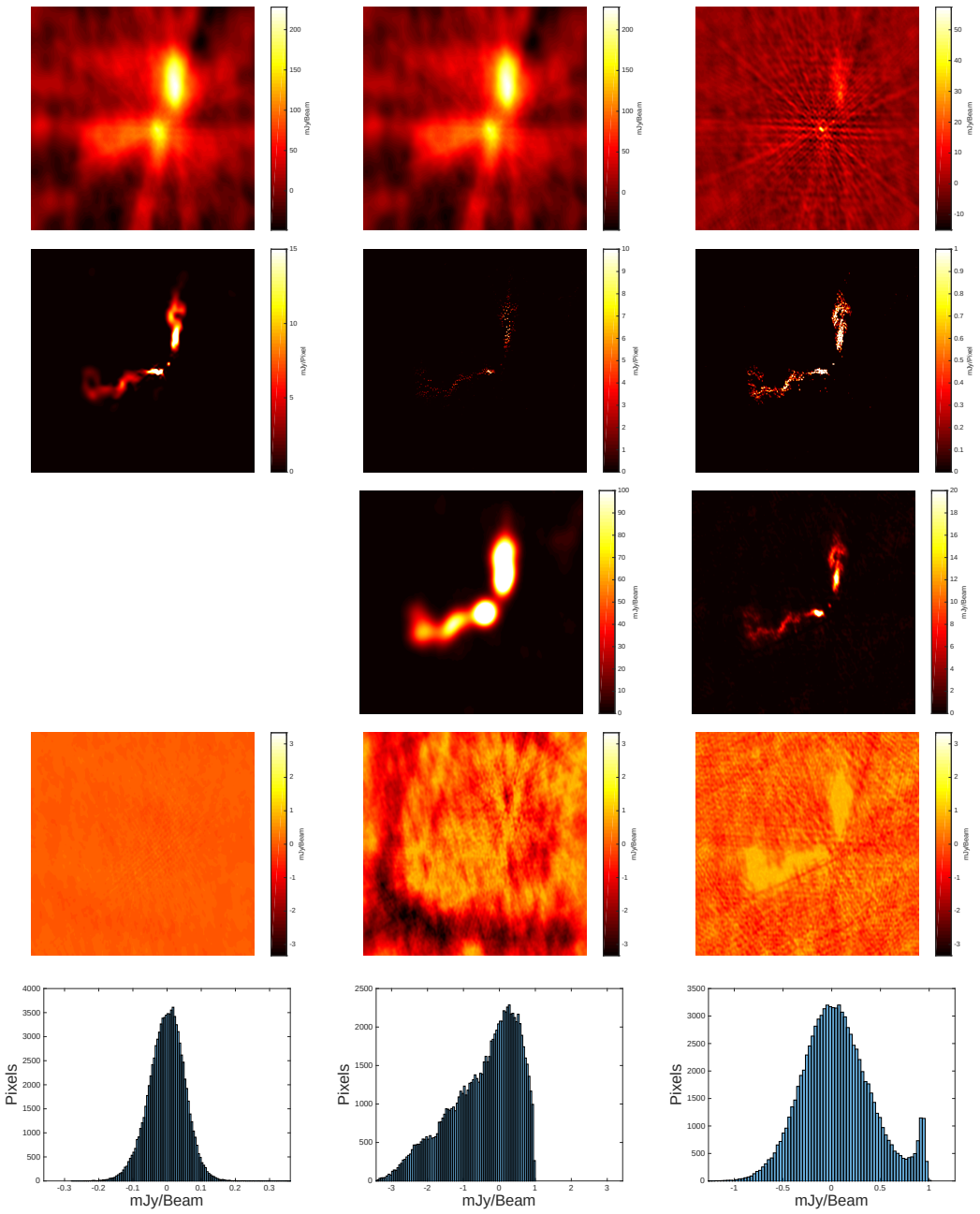

Figure 4.8: PURIFY and CLEAN reconstructions of PKS J0334-39. Each pixel is 2 arcseconds, and the images are 2048×2048 pixels. The pixels within $[862, 1162] \times [862, 1162]$ are shown in the images and histogram of this figure. Left column shows a PURIFY reconstruction with natural weighting. Middle and right columns show CLEAN reconstructions with natural and uniform weightings, respectively. From the top to bottom row: synthesised (*i.e.* dirty) image, model image, restored image, residual image, and a histogram of residual image. PURIFY does not require any post-processing and so does not produce a restored image.

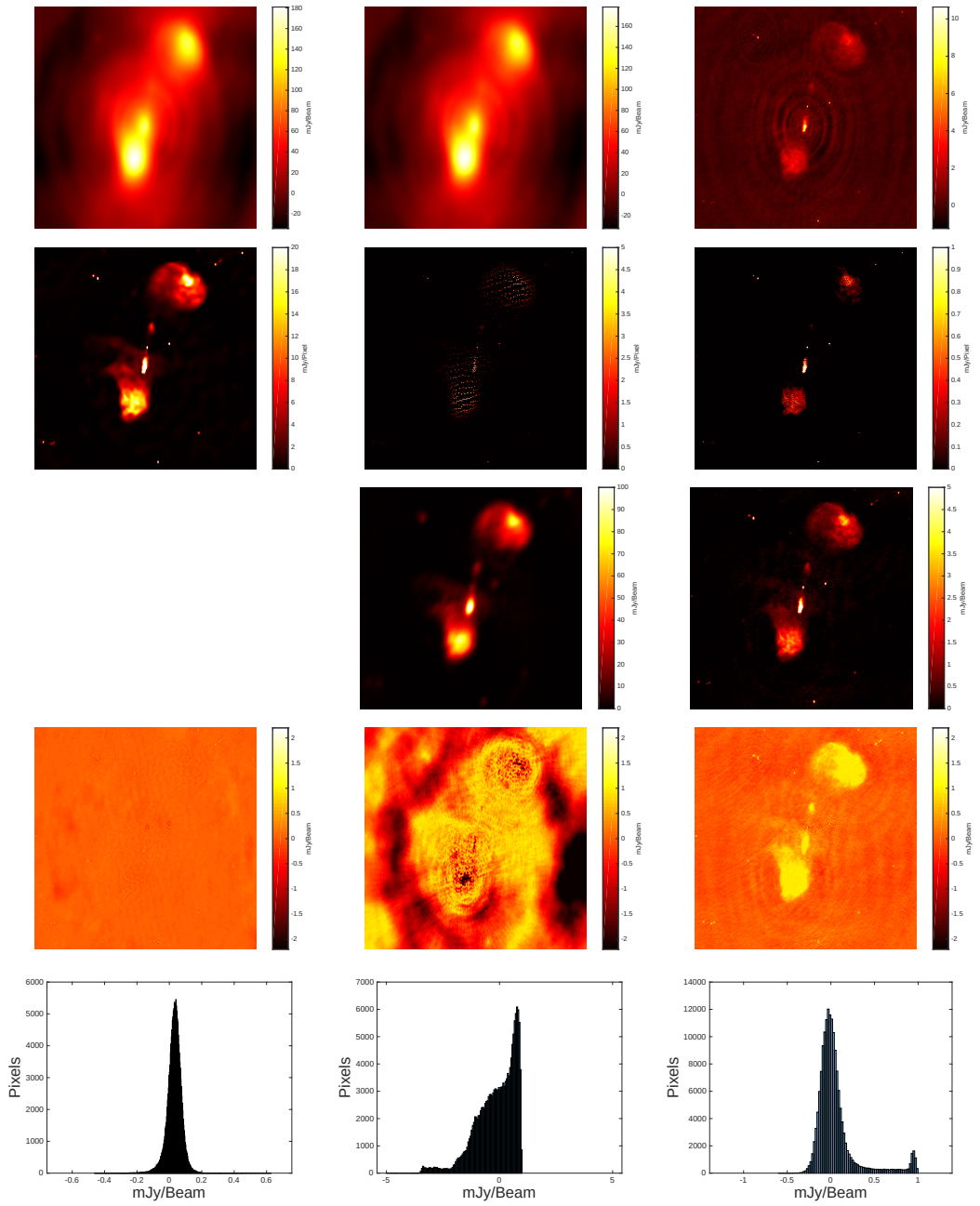

Figure 4.9: PURIFY and CLEAN reconstructions of PKS J0116-473. Each pixel is 2.4 arcseconds, and the images are 2048×2048 pixels. The pixels within $[800, 1200] \times [800, 1200]$ are shown in the images and histogram of this figure. Left column shows a PURIFY reconstruction with natural weighting. Middle and right columns show CLEAN reconstructions with natural and uniform weightings, respectively. From the top to bottom row: synthesised (*i.e.* dirty) image, model image, restored image, residual image, and a histogram of residual image. PURIFY does not require any post-processing and so does not produce a restored image.

Chapter 5

Distributed Forward-Backward ADMM

In this chapter we present new distributed big data sparse image reconstruction algorithms which have been implemented in the PURIFY (3.0.1) (`https://github.com/astro-informatics/purify`) and SOPT (3.0.1) (`https://github.com/astro-informatics/sopt`) software packages. These algorithms make use of degridding and gridding, wavelet transforms, and proximal operators to reconstruct high quality images of the radio sky while communicating data between compute nodes of a computing cluster using the Message Passing Interface (MPI). We distribute the data over a computing cluster to accommodate the large volume of measurements. We use multi-threaded parallelization on a Graphics Processing Unit (GPU) or via OpenMP to parallelize across cores of a CPU node. We show that the MPI distributed framework reduces the time it takes to compute an iteration, increase the volumes of data that can be included in image reconstruction, and can be used in connection with multi-threaded parallelization such as GPUs and OpenMP for further optimization.

In Section 5.1 we introduce the serial Dual Forward-Backward based Alternating Direction Method of Multipliers (ADMM) algorithm implemented in PURIFY. This sets the ground work for introducing computationally distributed wavelet and measurement operators and distributed ADMM

algorithm in Section 5.2. We demonstrate the implementations of the distributed algorithms in PURIFY in Sections 5.3 and 5.4 and end with a conclusion in Section 5.5.

5.1 Sparse Regularization using Dual Forward-Backward ADMM

As mentioned in (3.17), the standard constrained radio interferometry solution with ℓ_1 (sparse) regularization can be stated as

$$\boldsymbol{x}^\star = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_{\ell_1} + \iota_{\mathcal{C}}(\boldsymbol{x}) + \iota_{\mathcal{B}^{\varepsilon}_{\ell_2}(\boldsymbol{y})}(\boldsymbol{\Phi}\boldsymbol{x}) \right\}, \tag{5.1}$$

with $\mathcal{B}^{\varepsilon}_{\ell_2}(\boldsymbol{y}) = \{\boldsymbol{z} \in \mathbb{C}^M : \|\boldsymbol{z} - \boldsymbol{y}\|_{\ell_2} \leq \varepsilon\}$ being the set that satisfies the fidelity constraint and $\mathcal{C} = \mathbb{R}^N_+$ is the set that represents the positive and real constraint.

Let $\boldsymbol{r}$ be the slack variable with the constraint $\boldsymbol{r} = \boldsymbol{\Phi}\boldsymbol{x}$. As described in Section 3.3.3, to solve the above problem (5.1), ADMM can be applied by minimizing the Lagrangian of problem (5.1) corresponding to $\boldsymbol{x}$ and $\boldsymbol{r}$ alternatively, i.e.,

$$\min_{\boldsymbol{x}} \left\{ \mu \left[\|\boldsymbol{\Psi}^\dagger \boldsymbol{x}\|_{\ell_1} + \iota_{\mathcal{C}}(\boldsymbol{x}) \right] + \frac{1}{2} \left\| \boldsymbol{\Phi}\boldsymbol{x} - (\boldsymbol{r} - \boldsymbol{s}) \right\|^2_{\ell_2} \right\}, \tag{5.2}$$

$$\min_{\boldsymbol{r}} \left\{ \mu \left[\iota_{\mathcal{B}^{\varepsilon}_{\ell_2}(\boldsymbol{y})}(\boldsymbol{r}) \right] + \frac{1}{2} \left\| \boldsymbol{r} - (\boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{s}) \right\|^2_{\ell_2} \right\}, \tag{5.3}$$

where $\boldsymbol{s}$ represents the Lagrangian multiplier. Algorithm 5 shows the Dual Forward-Backward ADMM algorithm used to solve problem (5.1). Recall that it is the same as the standard ADMM algorithm, but uses Dual Forward-Backward splitting with a Forward-Backward step to minimize the subproblem (5.2). The distributed version of this algorithm is presented in [30]. The serial version of this algorithm has been implemented in PURIFY 2.0.0 and applied in [1] to simulated and real observations from radio interferometric telescopes previously.

Algorithm 5 Dual Forward-Backward ADMM.
The Dual Forward-Backward ADMM algorithm without MPI implementation. Lines 3–4 evaluate the ℓ_2-ball proximal operator (constraining the solution to the ℓ_2-ball), which is to address the solution of the subproblem (5.3). Line 5 is the Lagrangian dual variable update, connecting the two minimization problems (5.2) and (5.3). Lines 6–7 are a Forward-Backward step, which is to address the solution of the subproblem (5.2); particularly, line 6 is the forward (gradient) step, and line 7 is the backward step which is solved using the Dual Forward-Backward algorithm, as described between lines 9–16.

1: **given** $\boldsymbol{x}^{(0)}, \boldsymbol{r}^{(0)}, \boldsymbol{s}^{(0)}, \boldsymbol{q}^{(0)}, \gamma, \rho, \varrho$
2: **repeat for** $t = 1, \dots$
3: $\quad \boldsymbol{v}^{(t)} = \boldsymbol{\Phi}\boldsymbol{x}^{(t-1)}$
4: $\quad \boldsymbol{r}^{(t)} = \mathcal{P}^{\varepsilon}_{\mathcal{B}}\left(\boldsymbol{v}^{(t)} + \boldsymbol{s}^{(t-1)}\right)$
5: $\quad \boldsymbol{s}^{(t)} = \boldsymbol{s}^{(t-1)} + \varrho\left(\boldsymbol{v}^{(t)} - \boldsymbol{r}^{(t)}\right)$
6: $\quad \tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t-1)} - \rho\boldsymbol{\Phi}^{\dagger}\left(\boldsymbol{v}^{(t)} - \boldsymbol{r}^{(t)} + \boldsymbol{s}^{(t)}\right)$
7: $\quad \boldsymbol{x}^{(t)} = \textsc{DualFB}(\tilde{\boldsymbol{x}}^{(t)}, \gamma)$
8: **until convergence**
9: **function** $\textsc{DualFB}(\boldsymbol{z}, \gamma)$
10: $\quad$ **given** $\boldsymbol{d}_j^{(0)}, \eta$
11: $\quad \bar{\boldsymbol{z}}^{(0)} = \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{z}\right)$
12: $\quad$ **repeat for** $k = 1, \dots$
13: $\quad\quad \boldsymbol{d}^{(k)} = \frac{1}{\eta}\left(\mathcal{I} - \mathcal{S}_{\gamma}\right)\left(\eta\boldsymbol{d}^{(k-1)} + \boldsymbol{\Psi}^{\dagger}\bar{\boldsymbol{z}}^{(k-1)}\right)$
14: $\quad\quad \bar{\boldsymbol{z}}^{(k)} = \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{z}^{(k-1)} - \boldsymbol{\Psi}\boldsymbol{d}^{(k)}\right)$
15: $\quad$ **until convergence**
16: **return** $\bar{\boldsymbol{z}}^{(k)}$

5.2 Distributed Dual Forward-Backward ADMM

In the previous chapter, we covered serial proximal optimization algorithms and serial operators. It is well known that these algorithms can be distributed (see [93, 89, 30] and references therein).

In this section, we describe the details for how to modify these algorithms to be distributed over a computing cluster using the standard commonly known as MPI. For clarity, we describe MPI implementations of operators in PURIFY and SOPT. The measurements and MPI processes are distributed across the nodes of a computing cluster.

5.2.1 MPI Framework

The MPI standard is a framework where multiple process of the same program are run concurrently, communicating data and variables at sync points. This is commonly referred to as distributed memory parallelism. There are many independent processes (nodes) with their own data, but they can send messages containing data between them. This is different from the more typical shared memory parallelism, where a single process has access to all the data, but executes multiple threads for sections of the program (such as a loop with independent iterations). However, hybrids of shared and distributed memory parallelism are not uncommon, where nodes on a computing cluster send messages while performing multi-threaded operations. Please see [130] for a formal reference on MPI[1].

The MPI framework contains a total number of process n_d, each with a rank $0 \leq j < n_d$, all connected by a communicator for sending and receiving data. The most basic methods of a communicator consist of send and receive operations between individual processes. However, typically sending and receiving is performed in collective send and receive operations:

Broadcast (one to many) – Send a copy of a variable (scalar or array) from the root node to all nodes.

Scatter (one to many) – Scatter is where a root process contains an array; different sections of this array are sent to different nodes. The root process does not keep the sent data.

Gather (many to one) – Gather is where the root process receives data from all nodes. This could be sections of an array, or variables that are combined into an array on the root process.

All to All (many to many) – All to all is where data is communicated between all nodes at once. Each process sends and receives. This could be single variables or sections of arrays.

Reduce (many to one) – Reduce, or performing a reduction, is where

[1]Official versions of the MPI standard can be found online at `https://www.mpi-forum.org/docs/`.

a binary operation (assumed to be associative and commutative) is efficiently performed with a variable or array over the cluster with the result sent to the root process. Summation of variables across nodes is a common example of this. However, logical operations and max/min operations are also common.

All reduce (many to many) – All reduce is equivalent to a reduction, but the result is broadcasted to all nodes from the root process. All reduce with summation is called an all sum all operation.

The operation to broadcast a copy of $\boldsymbol{x}$ onto each node can be represented by the linear operation

$$\begin{bmatrix} \boldsymbol{I}_1 \\ \vdots \\ \boldsymbol{I}_{n_\mathrm{d}} \end{bmatrix} \boldsymbol{x} \tag{5.4}$$

where $\boldsymbol{I}_j$ is an $N \times N$ identity matrix. The adjoint of this operation is a reduction

$$\boldsymbol{x}_\mathrm{sum} = \begin{bmatrix} \boldsymbol{I}_1 & \dots & \boldsymbol{I}_{n_\mathrm{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{n_\mathrm{d}} \end{bmatrix}. \tag{5.5}$$

It is possible to view other MPI operations of sending data between nodes in the context of linear mappings.

5.2.2 Distributed Visibilities

The visibilities can be loaded on a root process then sorted into groups that are scattered to each node. This process splits and sorts the measurement vector $\boldsymbol{y}$ into groups $\boldsymbol{y}_j$, where j is the rank of a process:

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\mathrm{d}} \end{bmatrix}. \tag{5.6}$$

In this work, we sort the visibilities $\boldsymbol{y}$ via ordering them by baseline length and dividing $\boldsymbol{y}$ into sections of equal size $\boldsymbol{y}_j$ to be scattered to each node.

However, it is also possible to have each MPI process to read a different set of measurements. In principle, the weights and uvw coordinates are scattered with the visibilities.

If there is too much data to load the measurements onto one node, the data can be loaded in sections and then scattered to each node. After the data has been distributed, sorting into groups can be done using logical reductions, and then distributed to each node using an all to all operation. This has been done with the w-stacking algorithm in [2].

5.2.3 Distributed Measurement Operator

For each group of visibilities $\boldsymbol{y}_j$ on node j, there is a corresponding measurement operator $\boldsymbol{\Phi}_j$. However, there are many ways to relate $\boldsymbol{\Phi}_j$ to the measurement operator for $\boldsymbol{y}$, $\boldsymbol{\Phi}$; we show two examples.

5.2.3.1 Distributed Images

We can relate the MPI measurement operator to the serial operators by

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Phi}_{n_\mathrm{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_1 \\ \vdots \\ \boldsymbol{I}_{n_\mathrm{d}} \end{bmatrix}. \tag{5.7}$$

The forward operator can be expressed simply as independent measurement operators applied in parallel after broadcasting $\boldsymbol{x}$:

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Phi}_{n_\mathrm{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_1 \\ \vdots \\ \boldsymbol{I}_{n_\mathrm{d}} \end{bmatrix} \boldsymbol{x}. \tag{5.8}$$

The adjoint operator can be expressed as the adjoint of independent

measurement operators applied in parallel, followed by a reduction

$$\boldsymbol{x}_{\text{dirty}} = \begin{bmatrix} \boldsymbol{I}_1 & \dots & \boldsymbol{I}_{n_\text{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1^\dagger & & \\ & \ddots & \\ & & \boldsymbol{\Phi}_{n_\text{d}}^\dagger \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\text{d}} \end{bmatrix}. \tag{5.9}$$

However, with the MPI framework, it is efficient to always have a copy of the same image on each node so that other image domain operations can be performed in parallel (i.e. wavelet transforms). This can be ensured by combining the broadcast and reduction in a single all sum all operation during the adjoint. We work with the forward operator that applies each measurement operator independently on each node, with a copy of $\boldsymbol{x}$ located on each node

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\text{d}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Phi}_{n_\text{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \vdots \\ \boldsymbol{x} \end{bmatrix}, \tag{5.10}$$

and the adjoint operation can be performed by applying the adjoint of each measurement operator independently followed by an all sum all

$$\begin{bmatrix} \boldsymbol{x}_{\text{dirty}} \\ \vdots \\ \boldsymbol{x}_{\text{dirty}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_1 \\ \vdots \\ \boldsymbol{I}_{n_\text{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_1 & \dots & \boldsymbol{I}_{n_\text{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1^\dagger & & \\ & \ddots & \\ & & \boldsymbol{\Phi}_{n_\text{d}}^\dagger \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\text{d}} \end{bmatrix}. \tag{5.11}$$

We can normalize the operator with the operator norm, by using the power method to estimate the largest eigenvalue, and remove arbitrary scaling due to n_d and other normalization factors.

5.2.3.2 Distributed FFT Grid Sections

Another method, which is discussed in [30], is to distribute the grid points of the FFT grid, where the degridding can be performed on each node. This can be performed using a scatter and gather operation from a root process. We can define the operation of distributing the necessary grid points using the

operators $\boldsymbol{M}_j \in \mathbb{R}^{B_j \times 2N}$, where B_j is the number of non zero columns of $\boldsymbol{G}_j$. Additionally, we can remove the zero columns of $\boldsymbol{G}_j$, such that $\boldsymbol{G}_j \in \mathbb{R}^{M_j \times B_j}$.

The measurement operator is defined by

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{n_\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_1\boldsymbol{G}_1 & & \\ & \ddots & \\ & & \boldsymbol{W}_{n_\mathrm{d}}\boldsymbol{G}_{n_\mathrm{d}} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_1 \\ \vdots \\ \boldsymbol{M}_{n_\mathrm{d}} \end{bmatrix} \boldsymbol{F}\boldsymbol{Z}\boldsymbol{S}\boldsymbol{x}\,. \tag{5.12}$$

$[\boldsymbol{M}_1^\top, \cdots, \boldsymbol{M}_{n_\mathrm{d}}^\top]^\top$ can be seen as scattering the FFT grid points from the root process to the other nodes. The adjoint can be seen as gathering and summing gridded FFT grid points to the root process. While this method appears to reduce communication, this has the disadvantage that the result of the adjoint ends up only on the root process. In practice, this means a broadcast is eventually required after the adjoint of this measurement operator so that further image domain operations can be performed in parallel.

5.2.4 Distributed Wavelet Operator

The MPI wavelet operator can be distributed for each wavelet basis in the dictionary. Using the convention that $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$, each wavelet representation can be arranged as

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_{n_\mathrm{w}} \end{bmatrix}, \tag{5.13}$$

for n_w wavelet transforms. From this definition, it follows that each inverse transform is performed independently with a reduction at the end

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{I}_1 & \dots & \boldsymbol{I}_{n_\mathrm{w}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Psi}_{n_\mathrm{w}} \end{bmatrix}. \tag{5.14}$$

However, like with the distributed image measurement operator, we combine the reduction and broadcasting as an all sum all. In practice, we use the

forward operation

$$\begin{bmatrix} \boldsymbol{x} \\ \vdots \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_1 \\ \vdots \\ \boldsymbol{I}_{n_\mathrm{w}} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_1 & \dots & \boldsymbol{I}_{n_\mathrm{w}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Psi}_{n_\mathrm{w}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_{n_\mathrm{w}} \end{bmatrix}. \tag{5.15}$$

The adjoint operation is

$$\begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_{n_\mathrm{d}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Psi}_1^\dagger & & \\ & \ddots & \\ & & \boldsymbol{\Psi}_{n_\mathrm{w}}^\dagger \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \vdots \\ \boldsymbol{x} \end{bmatrix}. \tag{5.16}$$

5.2.5 Distributed Proximal Operator

The proximal operators for the ℓ_1-norm, ℓ_2-ball, and convergence criteria may require communication between nodes, which is discussed in this section.

5.2.5.1 Sparsity and Positivity Constraint

The ℓ_1-proximal norm does not need a communicator in itself. However, $\boldsymbol{\Psi}$ contains more than one wavelet transform. The proximal operator for the ℓ_1-norm is solved iteratively using the Dual Forward-Backward method. The objective function that proximal operator minimizes can be computed to check that the iterations have converged. For a given $\boldsymbol{x}$, the proximal operator returns

$$\underset{\boldsymbol{z}}{\operatorname{argmin}} \left[\iota_{\mathcal{C}}(\boldsymbol{z}) + \|\boldsymbol{\Psi}^\dagger \boldsymbol{z}\|_{\ell_1} + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{z}\|_{\ell_2} \right]. \tag{5.17}$$

To assert that the Dual Forward-Backward method has converged to a minimum when calculating the proximal operator requires checking the variation of the ℓ_1-norm; calculating the ℓ_1-norm requires an MPI all sum all operation over wavelet coefficients. Another assertion that can be made is that the relative variation of $\boldsymbol{x}$ is close to zero, which requires no communication.

5.2.5.2 Fidelity Constraint

In the constrained minimization problem, the solution is constrained to be within the ℓ_2-ball through the proximal operator $\text{prox}_{\mathcal{B}^{\varepsilon}_{\ell_2}(\boldsymbol{y})}(\boldsymbol{v})$. However, this proximal operator requires calculating the ℓ_2-norm of the residuals $\|\boldsymbol{v}-\boldsymbol{y}\|_{\ell_2}$. When the visibilities are distributed on each node $\boldsymbol{y}_i$, this calculation requires an all sum all.

However, if each node constrains the solution to an independent local ℓ_2-ball using $\text{prox}_{\mathcal{B}^{\varepsilon_j}_{\ell_2}(\boldsymbol{y}_j)}(\boldsymbol{v}_j)$ with radius ε_j, where $\varepsilon = \sqrt{\sum_{j=1}^{n_d} \varepsilon_j^2}$. The local ℓ_2-ball solution will also lie within the global ℓ_2-ball where we have used $\text{prox}_{\mathcal{B}^{\varepsilon}_{\ell_2}(\boldsymbol{y})}(\boldsymbol{v})$, which can be shown using the triangle inequality. This requires less communication (introducing a new ε_j for each node is suggested [30]). However, the communication overhead of calculating a distributed ℓ_2-norm is negligible compared to communicating entire images. Furthermore, using the global ℓ_2-ball is more robust in convergence rate as it is independent of how the measurements are grouped across the nodes.

5.2.6 Distributed Convergence

There are multiple methods that can be used to check that the solution has converged. For example, when the relative difference of the solution between iterations is small, i.e. $\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(i-1)}\|_{\ell_2}/\|\boldsymbol{x}^{(i)}\|_{\ell_2} < \delta$ for a small δ; when the relative difference of the objective function between iterations is small; and the condition that the residuals of the solution lie within the ℓ_2-ball[2]. These convergence criteria need to be communicated across the nodes. The convergence criteria need to be chosen carefully, since the quality of the output image can be degraded if the iterations have not converged sufficiently.

5.2.7 Distributed ADMM

With PURIFY, we build on the previous sections and combine the MPI distributed linear operators and proximal operators to solve the radio interferometric imaging inverse problem. The previous section discusses how

[2]A feature of ADMM is that it will not ensure that the residuals lie in the ℓ_2-ball for each iteration but it will converge to this condition.

to distribute $\boldsymbol{\Phi}$ for distributed $\boldsymbol{y}_j$, and how to distribute $\boldsymbol{\Psi}$ for distributed wavelet coefficients. In Algorithms 6 and 7, we outline MPI algorithms that use two variations of the measurement operator. Algorithm 6 uses an all sum all in the adjoint of the measurement operator following Section 5.2.3.1 and Algorithm 7 performs an FFT on the root node distributing parts of the grid following Section 5.2.3.2. In practice we recommend using Algorithm 6 as it can be easily modified to efficiently model wide-field effects, as demonstrated in [2]. Furthermore, Algorithm 6 is simpler to implement.

