

# **DESIGNING AN AUTOMATED IMAGING PIPELINE FOR THE uGMRT**

## **A THESIS**

*submitted in partial fulfillment of the requirements  
for the award of the degree of*

**Bachelor of Science - Master of Science**

*in*

**PHYSICS**

*by*

**ANANYA R  
(19035)**



**iiserb**

**DEPARTMENT OF PHYSICS  
INDIAN INSTITUTE OF SCIENCE EDUCATION  
AND RESEARCH  
BHOPAL - 462066**

**April 2024**



**भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान भोपाल**  
**Indian Institute of Science Education and Research**  
**Bhopal**  
**(Estb. By MHRD, Govt. of India)**

---

## CERTIFICATE

This is to certify that **ANANYA R**, BS-MS **PHYSICS**, has worked on the project entitled **DESIGNING AN AUTOMATED IMAGING PIPELINE FOR THE uGMRT** under my supervision and guidance. The content of this report is original and has not been submitted elsewhere for the award of any academic or professional degree.

April 2024  
IISER Bhopal

Dr. Mayuresh Surnis  
(IISER Bhopal)

Committee Member

Signature

Date

---

---

---

# **ACADEMIC INTEGRITY AND COPYRIGHT**

## **DISCLAIMER**

I hereby declare that this project is my own work and, to the best of my knowledge, it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at IISER Bhopal or any other educational institution, except where due acknowledgment is made in the document.

I certify that all copyrighted material incorporated into this document is in compliance with the Indian Copyright Act (1957) and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save IISER Bhopal from any and all claims that may be asserted or that may arise from any copyright violation.

**April 2024**  
**IISER Bhopal**

**Ananya R**

## **ACKNOWLEDGEMENT**

I express my deepest gratitude to my supervisor, Dr. Mayuresh Surnis, for his constant support and guidance throughout my Master's thesis project. His expertise, patience, and constructive feedback have been instrumental in shaping the direction and quality of this work. His mentorship was invaluable for the progress of my project for which I am profoundly indebted. In addition to my supervisor, I would like to thank all the individuals who have helped me in any way regarding my project. Their assistance, whether through providing resources, offering advice, or lending an ear, contributed significantly to the completion of this thesis.

I am deeply thankful to my parents and sister for their unwavering support, which helped me navigate through this journey's challenges and successes. Their encouragement has been a constant source of strength for me without which I wouldn't have been able to move forward.

Furthermore, I am grateful to all my friends from this college with whom I have developed invaluable bonds, including all 34 of my friends who could not be named here, for their friendship and support throughout these five years. I would like to heartily thank them for their unwavering presence and encouragement that have helped me thrive in all aspects of my life while filling it with immeasurable love and joy.

**Ananya R**

## ABSTRACT

This project aims to design an automated imaging pipeline for uGMRT (upgraded Giant Metrewave Radio Telescope) data, focusing on observations of the pulsar J1809-1943. Through the use of the imaging software program WSClean and the CAPTURE pipeline within the Common Astronomy Software Applications (CASA) architecture, we edited, calibrated, and imaged the radio visibility data of the pulsar J1809-1943 obtained using the uGMRT. The uGMRT pipeline, CAPTURE was modified for producing images of multiple sampling time intervals while using the imaging software WSClean. CAPTURE, modified to utilize WSClean for imaging, facilitated the production of self-calibrated radio images enhancing our understanding of the pulsar's behavior. The images produced so were analyzed to identify the sample time bin of alignment of the pulsar with the line of sight of the observer.

# **LIST OF SYMBOLS OR ABBREVIATIONS**

uGMRT	upgraded Giant Meterwave Radio Telescope
CASA	Common Astronomy Software Applications
CAPTURE	CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction

Additional abbreviations and symbols have been defined contextually wherever introduced.

# LIST OF FIGURES

1.1	GMRT Array configuration	4
1.2	Iterative image reconstruction	5
1.3	Schematic diagram showing the analysis steps performed by CAPTURE.	8
2.1	Simplified schematic diagram of a two-element interferometer	13
4.1	point spread function	23
4.2	primary beam	23
4.3	residual image	24
4.4	primary beam corrected image	24
5.1	Image of J1809-1943 after 4 rounds of phase-only self-calibration from original pipeline.	30
5.2	The cropped and color scaled image of the pulsar J1809-1943	31
5.3	Combined image obtained from the modified pipeline (niter=6400)	32
6.1	The image of the pulsar with different niter values. Top left niter=10000, top right niter=50000, bottom niter=100000	35
6.2	Contours of the above images with top left niter=10000, top right niter=50000, bottom niter=100000	36
6.3	Combined image obtained from the modified pipeline (niter= $10^6$ )	37
6.4	The figure shows the flux density map of the pulsar in each of the time bins. The visibility data of the pulsar was divided into 5 time bins and imaged to calculate the flux density.	38

# CONTENTS

Certificate	i
Academic Integrity and Copyright Disclaimer	ii
Acknowledgement	iii
Abstract	iv
LIST OF SYMBOLS OR ABBREVIATIONS	v
LIST OF FIGURES	vi
INTRODUCTION	1
<b>1 DATA COLLECTION AND PROCESSING TOOLS</b>	<b>3</b>
1.1 The Giant Metrewave Radio Telescope - GMRT . . . . .	3
1.2 Common Astronomy Software Applications - CASA . . . . .	4
1.2.1 <i>tclean</i> - CASA . . . . .	5
1.3 CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction - CAP-TURE . . . . .	6
1.4 WSClean . . . . .	7
1.5 Pulsars . . . . .	7
<b>2 RADIO INTERFEROMETRY THEORETICAL FORMULATION</b>	<b>9</b>
2.1 Simplifying Assumptions in Radio Interferometry . . . . .	9
2.2 Discrete Sampling and Antenna Reception . . . . .	12
2.3 Basics of Radio Interferometry . . . . .	13

2.4	The imaging coordinate system . . . . .	15
<b>3</b>	<b>CALIBRATION AND EDITING</b>	<b>17</b>
3.1	Calibration . . . . .	17
3.1.1	Initial calibration . . . . .	18
3.1.2	Bandpass calibration . . . . .	19
<b>4</b>	<b>IMAGING 3C391 USING CASA</b>	<b>20</b>
4.1	Process of editing and imaging . . . . .	20
4.2	Results of editing and imaging . . . . .	22
<b>5</b>	<b>EDITING THE CAPTURE PIPELINE</b>	<b>25</b>
5.1	Using the CAPTURE pipeline for imaging . . . . .	25
5.2	CAPTURE pipeline overview . . . . .	26
5.3	Modifying the CAPTURE pipeline . . . . .	28
<b>6</b>	<b>RESULTS AND DISCUSSION</b>	<b>33</b>
	<b>BIBLIOGRAPHY</b>	<b>39</b>

# INTRODUCTION

Since the discovery of cosmic radio emission by Karl Jansky in the 1930s, further confirmed by Grote Reber the field of radio astronomy has undergone significant advancements. These developments have enabled astronomers to observe and study astronomical phenomena not detectable through optical interferometry. The advancements in radio interferometry and imaging techniques have played a crucial role in the detection and study of various phenomena including pulsars, quasars, distant galaxies, and more. With continuous improvements in interferometry techniques and imaging algorithms, astronomers have been able to produce detailed images of radio sources and study their properties. Radio interferometers have an array of antennas that are spread out over a large area which simultaneously obtain signals from the source. The signals obtained from the component antennas are related to the radio image by a Fourier transform, which is later combined to produce detailed images of radio sources. To produce the image from the data obtained by a radio telescope, multiple stages of data processing are required, through which the data is edited to flag bad data, calibrated, and finally, Fourier transformed to obtain the final radio image.

The objective of this project was to design an automated imaging pipeline for uGMRT data, which can produce images at multiple sampling time intervals. For this, the data obtained by the upgraded Giant Metrewave Radio Telescope(uGMRT) of the pulsar J1809-1943(*RA* :  $18h9m51s$ , *DEC* :  $-19d43m52s$ , Date:18 Dec 2018) was utilized. Then CASA(Common Astronomy Software Applications), the radio imaging pipeline CAPTURE (CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction), and WSClean were used for processing and imaging this data. CAPTURE produces self-calibrated radio images after editing and calibrating raw radio interferometric data. It utilizes the *tclean* task from CASA for imaging. The CAPTURE

pipeline was modified to utilize the imaging software WSClean instead of *tclean* for imaging the calibrated interferometric data. WSClean which employs the w-stacking algorithm is a better option for imaging wide-field data. Also, the splitting of the radio visibility data into different time samples is easier in WSClean. The images of the pulsar while splitting the images into five time intervals were obtained using the modified pipeline. The flux density of each of the pulsar images was also studied.

The first section will go through a brief introduction of these tools utilized in our data analysis. It will go through the Giant Metrewave Radio Telescope (GMRT), the Common Astronomy Software Applications (CASA), the uGMRT pipeline CAPTURE, the imaging software tool WSClean, and a short description of pulsars. In the second section, the basics of radio interferometry required to understand the acquiring of radio signals, data processing, and combining the signals from multiple antennas to produce the radio image are explained. The formulation required for the Fourier transform and deconvolution required to convert the sampled visibility data into a radio image is explained here. Also, the response of a two-element interferometer is also reviewed in this section. In the third section, the process of editing and calibrating the radio visibility data is explained. Later the data collection and analysis will be discussed. We will go through the manual data processing and imaging of the supernova remnant 3C391 using CASA in the fourth section. In section five the editing of the CAPTURE pipeline to utilize WSClean during the imaging part along with the data reduction process is explained. Later we will go through the conclusions and findings of this project.

# Chapter 1

## DATA COLLECTION AND PROCESSING TOOLS

### 1.1 The Giant Metrewave Radio Telescope - GMRT

The GMRT (Giant Metrewave Radio Telescope) is a radio telescope array situated near Narayangaon, Pune, India. It is made up of 30 parabolic radio antenna dishes with a diameter of 45 meters, dispersed over a 25 km region. It has a central array of 12 antennas that form a random cluster in a region of size about 1 km, called the central square. The remaining antennas are located on the arms of a "Y" configuration. Figure 1.1 [1] shows the array configuration of GMRT.

The GMRT antennas are alt-azimuth-mounted parabolic prime-focus dishes. The longest and shortest baselines of the array are about 25 km and 100 m respectively. It was upgraded with new receivers after which it is called the uGMRT (upgraded Giant Metrewave Radio Telescope). The uGMRT replaced the 5 narrow band feed and front-end receiver systems of the legacy system with 4 wide-band receivers: 125 – 250 MHz (Band-2), 250 – 500 MHz (Band-3), 550 – 850 MHz (Band-4) and 1000 – 1460 MHz (Band-5). The GMRT correlator is an FX correlator. The correlator output, which is the raw visibilities is saved in LTA format(Long Term Accumulation format) which can later be processed to obtain radio images.

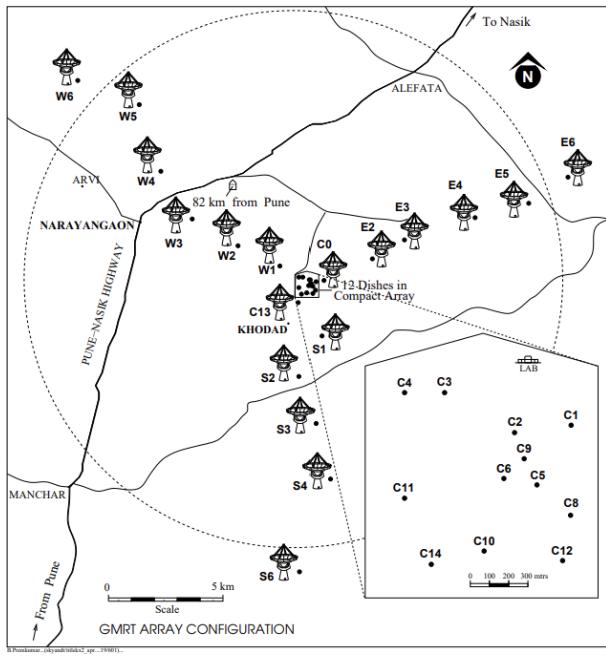


Figure 1.1: GMRT Array configuration

## 1.2 Common Astronomy Software Applications - CASA

Common Astronomy Software Application (CASA) is a suite of functions that can be used to reduce and analyze radio-astronomical data using a Python user interface. It is being developed by the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for VLBI European Research Infrastructure Consortium (JIV-ERIC), under the guidance of NRAO. It can be used for handling data from radio telescopes such as the AMLA, EVLA, and GMRT among others. The architectural components of CASA consist of the suite of libraries that are used for processing data, a command line interface that provides access to the libraries, and the user reference documentation. During this project the CASA 6.5.6.22 – Common Astronomy Software Applications [6.5.6.22] version with IPython 7.15.0 was used to process the data. CASA can be used to perform data editing by flagging bad data, data calibration imaging, and self-calibration. For processing and imaging the radio interferometric data obtained from the uGMRT we use the automatic pipeline,

CAPTURE (CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction) which utilizes CASA.[4],[5]

### 1.2.1 *tclean* - CASA

The CASA task *tclean* is the task used to produce the radio image from the edited and calibrated interferometric data. It is based on the CLEAN algorithm which is a technique used to reconstruct a model image from the interferometric data. It iteratively in each step removes and places a portion of the flux in the brightest pixel in a defined area of the dirty image in to the model image. The image reconstructions comprise of an outer loop of major cycles which implements transforms between the data and image domains and an inner loop of minor cycles which operates only in the image domain.

