How accurate are those ALMA measurements

An investigation of JvM correction factor for CLEAN implementation in ALMA data reduction pipeline

MSc thesis

Taotao Yang



Universiteit Leiden
Observatory

How accurate are those ALMA measurements

An investigation of JvM correction factor for CLEAN implementation in ALMA data reduction pipeline

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Astronomy

Author:	Taotao Yang
Student ID:	s4866835
Supervisor:	Prof. M.R. Hogerheijde
Second Supervisor:	Dr. J. van Marrewijk
Second Corrector:	Prof. H.J. van Langevelde
Project Duration:	October, 2025 – July, 2026

Cover Image:	ALMA and the centre of the Milky Way
Cover Credit:	ESO/B. Tafreshi (twanight.org)
Template Style:	Thesis style by Richelle F. van Capelleveen
Template Licence:	Licenced under CC BY-NC-SA 4.0



Universiteit Leiden
Observatory

P.O. Box 9513, 2300 RA Leiden, The Netherlands

Abstract

Context: Flux recovery is a crucial step in astronomy. However, the process of recovering accurate flux for interferometric observations is not trivial. Interferometric image reconstruction with `CLEAN` introduces a unit mismatch between the clean model, in units of clean beams, and the residual, in units of dirty beams. This results in potential discrepancies in the total source flux estimates. The Jorsater van Moorsel (JvM) correction addresses this problem by scaling the residual prior to combining with the convolved clean model. However, methods for JvM factor calculations are not rigorously defined for accurate total source flux estimates for extended sources.

Aim: We aim to determine the integration limit of the dirty beam area for JvM factor calculation that returns the most accurate total source flux estimates.

Methods: We compare two definitions of dirty beam integration limit. The first-null method that integrates to a first-null outlined in Czekala et al. (2021). The source-extent method that integrates to the source extent, where the same aperture where the total source flux is evaluated. We apply these methods over ALMA HD 163296 MAPS data and synthetic ALMA observations across extended, compact, and point sources with known input flux.

Results: With HD 163296 MAPS data, we record a JvM factor of 0.29 with the first-null method, consistent with the value reported in Öberg et al. (2021), and 0.95 for source-extent method. We observe the source-extent method produce approximately an order of magnitude more in-aperture source flux estimates than the first-null method. In simulated cases, the first-null method produce 12.6% total source flux input, while the source-extent method returns 98.1% for extended source. For compact and point sources, the two methods converge and provide $91 \pm 3\%$ total source flux estimates.

Conclusions: Accurate total source flux estimates are better achieved by adopting the source-extent method. The first-null method remains robust for compact sources in reducing side lobe response, while being less reliable for extended emissions. We recommend ALMA users to form JvM factors with the source-extent method for both compact and extended sources and report detailed data reduction parameters to ensure reproducibility.

Acknowledgments

We thank Prof. Michiel R. Hogerheijde and Dr. Joshiwa van Marrewijk at Leiden Observatory for their dedicated supervision, enlightening insights and discussions on radio interferometry, and patience throughout this thesis. It is with their guidance and support that clarified the direction of this thesis and led it to fruition. We also thank Prof. Huib J. van Langevelde for serving as the second reader for this thesis.

We thank Allegro, the European ALMA Regional Center node in the Netherlands, and the Allegro team for facilitating technical infrastructure and access to computational resources. We further thank the Allegro team and Dr. Joshiwa van Marrewijk for hosting biweekly group meetings on technical developments regarding ALMA and radio astronomy and allowing us to participate in this forum of knowledge.

We acknowledge that this thesis makes use of the following ALMA data: ADS/-JAO.ALMA#2018.1.01055.L. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), NSTC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

Contents

Abstract iii

Acknowledgments iv

List of Figures vii

Nomenclature viii

1 Introduction 1

1.1 Radio Interferometry and Image Reconstruction 1

1.1.1 Visibilities and the Dirty Image 1

1.1.2 The CLEAN Algorithm 2

1.2 The JvM Correction 2

1.2.1 The JvM Factor 2

1.2.2 The Integration Limit Problem 2

1.3 Thesis Introduction 3

2 Methods 4

2.1 The CLEAN Algorithm 4

2.2 JvM Factor Calculation 5

2.3 Validation Data 6

2.4 Simulations 6

3 Results 8

3.1 Validation Data 8

3.2 Simulations 9

3.2.1 Extended Source 9

3.2.2 Compact Source 10

3.2.3 Point Sources 11

4 Discussions 12
4.1 Which JvM Method is More Accurate 12
4.2 Implications on Reported Values in Literature 12
4.3 Improvements 13
5 Conclusions 14
Bibliography 15
Appendices 17
A CASA tclean 17
A.1 Parameter Assignment 17
A.1.1 Image Geometry 17
A.1.2 Spectral Parameters 17
A.1.3 Deconvolution Parameters 18
A.2 Data Products 19
B CASA simobserve 20
B.1 Skymodel Generation 20
B.1.1 Image Geometry 20
B.1.2 Building Headers 20
B.1.3 Source Characterization 21
B.2 Antenna Configurations 21
B.2.1 Multiple Observations 21
B.2.2 Filtering Antennas 22
B.2.3 Parameter Assignment 22
B.3 Simulation Setup 22
B.3.1 Necessary Ingredients 23
B.3.2 Image Geometry 23
B.3.3 Spectral Parameters 23
B.3.4 Observational Parameters 23
B.3.5 Pointing Parameters 24
B.4 Post Simulation 24
C Code 25
C.1 Workflow 25
C.2 Dependency 26
D Beam Diagnostics 28

List of Figures

2.1 Schematic overview of tclean 5
2.2 HD 163296 MAPS data 6
2.3 Sky model images 7

3.1 In-aperture flux estimates for HD 163296 9
3.2 In-aperture flux estimates for synthetic extended source 10
3.3 In-aperture flux estimates for compact sources 11

B.1 Antenna configuration for HD 163296 MAPS data 22

D.1 Beam diagnostics 28

Nomenclature

Acronyms

Notation	Description	Page List
ALMA	Atacama Large Millimeter/submillimeter Array	3, 6, 7, 9, 11, 14, 18, 21–23
CASA	Common Astronomy Software Applications	2, 7, 18, 21, 25–27
Dec	Declination	20, 21, 23
FWHM	Full Width at Half Maximum	4, 5, 9, 14
JvM	Jorsater van Moorsel	iii, 1–14, 23, 25
LOC	Local Tangent Plane Coordinates	22
LSRK	Local Standard of Rest, Kinematic	18
PSF	Point Spread Function	4, 12, 19
RA	Right Ascension	20, 21, 23
rms	Root Mean Square	4, 13, 18

Notation	Description	Page List
WCS	World Coordinate System	21, 26

Constants

Notation	Description	Unit
k	constant of Boltzmann	$1.381 \times 10^{-23}\,\mathrm{J\,K^{-1}}$
c	speed of light	$2.998 \times 10^{8}\,\mathrm{m\,s^{-1}}$
e	elementary charge	$1.602 \times 10^{-19}\,\mathrm{C}$
ϵ_0	vacuum permittivity	$8.854 \times 10^{-12}\,\mathrm{C^2\,kg^{-1}\,s^2}$
G	gravitational constant	$6.674 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$
m_0	atomic mass unit	$1.661 \times 10^{-27}\,\mathrm{kg}$
$M_{B\odot}$	absolute B-band magnitude of the sun	5.48
m_e	rest mass of an electron	$9.109 \times 10^{-31}\,\mathrm{kg}$
m_p	rest mass of a proton	$1.673 \times 10^{-27}\,\mathrm{kg}$
$M_{V\odot}$	absolute visual magnitude of the sun	4.83
$m_{V\odot}$	apparent visual magnitude of the sun	-26.75
N_A	constant of Avogadro	$6.022 \times 10^{23}\,\mathrm{mol^{-1}}$
h	constant of Planck	$6.626 \times 10^{-34}\,\mathrm{J\,Hz^{-1}}$
R	gas constant	$8.314\,\mathrm{J\,K^{-1}\,mol}$
σ_{SB}	constant of Stefan-Boltzmann	$5.670 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$
σ_e	Thomson crossection of an electron	$6.652 \times 10^{-29}\,\mathrm{m^2}$
$T_\odot$	effective temperature of the sun	5778K

Units

Notation	Description	Unit
$'$	arcminute	$\frac{1}{60}\,^\circ$
$''$	arcsecond	$\frac{1}{3600}\,^\circ$
AU	astronomical unit	$1.496 \times 10^{11}\,\mathrm{m}$
ρ	density	$\mathrm{kg\,m^{-3}}$
eV	electronvolt	$1.602 \times 10^{-19}\,\mathrm{J}$
F	flux	$\mathrm{J\,s^{-1}\,m^{-2}}$
Gyr	billion years (gigayear)	$1 \times 10^{9}\,\mathrm{year}$
Jy	Jansky	$1 \times 10^{-26}\,\mathrm{W\,m^{-2}\,Hz^{-1}}$