5.2.8 Global Fidelity Constraint ADMM

When the measurements are spread across the various nodes, communication is required to ensure that the same ℓ_2-ball constraint is enforced across all measurements. The proximal operator for the ℓ_2-ball is

$$\mathcal{P}^{\varepsilon}_{\mathcal{B}}(\boldsymbol{z}_j) \triangleq \begin{cases} \varepsilon \frac{\boldsymbol{z}_j - \boldsymbol{y}_j}{\sqrt{\text{AllSumAll}_j(\|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2}^2)}} + \boldsymbol{y}_j & \sqrt{\text{AllSumAll}_j(\|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2}^2)} > \varepsilon \\ \boldsymbol{z}_j & \sqrt{\text{AllSumAll}_j(\|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2}^2)} \leq \varepsilon \end{cases}. \tag{5.18}$$

5.2.9 Local Fidelity Constraint ADMM

We can split the single ℓ_2-ball into many, and restate a new constrained problem, i.e.,

$$\boldsymbol{x}^{\star} = \underset{\boldsymbol{x}}{\text{argmin}} \left\{ \|\boldsymbol{\Psi}^{\dagger}\boldsymbol{x}\|_{\ell_1} + \iota_{\mathcal{C}}(\boldsymbol{x}) + \sum_{j=1}^{n_{\text{d}}} \iota_{\mathcal{B}_j}(\boldsymbol{\Phi}_j \boldsymbol{x}) \right\}. \tag{5.19}$$

In particular, the alternating minimization involving the slack variable $\boldsymbol{r}$ is split into solving each $\boldsymbol{r}_j$ independently

$$\min_{\boldsymbol{r}_j} \left\{ \mu \left[\iota_{\mathcal{B}_j}(\boldsymbol{r}_j) \right] + \frac{1}{2} \left\| \boldsymbol{r}_j - \boldsymbol{\Phi}_j \boldsymbol{x} - \boldsymbol{s}_j \right\|_{\ell_2}^2 \right\}. \tag{5.20}$$

Algorithm 6 Distributed Image (Dual Forward-Backward ADMM):
Every node has access to a global ℓ_2-ball proximal and a serial version of the measurement operator $\boldsymbol{\Phi}_j$. After the adjoint of the measurement operator is applied, an AllSumAll is performed over the returned image of each node j, then each node has the combined image. An AllSumAll is also used after the forward wavelet operator $\boldsymbol{\Psi}_j$. Communication is needed in calculation of $\mathcal{P}_{\mathcal{B}}^{\varepsilon}$ with an AllSumAll in the ℓ_2-norm of the residuals. Using instead $\mathcal{P}_{\mathcal{B}_j}^{\varepsilon_j}$ removes this communication overhead but changes the minimization problem.

given $\boldsymbol{x}^{(0)}, \boldsymbol{r}_j^{(0)}, \boldsymbol{s}_j^{(0)}, \boldsymbol{q}_j^{(0)}, \gamma, \rho, \varrho$
repeat for $t = 1, \ldots$
  $\forall j \in \{1, \ldots, n_{\mathrm{d}}\}$ **do in parallel**
    $\boldsymbol{v}_j^{(t)} = \boldsymbol{\Phi}_j \boldsymbol{x}^{(t-1)}$
    $\boldsymbol{r}_j^{(t)} = \mathcal{P}_{\mathcal{B}}^{\varepsilon}\left(\boldsymbol{v}_j^{(t)} + \boldsymbol{s}_j^{(t-1)}\right)$
    $\boldsymbol{s}_j^{(t)} = \boldsymbol{s}_j^{(t-1)} + \varrho\left(\boldsymbol{v}_j^{(t)} - \boldsymbol{r}_j^{(t)}\right)$
    $\boldsymbol{q}_j^{(t)} = \boldsymbol{\Phi}_j^{\dagger}\left(\boldsymbol{v}_j^{(t)} - \boldsymbol{r}_j^{(t)} + \boldsymbol{s}_j^{(t)}\right)$
    $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t-1)} - \rho \mathrm{AllSumAll}_j(\boldsymbol{q}_j^{(t)})$
    $\boldsymbol{x}^{(t)} = \textsc{DualFB}\left(\tilde{\boldsymbol{x}}^{(t)}, \gamma\right)$
  end
until convergence
function $\textsc{DualFB}\left(\boldsymbol{z}, \gamma\right)$
  given $\boldsymbol{d}_j^{(0)}, \eta$
  $\bar{\boldsymbol{z}}^{(0)} = \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{z}\right)$
  repeat for $k = 1, \ldots$
    $\forall j \in \{1, \ldots, n_{\mathrm{w}}\}$ **do in parallel**
      $\boldsymbol{d}_j^{(k)} = \frac{1}{\eta}\left(\mathcal{I} - \mathcal{S}_{\gamma}\right)\left(\eta \boldsymbol{d}_j^{(k-1)} + \boldsymbol{\Psi}_j^{\dagger} \bar{\boldsymbol{z}}^{(k-1)}\right)$
      $\bar{\boldsymbol{z}}^{(k)} = \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{z} - \mathrm{AllSumAll}_{\boldsymbol{j}}\left(\boldsymbol{\Psi}_j \boldsymbol{d}_j^{(k)}\right)\right)$
    end
  until convergence
return $\bar{\boldsymbol{z}}^{(k)}$

Each ℓ_2-ball proximal operator acts on a different section of $\boldsymbol{y}_j$, so they can be performed in parallel with no communication [30]

$$\mathcal{P}_{\mathcal{B}_j}^{\varepsilon_j}(\boldsymbol{z}_j) \triangleq \begin{cases} \varepsilon_j \frac{\boldsymbol{z}_j - \boldsymbol{y}_j}{\|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2}} + \boldsymbol{y}_j & \|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2} > \varepsilon_j \\ \boldsymbol{z}_j & \|\boldsymbol{z}_j - \boldsymbol{y}_j\|_{\ell_2} \leq \varepsilon_j \end{cases}. \tag{5.21}$$

Algorithm 7 Distributed Fourier Grid (Dual Forward-Backward ADMM): Every node has access to a global ℓ_2-ball proximal operator. The measurement operator is split. First, the root process computes $\boldsymbol{FZS}$, and scatters parts of the FFT grid $\boldsymbol{b}_j^{(t)}$ to each node. Each node then applies $\boldsymbol{G}_j$ to predict the visibilities for the j^{th} node. After the ℓ_2-ball proximal operator is applied, each node applies $\boldsymbol{G}_j^\dagger$, then the root node gathers and adds the result. Then an update image is broadcast to the other nodes, which is needed for $\textsc{DualFB}\left(\tilde{\boldsymbol{x}}^{(t)}, \gamma\right)$. The rest of the algorithm is as in Algorithm 6.

given $\boldsymbol{x}^{(0)}, \boldsymbol{r}_j^{(0)}, \boldsymbol{s}_j^{(0)}, \boldsymbol{q}_j^{(0)}, \gamma, \rho, \varrho$
repeat for $t = 1, \ldots$
  $\forall j \in \{1, \ldots, n_{\text{d}}\}$ **do in parallel**
    Root process only: $\tilde{\boldsymbol{b}}^{(t)} = \boldsymbol{FZS}\boldsymbol{x}^{(t-1)}$
    $\boldsymbol{b}_j^{(t)} = \text{Scatter}_j(\boldsymbol{M}_j \tilde{\boldsymbol{b}}^{(t)})$
    $\boldsymbol{v}_j^{(t)} = \boldsymbol{G}_j \boldsymbol{b}_j^{(t)}$
    $\boldsymbol{r}_j^{(t)} = \mathcal{P}_{\mathcal{B}}^{\varepsilon}\left(\boldsymbol{v}_j^{(t)} + \boldsymbol{s}_j^{(t-1)}\right)$
    $\boldsymbol{s}_j^{(t)} = \boldsymbol{s}_j^{(t-1)} + \varrho(\boldsymbol{v}_j^{(t)} - \boldsymbol{r}_j^{(t)})$
    $\boldsymbol{q}_j^{(t)} = \boldsymbol{G}_j^\dagger\left(\boldsymbol{v}_j^{(t)} - \boldsymbol{r}_j^{(t)} + \boldsymbol{s}_j^{(t)}\right)$
    Root process only: $\boldsymbol{q}_j^{(t)} = \text{Gather}_j(\boldsymbol{q}_j^{(t)})$
    Root process only: $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t-1)} - \rho \boldsymbol{Z}^\dagger \boldsymbol{F}^\dagger \sum_{j=1}^{n_{\text{d}}} \boldsymbol{M}_j^\dagger \boldsymbol{q}_j^{(t)}$
    $\tilde{\boldsymbol{x}}^{(t)} = \text{Broadcast}(\tilde{\boldsymbol{x}}^{(t)})$
    $\boldsymbol{x}^{(t)} = \textsc{DualFB}(\tilde{\boldsymbol{x}}^{(t)}, \gamma)$
  end
until convergence
function $\textsc{DualFB}(\boldsymbol{z}, \gamma)$
  given $\boldsymbol{d}_i^{(0)}, \eta$
  $\bar{\boldsymbol{z}}^{(0)} = \mathcal{P}_{\mathcal{C}}(\boldsymbol{z})$
  repeat for $k = 1, \ldots$
    $\forall j \in \{1, \ldots, n_{\text{w}}\}$ **do in parallel**
      $\boldsymbol{d}_j^{(k)} = \frac{1}{\eta}\left(\mathcal{I} - \mathcal{S}_\gamma\right)\left(\eta \boldsymbol{d}_j^{(k-1)} + \boldsymbol{\Psi}_j^\dagger \bar{\boldsymbol{z}}^{(k-1)}\right)$
      $\bar{\boldsymbol{z}}^{(k)} = \mathcal{P}_{\mathcal{C}}\left(\boldsymbol{z} - \text{AllSumAll}_j\left(\boldsymbol{\Psi}_j \boldsymbol{d}_j^{(k)}\right)\right)$
    end
  until convergence
return $\bar{\boldsymbol{z}}^{(k)}$

By replacing $\mathcal{P}_{\mathcal{B}}^{\varepsilon}(\boldsymbol{z}_j)$ with $\mathcal{P}_{\mathcal{B}_j}^{\varepsilon_j}(\boldsymbol{z}_j)$ in Algorithms 6 and 7, the communication needed can be reduced.

The reduced communication overhead due to the local ℓ_2-ball constraint

is negligible compared to the overhead of the Broadcast and AllSumAll operations performed on $\boldsymbol{x}$, since $n_{\mathrm{d}} \ll N$. Additionally, there is a drawback when the convergence is sensitive to the distribution of $\boldsymbol{y}_j$, which is not the case for the global ℓ_2-ball ADMM. We thus advocate using the global fidelity constraint.

5.3 Algorithm Performance using PURIFY

We have implemented the MPI ADMM algorithms from the previous sections in PURIFY and SOPT. In this section, we benchmark the performance against the non-distributed counterpart [1], to show that such methods can decrease the time required for each iteration. We also implement and benchmark a GPU implementation of the measurement operator against its CPU implementation, to show that GPU implementations can further increase the performance (which can be used in conjunction with the MPI algorithms).

5.3.1 PURIFY Software Package

PURIFY has been developed as a software package that will perform distributed sparse image reconstruction of radio interferometric observations to reconstruct a model of the radio sky. The sparse convex optimization algorithms and MPI operations have been implemented in a standalone library known as SOPT. Previous versions of PURIFY are described in [29, 1]. In this section, we describe the latest release of PURIFY (Version 3.0.1) [7] and latest release of SOPT (Version 3.0.1) [8]. You can download and find details on PURIFY at `https://github.com/astro-informatics/purify` and SOPT at `https://github.com/astro-informatics/sopt`.

PURIFY and SOPT have been developed using the C++11 standard. We use the software package Eigen for linear algebra operations [101]. OpenMP is used to increase performance of the FFT, discrete planar wavelet transforms, and sparse matrix multiplication. The separable 2 dimensional discrete Daubechies wavelet transforms (1 to 30) have been implemented using a lifting scheme (more details on wavelet transforms can be found in [131, 132]),

and have been multi-threaded over application of filters. The sparse matrix multiplication is multi-threaded over rows, requiring the sparse matrix to be stored in row-major order for best performance. To perform operations on a GPU, we use the library ArrayFire, which can be used with a CPU, OpenCL, or CUDA back-end [133]. Within SOPT, we have implemented various MPI functionality (all sum all, broadcast, gather, scatter, etc.) to interface with data types and communicate the algorithm operations across the cluster. It is possible to read the measurements (and associated data) using UVFITS or CASA Measurement Set (MS) formats. The UVFITS format follows the format set by [134]. The output images are saved as two dimensional FITS file images, as a sine projection. Currently, the distributed algorithm supported is ADMM. Furthermore, w-projection and w-stacking algorithms are supported for wide-fields of view and are described in the following chapters.

5.3.2 Distribution of Visibilities

PURIFY can read visibilities $\{\boldsymbol{y}_i\}_{i=1}^{n_\mathrm{d}}$, and scatter them to each node on the cluster. How these measurements are distributed is not important when using the global ℓ_2-ball constraint. However, when using local ℓ_2-ball constraints, the way the visibilities are grouped for each node could make a difference to convergence, where it could be better to keep similar baselines on the same node. We do this by assigning different nodes to different regions of the FFT grid, or by assigning different nodes to regions in baseline length $\sqrt{u^2+v^2}$. However, when using the w-stacking algorithm k-means with MPI is used to redistribute the visibilities over the cluster using an all to all operation, as discussed in [2].

5.3.3 Benchmark Timings

In the remainder of this section, we time the operations of the MPI algorithms. We use Google Benchmark[3] to perform the timings of the mean and standard deviation for each operation benchmarked. The times provided are in real

[3] `https://github.com/google/benchmark`

time (incorporating communication), not CPU time, since multi-threaded operations are sensitive to this difference. Each benchmark configuration was timed for 10 runs, providing a mean and standard deviation used for timings and errors in the sections that follow.

The computing cluster Legion at University College London was used to measure the benchmark timings. We used the Type U nodes on Legion, which are configured as a 16 core device with 64GB RAM (160 Dell C6220 nodes - dual processor, eight cores per processor[4]).

In the benchmarks, the root node generates a random Gaussian density sampling distribution of baselines (u, v), ranging from $\pm\pi$ along each axis. The weights $\boldsymbol{W}_j$ and baseline coordinates $(\boldsymbol{u}_j, \boldsymbol{v}_j)$ are distributed to nodes $1 \leq j \leq n_\mathrm{d}$. This allows the construction of $\boldsymbol{W}_j\boldsymbol{G}_\mathrm{j}$ on each node. We use the Kaiser-Bessel kernel as the interpolation (anti-aliasing) convolution kernel for $\boldsymbol{G}_\mathrm{j}$, with a minimum support size of $J = 4$ (see [1] for more details). The identical construction of $\boldsymbol{FZS}$ can then be performed on each node or the root node (depending on the algorithm), and allow us to apply $\boldsymbol{\Phi}$ in each of the MPI algorithms.

5.3.4 MPI Measurement Operator Benchmarks

The AllSumAll($\boldsymbol{x}$) and Broadcast($\boldsymbol{x}$) in the $\boldsymbol{\Phi}^\dagger$ operations will be expensive in communication overheads for large N. Additionally, the calculation of the FFT $\boldsymbol{F}$ does not take advantage of MPI and will have the cost $\mathcal{O}(N \log N)$, albeit the FFT is multi threaded using FFTW and OpenMP to provide performance improvements. It is more likely that the time taken to compute the FFT will take longer than the communication of the image at large N. If we evenly distribute the visibilities so that each node has $M_j = M/n_\mathrm{d}$, the computational complexity of the sparse matrix multiplication $\boldsymbol{G}_j$ reduces to $\mathcal{O}(M_j J^2)$ per node, providing a large advantage at large M and n_d.

We benchmark the MPI $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^\dagger$ implementations against the non-

[4]More details can be found at `https://wiki.rc.ucl.ac.uk/wiki/RC_Systems#Legion_technical_specs`.

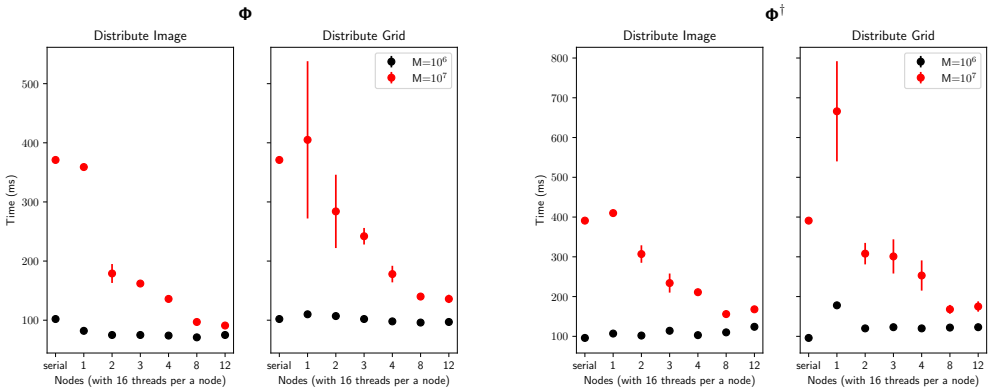

Figure 5.1: Time to apply forward $\boldsymbol{\Phi}$ (left) and adjoint $\boldsymbol{\Phi}^\dagger$ (right) as a function of the number of MPI nodes, benchmarked against the non MPI (serial) implementation. We fix the number of visibilities and image size at $N = 1024 \times 1024$, $M \in \{10^6, 10^7\}$. For the forward and adjoint operators, the left MPI implementation corresponds to using an all sum all MPI operation in the adjoint described in Section 5.2.3.1; on the right, the MPI implementation corresponds to distribution of the grid from the root node, as described in Section 5.2.3.2. Serial corresponds to the serial algorithm that contains no MPI and operates on a single node, but it uses multi-threading through OpenMP.

distributed equivalent using PURIFY. We use 10^6 and 10^7 visibilities, and a fixed image size of $N = 1024 \times 1024$. We vary the number of nodes from $1, 2, 3, 4, 8, 12$. Results are shown in Figure 5.1. For 10^6 visibilities there is no improvement on the measurement operator performance for each MPI implementation. However, for 10^7 it is clear that increasing the number of nodes increases the performance. The saturation for $n_\mathrm{d} \geq 5$ can be explained by the computational cost of the FFT $\boldsymbol{F}$ being greater than the sparse matrix multiplication $\boldsymbol{G}_j$. For small numbers of nodes n_d, i.e. 1 or 2, the application time is less reliable, but for larger n_d it becomes more stable. We also find that distributing sections of the grid (described in Section 5.2.3.2) is more expensive at low n_d than distributing the image (described in Section 5.2.3.1).

5.3.5 MPI Wavelet Operator Benchmarks

Like the measurement operator $\boldsymbol{\Phi}$, the wavelet operator $\boldsymbol{\Psi}$ requires an AllSumAll($\boldsymbol{x}$) operation. However, even with multi-threaded operations in the wavelet transform, computing $\boldsymbol{\Psi}_j$ is time consuming. When $n_\mathrm{w} > n_\mathrm{d}$,

multiple wavelet transforms are performed on some of the nodes, for example when $n_{\mathrm{w}} = 2n_{\mathrm{d}}$ there are two wavelet transforms per a node. In many cases we expect that the numbers of nodes is greater than the number of wavelet transforms, i.e. $n_{\mathrm{d}} \geq n_{\mathrm{w}}$, and the maximum benefit from MPI distribution of wavelet transforms can be seen. This can be seen in Figure 5.2, where there is a performance improvement with using MPI to distribute the wavelet transforms across nodes.

In the benchmarks, we use $n_{\mathrm{w}} = 9$ where $\mathbf{\Psi}_0$ is a Dirac basis and $\mathbf{\Psi}_1$ to $\mathbf{\Psi}_8$ are 2 dimensional (the product of 1 dimensional) Db wavelets 1 to 8. We perform the wavelet transform to three levels. Increasing the number of wavelet levels requires more computation, but much of this computation is in the first few levels. Furthermore, the low pass and high pass filters in the Db increase with size from 1 to 8, meaning Db 8 requires more computation than Db 7 at each wavelet level (but we have found the time difference small). The forward operator $\mathbf{\Psi}$ requires up-sampling, meaning that it requires a factor of 2 times more computation than the adjoint $\mathbf{\Psi}^{\dagger}$. The asymptotic behavior in Figure 5.2 shows that there is little improvement in application time by distributing the wavelet transforms for $n_{\mathrm{w}} > 4$ nodes. This could be due to communication or other factors around the method of implementation.

5.3.6 MPI Algorithm Benchmarks

As a demonstration the impact of the MPI operators, we benchmark the Algorithms 6 and Algorithm 7 against the serial Algorithm 5 (equivalent to $n_{\mathrm{d}} = 1$). We fix the number of visibilities and image size at $N = 1024 \times 1024$, $M \in \{10^6, 10^7\}$.

We use local ℓ_2-ball constraints for each node as described in Section 5.2.9. However, PURIFY also provides the ability to use the global ℓ_2-constraint. In practice, we do not find much difference in computation time between using a local or global ℓ_2-constraint.

In Figure 5.3, we time the application of one iteration of ADMM using one Dual Forward-Backward iteration. We find a clear increase in performance

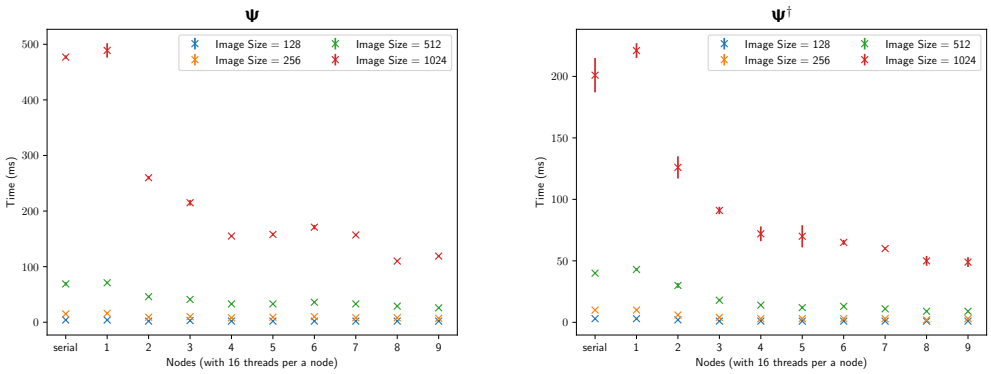

Figure 5.2: Time to apply forward $\mathbf{\Psi}$ (left) and adjoint $\mathbf{\Psi}^\dagger$ (right) as a function of the number of MPI nodes, benchmarked against the non MPI (serial) implementation. The forward operator requires 2 times more calculations than the adjoint due to the up sampling operations. Distributing the wavelet transforms across the nodes greatly decreases the time for calculation, but there is less improvement after 4 nodes. Serial corresponds to the serial algorithm that contains no MPI and operates on a single node, but it uses multi-threading through OpenMP.

when increasing the number of nodes used. This is predicted from the performance improvements from the previous sections. However, the improved times due to distribution seem to be greater than expected from the measurement and wavelet operators alone, suggesting that further aspects of this ADMM implementation improve with distribution.

5.3.7 GPU Measurement Operator Benchmarks

The MPI measurement operators in the previous subsection can also make use of graphics processing units (GPUs) to increase performance. We have implemented the MPI measurement operators using the software package ArrayFire [133], which provides the flexibility to chose a CPU, CUDA, or OpenCL back-end to perform computations. The hybrid MPI-GPU measurement operator works the same as the MPI measurement operator, but all operations on a given node are performed on a GPU. In this section, we show that the GPU can increase performance. We benchmark the ArrayFire implementation using a CUDA back-end, against the equivalent measurement operator. No MPI is used in these benchmarks, since it is clear from the

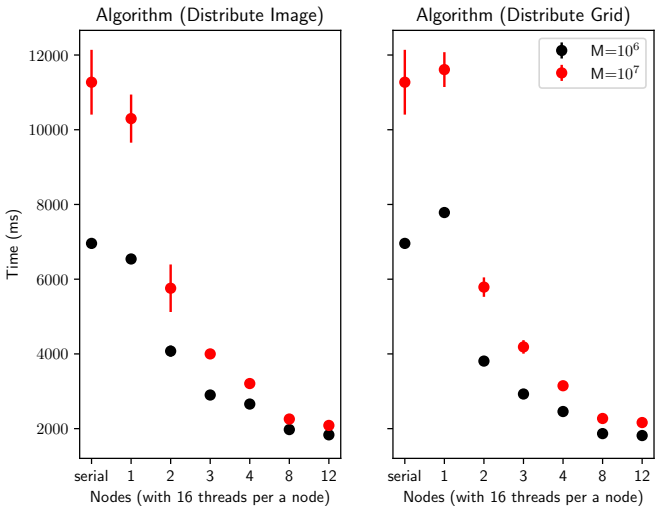

Figure 5.3: Time to apply a single iteration of the ADMM algorithm as a function of the number of MPI nodes, benchmarked against the non MPI (serial) implementation. The Dual Forward Backward algorithm is limited to one iteration. We fix the number of visibilities and image size at $N = 1024 \times 1024$, $M \in \{10^6, 10^7\}$. On the left the MPI implementation corresponds to using Algorithm 6 (which uses the MPI measurement operator from Section 5.2.3.1 where the image is distributed); on the right MPI implementation corresponds to using Algorithm 7 (which uses the MPI measurement operator from Section 5.2.3.2 where the Fourier grid is distributed). The improvements due to distribution seem to be greater than expected from the measurement and wavelet operators alone, suggesting that other aspects of this ADMM implementation improve with distribution.

previous section that MPI will also increase performance. We perform the benchmarks on a high performance workstation, using an NVIDIA Quadro K4200 GPU (with 4GB RAM). We use 5×10^6 visibilities, and use the image sizes of 256×256, 512×512, 1024×1024 and 2048×2048. We find that the GPU implementation of degridding and gridding is about 10 times faster than the CPU counter part when there is a limit to using one thread. Figure 5.4 shows the application of $\mathbf{\Phi}$ and $\mathbf{\Phi}^\dagger$, with a large performance improvement when using the GPU. Interestingly, there is much less increase in performance using a GPU over using 48 CPU threads. Considering that GPUs have hundreds to thousands of threads, this suggests that using a CPU with more threads could be more beneficial. Furthermore, the 4 GB memory limit on the

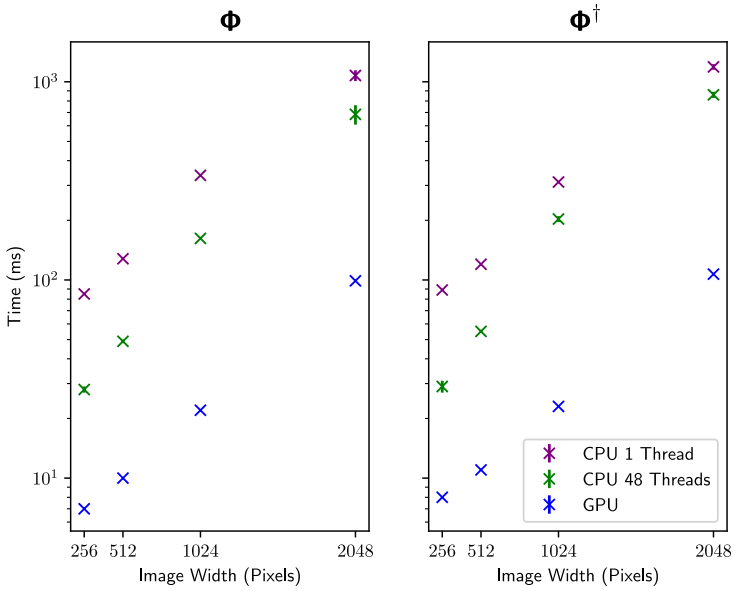

Figure 5.4: Time to apply forward $\mathbf{\Phi}$ and adjoint $\mathbf{\Phi}^\dagger$ as a function of image size, using CPU implementation and ArrayFire with GPU CUDA back-end implementation. We fix the number of visibilities at $M = 5 \times 10^6$, and vary the width of a square image. The CPU times for 1 and 48 threads show that there is some improvement by using threading for the CPU. However, it is clear that GPU implementation remains almost an order of magnitude faster for both gridding and degridding, especially at larger image sizes.

GPU can make scaling to large images or data sets on one node difficult.

5.4 Big Data Interferometric Image Reconstruction Using PURIFY

We follow [1] in performing reconstruction of a simulated M31 observation, but using Algorithm 6 where we distribute the image (rather than distributing the Fourier grid as proposed by [30]), with a global ℓ_2 fidelity constraint (rather than separate local fidelity constraints as proposed by [30]), to reconstruct an observation of 1 billion measurements. For simulation purposes, we use PURIFY to generate 20 million measurements on each of the 50 nodes considered. 50 nodes are not just needed for the speed of iterations, but for the memory required during image reconstruction from the measurements and their interpolation weights $\boldsymbol{G}_\mathrm{j}$. The measurements are created with the

Kaiser-Bessel kernel as the interpolation (anti-aliasing) convolution kernel for $\boldsymbol{G}_{\mathrm{j}}$, with a minimum support size of $J = 8$ (see [1] for more details). We use a Gaussian sampling pattern for (u, v), with the standard deviation of $\pi/3$ for a range of $u, v \in [-\pi, \pi]$.

To simulate the observation measurements, we calculate

$$\boldsymbol{y} = \boldsymbol{\Phi}_j \boldsymbol{x}_{\mathrm{GroundTruth}} + \boldsymbol{n}_j \,. \tag{5.22}$$

where $\boldsymbol{n}_j$ is sampled from identically independently distributed Gaussian noise and $\boldsymbol{x}_{\mathrm{GroundTruth}}$ is the simulated ground truth image of the sky. The standard deviation of the Gaussian noise for each measurement is determined by

$$\sigma_i = \frac{\|\boldsymbol{\Phi}\boldsymbol{x}_{\mathrm{GroundTruth}}\|_{\ell_2}}{\sqrt{M}} \times 10^{-\frac{\mathrm{ISNR}}{20}} \,, \tag{5.23}$$

where the ISNR is the input signal to noise ratio (SNR) on the simulated visibilities with

$$\|\boldsymbol{\Phi}\boldsymbol{x}_{\mathrm{GroundTruth}}\|_{\ell_2} = \sqrt{\mathrm{AllSumAll}(\|\boldsymbol{\Phi}_j \boldsymbol{x}_{\mathrm{GroundTruth}}\|_{\ell_2}^2)} \,. \tag{5.24}$$

See the previous chapter for more explanation on ISNR. We estimate a global ε from σ_i by [1]

$$\varepsilon^2 = (2M + 2\sqrt{4M})\frac{\sigma_i^2}{2} \,. \tag{5.25}$$

The local ε_j can be estimated as

$$\varepsilon_j^2 = (2M_j + 2\sqrt{4M_j})\frac{\sigma_i^2}{2} \,. \tag{5.26}$$

The term $\sqrt{4M_j}$ means that $\varepsilon^2 \neq \sum_j \varepsilon_j^2$;[5] however, we find that in the limit that M is large and dominates over $\sqrt{M}$ and $\sum_j \sqrt{M_j}$. For reconstruction, the support size is lowered to $J = 4$.

We simulate an observation using an ISNR of 30 dB, and an image of

[5] [30] redefines the local ε_j such that $\varepsilon^2 = \sum_j \varepsilon_j^2$.

M31 that is 1024×1024 pixels. To perform the reconstruction we use 1 billion visibilities distributed over 50 nodes of the Grace computing cluster at University College London. Each node of Grace contains two 8 core Intel Xeon E5-2630v3 processors (16 cores total) and 64 Gigabytes of RAM.[6]

Figure 5.5 shows the ground truth, reconstruction, and residuals of M31 using an image size of 1024 by 1024 pixels. We use the wavelet dictionary of a Dirac basis, followed by Daubechies 1 to 8, each with 4 wavelet levels. We use a positive valued constraint. The Dual Forward-Backward iterations for the ℓ_1-proximal operator converge at a relative difference of less than 10^{-3}. Five Dual Forward-Backward iterations are needed per ADMM iteration. The reconstruction has a reconstructed SNR of 31.9554 dB using the formula

$$\text{SNR} = 20\log_{10}\left[\frac{\|\boldsymbol{x}\|_{\ell_2}}{\|\boldsymbol{x} - \boldsymbol{x}_{\text{GroundTruth}}\|_{\ell_2}}\right], \tag{5.27}$$

where $\boldsymbol{x}^*$ is the ground truth and $\boldsymbol{x}$ is the reconstructed image. The residuals show that most of the small and large scale structures of M31 are modeled, which qualitatively shows that the reconstruction models the data well. While there were only 9 iterations of ADMM and this took less than 2 minutes to reach the convergence criteria, we expect that performing more iterations could produce better results. The structure left in the residuals suggests a tighter convergence criteria should be given. Such a reconstruction using 1 billion measurements would not be possible without making use of the distributed Dual Forward-Backward ADMM algorithm. A similar reconstruction was performed with wide-field corrections in [2], where the same Dual Forward-Backward ADMM algorithm was used.