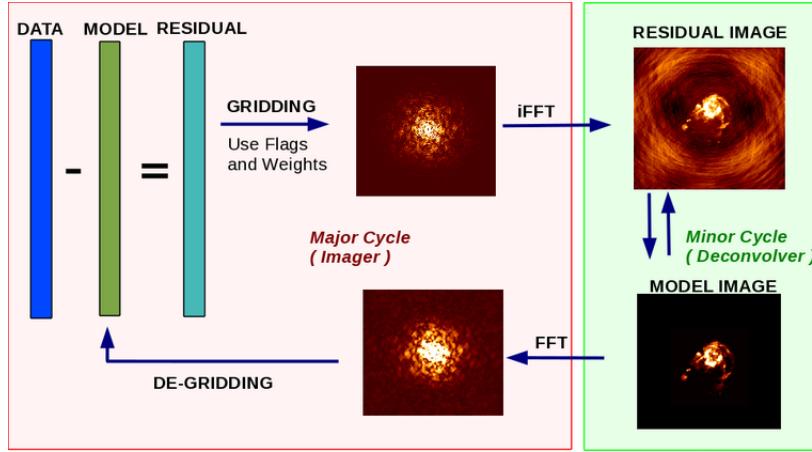


Figure 1.2: Iterative image reconstruction

The figure 1.2 [7] shows a diagram of the process of iterative image reconstruction using the minor and major cycles based on the Cotton-Schwab Clean algorithm but Hogbom and Clark Clean are also available as subsets of this framework. The data to image transformation is called imaging. The data is the calibrated visibilities of the object being imaged, data weights, and the u-v sampling function, the input is the Algorithm and the iteration controls and the output is the model image, residual image, and restored image. The weighted visibilities are convolutionally resampled onto a grid of spatial-frequency cells, inverse Fourier transformed and normalized. The

image to data transform is called the prediction step, which converts a model of the sky brightness into a list of model visibilities that are later subtracted from the data to form the residual visibilities. The directional dependent effects for both the transforms can be taken care of by the convolution function. Initially, a guess for the image model is taken as the iterations start. During the major cycle model, visibilities are predicted, residual visibilities are calculated, and the residual image is produced. It can be written as a convolution of the true image with a point spread function. During the minor cycle, a model of the true sky is iteratively built up by separating it from the point spread function which can be called deconvolution.[16]

### 1.3 CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction - CAPTURE

CAsa Pipeline-cum-Toolkit for Upgraded GMRT data REduction - CAPTURE is a continuum imaging pipeline developed for the uGMRT. It was developed by the National Centre for Radio Astrophysics (NCRA-TIFR), Pune, India. It is a fully automated pipeline that can convert the interferometric data obtained from the upgraded Giant Metrewave Radio Telescope into radio images. The Python program utilizes tasks from CASA (Common Astronomy Software Applications) to process and image the data. It performs the flagging of bad data, calibration, imaging, and self-calibration automatically to convert raw data to a self-calibrated continuum image. CAPTURE takes visibility data as input in the LTA, FITS, or Measurement Set(MS) format and gives calibration tables, calibrated target visibilities, and self-calibrated images as output. First, it locates and flags non-working antennas based on calibrator scans, where the source scans are flagged if the calibrator scans are bad. Then multiple rounds of calibration and flagging are performed. Then this calibrated data is averaged in frequency according to user preference and inspected for bad data. Then this calibrated visibility is used for imaging and self-calibration. The figure 1.3 [6] shows a schematic diagram of the analysis steps that are performed by CAPTURE during the data processing and imaging. The pipeline is available at <https://github.com/ruta-k/CAPTURE-CASA6> for public usage. The input variables can be set in “config\_capture.ini”, the configuration file which has two parts that are the “basic” block and the “default” block. The

Python functions that uses the CASA tasks to operate the CAPTURE pipeline are all defined in the “ugfunctions.py” file. The “vla-cals.list” contains the list of the names of calibrators and the “listscan” and “gvfits” contain precompiled binaries for lta to fits conversion. The “capture-casa6.py” file contains the pipeline code which utilises all the files mentioned before to perform the editing calibration and imaging.[6]

## 1.4 WSClean

WSClean stands for “w-stacking clean”. WSClean is a software tool used in radio astronomy to perform imaging and deconvolution of radio interferometric data. It is a wide-field interferometric imager that uses the w-stacking algorithm which can make use of the w-snapshot algorithm. The w-stacking algorithm is developed as an alternative to the w-projection algorithm. In the w-stacking algorithm, the uv-samples are corrected with a multiplication after the FFT (Fast Fourier Transform) instead of convolving with a w-term correcting kernel before the FFT. Multiplying each pixel is generally faster than convolving an image, but samples cannot be added together on one grid until the w-term correction is done. So each w-layer has to be stored and FFT-ed individually. Hence w-stacking requires more memory and spends relatively more time doing the FFTs, instead of gridding+convolving. It is specifically developed for wide-field radio interferometers. This method was found to be an order of magnitude faster than the w-projection of CASA on data from the MWA (Murchison Widefield Array), and also capable of full-sky imaging at full resolution with correct polarization correction.[8]

## 1.5 Pulsars

Pulsars or Pulsating Radio Stars are rapidly rotating highly magnetized neutron stars formed during the supernova explosions of massive stars. Charged particles are accelerated along the magnetic field which emits electromagnetic radiation as the neutron star spins which forms a beam. The electromagnetic radiation is observed as pulses, where the pulses are produced when the magnetic axis crosses the observer’s line of sight during each rotation. The repetition period of the pulse is simply the rotation period of the neutron star.

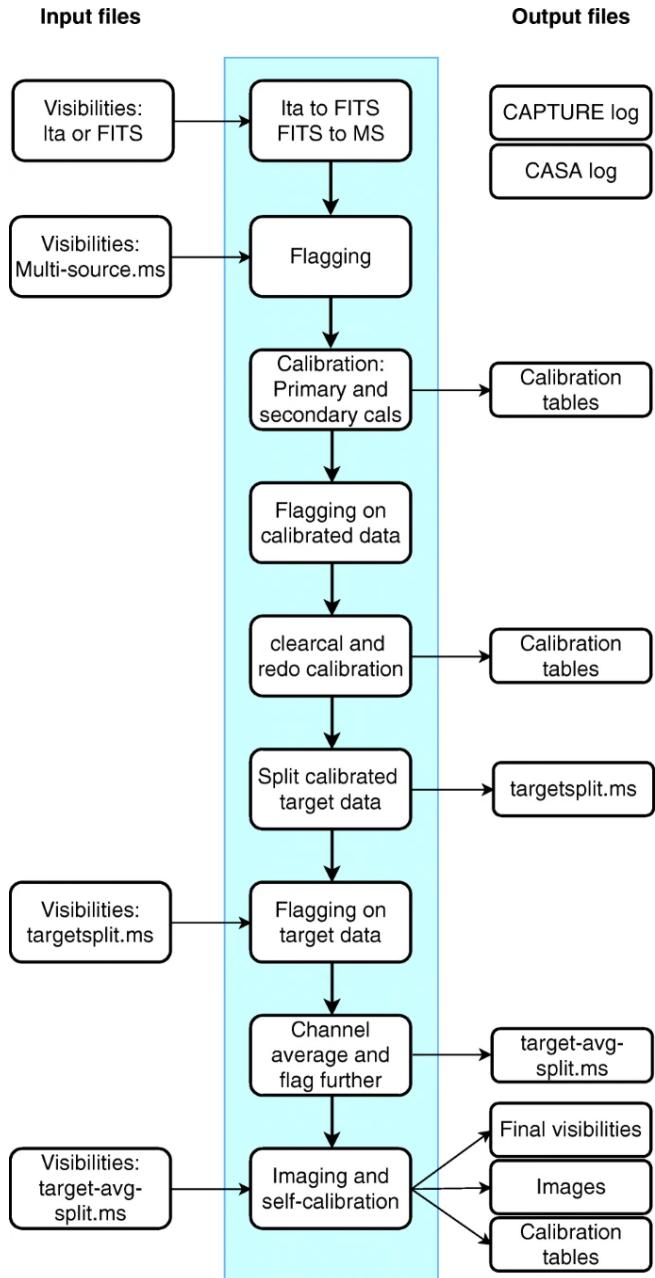


Figure 1.3: Schematic diagram showing the analysis steps performed by CAPTURE.

## Chapter 2

# RADIO INTERFEROMETRY

## THEORETICAL FORMULATION

### 2.1 Simplifying Assumptions in Radio Interferometry

Radio interferometry utilizes some simplifying assumptions to make the analysis manageable. We will go through the simplifying assumptions while formulating the main principles of synthesis imaging.

When an astrophysical phenomenon occurs at a position  $\mathbf{R}$  in space it causes a time-variable electromagnetic field,  $E(\mathbf{R}, t)$  at  $\mathbf{R}$  that will propagate through space and arrive at a point  $\mathbf{r}$  where the observer is at. The magnitude of the electric field for a finite time interval can be expressed as the real part of the sum of the Fourier series where the only time-variable functions are exponentials. The coefficients of the Fourier series,  $E_\nu(\mathbf{R})$  are quasi-monochromatic components of the electric field  $E(\mathbf{R}, t)$ .

The electric field at  $\mathbf{r}$ , the point from which the observation is being taken can be written as,

$$E_\nu(\mathbf{r}) = \iiint P_\nu(\mathbf{R}, \mathbf{r}) E_\nu(\mathbf{R}) dx dy dz, \quad (2.1)$$

where  $P_\nu(\mathbf{R}, \mathbf{r})$  is called the propagator and it describes how the electric field and  $\mathbf{R}$  influences the electric field at  $\mathbf{r}$ . The first simplifying assumption is such that we ignore the fact that electromagnetic radiation is a vector phenomenon and we will treat it as a scalar field. The second simplifying assumption is that our source of interest is so far away from the observer that what we measure is the surface brightness of the emitting

phenomenon. The third simplifying assumption is that the space within the celestial sphere (a very large sphere of radius  $|\mathbf{R}|$ ) is empty. The propagator here takes a simple form so that the equation (1) simplifies to

$$E_\nu(\mathbf{r}) = \int \mathcal{E}_\nu(\mathbf{R}) \frac{e^{2\pi i \nu |\mathbf{R} - \mathbf{r}|/c}}{|\mathbf{R} - \mathbf{r}|} dS, \quad (2.2)$$

where  $\mathcal{E}_\nu(\mathbf{R})$  is the electric field at the surface of the celestial sphere, of radius  $|\mathbf{R}|$  and  $dS$  is the surface area element of the celestial sphere. During observation of radio sources in the sky what the interferometer measures is the spatial coherence function of the electric field.