Notation	Description	Unit
$L_\odot$	solar luminosity	$3.828 \times 10^{26} \mathrm{J\,s^{-1}}$
L	luminosity	$\mathrm{J\,s^{-1}}$
mas	milliarcsecond	$0.001''$
$M_\oplus$	Earth mass	$5.972 \times 10^{24} \mathrm{kg}$
M_J	Jupiter mass	$1.898 \times 10^{27} \mathrm{kg}$
$M_\odot$	solar mass	$1.988 \times 10^{30} \mathrm{kg}$
κ	opacity	$\mathrm{m^2\,kg^{-1}}$
pc	parsec	$3.086 \times 10^{16} \mathrm{m}$
I	intensity	$\mathrm{J\,s^{-1}\,m^{-2}\,sr^{-1}}$
$R_\oplus$	Earth radius	$6.378 \times 10^{6} \mathrm{m}$
R_J	Jupiter radius	$7.149 \times 10^{7} \mathrm{m}$
$R_\odot$	solar radius	$6.957 \times 10^{8} \mathrm{m}$
F_λ	spectral flux	$\mathrm{J\,s^{-1}\,m^{-3}}$
F_ν	spectral flux	$\mathrm{J\,s^{-1}\,Hz^{-1}\,m^{-2}}$
I_λ	specific intensity	$\mathrm{J\,s^{-1} m^{-3}\,sr^{-1}}$
I_ν	specific intensity	$\mathrm{J\,s^{-1}\,Hz^{-1}\,m^{-2}\,sr^{-1}}$

1. Introduction

Modern radio interferometry enables high-resolution observations by combining signals from multiple antennas. The measured visibilities are connected to the sky brightness through a Fourier transform. Due to partial sampling on the uv-plane, the direct Fourier transform returns a dirty image, where the true sky brightness is convolved with the dirty beam. Careful implementation of deconvolution and image reconstruction is needed to obtain accurate source flux estimates. In this chapter, we introduce the relevant concepts, the JvM correction, and the research question of this thesis.

1.1 Radio Interferometry and Image Reconstruction

The need for interferometry stems from the diffraction-limited angular resolution in a single telescope (Thompson et al., 2017). The angular resolution θ is dependent on the main aperture diameter of a telescope D when observing light at some wavelength λ:

$$\theta = 1.22\frac{\lambda}{D} \tag{1.1}$$

Constructing a single telescope with a large aperture for sub-arcsecond resolution would quickly become unfeasible as aperture diameter increases. Radio interferometry allows for the placement of individual telescopes to synthesize a larger aperture. These spatially distributed telescopes would have a baseline B between each other, where B represents the spatial separation between the two antennas. We write the resolution θ for an interferometer with a baseline B as:

$$\theta = 1.22\frac{\lambda}{B} \tag{1.2}$$

1.1.1 Visibilities and the Dirty Image

A radio interferometer measures visibilities $V(u, v)$, which are connected to the true sky brightness $I(l, m)$ (Briggs et al., 1999). The two quantities are associated via Fourier transform through the van Cittert-Zernike theorem:

$$V(u, v) = \iint I(l, m) \exp\left[-2\pi i(ul + vm)\right] dl\, dm \tag{1.3}$$

$$I(l, m) = \iint V(u, v) \exp\left[2\pi i(ul + vm)\right] du\, dv \tag{1.4}$$

In practice, the uv-plane is only partially sampled. The direct Fourier transform of the visibilities therefore yields a dirty image, where the true sky brightness is convolved with the dirty beam. Additionally, the dirty beam is the Fourier transformed uv-plane coverage, where side lobe responses may lead to complications in flux measurements.

1.1.2 The CLEAN Algorithm

Common image reconstruction methods involve the CLEAN algorithm and its modern implementation `tclean` in Common Astronomy Software Applications (CASA) (Högbom, 1974; McMullin et al., 2007; The CASA Team et al., 2022). The algorithm iteratively models the image as a collection of point sources. During each iteration, the algorithm searches for the brightest peak in the residual, subtracts a scaled dirty beam, and builds a CLEAN component. After convergence, the CLEAN components are convolved with a clean beam. This clean beam is commonly approximated as a Gaussian fitted to the main lobe of the dirty beam. The final reconstructed image is therefore formed by combining the convolved clean model C with the residual R:

$$\text{Reconstructed Image} = C + R. \tag{1.5}$$

1.2 The JvM Correction

The process of `CLEAN` results in a unit mismatch issue. The dirty image and the residual are shown in units of Jy / Dirty Beam, while the CLEAN model is expressed in units of Jy / px. After convolving the CLEAN model with the clean image, the resulting convolved clean model is in units of Jy / Clean Beam, whereas the residual remains in units of Jy / Dirty Beam. We therefore employ a JvM factor to correct for the unit mismatch in the final image by scaling the residual (Jorsater & van Moorsel, 1995).

1.2.1 The JvM Factor

The unit mismatch can bias total source flux estimates. We write the JvM correction factor ϵ in terms of a corrected flux G, the convolved clean model C, and some residual R:

$$G = C + \epsilon R \tag{1.6}$$

In practice, we commonly adopt a beam-area method for JvM factor calculation:

$$\epsilon = \frac{\Omega_{\text{clean}}}{\Omega_{\text{dirty}}}. \tag{1.7}$$

Here Ω_{clean} and Ω_{dirty} are the integrated areas of the clean and dirty beams, respectively. The clean beam area is computed by its Gaussian parameters, while the dirty beam area requires integrating the dirty beam out to some limit.

1.2.2 The Integration Limit Problem

The integration limit for Ω_{dirty} is not rigorously defined. Czekala et al. (2021) suggests integrating the dirty beam out to its first null. The first null is chosen to represent the main lobe, thus eliminating side lobe response. While a first-null method is commonly

adopted for compact sources, this method becomes problematic for more extended sources. The proposed first null may be contained within the extent of an extended source, causing the JvM factor to be evaluated over a regime that contains only a fraction of the total source flux.

We hypothesize that a more accurate JvM factor is obtained by integrating the dirty beam area over the same aperture that is used to measure the total source flux. That is, we ensure that the total source flux measurements match with the source's extent.

We compare two definitions of Ω_{dirty}: 1) the first-null method of integrating out to the first null, r_{null}, as outlined by Czekala et al. (2021), and 2) the source-extent method of integrating over the same aperture that is used to measure the total source flux, r_{extent}. We hypothesize that the latter would yield more accurate JvM factor, as it ensures that Ω_{dirty} corresponds to the source's extent, where the source flux is measured.

1.3 Thesis Introduction

In this thesis, we aim to investigate the proper procedure in raw Atacama Large Millimeter/submillimeter Array (ALMA) data reduction, with the ultimate goal of understanding and implementing a standard operation procedure for optimal total source flux and error estimates through JvM corrections. We search for an ideal integration limit for JvM factor calculation for accurate total source flux estimates. We investigate this problem with complementary approaches. First, we validate the JvM correction factor results with real data (Öberg et al., 2021; Czekala et al., 2021). Second, we employ synthetic ALMA observations via `simobserve` with known input fluxes for comparisons.

We note that the JvM factor finding problem is distinct from the missing-flux problem. This is because the missing flux arises from incomplete uv-coverage, as a baseline B must be greater than the sum of two antenna dish radii (Braun & Walterbos, 1985; Cornwell et al., 1999). This thesis addresses only the unit mismatch and the choice of integration limit in the JvM correction.

The rest of this thesis is organized as follows: Chapter 2 explains the technical execution for achieving `CLEAN` and JvM correction, as well as their data products; Chapter 3 describes the simulation setup and results; and lastly, Chapter 4 performs analysis and discussion on the data, followed by a summary in Chapter 5.

2. Methods

In this chapter, we illustrate the CLEAN algorithm and explain the associated data products in detail. We then showcase the steps involved for JvM factor calculation following the beam-area method outlined in Eq. (1.7). We then present the real dataset for validation and the simulation setup.

2.1 The CLEAN Algorithm

As mentioned in Section 1.1.2`tclean` reconstructs images by iteratively subtracting point-source responses from the dirty images. The `tclean` algorithm consists of major and minor cycles. We present the algorithm in Fig. 2.1.

The `tclean` algorithm iteratively populates the clean model during minor cycles, while major cycles recalculate the residual by the updated model and visibilities. This repeats until a stopping condition is met, such as a noise threshold represented in units of source image Root Mean Square (rms). The rms is commonly set by the out-of-mask region in the dirty image (Condon, 1997). Shown in Eq. (1.5), the final reconstructed image is built from linear combination between the convolved clean model C and a residual R.

The convolved clean model C is build by first forming a Gaussian kernel for convolution from the clean beam, where the Full Width at Half Maximum (FWHM) values can be accessed via headers. We therefore transform the semi-major and semi-minor axes $\theta_{\mathrm{maj}}, \theta_{\mathrm{min}}$ into Gaussian standard deviations σ_{maj} and σ_{min} via:

$$\begin{aligned} \sigma_{\mathrm{maj}} &= \frac{\theta_{\mathrm{maj}}}{2\sqrt{2\ln 2}} \\ \sigma_{\mathrm{min}} &= \frac{\theta_{\mathrm{min}}}{2\sqrt{2\ln 2}} \end{aligned} \tag{2.1}$$

Several data products are available from `tclean`: 1) the reconstructed clean image `.image`, 2) the residual `.residual`, 3) the clean model `.model`, 4) the mask `.mask`, 5) and the Point Spread Function (PSF) `.psf`. We explain the detailed information of these data products and the associated standard cleaning parameters in Appendix A. We note that the standard cleaning parameters are setup in alignment with Czekala et al. (2021), and we refer to this `tclean` call as the standard cleaning methods for this thesis.