5.5 Conclusion

In this chapter we have used the mathematics of convex optimization methods with their application to distributed interferometric image reconstruction as

[6]More details can be found at `https://wiki.rc.ucl.ac.uk/wiki/RC_Systems#Grace_technical_specs`

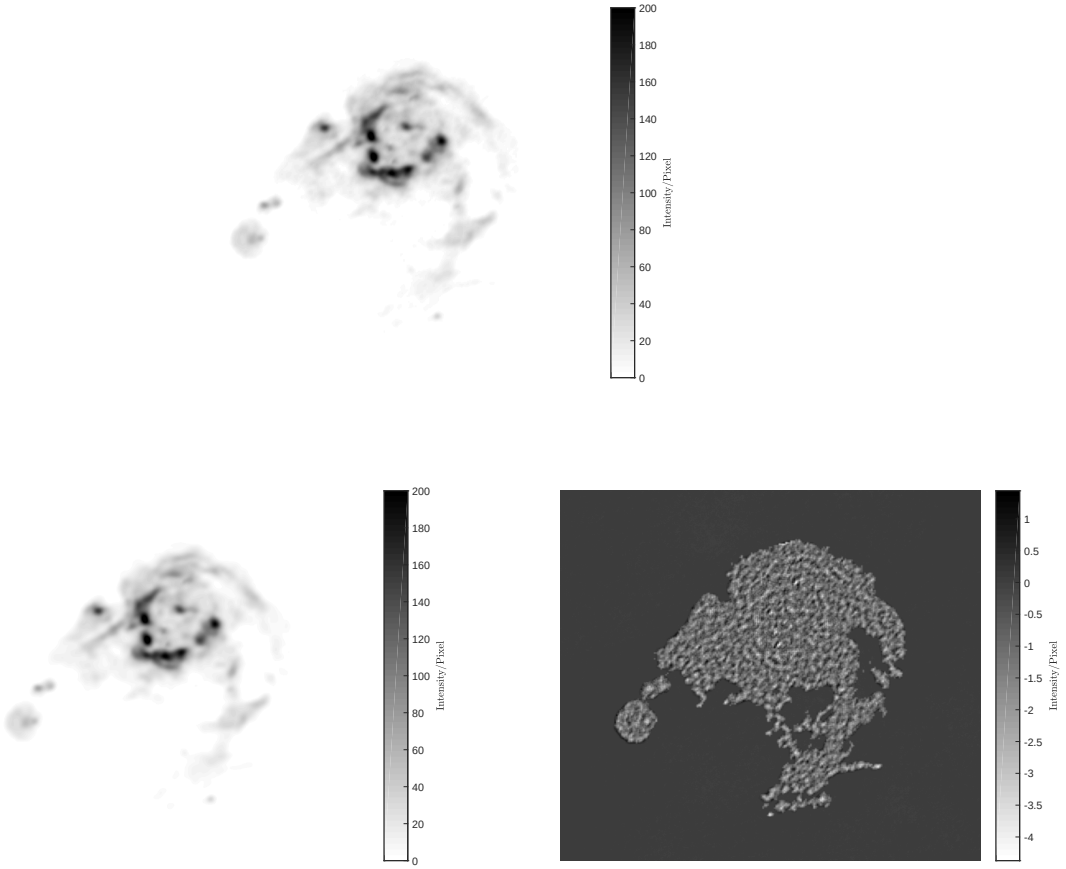

Figure 5.5: Reconstruction of simulated M31 observation using Algorithm 6 with a global ℓ_2 constraint, with the ground truth (top), reconstructed sky model (bottom left), and residuals (bottom right). The simulation is 1024 pixels in width and height. One billion visibilities were evenly distributed across 50 nodes, with one MPI process and 20 million measurements per compute node. With 9 iterations of ADMM, an accurate full sky model is reconstructed. There are structures left in the residual that suggest the convergence criteria could be improved, and more iterations would be helpful.

implemented by the PURIFY 3.0.1 and SOPT 3.0.1 software packages [7, 8]. We reviewed the planar radio interferometric measurement equation and derive the objective functions that can be minimized to obtain a solution to the inverse imaging problem. We then introduce various convex optimization techniques that can be used to minimize these objective functions. We develop algorithms to distribute and parallelize these approaches when dealing with very large data sets, where both computations and data are distributed across the nodes of a computing cluster, while on each node multi-threading is exploited on GPUs or across CPU cores. We then benchmark the implementations, demonstrating considerable computational savings compared to the serial equivalents.

With next generation radio interferometric telescopes coming online, distributed and parallel image reconstruction and data analysis will be necessary to deal with the large image sizes and large volumes of data of forthcoming telescopes. This work is an important step on the path to developing computational algorithms that will be required for telescopes to reach the high resolution and sensitivity needed for science goals of telescopes such as the SKA.

Chapter 6

Fast and Exact w-stacking w-projection Hybrid Algorithm

Since the advent of radio interferometry in the 1940s [135, 136] radio astronomers have built an impressive suite of interferometric imaging techniques to allow signals from collections of antennas to be used collectively to image astronomical sources. As successive generations of interferometric arrays were built and operated, techniques were developed to obtain an estimate of the true sky brightness distribution, and to correct for different instrumental affects inherent in the process. Among these methods are processes such as deconvolution of the antenna response, so-called 'CLEANing' [17, 137, 47, 51], and methods to account for wide-field and other direction dependent effects (DDEs) such as w-projection [76] and a-projection [138].

However, the w-projection algorithm kernels, used to correct for non-coplanar array and sky curvature, to date have been computationally expensive to calculate, with kernel generation dominated by the Fast Fourier Transform (FFT) [139]. In particular the gridding kernel (anti-aliasing kernel) and w-chirp are multiplied in image space, and then an FFT is applied to generate the w-projection kernel [140]. This means it has not been possible to generate a kernel for each w-term individually, instead they are generated as w-planes, approximately correcting for a group of w-terms.

For extremely wide-fields of view, this becomes expensive in computation

and memory, and requires both high resolution sampling to model the spherical curvature and extra zero padding to increase sub-pixel accuracy in the uv-domain. Such a cost in kernel construction has motivated alternative imaging strategies, such as image domain gridding [141]. Even for small fields of view with high resolution, it is not possible to perform an FFT for each visibility on large data sets, limiting the kernel calculation to a small number of w-planes. However, [142] mathematically showed that for narrow fields of view the w-projection kernel can be approximated as separable into a product of two 1 dimensional kernels, reducing the resources required to generate w-planes.

In this chapter, we set out to improve the analytic understanding of wide-field interferometry, in the hopes that it would provide clues on how to improve the strategy of expensive kernel construction. We start by presenting the non-standard analytic expression for the 3 dimensional Fourier transform used to create the w-projection kernel. Then using the analytic expression for the Fourier transform of a spherical shell and enforcing the horizon window with a convolution kernel, we arrive at the 3 dimensional expression for the sky curvature and horizon in the uvw-domain. The real component of the kernel is a radial Sinc function in uvw. It is also clear that the horizon window produces the imaginary component, which is a Hilbert transform of the real component. With this understanding, we investigate construction through 3 dimensional convolution in the uvw-domain to generate gridding kernels. However, this proves computationally challenging due to rapid osculations and large function support[1], and an alternative strategy is provided in the next chapter.

We find it is less challenging to generate the w-projection kernel via a Fourier integral using 2 dimensional adaptive quadrature, due to the smoothness of the window function and the chirp. However, under the condition that the window function has radial symmetry, this 2 dimensional Fourier integral is equivalent to 1 dimensional Hankel transform. We show that such a 1 dimensional Hankel transform can be fast and accurately computed

[1]By the support of a function we mean the region of the domain where the function has non-zero output.

with adaptive quadrature compared to the 2 dimensional Fourier integral, and produces the same imaging results.

We discuss the computational impact of having a 1 dimensional radially symmetric w-projection kernel, such as reducing the dimension of w-planes from 2 dimensional to 1 dimensional radial planes, allowing new possibilities for reducing kernel construction costs.

Lastly, we provide a demonstration of exact correction of the w-component to a MWA observation of the Puppis A and Vela supernova remnants using the sparse image reconstruction using the software package PURIFY [29, 1], using the hybrid of w-stacking and w-projection with distributed computation on a high performance computing cluster. Correction of the w-component for each measurement is only possible with the developments in this work, a radially symmetric w-projection kernel and distributed computation with w-stacking.

The developments presented here provide an accurate route for reducing the computational overhead for next generation wide-field imaging, thus providing a step forward on the path to realizing the SKA.

The calculation of a 1 dimensional radially symmetric w-projection kernel is derived in Section 6.3. The 1 dimensional radially symmetric kernel is then numerically validated and benchmarked in Section 6.4. Section 6.5 details and demonstrates the computationally distributed w-stacking and w-projection hybrid algorithm that is possible with a 1 dimensional w-projection kernel. This chapter is concluded in Section 6.6.

This chapter starts with an introduction to the w-projection algorithm in Section 6.1, Section 6.2 extends the w-projection derivation starting from a 3 dimensional setting.

6.1 The projection algorithm

The projection algorithm has been developed to model baseline dependent effects. Typically, DDEs in the measurement equation such as the primary beam and w-term are multiplied with the sky intensity in the image domain.

Since they are baseline dependent, a separate primary beam and w-term would need to be multiplied for each baseline – which is computationally inefficient as this involves applying a different gridding/degridding process for each baseline.

If we define our baseline dependent DDEs as

$$c(l,m;w) = a(l,m)\frac{\mathrm{e}^{-2\pi i w(\sqrt{1-l^2-m^2}-1)}}{\sqrt{1-l^2-m^2}}\,, \tag{6.1}$$

the measurement equation can be expressed as

$$\begin{aligned} y(u,v,\bar{w}+w) = \int x(l,m)\mathrm{e}^{-2\pi i\bar{w}(\sqrt{1-l^2-m^2}-1)} \\ \times c(l,m;w)\mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,. \end{aligned} \tag{6.2}$$

We can use the convolution theorem, which states that for functions f and g we have $\mathcal{F}^{-1}\{\mathcal{F}\{f\}\mathcal{F}\{g\}\} = f \star g$, where convolution in 3 dimensional is defined as

$$(f\star g)(x,y,z) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t,r,q)g(x-t,y-r,z-q)\,\mathrm{d}t\mathrm{d}r\mathrm{d}q\,. \tag{6.3}$$

This produces the expression

$$y(u,v,w) = \tilde{y}(u,v,0)\star C(u,v,w)\,, \tag{6.4}$$

where $\tilde{y}(u,v,0)$ is the Fourier transform of the sky brightness

$$\tilde{y}(u,v,0) = \int x(l,m)\mathrm{e}^{-2\pi i\bar{w}(\sqrt{1-l^2-m^2}-1)}\mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,. \tag{6.5}$$

where the projection kernel C is the Fourier representation of c, and $\star$ is the convolution operation.

6.1.1 Projection with convolutional degridding

Since the convolution with gridding kernels is already baseline dependent, we can include the projection convolution in the gridding process. If we let $G(u,v)$

be a gridding kernel, and the Fourier transform of the window function $g(l,m)$, we find

$$\begin{aligned} y(u,v,w) = \int \left[\frac{x(l,m)}{g(l,m)}\right] \mathrm{e}^{-2\pi i\bar{w}(\sqrt{1-l^2-m^2}-1)} \\ \times\, g(l,m)c(l,m;w)\mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,, \end{aligned} \tag{6.6}$$

this suggests that we should define a new convolutional kernel

$$[GC]\,(u,v,w) = G(u,v)\star C(u,v,w) \tag{6.7}$$

$$y(u,v,w) = \tilde{y}(u,v,0)\star[GC]\,(u,v,w)\,, \tag{6.8}$$

where $\tilde{y}(u,v,0)$ is now the Fourier transform of the gridding corrected sky brightness

$$\tilde{y}(u,v,0) = \int \frac{x(l,m)\mathrm{e}^{-2\pi i\bar{w}(\sqrt{1-l^2-m^2}-1)}}{g(l,m)}\mathrm{e}^{-2\pi i(lu+mv)}\,\mathrm{d}l\mathrm{d}m\,. \tag{6.9}$$

Traditionally, the kernel is window separable in l and m, i.e. $g(l,m) = g(l)g(m)$. But, as relevant for the later sections of this work, it can be a radial function, i.e. a function of $\sqrt{l^2+m^2}$ only.

This shows that we can include the projection convolution in the gridding process through the kernel GC in Equation 6.8 and the operator $\boldsymbol{GC}$ seen in Equation 2.24. In the next section, we derive expressions for the chirp kernel C in uvw-space from a 3 dimensional setting.

6.2 Projection algorithm in a 3 dimensional setting

In this section, we derive the 3 dimensional w-projection kernel C_{H} formula including the horizon. We start using a measurement equation which can be expressed to include the horizon explicitly and any restrictions of our signal to the sphere. We restrict the signal above horizon in 3 dimensional through the

Heaviside step function

$$\Theta(n) = \begin{cases} 1 & n > 0 \\ \frac{1}{2} & n = 0 \\ 0 & n < 0 \end{cases} \tag{6.10}$$

and to the sphere through the Dirac delta function, yielding $\delta(1-l^2-m^2-n^2)$,

$$c_{\mathrm{H}}(l,m,n;w') = \Theta(n)\,\delta(1-l^2-m^2-n^2)\mathrm{e}^{+2\pi i w'}\,. \tag{6.11}$$

This leads to the measurement equation

$$\begin{aligned} y(u,v,w') = \int_{-\infty,-\infty,-\infty}^{\infty,\infty,\infty} & x(l,m)a(l,m)c_{\mathrm{H}}(l,m,n;w') \\ & \times \mathrm{e}^{-2\pi i(lu+mv+nw')}\mathrm{d}l\mathrm{d}m\mathrm{d}n\,. \end{aligned} \tag{6.12}$$

where equivalent 3 dimensional equations can be found in [36, 76, 12]. Unlike the previous section, the above equation has no $1/n$ term. This term is provided by the Dirac composition rule, which is shown in the next subsection.

For telescopes that make use of Earth Rotation Synthesis and track a source location across the sky, some pointing locations during the observation could be closer to the horizon. There might be times when a source is below the horizon and not detected by the telescope, but can be detected above the horizon at other times. In most cases this effect would be small, but could in principle be modeled in the primary beam for telescopes that are sensitive at the horizon. Many telescopes with an extremely wide-field of view use the drift scan observation strategy where the horizon is fixed as a function of time [112].

6.2.1 w-projection including the horizon directly

In section, we show that the kernel in the work of [76] is equivalent to including both the horizon and spherical effects in the projection algorithm in a full 3

dimensional setting. The Fourier transform of Equation 6.11 is

$$C_{\rm H}(u,v,w) = \int_{0,-\infty,-\infty}^{\infty,\infty,\infty} \delta(1-l^2-m^2-n^2)\mathrm{e}^{-2\pi i(lu+mv+nw)}\mathrm{e}^{+2\pi iw}\mathrm{d}l\mathrm{d}m\mathrm{d}n\,. \tag{6.13}$$

We find that the Dirac delta function argument is zero at two values of $n = n_{\pm}$, where $n_{\pm} = \pm\sqrt{1-l^2-m^2}$ are the two roots. In addition, we have $\delta(n^2 - n_+^2) = (\delta(n-n_+) - \delta(n-n_-))/(2n_+)$, however, the horizon eliminates the $n = n_-$ root from the integral. Using the composition rule for the Dirac delta function we have

$$\begin{aligned} C_{\rm H}(u,v,w) = \int_{0,-1,-1}^{1,1,1} \frac{\delta(n-n_+)}{2} \frac{\mathrm{e}^{-2\pi iw\sqrt{1-l^2-m^2}}}{\sqrt{1-l^2-m^2}} \\ \times \mathrm{e}^{-2\pi i(ul+mv)}\mathrm{e}^{+2\pi iw}\mathrm{d}l\mathrm{d}m\mathrm{d}n\,, \end{aligned} \tag{6.14}$$

where the bounds of integration are now restricted to the sphere. Doing an integral over n we find

$$C_{\rm H}(u,v,w) = \int_{-1,-1}^{1,1} \frac{\mathrm{e}^{-2\pi iw(\sqrt{1-l^2-m^2}-1)}}{2\sqrt{1-l^2-m^2}} \mathrm{e}^{-2\pi i(ul+mv)}\mathrm{d}l\mathrm{d}m\,. \tag{6.15}$$

This is the standard expression used for the w-projection kernel in [76], with the inclusion of a factor of $1/2$ from there being two roots and normalization of the Dirac Delta function. To date, there is no analytical solution for this integral beyond approximations. One reason this integral may be difficult to solve analytically, is the breaking of spherical symmetry when including the horizon.

Having no analytic solution to this integral poses a problem in understanding the properties of $C_{\rm H}(u,v,w)$. This has lead to various approximations of $C_{\rm H}(u,v,w)$, where the solution can be used estimate its support and amplitude.

We can expand $w(\sqrt{1-l^2-m^2}-1)$ in a Taylor expansion to a given order.

We can expand in $(\sqrt{1-l^2-m^2}-1)$ to first order, we find

$$w(\sqrt{1-l^2-m^2}-1) = -\frac{w(l^2+m^2)}{2} + \mathcal{O}(w(l^2+m^2)^2)\,. \tag{6.16}$$

This has the assumption $w(l^2+m^2)^2 \ll 1$. Also choosing a small field of view $(l^2+m^2)^2 \ll 1$ leads to

$$\frac{\mathrm{e}^{-2\pi i w(\sqrt{1-l^2-m^2}-1)}}{2\sqrt{1-l^2-m^2}} \to \frac{\mathrm{e}^{\pi i w(l^2+m^2)}}{2}\,. \tag{6.17}$$

In [76], they state the above small field of view approximation, which is a Gaussian. The Fourier transform of a Gaussian function is also Gaussian, and leads to

$$C_{\mathrm{H}}(u,v,w) \propto \frac{\mathrm{e}^{i\pi\frac{(u^2+v^2)}{w}}}{iw}\,, \tag{6.18}$$

however, they comment that this expression breaks down at large fields of view and diverges at $w=0$. By choosing to fix the sky to a parabola, rather than the sphere, we arrive at the same approximation above. First we choose

$$c_{\mathrm{H}}(l,m,n;w') = \frac{1}{2}\delta\left(n + \frac{l^2+m^2}{2}\right), \tag{6.19}$$

then by integrating over n in Equation 6.12 we arrive at same small field of view approximation.

6.2.2 w-projection with exact spherical correction

We choose to replace the horizon with a window function, where the expression for the full sphere is

$$c_{\mathrm{H}}(l,m,n;w') = h(n)\delta(1-l^2-m^2-n^2)\,. \tag{6.20}$$

Any scaling from this window function can be corrected in the upper hemisphere of the measurement equation

$$y(u,v,w') = \int_{-\infty,-\infty,-\infty}^{\infty,\infty,\infty} \frac{x(l,m)a(l,m)}{h(\sqrt{1-l^2-m^2})} c_{\mathrm{H}}(l,m,n;w') \times \mathrm{e}^{-2\pi i(ul+mv+nw')} \mathrm{e}^{+2i\pi w'} \mathrm{d}l\mathrm{d}m\mathrm{d}n\,. \tag{6.21}$$

6.2.2.1 No horizon

When $h(n) = 1$ there is no horizon and the w-projection kernel is calculated from

$$C(u,v,w) = \int_{-\infty,-\infty,-\infty}^{\infty,\infty,\infty} \delta(1-l^2-m^2-n^2) \mathrm{e}^{-2\pi i(ul+mv+nw)} \mathrm{e}^{+2\pi iw} \mathrm{d}l\mathrm{d}m\mathrm{d}n\,. \tag{6.22}$$

The Fourier transform of this equation has an analytic solution that can be simply expressed as a real valued function

$$C(u,v,w) = 2\pi \mathrm{sinc}\left(2\pi\sqrt{u^2+v^2+w^2}\right) \mathrm{e}^{+2\pi iw}\,, \tag{6.23}$$

as shown in [143], which is solved in spherical coordinates due to symmetry. This solution dates back as far as [144], and similar problems have been solved in 2 dimensions in [145]. The units of (u,v,w) are implicitly chosen to depend on the directional cosines (l,m,n), meaning $\sqrt{u^2+v^2+w^2} = 1$ corresponds to the largest spatial scales.

The Sinc function above represents limits on the resolution in (u,v,w) due to the field of view being bounded to the sphere. The uncertainty principle states that restricting the field of view is equivalent to enforcing a resolution limit on $C(u,v,w)$. At a small field of view, this kernel is effectively a delta function of small support. However, as the field of view increases, the kernel becomes a radial Sinc function with extended support and rapid oscillations. When mosaicking multiple fields of view, resolution in (u,v,w) is increased (as discussed in [146] and [36]), however, the total field of view will be limited to

the sphere as represented by this radial Sinc function.

Since $x(l,m)$ is independent of n it will project both onto the sphere for n and $-n$. While $C(u,v,w)$ models the curvature of the sphere, it allows a reflection of $x(l,m)$ for $-1 \leq n < 0$. This is why a horizon window function needs to be included in the analysis.

6.2.2.2 Projecting above the Horizon

If we let $H(w)$ be the Fourier transform of $h(n)$, we find that the horizon effect can be understood through the convolution theorem

$$C_{\mathrm{H}}(u,v,w) = H(w) \star C(u,v,w)\,. \tag{6.24}$$

We can get an expression for the horizon limited w-projection kernel in the (u,v,w) domain in terms of the w-projection kernel for the full sphere. Choosing $h(n) = \Theta(n)$ with $H(w) = \frac{1}{2}\left[\delta(w) - \frac{i}{\pi w}\right]$, we find an expression equivalent to Equation 6.15 in the (u,v,w) domain

$$C_{\mathrm{H}}(u,v,w) = \frac{1}{2}C(u,v,w) - \frac{i}{2\pi}\int_{-\infty}^{\infty}\frac{C(u,v,t)}{w-t}\mathrm{d}t\,, \tag{6.25}$$

where the second term is a Hilbert transform of the sphere along the w-axis. Another equivalent expression can be found by choosing a box function $h(n) = \Pi(n+\frac{1}{2})$ for the horizon window, by setting $H(w) = \mathrm{e}^{i\pi w}\frac{\sin(\pi w)}{\pi w}$,

$$C_{\mathrm{H}}(u,v,w) = \int_{-\infty}^{\infty}\mathrm{d}t\,\mathrm{e}^{i\pi t}\mathrm{sinc}(\pi t)C(u,v,w-t)\,. \tag{6.26}$$

We are not aware of an analytic solution to this convolution, which could improve understanding of the behavior of wide field effects.

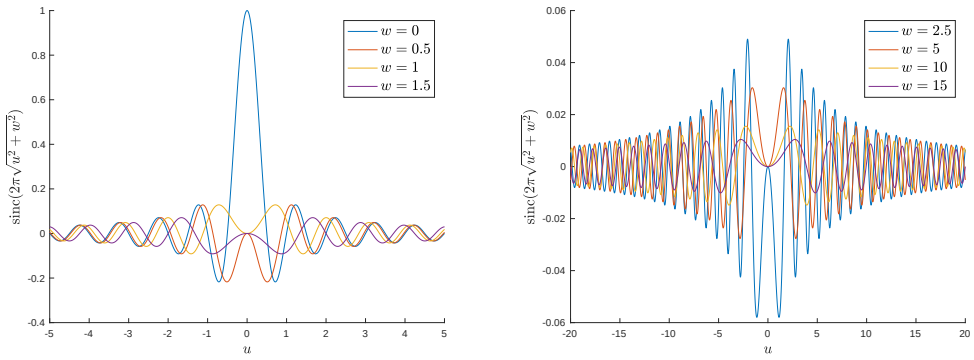

Figure 6.1: The oscillations of C, without the complex phase, as a function of u for given w. Equation 6.28, which is used to calculate the pixel size of a uv-grid, shows that many of these oscillations can occur over the convolution window, making numerical integration difficult for convolution with the gridding kernels G and the horizon H. Hence, we find that convolution by numerical integration is difficult. Additionally, we see that C has a large support that increases with w. The top figure shows the standard Sinc function at $w = 0$, and the bottom figure shows the spread of C over a wider range of u as w increases.

6.2.3 Convolution with a gridding kernel

To calculate the w-projection kernel, we could convolve the chirp with the gridding kernel in the (u, v, w) domain

$$[GC](u,v,w) = \int_{-\infty,-\infty,-\infty}^{\infty,\infty,\infty} G(p)G(q)H(r)C(u-p, v-q, w-r)\mathrm{d}p\mathrm{d}q\mathrm{d}r \; . \tag{6.27}$$

However, the challenge with computing this three dimensional integral is the extended support of H and C in w. Additionally, $C(u, v, w)$ will have rapid oscillation in (u, v) for small values of w, making accurate numerical integration and convolution expensive, see Figure 6.1. Therefore, we avoid this approach in kernel calculation, and present an alternative approach in the next chapter.

6.2.4 Summary

In this section, we investigated exact analytic expressions for modeling curvature in wide-field interferometry, for extremely wide-fields of view. This expression has traditionally been stated in the (l, m, n) domain. However,

this work provides the first exact analytic expression for sky curvature and horizon seen in wide-field interferometry in the (u, v, w) domain. Unlike the previous small field of view approximations, this exact kernel does not diverge and is continuous. Furthermore, it provides more insight and understanding of spherical imaging, i.e. it describes a fundamental resolution limit for the measurement of a visibility from a sphere, and the impact of the horizon window in the (u, v, w) domain. While this expression provides insight, the rapid oscillations due to the spherical sky and large support make calculation difficult. These insights suggest that exact computation of projection kernels is more feasible through a Fourier integral from the (l, m, n) domain.

6.3 Kernel Calculation Methods

In the previous chapter, we discussed the properties of the w-projection kernel in the (l, m, n) and (u, v, w) domains. We expected that the properties for numerical convolution with the chirp and the gridding kernel are more favorable by multiplying the window and the chirp in the image domain, then performing a Fourier transform to generate the kernel in the Fourier domain. This should increase accuracy and reduce the total computation.

In principle we can create an image of $g(l, m)$ and $c(l, m; w)$ and perform an FFT to calculate the w-projection kernel in the Fourier domain. However, this FFT will only calculate kernel values that lie on a regular grid which is a problem since we want to evaluate $[GC](u, v, w)$ off of a grid. The grid can be made finer with an FFT but this typically requires zero padding by a factor of 2 or 8 [76]. For large image sizes this can consume a lot of memory and time during kernel construction for large image sizes and wide-fields of view. However, there is no need to use the FFT with $g(l, m)$ and $c(l, m; w)$ to calculate a Fourier transform because the functions have a closed form. We can then use efficient and low memory adaptive quadrature methods, which are fast for smooth functions.

In this section, we describe two methods for calculating the w-projection kernel using the Fourier transform. The first is numerical integration using adaptive quadrature in 2d, the second is to restrict the imaged region to a radial field of view allowing for a radially symmetric kernel that can be integrated with adaptive quadrature in 1d. In the following section we compare the numerical accuracy and speed of the two kernel construction methods. The scaling $\Theta(1 - l^2 - m^2)/\sqrt{1 - l^2 - m^2}$ is included in the gridding and primary beam correction, because it is baseline independent. We do not include this term in the gridding kernel, and we apply this in the image domain with all other baseline independent effects.

6.3.1 Cartesian integration

To calculate the Fourier coefficients of the w-projection corrected gridding kernel, we need to perform a Fourier series with boundary conditions determined by the size of the window. We let Δu and Δv determine the conversion between pixel and baseline coordinates, $u = u_{\text{pix}}\Delta u$ and $v = v_{\text{pix}}\Delta v$ where u_{pix} and v_{pix} are integer pixel values. This factor is given by

$$\Delta u = \left[2\alpha \sin\left(\frac{N_x \pi \text{cell}}{2 \times 60 \times 60 \times 180.}\right)\right]^{-1}. \tag{6.28}$$

where cell is the size of a pixel in arc-seconds, α is the oversampling ratio, and N_x is the image width of the x-axis. A similar formula is given for Δv, with respect to the y-axis. We use this field of view to integrate over the imaged region, and including the bounds of the sphere

$$\begin{aligned}[GC](u_{\text{pix}}, v_{\text{pix}}, w, \Delta u, \Delta v) = \int_{-\alpha/(2\Delta u),-\alpha/(2\Delta v)}^{\alpha/(2\Delta u),\alpha/(2\Delta v)} & \mathrm{e}^{-2\pi i w(\sqrt{1-l^2-m^2}-1)} \\ & \times g(\Delta u l) g(\Delta v m) \mathrm{e}^{-2\pi i(\Delta u u_{\text{pix}} l + \Delta v v_{\text{pix}} m)} \mathrm{d}l \mathrm{d}m .\end{aligned} \tag{6.29}$$

We then change coordinates $l = x/\Delta u$ and $m = y/\Delta v$ to be relative to the imaged region

$$\begin{aligned}[GC](u_{\text{pix}}, v_{\text{pix}}, w, \Delta u, \Delta v) = \frac{1}{\Delta u \Delta v} \int_{-\alpha/2,-\alpha/2}^{\alpha/2,\alpha/2} & \mathrm{e}^{-2\pi i w(\sqrt{1-x^2/\Delta u^2-y^2/\Delta v^2}-1)} \\ & \times g(x) g(y) \mathrm{e}^{-2\pi i(u_{\text{pix}} x + v_{\text{pix}} y)} \mathrm{d}x \mathrm{d}y .\end{aligned} \tag{6.30}$$

Here $g(l)$ is the window function that determines the gridding kernel and $[GC]$ is the w-projection corrected gridding kernel. It is worth noticing that when $w = 0$, ignoring normalization there is no dependence on Δu or Δv, unless the condition $l^2 + m^2 \leq 1$ is to be enforced.

Depending on the convention of the FFT operation $\boldsymbol{F}$ in the measurement operator, there could be a phase offset of $\mathrm{e}^{\pm 2\pi i u_{\text{pix}}/2}$ and $\mathrm{e}^{\pm 2\pi i v_{\text{pix}}/2}$ required to

centre the image[2]. The region of integration is determined by the zero padded field of view (we have used zero padding by a factor of $\alpha = 2$).

6.3.2 Polar integration

By performing a change of coordinates, this integral can also be evaluated in polar coordinates

$$[GC](u_{\text{pix}}, v_{\text{pix}}, w, \Delta u, \Delta v) = \frac{1}{\Delta u \Delta v} \int_{0,0}^{\alpha/2, 2\pi} g(r\cos(\theta)) g(r\sin(\theta)) \mathrm{e}^{-2\pi i w(\sqrt{1 - r^2\cos^2(\theta)/\Delta u^2 - r^2\sin^2(\theta)/\Delta v^2} - 1)} \times \mathrm{e}^{-2\pi i (u_{\text{pix}} r\cos(\theta) + v_{\text{pix}} r\cos(\theta))} r \mathrm{d}r \mathrm{d}\theta \,, \tag{6.31}$$

The region is circular rather than rectangular, which is a fundamental difference with the Cartesian expression in Equation 6.30 (the boundary conditions for the Fourier series lie on a circle, rather than a square).

The enforces a Sinc convolution with the w-projection for the rectangular boundary condition, and a Airy Pattern convolution (first order Bessel Function) for the circular boundary condition. This translates to a slightly different interpolation when up-sampling the w-projection kernel, Sinc interpolation in the rectangular case, and $J_1(4\pi\sqrt{u^2+v^2}/\alpha)/(2\sqrt{u^2+v^2}/\alpha)$ interpolation in the circular case, both enforcing a band-limit.

It is important to state, this boundary is at the edge of the zero-padded region, which suggests that there would be little difference in practice because it is far outside of the gridding corrected region, and will not change suppression of aliasing error (which is the purpose of the window function/gridding convolution function). This means that while the kernels are fundamentally different due to the boundary condition, they will perform the same role, and the entire measurement operators will be equivalent after gridding correction and zero-padding.

[2]This is due the difference of centering the coordinates in the middle or at the corner of the image, which can require an FFT shift.

6.3.3 Radial symmetry

We now make our window function radially symmetric $g(l)g(m) \to g(\sqrt{l^2+m^2})$, and choose $\Delta u = \Delta v$ so that the chirp is also radially symmetric. This allows us to take the Fourier transform of a radially symmetric function, which is calculated using a 1 dimensional integral rather than the 2 dimensional polar integral in Equation 6.31, and is known as a Hankel transform[3]. This is given by

$$[GC](\sqrt{u_{\text{pix}}^2+v_{\text{pix}}^2}, w, \Delta u) = \frac{2\pi}{\Delta u^2}\int_0^{\alpha/2} g(r)\mathrm{e}^{-2\pi i w(\sqrt{1-r^2/\Delta u^2}-1)} \times J_0\left(2\pi r\sqrt{u_{\text{pix}}^2+v_{\text{pix}}^2}\right) r\mathrm{d}r\,, \tag{6.32}$$

where J_0 is a zeroth order Bessel function. The restriction of $r/\Delta u < 1$ is built into the bounds of the integration. This has the large computational advantage of only sampling along the radius, reducing how the computation scales with field of view and w. There is also an increase in accuracy, since there is no sampling in θ. Furthermore, the condition that we require $\Delta u = \Delta v$ is not difficult to accommodate in many cases.

6.3.4 Adaptive quadrature

To compute Equation 6.30, we use adaptive multidimensional integration. In a multi-variate setting, quadrature is also known as cubature.

We use the software package Cubature[4] which has implementations of these algorithms. We use the h-adaptive cubature method to evaluate the integrals in this work, which uses the work of [148] and [149] to perform integration using an adaptive mesh to approximate the integral, until convergence is reached (h is in reference to a length parameter of the mesh). Cubature also has a p-adaptive method [150], which uses polynomial based quadrature, increasing the polynomial order of the integrand until the integration has converged, and is expected to converge faster than h-adaptive

[3] [147] suggested that convolutions between radially symmetric functions can be efficiently computed using a Hankel Transform but in different astronomical contexts.

[4] https://github.com/stevengj/cubature

methods for smooth integrands.