The correlation of the electric field at two observation points  $\mathbf{r}_1$  and  $\mathbf{r}_2$  is defined as,

$$V_\nu(\mathbf{r}_1, \mathbf{r}_2) = \langle E_\nu(\mathbf{r}_1) E_\nu^*(\mathbf{r}_2) \rangle. \quad (2.3)$$

Substituting the equation for the electric field in this we get,

$$V_\nu(\mathbf{r}_1, \mathbf{r}_2) = \left\langle \iint \mathcal{E}_\nu(\mathbf{R}_1) \mathcal{E}_\nu^*(\mathbf{R}_2) \frac{e^{2\pi i \nu |\mathbf{R}_1 - \mathbf{r}_1|/c}}{|\mathbf{R}_1 - \mathbf{r}_1|} \frac{e^{-2\pi i \nu |\mathbf{R}_2 - \mathbf{r}_2|/c}}{|\mathbf{R}_2 - \mathbf{r}_2|} dS_1 dS_2 \right\rangle. \quad (2.4)$$

The fourth simplifying assumption says that radiations from astronomical objects are not spatially coherent,

$$\langle \mathcal{E}_\nu(\mathbf{R}_1) \mathcal{E}_\nu^*(\mathbf{R}_2) \rangle = 0 \text{ for } \mathbf{R}_1 \neq \mathbf{R}_2. \quad (2.5)$$

Our equation becomes,

$$V_\nu(\mathbf{r}_1, \mathbf{r}_2) = \int \langle |\mathcal{E}_\nu(\mathbf{R})|^2 \rangle |\mathbf{R}|^2 \frac{e^{2\pi i \nu |\mathbf{R} - \mathbf{r}_1|/c}}{|\mathbf{R} - \mathbf{r}_1|} \frac{e^{-2\pi i \nu |\mathbf{R} - \mathbf{r}_2|/c}}{|\mathbf{R} - \mathbf{r}_2|} dS. \quad (2.6)$$

We will denote the unit vector  $\mathbf{R}/|\mathbf{R}|$  as  $\mathbf{s}$  and, the observed intensity,  $\langle |\mathcal{E}_\nu(\mathbf{s})|^2 \rangle |\mathbf{R}|^2$  as  $I_\nu(\mathbf{s})$ . Assuming that the source is very far away we can rewrite the equation neglecting the smaller terms,

$$V_\nu(\mathbf{r}_1, \mathbf{r}_2) \approx \int I_\nu(\mathbf{s}) e^{-2\pi i \nu \mathbf{s} \cdot (\mathbf{r}_1 - \mathbf{r}_2)/c} d\Omega. \quad (2.7)$$

This is the spatial autocorrelation function or the spatial coherence function of,  $E_\nu(\mathbf{R})$ . It only depends on the separation,  $\mathbf{r}_1 - \mathbf{r}_2$  of the observation points and not the absolute positions  $\mathbf{r}_1$  and  $\mathbf{r}_2$ . The interferometer measures this spatial coherence function.

The fifth simplifying assumption has 2 forms one being that starting points of  $\mathbf{r}_1$  and  $\mathbf{r}_2$  are on the same plane, that is the antenna elements lie in the same plane and

measuring the distances in wavelengths,  $\mathbf{r}_2 - \mathbf{r}_2 = \lambda(u, v, 0)$ . The components of  $\mathbf{s}$  in this coordinate system is  $(l, m, \sqrt{1 - l^2 - m^2})$ . The equation for spatial coherence function becomes,

$$V_\nu(u, v, w \equiv 0) = \iint I_\nu(l, m) \frac{e^{-2\pi i(ul+vm)}}{\sqrt{1 - l^2 - m^2}} dl dm. \quad (2.8)$$

This shows that  $V_\nu(u, v, w \equiv 0)$  and  $I_\nu(l, m)/\sqrt{(1 - l^2 - m^2)}$  are related to each other by Fourier transform.

Alternatively, we can assume that all the electromagnetic radiation comes from only a small part of the sky. That is, we can take  $\mathbf{s} = \mathbf{s}_0 + \boldsymbol{\sigma}$  and neglect all terms with the square of  $\boldsymbol{\sigma}$ . Both  $\mathbf{s}$  and  $\mathbf{s}_0$  being unit vectors we can write,

$$\begin{aligned} 1 &= |\mathbf{s}| = \mathbf{s} \cdot \mathbf{s} = \mathbf{s}_0 \cdot \mathbf{s}_0 \\ &= \mathbf{s}_0 \cdot \mathbf{s}_0 + 2\mathbf{s}_0 \cdot \boldsymbol{\sigma} + \boldsymbol{\sigma} \cdot \boldsymbol{\sigma} \\ &\approx 1 + 2\mathbf{s}_0 \cdot \boldsymbol{\sigma}. \end{aligned}$$

Hence,  $\mathbf{s}_0$  and  $\boldsymbol{\sigma}$  are perpendicular to each other. Now we take  $\mathbf{s}_0$  (called phase tracking center) as  $\mathbf{s}_0 = (0, 0, 1)$ . We get a different form for the spatial coherence function,

$$V'_\nu(u, v, w) = e^{-2\pi iw} \iint I_\nu(l, m) e^{-2\pi i(ul+vm)} dl dm. \quad (2.9)$$

We see that  $V_\nu(u, v, w) = e^{2\pi iw} V'_\nu(u, v, w)$  will be independent of  $w$ . Hence,

$$V_\nu(u, v) = \iint I_\nu(l, m) e^{-2\pi i(ul+vm)} dl dm. \quad (2.10)$$

Which is a Fourier transform whose inverse Fourier transform gives us the intensity,

$$I_\nu(l, m) = \iint V_\nu(u, v) e^{2\pi i(ul+vm)} du dv. \quad (2.11)$$

## 2.2 Discrete Sampling and Antenna Reception

During observation, the spatial coherence function,  $V$  is only measured at specific places in the  $u$ - $v$  plane. It is not known at all points in the sky. The sampling function,  $S(u, v)$ , describes the sampling, which would be zero at positions where no measurement is made.

$$I_\nu^D(l, m) = \iint V_\nu(u, v) S(u, v) e^{2\pi i(ul+mv)} du \ dv. \quad (2.12)$$

$I_\nu^D(l, m)$  is called the dirty beam. The dirty beam is related to the desired intensity distribution,  $I_\nu(l, m)$  by convolution,

$$I_\nu^D = I_\nu * B, \quad (2.13)$$

where  $B(l, m)$  is the point spread function (psf, also called synthesized beam),

$$B(l, m) = \iint S(u, v) e^{2\pi i(ul+mv)} du \ dv. \quad (2.14)$$

Interferometers have some sensitivity to the direction of arrival of the radio signals. Hence the equation of electric field, and hence the equations for spatial coherence function all have an additional term of  $\mathcal{A}_\nu(\mathbf{s})$  called the primary beam (pb, also called the normalized antenna reception pattern). The primary beam explains how the antenna's sensitivity changes with the direction of orientation. So the equation for visibility can be rewritten as,

$$V_\nu(u, v) = \iint \mathcal{A}_\nu(l, m) I_\nu(l, m) e^{-2\pi i(ul+vm)} dl \ dm. \quad (2.15)$$

## 2.3 Basics of Radio Interferometry

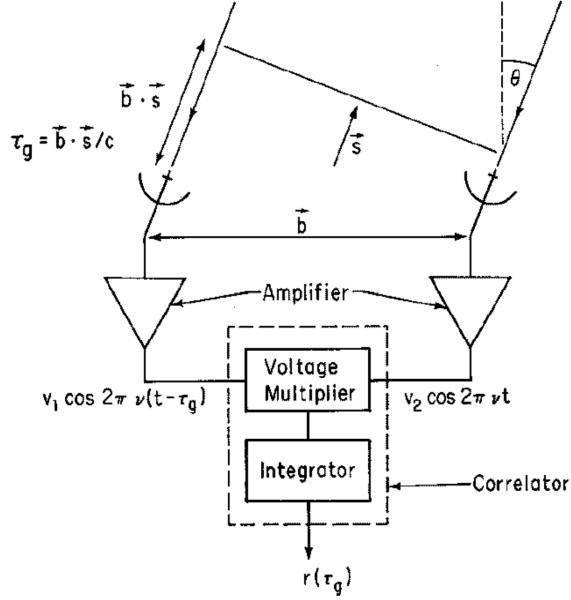


Figure 2.1: Simplified schematic diagram of a two-element interferometer

We can go through a two-element interferometer as in the figure 2.1 to better understand the interferometer response. Let two antennas point towards a source in a direction  $\mathbf{s}$  and the interferometer baseline is denoted by  $\mathbf{b}$  the wavefront front from the source reaches one antenna a time  $\tau_g$  later than the other, which is called the geometrical delay given by,

$$\tau_g = \mathbf{b} \cdot \mathbf{s} / c. \quad (2.16)$$

The signals are passed through amplifiers with filters that select the required frequency band of width  $\Delta\nu$  with central frequency  $\nu$ . It then passes through the correlator which combines the signals, which is a voltage multiplier followed by a time-averaging circuit. The output from the correlator is proportional to  $\langle V_1(t)V_2(t) \rangle$ , with the brackets denoting time averaging. We can denote the received signals as  $V_1(t) = v_1 \cos 2\pi\nu(t - \tau_g)$  and  $V_2(t) = v_2 \cos 2\pi\nu(t - \tau_g)$ , then the correlator output will be,

$$r(\tau_g) = v_1 v_2 \cos 2\pi\nu\tau_g. \quad (2.17)$$

The signal power received in bandwidth  $\Delta\nu$  from source element  $d\Omega$  is  $A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega$

where  $A(\mathbf{s})$  is the effective collecting area in direction  $\mathbf{s}$  and  $A(\mathbf{s})$  is the radio brightness in the direction  $\mathbf{s}$ . The correlator output from solid angle  $d\Omega$  is given by,

$$dr = A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega \cos 2\pi\nu\tau_g, \quad (2.18)$$

$$r = \Delta\nu \int_S A(\mathbf{s})I(\mathbf{s}) \cos \frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}}{c} d\Omega. \quad (2.19)$$

Representing the position vector by,  $\mathbf{s} = \mathbf{s}_0 + \boldsymbol{\sigma}$  where  $\mathbf{s}_0$  is the phase tracking center (or phase reference position), we obtain,

$$\begin{aligned} r &= \Delta\nu \cos \frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}_0}{c} \int_S A(\boldsymbol{\sigma})I(\boldsymbol{\sigma}) \cos \frac{2\pi\nu\mathbf{b} \cdot \boldsymbol{\sigma}}{c} d\Omega \\ &\quad - \Delta\nu \sin \frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}_0}{c} \int_S A(\boldsymbol{\sigma})I(\boldsymbol{\sigma}) \sin \frac{2\pi\nu\mathbf{b} \cdot \boldsymbol{\sigma}}{c} d\Omega. \end{aligned} \quad (2.20)$$

The complex visibility of the source can be regarded as the unnormalized measure of the coherence of the electric field, which is modified by the interferometer characteristic to an extent. It is given by,

$$V \equiv |V|e^{i\phi_V} = \int_S \mathcal{A}(\boldsymbol{\sigma})I(\boldsymbol{\sigma})e^{-2\pi i\nu\mathbf{b} \cdot \boldsymbol{\sigma}/c} d\Omega. \quad (2.21)$$

$\mathcal{A}(\boldsymbol{\sigma}) \equiv A(\boldsymbol{\sigma})/A_0$  is the normalized antenna reception pattern, where  $A_0$  is the antenna response at the beam center. When the antennas track the source, the system then responds to the modified brightness distribution given by,  $\mathcal{A}(\boldsymbol{\sigma})I(\boldsymbol{\sigma})$ . Separating the real and imaginary parts of  $V$  in equation (2.21) we get,

$$A_0|V|\cos\phi_V = \int_S A(\boldsymbol{\sigma})I(\boldsymbol{\sigma}) \sin \frac{2\pi\nu\mathbf{b} \cdot \boldsymbol{\sigma}}{c} d\Omega, \quad (2.22)$$

$$A_0|V|\sin\phi_V = - \int_S A(\boldsymbol{\sigma})I(\boldsymbol{\sigma}) \cos \frac{2\pi\nu\mathbf{b} \cdot \boldsymbol{\sigma}}{c} d\Omega. \quad (2.23)$$

Substituting these equations in the response equation (2.20) we get,

$$r = A_0\Delta\nu|V| \cos \left( \frac{2\pi\nu\mathbf{b} \cdot \mathbf{s}_0}{c} - \phi_V \right). \quad (2.24)$$

The usual procedure is to measure the amplitude and phase of the fringe pattern as represented by the cosine term in the above equation and then to derive the amplitude and phase of  $V$  by calibration. The brightness distribution of the source is obtained from the visibility data by the inversion of the transform in equation 2.21. Thus,  $V$  must be measured in a sufficiently wide range of  $\nu\mathbf{b} \cdot \boldsymbol{\sigma}/c$ , which is the component of the baseline normal to the direction of the source measured in wavelengths.