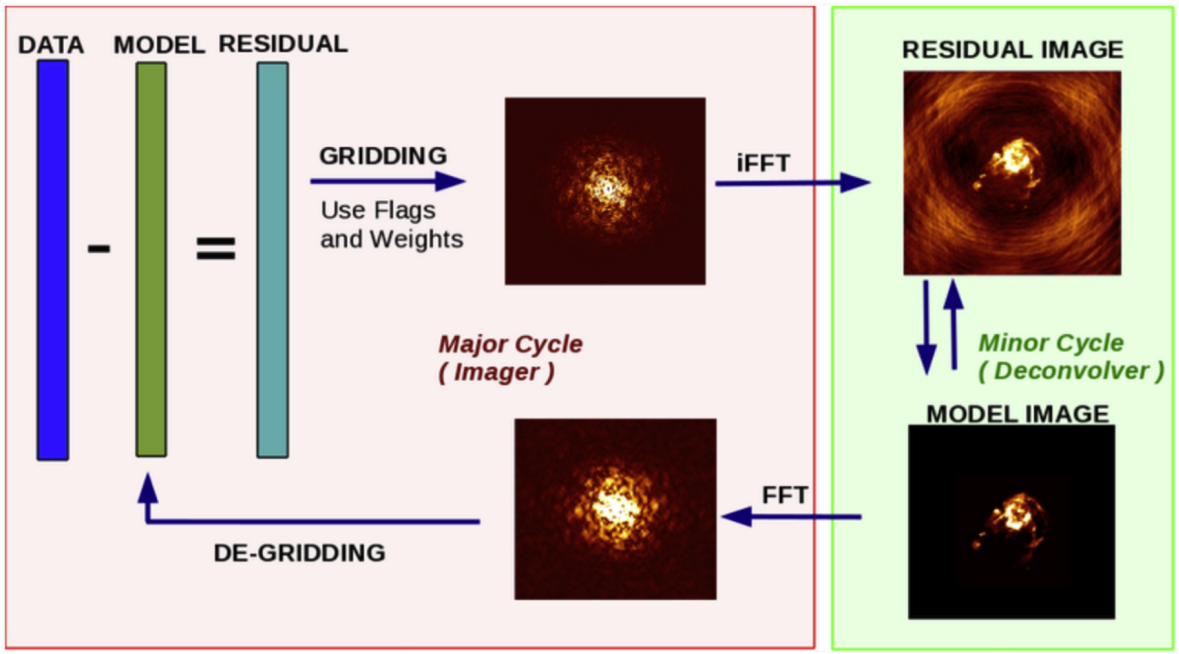


Figure 2.1: Schematic overview of the iterative image reconstruction in the CASA task `tclean`. The red-shaded region indicates the major cycle of `tclean` algorithm, while the green area represents the minor cycles. The Fourier transforms take place during the major cycles to convert raw data into images, while minor cycles occur only on the images for deconvolution. *Image credit by The CASA Team et al. (2022).*

2.2 JvM Factor Calculation

The JvM factors following Eq. (1.7) requires two elements: 1) the clean beam area $\Omega_{\rm clean}$ and 2) the dirty beam area $\Omega_{\rm dirty}$. The clean beam area $\Omega_{\rm clean}$ is characterized by its FWHM as semi-major axis $\theta_{\rm maj}$ and semi-minor axis $\theta_{\rm min}$, and its position angle ϕ. That is, we write $\Omega_{\rm clean}$ as:

$$\Omega_{\rm clean} = \frac{\pi}{4\ln 2}\theta_{\rm maj}\theta_{\rm min} \tag{2.2}$$

The dirty beam area is dependent on the integration limit of the dirty beam. Since the dirty beam is elliptical, we must first deproject it into a circular geometry. This deprojection is performed on its coordinates (l, m). We first rotate the (l, m) coordinates by the position angle ϕ of the clean beam:

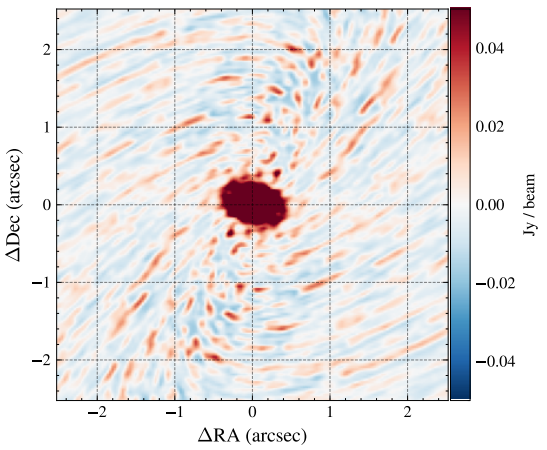
$$\begin{pmatrix} l' \\ m' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} l \\ m \end{pmatrix}. \tag{2.3}$$

We then compute an effective axes length $\sigma_{\rm eff}$ from its semi-major axis $\sigma_{\rm maj}$ and semi-minor axis $\sigma_{\rm min}$:

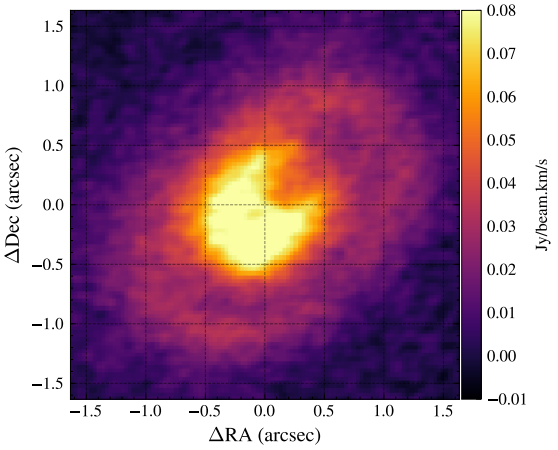
$$\sigma_{\rm eff} = \sqrt{\sigma_{\rm maj}\sigma_{\rm min}} \tag{2.4}$$

Thus, we obtain the deprojected coordinates $(l_{\rm deproj}, m_{\rm deproj})$ as:

$$\begin{pmatrix} l_{\rm deproj} \\ m_{\rm deproj} \end{pmatrix} = \sigma_{\rm eff} \begin{pmatrix} l'/\sigma_{\rm min} \\ m'/\sigma_{\rm maj} \end{pmatrix}. \tag{2.5}$$



(a) The dirty beam for HD 163296. This is the Fourier transformed uv-plane coverage.



(b) The moment0 map for HD 162396. This is the integrated intensity of the spectrum.

Figure 2.2: Unprocessed HD 163296 MAPS data showing the dirty beam in Fig. 2.2a and the integrated spectral intensity in Fig. 2.2b.

Consequently, we obtain the deprojected radial coordinate r_{deproj} as:

$$r_{\mathrm{deproj}} = \sqrt{l^2_{\mathrm{deproj}} + m^2_{\mathrm{deproj}}} \tag{2.6}$$

The dirty beam area is then the integration of deprojected and radially averaged dirty beam profile $I_{\mathrm{dirty}}(r_{\mathrm{deproj}})$ along r_{deproj} up to some integration limit. In Czekala et al. (2021), this integration limit is defined as the first null r_{null}, which represents the first zero crossing of r_{deproj}:

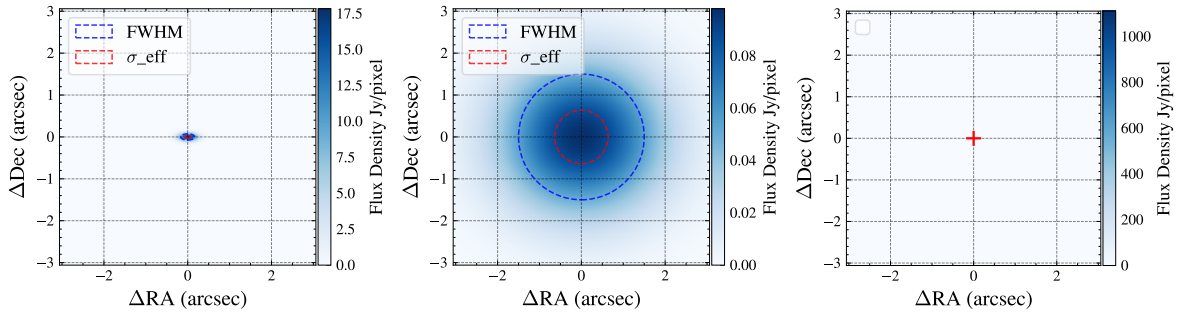
$$\Omega_{\mathrm{dirty}} = \int_0^{r_{\mathrm{null}}} I_{\mathrm{dirty}}(r_{\mathrm{deproj}})\, dr_{\mathrm{deproj}} \tag{2.7}$$

2.3 Validation Data

We validate the JvM factor calculation method with HD 163296 MAPS data (Öberg et al., 2021). We refer to this data set as the validation data set. We study the HCN $J = 3 - 2$ molecular line. This real data set combines extended and compact ALMA antenna configurations for observation, resulting in a dirty beam with a pronounced shelf due to disk substructures and significant difference in the clean and dirty beam areas (Isella et al., 2016; Andrews et al., 2018). We adhere to the standard cleaning parameters to produce the initial data products from `tclean`. We then compare the JvM correction factors derived from r_{null} and r_{extent}, and their associated total source flux estimates. We showcase the dirty beam and the moment 0 map of the validation data in Fig. 2.2.