The p-adaptive method tends to converge faster than the h-adaptive method for the 1d-integration, while providing results as accurate within numerical error. However, the accuracy of the p-adaptive method was not as accurate for 2d-integration, especially in the presence of discontinuities. For this reason, we use the p-adaptive method for 1d-integration but the h-adaptive method for 2d-integration.

6.3.5 Kaiser-Bessel gridding kernel

In this work, we use a Kaiser-Bessel gridding kernel. Kaiser-Bessel functions have been used as convolutional gridding kernels for decades [64, 73, 63], and have a simpler form than the prolate spheroidal wave functions, while providing similar performance [64]. The zeroth order Kaiser-Bessel function can be expressed as

$$G(u_{\text{pix}}) = \frac{I_0\left(\beta\sqrt{1-\left(\frac{2u_{\text{pix}}}{J}\right)^2}\right)}{I_0(\beta)}, \tag{6.33}$$

where u_{pix} has units of pixels, J is the support in units of pixels, I_0 is the zeroth order modified Bessel function of the first kind, and β determines the spread of the Kaiser-Bessel function [73, 63]. The Fourier Transform of $G(u_{\text{pix}})$ is

$$g(x) = \text{sinc}\left(\sqrt{\pi^2 x^2 J^2 - \beta^2}\right). \tag{6.34}$$

To correct for the convolution, the image is divided by $g(l)$ [73, 63]

$$s(x) = [g(x)]^{-1}. \tag{6.35}$$

The work of [63] shows that for $\beta = 2.34J$ the Kaiser-Bessel kernel performs similarly to the optimal min-max kernel considered.

In this chapter, we use the Kaiser-Bessel gridding kernel to calculate w-projection kernels, by using $g(x)$ in Equations 6.30 and 6.32. For other possible

window functions and anti-aliasing kernels, see [12] and [1].

6.4 Validation of Radially Symmetric Kernel

In this section we numerically evaluate Equation 6.30, and present a cross section of the kernel, showing its variation with sub-pixel accuracy. We then numerically evaluate Equation 6.32, showing that it provides the same accurate sub-pixel accuracy, with orders of magnitude less function evaluations during the quadrature computation.

6.4.1 Quadrature convergence conditions

The kernel function is normalized to one when $(u, v, w) = (0, 0, 0)$, and an estimate error tolerance η on the quadrature calculated kernel $[GC]^{\eta}(u_{\text{pix}}, v_{\text{pix}}, w)$ is used for quadrature convergence of the kernel, such that the absolute difference is less than η

$$|[GC](u_{\text{pix}}, v_{\text{pix}}, w) - [GC]^{\eta}(u_{\text{pix}}, v_{\text{pix}}, w)| \leq \eta\,. \tag{6.36}$$

It is also possible to use the relative difference

$$\frac{|[GC](u_{\text{pix}}, v_{\text{pix}}, w) - [GC]^{\eta}(u_{\text{pix}}, v_{\text{pix}}, w)|}{|[GC]^{\eta}(u_{\text{pix}}, v_{\text{pix}}, w)|} \leq \eta\,, \tag{6.37}$$

which would constrain smaller values of $[GC]^{\eta}(u_{\text{pix}}, v_{\text{pix}}, w)$ to be calculated more accurately, at the cost of more computation.

There is a downside of using absolute difference, for example, if you are calculating kernels to an absolute accuracy of 10^{-2} and the kernels have values below 10^{-2} then these values may not be accurate. The relative difference is an ideal alternative, but it can cause an inconsistent level of accuracy across the measurement operator, and more computation can go into small values that may not contribute much in practice. If the support size is known accurately before computation, this may help.

We assume that the support size of the w-projection GC kernel is proportional to $2w/\Delta u$ and at least the support size of the gridding kernel

G. With the support size known, we use the absolute different criteria with $\eta = 10^{-6}$.

6.4.2 Kernel cross-section

Figure 6.2 shows a cross section of the w-projection kernel $[GC](u_{\text{pix}}, 0, w)$, the real and imaginary components, and the absolute value, for $0 \leq u_{\text{pix}} \leq 19$ and $0 \leq w \leq 99$. We find that the convolution of C_{H} with $G(u)$ and $G(v)$ creates a smooth varying w-projection kernel in both real and imaginary components. The imaginary component is zero at $w = 0$, which is consistent with Equation 6.25. We find that the decay in the kernel as a function of w is more extreme with wider fields of view.

We then evoke radial symmetry in the gridding kernel and field of view, and evaluate Equation 6.32 in Figure 6.3. We find that the features of the radially symmetric gridding kernel from Equation 6.30 match the cross section of Equation 6.32, suggesting little difference between the two kernels. Additionally, when N samples are required to evaluate the 1 dimensional radially symmetric kernel, approximately N^2 are required to evaluate the 2 dimensional kernel, as shown in Figure 6.4. This suggests that the symmetric kernel calculation scales with radius, not total area as in the 2 dimensional case. This has enormous general implications for computation and storage for w-projection kernels at large fields of view.

6.4.3 Numerical equivalence of radially symmetric kernel

Next, we show that using the radially symmetric gridding kernel is consistent with the non radially symmetric kernel. To test this, we constructed three measurement operators $\mathbf{\Phi}_{\text{standard}}$ (standard w-projection kernel), $\mathbf{\Phi}_{\text{radial}}$ (symmetric w-projection kernel), and $\mathbf{\Phi}_{\text{no-projection}}$ (no w-term), and show that $\mathbf{\Phi}_{\text{standard}} \approx \mathbf{\Phi}_{\text{radial}}$ within some error (suggesting that they agree), and use $\mathbf{\Phi}_{\text{no-projection}}$ as a reference operator.

To show that two operators are equivalent, we need the notion of an

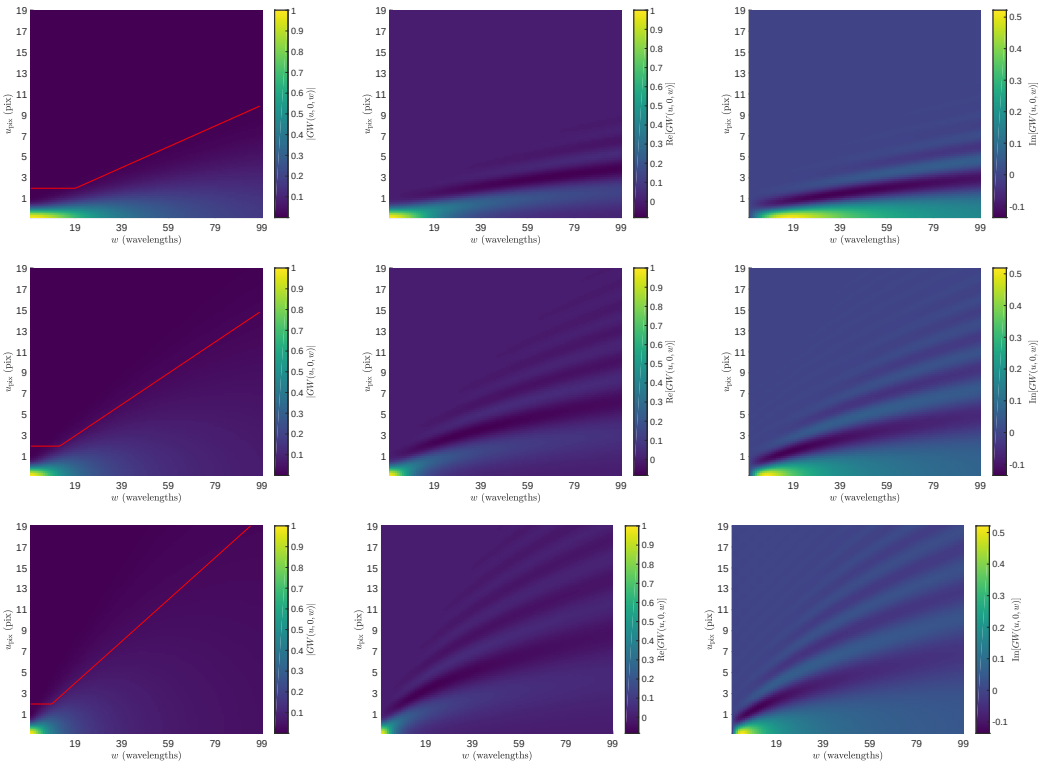

Figure 6.2: Plot of the kernels calculated using Equation 6.30, as a function of u_{pix} and w, with $v_{\text{pix}} = 0$, for absolute (left column), real (middle column), and imaginary (right column) values. Each row has a different field of view, $11.3778° \times 11.3778°$ (top), $17.0667° \times 17.0667°$ (middle), and $22.7556° \times 22.7556°$ (bottom). We see that the kernel spreads as a function of increasing w. The support size in pixels increases with field of view, due to a large field increasing the sampling rate of the kernel. It is also clear that the kernel decreases in value with increasing w, faster at wider fields of view. The real and imaginary components both show oscillations. We find the imaginary component is zero at $w = 0$ as expected. The values have been calculated using adaptive quadrature within an absolute error of $\eta = 10^{-6}$. There are 100 uniform samples in each of u_{pix} and w, making 10^4 for each plot. The red line shows $\max(4, 2w/\Delta u)/2$ for reference, which is assumed to be the support size for this work. The features of this kernel are also consistent with w-projection kernels used by ASKAPSoft [140].

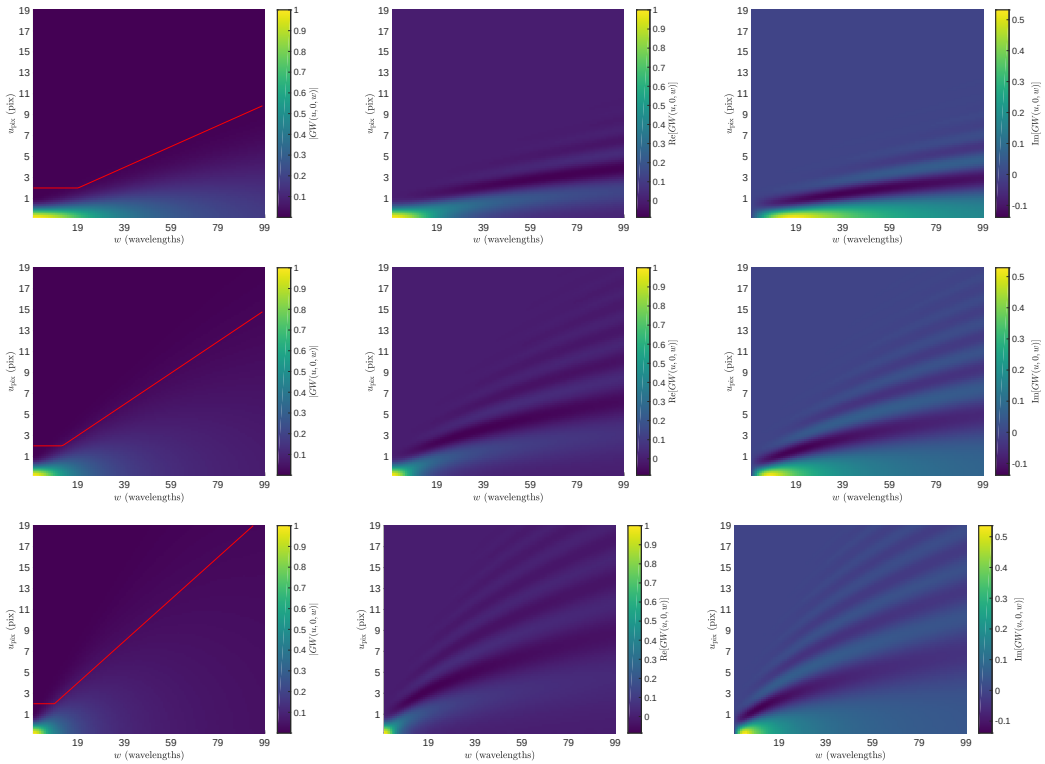

Figure 6.3: Plot of the kernels calculated from Equation 6.32, as a function of u_{pix} and w, with $v_{\text{pix}} = 0$, for absolute (left column), real (middle column), and imaginary (right column) values. Each row has a different field of view, $11.3778° \times 11.3778°$ (top), $17.0667° \times 17.0667°$ (middle), and $22.7556° \times 22.7556°$ (bottom). We find the same features in Figure 6.2, showing that it is consistent with Equation 6.30. The values have been calculated using adaptive quadrature within an absolute error of $\eta = 10^{-6}$. There are 100 uniform samples in each of u_{pix} and w, making 10^4 for each plot. The red line shows $\max(4, 2w/\Delta u)/2$ for reference.

operator norm $\|\cdot\|_{\text{op}}$. The operator norm for an operator that maps between Hilbert spaces (ℓ_2) has the property that

$$\|\boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \|\boldsymbol{\Phi}\|_{\text{op}} \|\boldsymbol{x}\|_{\ell_2} \quad \forall \boldsymbol{x} \in \mathbb{R}^N . \tag{6.38}$$

$\|\boldsymbol{\Phi}\|_{\text{op}}$ is the smallest value for which this is true for all $\boldsymbol{x}$. This allows us to put bounds on the output of $\|\boldsymbol{\Phi}\|_{\text{op}}$ for each input. We also have the properties that $\|\boldsymbol{\Phi}\|_{\text{op}} = \|\boldsymbol{\Phi}^\dagger\|_{\text{op}}$ and $\|\boldsymbol{\Phi}^\dagger\boldsymbol{\Phi}\|_{\text{op}} = \|\boldsymbol{\Phi}\|_{\text{op}}^2$.

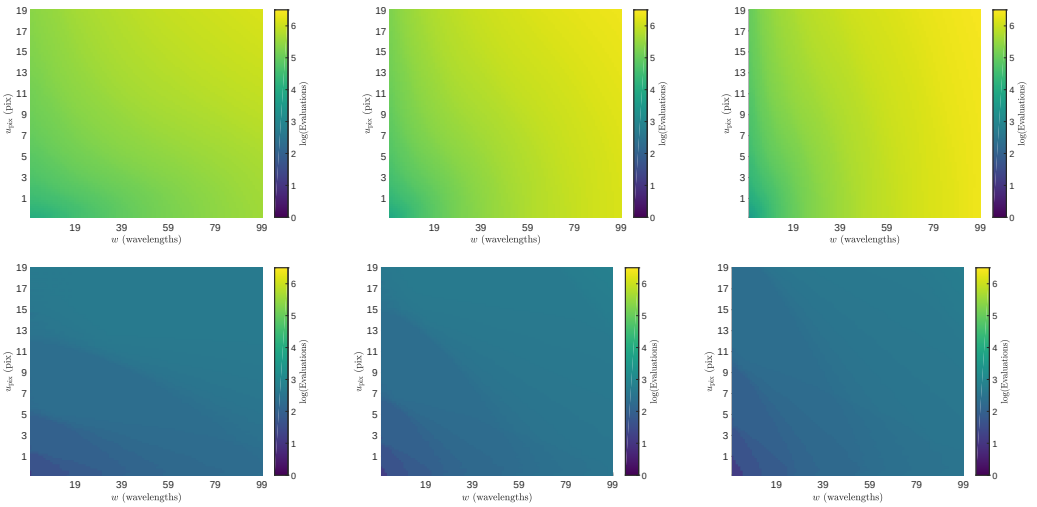

Figure 6.4: The plots above show the number of function evaluations in the quadrature method required to produce Figures 6.2 (top row) and 6.3 (bottom row). Each column corresponds to a field of view of $11.3778° \times 11.3778°$(left), $17.0667° \times 17.0667°$ (middle), and $22.7556° \times 22.7556°$ (right). The top row shows two times the values in the bottom row, suggesting that if Equation 6.32 takes N evaluations, then Equation 6.30 takes N^2 evaluations to compute. This shows the computation of Equation 6.32 scales with radius vs. the computation of Equation 6.30 that scales with area. The number of evaluations required can be greatly reduced by increasing the absolute error η.

The operator norm allows the following statement

$$\begin{aligned} &\frac{\|(\boldsymbol{\Phi}_{\text{standard}} - \boldsymbol{\Phi}_{\text{radial}})\boldsymbol{x}\|_{\ell_2}}{\|\boldsymbol{x}\|_{\ell_2}} \\ &\leq \|\boldsymbol{\Phi}_{\text{standard}} - \boldsymbol{\Phi}_{\text{radial}}\|_{\text{op}} \quad \forall \boldsymbol{x} \in \mathbb{R}^N. \end{aligned} \tag{6.39}$$

For every input sky model $\boldsymbol{x}$, the root-mean-squared (RMS) difference between the model visibilities is bounded by the product of the RMS of the input sky model and the operator norm $\|\boldsymbol{\Phi}_{\text{standard}} - \boldsymbol{\Phi}_{\text{radial}}\|_{\text{op}}$. Additionally, for visibilities $\boldsymbol{y}$

$$\begin{aligned} &\frac{\|(\boldsymbol{\Phi}^\dagger_{\text{standard}} - \boldsymbol{\Phi}^\dagger_{\text{radial}})\boldsymbol{y}\|_{\ell_2}}{\|\boldsymbol{y}\|_{\ell_2}} \\ &\leq \|\boldsymbol{\Phi}_{\text{standard}} - \boldsymbol{\Phi}_{\text{radial}}\|_{\text{op}} \quad \forall \boldsymbol{y} \in \mathbb{R}^M. \end{aligned} \tag{6.40}$$

This statement says that the RMS difference between dirty maps is bounded

by the product of the RMS of the input visibilities and the operator norm $\|\mathbf{\Phi}_{\text{standard}} - \mathbf{\Phi}_{\text{radial}}\|_{\text{op}}$. When $\|\mathbf{\Phi}_{\text{standard}} - \mathbf{\Phi}_{\text{radial}}\|_{\text{op}} = 0$, the two operators will clearly be the same.

Since our linear operators map between two Hilbert spaces, the operator norm of $\mathbf{\Phi}$ is the square root of the largest Eigenvalue of $\mathbf{\Phi}^\dagger\mathbf{\Phi}$. To calculate the largest Eigenvalue, we use the power method (as used in [1]).

First we normalize each operator, such that $\|\mathbf{\Phi}\| = 1$, so there is no arbitrary scaling. Then we calculate $\|\mathbf{\Phi}_{\text{standard}} - \mathbf{\Phi}_{\text{radial}}\|_{\text{op}}$ and $\|\mathbf{\Phi}_{\text{standard}} - \mathbf{\Phi}_{\text{no-projection}}\|_{\text{op}}$.

To construct the measurement operators, we use a variable Gaussian sampling density in (u, v, w), with a root-mean-squared spread of 100 wavelengths. We scale w to have an RMS value of 20 wavelengths. We choose a cell size of 240 arcseconds and an image size of 256 by 256 pixels. This provides a full width field of view of $17.0667^\circ \times 17.0667^\circ$. It is important to note that the w-kernels are a function of the field of view, and not the cell size. The kernel support size is estimated by the w-value for each measurement to be $\min(\max(4, 2w/\Delta u), 40)$. This support has a minimum size of 4 and a largest size of 40, and in between a size of $2w/\Delta u$. The benchmarking was performed on a high performance workstation comprised of two Intel Xeon Processors (E5-2650Lv3) with 12 cores each with 2 times hyper-threading per core (at 1.8 GHz) and 256 Gigabytes of DDR4 RAM (at 2133 MHz).

We found the construction time of a radially symmetric kernel was almost two orders of magnitude faster to calculate. An absolute difference of 10^{-4} was used for quantifying quadrature convergence. The power method was considered converged with a relative difference of 10^{-6}.

In Figure 6.5, we show the operator construction time (excluding the normalization), and the operator norm of the difference. Each data point was generated by averaging over 5 realizations. The number of measurements M ranges from only 100 to 1000. From this figure, it is clear that that the operator difference is consistently on the order of 10^{-3}, suggesting that we

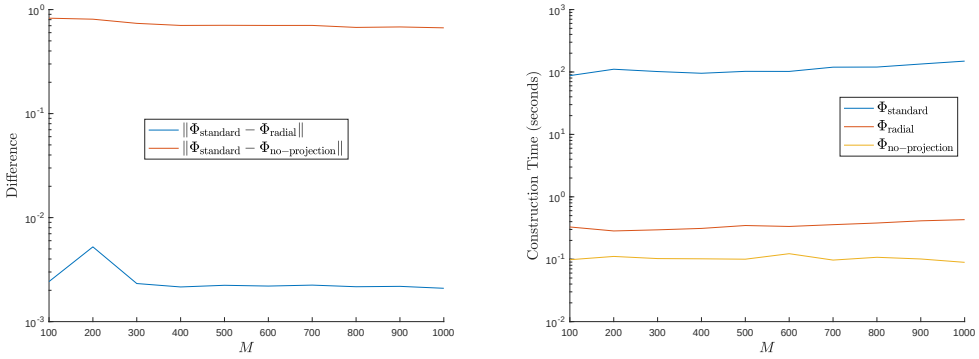

Figure 6.5: Figures comparing 3 types of measurement operators. One with a standard 2 dimensional w-projection kernel $\mathbf{\Phi}_{\text{standard}}$, a radially symmetric kernel $\mathbf{\Phi}_{\text{radial}}$, and one with no w-projection kernel $\mathbf{\Phi}_{\text{no-projection}}$. The comparisons were performed for 100 to 1000 measurements. (top) The difference in operator norms. We find that the full 2 dimensional and radially symmetric kernels are bounded to be the same within about 3×10^{-3}. We find that assuming no w-projection kernel produces a difference close to 1. (Bottom) A plot of the construction time for each operator (excluding normalization). We find that using an analytic expression for the Kaiser-Bessel with no w-projection, $\mathbf{\Phi}_{\text{no-projection}}$, is fastest for two reasons. There is no quadrature integral to calculate, and minimal amount of coefficients to store into memory. The quadrature calculation with variable kernel size means that $\mathbf{\Phi}_{\text{radial}}$ will always take more time to calculate, even for $w = 0$, which is computationally cheap for quadrature (see Figure 6.4). We find $\mathbf{\Phi}_{\text{standard}}$ is the most expensive in time to calculate. This is consistent with the number of function evaluations required to calculate each coefficient.

have the bounds of $\frac{\|(\mathbf{\Phi}^{\dagger}_{\text{standard}} - \mathbf{\Phi}^{\dagger}_{\text{radial}})\boldsymbol{y}\|_{\ell_2}}{\|\boldsymbol{y}\|_{\ell_2}} \leq 10^{-3}$, which translates to an upper bound dirty map RMS difference of the order of less than 1%. However, the difference will in principle be less. Similar can be said for generating model visibilities.

It is also clear that the construction times are dramatically different between the two. The construction time is greatly improved by the threading, since the kernel construction was performed in parallel. However, due to the small value of M, this improvement has reached saturation. It is clear in this example that construction is hundreds of times faster when using a radial symmetric kernel.

6.4.4 Imaging of the directionally dependent w-effect via the zero-spacing

The previous tests have indirectly verified that the radially symmetric w-projection kernel is consistent with the 2 dimensional w-projection kernel, suggesting that the entire degridding and gridding process is self consistent. In this section, we image the generated radially symmetric kernels directly and compare against the theoretically expected values that are independent of implementation.

In the image domain, we expect the w-projection kernel to be a chirp with the form of

$$c(l,m;w) = \mathrm{e}^{-2\pi i w(\sqrt{1-l^2-m^2}-1)}, \tag{6.41}$$

then by considering a zero length baseline (also known as an auto-correlation) with an artificial w-component, which can be done by choosing $y(0,0,w)=1$ and $\bar{w}=0$ in the measurement equation, we find that the adjoint application of the measurement operator and then taking the complex conjugate will result in

$$dde_{\text{expected}}(l,m;w) = a(l,m)\frac{c(l,m;w)}{\sqrt{1-l^2-m^2}}. \tag{6.42}$$

It follows that in the discrete setting, gridding a visibility at $(u,v)=(0,0)$ and $\bar{w}=0$ will produce the same result

$$dde_{\text{calculated}}(l_i,m_i;w) = \sqrt{N}(\mathbf{\Phi}^{\dagger}_{(u=0,v=0,w)})^{*}_{i}. \tag{6.43}$$

We calculate the average relative difference of *dde* for the imaginary and real parts, using the formula

$$\delta(q,p) = 2\left[\frac{q-p}{|q|+|p|}\right], \tag{6.44}$$

this suppresses divergences for when q or p are close to zero. We choose $a(l,m)=1$, and values of $w=10$ and $w=100$ wavelengths using an image with 4096 by 4096 pixels and a pixel height and width of 15 arcseconds. This

leads to a field of view of $17.0667^\circ \times 17.0667^\circ$. We compare using a support size linear in w, $\frac{2w}{\Delta u}$, rounded to the nearest pixel. We choose an accuracy of 10^{-6} in absolute and relative error for numerical quadrature.

Figure 6.6 and 6.7 show that the radially symmetric w-projection kernel has an error on the order of 1% for both the real and imaginary parts. Where the w-effect goes through zero in the real and imaginary parts the average relative difference diverges. It is clear that the w-projection kernel still matches the expected w-effect, and that these divergences are due to instabilities of the average relative difference for values close to zero.

We find that increasing the support size and reducing the error in numerical quadrature can reduce the average relative difference. We also find that the support size $\frac{2w}{\Delta u}$ and accuracy of 10^{-6} in absolute and relative error for numerical quadrature is sufficient for relative error on the order of 1%. However, if we do not require this accuracy, we can reduce the needed computation by reducing the support size and reducing the accuracy of the numerical quadrature.

6.5 Distributed w-stacking w-projection hybrid algorithm

Until now, it has not been realistic to generate a w-projection kernel for each individual w value for each visibility in a wide-fields of view observation. We show how this can be done, first using adaptive quadrature to calculate radially symmetric w-projection kernels tailored for each visibility and then using a MPI distributed w-stacking method to perform further image domain corrections. This allows for exact non-coplanar corrections for each visibility over wide-fields of view which hasn't been practical previously in any realistic wide-field observation. In this section, we provide a brief demonstration of using radially symmetric w-projection kernels in image reconstruction. We show for the first time that fast and accurate kernel construction, in conjunction with w-stacking, enables the ability for modeling sky curvature

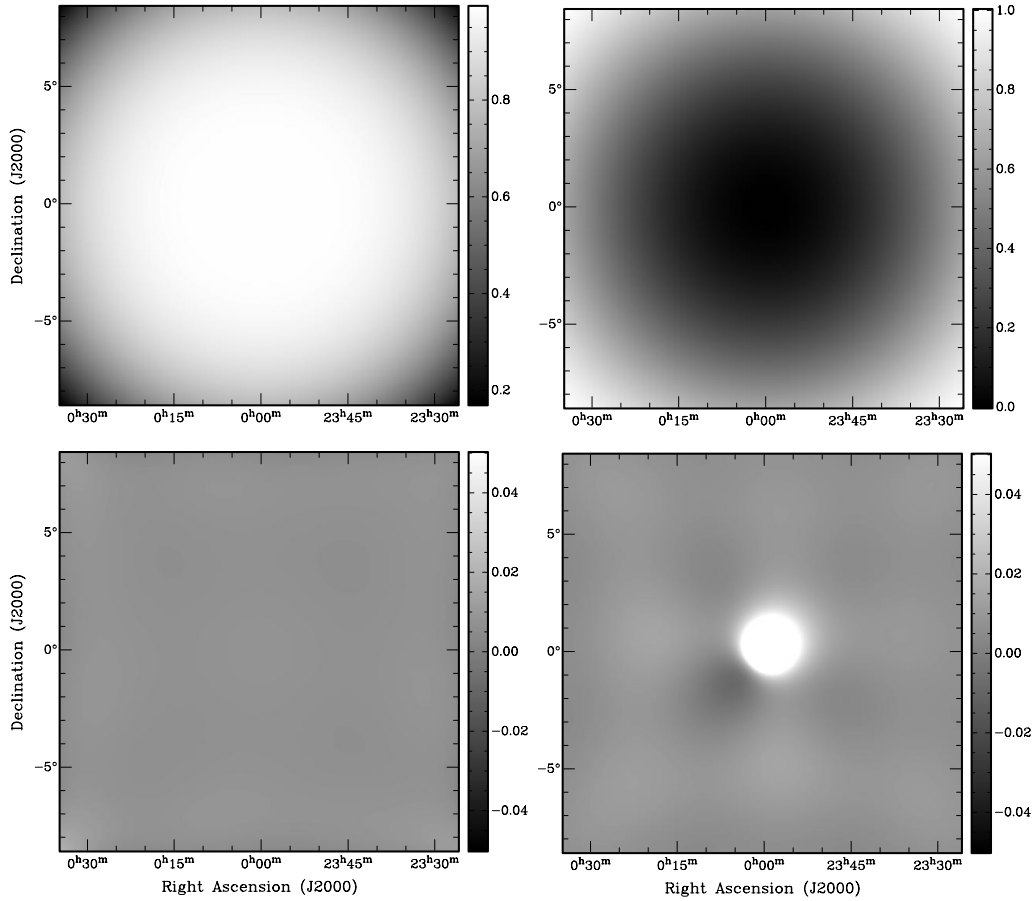

Figure 6.6: Here we show the calculated radial w-projection chirp in the image domain along with the average relative difference of the expected and calculated chirp for both the real and imaginary parts. The left column displays the real component of the chirp, and the right column the imaginary component. The top row is the radial w-projection chirp in the image domain calculated using $dde_{\text{calculated}}$ with 4096 pixels and a pixel size of 15 arcseconds, calculated for a $w = 10$ wavelengths using a kernel support size of 10 by 10 pixels. The bottom row is the average relative difference $\delta(dde_{\text{expected}}, dde_{\text{calculated}})$. We find that average relative difference is on the order of 1%, excluding where $dde_{\text{calculated}}$ and dde_{expected} are close to zero and the average relative difference diverges. This shows that the radial symmetric w-projection kernel accurately models the directionally dependent w-effect at high resolution over wide-fields of view.

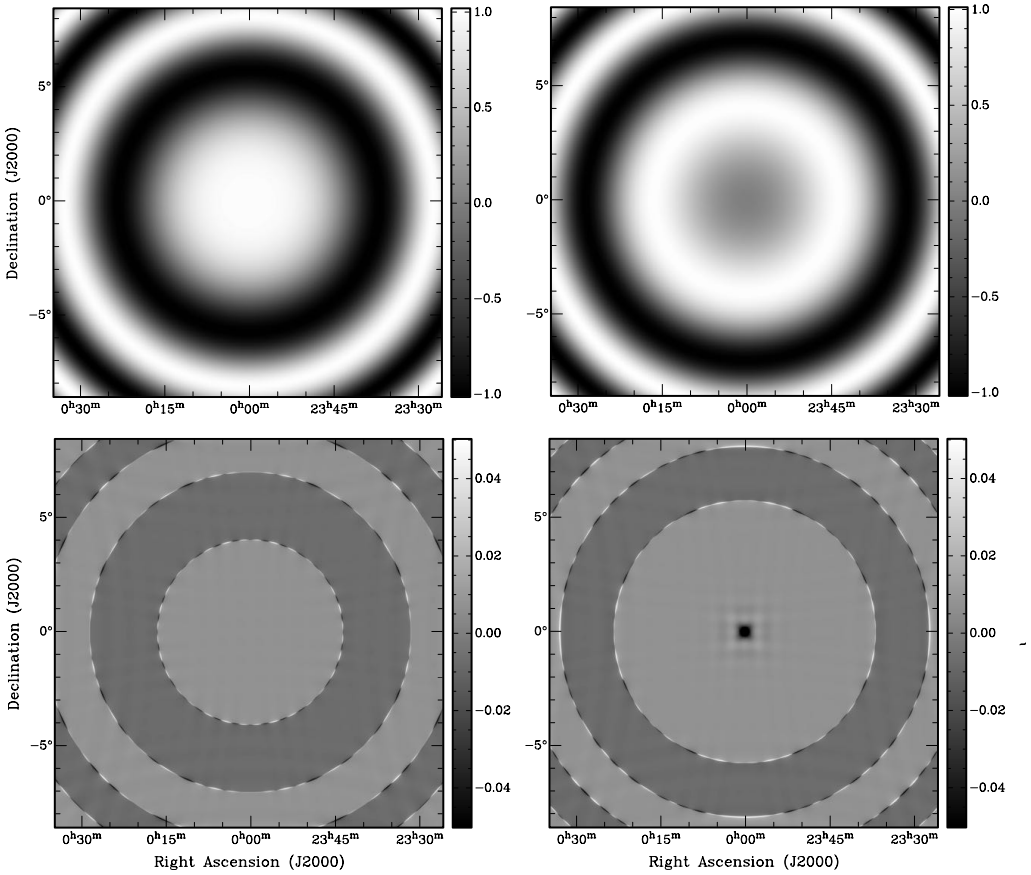

Figure 6.7: As in Figure 6.6, but for $w = 100$ wavelengths and using a kernel support size of 118 by 118 pixels. Again we find that average relative difference is on the order of 1%, demonstrating that even for larger w, the radial symmetric w-projection kernel accurately models the directionally dependent w-effect at high resolution over wide-fields of view.

and non-coplanar baselines to extremely wide-fields of view for each visibility. The kernels are calculated to an absolute accuracy of 10^{-6}, making the kernel extremely accurate for each w and very wide-fields of view. We present a hybrid of w-stacking and w-projection algorithm that uses the Message Passing Interface (MPI) standard and show its application to image reconstruction of an MWA observation of Puppis A and Vela. This algorithm is made practical with the developments of the previous section and the use of distributed computation.