## 2.4 The imaging coordinate system

The baseline has the coordinates  $(u, v, w)$  where  $w$  points in the direction of the source,  $\mathbf{s}_0$  which would be the center of the synthesized image.  $u$ ,  $v$ , and  $w$  are all measured in wavelengths at the central frequency and in the direction East, North, and the phase tracking center. The positions on the sky are defined in the direction cosines  $l$  and  $m$  measured with respective to the  $u$  and  $v$  axes. The synthesized image in the  $(l, m)$  plane represents a projection of the celestial sphere onto a tangent plane at the  $(l, m)$  origin.

The parameters in the interferometer response can be written in these coordinates as,

$$\begin{aligned} \frac{2\pi\nu\mathbf{b}\cdot\mathbf{s}}{c} &= ul + vm + wn, \\ \frac{2\pi\nu\mathbf{b}\cdot\mathbf{s}_0}{c} &= w, \\ d\Omega = \frac{dl \ dm}{n} &= \frac{dl \ dm}{\sqrt{1 - l^2 - m^2}}. \end{aligned} \quad (2.25)$$

The equation for the complex visibility (2.21) can be rewritten using these as,

$$V(u, v, w) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{A}(l, m) I(l, m) e^{-2\pi i [ul + vm + w(\sqrt{1 - l^2 - m^2} - 1)]} \frac{dl \ dm}{\sqrt{1 - l^2 - m^2}}. \quad (2.26)$$

To simplify the inversion this equation can be reduced to a 2 dimensional Fourier transform which can be done under 2 sets of conditions. First taking the baselines to be coplanar where the rotation of the earth traces a circle concentric with the earth's axis. With an East-West array, all the baselines would be coplanar and there would be no component parallel to the axis of Earth. Now if we choose the w-axis to lie in the direction of the celestial pole,  $w = 0$ , The above equation can be rewritten as,

$$V(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{A}(l, m) I(l, m) e^{-2\pi i [ul + vm]} \frac{dl \ dm}{\sqrt{1 - l^2 - m^2}}. \quad (2.27)$$

Whose inversion will give us the equation,

$$\frac{\mathcal{A}(l, m) I(l, m)}{\sqrt{1 - l^2 - m^2}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V(u, v) e^{2\pi i [ul + vm]} du \ dv. \quad (2.28)$$

For an East-West array, the projected spacing of the baselines in the v direction becomes shortened for low declination observations which makes having baselines with

a significant component parallel to the Earth's axis. Thus the baseline vectors don't remain coplanar for a 2-dimensional array in  $(u, v, w)$  space. A system of 3 coordinates is necessary here. So the second set of conditions where the equation reduces to a 2 dimensional Fourier transform depends upon  $|l|$  and  $|m|$  being small enough so that we can write,

$$(\sqrt{1 - l^2 - m^2} - 1)w \approx -\frac{1}{2}(l^2 + m^2)w \approx 0. \quad (2.29)$$

Then equation 2.26 becomes,

$$V(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{A}(l, m) I(l, m) e^{-2\pi i [ul + vm]} dl dm. \quad (2.30)$$

From this we can write,

$$\mathcal{A}(l, m) I(l, m) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V(u, v) e^{2\pi i [ul + vm]} du dv. \quad (2.31)$$

## Chapter 3

# CALIBRATION AND EDITING

The data from antenna pairs recorded are the observed visibility data,  $\tilde{V}_{ij}$ , and is different from the true visibility,  $V_{ij}$ . Calibration is done in order to determine the true visibility from the visibility that is observed. Calibration of the data is identifying and measuring the different parameters that induce errors in the observed data and using these parameters to estimate the true visibility. Editing is the process of locating and removing faulty and inconsistent data and discarding the data is referred to as data flagging. Improper observing parameters, interference from terrestrial sources, inaccurate antenna tracking, weather-related issues, and malfunctioning receivers are all some problems that may require data flagging. Certain antenna pairs may have periods of time where their gain value cannot be established; in these cases, data flagging has to be done to get rid of these data.

### 3.1 Calibration

The basic calibration equation is given by,

$$\tilde{V}_{ij}(t) = \mathcal{G}_{ij}(t)V_{ij}(t) + \epsilon_{ij}(t) + \eta_{ij}(t), \quad (3.1)$$

where,  $t$  is the time of observation,  $\mathcal{G}_{ij}(t)$  is the baseline-based complex gain,  $\epsilon_{ij}(t)$  is a baseline-based complex offset, and  $\eta_{ij}(t)$  is a stochastic complex noise.

The baseline-based complex gain,  $\mathcal{G}_{ij}(t)$  can be written approximately as the product of antenna-based complex gains,  $g_i(t)$  and  $g_j(t)$ ,

$$\mathcal{G}_{ij}(t) = g_i(t)g_j^*(t) = a_i(t)a_j(t)e^{i(\phi_i(t)-\phi_j(t))}, \quad (3.2)$$

where  $a_i(t)$  is the antenna-based amplitude correction and  $\phi_i(t)$  is the antenna-based phase correction. By observations of the calibrator source, we can determine the complex gain,  $\mathcal{G}_{ij}(t)$  for each of the  $N(N-1)/2$  baseline which can then be used to calculate the  $N$  values of  $g_i$ .

The visibilities are averaged over time intervals during which it and the complex gain are expected not to change considerably. Calibration methods can be divided into 3 main categories (1) Direct calibrations: Interferometer arrays require that they meet tolerance specification to ensure stable and linear operation. Special feedback circuitry and monitoring critical parameters within the array system and array environment can be used to correct for changes during observation. (2) Calibrator sources in the sky: Interferometers measure phase difference instead of absolute phase. During observation, visibility phases are measured in reference to the phase tracking center. Antenna phase offset measurements necessitate the monitoring of a sky calibrator. In case the array is not stable for phase or gain periodic observations of the calibrator are necessary. (3) Self-calibration: During self-calibration, we use the signal from the source as a test signal to calibrate the array.

### 3.1.1 Initial calibration

Initial calibration before data is taken includes calibration of antenna position, delays, gain, and the positioning of the calibrator source used for system gain monitoring. The variation between the desired and actual pointing positions of the antenna is the antenna pointing error and is caused by a number of factors. The error has directional dependence due to error in alignment of the polar or elevation axis, gravitational misshaping of the structure, non-perpendicularity of the axes, atmospheric refraction and other effects, and time dependence due to differential heating of the structure. The error can be determined by tiltmeters placed in the antenna structure.

The integrated visibility function for a finite bandwidth is given by,

$$V_{ij}(t) = \int_0^\infty \left( \int_{-\infty}^\infty \int_{-\infty}^\infty \mathcal{A}_\nu(l, m) I_\nu(l, m) e^{-2\pi i \nu \Delta \tau_g} dl dm \right) e^{2\pi i \nu \Delta \tau_r} \mathcal{G}_{ij}(\nu) d\nu, \quad (3.3)$$

where  $\Delta \tau_g$  is the delay relative to the delay tracking center and  $\Delta \tau_r$  is the error in inserted delay for the delay tracking center. The difference in phase  $\Delta \phi$  between the

band's ends, that results from the residual delay is given by  $\Delta\phi = 2\pi\Delta\nu(\Delta\tau_g - \Delta\tau_r)$ . The delay errors occur because only the delay for the delay tracking center can be offset by inserting the delay in the system, hence the emission from other directions will have some loss of coherence, and because of the error in the inserted delay due to miscalculation, miscalibration or quantization of the delay inserted. During delay calibration, these residual delays are measured. There are 2 methods for delay calibration one is to observe a strong source and vary the delay until maximum coherence is found and the other is to make the phase slope across the pass band zero by varying the delay.

### 3.1.2 Bandpass calibration

In continuum observations, the visibility is integrated over a wide bandwidth to obtain the best snr. If the delays are set correctly, there are no significant changes in visibility over the bandwidth, and the chromatic aberration is negligible then it is fine to sum the emission over the bandwidth before correlation. But these conditions may not be met and the complex gain may vary with frequency. The delay calibration removes the phase gradient and the propagation difference in signal paths. The baseline-based complex gain changes with frequency. Bandpass calibration is the compensation for the change in gain with frequency. To determine the relative frequency response of respective frequency channels a strong calibrator source is monitored for an adequate amount of time. The observed visibility,  $\tilde{V}_{ij}(\nu_k)$  divided by the correct visibility,  $V_{ij}(\nu_k)$  gives the bandpass complex gain function, where the correct visibility is found by observing a strong source.

## Chapter 4

# IMAGING 3C391 USING CASA

### 4.1 Process of editing and imaging

To be familiar with CASA (Common Astronomy Software Applications) and to learn the process of imaging, a tutorial on calibration and imaging of the interferometric data of 3C391 supernova remnant obtained using VLA (Karl G. Jansky Very Large Array) was used. There are 10 sources observed where,  $J1331 + 3030 = 3C286$ , will be the calibrator for visibility amplitudes and bandpass,  $J1822 - 0938$ , will be the visibility phase calibrator,  $J0319 + 4130 = 3C84$  will be the polarization calibrator and the 7 fields located within the supernova remnant are 3C391 C1–C7.

We start with reviewing the data and editing it. We use the observer's log to find faulty antennas during the observation period so that we can flag their scan data. We use *listobs* to obtain an overview of the data set. We see that the first scan is a setup scan and flag that scan, the antenna that doesn't have the C-band receiver, and the corrupted data using the *flagdata* task. Arrays often require some brief period of time to stabilize at the beginning of the scan, so we also flag the first few samples at the beginning of scans which is called quack flagging. We also use *plotms* to examine and edit the data. The observed visibility  $V'$  between two antennas i and j are related to the true visibility  $V$  by,

$$V'_{i,j}(u, v, f) = b_{ij}(t) [B_i(f, t)B_j^*(f, t)] g_i(t)g_j(t)V_{i,j}(u, v, f)e^{i[\theta_i(t)-\theta_j(t)]}, \quad (4.1)$$

where  $g_i$  and  $\theta_i$  are the amplitude and phase of complex gain,  $B_i$  is the complex band-pass and  $b(t)$  is the often neglected baseline term.

The *gencal* task is used to specify various types of calibration values. We use it to insert calculations of the antenna position corrections. The initial flux scaling is done by monitoring a primary calibrator whose flux density is already known and by comparison, the antenna gain values,  $g_i$  are determined. We use *setjy* to get a list of available calibrator models. From that, the C-band model will be used on 3C286. Now using the calibrator model 3C286\_C.im the flux density can be scaled using *setjy* for  $J1331 + 3030$ . An initial phase calibration is done before the bandpass calibration by using *gaincal* to calculate the phase versus time for the central channels for the 3 calibrators while applying the antenna position correction. Then *plotms* is used to plot the phase and we see that there are phase jumps for the ea05 antenna hence we flag the antenna.

Now *gaincal* is used on the bandpass calibrator alone and a separate phase calibration table is made. In the first phase of bandpass calibration, we solve for the delays with respect to the antennas. To solve for the relative delays of the antennas with respect to the antenna selected as the reference we use the gain type K in the task *gaincal* while applying the antenna position correction and the gain calibration for the bandpass calibrator. The visibility function should be constant with frequency but due to the antenna bandpasses, it changes with frequency. The next stage is to solve for  $B_i$  (complex bandpass). The *bandpass* task solves for the complex bandpass by using the phase solutions derived.

Next, we determine corrections for  $g_i$  and  $\theta_i$  that is the complex gains. Firstly we determine the complex gains for 3C286 (flux density calibrator) using *gaincal*. In the next step, we calculate the complex gain in the direction of the source from the phase calibrator J1822-0938 using *gaincal*. The *calmode* '*ap*' is used to get both amplitude and phase values of the gain. We can also check the phase stability of the reference antenna during this time using *plotms*. To determine the response of the system to a source of known flux density and hence the gain of the system we use the primary flux calibrator. We take it such that the mean gain amplitude is equal for both the primary calibrator and the secondary calibrator. Initially, the correct flux density of the secondary calibrator is unknown and now we scale using the gain of the primary calibrator to find it. for the secondary calibrator, we utilize the task *fluxscale* to create a table with appropriately scaled amplitude gains.