2.4 Simulations

With HD 163296 MAPS data, we test the implementation of JvM factor calculation listed in Eq. (1.7). However, real data is populated with noise and therefore is deficient



(a) Elliptical Gaussian. **(b)** Extended Gaussian. **(c)** Point-source.

Figure 2.3: Sky model images. Sources are approximated with Gaussian $\sigma_{\rm eff}$. Figure shows elliptical and circular Gaussian sources, along with point-source sky model. Red marker indicate point source pixel location. The FWHM and $\sigma_{\rm eff}$ is overlaid for visualizing the underlying parameters used for sky model images generation.

in a known ground truth of total source flux present. We employ CASA `simobserve` to create synthetic ALMA observations, where artificial sky model images allow for prior knowledge on total input fluxes. Therefore, we enable comparison between the total source flux estimations derived from JvM corrections and the total input flux. The simulation setup requires two components: 1) the sky model images and 2) the antenna configurations. We list the parameters used for `simobserve` in Appendix B.

We build synthetic sky model images for: 1) point sources, 2) compact sources, and 3) extended sources. The non-point sources are represented by some Gaussian. We define its semi-major and semi-minor axes in arcsecond units, which are then translated into Gaussian effective axis $\sigma_{\rm eff}$ following Eq. (2.1). For point sources, we assign all input fluxes onto a single pixel at image center. We show the examples of generated sky model images in Fig. 2.3. The synthetic observations are then processed with `tclean` following the parameters listed in Appendix A. In this step, we derive the extent $r_{\rm extent}$ of the simulated observation targets by the convergence of their cumulative source flux along the deprojected radial.

Shown in Appendix B, the simulation antenna configurations are built to reproduce that of the HD 163296 MAPS data set. We replicate the same uv-plane coverage, and consequently the dirty beam of the validation data set. We export antenna positions and station IDs with `analysisUtils` and `listobs` (Hunter et al., 2023). This allows us to generate synthetic observations with the same compact and extended configurations that produced the dirty-beam shelf in the real data.

3. Results

In this chapter, we illustrate the results of applying beam-area JvM correction factor calculation on the validation and synthetic datasets, as outline in Eq. (1.7). For synthetic datasets, we include comparison between known total input fluxes and estimated total source fluxes. Every measurement set in this chapter is cleaned with the standard `tclean` parameters outlined in Appendix A to a stopping criteria at noise threshold of 4σ. The four test cases are listed in Table 3.1.

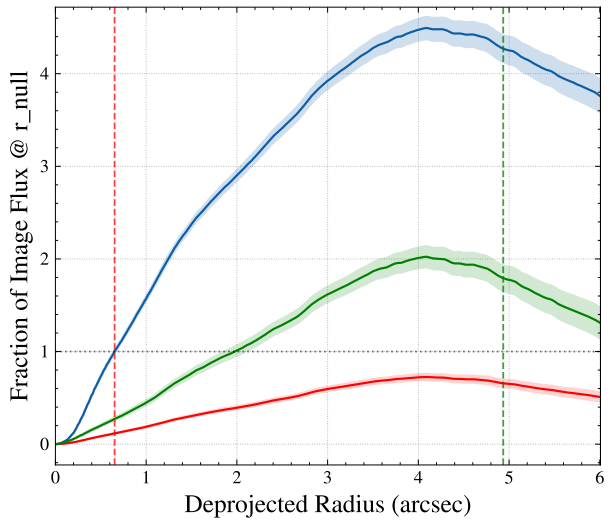
Table 3.1: Definition of the four test cases. The JvM radius is where the JvM factor is evaluated. The aperture radius is the aperture over which the total source flux is integrated.

Case	JvM Radius	Aperture Radius
C1	$r_{\rm null}$	$r_{\rm null}$
C2	$r_{\rm null}$	$r_{\rm extent}$
C3	$r_{\rm extent}$	$r_{\rm null}$
C4	$r_{\rm extent}$	$r_{\rm extent}$

3.1 Validation Data

We derive JvM factors for HD 163296 MAPS HCN $J = 3 - 2$ data following Eq. (1.7). The first-null method gives $\epsilon_{\rm null} = 0.29$, consistent with the value of $\epsilon = 0.30$ reported by Öberg et al. (2021) for the same dataset. The full diagnostic plot following Figure 4 notation in Czekala et al. (2021) is listed in Appendix D. The source-extent method, which integrates the dirty beam out to the extent of the `tclean` mask along the deprojected radial, gives $\epsilon_{\rm extent} = 0.95$. The two values differ by a factor of ~ 3. The test cases for the validation dataset is shown in Fig. 3.1.

Figure 3.1 shows the cumulative in-aperture flux estimates and errors for the validation dataset. Since no ground truth is present for real data, we conveniently normalize the flux curves to the in-aperture clean image flux evaluated at $r_{\rm null}$. At small radii, the JvM-corrected clean image flux curves derived from JvM factors at $r_{\rm null}$ and $r_{\rm extent}$ situate below the clean image flux curve. This is the expected behavior of JvM correction, as a JvM factor downweights the residual R. We notice C1 case estimates 11.6% of clean image flux reference, while C4 case estimates at 179.2%. That is, by evaluating a JvM factor at the object's extent, and consequently integrating over the same aperture for estimates, we obtain ~ 2 factor of in-aperture flux when compared




```
In-aperture Total Src Flux Estimates
--------  -----------  --------------
JvM(r)    Aperture(r) Total Src Flux
--------  -----------  --------------
r_null    r_null               11.62%
r_null    r_extent             65.65%
r_extent  r_null               27.24%
r_extent  r_extent            179.24%

r_null           = 0.65 arcsec
r_extent         = 4.94 arcsec
JvM @ r_null     = 0.29
JvM @ r_extent   = 0.95
Source flux ref  = 4.10 Jy
```

Clean img $\pm 2\sigma$
JvM corrected img | JvM(r_null) $\pm 2\sigma$
JvM corrected img | JvM(r_extent) $\pm 2\sigma$
r_null = 0.65"
r_extent = 4.94"

Figure 3.1: In-aperture flux estimates for HD 163296. The left pane shows cumulative in-aperture flux with shaded 2σ error envelopes as a function of deprojected radius. The red and green vertical dashed lines mark $r_{\rm null}$ and $r_{\rm extent}$, respectively. The right pane prints the in-aperture flux for the four test cases listed in Table 3.1. The flux normalization is based off of the in-aperture clean image flux at $r_{\rm null}$.

with reference. In comparison, C4 case produces in-aperture flux estimates by ~ 1.5 orders more than that of C1 case. This difference stems from the different choice of integration limits used for $\Omega_{\rm dirty}$, where integrating at $r_{\rm extent}$ contributes more flux from the residual in Eq. (1.6).

We confirm with the validation dataset that the reported JvM factor in Öberg et al. (2021) following the `tclean` methods listed in (Czekala et al., 2021) yields consistent result with our own approach. We further show that the choice of integration limit substantially alters the in-aperture flux estimates. However, we are unable to confirm which integration limit is the correct choice for accurate in-aperture flux estimates. We continue the investigations with synthetic datasets.

3.2 Simulations

The synthetic dataset is constructed from three distinct classes of sky model images: 1) extended Gaussian, 2) compact Gaussian, and 3) point sources. These sky model images are visualized in Fig. 2.3. We simulate the ALMA observations following the same antenna configurations for HD 163296 MAPS data, shown in Section 2.4. The simulated measurement sets are then cleaned with parameters specified in Appendix A to reach a stopping criteria of 4σ noise threshold. Since the total input flux is a known quantity for each synthetic sky model image, we establish these total source fluxes as the ground truth for comparison.