6.5.1 w-stacking-w-projection measurement operator

In the past the w-stacking and the w-projection algorithms were treated as

separate methods that could only correct average w values. However, with a fast and accurate method of calculating w-projection kernels, we show that the w-stacking and the w-projection algorithms can be combined into a hybrid algorithm, allowing exact w-term correction for each visibility over wide-fields of views. First, we distribute the measurements into w-stacks using MPI. Then, we generate a w-projection kernel for each visibility in a w-stack.

The measurement operator corrects for the average w-value in the w-stack, then applies a further correction to each visibility with the w-projection. Each w-stack $\boldsymbol{y}_k$ has the measurement operator of

$$\boldsymbol{\Phi}_k = \boldsymbol{W}_k \boldsymbol{G}\boldsymbol{C}_k \boldsymbol{F} \boldsymbol{Z} \tilde{\boldsymbol{S}}_k \,. \tag{6.45}$$

The gridding correction has been modified to correct for the w-stack dependent effects, such as the average $\bar{w}_k$ and $1/n(\boldsymbol{l})$

$$\left[\tilde{\boldsymbol{S}}_k\right]_{ii} = \frac{a_k(l_i, m_i) \mathrm{e}^{-2\pi i \bar{w}_k (\sqrt{1-l_i^2-m_i^2}-1)}}{g(\sqrt{l_i^2+m_i^2})\sqrt{1-l_i^2-m_i^2}} \,. \tag{6.46}$$

We choose no primary beam effects within the stack $a_k(l_i, m_i)$. This gridding correction shifts the relative w value in the stack. This can reduce the effective w value in the stack, especially when the stack is close to the mean $\bar{w}_k$, i.e. to the value of $w_i - \bar{w}_k$[5]. This reduces the size of the support needed in the w-projection gridding kernel for each stack,

$$\begin{aligned}[\boldsymbol{G}\boldsymbol{C}_k]_{ij} = [GC](\sqrt{(u_i/\Delta u - q_{u,j})^2 + (v_i/\Delta u - q_{v,j})^2} \\ , w_i - \bar{w}_k, \Delta u) \,.\end{aligned} \tag{6.47}$$

$(q_{u,j}, q_{v,j})$ represents the nearest grid points. For each stack $\boldsymbol{y}_k \in \mathbb{C}^{M_k}$ we have the measurement equation $\boldsymbol{y}_k = \boldsymbol{\Phi}_k \boldsymbol{x}$.

To cluster the visibilities into w-stacks, it is ideal to minimize the kernel sizes across all stacks, minimizing the memory and computation costs of the

[5]Another good choice may be to minimize the median w in a stack rather than the mean w in a stack.

kernel. A k-means clustering can be used, which greatly improves performance by reducing the values of $|w_i - \bar{w}_k|^2$ across the w-stacks.

It is clear that each stack has an independent measurement equation. However, the full measurement operator is related to the stacks in the adjoint operators such that

$$\boldsymbol{x}_{\text{dirty}} = \begin{bmatrix} \boldsymbol{\Phi}_1^\dagger, & \ldots, & \boldsymbol{\Phi}_{k_{\max}}^\dagger \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{k_{\max}} \end{bmatrix} = \boldsymbol{\Phi}^\dagger \boldsymbol{y}\,. \tag{6.48}$$

When applying the w-stacks in parallel, an MPI all reduce can be used to sum over the dirty maps generated from each node. The full operator $\boldsymbol{\Phi}$ can be normalized using the power method.

6.5.2 Distributed Image Reconstruction

For image reconstruction, we use alternating direction method of multipliers as implemented in PURIFY (ADMM) [1], but built using MPI to operate on a computing cluster. The algorithm solves the same minimisation problem stated in [1]

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} \left\| \boldsymbol{\Psi}^\dagger \boldsymbol{x} \right\|_{\ell_1} \quad \text{subject to} \quad \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \epsilon\,. \tag{6.49}$$

The term $\left\| \boldsymbol{\Psi}^\dagger \boldsymbol{x} \right\|_{\ell_1}$ is a penalty on the number of non-zero wavelet coefficients, while $\|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_{\ell_2} \leq \epsilon$ is the condition that the measurements fit within a Gaussian error bound ϵ. The wavelet operator $\boldsymbol{\Psi}$ uses a wavelet dictionary of 9 wavelets, which includes a Dirac basis, and Debauches 1 to 8. Each basis in the dictionary $\boldsymbol{\Psi}_k$ has its own node, and is performed in parallel. Like with the adjoint measurement operator, an MPI reduction is performed to sum over

the nodes for the forward wavelet operator[6]

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\Psi}_1, & \ldots, & \boldsymbol{\Psi}_9 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_9 \end{bmatrix} = \boldsymbol{\Psi}\boldsymbol{\alpha}\,. \tag{6.50}$$

6.5.3 MWA observation of Puppis A and Vela

We use PURIFY [1] and the MPI w-stacking w-projection hybrid algorithm to reconstruct an observation of Puppis A performed with the MWA telescope. The observation is from the Phase 1 configuration of the MWA taken on 16 May 2013. The data was collected with XX and YY linear polarizations and then calibrated and flagged following the standard MWA data reduction process, more details on this process be found in [50]. The observation is centered at (RA = 08:19:59.99, DEC = -42:45:00), with a 112 second integration, and a central frequency of 149.115 MHz with a bandwidth of 30.720 MHz. Figure 6.8 shows a histogram of the visibilities as a function of w, the w-coverage of the observation ranges between ± 600 wavelengths. The observation contains on the order of 17 million visibilities, and the XX and YY correlations are combined to generate the Stokes I visibilities.

We implemented a k-means algorithm with MPI to sort and distribute the visibilities into 50 w-stacks, spread over 25 nodes (2 processes per node, with 1 process per stack), this sorting algorithm took approximately 5 seconds. Most w-stacks contain w-values between 0 and ± 12 wavelengths, however, some stacks contain w-values of up to 22 wavelengths. The image reconstruction was performed over a 25° by 25° field of view, using 2048^2 pixels and a pixel width of $45''$. Generating the radial w-projection kernels took close to 40 minutes, this time can be changed by using more or fewer w-stacks. Furthermore, the measurement operator was computed in parallel with over 25 nodes, and used in combination with sparse image reconstruction algorithms used in [1]. We used the Galaxy Supercomputer (located in the Pawsey Supercomputing

[6] We use the convention that $\boldsymbol{x} = \boldsymbol{\Psi}\boldsymbol{\alpha}$ and $\boldsymbol{\Psi}^\dagger \boldsymbol{x} = \boldsymbol{\alpha}$.

Centre[7]).

This observation contains the Puppis A and Vela supernova remnants, a mix of many bright compact sources and extended structures of the galactic plane. With PURIFY, we use natural weighting, as it provides the best performance in modeling both extended and compact structures. We do not include primary beam corrections when solving for the reconstructed image.

Figure 6.9 shows the dirty map, residuals, and the reconstructed image. As described in [1], we do not include the restored map, and the reconstructed image is a sky model that is the equivalent to a CLEAN component model. We also follow [1] by using the same wavelet dictionary, and scale the epsilon by 275 because the weights are relative not absolute. We can correct the scale of flux due to the field of view by using the Fourier relation $F(\Delta u u_{\text{pix}}, \Delta v v_{\text{pix}})$ being paired with $\frac{f(l/\Delta u, m/\Delta v)}{\Delta u \Delta v}$.

The dirty map and residual map were converted to Jy/Beam. To do this, we image the weights of the visibilities to obtain the peak pixel value of the point spread function, the dirty map and residuals are then divided by the peak value to convert from arbitrary units to Jy/Beam. We find that the residual map has a RMS value of approximately 190 mJy/Beam, with many of the extended structures removed from the residuals. The large scale structures of Vela are accurately removed, with only a few positive regions in the residuals where the negative side-lobes of Vela are located. This shows that the majority of the large scale structures and more compact detailed sources such as Puppis A are accurately modeled using PURIFY over a 25 by 25 degree field of view.

6.6 Conclusion

As described previously, the effect of the w-projection kernel for non-coplanar baselines ($w \neq 0$) becomes greater at larger fields of view. At these extremely wide-fields of view, the construction cost of a w-projection kernel is expensive

[7] https://www.pawsey.org.au/our-systems/

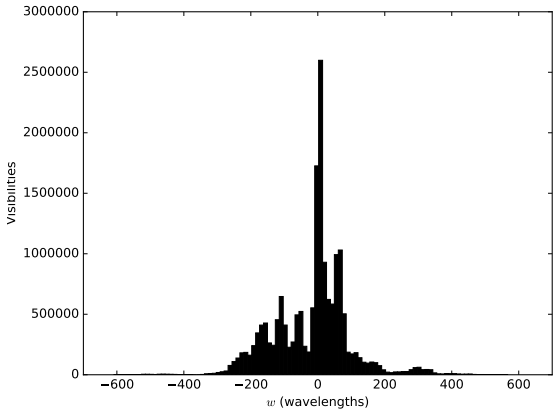

Figure 6.8: A histogram of the w-coverage of the imaged data using 100 bins. The w-values span over ± 600 wavelengths. This w-coverage represents 17,529,644 visibilities after flagging of Radio Frequency Interference (RFI) has been applied.

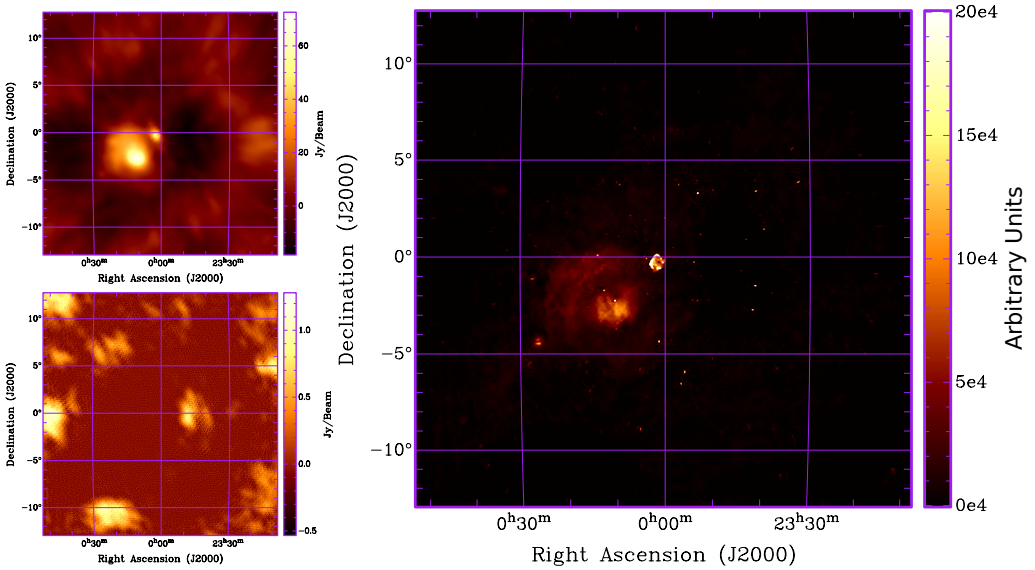

Figure 6.9: The dirty map (Top Left), residuals (Bottom Left), and sky model reconstruction (Right) of the 112 second MWA Puppis A observation centered at 149.115 MHz, using 17.5 million visibilities and an image size of 2049^2 (each pixel is 45 arcseconds and the field of view is approximately 25 by 25 degrees). This image was reconstructed using the MPI distributed w-stacking-w-projection hybrid algorithm, using the radial symmetric w-projection kernels, in conjunction with the ADMM algorithm. The RMS of the residuals is 0.189 Jy/Beam, the dynamic range of the reconstruction is 19,850.

when using FFT based methods. In this work, we have found that calculations are extremely fast and accurate using adaptive quadrature to compute a radially symmetric gridding kernel. This dramatically reduces the amount of calculations for a numerically exact kernel calculation, reducing the number of samples in the 2 dimensional case from N^2 to N in the radially symmetric case. This immediately makes such a quadrature method computationally competitive. It has low memory usage, it can be distributed in parallel, and scales to extremely wide-fields of view. Furthermore, the calculation is analytic up to a chosen numerical error, allowing the tuning of speed vs. accuracy that is not possible with FFT based methods for large images.

In this work, we developed a new technique to validate the calculation and application of a DDE. We show that by applying the modeled DDE when gridding a visibility with an artificial zero length baseline, we can provide an image of the DDE model where it can be directly verified. We applied this to the radial w-projection kernel to show the w-effect corrections to be accurate on the order of 1%. This accuracy value is tunable through the support size and the accuracy of the quadrature integration.

Wide-field modeling effects are critical not just for imaging, but need to be included during calibration of instrumental and ionospheric effects, where the w-projection can be used to simulate non-coplanar baselines over extremely wide-fields of view. This is important for generating visibilities from a sky model for non-imaging experiments. Visibilities generated from a sky model could be critical for physical scientific results. For example, any physical model of the EoR that is to be compared with data collected from a wide-field interferometer needs to have wide-field effects simulated during the comparison in image or Fourier space, just as any other instrumental effect (such as the primary beam). In summary, imaging methods are generally not important for non-imaging experiments, but the wide-field and instrumental response still needs to be considered when performing analysis with visibilities. The fast and exact correction via quadrature using a radially symmetric kernel is

new, and makes fast, exact, spherical and non co-planar baseline corrections possible with a w-stacking w-projection hybrid. The process works by first correcting for the average w-value in a stack to reduce kernel size and total computation, then correcting the exact difference for each visibility using quadrature calculated kernels. This method was then demonstrated on an MWA observation of the Puppis A and Vela supernova remnants for a 25 by 25 degree field of view and over 17.5 million measurements.

We have shown that this distributed and paralleled algorithm is extremely powerful for wide-field imaging. Furthermore, these algorithms can be accelerated using multi-threaded parallelism, i.e. General Purpose Graphics Processing Units, in addition to MPI.

With this work, we provide an important step forward in the fast and accurate evaluation of wide-field interferometric imaging, bringing us closer to solving the computational challenges of the SKA and thus realizing its enormous scientific potential.

Chapter 7

w-stacking w-projection Algorithm: Details and Improvements

Two recent developments from the previous Chapter have allowed individual correction for each data set. The first is the use of adaptive quadrature and radial symmetry to calculate w-projection kernels orders of magnitude faster than the full 2 dimensional calculation [2]. The second is the developments in distributed image reconstruction from state of the art convex optimization algorithms, which provide a natural framework for the Message Passing Interface (MPI) distribution of FFTs and degridding for radio interferometric imaging [3]. An MPI hybrid w-stacking w-projection algorithm demonstrating these developments was applied on a super computing cluster, where 17.5 million measurements were individually corrected over a 25 by 25 degree field of view from an MWA observation. Such individual correction has not been previously possible.

After reviewing the w-stacking w-projection algorithm, we provide the algorithmic details of how to distribute the measurements through a k-means clustering algorithm to improve computational performance, the use of conjugate symmetry to reduce the range of w values, and show the application of these algorithms to a larger data set to demonstrate the improvement. We

end with a discussion of future strategies for kernel calculation and adapting the algorithm to model other DDEs.

The Chapter is laid out as follows. Section 7.1 describes the distributed k-means clustering algorithm used to create the w-stacks and the reconstruction algorithm used to generate a sky model of the observed data. Section 7.2 demonstrates the application of the algorithm for this implementation on an observation of Fornax A. Section 7.3 proposes possible improvements in kernel calculation for large data sets, and discusses how other directional dependent effects can be included into the algorithm. The work is concluded in Section 7.4.

7.1 Clustering w-stacks

It is ideal to minimize the kernel sizes across all stacks, minimizing the memory and computation costs of the kernel. We develop an MPI k-means clustering algorithm which greatly improves performance by reducing the values of $|w_i - \bar{w}_k|^2$ across the w-stacks. Each MPI node finds the w-stack to which a visibility belongs, updating the cluster centers across all MPI nodes with each iteration. This is then followed by an all-to-all MPI operation to distribute the visibilities to their w-stacks. There already exist parallel and distributed k-means clustering algorithms for big data [152, 153]. The k-means w-clustering algorithm is presented in Algorithm 8. This algorithm is necessary to reduce computation and operating memory when applying the w-projection kernels by reducing the support size of each kernel.

Typically, the visibilities are read in and distributed across a computing cluster one measurement set at a time. Then the k-means algorithm (Algorithm 8) is used to assign a w-stack for each visibility. Then the visibilities are redistributed across the cluster, so that each MPI process corresponds to a w-stack.

Algorithm 8 k-means w-stacking:
The k-means algorithm sorts the visibilities into clusters (w-stacks) by minimizing the average w deviation, $(\bar{w} - w)^2$, within each cluster. The algorithm returns two arrays: $\boldsymbol{n}$ is the array of indices that labels the w-stack for each visibility; $\bar{\boldsymbol{w}}$ is the average w value within each w-stack. The algorithm requires a starting w-stack distribution $\bar{\boldsymbol{w}}^{(0)}$, which we choose to be evenly distributed between the minimum and maximum w-values. The algorithm should iterate until $\bar{\boldsymbol{w}}^{(t)}$ has converged, which we choose to be a relative difference of 10^{-3}. Note p is the index of visibility, q is the index for w-stacks, and c is the place holder for the minimum deviation for the visibility at index p. The AllSumAll(x) operation is an MPI reduction of a summation followed by broadcasting the result to all compute nodes.

given $\bar{\boldsymbol{w}}^{(0)}, \boldsymbol{n}^{(0)}, w_{\text{total}}, n_{\text{total}}, \boldsymbol{w}_{\text{sum}}, w_{\text{count}}$
repeat for $t = 1, \ldots$
  $\boldsymbol{w}_{\text{sum}} = \boldsymbol{0}$
  $\boldsymbol{w}_{\text{count}} = \boldsymbol{0}$
  repeat for $p = 1, \ldots$
    $m := 2(w_{\max} - w_{\min})^2$
    repeat for $q = 1, \ldots$
      $c := (\bar{\boldsymbol{w}}_q^{(t)} - \boldsymbol{w}_p)^2$
      if $c < m$ **then**
        $m := c$
        $\boldsymbol{n}_p^{(t+1)} = q$
      end if
    until $q > n_{\text{total}}$
    $\boldsymbol{w}_{\text{sum}_{\boldsymbol{n}_p^{(t+1)}}} = \boldsymbol{w}_{\text{sum}_{\boldsymbol{n}_p^{(t+1)}}} + \boldsymbol{w}_p$
    $\boldsymbol{w}_{\text{count}_{\boldsymbol{n}_p^{(t+1)}}} = \boldsymbol{w}_{\text{count}_{\boldsymbol{n}_p^{(t+1)}}} + 1$
  until $p > w_{\text{total}}$
  repeat for $q = 1, \ldots$
    $\bar{\boldsymbol{w}}_q^{(t+1)} = 0$
    if AllSumAll($\boldsymbol{w}_{\text{count}q}$) > 0 **then**
      $\bar{\boldsymbol{w}}_q^{(t+1)} = \text{AllSumAll}(\boldsymbol{w}_{\text{sum}q}) / \text{AllSumAll}(\boldsymbol{w}_{\text{count}q})$
    end if
  until $q > n_{\text{total}}$
until convergence

7.1.1 Conjugate symmetry

Prior to w-stacking with the k-means algorithm, conjugate symmetry may be used to restrict the w-values onto the positive w-domain. The origin of the w-effect stems from the 3d Fourier transform of a spherical shell and a horizon window, with the w component probing the Fourier coefficient of the signal along the line of sight. The sky, the horizon window, the spherical shell, and

the primary beam can all be interpreted as a real valued signal. This provides a conjugate symmetry between $-|w|$ and $+|w|$, i.e.

$$y^*(u,v,-|w|) = y(-u,-v,|w|)\,. \tag{7.1}$$

Properties of noise remain unchanged under conjugate symmetry, meaning that measurements can be restricted to positive w, i.e. $w \in \mathbb{R}_+$. Other modelled instrumental effects may need to be conjugated, which is only important when they are complex valued signals. In particular, polarized signals, e.g. Stokes Q, U, and V, are independent real valued signals. Thus, linear polarization has a slightly different relation

$$y_P^*(u,v,-|w|) = y_Q(-u,-v,|w|) - iy_U(-u,-v,|w|)\,, \tag{7.2}$$

suggesting the reflection should be done to the Stokes Q and U visibilities before combination into linear polarization, and then combined with $-i$ rather than $+i$. This combination is important for accurate polarimetirc image reconstruction [51].

7.2 Application to MWA observation of Fornax A

In this section we show an example of how conjugate symmetry allows exact non-coplanar correction to a larger data set than the previous chapter. The increased efficiency of the w-stacking due to conjugate symmetry reduces the construction time and application time of the w-projection kernels.

We use PURIFY (version 3.0.1, [7]) to perform wide-field image reconstruction of an observation of Fornax A taken with the MWA. The observation has a pointing centre of (03h 22m 41.7s, -37d 12m 30s), and the integration time is 112 seconds. Fornax A was observed using XX and YY polarizations, with the visibilities transformed into Stokes I. The bandwidth was 30.72 MHz with a central frequency of 184.955 MHz and using 768

channels, which is a standard observational mode for the MWA [154, 155]. The data reduction, including flagging and calibration, is as per [156].

To perform the reconstruction we use 50 nodes of the Grace computing cluster at University College London. Each node of Grace contains two 8 core Intel Xeon E5-2630v3 processors (16 cores total) and 64 Gigabytes of RAM.[1]

The reconstructed image is of 2048 by 2048 pixels, with a pixel width of 45 arc-seconds and a field of view of 25 by 25 degrees. The w values range between 0 and approximately 600 wavelengths for the total of 126.6 million visibilities, after conjugating the visibilities for negative w values, i.e. a range of 1200 wavelengths originally.

Sorting the visibilities into 50 w-stacks (one per MPI node) took under 5 seconds using the MPI distributed k-means algorithm described in Algorithm 8. If the average relative difference of each w-stack centre $\bar{\boldsymbol{w}}_i$ between k-means iterations is less than 10^{-3} we consider the algorithm has converged. We do not expect the w-projection algorithm performance to improve beyond this level of accuracy in clustering as a function of the number of iterations. In this case, the algorithm converged in 6 iterations.

It took a total of 15 minutes to construct a w-projection kernel for all visibilities, using quadrature accuracy of 10^{-6} in relative and absolute error, as described in the previous Chapter. The w-projection kernel construction time in the previous Chapter was 40 minutes for 50 w-stacks (over 25 compute nodes), with the same field of view and same image size, over the same range of w values, but for only 17.5 million visibilities. We find that the use of conjugate symmetry before the k-means clustering algorithm allows for more efficient computation of the w-projection kernels due to more efficient w-stacking because of the reduced range of w-values, allowing for 2.6 times faster kernel construction for approximately 7 times as many measurements (126.6 million visibilities), i.e. an overall saving of approximately by a factor of 18.

[1] More details can be found at `https://wiki.rc.ucl.ac.uk/wiki/RC_Systems#Grace_technical_specs`

Reconstruction time took 12 hours, with a total of 2475 iterations, with the FFT and wavelet operations contributing to much of this time due to the large image size. Note that we elected to run the reconstruction for a much longer time than needed to produce an acceptable image. We erred on the side of a higher number of iterations than strictly necessary in order to get a very high quality reconstruction.

The reconstructed image can be seen in Figure 7.1, which also shows the residual and dirty maps. The bright, extended source Fornax A is visible at the field centre, with the rest of the field consisting mostly of point sources. The residual map shows that the reconstruction models many of the sources in the field of view, however, the point spread function sidelobes from bright sources outside the FoV are still present in the residuals. Despite outside sources disrupting the reconstruction, the root mean squared (RMS) value of the residual map is 15 mJy/beam.

Figure 7.2 shows a zoom in of Figure 7.1, with the colour scale adjusted to show the reconstruction of Fornax A in greater detail. From the scaled residuals it is clear that this reconstruction accurately models the extended structure of Fornax A.

7.3 Improvements for the Future

We discuss two classes of possible improvements: kernel interpolations and correction for non-standard direction dependent effects.

7.3.1 Kernel interpolation

While we have shown that the use of k-means clustering and complex conjugation can aid in kernel construction, w-projection kernels can still be expensive in construction time due to the large number of coefficients in $\boldsymbol{GC}$. This construction overhead can be further reduced using interpolation methods, such as bilinear interpolation between 1 dimensional w-planes, or parametric fitting. This may allow for on the fly calculation of kernels during imaging. We discuss how a radially symmetric kernel could affect such methods

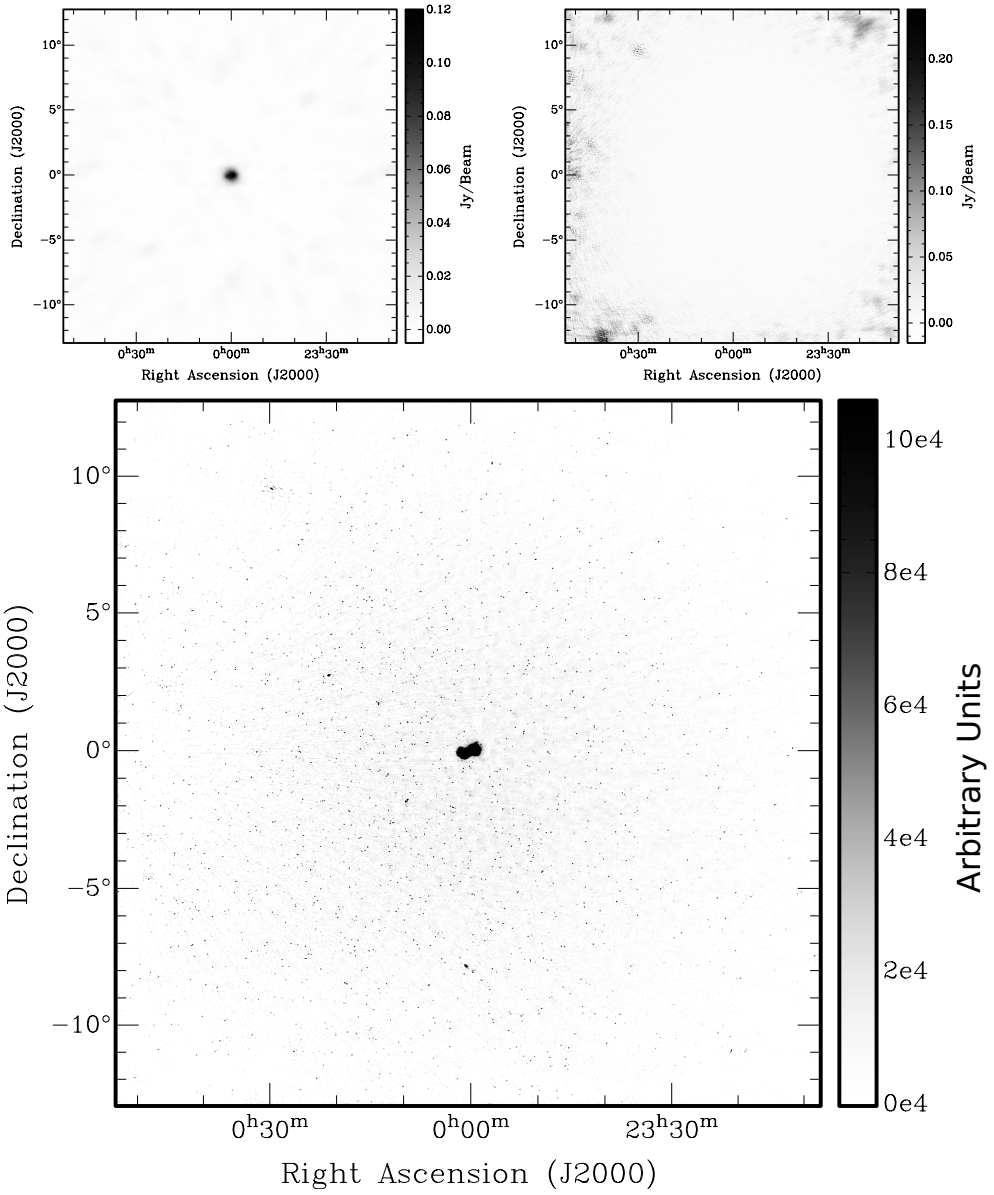

Figure 7.1: The dirty map (Top Left), residuals (Top Right), and sky model reconstruction (Bottom) of the 112 second MWA Fornax A observation centered at 184.955 MHz, using 126.6 million visibilities and an image size of 2049^2 (each pixel is 45 arcseconds and the field of view is approximately 25 by 25 degrees). This image was reconstructed using the MPI distributed w-stacking-w-projection hybrid algorithm, exploiting conjugate symmetry and the k-means clustering algorithm for distribution of w-stacks presented herein, and using the radial symmetric w-projection kernels, in conjunction with the ADMM algorithm. The dynamic range of the reconstruction is 844,000. The RMS of the residuals is approximately 15 mJy/beam over the entire field of view.