Now we apply all the calibration corrections to the actual data using *applycal*. First, we apply the antenna position adjustments, scaled gain calibration for amplitude and phase, delay calibration, and bandpass calibration to each, primary and secondary calibrator. Then we apply it to all 7 source fields of the target. We use the gain calibration of the secondary calibrator for the process as it is closer to the source fields. Then we split the corrected data of only the target source using the *split* task and utilize *statwt* to adjust the data weights which will eliminate the impact of the relative noise scatter added during data flagging.

The *tclean* task in CASA after Fourier transforming the data gives the dirty image and also deconvolves the image obtained after Fourier transform. It forms the model image from the dirty image by iteratively removing at each step a fraction of flux in the brightest pixel from the defined region. *tclean* is based on the CLEAN algorithm which is used for reconstructing the image from the interferometric data. Högbom's CLEAN algorithm operates under the assumption that the radio sky can be accurately depicted by point sources and uses an iterative procedure to find their strength and position. The CLEAN algorithm has 2 main variants, the *Högbom algorithm* and the *Clark algorithm*. To correct the image with the primary beam, we divide the image obtained after deconvolution with the primary beam using the task *impbcor*.

## 4.2 Results of editing and imaging

The task *tclean* creates output files of .image, .pb, .mask, .model, .psf, .residual, .weight, and .sumwt. The figures from 4.1 to 4.4 are the output after all the editing, calibration, and imaging. The figure 4.1 is the image of the point spread function (psf, also called synthesized beam). It is the dirty beam that was deconvolved from the true visibility while running the clean process using *tclean*. It is the telescope's spatial response to a point source in the sky. The figure 4.2 is the image of the primary beam which is the effective response of the telescope to radiation from a source in the sky. The figure 4.3 is the residual image that is left at the end of the deconvolution process. It is useful to diagnose whether or not to clean the image more deeply. The figure 4.4 is the image of the supernova obtained after running the task *impbcor*. It is the primary beam-corrected image of the supernova.

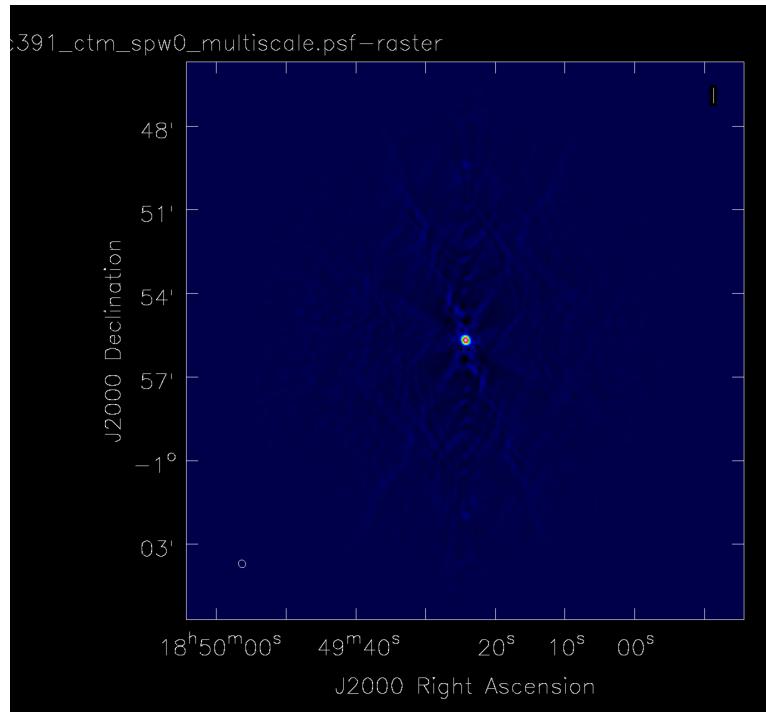


Figure 4.1: point spread function

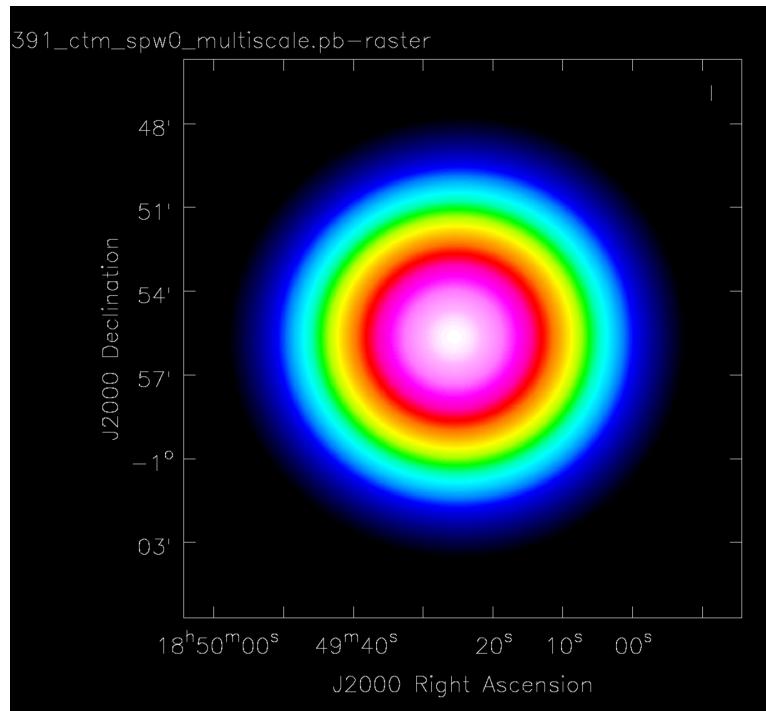


Figure 4.2: primary beam

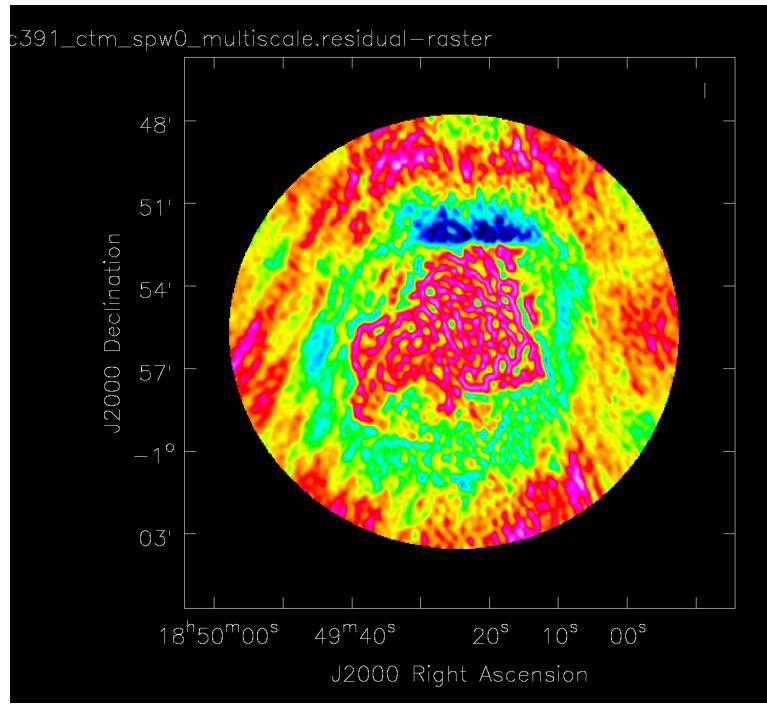


Figure 4.3: residual image

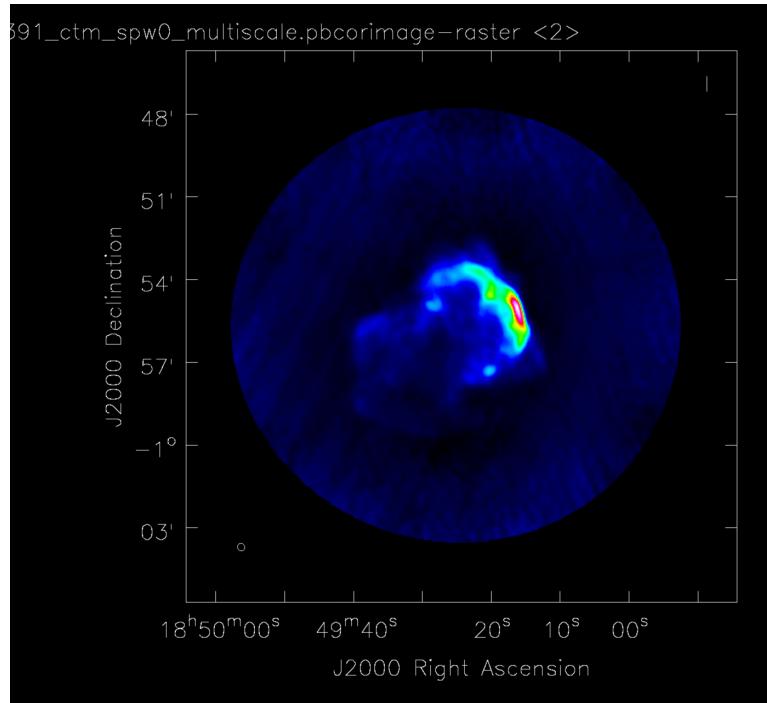


Figure 4.4: primary beam corrected image

# Chapter 5

## EDITING THE CAPTURE PIPELINE

### 5.1 Using the CAPTURE pipeline for imaging

CAPTURE is an automatic continuum imaging pipeline tailored for processing the uGMRT data efficiently. To utilize the CAPTURE pipeline, CASA is required to be installed since it relies on the CASA tasks for data processing and imaging. Also, the pipeline files including the uGMRT visibility file need to be in the same location while using the pipeline. The main pipeline in “capture-casa6.py” uses the functions in “ugfunctions.py” to obtain the radio image from the processed radio visibility file while taking input for variables required for the processing and imaging from the configuration file, “config\_capture.ini”. The first part of the file is of boolean values that determine the processes that would be performed and the second part has the values required for executing these processes. The output files obtained after executing the pipeline are the calibration tables, measurement sets, and image files generated in each iteration of self-calibration. Only the visibility file from the last iteration is retained while the visibility files produced before are removed. All the calibration tables are also retained.

The CAPTURE pipeline was used in a particular application to image the pulsar J1809-1943 (proposal ID: ddtC042, Observation No.: 10740). The pulsar was observed in Band-4 (550 MHz) at the coordinates RA:18h9m51s DEC:-19d43m52s. 3C286 was

used as the flux density calibrator and 1822-096 was used as the phase calibrator where J1809-1943 is the target source. The variables required for the pipeline were set in the configuration file. The reference antenna was set as E04. We set the configuration file so that it performs 4 rounds of self-calibration on the split average visibility file. It was specified in the configuration file that we would only be performing phase only self-calibration and not amplitude and phase self-calibration. The cell size was set to 1 arcsec and the image size in pixels was set to 5000. The fits file of the pulsar observation was also provided. The pipeline converts it to an ms file to perform the data processing and calibration. After flagging and calibration, it produces an averaged split MS file of the visibility. Also, the plots produced during the editing and calibration are all saved in a file named diagnostic plots. The split average visibility file is then used to produce a dirty image and then self-calibration rounds are performed and images are developed using *tclean* during each iteration of the self-calibration loop.

The pipeline produces an image at each self-calibration round. The figure 5.1 is the image obtained after using the CAPTURE pipeline on the visibility data of the J1809-1943 pulsar data after 4 iterations of self-calibration loop, which included phase-only calibration and not phase and amplitude self-calibration. During each self-calibration iteration, the pipeline produces an image using *tclean* while doubling the niter. In the figure 5.1 the white circle marks the pulsar, J1809-1943 which is the point source inside it. The image was visualized while using the plotting package Aplpy. The figure 5.2 is a color scaled and cropped version of the image 5.1 with the pulsar at the center of the image. The casa task imfit was used to calculate the flux density of the point source which is mentioned in the table given below.