3.2.1 Extended Source

For extended sources, we construct a circular Gaussian with FWHM=3.00″. This translates to $\sigma_{eff} = 1.27''$. We showcase the results of this case in Fig. 3.2. Here we

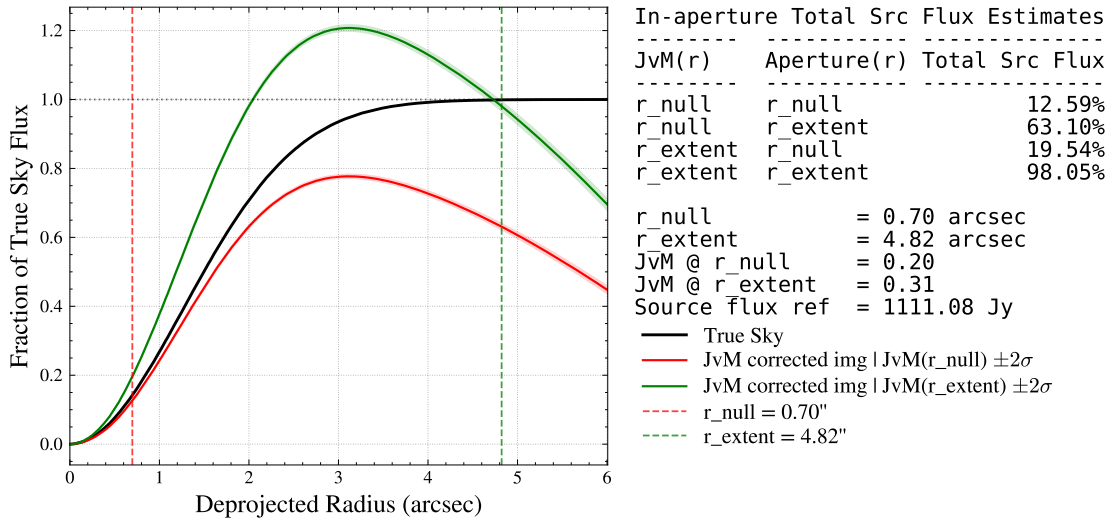
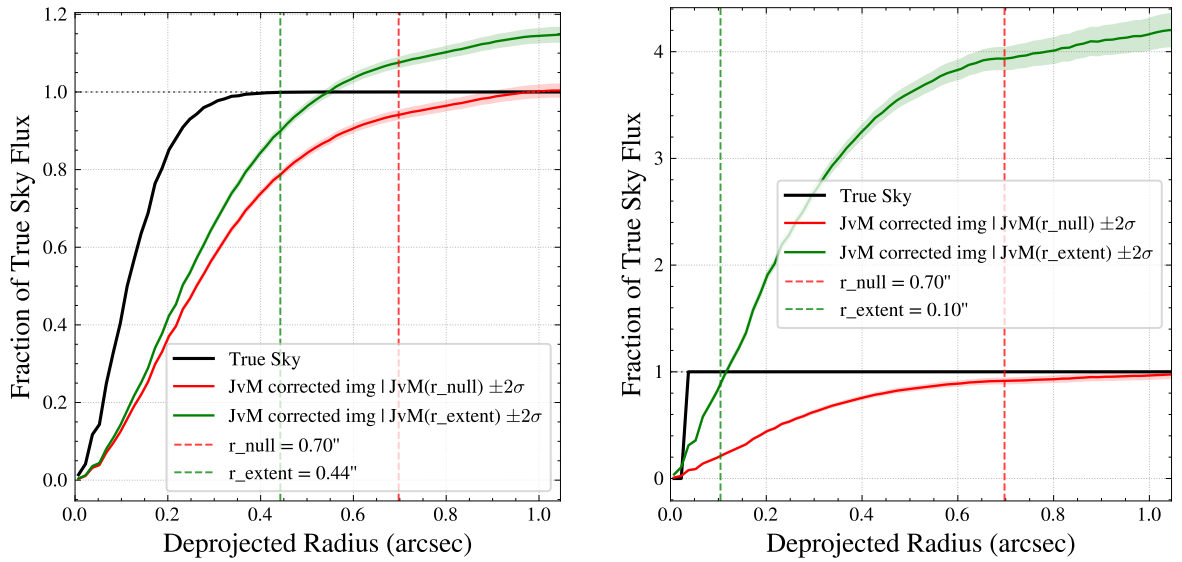


Figure 3.2: In-aperture flux estimates for synthetic extended source. The sky model image represents a Gaussian with $\sigma_{\rm eff} = 1.27''$. The left pane shows cumulative in-aperture flux with shaded 2σ error envelopes as a function of deprojected radius. The red and green vertical dashed lines mark $r_{\rm null}$ and $r_{\rm extent}$, respectively. The right pane prints the in-aperture flux for the four test cases listed in Table 3.1. The flux normalization is based off of the total flux of the true sky model image.

notice the integration limit varies significantly across $r_{\rm deproj}$, with $r_{\rm null} = 0.70''$ and $r_{\rm extent} = 4.82''$. This connects to a 12.6% total source flux estimates with integration limit $r_{\rm null}$ and a 98.1% for $r_{\rm extent}$. By adopting the integration limit that is consistent with source extent on extended sources, we notice a factor ~ 8 difference in total source flux estimates. This approximates an order of magnitude difference. The 2σ error bands contributes little to the percentage of total source flux estimates for extended sources. Here in test case C2, changing the total flux estimation limit from $r_{\rm null}$ to $r_{\rm extent}$ for a JvM correct flux curve derived from JvM factor evaluated at $r_{\rm null}$ significantly improves the results from test case C1 by factor ~ 5. There is a similar improve when compared between test case C3 and C4. Therefore, for extended sources, where the source extent is significantly larger than that of the beam size, adopting $r_{\rm extent}$ as the aperture for both JvM factor evaluation and source flux integration is the optimal solution.

3.2.2 Compact Source

The results for synthetic compact Gaussian sources is shown in Fig. 3.3a. The source sky model image is an elliptical Gaussian. Here we notice the source extent yields similar JvM factors with the first-null method. With C1 showing 94.1% total source flux in-aperture, the first-null method remains robust for compact sources. The results from source extent integration method is consistent with the first-null method at 89.9% total source flux estimates. We expect the compact source geometry, an elliptical in this case, to contribute little to the overall total source flux estimates. This is because the radial averaging process at small radii effectively approximate the elliptical geometry as circular. In this case, test case C2 and C3 becomes more relevant in terms of total source flux estimates, at 78.7% and 107.6%, respectively. However, we do not see the



(a) In-aperture flux estimates for synthetic compact sources. The sky model image represents an elliptical Gaussian with $\sigma_{\rm maj} = 0.14''$ and $\sigma_{\rm min} = 0.06''$.

(b) In-aperture flux estimates for synthetic point sources. All input fluxes assigned to a single central pixel in the source synthetic sky model image.

Figure 3.3: In-aperture flux estimates for compact Gaussian and point sources. The left pane shows cumulative in-aperture flux with shaded 2σ error envelopes as a function of deprojected radius. The red and green vertical dashed lines mark $r_{\rm null}$ and $r_{\rm extent}$, respectively. The right pane prints the in-aperture flux for the four test cases listed in Table 3.1. The flux normalization is based off of the total flux of the true sky model image.

benefit of choosing either of these two test cases for accurate total source flux estimates over test case C1 and C4. This is because test case C2 underestimates the total source flux while test case C3 overestimates it.

3.2.3 Point Sources

We present the results of point sources in Fig. 3.3b. For test case C1 and C4, we observe similar in-aperture flux estimates at 91.3% and 88.7%, respectively. Since the point sources have all fluxes assigned to a single pixel in the synthetic sky model image, the resulting simulated ALMA observation becomes beam-limited. Effectively, all in-source flux converges immediately along $r_{\rm deproj}$ at beam size of $0.030''$. Consequently, any aperture larger than the beam size sufficiently captures the entire source. The first-null methods is therefore conceptually identical to the source-extent method. We included test case C2 and C3 to investigate the JvM factor evaluation and source flux integration across different apertures. We immediately notice that the integration limit must be set consistently across JvM factor evaluation and source flux estimation.

4. Discussions

In this chapter, we discuss the results presented in Chapter 3. We answer the question of optimal integration limit for JvM factor calculation and subsequent total source flux estimations. We consider potential implications on reported flux estimates in literature and further improvements to this thesis.

4.1 Which JvM Method is More Accurate

For compact and point sources, employing the first-null method for evaluating JvM correction factor and perform total source flux estimates at a first-null radius r_{null} is robust. We produce consistent JvM correction factor when compared with (Öberg et al., 2021). Since the source extent rapidly converges at small deprojected radii for compact and point sources, a first-null radius consistently envelopes the source extent. Consequently, adopting a source extent integration limit at this regime yields near identical results with the first-null integration limit. Both methods succeed in estimating the total in source flux for simulated datasets, with $\sim 91 \pm 3\%$ total source flux estimated in integration aperture. In the case of extended sources, the first-null methods yields 12.6% total source flux while the source-extent method returns 98.1%. Since the first-null radius r_{null} is only dependent on the PSF of the measurements set, it poorly describes the geometry of the actual observation target extent. By setting the integration limit at the source extent, we make sure all flux is considered to constrain the JvM correction factor, and consequently the integration total source flux estimates. Employing different integration limit as apertures for JvM factor and source flux integration is ill-advised. In these cases, the calculation of JvM factor and source flux estimation becomes inconsistent as the dirty beam area is now inconsistent. We recommend adhering to the source extent r_{extent} as a unified method for evaluating JvM factors and the subsequent total source flux estimations.

4.2 Implications on Reported Values in Literature

Assuming the source-extent method is correct, total source fluxes derived from the first-null method would be consistently and systematically underestimated for any cleaned sources. Stated in Section 1.3, this underestimated flux is not connected to the under sampled nature of uv-plane coverage (Helfer et al., 2003). That is, in the case of compact and point sources, we expect a loss of $9 \pm 3\%$ total source flux during estimation.