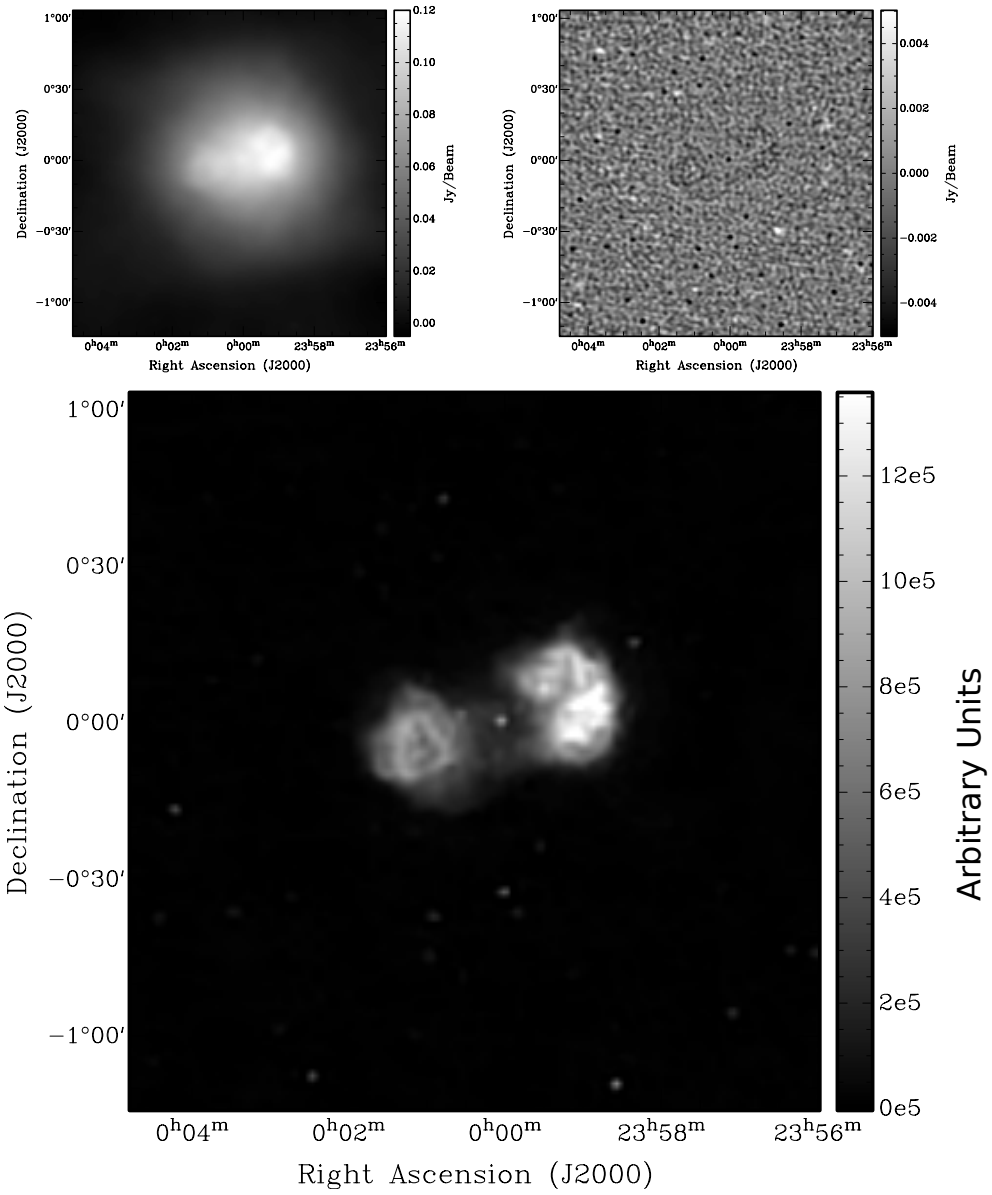

Figure 7.2: Same as Figure 7.1 zoomed view centered on Fornax A, showing the recovered structure of the double lobed radio galaxy. The residuals have been scaled to show the details. The residuals over the zoomed region have an RMS of 1.2 mJy/beam.

in the future.

7.3.1.1 w-planes: bilinear interpolation

The radially symmetric kernel allows fast and accurate calculation. It also reduces the dimensions of the kernel from 2 dimensions to 1 dimension. This allows for fast and accurate pre-sampling of the w-projection kernel directly in the uvw-domain. Pre-sampling could speed up the radially symmetric kernel construction time and allow for on the fly calculation, while reducing the total memory in stored gridding kernels as discussed below. We discuss how radial symmetry can lead to an improvement in pre-sampling by reducing memory and pre-sampling time.

A non-radially symmetric kernel would mean pre-sampling in $(u_{\text{pix}}, v_{\text{pix}}, w)$, which is a computational challenge. For $N_u \times N_v$, samples in (u, v), we would have N_w w-projection planes. This requires in total $N_u N_v N_w$ samples. The total memory required in pre-samples is $16 \times 10^{-6} \times N_u N_v N_w$[Megabytes].

With radial symmetry the w-projection kernel can be computed as a function of $(\sqrt{u_{\text{pix}}^2 + v_{\text{pix}}^2}, w)$. For N_{uv} radial samples in $\sqrt{u_{\text{pix}}^2 + v_{\text{pix}}^2}$, and N_w samples in w, we have only $N_{uv} N_w$ samples. This can be thought of as pre-computing 1 dimensional w-planes, rather than 2 dimensional w-planes. Additionally, each sample only requires a 1 dimensional integral by quadrature that reduces the pre-sampling time.

The 1 dimensional nature of the problem suggests better scaling of pre-sampling computation time and memory, allowing extremely accurate w-projection kernels. The total memory required in pre-samples is $16 \times 10^{-6} \times N_{uv} N_w$[Megabytes].

It is also worth noting that pre-sampling is only required for positive (u, v, w), since the complex conjugate can be used to estimate $(u, v, -w)$ and radial symmetry can be used for negative u and v. This leads to additional memory savings in pre-sampling.

Pre-sampling can be optimized for accuracy and storage by using an adaptive sampling density. The pre-samples could be stored permanently in

cases where kernel construction is performed repetitively.

Bilinear interpolation is computationally cheap, and could make accurate on-the-fly construction of w-projection kernels possible, which could be needed for large data such as for the Square Kilometre Array (SKA) [157]. In the case where storing the gridding kernels consumes more memory than the pre-sampled kernel, on-the-fly construction can be built into the ***GC*** operator, where bilinear interpolation is used on application. However, memory layout of the pre-samples would be important, since the sample look-up time could reduce the speed of the calculation considerably.

7.3.1.2 Function fitting

Another powerful solution to improve kernel construction costs can be found from the well-known prolate spheroidal wave function (PSWF) gridding kernels, which do not have an analytic form.

PSWFs can be defined multiple ways, such as having optimal localization of energy in both image and harmonic space, making them difficult to compute. They can be calculated directly through Sinc interpolation after solving a discrete eigenvalue problem, but this can be computationally expensive, or they can be calculated using a series expansion. However, this has not stopped radio astronomers using the PSWFs for decades, ever since the work of [65, 62] described a custom made PSWF that has been used in CASA [102], AIPS [158], MIRIAD [71], and PURIFY [29]. In [65, 62], a rational approximation is used to provide a stable and accurate fit to the PSWF, which has stood the test of time.

A similar approach can be used to provide an accurate fit to w-projection kernels. Put simply, it is possible to fit a radially symmetric kernel as a function of three parameters $\left(\sqrt{u_{\text{pix}}^2 + v_{\text{pix}}^2}, w, \Delta u\right)$, i.e. polynomial fitting. This has various advantages over the pre-sampling method, such as reduced storage, no pre-sampling time, and reduced look up time (which could be critical for on-the-fly application). However, stability and reliability of the fit is not guaranteed and would require further investigation.

7.4 Conclusion

We have discussed details of the w-stacking w-projection algorithm implementation, including details of the k-means clustering, introduction of conjugate symmetry to improve the computational efficiency of the current algorithm, and possible extensions to the current algorithms and code base to further improve efficiency and accuracy of the reconstructions.

We use the MPI distributed ADMM implementation in PURIFY to reconstruct an MWA observation of Fornax A, recovering accurate sky models of the complex source Fornax A and of point sources over the entire 25 by 25 degree field of view. We find that we can construct w-projection kernels for 7 times the number of measurements, 2.6 times faster than the time taken in the previous Chapter (an overall saving of approximately 18 times), using the same image size, field of view, and range of w values.

We conclude the work with proposals to modify the implementation of the 1 dimensional radial w projection kernels for large data sets, such as the use of kernel interpolation. Accurate correction of wide-field and instrumental effects is critical in the era of next generation radio interferometers and are vital to achieving science goals ranging from the detection of the Epoch of Reionization to accurately reconstructing cosmic magnetic fields.

Chapter 8

Balancing Compute Load for Wide-field Reconstruction

Recent novel developments in fast construction of w-projection kernels and the distributed w-stacking w-projection hybrid algorithm [76, 2] has allowed fast and accurate modeling of non-coplanar effects over extremely wide-fields of view from the MWA for over 100 million measurements [4]. The algorithm allows parallel construction of w-projection kernels while also distributing their storage for application, proving to be an effective method of tackling the most computational and memory intensive components of radio interferometric imaging [159, 160, 157, 2]. However, while this distribution reduces the size and computational cost of the w-projection kernel, it does not ensure that computational resources are being used most effectively across the compute cluster. This makes it vulnerable to bottlenecks in computation without the modifications presented in this work.

This work presents a new distributed gridding algorithm that evenly balances the computational load across a computing cluster, extending the distributed gridding methods developed in [3]. This work combines the two measurement operator algorithms described in Section 5.2.3.2 (distributing sections of the grid) and 5.2.3.1 (distributing images) to create a new novel algorithm. This method allows all w-stacks to distribute sections of their grid to different nodes for gridding and degridding using an all to all operation.

Such an approach allows full memory and computational use across the nodes of the computing cluster when performing fast Fourier transforms (FFTs) of w-stacks and when degridding with w-projection kernels, which has not been possible previously, removing resource bottlenecks when imaging wide-fields of view for large data sets. Such distributed degridding and gridding algorithms will be vital for next-generation radio interferometers with large data sets, such as the Square Kilometer Array (SKA). In particular, such an algorithm is needed for effectively correcting instrumental effects via the image and Fourier domain, while using the full performance of a computing cluster.

The remaining sections of this Chapter are as follows. Section 8.1 introduces the distributed w-stacking w-projection hybrid algorithm with compact notation. Section 8.2 discusses the computational and memory bottlenecks of this method. Section 8.3 presents the new algorithm that evenly distributes the computational load across compute nodes. Section 8.4 demonstrates the application of this algorithm that has been implemented in the interferometric imaging software package PURIFY (in an upcoming release after version 3.0.1)[1].

8.1 Distributed wide-field measurement operator

In the distributed w-stacking w-projection algorithm [2], the measurement operator corrects for the average w-value in each w-stack, then applies an extra correction to each visibility with the w-projection. Each w-stack $\boldsymbol{y}_k$ has the measurement operator of

$$\boldsymbol{\Phi}_k = \boldsymbol{W}_k [\boldsymbol{GC}]_k \boldsymbol{F} \boldsymbol{Z} \tilde{\boldsymbol{S}}_k \,. \tag{8.1}$$

[1] `https://github.com/astro-informatics/purify`

The gridding correction, $\tilde{\boldsymbol{S}}_k$, has been modified to correct for the w-stack dependent effects, such as the average w-value of the stack $\bar{w}_k$

$$\tilde{\boldsymbol{S}}_{k,ii} = \frac{a_k(l_i, m_i)\mathrm{e}^{-2\pi i \bar{w}_k(\sqrt{1-l_i^2-m_i^2}-1)}}{g(l_i^2+m_i^2)\sqrt{1-l_i^2-m_i^2}}\,. \tag{8.2}$$

We leave the option of choosing different primary beam effects in a stack $a_k(l_i, m_i)$. The chirp shifts the relative w-value in the stack indexed by k. The stacks can be clustered carefully to reduce the effective w-value in the stack, especially when the stack is close to the mean $\bar{w}_k$, i.e. to the value of $w_i - \bar{w}_k$. This reduces the size of the support needed in the w-projection gridding kernel for each stack,

$$[\boldsymbol{GC}]_{k,ip} = [GC]\Big(\sqrt{(u_i/\Delta u - q_{u,p})^2 + (v_i/\Delta u - q_{v,p})^2}, w_i - \bar{w}_k, \Delta u\Big)\,. \tag{8.3}$$

$(q_{u,p}, q_{v,p})$ represents the nearest grid points, and we use adaptive quadrature to calculate

$$\begin{aligned}[GC]\Big(\sqrt{u_{\mathrm{pix}}^2 + v_{\mathrm{pix}}^2}, w, \Delta u\Big) = \frac{2\pi}{\Delta u^2}\int_0^{\alpha/2} g(r)\mathrm{e}^{-2\pi i w(\sqrt{1-r^2/\Delta u^2}-1)} \\ \times J_0\left(2\pi r\sqrt{u_{\mathrm{pix}}^2 + v_{\mathrm{pix}}^2}\right) r\mathrm{d}r\,,\end{aligned} \tag{8.4}$$

where $g(r)$ is the radial anti-aliasing filter in the image domain (i.e. the Fourier transform of the Kaiser-Bessel function), Δu is the resolution of the Fourier grid as determined by the zero padded field of view, and $(u_{\mathrm{pix}}, v_{\mathrm{pix}})$ are the pixel coordinates on the Fourier grid.

For each stack $\boldsymbol{y}_k \in \mathbb{C}^{M_k}$ we have the measurement equation $\boldsymbol{y}_k = \boldsymbol{\Phi}_k \boldsymbol{x}$. It is clear that each stack has an independent measurement equation. However, the full measurement operator is related to the stacks in the adjoint operators such that

$$\boldsymbol{x}_{\mathrm{dirty}} = \mathrm{AllSumAll}_k\left(\boldsymbol{\Phi}_k^\dagger \boldsymbol{y}_k\right) = \boldsymbol{\Phi}^\dagger \boldsymbol{y}\,. \tag{8.5}$$

We use an MPI all-sum-all to generate the same dirty map on each node. The

full MPI operator $\tilde{\mathbf{\Phi}}$ is normalized using the power method. For further details see [2].

8.2 Bottleneck of the distributed stacking method

To minimize the time taken to perform kernel calculation and increase accuracy of the non-coplanar correction, the visibilities need to be sorted into w-stacks using a cluster algorithm. We do this by using the k-means clustering algorithm after using complex conjugation to reflect the visibilities to have positive w [4]. Because the w-stacks are clustered to minimize error, the memory and computational load of each $[\boldsymbol{GC}]_k$ has previously been ignored when assigning one stack k per compute node. When the majority of visibilities lie in only a few stacks, the total available memory and resources for construction and application of $[\boldsymbol{GC}]_k$ is bottlenecked. This is especially the case when there is one $[\boldsymbol{GC}]_k$ per MPI node. This problem is emphasized for extremely wide-fields of view and large values of w, where the w-projection kernel size scales as $\frac{2w}{\Delta u}$, with $\frac{1}{\Delta u} \propto$ field of view, and for large numbers of visibilities. Hence, these factors have a large impact on the required computational resources in kernel construction and application, as we demonstrate in Section 8.4.

In the next section we describe an algorithm that solves this bottleneck. We split the operator $[\boldsymbol{GC}]_k$ into smaller operators $[\boldsymbol{GC}]_{jk}$ that can be spread across multiple nodes j for w-stacks indexed by k. We remove the requirement that image domain correction and Fourier domain correction are applied on the same node. We restrict the index j for nodes that apply Fourier domain correction and index k for nodes that apply image domain correction. This allows even distribution of the memory load, kernel construction, and application of the operator $[\boldsymbol{GC}]$ to ensure scalability as demonstrated in Section 8.4.

8.3 All-to-all distributed measurement operator

In this section we introduce a new MPI distribution strategy for the application of a wide-field measurement operator. This process allows the FFTs of the w-stacks to be evenly distributed across all nodes while allowing the sparse matrix operations to be distributed evenly across all nodes. Communicating only the grid points that are needed for degridding minimizes communication in an intermediate all-to-all operation.

8.3.1 Distributing measurements for computational load

First the k-means algorithm is used to sort the visibilities into w-stacks $\boldsymbol{y}_k$. The visibilities of each stack $\boldsymbol{y}_k$ are distributed across MPI nodes $\boldsymbol{y}_{jk}$, where $1 \leq j \leq n_\mathrm{d}$, to evenly distribute the computation of $\boldsymbol{[GC]}$. The computational load of an individual visibility $y_{\mathsf{k}i}$ is determined by the support size

$$\mathrm{support}(w_i - \bar{w}_k, \Delta u) = \left[\max\{J_\mathrm{min}, 2(w_i - \bar{w}_k)/\Delta u\}\right]^2 , \tag{8.6}$$

where J_min is the 1d support size of the anti-aliasing kernel [2]. It is then straightforward to determine the total computational load of $\boldsymbol{[GC]}$ and then distribute it evenly across nodes j. This is done by calculating the average computational load across all nodes from $j = 1$ to $j = n_\mathrm{d}$ in order, filling each node j with visibilities until it reaches the average computational load.

In practice, it is difficult to fill each node with the exact average computational load, because each visibility has its own integral (indivisible) computational load. This can be accommodated by allowing the last node to overfill slightly and keeping the rest of the nodes under the average load. Testing has shown that the overfill amount on the last node is insignificant.

8.3.2 All-to-all distribution of Fourier grid subsections

With the computational load of $[\boldsymbol{GC}]$ distributed across the nodes, the measurement equation needs to map sections of the grid that need to be sent to each node j from each stack k to minimize communication. Without loss of generality, we let $1 \leq k, j \leq n_{\mathrm{d}}$. The MPI measurement equation reads

$$\boldsymbol{y}_{jk} = \boldsymbol{W}_{jk}[\boldsymbol{GC}]_{jk}\mathrm{AllToAll}_{jk}\left(\boldsymbol{M}_{jk}\boldsymbol{F}\boldsymbol{Z}\tilde{\boldsymbol{S}}_k\boldsymbol{x}\right), \tag{8.7}$$

where the chirp multiplication and FFT are applied on node k (assuming one $\tilde{\boldsymbol{S}}_{\boldsymbol{k}}$ per node for simplicity), the operator $\boldsymbol{M}_{jk} \in \mathbb{R}^{K_{jk} \times K}$ selects only the grid sections (of size K_j) of the FFT grid (of size K) of stack k that are needed for degridding on node j, which are then sent to node j with the MPI all-to-all operation. This is followed by degridding to the visibilities on node j that belong to stack k using $[\boldsymbol{GC}]_{jk} \in \mathbb{C}^{M_{jk} \times K_{jk}}$. In practice, $[\boldsymbol{GC}]_{jk}$ are combined into one sparse matrix on each node that has $\sum_k M_{jk}$ rows and $\sum_j K_{jk}$ columns. This entire process is visualized in Figure 8.1.

The application of the adjoint operator reads

$$\begin{aligned}\boldsymbol{x}_{\mathrm{dirty}} = \mathrm{AllSumAll}_k\Bigg(&\tilde{\boldsymbol{S}}_k^{\dagger}\boldsymbol{Z}^{\dagger}\boldsymbol{F}^{\dagger}\times \\ &\sum_{j=1}^{n_{\mathrm{d}}}\left[\boldsymbol{M}_{jk}^{\dagger}\mathrm{AllToAll}_{kj}\left([\boldsymbol{GC}]_{jk}^{\dagger}\boldsymbol{W}_{jk}^{\dagger}\boldsymbol{y}_{jk}\right)\right]\Bigg),\end{aligned} \tag{8.8}$$

where node j grids visibilities from stack k, these grid sections are sent from node j to stack k through an all-to-all operation. The grid sections from each node j are added to the full FFT grid of each stack k. An inverse FFT is applied followed by cropping of the image. Multiplication of the conjugate chirp is applied on each stack k followed by an all-sum-all of the images to produce the same dirty map on each MPI node.

Extensive unit testing has shown that the distributed computation is equivalent to the non distributed computation and the standard w-stacking w-projection algorithm. It is worth noting that when $n_{\mathrm{d}} \times K > 2^{32} - 1$, 64 bit

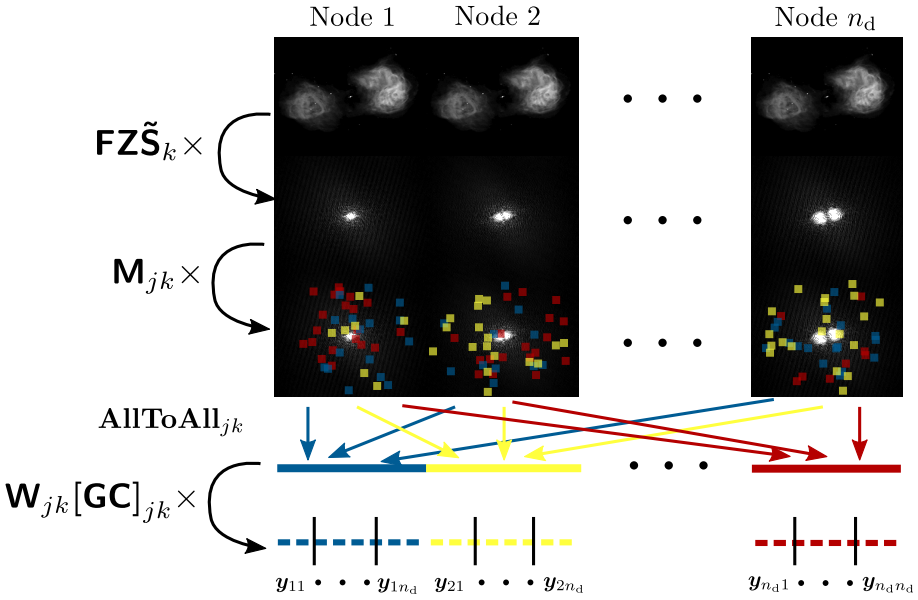

Figure 8.1: Each node starts with a copy of $\boldsymbol{x}$. The linear operation $\tilde{\boldsymbol{S}}_k$ applies the gridding correction and multiplication of the chirp on node k. Each node performs zero padding and an FFT with the operation $\boldsymbol{FZ}$. The operation $\boldsymbol{M}_{jk}$ selects sections of the FFT grid on node k that are required on node j for degridding (this is determined by the columns of $[\boldsymbol{GC}]_{jk}$). The colored squares show regions of the grid that are to be sent to each node, with each color corresponding to a value of j. The sections of the FFT grid are distributed through a distributed MPI all-to-all communication step. This is followed by the application of $[\boldsymbol{GC}]_{jk}$ for the k^{th} w-stack on node j, to interpolate the visibilities $\boldsymbol{y}_{jk}$ off of the grid, with the w-projection kernel performing the correction for the offset $w - \bar{w}_k$. The adjoint process corresponds to performing each step in reverse, followed by an all-sum-all operation over the w-stacks.

integer types are specifically needed for indexing across $n_d \times K$ FFT pixels without overflow.

8.4 Implementation and Application

In this section we demonstrate the effectiveness of evenly distributing the computational load using the algorithm presented in Section 8.3. This algorithm has been implemented in the interferometric imaging software package PURIFY using C++ and MPI, where this method is ready for an upcoming release. Similarly, to Section 5.4, we apply this algorithm to a simulated data set. However, we point out that the standard w-stacking

w-projection algorithm cannot be applied due to memory limitations and bottlenecks on each compute node, which is the purpose of this demonstration of the load-balanced operator.

To demonstrate the effectiveness of the algorithm, we simulate reconstruction of a 25 by 25 deg field of view, using a Gaussian variable sampling density in uvw following [1]. u and v are represented in radians, with a standard deviation of $\pi/3$. w is represented in wavelengths, with a standard deviation of 200 wavelengths, but was constrained to values between ± 600 wavelengths. An 1024 by 1024 pixel image of M31 is considered, where the pixel size is 90 by 90 arcseconds. We add Gaussian noise to the measurements, so that the visibilities have an input signal to noise ratio of 30 decibels [1]. We then apply the alternating direction method of multipliers (ADMM) algorithm as performed in [1, 2, 3], see Chapter 5 for more details. We used a minimal gridding kernel support size of $J_{\min} = 4$ for the Kaiser-Bessel kernel.

First we use conjugate symmetry to reflect the visibilities to have $w \geq 0$. Then we use the k-means clustering algorithm to assign each visibility to a w-stack indexed by k and to calculate each $\bar{w}_k$. Then we iterate through the visibilities to assign the computational load across the nodes, following Section 8.3. The visibilities and w-stack indexes are redistributed using an all-to-all operation. Then the w-projection kernels shown in Equation 8.4 are constructed using adaptive quadrature to an accuracy of 10^{-6} in absolute and relative error, which has shown to be accurate to 1% in the image domain [2]. This corrects each visibility for the w offset determined by $\bar{w}_k$ and the w-stack index k.

We perform reconstruction using 2 billion visibilities with 50 nodes of the Grace supercomputing cluster at University College London (UCL). Each node has two 8 core Intel Xeon E5-2630v3 processors and 64 Gigabytes of RAM[2]. Note that this is exactly the same configuration used in the recent work of [4], where an MWA Fornax A observation was reconstructed using 126 million

[2]More details can be found at `https://wiki.rc.ucl.ac.uk/wiki/RC_Systems#Grace_technical_specs`

visibilities.

The memory used to store $[\boldsymbol{GC}]$ is distributed across 50 compute nodes. The memory needed to store $[\boldsymbol{GC}]$ was approximately 21 Gigabytes on each node (3 Tb across all nodes). However, for efficient layout for memory access $[\boldsymbol{GC}]^{\dagger}$ was also stored, requiring an additional 3 Tb across all nodes. The 2 billion visibilities amounts to 32 Gigabytes spread evenly across the nodes. To store the weights and uvw-coordinates during construction of $[\boldsymbol{GC}]$ requires 64 Gigabytes of memory spread evenly over the cluster.

Sorting and distributing the visibilities took approximately 2 minutes. Kernel construction took 1 hour and 5 minutes. Application of the combined gridding and degridding operation took approximately 25 seconds. The ADMM algorithm converged in approximately 20 minutes with 9 iterations. The signal to noise ratio of the reconstruction was calculated as in [1] to be 31.49 decibels.

Applying the standard distribution method of the w-stacking w-projection hybrid algorithm was not possible for the scenario considered due to memory requirements, where each $[\boldsymbol{GC}]_k$ requires approximately 1 to 50 Gigabytes of memory. Additionally, even if there was enough memory on each node, run time would increase greatly due to lack of CPU cores on the heavily loaded nodes acting as a bottleneck. In this case the load balanced distributed method presented in this work circumvents this bottleneck in resources and enables accurate interferometric image reconstruction over extremely wide-fields of view for a larger data set than previously possible.

Chapter 9

Interferometric Imaging with the SPIDER Telescope

The Hubble Space Telescope (HST) has changed the way astronomers have looked at the Universe. The number of astronomical studies that have used observations from the HST make it one of the most important observatories in history. More than 15,000 articles have used HST data, in total collecting 738,000 citations.[1] However, telescopes such as the HST and its scientific successor, the James Webb Space Telescope (JWST), are extremely heavy and large, while being expensive in cost and power consumption. Nevertheless such next generation optical telescopes like JWST are critical to address astronomy and cosmology science goals such as answering questions about dark matter through weak lensing and understanding the history and formation of our universe.

Recently, the concept of an instrument known as the Segmented Planar Imaging Detector for Electro-optical Reconnaissance (SPIDER) has been developed [9, 161]. The SPIDER is a small-scale interferometric optical imaging device that first uses a lenslet array to measure multiple interferometer baselines, then uses photonic integrated circuits (PICs) to miniaturize the measurement acquisition. The goal of the SPIDER is to reduce the weight, cost, and power consumption of optical telescopes. Furthermore, additional

[1] See `https://www.nasa.gov/mission_pages/hubble/story/index.html`

designs have been proposed that could increase the efficiency of imaging using fewer measurements [162, 163]. Recent visibility measurements using lenslet arrays and PICs have shown to match theoretical predictions [164]. Unlike traditional optical interferometry, the SPIDER telescope can accurately retrieve both phase and amplitude information [164], making the measurement process analogous to a radio interferometer. Accurate interferometric image reconstruction methods from radio astronomy can thus be applied to image SPIDER observations.

Radio astronomy has a long history of using interferometry to push beyond the limits of resolution and size, at the computational cost of image reconstruction [11]. An interferometer is a device that measures the cross-correlation function of the signals. Interferometric imaging in the radio has proven to be a popular approach between 50 MHz and 100 GHz, with telescopes such as the Very Large Array (VLA) that have antenna arrays spread over 36 kilometers [12]. The cross-correlation between voltages from each pair of antenna is computed to generate the complex valued measurements known as visibilities. A visibility represents a Fourier coefficient for the sky brightness, with the Fourier coordinate determined by the antenna pair separation. Typically an antenna pair is known as a baseline, with the baseline length corresponding to the antenna separation [12].

Recently, sparse image reconstruction algorithms that exploit developments from the field of convex optimization have shown to improve the quality of reconstructed observations from radio interferometers considerably, on both simulations and real data [1, 165, 2, 3]. In this chapter we take recent developments from radio interferometric imaging and sparse image reconstruction, and put them into the context of the proposed SPIDER instrument. Such methodology would prove useful in future space based telescopes and space missions based on the SPIDER technology (e.g. aerial observations of planetary surfaces). Ultimately it is evident that recent algorithmic developments for radio interferometric imaging can be directly

applied to the SPIDER optical interferometer.

In Section 9.1 we introduce the background and current developments behind the SPIDER concept. Section 9.2 shows image reconstruction from a simulated SPIDER observation.

9.1 SPIDER

Key to the concept design of the SPIDER is the use of lenslets to collect signals from incoming light. These signals are combined using a PIC to produce an interferometric measurement (visibility), i.e. a Fourier coefficient of the observation. The Fourier coordinates, (u, v), are determined by the separation size in wavelengths (baseline length) between the lenslets that were used to generate the measurement, with larger separations resulting in higher resolution measurements. However, unlike radio interferometry where all possible pairs of antennas in an array can be combined in an observation, lenslets can only be paired once. If there are N_l lenslets, the lenslet array will produce $N_l/2$ correlations. This differs to the $N(N-1)/2$ correlations expected from a radio array [12, 163]. To compensate for this lenslets can be combined with the PIC to split the signal into spectral bins (channels), allowing for increased sampling coverage due to variation of baseline length over wavelength. This strategy has been successful in radio astronomy for decades, and is known as multi-frequency synthesis [12].

The concept design of the SPIDER proposed in [9] is to put a linear array of lenslets onto a PIC card. The PIC cards are mounted as radial spokes on a disc, producing a radial sampling pattern in the uv-plane (however, other sampling patterns are considered in [9]). The proposed operating wavelengths are between 500 nm and 900 nm. The operating wavelength divided by the size of a lenslet (8.75 mm) determines the field of view to be approximately between 0.5 and 1 arc minutes. The longest baseline along a spoke is 0.5 m, which is sensitive to resolutions between 0.65 and 1.2 arcseconds. Parameters of the SPIDER design adopted from [9] are listed in Table 9.1, which leads to

Table 9.1: SPIDER configuration parameters adopted from 9.

Parameter	Value
Spectral Coverage	500-900 nm
Lenslet Diameter	8.75 mm
Longest Baseline	0.5 m
Number of Lenslets per PIC spoke	24
Number of PIC spoke	37
Number of Spectral Bins	10
FoV at 500 (900) nm	35″ (65″)
Maximum Resolution at 500 (900) nm	0.7″ (1.2″)
Total Measurements	4440

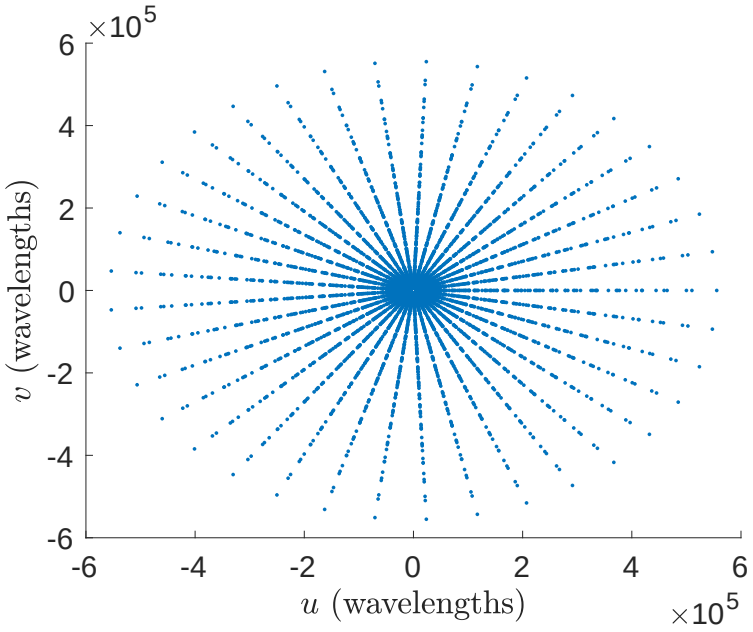

Figure 9.1: The sampling pattern of SPIDER in the uv-plane in units of wavelengths using 24 lenslets over 37 PIC cards for the combined coverage of 10 spectral bins. The sampling pattern was generated using the parameters in Table 9.1. Since the Fourier coordinates are relative to wavelength, using the spectral bins (channels) will increase the uv-coverage of the instrument substantially. The number of measurements in the single channel corresponds to 444, which makes 4440 measurements over the entire band.

the (u, v) sampling coverage shown in Figure 9.1.