## 5.2 CAPTURE pipeline overview

First, the pipeline reads the configuration file “config\_capture.ini” and stores the values provided in it in appropriate variables. The pipeline starts the data processing by converting the input visibility file into a measurement set (MS) file depending on whether it is an LTA or FITS file. Then the MS file is checked to find the channel range for calibration, the assumed clean channel range, the channel range for flagging, and the polarizations in the file. Then the amplitude calibrator, the phase calibrator, and

the target sources are identified. The amplitude, phase calibrator scans, and target scans are also then identified. Next, the list of the antennas in the file is collected.

Now the data flagging and editing of the visibility file starts. It starts with finding and flagging the bad antennas and bad channels. It then starts flagging the first channel and performing quack flagging as part of the initial flagging. It then identifies and flags the data by clipping at high amplitude levels, that is flagging data exceeding a predetermined amplitude threshold. It then further uses tfcrop for tighter flagging of the flux and phase calibrator. Then the flagging is extended to neighboring data points also. Next, it flags the target in the same way and then begins calibration. In the end, a flagging summary is generated.

The calibration starts by performing initial gain calibration using the *setjy* task on the amplitude calibrator by filling the model column with the visibility of the amplitude calibrator. Then it performs delay calibration using *gaincal* where it corrects for the antenna-based delays. It then goes on to perform the initial bandpass calibration where the frequency-dependent gains are corrected using the amplitude calibrator and then a final bandpass calibration is performed to refine the amplitude and phase solutions. Then additional gain calibration is performed on the calibrators and then a flux scale calibration is performed on the data. The calibration solutions are then applied to the phase calibrator and then to the target sources. Then after this calibration, diagnostic plots are generated to visualize the results of the calibration. After the first round of flagging and calibration, the pipeline performs a post-calibration flagging of the measurement set file. It performs clip flagging to remove outliers in calibrated data, TFCrop flagging and RFlag flagging are performed for calibrator tight flagging and the flagging is extended. Then the target is flagged based on the antennas and baselines. Now the calibration is redone. The previous calibration solutions are clear at the beginning. Using *setjy* the flux density of the amplitude calibrators is then estimated. Then the delay calibration is performed using the first amplitude calibrator and then the bandpass calibration is performed where the bandpass solutions are computed based on the calibrators and the previous calibration tables. Then the gain calibration solutions of all the calibrators and the flux scale for the phase calibrators are computed and then all the calibration solutions are applied to the MS data for the amplitude calibrators, phase calibrators, and targets. Once again after the flagging and calibration

diagnostic plots are generated. Now depending on the input from the configuration file, the pipeline performs splitting of the measurement set file. It compares the fields with standard amplitude calibrators and the list of phase calibrators from the “vla-cal.list” file and the fields that are in neither of them are set as the target source and split into a separate file. Now it performs flagging of this split file where a time-frequency flagging (*tfcrop*) is done. Then a list of baselines is obtained to perform rflag flagging. After this, the pipeline performs frequency averaging of the split MS file and then it performs rflag flagging of the split average file. Now this split average file can be used to make a dirty image using *tclean* and then perform self-calibration to later generate clean images using *tclean*.

### 5.3 Modifying the CAPTURE pipeline

While the CAPTURE pipeline used *tclean* for the imaging, my goal was to modify the pipeline so that the software tool WSClean could be used instead of *tclean* and to use it to generate multiple images of different time intervals. The most recent version of WSClean and its manual was available at <https://gitlab.com/aroffringa/wsclean/> and <https://wsclean.readthedocs.io/> respectively for public usage. WSClean uses the w-stacking algorithm for wide-field imaging while *tclean* uses the w-projection algorithm. It was installed in order to use it for imaging by combining it with the pipeline. Initially, it was used on the self-calibrated measurement set file to be familiarized with the imaging.

The data flagging and editing part of the CAPTURE pipeline need not be changed. Only the imaging part after generating the split average visibility file needs to be modified. For incorporating WSClean with the Python code of the pipeline it needed to be called through an executable file because it could not be called directly through the code. The executable file obtains inputs for the variables required for running WSClean from the Python code to perform the imaging. It takes input for the name of the input ms file, the name of the output files, output image size in number of pixels, the scale of a pixel, niter (maximum number of clean iterations to perform), threshold, and the number of intervals to image inside the selected global interval. The CAPTURE files “gvfits”, “listscan”, “vla-cals.list”, and “config\_capture.ini” are

retained as the same. The alterations are made in the main pipeline “capture-casa6.py” and “ugfunctions.py”, which contain the functions.

In the file “ugfunctions.py” a new function was added which can call WSClean using the values provided while running the pipeline. The function utilizes the Python library subprocess to run the executable file “clean.sh” that was designed to run the WSClean software. The function takes the values required to run WSClean and passes it on to the executable file which in turn performs the imaging and gives us the images as output. The executable file I have written first checks the command line takes input from the command line arguments and then runs WSClean using those parameters. The dirty image, the cleaned image, the model image, the psf, and the residual image obtained as the output after operating WSClean through the executable file are all saved as fits files. This function added to the “ugfunctions.py” file can be used to call WSClean in the main pipeline by providing appropriate arguments for the function. The self-calibration also needed to be rewritten to utilize WSClean during its imaging. The WSClean is called in the self-calibration with the *-intervals-out* set as 5 since we need to split the measurement set in 5 intervals and image each interval separately. Also, a separate call for WSClean is done with *-intervals-out* set as 1 to make an image with all the measurement intervals combined. Since the dirty image would be produced in each run of WSClean the command to produce the dirty image was removed from the pipeline. Necessary changes are made to the main pipeline to include the new functions for self-calibration and imaging. Now as the output, we get 5 images of the self-calibrated visibility of split time intervals and a combined image.

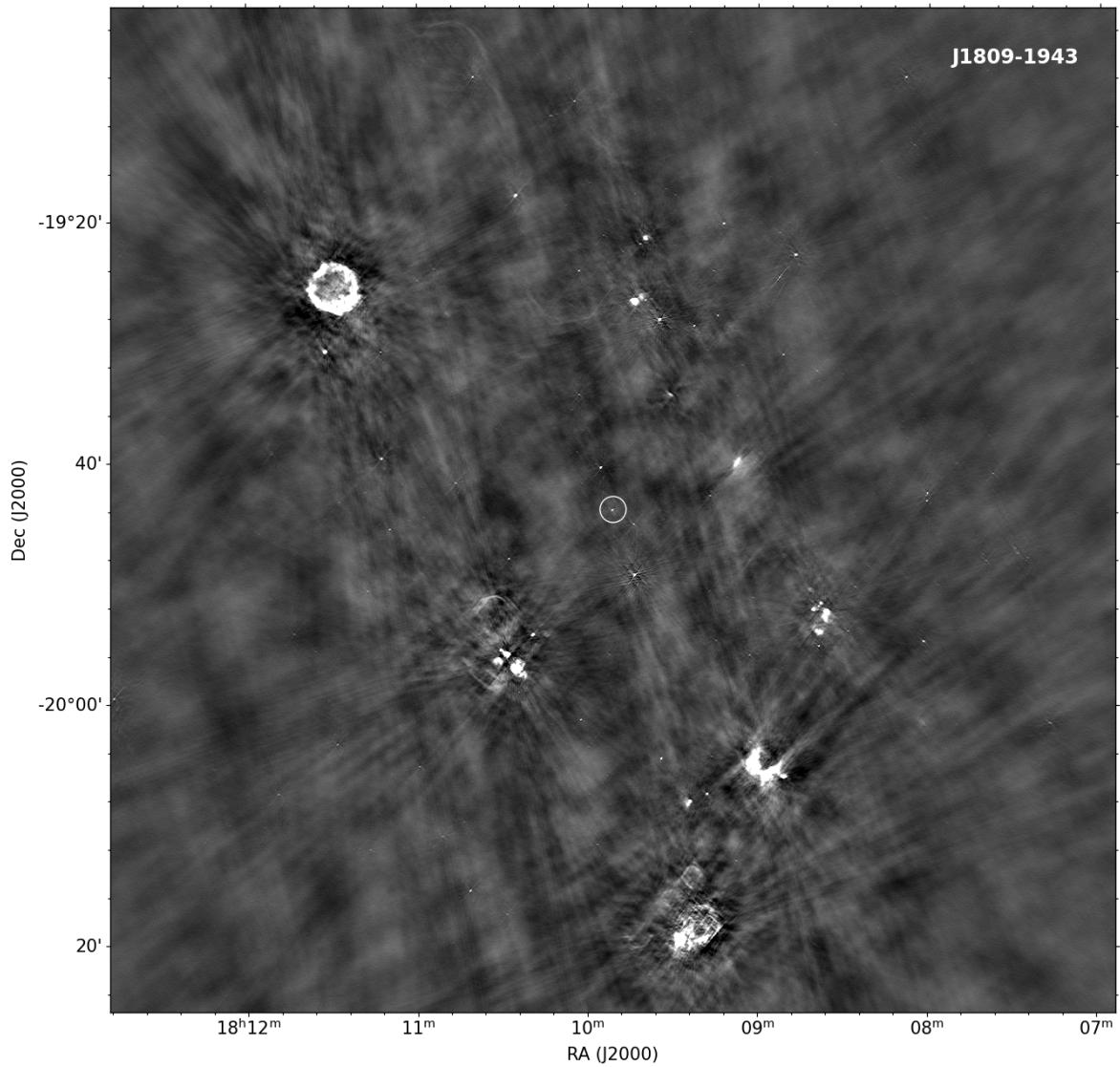


Figure 5.1: Image of J1809-1943 after 4 rounds of phase-only self-calibration from original pipeline.