For extended sources, as much as 87.4% of total source flux is lost when adopting the first-null method, with 1.9% when using the source-extent method. In effect, this translates to a factor of $\sim$ 3 for validation HD 163296 MAPS dataset and a factor of $\sim$ 8 for synthetic, extended dataset. Any observations that combine extended and compact baselines are expected to suffer from this discrepancy. Antenna configurations therefore contain systematic error contribution to the total source flux estimates. In Section 1.3, we state the difference in the missing flux problem. We emphasize that the total source flux estimates cannot physically recover flux that is never observed by the radio interferometer due to incomplete uv-plane coverage.

4.3 Improvements

In Section 2.4, we have showcased the synthetic sky model images that are represented by smooth Gaussian geometries. Real observation targets such as HD 163296 contains substructures that may alter the deprojected radial profile of the fluxes without affecting the total source flux. However, since the source-extent method employs an aperture that captures the source extent, we expect little to no effect of source morphology on the resulting JvM factor calculation and total source flux estimation. However, this speculation require systematic tests with synthetic sky models with substructures.

We have generated the synthetic sky model images without noise. This is never the case for real observations, where noise sources from instrumentation and environment always contribute to the measurement sets. The extent for the validation dataset is therefore estimated at $\sim 4.9''$, consistent with `tclean` mask geometry. Since `tclean` automatically draws mask boundary by specific signal-to-noise ratio, noisier measurement sets may therefore overestimate or underestimate the true source extent. This requires a more robust implementation of source extent estimation. These results must be tested with synthetic datasets that populate with artificial noise elements.

The current error estimates σ is drawn from per-channel rms of the residual beyond source mask. Numerically, these values are small, we therefore visualize in plots 2σ error estimates. That is, the contributing factor to total source flux estimates and JvM factor calculation is not the error estimates, but rather the integration limits themselves.

5. Conclusions

In this thesis, we investigated the choice of integration limits as apertures for JvM factor calculation and total source flux estimations. We tested both the first-null method and the source-extent method on real HD 163296 MAPS data for validation. We then performed the study on simulated ALMA observations on synthetic point, compact, and extended sources. The results are compared with Öberg et al. (2021) following methods introduced in Czekala et al. (2021). We aim to understand which integration limit produce more accurate JvM factor and total source flux estimate.

We find the source-extent method accurately estimates total source flux. This method employs an integration limit that captures the source extent for both the JvM factor calculation and total source flux estimations. With synthetic extended Gaussian source of FWHM at 3.00″, the first-null method produce 12.6% total source flux estimates, while the source-extent method yields 98.1%. For point source and compact source, the two methods converge, returning $91 \pm 3\%$ total source flux estimates.

We recommend users of ALMA data reduction employ the JvM factor calculation at source extent or source aperture instead of the first-null. The source-extent method is recommended for both compact and extended sources. We further encourage adoption of detailed cleaning parameters report for reproducibility. Further work should be focused on testing synthetic observations with noise and complex source substructures to confirm the conclusions of this thesis.

Bibliography

Andrews, S. M., Huang, J., Pérez, L. M., et al. 2018, ApJL , 869, L41, doi: 10.3847/2041-8213/aaf741

Braun, R., & Walterbos, R. A. M. 1985, A&A , 143, 307

Briggs, D. S., Schwab, F. R., & Sramek, R. A. 1999, in Astronomical Society of the Pacific Conference Series, Vol. 180, Synthesis Imaging in Radio Astronomy II, ed. G. B. Taylor, C. L. Carilli, & R. A. Perley, 127

Condon, J. J. 1997, PASP , 109, 166, doi: 10.1086/133871

Cornwell, T., Braun, R., & Briggs, D. S. 1999, in Astronomical Society of the Pacific Conference Series, Vol. 180, Synthesis Imaging in Radio Astronomy II, ed. G. B. Taylor, C. L. Carilli, & R. A. Perley, 151

Cornwell, T. J. 2008, IEEE Journal of Selected Topics in Signal Processing, 2, 793, doi: 10.1109/JSTSP.2008.2006388

Czekala, I., Loomis, R. A., Teague, R., et al. 2021, ApJS , 257, 2, doi: 10.3847/1538-4365/ac1430

Helfer, T. T., Thornley, M. D., Regan, M. W., et al. 2003, The Astrophysical Journal Supplement Series, 145, 259, doi: 10.1086/346076

Högbom, J. A. 1974, A&AS , 15, 417

Hunter, T. R., Petry, D., Barkats, D., Corder, S., & Indebetouw, R. 2023, analysisUtils, 2.90, Zenodo, doi: 10.5281/zenodo.8408419

Isella, A., Guidi, G., Testi, L., et al. 2016, PhRvL , 117, 251101, doi: 10.1103/PhysRevLett.117.251101

Jorsater, S., & van Moorsel, G. A. 1995, AJ , 110, 2037, doi: 10.1086/117668

Kepley, A. A., Tsutsumi, T., Brogan, C. L., et al. 2020, PASP , 132, 024505, doi: 10.1088/1538-3873/ab5e14

McMullin, J. P., Waters, B., Schiebel, D., Young, W., & Golap, K. 2007, in Astronomical Society of the Pacific Conference Series, Vol. 376, Astronomical Data Analysis Software and Systems XVI, ed. R. A. Shaw, F. Hill, & D. J. Bell, 127

Öberg, K. I., Guzmán, V. V., Walsh, C., et al. 2021, ApJS , 257, 1, doi: 10.3847/1538-4365/ac1432

Rau, U., & Cornwell, T. J. 2011, A&A , 532, A71, doi: 10.1051/0004-6361/201117104

The CASA Team, Bean, B., Bhatnagar, S., et al. 2022, Publications of the Astronomical Society of the Pacific, 134, 114501, doi: 10.1088/1538-3873/ac9642

Thompson, A. R., Moran, J. M., & Swenson, Jr., G. W. 2017, Interferometry and Synthesis in Radio Astronomy, 3rd Edition, doi: 10.1007/978-3-319-44431-4

A. CASA tclean

A.1 Parameter Assignment

The parameter assignment for `tclean` can be categorized into three categories for this thesis: 1) the image geometry, 2) the spectral parameters, and 3) the deconvolution parameters.

A.1.1 Image Geometry

The geometry of the image product output is controlled by the following arguments:

`imsize` The dimension of the image product in pixel units. We adopt `imsize=[1024, 1024]`. Since the image reconstruction pipeline in `tclean` scales with the number of pixels present in the final products, we set the argument to balance the precision of the reconstructed image with CPU time cost, ensuring sufficiently quick image processing.

`cell` The size of the pixels present in the imaging products, usually in units of arcseconds. We set this value to `cell="0.065arcsec"` and `cell="0.030arcsec"`, allowing for cross-examination of image products across resolutions.

A.1.2 Spectral Parameters

There exist many options for spectral imaging via `tclean`. We control the type of operation and its corresponding setup with the following parameters, with some help from `listobs` for data-dependent arguments.

`specmode` The master argument to set the type of spectral definition modes. When dealing with continuum emissions or source measurements with a single channel, we employ the `specmode=mfs` mode, which performs multi-frequency synthesis (Rau & Cornwell, 2011). More commonly, when inspecting spectral profiles of the source dataset, we utilize `specmode=cube` mode, where a spectral cube with multiple channels is generated.

`field` The data field to select for image processing. We set `field="0"`.

`spw` The spectral window to select for image processing. We set `spw="0"`.

`restfreq` The rest frequency when `specmode=cube` is selected. Here we set `restfreq="265886.433900MHz"`.

`nchan` The number of output channels for spectral analysis. This is set with `nchan=20` to allow for sufficient spectrum coverage.

`start` The first channel. We set `start="0.0km/s"`.

`width` The channel width in units of km/s. We set `width="0.5km/s"`.

`outframe` The spectral reference frame for interpreting `start` and `width`. We commonly use Local Standard of Rest, Kinematic (LSRK) with `outframe="LSRK"`, following ALMA data reduction conventions.

A.1.3 Deconvolution Parameters

The deconvolution pipeline in `tclean` is declared by the following arguments.

`usemask` The type of masks for deconvolution. There are many options available for this argument. Namely, we could let `tclean` automatically decide the mask geometry, using the primary beam geometry as a mask, or parsing in some manual mask. We employ the new methods of auto-masking present in the newer version of CASA through `usemask="auto-multithresh"` (Kepley et al., 2020).

`niter` The maximum number of iterations to run `tclean`. We use `niter` as an exit condition. Therefore, `niter` is adaptively set to different numbers.

`threshold` The flux threshold to exit `tclean` runs. We set `threshold="4.0mJy"` as default.

`nsigma` The flux threshold exiting condition based on rms. We use this option interchangeably with `threshold`. We set `nsigma=4.0`.

`weighting` The weighting schemes. We commonly use Briggs weighting with a robust parameter via `weighting="briggs"`.

`robust` The robust parameter used for the Briggs weighting scheme. By controlling the robust parameter, we can map to natural weighting with `robust=-2.0` and map to uniform weighting with `robust=2.0`. A robust parameter in $[-2.0, 2.0]$ can be understood as a blend of uniform and natural weighting. We set `robust=0.5` following the `tclean` convention.