9.2 Reconstructions

In this section we demonstrate reconstruction of simulated SPIDER observations using the ADMM algorithm, where a solution is found from

the constrained problem. We use the software package PURIFY[2] to perform interferometric image reconstruction, powered by the convex optimization package SOPT[3]. The SPIDER telescope is a planar interferometric telescope, and the standard planar interferometric measurement equation can be applied through gridding and degridding.

To generate the measurement operator used to simulate the observation we use the Kaiser-Bessel kernel with a support size $J = 8$ pixels to reduce aliasing error in the ground truth measurements. For reconstruction, we use a measurement operator with a kernel support size of $J = 4$ pixels. The number of pixels in $\boldsymbol{x}$ are determined by the ground truth image, $\boldsymbol{x}_{\text{GroundTruth}} \in \mathbb{R}_+^N$. We do not include the decrease in sensitivity of the SPIDER instrument away from the center of the field, but this can be included in simulations if it is well characterized. To simulate the observation we follow [1] and add i.i.d. Gaussian noise to the observational data. We define an input signal to noise ratio (ISNR) to determine the standard deviation of the Gaussian noise, where this standard deviation is defined as

$$\sigma_i = \frac{\|\boldsymbol{\Phi}\boldsymbol{x}_{\text{GroundTruth}}\|_{\ell_2}}{\sqrt{M}} \times 10^{-\frac{\text{ISNR}}{20}} . \tag{9.1}$$

The Fourier sampling pattern of the observation (i.e. the uv-coverage) is determined by the design of the SPIDER instrument and the optical spectral coverage. By combining the entire spectra it is possible to increase the sampling coverage, as explained in Section 9.1. We use the configuration of Table 9.1 (shown in Figure 9.1).

The results presented in Figure 9.2 show that an observation using the proposed SPIDER design can be effectively reconstructed using PURIFY. Reconstruction was performed using a Dirac basis and Daubechies wavelets 1 to 8. While we have used the design from [9], where the baselines lie on radial spokes, different baseline configurations may lead to higher quality

[2] `https://github.com/astro-informatics/purify`
[3] `https://github.com/astro-informatics/sopt`

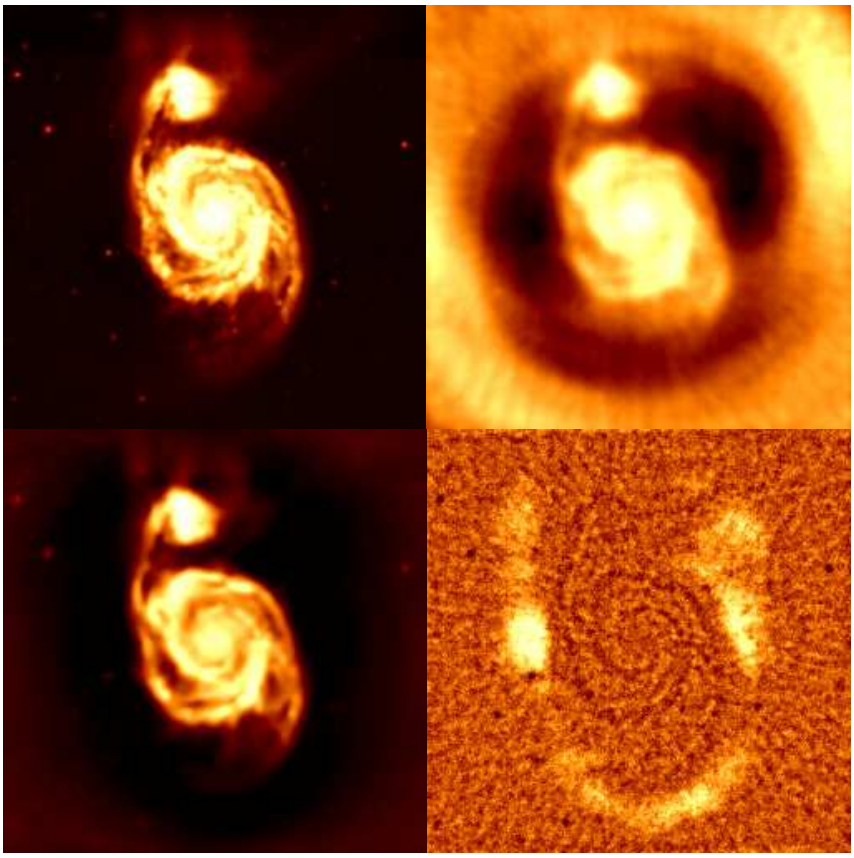

Figure 9.2: Simulation of observation and reconstruction of the spiral galaxy M51 using ADMM implemented with PURIFY, including the ground truth (top left), the observed image (top right), the PURIFY reconstruction (bottom left), and the residuals (bottom right). We used an ISNR of 30dB, a pixel size of $0.3''$, and an image size of 256 by 256 pixels, with the sampling pattern for 10 spectral bins as shown in Figure 9.1 resulting in 4440 measurements. The structure of the spiral arms and point sources are recovered well using PURIFY.

reconstruction. Depending on the structures in the ground truth sky, different baseline configurations will be more effective at sampling the sky, leading to more effective reconstruction of objects and their details. It was recently shown that the theory of compressive sensing might lead to more efficient designs [163].

In summary, we adapt recent developments in radio interferometric imaging, leveraging sparsity and convex optimisation, and show that they are effective for imaging SPIDER observations. Moreover, recent developments in efficient uncertainty quantification for radio interferometric imaging can

also be adapted for use with SPIDER [165]. The computational performance of these algorithms can be further increased using GPU multi-threading and distribution across nodes of a computing cluster [as implemeneted in PURIFY already; 3].

Chapter 10

Conclusions

There are two major challenges with next generation imaging. The first is to create accurate images of the radio sky for both compact sources and medium to large extended structures. The second challenge is to develop methods of image reconstruction that are computationally efficient enough to scale for large data sets and not require excessive computation.

This thesis has made contributions to both of these challenges. With the application of convex optimization to real data sets in Chapter 4 we showed that we can obtain detailed and high quality images of compact and extended radio sources. In Chapter 5, we describe and demonstrate implementations of computationally distributed degridding/gridding operators, wavelet transforms, and proximal operators, then use them to distribute the ADMM algorithm. This makes it possible to perform high image reconstruction to large data sets. Then, in Chapters 6 and 7, we introduced new calculation and computational distribution methods for wide-field non-coplanar interferometric telescopes that make it possible to correct each individual measurement from next generation low frequency interferometric telescopes, this was not previously possible. This shows an improvement in calculation scalability over previous wide-field correction techniques, and allows for more accurate modeling of the measurement equation leading to more accurate reconstructions. In Chapter 8, we improve the distributed computational efficiency of constructing and applying instrumental corrections

in the Fourier domain. This allows for accurate imaging of wider fields of view and larger data sets without increasing the required computational resources. Lastly, in Chapter 9 we show that these imaging developments can be applied to interferometric imaging outside of radio astronomy.

The developments listed above are not the end point for distributed image reconstruction methods. Future challenges include distribution of the wavelet transforms and FFTs for large image sizes and the ability to perform directional dependent calibration. However, it is more important that these methods are used routinely within radio astronomy. The concept of using a reconstructed model of the radio sky over the restored CLEAN image is new for radio astronomers because the reconstruction quality has not been available.

Furthermore, there are many things that need to be understood about the application of new imaging methods. This includes recognizing the impact of artifacts due to calibration error and insufficient modeling of the measurement equation, this is important for understanding scientific analysis. And lastly, the convergence criteria can have an impact on the total computation and reconstruction quality, and is something that needs to be understood, in many cases the quality is good enough for scientific analysis before convergence has been reached, suggesting that less iterations and computation could be needed depending on the image quality needed for the study.

We expect that developments from this thesis can be applied to wide band deconvolution, which is becoming increasingly important. Wide band deconvolution not only has the challenge of increased data and images due to more spectral channels, but the challenge of reconstructing the different spectra of both compact and extended radio sources. Typical radio sources are expected to be broad-band source with smooth spectra, but some cn have narrow band spectral features, making the task of modeling spectra especially challenging. However, many tools from convex optimization are built for reconstructing both compact and extended sources, which could prove valuable for reconstructing broad-band and narrow band signals.

However, most importantly, accurate and computationally scalable image reconstruction methods will be required for meeting next generation science goals. This thesis has taken a step in this direction by developing, implementing, and applying new interferometric image reconstruction that have been distributed on computing clusters. This thesis can be used as a foundation for building more efficient methods that can be applied to even larger data sets from next generation interferometric telescopes.

Bibliography

[1] L. Pratley, J. D. McEwen, M. d'Avezac, R. E. Carrillo, A. Onose, and Y. Wiaux. Robust sparse image reconstruction of radio interferometric observations with purify. MNRAS, 473:1038–1058, January 2018.

[2] L. Pratley, M. Johnston-Hollitt, and J. D. McEwen. A Fast and Exact w-stacking and w-projection Hybrid Algorithm for Wide-field Interferometric Imaging. ApJ, 874:174, April 2019.

[3] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Xiaohao Cai, David Perez-Suarez, Ilektra Christidi, and Roland Guichard. Distributed and parallel sparse convex optimization for radio interferometry with PURIFY. *Astronomy and Computing, submitted, arXiv:1903.04502*, 2019.

[4] L. Pratley, M. Johnston-Hollitt, and J. D. McEwen. w-stacking w-projection hybrid algorithm for wide-field interferometric imaging: implementation details and improvements. *PASA, submitted*, Mar 2019.

[5] L. Pratley and J. D. McEwen. Sparse Image Reconstruction for the SPIDER Optical Interferometric Telescope. *MNRAS, submitted*, March 2019.

[6] L. Pratley and J. D. McEwen. Load balancing for distributed interferometric image reconstruction. *MNRAS, submitted*, March 2019.

[7] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Rafael Carrillo,

Ilektra Christidi, Roland Guichard, David Pérez-Suárez, and Yves Wiaux. PURIFY, February 2019.

[8] Luke Pratley, Jason D. McEwen, Mayeul d'Avezac, Rafael Carrillo, Ilektra Christidi, Roland Guichard, David Pérez-Suárez, and Yves Wiaux. SOPT, February 2019.

[9] R. Kendrick et al. Flat Panel Space Based Space Surveillance Sensor. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page E45, September 2013.

[10] L. L. McCready, J. L. Pawsey, and R. Payne-Scott. Solar Radiation at Radio Frequencies and Its Relation to Sunspots. *Proceedings of the Royal Society of London Series A*, 190:357–375, August 1947.

[11] M. Ryle and A. Hewish. The synthesis of large radio telescopes. MNRAS, 120:220, 1960.

[12] A.R. Thompson, J.M. Moran, and G.W. Swenson. *Interferometry and Synthesis in Radio Astronomy*. Wiley, 2008.

[13] Fabian Walter, Elias Brinks, W. J. G. de Blok, Frank Bigiel, Jr. Kennicutt, Robert C., Michele D. Thornley, and Adam Leroy. THINGS: The H I Nearby Galaxy Survey. AJ, 136(6):2563–2647, Dec 2008.

[14] L. Koopmans, J. Pritchard, G. Mellema, J. Aguirre, K. Ahn, R. Barkana, I. van Bemmel, G. Bernardi, A. Bonaldi, F. Briggs, A. G. de Bruyn, T. C. Chang, E. Chapman, X. Chen, B. Ciardi, P. Dayal, A. Ferrara, A. Fialkov, F. Fiore, K. Ichiki, I. T. Illiev, S. Inoue, V. Jelic, M. Jones, J. Lazio, U. Maio, S. Majumdar, K. J. Mack, A. Mesinger, M. F. Morales, A. Parsons, U. L. Pen, M. Santos, R. Schneider, B. Semelin, R. S. de Souza, R. Subrahmanyan, T. Takeuchi, H. Vedantham, J. Wagg, R. Webster, S. Wyithe, K. K. Datta, and C. Trott. The Cosmic Dawn and Epoch of Reionisation with SKA. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 1, April 2015.

[15] M. Johnston-Hollitt, F. Govoni, R. Beck, S. Dehghan, L. Pratley, T. Akahori, G. Heald, I. Agudo, A. Bonafede, E. Carretti, T. Clarke, S. Colafrancesco, T. A. Ensslin, L. Feretti, B. Gaensler, M. Haverkorn, S. A. Mao, N. Oppermann, L. Rudnick, A. Scaife, D. Schnitzeler, J. Stil, A. R. Taylor, and V. Vacca. Using SKA Rotation Measures to Reveal the Mysteries of the Magnetised Universe. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 92, April 2015.

[16] R. Braun, T. Bourke, J. A. Green, E. Keane, and J. Wagg. Advancing Astrophysics with the Square Kilometre Array. In *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 174, Apr 2015.

[17] J. A. Högbom. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. A&AS, 15:417, June 1974.

[18] M. P. van Haarlem, M. W. Wise, A. W. Gunst, G. Heald, J. P. McKean, J. W. T. Hessels, A. G. de Bruyn, R. Nijboer, J. Swinbank, R. Fallows, M. Brentjens, A. Nelles, R. Beck, H. Falcke, R. Fender, J. Hörandel, L. V. E. Koopmans, G. Mann, G. Miley, H. Röttgering, B. W. Stappers, R. A. M. J. Wijers, S. Zaroubi, M. van den Akker, A. Alexov, J. Anderson, K. Anderson, A. van Ardenne, M. Arts, A. Asgekar, I. M. Avruch, F. Batejat, L. Bähren, M. E. Bell, M. R. Bell, I. van Bemmel, P. Bennema, M. J. Bentum, G. Bernardi, P. Best, L. Bîrzan, A. Bonafede, A.-J. Boonstra, R. Braun, J. Bregman, F. Breitling, R. H. van de Brink, J. Broderick, P. C. Broekema, W. N. Brouw, M. Brüggen, H. R. Butcher, W. van Cappellen, B. Ciardi, T. Coenen, J. Conway, A. Coolen, A. Corstanje, S. Damstra, O. Davies, A. T. Deller, R.-J. Dettmar, G. van Diepen, K. Dijkstra, P. Donker, A. Doorduin, J. Dromer, M. Drost, A. van Duin, J. Eislöffel, J. van Enst, C. Ferrari, W. Frieswijk, H. Gankema, M. A. Garrett, F. de Gasperin, M. Gerbers, E. de Geus, J.-M. Grießmeier, T. Grit, P. Gruppen, J. P. Hamaker,

T. Hassall, M. Hoeft, H. A. Holties, A. Horneffer, A. van der Horst, A. van Houwelingen, A. Huijgen, M. Iacobelli, H. Intema, N. Jackson, V. Jelic, A. de Jong, E. Juette, D. Kant, A. Karastergiou, A. Koers, H. Kollen, V. I. Kondratiev, E. Kooistra, Y. Koopman, A. Koster, M. Kuniyoshi, M. Kramer, G. Kuper, P. Lambropoulos, C. Law, J. van Leeuwen, J. Lemaitre, M. Loose, P. Maat, G. Macario, S. Markoff, J. Masters, R. A. McFadden, D. McKay-Bukowski, H. Meijering, H. Meulman, M. Mevius, E. Middelberg, R. Millenaar, J. C. A. Miller-Jones, R. N. Mohan, J. D. Mol, J. Morawietz, R. Morganti, D. D. Mulcahy, E. Mulder, H. Munk, L. Nieuwenhuis, R. van Nieuwpoort, J. E. Noordam, M. Norden, A. Noutsos, A. R. Offringa, H. Olofsson, A. Omar, E. Orrú, R. Overeem, H. Paas, M. Pandey-Pommier, V. N. Pandey, R. Pizzo, A. Polatidis, D. Rafferty, S. Rawlings, W. Reich, J.-P. de Reijer, J. Reitsma, G. A. Renting, P. Riemers, E. Rol, J. W. Romein, J. Roosjen, M. Ruiter, A. Scaife, K. van der Schaaf, B. Scheers, P. Schellart, A. Schoenmakers, G. Schoonderbeek, M. Serylak, A. Shulevski, J. Sluman, O. Smirnov, C. Sobey, H. Spreeuw, M. Steinmetz, C. G. M. Sterks, H.-J. Stiepel, K. Stuurwold, M. Tagger, Y. Tang, C. Tasse, I. Thomas, S. Thoudam, M. C. Toribio, B. van der Tol, O. Usov, M. van Veelen, A.-J. van der Veen, S. ter Veen, J. P. W. Verbiest, R. Vermeulen, N. Vermaas, C. Vocks, C. Vogt, M. de Vos, E. van der Wal, R. van Weeren, H. Weggemans, P. Weltevrede, S. White, S. J. Wijnholds, T. Wilhelmsson, O. Wucknitz, S. Yatawatta, P. Zarka, A. Zensus, and J. van Zwieten. LOFAR: The LOw-Frequency ARray. A&A, 556:A2, August 2013.

[19] S. J. Tingay, R. Goeke, J. D. Bowman, D. Emrich, S. M. Ord, D. A. Mitchell, M. F. Morales, T. Booler, B. Crosse, R. B. Wayth, C. J. Lonsdale, S. Tremblay, D. Pallot, T. Colegate, A. Wicenec, N. Kudryavtseva, W. Arcus, D. Barnes, G. Bernardi, F. Briggs, S. Burns, J. D. Bunton, R. J. Cappallo, B. E. Corey, A. Deshpande, L. Desouza,

B. M. Gaensler, L. J. Greenhill, P. J. Hall, B. J. Hazelton, D. Herne, J. N. Hewitt, M. Johnston-Hollitt, D. L. Kaplan, J. C. Kasper, B. B. Kincaid, R. Koenig, E. Kratzenberg, M. J. Lynch, B. Mckinley, S. R. Mcwhirter, E. Morgan, D. Oberoi, J. Pathikulangara, T. Prabu, R. A. Remillard, A. E. E. Rogers, A. Roshi, J. E. Salah, R. J. Sault, N. Udaya-Shankar, F. Schlagenhaufer, K. S. Srivani, J. Stevens, R. Subrahmanyan, M. Waterson, R. L. Webster, A. R. Whitney, A. Williams, C. L. Williams, and J. S. B. Wyithe. The Murchison Widefield Array: The Square Kilometre Array Precursor at Low Radio Frequencies. PASA, 30:7, January 2013.

[20] A. W. Hotan, J. D. Bunton, L. Harvey-Smith, B. Humphreys, B. D. Jeffs, T. Shimwell, J. Tuthill, M. Voronkov, G. Allen, S. Amy, K. Ardern, P. Axtens, L. Ball, K. Bannister, S. Barker, T. Bateman, R. Beresford, D. Bock, R. Bolton, M. Bowen, B. Boyle, R. Braun, S. Broadhurst, D. Brodrick, K. Brooks, M. Brothers, A. Brown, C. Cantrall, G. Carrad, J. Chapman, W. Cheng, A. Chippendale, Y. Chung, F. Cooray, T. Cornwell, E. Davis, L. de Souza, D. DeBoer, P. Diamond, P. Edwards, R. Ekers, I. Feain, D. Ferris, R. Forsyth, R. Gough, A. Grancea, N. Gupta, J. C. Guzman, G. Hampson, C. Haskins, S. Hay, D. Hayman, S. Hoyle, C. Jacka, C. Jackson, S. Jackson, K. Jeganathan, S. Johnston, J. Joseph, R. Kendall, M. Kesteven, D. Kiraly, B. Koribalski, M. Leach, E. Lenc, E. Lensson, L. Li, S. Mackay, A. Macleod, T. Maher, M. Marquarding, N. McClure-Griffiths, D. McConnell, S. Mickle, P. Mirtschin, R. Norris, S. Neuhold, A. Ng, J. O'Sullivan, J. Pathikulangara, S. Pearce, C. Phillips, R. Y. Qiao, J. E. Reynolds, A. Rispler, P. Roberts, D. Roxby, A. Schinckel, R. Shaw, M. Shields, M. Storey, T. Sweetnam, E. Troup, B. Turner, A. Tzioumis, T. Westmeier, M. Whiting, C. Wilson, T. Wilson, K. Wormnes, and X. Wu. The Australian Square Kilometre Array Pathfinder: System Architecture and Specifications of the Boolardy

Engineering Test Array. PASA, 31:e041, November 2014.

[21] 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*, 1(1), 2013.

[22] R. Maartens, F. B. Abdalla, M. Jarvis, and M. G. Santos. Overview of Cosmology with the SKA. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 16, April 2015.

[23] Y. Wiaux, L. Jacques, G. Puy, A. M. M. Scaife, and P. Vandergheynst. Compressed sensing imaging techniques for radio interferometry. MNRAS, 395:1733–1742, May 2009.

[24] R. E. Carrillo, J. D. McEwen, and Y. Wiaux. Sparsity Averaging Reweighted Analysis (SARA): a novel algorithm for radio-interferometric imaging. MNRAS, 426:1223–1234, October 2012.

[25] R. E. Carrillo, J. D. McEwen, D. Van De Ville, J.-P. Thiran, and Y. Wiaux. Sparsity Averaging for Compressive Imaging. *IEEE Signal Processing Letters*, 20:591–594, June 2013.

[26] J. D. McEwen and Y. Wiaux. Compressed sensing for wide-field radio interferometric imaging. MNRAS, 413:1318–1332, May 2011.

[27] Y. Wiaux, G. Puy, Y. Boursier, and P. Vandergheynst. Spread spectrum for imaging techniques in radio interferometry. MNRAS, 400:1029–1038, December 2009.

[28] L. Wolz, J. D. McEwen, F. B. Abdalla, R. E. Carrillo, and Y. Wiaux. Revisiting the spread spectrum effect in radio interferometric imaging: a sparse variant of the w-projection algorithm. MNRAS, 436:1993–2003, December 2013.

[29] R. E. Carrillo, J. D. McEwen, and Y. Wiaux. PURIFY: a new approach to radio-interferometric imaging. MNRAS, 439:3591–3604, April 2014.

[30] A. Onose, R. E. Carrillo, A. Repetti, J. D. McEwen, J.-P. Thiran, J.-C. Pesquet, and Y. Wiaux. Scalable splitting algorithms for big-data interferometric imaging in the SKA era. MNRAS, 462:4314–4335, November 2016.

[31] F. Li, T. J. Cornwell, and F. de Hoog. The application of compressive sampling to radio astronomy. I. Deconvolution. A&A, 528:A31, April 2011.

[32] A. Dabbech, C. Ferrari, D. Mary, E. Slezak, O. Smirnov, and J. S. Kenyon. MORESANE: MOdel REconstruction by Synthesis-ANalysis Estimators. A sparse deconvolution algorithm for radio interferometric imaging. A&A, 576:A7, April 2015.

[33] H. Garsden, J. N. Girard, J. L. Starck, S. Corbel, C. Tasse, A. Woiselle, J. P. McKean, A. S. van Amesfoort, J. Anderson, I. M. Avruch, R. Beck, M. J. Bentum, P. Best, F. Breitling, J. Broderick, M. Brüggen, H. R. Butcher, B. Ciardi, F. de Gasperin, E. de Geus, M. de Vos, S. Duscha, J. Eislöffel, D. Engels, H. Falcke, R. A. Fallows, R. Fender, C. Ferrari, W. Frieswijk, M. A. Garrett, J. Grießmeier, A. W. Gunst, T. E. Hassall, G. Heald, M. Hoeft, J. Hörandel, A. van der Horst, E. Juette, A. Karastergiou, V. I. Kondratiev, M. Kramer, M. Kuniyoshi, G. Kuper, G. Mann, S. Markoff, R. McFadden, D. McKay-Bukowski, D. D. Mulcahy, H. Munk, M. J. Norden, E. Orru, H. Paas, M. Pandey-Pommier, V. N. Pandey, G. Pietka, R. Pizzo, A. G. Polatidis, A. Renting, H. Röttgering, A. Rowlinson, D. Schwarz, J. Sluman, O. Smirnov, B. W. Stappers, M. Steinmetz, A. Stewart, J. Swinbank, M. Tagger, Y. Tang, C. Tasse, S. Thoudam, C. Toribio, R. Vermeulen, C. Vocks, R. J. van Weeren, S. J. Wijnholds, M. W. Wise, O. Wucknitz, S. Yatawatta, P. Zarka, and A. Zensus. LOFAR sparse image reconstruction. A&A, 575:A90, March 2015.

[34] F. Li, S. Brown, T. J. Cornwell, and F. de Hoog. The application of

compressive sampling to radio astronomy. II. Faraday rotation measure synthesis. A&A, 531:A126, July 2011.

[35] X. H. Sun, L. Rudnick, T. Akahori, C. S. Anderson, M. R. Bell, J. D. Bray, J. S. Farnes, S. Ideguchi, K. Kumazaki, T. O'Brien, S. P. O'Sullivan, A. M. M. Scaife, R. Stepanov, J. Stil, K. Takahashi, R. J. van Weeren, and M. Wolleben. Comparison of Algorithms for Determination of Rotation Measure and Faraday Structure. I. 1100-1400 MHz. AJ, 149:60, February 2015.

[36] A. R. Thompson. Fundamentals of Radio Interferometry. In G. B. Taylor, C. L. Carilli, and R. A. Perley, editors, *Synthesis Imaging in Radio Astronomy II*, volume 180 of *Astronomical Society of the Pacific Conference Series*, page 11, 1999.

[37] J. D. McEwen and A. M. M. Scaife. Simulating full-sky interferometric observations. MNRAS, 389:1163–1178, September 2008.

[38] T. D. Carozzi and G. Woan. A generalized measurement equation and van Cittert-Zernike theorem for wide-field radio astronomical interferometry. MNRAS, 395:1558–1568, May 2009.

[39] O. M. Smirnov. Revisiting the radio interferometer measurement equation. IV. A generalized tensor formalism. A&A, 531:A159, July 2011.

[40] D. C. Price and O. M. Smirnov. Generalized formalisms of the radio interferometer measurement equation. MNRAS, 449:107–118, May 2015.

[41] F. Zernike. The concept of degree of coherence and its application to optical problems. *Physica*, 5:785–795, August 1938.

[42] T. J. Cornwell and K. F. Evans. A simple maximum entropy deconvolution algorithm. A&A, 143:77–83, February 1985.

[43] K. A. Marsh and J. M. Richardson. The objective function implicit in the CLEAN algorithm. A&A, 182:174–178, August 1987.

[44] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[45] B. G. Clark. An efficient implementation of the algorithm 'CLEAN'. A&A, 89:377, September 1980.

[46] F. R. Schwab. Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry. AJ, 89:1076–1081, July 1984.

[47] D. G. Steer, P. E. Dewdney, and M. R. Ito. Enhancements to the deconvolution algorithm 'CLEAN'. A&A, 137:159–165, August 1984.

[48] R. J. Sault, L. Staveley-Smith, and W. N. Brouw. An approach to interferometric mosaicing. A&AS, 120:375–384, December 1996.

[49] T. J. Cornwell. Multiscale CLEAN Deconvolution of Radio Synthesis Images. *IEEE Journal of Selected Topics in Signal Processing*, 2:793–801, November 2008.

[50] A. R. Offringa, B. McKinley, N. Hurley-Walker, F. H. Briggs, R. B. Wayth, D. L. Kaplan, M. E. Bell, L. Feng, A. R. Neben, J. D. Hughes, J. Rhee, T. Murphy, N. D. R. Bhat, G. Bernardi, J. D. Bowman, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, A. Ewall-Wice, B. M. Gaensler, R. Goeke, L. J. Greenhill, B. J. Hazelton, L. Hindson, M. Johnston-Hollitt, D. C. Jacobs, J. C. Kasper, E. Kratzenberg, E. Lenc, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, N. Kudryavtseva, D. Oberoi, S. M. Ord, B. Pindor, P. Procopio, T. Prabu, J. Riding, D. A. Roshi, N. U. Shankar, K. S. Srivani, R. Subrahmanyan, S. J. Tingay, M. Waterson,

R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. WSCLEAN: an implementation of a fast, generic wide-field imager for radio astronomy. MNRAS, 444:606–619, October 2014.

[51] L. Pratley and M. Johnston-Hollitt. An improved method for polarimetric image restoration in interferometry. MNRAS, June 2016.

[52] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, feb 2006.

[53] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure and Appl. Math.*, 59(8):1207–1223, March 2006.

[54] D.L. Donoho. Compressed sensing. *ieeeit*, 52(4):1289–1306, apr 2006.

[55] E. J. Candes and M. B. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25:21–30, March 2008.

[56] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[57] Patrick L. Combettes and Jean-Christophe Pesquet. *Proximal Splitting Methods in Signal Processing*, pages 185–212. Springer New York, New York, NY, 2011.

[58] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing Sparsity by Reweighted L1 Minimization. *ArXiv e-prints*, November 2007.

[59] D. S. Briggs, F. R. Schwab, and R. A. Sramek. Imaging. In G. B. Taylor, C. L. Carilli, and R. A. Perley, editors, *Synthesis Imaging in Radio Astronomy II*, volume 180 of *Astronomical Society of the Pacific Conference Series*, page 127, 1999.

[60] Edmund Taylor Whittaker. Xviii.—on the functions which are represented by the expansions of the interpolation-theory. *Proceedings of the Royal Society of Edinburgh*, 35:181–194, 1915.

[61] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[62] F. R. Schwab. Optimal Gridding. VLA SCIENTIFIC MEMORANDUM 132, National Radio Astronomy Observatory, Charlottesville, Virginia, 1980.

[63] J. A. Fessler and B. P. Sutton. Nonuniform fast fourier transforms using min-max interpolation. *IEEE Transactions on Signal Processing*, 51:560–574, February 2003.

[64] E. W Greisen. The Effects of Various Convolving Functions on Aliasing and Relative Signal-to-Noise Ratios. VLA SCIENTIFIC MEMORANDUM 131, National Radio Astronomy Observatory, Charlottesville, Virginia, 1979.

[65] F. R. Schwab. Suppression of Aliasing by Convolutional Gridding Schemes. VLA SCIENTIFIC MEMORANDUM 129, National Radio Astronomy Observatory, Charlottesville, Virginia, 1978.

[66] J. A. Stratton. Spheroidal Functions. *Proceedings of the National Academy of Science*, 21:51–56, January 1935.

[67] David Slepian and Henry O Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—i. *Bell System Technical Journal*, 40(1):43–63, 1961.

[68] Henry J Landau and Henry O Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—ii. *Bell System Technical Journal*, 40(1):65–84, 1961.

[69] Henry J Landau and Henry O Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty—iii: The dimension of the space of essentially time-and band-limited signals. *Bell System Technical Journal*, 41(4):1295–1336, 1962.

[70] F. R. Schwab. Optimal Gridding of Visibility Data in Radio Interferometry. In J. A. Roberts, editor, *Indirect Imaging. Measurement and Processing for Indirect Imaging*, pages 333–346, 1984.

[71] R. J. Sault, P. J. Teuben, and M. C. H. Wright. A Retrospective View of MIRIAD. In R. A. Shaw, H. E. Payne, and J. J. E. Hayes, editors, *Astronomical Data Analysis Software and Systems IV*, volume 77 of *Astronomical Society of the Pacific Conference Series*, page 433, 1995.

[72] E. W Greisen. The Creation of AIPS. AIPS MEMORANDUM 100, National Radio Astronomy Observatory, Charlottesville, Virginia, 1998.

[73] J. I. Jackson, C. H. Meyer, D. G. Nishimura, and A. Macovski. Selection of a convolution function for fourier inversion using gridding [computerised tomography application]. *IEEE Transactions on Medical Imaging*, 10(3):473–478, Sep 1991.