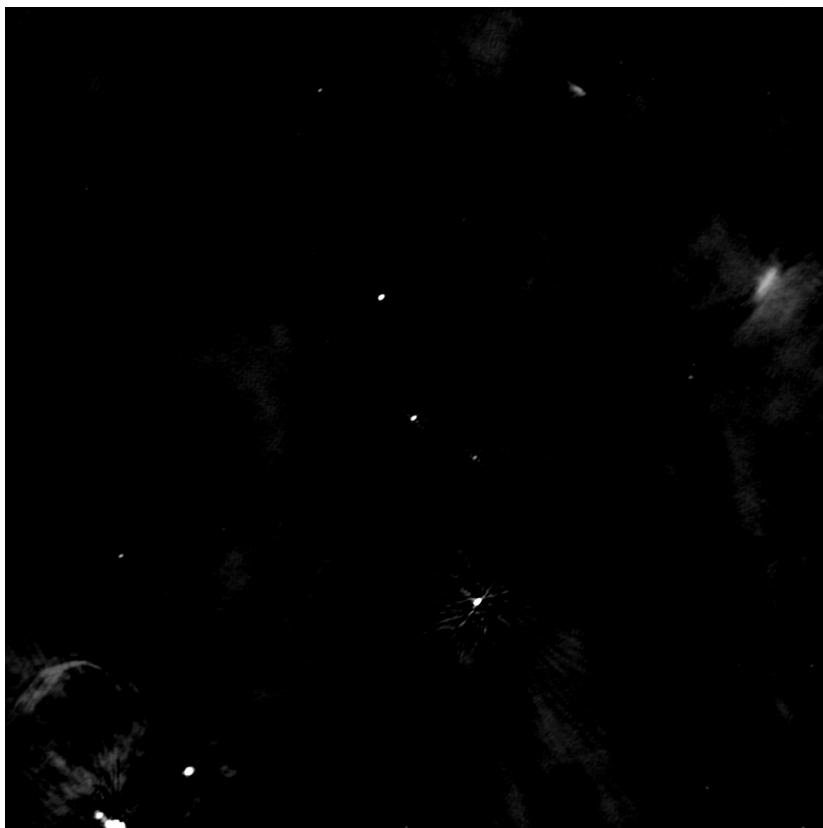


Figure 5.2: The cropped and color scaled image of the pulsar J1809-1943

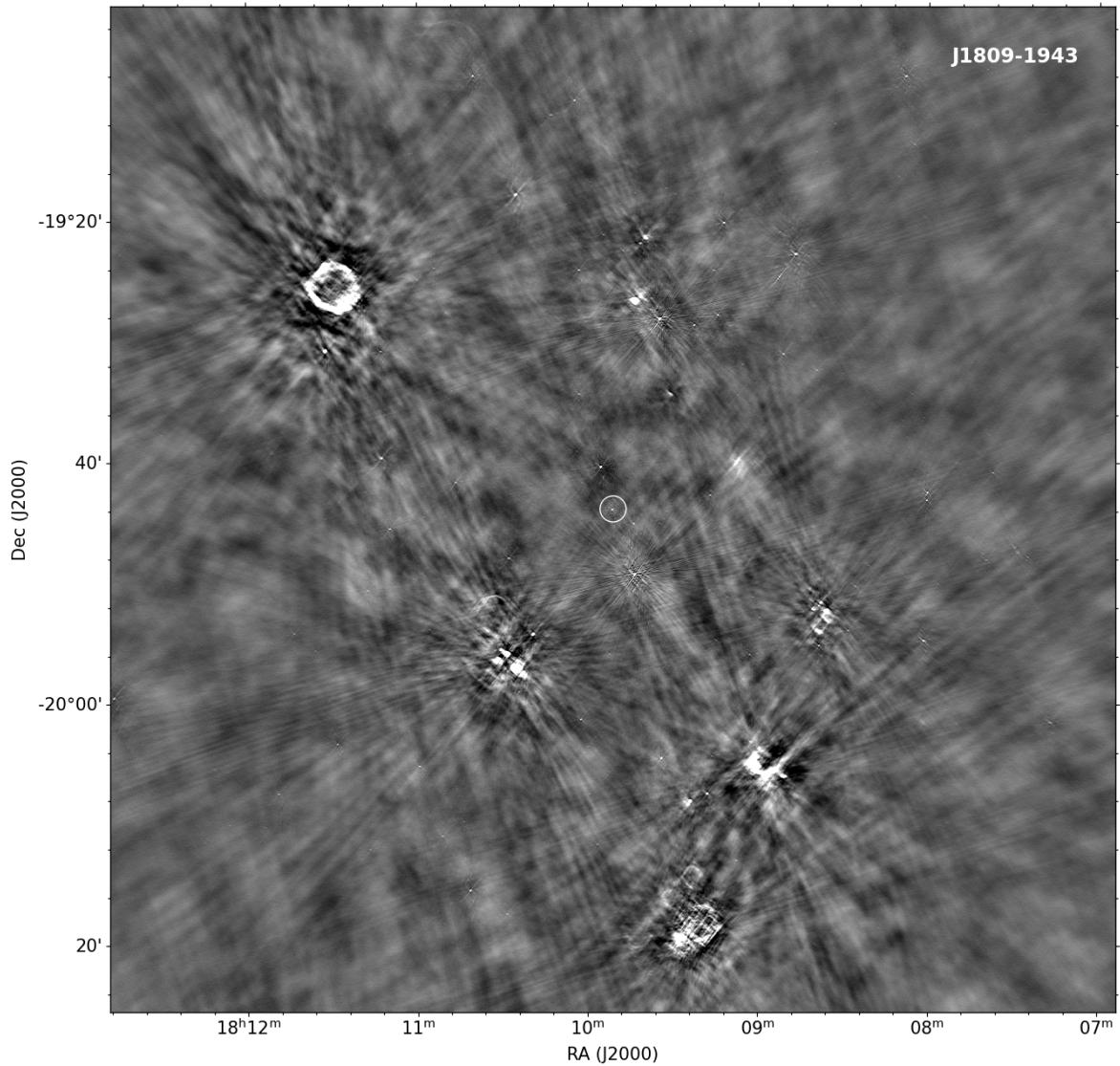


Figure 5.3: Combined image obtained from the modified pipeline (niter=6400)

## Chapter 6

# RESULTS AND DISCUSSION

The aim was to develop a pipeline so that it could visualize the electromagnetic pulse produced by the pulsar. At <https://github.com/ananya176/pipelinetwsclean.git> all of the files required for running the pipeline and the edited pipeline are made available. No change is done to “vla-cals.list”, “listscan”, and “gvfits” since they are standard files required for the data reduction. These files along with the visibility file are needed to use the pipeline for imaging. This pipeline can successfully perform the data reduction and then generate self-calibrated images while splitting the samples into five time intervals. It can also make an image that combines the whole of the sampling time interval. The pipeline was implemented on the visibility data of the pulsar, J1809-1943. Also, a small code was written that uses the casa task *imfits* to calculate the flux density of the pulsar from the image. We first ran the pipeline using the default values provided in their GitHub repository with very few changes. So we used niter start as 400 with it doubling in each run of the self-calibration loop and in the end the last image was generated with niter 6400, figure 5.3.

According to theory while binning the sample scans the flux density of the image in the time interval where the pulsar aligns with the observer should be higher than when it is not aligned. However, for our images, we could not find consistent values. To understand the reason behind this we used the *imfit* task to calculate the flux density of some other point in the images to find if they are consistent. But it also was found to be inconsistent. Running the whole pipeline also takes about 7 hours so to perform only imaging we separated the self-calibrated measurement set file and used WSClean

on it separately for multiple parameter values. There seemed to be some problem with the cleaning process. Increasing the niter considerably appears to yield better images. The niter given for *tclean* was 6400. By changing the niter to 10000, 50000, 100000 we could get images for the complete interval given in figure 6.1. The contours are also given in figure 6.2. In the end, we decided to set the niter as  $10^6$  and also use -auto-threshold 3 to clean down to 3 sigma of noise level.

Image Name	Flux Density (mJy)	Frequency (MHz)
J1809-1943-selfcalimg4-t0000-image.fits	$13.24 \pm 0.35$	642.82
J1809-1943-selfcalimg4-t0001-image.fits	$13.22 \pm 0.52$	642.82
J1809-1943-selfcalimg4-t0002-image.fits	$11.33 \pm 0.67$	642.82
J1809-1943-selfcalimg4-t0003-image.fits	$15.25 \pm 1.19$	642.82
J1809-1943-selfcalimg4-t0004-image.fits	$11.62 \pm 0.44$	642.82
J1809-1943-combined-img-t0004-image.fits	$13.28 \pm 0.48$	642.82

Table 6.1: Integrated Flux Density and Frequency for different images of J1809-1943 (niter= $10^6$ ).

We could consider the high value in the 4th time bin to be a significant peak in the flux density of the pulsar. This could mean that the pulsar is aligned with the line of sight of the telescope when it is the 4th time bin. Also, the first 2 bins have flux density values higher than the values of the 3rd and 5th bins. This could suggest some emissions in the off-pulse region of the pulsar. The figure 6.4 plots the flux density versus the time bins to visualize the electromagnetic pulse of the pulsar.

As an extension of this project, the pipeline could be further optimized to obtain the images using *tclean* or WSClean depending on the user's choice, which can be given through the configuration file. Also, the self-calibration could also be edited so that the pipeline only makes the images after all the rounds of self-calibration are done. So that we obtain the images of the five time intervals only of the visibility file whose self-calibration is complete. We could also investigate why the flux density in the images had inconsistencies. Further testing of the modified pipeline may also be conducted using different data sets.

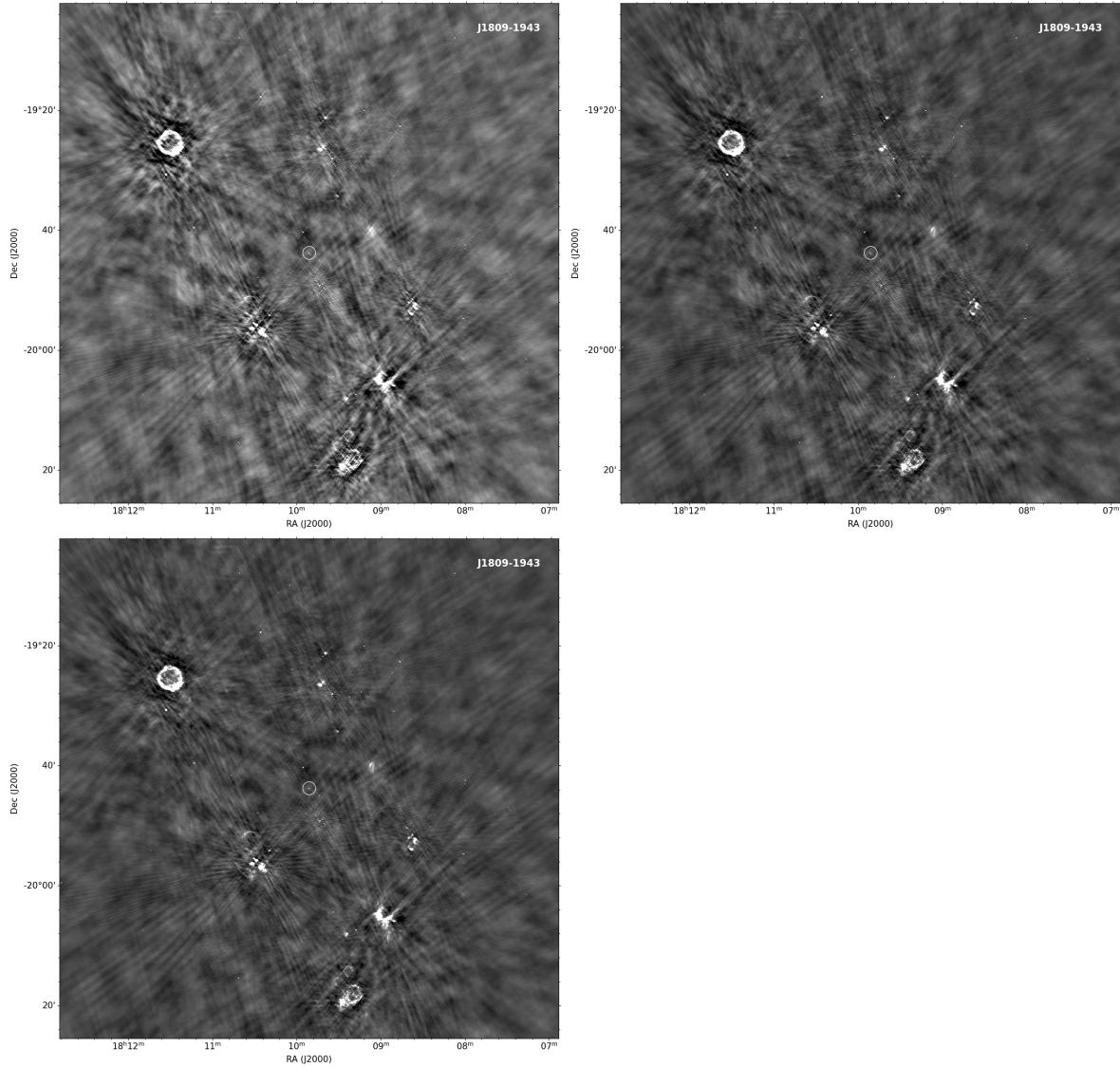


Figure 6.1: The image of the pulsar with different niter values. Top left niter=10000, top right niter=50000, bottom niter=100000