`uvtaper` The *uv*-taper parameter. This tapers the image on *uv*-plane to allow the clean beam to reach a desired geometry. The value relies on target-geometry-dependent entries. In essence, this is a custom weighting scheme to complement the set `weighting` argument on the outer baselines in the *uv*-plane.

`deconvolver` The deconvolution algorithm. For an adapted version of the original `CLEAN` algorithm, we can set `deconvolver="hogbom"`. And to employ multi-scale deconvolution, we employ `deconvolver="multiscale"` (Cornwell, 2008).

`scales` The list of scales in pixel units for the `deconvolver="multiscale"` algorithm. For multi-scale deconvolution, `tclean` fits extended clean components as supplements to the Dirac delta distributions for building clean models. This allows the clean model to be represented by clean components of various sizes, ultimately yielding a more accurate representation of the sky image over the original `CLEAN` implementation. We set `scales=[0, 5, 15, 25]`.

A.2 Data Products

Once `tclean` exits upon reaching predefined exit conditions, we arrive at the following data products, as mentioned in Chapter 2.

`.image` The reconstructed image. This is the end product from adding residual to the convolved clean model. The unit is in Jy/Clean Beam.

`.model` The CLEAN model. This is the collection of CLEAN components, representing the deconvolved sky image, where true sources of signal are assumed to be. The unit is in Jy/px.

`.residual` The residual image or the dirty image. This contains the part of the source image where no CLEAN model was fitted, the noise, and part of the source emissions.

`.psf` The PSF or the dirty beam. This is the per-interferometer setup response to the observation source.

`.pb` The primary beam. This is the primary beam sensitivity profile across the field of view.

`.mask` The mask for deconvolution. This manages the geometry where `tclean` can fit CLEAN components onto.

`.sumwt` The sum of weights. This represents the weighting applied to the image.

B. CASA simobserve

B.1 Skymodel Generation

We generate synthetic skymodels by defining the following parameters: 1) image geometry, 2) headers, and 3) source characterization.

B.1.1 Image Geometry

We control the image geometry of the generated skymodel image via the following parameters:

`cellsize` The per pixel resolution of the image. This follows the convention set in Appendix A. We align this parameter with the `cell` parameter present in `tclean` at `cellsize="0.030arcsec"` for image consistency.

`shape` The resolution of the generated image. We follow similar settings present in Appendix A, at `shape=(1024, 1024)`, for mimicking a wide field of view.

B.1.2 Building Headers

At the time of skymodel image generation, we obtain both the image file for visual inspection and a `.fits` for the `simobserve` pipeline. The necessary headers needed for creating a synthetic skymodel image are listed in the following:

`ra_dec_center` The Right Ascension (RA) and Declination (Dec) center. This defines where the center of the generated image is located in the sky. We match this parameter with `listobs` readouts from HD 163296 MAPS data at `ra_dec_center="17h56m21.278890s -21d57m22.64064s"`.

`restfreq` The rest frequency. This parameter is identical to the one present in Appendix A. We set it at `restfreq=265886.433900e6` Hz.

`bunit` The generated image unit. We assign `bunit="Jy/pixel"`, matching the deconvolved clean model image.

`date` This is the data. We assign arbitrary data in this parameter space, as this entry is likely to be overwritten when `simobserve` is called.

Since the image center is defined with RA and Dec coordinates in the J2000 definition, we translate via World Coordinate System (WCS) to assign the center pixel of the generated image at the defined image center. This is an especially important step, as point sources are commonly represented as a single pixel in our simulated skymodel image, and when the image coordinates are misaligned with that of the assigned image center at `ra_dec_center`, we would be generating an off-center image, resulting in ineffective visualization.

B.1.3 Source Characterization

We now characterize the observational target with the following parameters:

`source_type` The source type. This can either be a point source or a Gaussian source. For compact sources, we utilize `source_type="point"`. And when non-point sources or more extended sources are present, we approximate these sources via a Gaussian distribution by setting `source_type="gauss"`.

`flux_density` The total flux. We manually define a total flux present in the generated image and smear it across the area where the source is present. We commonly set this to unity with `flux_density=1.0`.

`major_arcsec` The semi-major axis length. This defines the semi-major axis of the synthetic Gaussian observation source in arcsecond units.

`minor_arcsec` The semi-minor axis length. This defines the semi-minor axis of the synthetic Gaussian observation source in arcsecond units.

`pa_deg` The position angle. This defines the position angle of the synthetic observational target, be it a point source or a Gaussian source, in degrees.

B.2 Antenna Configurations

We now work to obtain the second ingredient necessary for `simobserve` pipeline, the antenna configurations. This is achieved through the use of `analysisUtils`, we evoke `analysisUtils.buildConfigurationFile` for directly exporting the antenna configurations from existing measurements `.ms` files.

B.2.1 Multiple Observations

It is common for ALMA observation to consist of many observations. These observations may take place across different times of the year, with different antenna configurations for individual observations. We therefore utilize the `table` utility from CASA to directly examine the associated observation ID present in the source measurements file. Moving on, this allows us to split the observations into independent measurements via the `split` task in CASA.

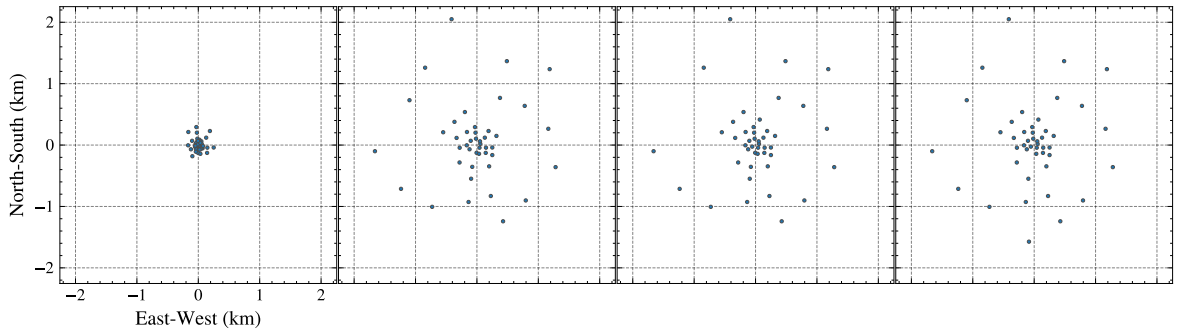


Figure B.1: Antenna configuration for HD 163296 MAPS data. The source measurements contain four distinct observation IDs with observation dates and antennas used. From left to right, these figures indicate the antenna configurations across HD 163296 MAPS data observation ID 0 through 3. A compact antenna configuration is shown in the leftmost figure, while the other three configurations are largely similar with respect to each other.

B.2.2 Filtering Antennas

When working directly with `buildConfigurationFile`, the exported antenna configuration consists of all the antennas used across observations. This issue is not mitigated at the time of `split`, but rather, we must use `listobs` to export the actual antenna names and their associated stations at ALMA, such that a second iteration filtering of all the present antennas could take place. During this process, we build the list of antennas used by some observations with its associated `listobs` results. The resulting antenna configurations is shown in Fig. B.1:

B.2.3 Parameter Assignment

The parameters present in `analysisUtils.buildConfigurationFile` and the antenna configuration filtering pipeline are presented below, and all parameters not mentioned in the description are considered to be set at their default values:

`vis` — The visibility. This is the measurements data set. We set this to target the HD 163296 MAPS data.

`coordsys` — The coordinate system. For ALMA data, default to `coordsys="LOC"`. This utilizes a Local Tangent Plane Coordinates (LOC) for defining the antenna locations.

`observatory` — The name of the observatory. We default to `observatory="ALMA"` for this thesis.

B.3 Simulation Setup

Now that we have obtained the ingredients for simulations, we feed them into the simulation pipeline via `simobserve`. We describe the parameters associated with this step in the following categories: 1) the necessary ingredients, 2) image geometry, 3) spectral parameters, 4) observational parameters, and 5) pointing parameters.

B.3.1 Necessary Ingredients

We control the skymodel image and antenna configuration parsing via the following parameters:

`skymodel` The list of skymodels.

`antennalist` The list of antenna configurations.

B.3.2 Image Geometry

The following parameters define the geometry of the simulated observation, along with its resulting resolution:

`indirection` The center of observation. This controls the direction of observation. We set this to align with the J2000 RA and Dec declared in synthetic skymodel image generation at `indirection="J2000 17h56m21.278890s -21d57m22.64064s"`.

`incell` The cell size of the input model. This controls the resolution of the resulting image, and we set this in alignment with other cell size parameters at `incell="0.030arcsec"`.

B.3.3 Spectral Parameters

We define the spectral characteristics of the simulated data products via:

`incenter` The central frequency. This is read from `listobs` readout of HD 163296 MAPS data and set at `incenter="265881.3786MHz"`.

`inwidth` The bandwidth. We define `inwidth` to set the simulation bandwidth. A generous `inwidth="8GHz"` is set to allow for sufficient coverage around the central frequency.

B.3.4 Observational Parameters

The observational parameters are set to control the simulated ALMA observation itself:

`totaltime` The total on-source integration time. This defines how long in total a simulated ALMA observation spends on observing the given skymodel image. We set `totaltime="3600s"`.