[74] M. P. van Haarlem, M. W. Wise, A. W. Gunst, G. Heald, J. P. McKean, J. W. T. Hessels, A. G. de Bruyn, R. Nijboer, J. Swinbank, R. Fallows, M. Brentjens, A. Nelles, R. Beck, H. Falcke, R. Fender, J. Hörandel, L. V. E. Koopmans, G. Mann, G. Miley, H. Röttgering, B. W. Stappers, R. A. M. J. Wijers, S. Zaroubi, M. van den Akker, A. Alexov, J. Anderson, K. Anderson, A. van Ardenne, M. Arts, A. Asgekar, I. M. Avruch, F. Batejat, L. Bähren, M. E. Bell, M. R. Bell, I. van Bemmel, P. Bennema, M. J. Bentum, G. Bernardi, P. Best, L. Bîrzan, A. Bonafede, A.-J. Boonstra, R. Braun, J. Bregman, F. Breitling, R. H. van de Brink, J. Broderick, P. C. Broekema, W. N. Brouw, M. Brüggen, H. R. Butcher, W. van Cappellen, B. Ciardi, T. Coenen, J. Conway,

A. Coolen, A. Corstanje, S. Damstra, O. Davies, A. T. Deller, R.-J. Dettmar, G. van Diepen, K. Dijkstra, P. Donker, A. Doorduin, J. Dromer, M. Drost, A. van Duin, J. Eislöffel, J. van Enst, C. Ferrari, W. Frieswijk, H. Gankema, M. A. Garrett, F. de Gasperin, M. Gerbers, E. de Geus, J.-M. Grießmeier, T. Grit, P. Gruppen, J. P. Hamaker, T. Hassall, M. Hoeft, H. A. Holties, A. Horneffer, A. van der Horst, A. van Houwelingen, A. Huijgen, M. Iacobelli, H. Intema, N. Jackson, V. Jelic, A. de Jong, E. Juette, D. Kant, A. Karastergiou, A. Koers, H. Kollen, V. I. Kondratiev, E. Kooistra, Y. Koopman, A. Koster, M. Kuniyoshi, M. Kramer, G. Kuper, P. Lambropoulos, C. Law, J. van Leeuwen, J. Lemaitre, M. Loose, P. Maat, G. Macario, S. Markoff, J. Masters, R. A. McFadden, D. McKay-Bukowski, H. Meijering, H. Meulman, M. Mevius, E. Middelberg, R. Millenaar, J. C. A. Miller-Jones, R. N. Mohan, J. D. Mol, J. Morawietz, R. Morganti, D. D. Mulcahy, E. Mulder, H. Munk, L. Nieuwenhuis, R. van Nieuwpoort, J. E. Noordam, M. Norden, A. Noutsos, A. R. Offringa, H. Olofsson, A. Omar, E. Orrú, R. Overeem, H. Paas, M. Pandey-Pommier, V. N. Pandey, R. Pizzo, A. Polatidis, D. Rafferty, S. Rawlings, W. Reich, J.-P. de Reijer, J. Reitsma, G. A. Renting, P. Riemers, E. Rol, J. W. Romein, J. Roosjen, M. Ruiter, A. Scaife, K. van der Schaaf, B. Scheers, P. Schellart, A. Schoenmakers, G. Schoonderbeek, M. Serylak, A. Shulevski, J. Sluman, O. Smirnov, C. Sobey, H. Spreeuw, M. Steinmetz, C. G. M. Sterks, H.-J. Stiepel, K. Stuurwold, M. Tagger, Y. Tang, C. Tasse, I. Thomas, S. Thoudam, M. C. Toribio, B. van der Tol, O. Usov, M. van Veelen, A.-J. van der Veen, S. ter Veen, J. P. W. Verbiest, R. Vermeulen, N. Vermaas, C. Vocks, C. Vogt, M. de Vos, E. van der Wal, R. van Weeren, H. Weggemans, P. Weltevrede, S. White, S. J. Wijnholds, T. Wilhelmsson, O. Wucknitz, S. Yatawatta, P. Zarka, A. Zensus, and J. van Zwieten. LOFAR: The LOw-Frequency ARray. A&A, 556:A2, August 2013.

[75] D. R. DeBoer, A. R. Parsons, J. E. Aguirre, P. Alexander, Z. S. Ali, A. P. Beardsley, G. Bernardi, J. D. Bowman, R. F. Bradley, C. L. Carilli, C. Cheng, E. de Lera Acedo, J. S. Dillon, A. Ewall-Wice, G. Fadana, N. Fagnoni, R. Fritz, S. R. Furlanetto, B. Glendenning, B. Greig, J. Grobbelaar, B. J. Hazelton, J. N. Hewitt, J. Hickish, D. C. Jacobs, A. Julius, M. Kariseb, S. A. Kohn, T. Lekalake, A. Liu, A. Loots, D. MacMahon, L. Malan, C. Malgas, M. Maree, Z. Martinot, N. Mathison, E. Matsetela, A. Mesinger, M. F. Morales, A. R. Neben, N. Patra, S. Pieterse, J. C. Pober, N. Razavi-Ghods, J. Ringuette, J. Robnett, K. Rosie, R. Sell, C. Smith, A. Syce, M. Tegmark, N. Thyagarajan, P. K. G. Williams, and H. Zheng. Hydrogen Epoch of Reionization Array (HERA). PASP, 129(4):045001, April 2017.

[76] 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*, 2:647–657, November 2008.

[77] C. Tasse, S. van der Tol, J. van Zwieten, G. van Diepen, and S. Bhatnagar. Applying full polarization A-Projection to very wide field of view instruments: An imager for LOFAR. A&A, 553:A105, May 2013.

[78] C Tasse, B Hugo, M Mirmont, O Smirnov, M Atemkeng, L Bester, MJ Hardcastle, R Lakhoo, S Perkins, and T Shimwell. Faceting for direction-dependent spectral deconvolution. *Astronomy & Astrophysics*, 611:A87, 2018.

[79] J. G. Ables. Maximum Entropy Spectral Analysis. A&AS, 15:383, June 1974.

[80] A. Dabbech, A. Onose, A. Abdulaziz, R. A. Perley, O. M. Smirnov, and Y. Wiaux. Cygnus A super-resolved via convex optimization from VLA data. MNRAS, 476:2853–2866, May 2018.

[81] M. Pereyra, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Maximum-a-posteriori estimation with unknown regularisation parameters. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 230–234, Aug 2015.

[82] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. In *2006 14th European Signal Processing Conference*, pages 1–5, Sept 2006.

[83] K. Akiyama, S. Ikeda, M. Pleau, V. L. Fish, F. Tazaki, K. Kuramochi, A. E. Broderick, J. Dexter, M. Mościbrodzka, M. Gowanlock, M. Honma, and S. S. Doeleman. Superresolution Full-polarimetric Imaging for Radio Interferometry with Sparse Modeling. AJ, 153:159, April 2017.

[84] J. Birdi, A. Repetti, and Y. Wiaux. Sparse interferometric Stokes imaging under the polarization constraint (Polarized SARA). MNRAS, 478:4442–4463, August 2018.

[85] J. Birdi, A. Repetti, and Y. Wiaux. Scalable algorithm for polarization constrained sparse interferometric stokes imaging. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 465–469, July 2018.

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

[87] J. Deguignet, A. Ferrari, D. Mary, and C. Ferrari. Distributed multi-frequency image reconstruction for radio-interferometry. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 1483–1487, Aug 2016.

[88] S. Boyd, S.P. Boyd, L. Vandenberghe, and Cambridge University Press.

Convex Optimization. Berichte über verteilte messysteme. Cambridge University Press, 2004.

[89] N. Komodakis and J. Pesquet. Playing with duality: An overview of recent primal?dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, 32(6):31–54, Nov 2015.

[90] J.N. Girard, H. Garsden, J.L. Starck, S. Corbel, A. Woiselle, C. Tasse, J.P. McKean, and J. Bobin. Sparse representations and convex optimization as tools for lofar radio interferometric imaging. *Journal of Instrumentation*, 10(08):C08013, 2015.

[91] N.S. Papageorgiou and S.T. Kyritsi-Yiallourou. *Handbook of Applied Analysis*. Advances in Mechanics and Mathematics. Springer US, 2009.

[92] Pontus Giselsson and Stephen Boyd. Metric selection in fast dual forward–backward splitting. *Automatica*, 62:1 – 10, 2015.

[93] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[94] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[95] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[96] P. L. Combettes and J. Pesquet. A douglas–rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, Dec 2007.

[97] R. Bot and C. Hendrich. A douglas–rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators. *SIAM Journal on Optimization*, 23(4):2541–2565, 2013.

[98] J. Yang and Y. Zhang. Alternating direction algorithms for l1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.

[99] S. Setzer, G. Steidl, and T. Teuber. Deblurring poissonian images by split bregman techniques. *Journal of Visual Communication and Image Representation*, 21(3):193 – 199, 2010.

[100] Patrick L. Combettes, Đinh Dũng, and Bang Công Vũ. Dualization of signal recovery problems. *Set-Valued and Variational Analysis*, 18(3):373–404, Dec 2010.

[101] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[102] J. P. McMullin, B. Waters, D. Schiebel, W. Young, and K. Golap. CASA Architecture and Applications. In R. A. Shaw, F. Hill, and D. J. Bell, editors, *Astronomical Data Analysis Software and Systems XVI*, volume 376 of *Astronomical Society of the Pacific Conference Series*, page 127, October 2007.

[103] S. Bhatnagar and T. J. Cornwell. Scale sensitive deconvolution of interferometric images. I. Adaptive Scale Pixel (Asp) decomposition. A&A, 426:747–754, November 2004.

[104] L. Zhang, S. Bhatnagar, U. Rau, and M. Zhang. Efficient implementation of the adaptive scale pixel decomposition algorithm. A&A, 592:A128, August 2016.

[105] D. Oberoi, J. Attridge, and S. Doeleman. PSFs and best fits beams in AIPS and MIRIAD. LOFAR MEMORANDUM 6, Haystack Observatory, Massachusetts Institute of Technology, WESTFORD, MASSACHUSETTS, 2003.

[106] P. Patel, I. Harrison, S. Makhathini, F. B. Abdalla, D. Bacon, M. Brown, I. Heywood, M. Jarvis, and O. Smirnov. Weak Lensing Simulations for the SKA. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 30, April 2015.

[107] N. Hurley-Walker, J. R. Callingham, P. J. Hancock, T. M. O. Franzen, L. Hindson, A. D. Kapińska, J. Morgan, A. R. Offringa, R. B. Wayth, C. Wu, Q. Zheng, T. Murphy, M. E. Bell, K. S. Dwarakanath, B. For, B. M. Gaensler, M. Johnston-Hollitt, E. Lenc, P. Procopio, L. Staveley-Smith, R. Ekers, J. D. Bowman, F. Briggs, R. J. Cappallo, A. A. Deshpande, L. Greenhill, B. J. Hazelton, D. L. Kaplan, C. J. Lonsdale, S. R. McWhirter, D. A Mitchell, M. F. Morales, E. Morgan, D. Oberoi, S. M. Ord, T. Prabu, N. Udaya Shankar, K. S. Srivani, R. Subrahmanyan, S. J. Tingay, R. L. Webster, A. Williams, and C. L. Williams. Galactic and extragalactic all-sky murchison widefield array (gleam) survey i: A low-frequency extragalactic catalogue. *Monthly Notices of the Royal Astronomical Society*, 2016.

[108] L. Pratley, M. Johnston-Hollitt, S. Dehghan, and M. Sun. Using head-tail galaxies to constrain the intracluster magnetic field: an in-depth study of PKS J0334-3900. MNRAS, 432:243–257, June 2013.

[109] D.S. Briggs. *High Fidelity Deconvolution of Moderately Resolved Sources*. D. Briggs, 1995.

[110] Peter J. Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

[111] D.C. Hoaglin, F. Mosteller, and J.W. Tukey. *Understanding robust and exploratory data analysis*. Wiley Classics Library Editions. Wiley, 2000.

[112] R. B. Wayth, E. Lenc, M. E. Bell, J. R. Callingham, K. S. Dwarakanath, T. M. O. Franzen, B.-Q. For, B. Gaensler, P. Hancock, L. Hindson, N. Hurley-Walker, C. A. Jackson, M. Johnston-Hollitt, A. D. Kapińska, B. McKinley, J. Morgan, A. R. Offringa, P. Procopio, L. Staveley-Smith, C. Wu, Q. Zheng, C. M. Trott, G. Bernardi, J. D. Bowman, F. Briggs, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, R. Goeke, L. J. Greenhill, B. J. Hazelton, D. L. Kaplan, J. C. Kasper, E. Kratzenberg, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, D. Oberoi, S. M. Ord, T. Prabu, A. E. E. Rogers, A. Roshi, N. U. Shankar, K. S. Srivani, R. Subrahmanyan, S. J. Tingay, M. Waterson, R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. GLEAM: The GaLactic and Extragalactic All-Sky MWA Survey. PASA, 32:e025, June 2015.

[113] D. C. Jacobs, B. J. Hazelton, C. M. Trott, J. S. Dillon, B. Pindor, I. S. Sullivan, J. C. Pober, N. Barry, A. P. Beardsley, G. Bernardi, J. D. Bowman, F. Briggs, R. J. Cappallo, P. Carroll, B. E. Corey, A. de Oliveira-Costa, D. Emrich, A. Ewall-Wice, L. Feng, B. M. Gaensler, R. Goeke, L. J. Greenhill, J. N. Hewitt, N. Hurley-Walker, M. Johnston-Hollitt, D. L. Kaplan, J. C. Kasper, H. Kim, E. Kratzenberg, E. Lenc, J. Line, A. Loeb, C. J. Lonsdale, M. J. Lynch, B. McKinley, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, A. R. Neben, N. Thyagarajan, D. Oberoi, A. R. Offringa, S. M. Ord, S. Paul, T. Prabu, P. Procopio, J. Riding, A. E. E. Rogers, A. Roshi, N. Udaya Shankar, S. K. Sethi, K. S. Srivani, R. Subrahmanyan, M. Tegmark, S. J. Tingay, M. Waterson, R. B. Wayth, R. L. Webster, A. R. Whitney, A. Williams, C. L. Williams, C. Wu, and J. S. B. Wyithe. The Murchison Widefield Array 21 cm Power Spectrum Analysis Methodology. ApJ, 825:114, July 2016.

[114] E. Lenc, B. M. Gaensler, X. H. Sun, E. M. Sadler, A. G. Willis, N. Barry, A. P. Beardsley, M. E. Bell, G. Bernardi, J. D. Bowman, F. Briggs, J. R. Callingham, R. J. Cappallo, P. Carroll, B. E. Corey, A. de Oliveira-Costa, A. A. Deshpande, J. S. Dillon, K. S. Dwarkanath, D. Emrich, A. Ewall-Wice, L. Feng, B.-Q. For, R. Goeke, L. J. Greenhill, P. Hancock, B. J. Hazelton, J. N. Hewitt, L. Hindson, N. Hurley-Walker, M. Johnston-Hollitt, D. C. Jacobs, A. D. Kapinska, D. L. Kaplan, J. C. Kasper, H.-S. Kim, E. Kratzenberg, J. Line, A. Loeb, C. J. Lonsdale, M. J. Lynch, B. McKinley, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, J. Morgan, T. Murphy, A. R. Neben, D. Oberoi, A. R. Offringa, S. M. Ord, S. Paul, B. Pindor, J. C. Pober, T. Prabu, P. Procopio, J. Riding, A. E. E. Rogers, A. Roshi, N. Udaya Shankar, S. K. Sethi, K. S. Srivani, L. Staveley-Smith, R. Subrahmanyan, I. S. Sullivan, M. Tegmark, N. Thyagarajan, S. J. Tingay, C. Trott, M. Waterson, R. B. Wayth, R. L. Webster, A. R. Whitney, A. Williams, C. L. Williams, C. Wu, J. S. B. Wyithe, and Q. Zheng. Low frequency observations of linearly polarized structures in the interstellar medium near the south Galactic pole. *ArXiv e-prints*, July 2016.

[115] T. Murphy, M. E. Bell, D. L. Kaplan, B. M. Gaensler, A. R. Offringa, E. Lenc, N. Hurley-Walker, G. Bernardi, J. D. Bowman, F. Briggs, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, R. Goeke, L. J. Greenhill, B. J. Hazelton, J. N. Hewitt, M. Johnston-Hollitt, J. C. Kasper, E. Kratzenberg, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, D. Oberoi, S. M. Ord, T. Prabu, A. E. E. Rogers, D. A. Roshi, N. U. Shankar, K. S. Srivani, R. Subrahmanyan, S. J. Tingay, M. Waterson, R. B. Wayth, R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. Limits on low-frequency radio emission from southern exoplanets with the Murchison Widefield Array. MNRAS, 446:2560–2565, January 2015.

[116] A. R. Offringa, C. M. Trott, N. Hurley-Walker, M. Johnston-Hollitt,

B. McKinley, N. Barry, A. P. Beardsley, J. D. Bowman, F. Briggs, P. Carroll, J. S. Dillon, A. Ewall-Wice, L. Feng, B. M. Gaensler, L. J. Greenhill, B. J. Hazelton, J. N. Hewitt, D. C. Jacobs, H.-S. Kim, P. Kittiwisit, E. Lenc, J. Line, A. Loeb, D. A. Mitchell, M. F. Morales, A. R. Neben, S. Paul, B. Pindor, J. C. Pober, P. Procopio, J. Riding, S. K. Sethi, N. U. Shankar, R. Subrahmanyan, I. S. Sullivan, M. Tegmark, N. Thyagarajan, S. J. Tingay, R. B. Wayth, R. L. Webster, and J. S. B. Wyithe. Parametrizing Epoch of Reionization foregrounds: a deep survey of low-frequency point-source spectra with the Murchison Widefield Array. MNRAS, 458:1057–1070, May 2016.

[117] Lakshmi Saripalli, Ravi Subrahmanyan, and N. Udaya Shankar. A case for renewed activity in the giant radio galaxy j0116–473. *The Astrophysical Journal*, 565(1):256, 2002.

[118] L. Hindson, M. Johnston-Hollitt, N. Hurley-Walker, K. Buckley, J. Morgan, E. Carretti, K. S. Dwarakanath, M. Bell, G. Bernardi, N. D. R. Bhat, J. D. Bowman, F. Briggs, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, A. Ewall-Wice, L. Feng, B. M. Gaensler, R. Goeke, L. J. Greenhill, B. J. Hazelton, D. Jacobs, D. L. Kaplan, J. C. Kasper, E. Kratzenberg, N. Kudryavtseva, E. Lenc, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, B. McKinley, D. A. Mitchell, M. F. Morales, E. Morgan, D. Oberoi, S. M. Ord, B. Pindor, T. Prabu, P. Procopio, A. R. Offringa, J. Riding, A. E. E. Rogers, A. Roshi, N. U. Shankar, K. S. Srivani, R. Subrahmanyan, S. J. Tingay, M. Waterson, R. B. Wayth, R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. The First Murchison Widefield Array low-frequency radio observations of cluster scale non-thermal emission: the case of Abell 3667. MNRAS, 445:330–346, November 2014.

[119] G. K. Miley, G. C. Perola, P. C. van der Kruit, and H. van der Laan.

Active Galaxies with Radio Trails in Clusters. Nature, 237:269–272, June 1972.

[120] J. E. Gunn and J. R. Gott, III. On the Infall of Matter Into Clusters of Galaxies and Some Effects on Their Evolution. ApJ, 176:1, August 1972.

[121] E. Freeland, R. F. Cardoso, and E. Wilcots. Bent-Double Radio Sources as Probes of Intergalactic Gas. ApJ, 685:858–862, October 2008.

[122] E. M. Douglass, E. L. Blanton, T. E. Clarke, S. W. Randall, and J. D. Wing. The Merger Environment of the Wide Angle Tail Hosting Cluster A562. ApJ, 743:199, December 2011.

[123] C. Pfrommer and T. W. Jones. Radio Galaxy NGC 1265 Unveils the Accretion Shock Onto the Perseus Galaxy Cluster. ApJ, 730:22, March 2011.

[124] L. Pratley, M. Johnston-Hollitt, S. Dehghan, and M. Sun. Using radio jets of PKS J0334-3900 to probe the intra-cluster medium in A3135. In F. Massaro, C. C. Cheung, E. Lopez, and A. Siemiginowska, editors, *Extragalactic Jets from Every Angle*, volume 313 of *IAU Symposium*, pages 301–302, March 2015.

[125] G Brunetti, S Giacintucci, R Cassano, W Lane, D Dallacasa, T Venturi, NE Kassim, G Setti, WD Cotton, and M Markevitch. A low-frequency radio halo associated with a cluster of galaxies. *Nature*, 455(7215):944–947, 2008.

[126] S. Shakouri, M. Johnston-Hollitt, and G. W. Pratt. The ATCA REXCESS Diffuse Emission Survey (ARDES) - I. Detection of a giant radio halo and a likely radio relic. MNRAS, 459:2525–2538, July 2016.

[127] G. Martinez Aviles, C. Ferrari, M. Johnston-Hollitt, L. Pratley, G. Macario, T. Venturi, G. Brunetti, R. Cassano, D. Dallacasa, H. T.

Intema, S. Giacintucci, G. Hurier, N. Aghanim, M. Douspis, and M. Langer. ATCA observations of the MACS-Planck Radio Halo Cluster Project. I. New detection of a radio halo in PLCK G285.0-23.7. A&A, 595:A116, November 2016.

[128] R. Cassano, S. Ettori, G. Brunetti, S. Giacintucci, G. W. Pratt, T. Venturi, R. Kale, K. Dolag, and M. Markevitch. Revisiting Scaling Relations for Giant Radio Halos in Galaxy Clusters. ApJ, 777:141, November 2013.

[129] R. Cassano, G. Bernardi, G. Brunetti, M. Brüggen, T. Clarke, D. Dallacasa, K. Dolag, S. Ettori, S. Giacintucci, C. Giocoli, M. Gitti, M. Johnston-Hollitt, R. Kale, M. Markevich, R. Norris, M. P. Pommier, G. Pratt, H. J. A. Rottgering, and T. Venturi. Cluster Radio Halos at the crossroads between astrophysics and cosmology in the SKA era. *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, page 73, April 2015.

[130] M. Snir, W. Gropp, S. Otto, S. Huss-Lederman, J. Dongarra, and D. Walker. *MPI–the Complete Reference: The MPI core*. MPI.: The Complete Reference : The MPI Core. Mass, 1998.

[131] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier analysis and applications*, 4(3):247–269, 1998.

[132] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[133] Pavan Yalamanchili, Umar Arshad, Zakiuddin Mohammed, Pradeep Garigipati, Peter Entschev, Brian Kloppenborg, James Malcolm, and John Melonakos. ArrayFire - A high performance software library for parallel computing with an easy-to-use API, 2015.

[134] E. W Greisen. AIPS FITS File Format (revised). AIPS

MEMORANDUM 117, National Radio Astronomy Observatory, Charlottesville, Virginia, 2016.

[135] J. L. Pawsey, R. Payne-Scott, and L. L. McCready. Radio-Frequency Energy from the Sun. Nature, 157:158–159, February 1946.

[136] M. Ryle and D. D. Vonberg. An Investigation of Radio-Frequency Radiation from the Sun. *Proceedings of the Royal Society of London Series A*, 193:98–120, April 1948.

[137] U. J. Schwarz. Mathematical-statistical Description of the Iterative Beam Removing Technique (Method CLEAN). A&A, 65:345, April 1978.

[138] S. Bhatnagar, T. J. Cornwell, K. Golap, and J. M. Uson. Correcting direction-dependent gains in the deconvolution of radio interferometric images. A&A, 487:419–429, August 2008.

[139] A. Scaife. PDR.02.05.03 Imaging Pipeline. Ska sdp scientific memorandum, 2015.

[140] T. J. Cornwell, B. Humphreys, E. Lenc, V. Voronkov, and M. Whiting. ASKAP-SW-0020: ASKAP Science Processing. Askap memorandum, 2011.

[141] S. van der Tol, B. Veenboer, and A.R. Offringa. Image domain gridding: a fast method for convolutional resampling of visibilities. *Astronomy & Astrophysics*, apr 2018.

[142] B. Merry. Approximating W projection as a separable kernel. MNRAS, 456:1761–1766, February 2016.

[143] S Vembu. Fourier transformation of the n-dimensional radial delta function. *The Quarterly Journal of Mathematics*, 12(1):165–168, 1961.

[144] S. D. Poisson. *Mémoire sur l'intégration de quelques équations linéaires aux différences partielles, et particuliérement de l'équation générale du mouvement des fluides élastiques*, volume 3. Firmin Didot, 1820.

[145] M. A. Parseval. *Mémoire sur les séries et sur l'intégration complète d'une équation aux différences partillies linéaires du second ordre, á coé'fficiens constans*. 1805.

[146] R. D. Ekers and A. H. Rots. Short Spacing Synthesis from a Primary Beam Scanned Interferometer. In C. van Schooneveld, editor, *IAU Colloq. 49: Image Formation from Coherence Functions in Astronomy*, volume 76 of *Astrophysics and Space Science Library*, page 61, 1979.

[147] M. Birkinshaw. Radially-symmetric Fourier Transforms. In D. R. Crabtree, R. J. Hanisch, and J. Barnes, editors, *Astronomical Data Analysis Software and Systems III*, volume 61 of *Astronomical Society of the Pacific Conference Series*, page 249, 1994.

[148] A.C. Genz and A.A. Malik. Remarks on algorithm 006: An adaptive algorithm for numerical integration over an n-dimensional rectangular region. *Journal of Computational and Applied Mathematics*, 6(4):295 – 302, 1980.

[149] Jarle Berntsen, Terje O. Espelid, and Alan Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Trans. Math. Softw.*, 17(4):437–451, December 1991.

[150] Rank Ernst. Adaptive h-, p- and hp-versions for boundary integral element methods. *International Journal for Numerical Methods in Engineering*, 28(6):1335–1349, 1989.

[151] A. Dabbech, L. Wolz, L. Pratley, J. D. McEwen, and Y. Wiaux. The w-effect in interferometric imaging: from a fast sparse measurement operator to superresolution. MNRAS, 471:4300–4313, November 2017.

[152] Kilian Stoffel and Abdelkader Belkoniene. Parallel k/h-means clustering for large data sets. In Patrick Amestoy, Philippe Berger, Michel Daydé,

Daniel Ruiz, Iain Duff, Valérie Frayssé, and Luc Giraud, editors, *Euro-Par'99 Parallel Processing*, pages 1451–1454, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[153] C.C. Aggarwal and C.K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis, 2013.

[154] T. Prabu, K. S. Srivani, D. A. Roshi, P. A. Kamini, S. Madhavi, D. Emrich, B. Crosse, A. J. Williams, M. Waterson, A. A. Deshpande, N. U. Shankar, R. Subrahmanyan, F. H. Briggs, R. F. Goeke, S. J. Tingay, M. Johnston-Hollitt, G. M. R, E. H. Morgan, J. Pathikulangara, J. D. Bunton, G. Hampson, C. Williams, S. M. Ord, R. B. Wayth, D. Kumar, M. F. Morales, L. deSouza, E. Kratzenberg, D. Pallot, R. McWhirter, B. J. Hazelton, W. Arcus, D. G. Barnes, G. Bernardi, T. Booler, J. D. Bowman, R. J. Cappallo, B. E. Corey, L. J. Greenhill, D. Herne, J. N. Hewitt, D. L. Kaplan, J. C. Kasper, B. B. Kincaid, R. Koenig, C. J. Lonsdale, M. J. Lynch, D. A. Mitchell, D. Oberoi, R. A. Remillard, A. E. Rogers, J. E. Salah, R. J. Sault, J. B. Stevens, S. Tremblay, R. L. Webster, A. R. Whitney, and S. B. Wyithe. A digital-receiver for the Murchison Widefield Array. *Experimental Astronomy*, 39:73–93, March 2015.

[155] S. M. Ord, B. Crosse, D. Emrich, D. Pallot, R. B. Wayth, M. A. Clark, S. E. Tremblay, W. Arcus, D. Barnes, M. Bell, G. Bernardi, N. D. R. Bhat, J. D. Bowman, F. Briggs, J. D. Bunton, R. J. Cappallo, B. E. Corey, A. A. Deshpande, L. deSouza, A. Ewell-Wice, L. Feng, R. Goeke, L. J. Greenhill, B. J. Hazelton, D. Herne, J. N. Hewitt, L. Hindson, N. Hurley-Walker, D. Jacobs, M. Johnston-Hollitt, D. L. Kaplan, J. C. Kasper, B. B. Kincaid, R. Koenig, E. Kratzenberg, N. Kudryavtseva, E. Lenc, C. J. Lonsdale, M. J. Lynch, B. McKinley, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, D. Oberoi,

A. Offringa, J. Pathikulangara, B. Pindor, T. Prabu, P. Procopio, R. A. Remillard, J. Riding, A. E. E. Rogers, A. Roshi, J. E. Salah, R. J. Sault, N. Udaya Shankar, K. S. Srivani, J. Stevens, R. Subrahmanyan, S. J. Tingay, M. Waterson, R. L. Webster, A. R. Whitney, A. Williams, C. L. Williams, and J. S. B. Wyithe. The Murchison Widefield Array Correlator. PASA, 32:e006, March 2015.

[156] B. McKinley, R. Yang, M. López-Caniego, F. Briggs, N. Hurley-Walker, R. B. Wayth, A. R. Offringa, R. Crocker, G. Bernardi, P. Procopio, B. M. Gaensler, S. J. Tingay, M. Johnston-Hollitt, M. McDonald, M. Bell, N. D. R. Bhat, J. D. Bowman, R. J. Cappallo, B. E. Corey, A. A. Deshpande, D. Emrich, A. Ewall-Wice, L. Feng, R. Goeke, L. J. Greenhill, B. J. Hazelton, J. N. Hewitt, L. Hindson, D. Jacobs, D. L. Kaplan, J. C. Kasper, E. Kratzenberg, N. Kudryavtseva, E. Lenc, C. J. Lonsdale, M. J. Lynch, S. R. McWhirter, D. A. Mitchell, M. F. Morales, E. Morgan, D. Oberoi, S. M. Ord, B. Pindor, T. Prabu, J. Riding, A. E. E. Rogers, D. A. Roshi, N. Udaya Shankar, K. S. Srivani, R. Subrahmanyan, M. Waterson, R. L. Webster, A. R. Whitney, A. Williams, and C. L. Williams. Modelling of the spectral energy distribution of Fornax A: leptonic and hadronic production of high-energy emission from the radio lobes. MNRAS, 446:3478–3491, February 2015.

[157] C. Hollitt, M. Johnston-Hollitt, S. Dehghan, M. Frean, and T. Butler-Yeoman. An Overview of the SKA Science Analysis Pipeline. In N. P. F. Lorente, K. Shortridge, and R. Wayth, editors, *Astronomical Data Analysis Software and Systems XXV*, volume 512 of *Astronomical Society of the Pacific Conference Series*, page 367, December 2017.

[158] E. W. Greisen. AIPS, the VLA, and the VLBA. In A. Heck, editor, *Information Handling in Astronomy - Historical Vistas*, volume 285 of *Astrophysics and Space Science Library*, page 109, March 2003.

[159] P. Wortmann. Gridding Computational Intensity. SKA SDP SCIENTIFIC MEMORANDUM 28, 2016.

[160] P. Braam and P. Wortmann. Kernel Prototyping SOW. Ska sdp scientific memorandum, 2016.

[161] A. Duncan et al. SPIDER: Next Generation Chip Scale Imaging Sensor. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 27, 2015.

[162] G. Liu, D. Wen, and Z. Song. Rearranging the lenslet array of the compact passive interference imaging system with high resolution. In *AOPC 2017: Space Optics and Earth Imaging and Space Navigation*, volume 10463 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 1046310, October 2017.

[163] G. Liu, D.-S. Wen, and Z.-X. Song. System design of an optical interferometer based on compressive sensing. MNRAS, 478:2065–2073, July 2018.

[164] Tiehui Su et al. Experimental demonstration of interferometric imaging using photonic integrated circuits. *Opt. Express*, 25(11):12653–12665, May 2017.

[165] Xiaohao Cai, Marcelo Pereyra, and Jason D. McEwen. Uncertainty quantification for radio interferometric imaging: II. MAP estimation. MNRAS, 480:4170–4182, Nov 2018.

[166] Xiaohao Cai, Marcelo Pereyra, and Jason D. McEwen. Uncertainty quantification for radio interferometric imaging - I. Proximal MCMC methods. MNRAS, 480:4154–4169, Nov 2018.