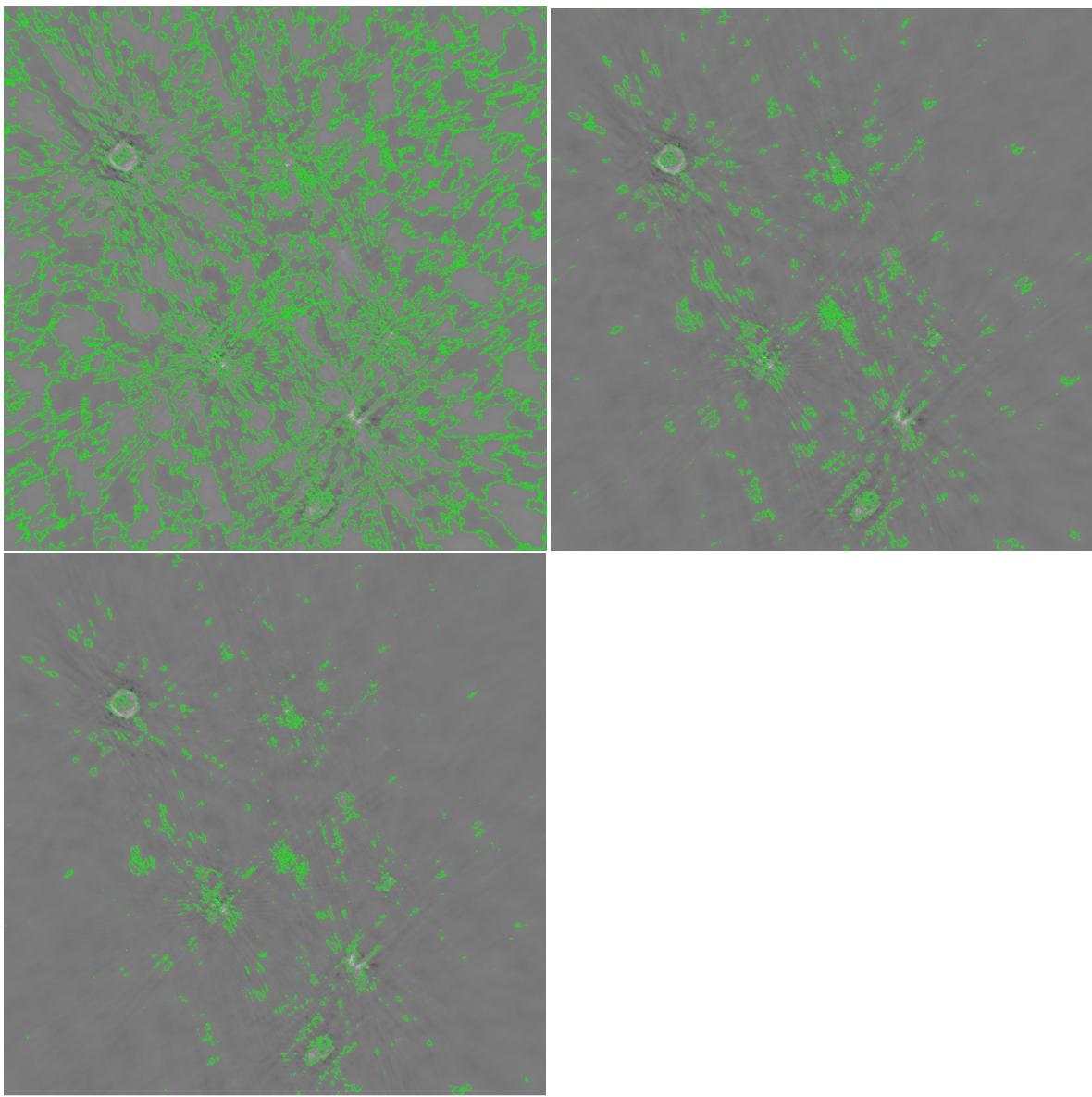


Figure 6.2: Contours of the above images with top left niter=10000, top right niter=50000, bottom niter=100000

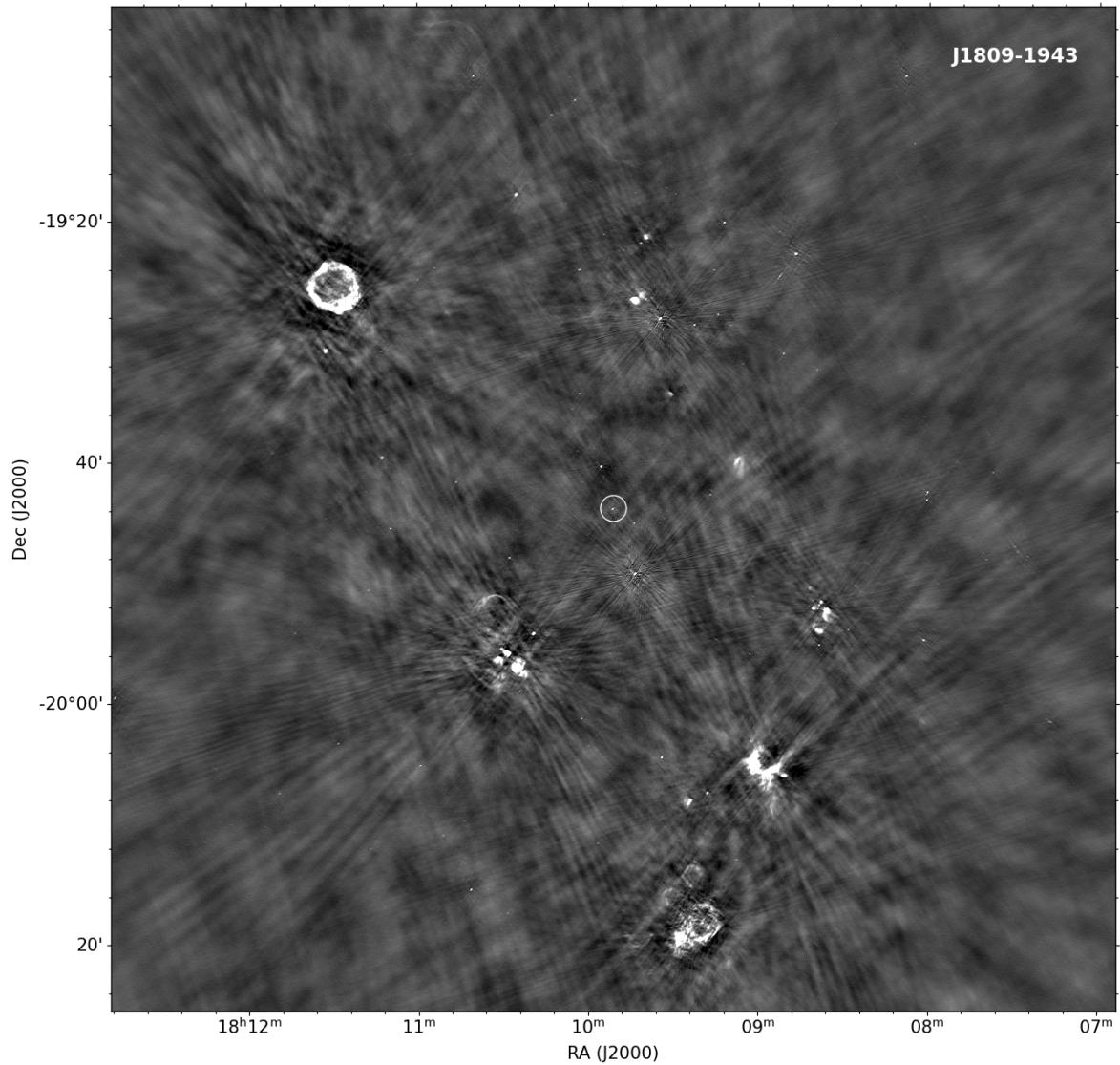


Figure 6.3: Combined image obtained from the modified pipeline (niter=10<sup>6</sup>)

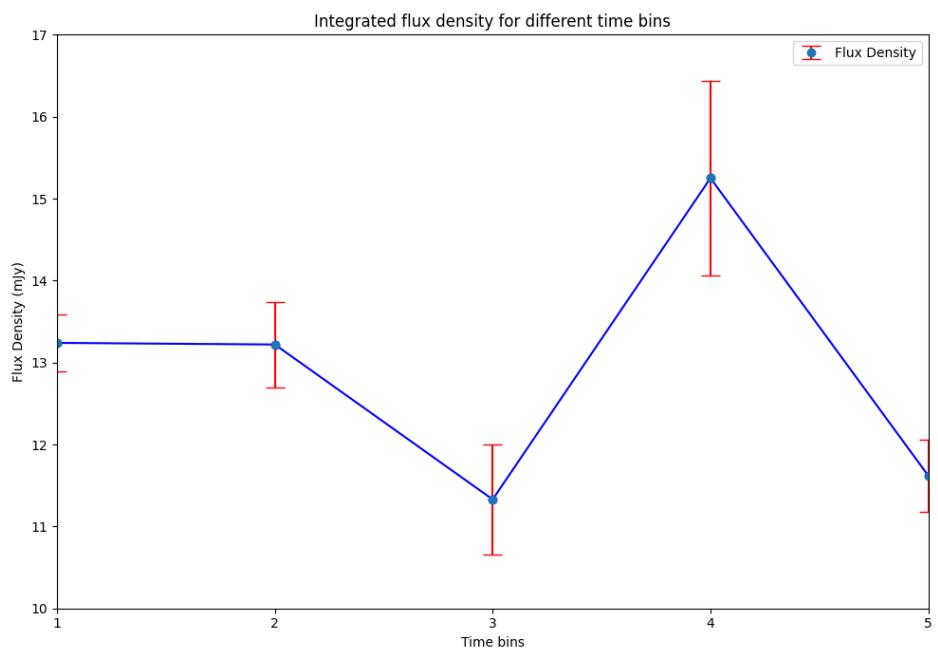


Figure 6.4: The figure shows the flux density map of the pulsar in each of the time bins. The visibility data of the pulsar was divided into 5 time bins and imaged to calculate the flux density.

# Bibliography

- [1] *The GMRT: System Parameters and Current Status*, 15 June 2023, [http://www.gmrt.ncra.tifr.res.in/doc/GMRT\\_specs.pdf](http://www.gmrt.ncra.tifr.res.in/doc/GMRT_specs.pdf).
- [2] Gupta, Y., et al, *The upgraded GMRT: opening new windows on the radio Universe.*, *Current Science*, 113(4), 2017, <http://www.gmrt.ncra.tifr.res.in/doc/ugmrt.pdf>.
- [3] Ananthakrishnan, S., *The Giant Metrewave Radio Telescope*, *J. Astrophys. Astron.*, 1995, 16, 427-435.
- [4] McMullin, J. P., Waters, B., Schiebel, D., Young, W., Golap, K., *CASA Architecture and Applications, Astronomical Data Analysis Software and Systems XVI*, ASP Conference Series, Vol. 376, 2007, R. A. Shaw, F. Hill and D. J. Bell, eds.
- [5] The CASA Team, et al., *CASA, the Common Astronomy Software Applications for Radio Astronomy*, *PASP*, 134, 114501. DOI: 10.1088/1538-3873/ac9642
- [6] Kale, Ruta. Ishwara-Chandra, C. H., *CAPTURE: a continuum imaging pipeline for the uGMRT*, *Experimental Astronomy*, 51:95-108, 2021, DOI: 10.1007/s10686-020-09677-6
- [7] Synthesis imaging, [https://casadocs.readthedocs.io/en/latest/notebooks/synthesis\\_imaging.html](https://casadocs.readthedocs.io/en/latest/notebooks/synthesis_imaging.html)
- [8] A. R. Offringa, B. McKinley, Hurley-Walker, and others, *WSClean: an implementation of a fast, generic wide-field imager for radio astronomy*, *MNRAS*, vol. 444, no. 1, pp. 606-619, 2014, DOI: 10.1093/mnras/stu1368.

- [9] B. G. Clark, *Coherence in Radio Astronomy*. Taylor, G. B., Carilli, C. L., & Perley, R. A. (Eds.). (1999). *Synthesis imaging in radio astronomy II* (Vol. 180).
- [10] A. Richard Thompson, *Fundamentals of Radio Interferometry*. Taylor, G. B., Carilli, C. L., & Perley, R. A. (Eds.). (1999). *Synthesis imaging in radio astronomy II* (Vol. 180).
- [11] Ed B. Fomalont, *Calibration and Editing*. Taylor, G. B., Carilli, C. L., & Perley, R. A. (Eds.). (1999). *Synthesis imaging in radio astronomy II* (Vol. 180).
- [12] CASA, *Common Astronomy Software Applications*, <https://casa.nrao.edu/>
- [13] VLA Continuum Tutorial 3C391–CASA6.4.1 – CASA Guides, (n.d.), [https://casaguides.nrao.edu/index.php?title=VLA\\_Continuum\\_Tutorial\\_3C391-CASA6.4.1](https://casaguides.nrao.edu/index.php?title=VLA_Continuum_Tutorial_3C391-CASA6.4.1)
- [14] Robitaille, T. and Bressert, E., *APLpy: Astronomical Plotting Library in Python, Astrophysics Source Code Library*, 2012, aug, <http://adsabs.harvard.edu/abs/2012ascl.soft08017R>
- [15] Robitaille, Thomas, *APLpy v2.0: The Astronomical Plotting Library in Python*, feb, 2019, DOI: 10.5281/zenodo.2567476, <https://doi.org/10.5281/zenodo.2567476>
- [16] Common Astronomy Software Applications,<https://casadocs.readthedocs.io/en/stable/>