`integration` The integration time per visibility. With `integration="10s"`, the simulated observation generates visibilities at 10s intervals.

`compwidth` The component spectral width. This is similar to the `width` parameter in Appendix A. We define `compwidth="8GHz"`, aligning with `inwidth` for single-channel data products.

`obsmode` The mode for observation. We set `obsmode="int"` for observations through ALMA interferometers.

Since we are only interested in the total amount of source skymodel image flux recovered via JvM correction, we produce a single channel in the resulting measurements data set, mitigating the need for per-channel analysis.

B.3.5 Pointing Parameters

And lastly, we define the pointing parameters. These parameters control how the simulated observation deals with extended sources. Since the generated skymodel images are large in their `shape`, we effectively create a large field of view with its `cellsize` that encompasses a 30 arcsecond by 30 arcsecond area.

`setpointings` The option to set pointings. The default value of true becomes problematic for extended skymodel images with a hexagonal mosaic pattern in the pointings. We therefore set `setpointings=False` to allow for manual `ptgfile` parsing.

`ptgfile` The `.ptg` file path. We generate a manual `.ptg` file in `.txt` format to represent the `indirection` settings. This allow `simobserve` to treat the extended synthetic skymodel image as if it were compact. This is a viable alternative, as the true object of observation in the synthetic skymodel image is often much smaller in scale.

B.4 Post Simulation

As `simobserve` draws to a close, we are left with the per-observation `.ms` files. These measurements are then concatenated via `concat` with a set `dirtol="1arcsec"` to allow for slight offsets in the direction of simulated measurement components. Next, we convert these simulated measurements from `.ms` files into actionable data products via `tclean`, as mentioned in Appendix A.

C. Code

C.1 Workflow

The script used for this thesis is designed in the following manner: 1) atomic helpers, 2) single-task utilities, and 3) parallelization methods. We then execute the script within a live CASA session via terminal commands.

Helpers The helper functions. These helper functions make up the foundation of the script. We designed these functions to be self-explanatory. Tasks such as searching and compiling a list of measurements by keywords, extracting `.fits` file headers, and performing deprojection fall within this category.

Utilities The utility functions. We then assemble the helpers coherently in utility functions for more complex executions. These utilities could perform tasks such as calculating a radius-dependent JvM factor, performing image correction with the calculated JvM factor, exporting the results in a constructed `.fits` file, and producing associated plots. We commonly build plotters and data cube handlers in this phase.

Parallelization The parallelization step. We assign the utilities as workhorses in the script for parallelization. This is handled by a uniform wrapper `run_parallel` that accepts zero-argument functions in lists. We then distribute the tasks on CPU cores via `functools.partial`. Effectively, we execute multiple utility functions in parallel, thus allowing for rapid, automatic diagnostics to be run.

Mode The mode selection. We group utilities and their parallelization into categories. This is then assigned to modes at the end of the script, as CASA terminal favors scripted `Python` syntax over functional designs. Therefore, we simply list the functions we desire for specific modes, such as performing `tclean` with multiple configurations, executing automatic JvM factor diagnostics, and generating multiple plots simultaneously across multiple data products.

C.2 Dependency

The script is dependent on the following packages for identical environment setup.

Code Modules

This thesis relies on CASA as its core module.

`casa` The CASA environment. This is the main element for this thesis script file. We call for packages such as `casatasks` and `casatools` with CASA. During this thesis, we work with CASA `v6.4.1`.

The key tasks involved with this thesis in `casatasks` and `casatools` contains the following.

`tclean` The main algorithm for the cleaning process. This is explained in detail in Appendix A.

`exportfits` The main handler for `.fits` files. We save the processed data cubes into `.fits` files with this method.

`simobserve` The primary method for simulations. This is elaborated in Appendix B.

`split` The method for splitting a single measurement set into multiple ones. This

`concat` The method for concatenating multiple measurements into a singular one.

`listobs` The utility function for accessing the summary metadata of measurements.

`imhead` The utility for directly accessing and manipulating image headers.

`immoments` The function for generating image moment maps.

Scientific Computing

Key packages used for handling and generating data products are listed here.

`astropy` The `astropy` package. We rely on `astropy` for WCS handling `.fits` file manipulation, convolution via Fast Fourier Transform, and kernel building.

`numpy` The `numpy` package. This allows us to access advanced array manipulation techniques for data cube processing.

`scipy` The `scipy` package. This enables accurate and efficient integration in beam area calculations and minimization methods for `uvtaper` parameter finding.

`analysisUtils` The custom analysis utility suite. We employ this to look up and build antenna configurations, as stated in Appendix B.

`matplotlib` The core plotting utility suite. We employ `pyplot` for plot generation.

Parallelism

The packages used for parallelization are listed here.

`functools` The main methods for parsing in zero-argument functions for simultaneous execution.

`multiprocessing`
The method for spawning processes instead of threads to allow CASA to run multiple tasks in parallel.

`concurrent.futures`
The method for asynchronously performing multiple tasks. We make high-level parallelization through this method, over conventional `multiprocessing` usage.

Miscellaneous

The key additional packages not mentioned above are listed here.

`logging` We employ this method to spawn and create terminal logger output for inspection.

`re` The regular expression package is used to handle string manipulations involved with automatic path discovery and assignment.

`pathlib.Path` The package for cross-platform support in directory handling.

`shutil` We employ this package for performing automatic shell commands for temporary file handling. This must be set with caution to avoid accidental removal of key data files.

D. Beam Diagnostics

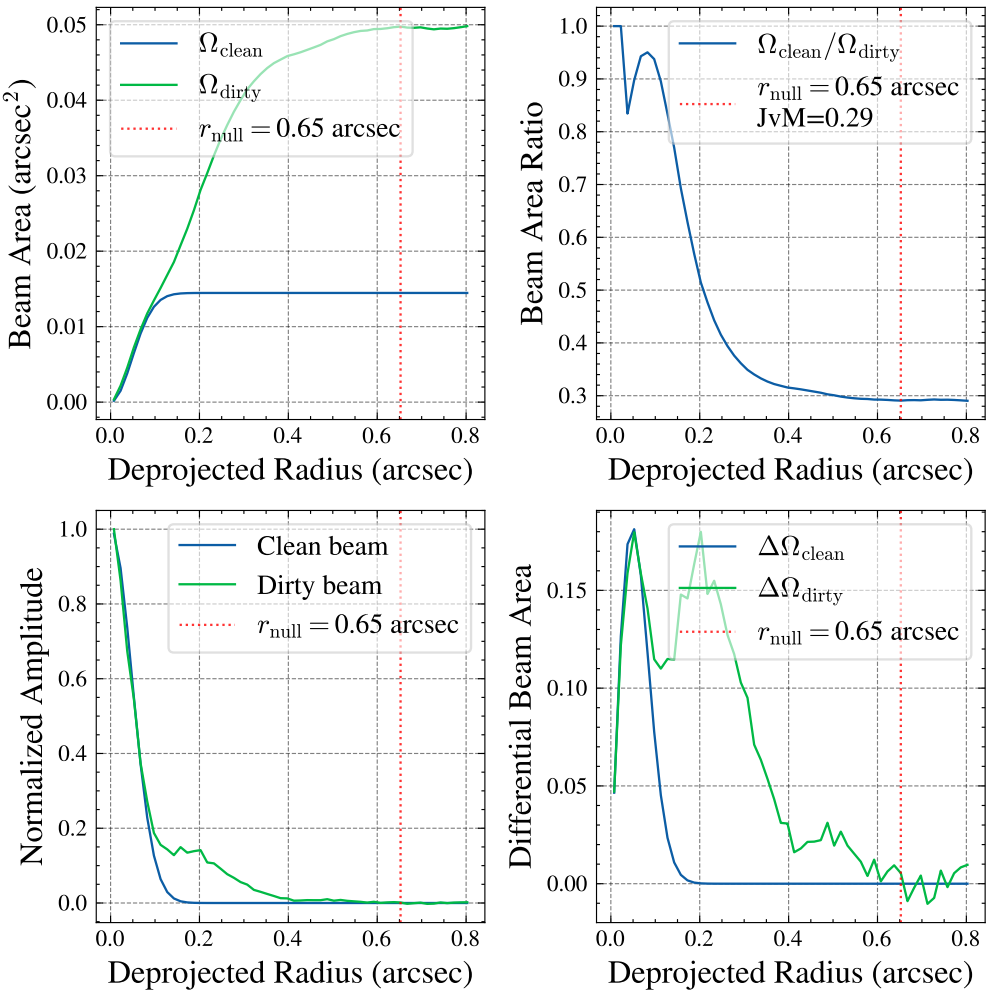


Figure D.1: Beam diagnostics. The diagnostic plot for HD 163296 MAPS data when standard `tclean` parameters are used to clean to `nsigma=4`. We produce: 1) the cumulative beam areas of the clean and dirty beams, 2) the beam area ratios and the associated first null for JvM factor finding, 3) the normalized beam amplitudes on the deprojected radial, where a "shelf" is present in the dirty beam around 0.1″ to 0.4″, and 4) the differential profiles of the clean and dirty beams for beam area calculation. We reproduces the diagnostic plot present in Czekala et al. (2021) and visualize the beam geometry and the extended shelf.