

# VLA CASA Imaging Tutorial - MWAL Module 5, A.Y. 2025/2026

---

Prof. Myriam Gitti, Tutor: Dr. Francesco Ubertosi

Adapted from the [CASA Imaging Tutorial](#)

The data used in this tutorial can be found on [Virtuale](#) under the name **MS0735-VLA\_calibrated.zip**

## Introduction

---

This tutorial provides guidance on imaging procedures in CASA. The tutorial covers calibrated data inspection, basic continuum cleaning and the influence of image weights. In this tutorial, we display all images with the VIEWER tool in CASA. Note that another software, named CARTA (Cube Analysis and Rendering Tool for Astronomy) is now the recommended tool to display images in the most recent versions of CASA (being the only available tool for macOS versions > 14). CARTA is developed for the VLA, ALMA, and other radio telescopes. We will NOT cover CARTA in this tutorial (or in the lab course), but if you are interested you can find more details about how to use and display images in CARTA [at this link](#).

We will be utilizing observations taken with the Karl G. Jansky Very Large Array of the central radio galaxy in the galaxy cluster [MS 0735.6+7421](#). The data were taken in 2018 in the C-configuration of the VLA, using the L band receiver (central frequency: 1.5 GHz). **For practical reasons, we have reduced the size of the dataset, by averaging the (calibrated) data in time and selecting 2 spectral windows only. The effective central frequency of the data for this tutorial is 1.2 GHz.**

We will skip the calibration process in this guide, as examples of calibration can be found in several other guides, including the [VLA Continuum Tutorial for 3C391](#).

A copy of the calibrated data (<200 MB) can be found on [Virtuale](#).

Your first step will be to unzip the file in your working directory. From the command line (before you start CASA):

```
unzip MS0735-VLA_calibrated.zip
```

Then start CASA as usual via the `casa` command, which will bring up the CASA prompt and launches the logger (**do not close the logger window! if you do, exit CASA and relaunch it to re-open it**). If you want to have a look at the data structure, you can do so with the task [listobs](#).

## The CLEAN Algorithm

---

The CLEAN algorithm, developed by [J. Högbom \(1974\)](#) enabled the synthesis of complex objects, even if they have relatively poor Fourier uv-plane coverage. Poor coverage occurs with partial Earth rotation synthesis, or with arrays composed of few antennas. The "dirty" image is formed by a simple Fourier inversion of the sampled visibility data, with each point on the sky being represented by a suitably scaled and centered Point Spread Function (PSF), or dirty beam, which itself is the Fourier inversion of the visibility (u,v) coverage.

The convolution with the dirty beam creates artifacts in the image and limits the dynamic range. The CLEAN algorithm attempts to remove the dirty beam pattern from the image via deconvolution. This implies that it interpolates from the measured (u,v) points across gaps in the (u,v) coverage. In short, CLEAN provides solutions to the convolution equation by representing radio sources by a number of point sources in an empty field. The brightest points are identified in the dirty image, and their associated PSFs are subtracted. The process is then repeated for the next brighter points. Variants of CLEAN, such as multi-scale CLEAN, take into account extended kernels which may be better suited for extended objects.

For single pointings, CASA uses the Hogbom cleaning algorithm by default in the task `tclean` (`deconvolver='hogbom'`), which breaks the process into major and minor cycles (see Figure 1). To start with, the visibilities are gridded, weighted, and Fourier transformed by computing the so-called fast Fourier transform (FFT) to create a (dirty) image. The minor cycles then operate in the image domain to find the *clean components* (i.e., the brightest point sources) that are added to the clean model: repeatedly performing a search to find the next bright point, then subtracting its PSF from the image. The model image is Fourier transformed back to the visibility domain, degridded, and subtracted from the visibilities. This creates a new residual that is then gridded, weighted, and FFT'ed again to the image domain for the next iteration. The gridding, FFT, degridding, and subtraction processes form a major cycle. This iterative process is continued until a stopping criterion is reached, such as a maximum number of clean components, or a flux threshold in the residual image.

In CASA `tclean`, two versions of the PSF can be used: setting "`deconvolver='hogbom'`" uses the full-sized PSF for subtraction. This is a thorough but slow method. All other options use a smaller beam patch, which increases the speed. The patch size and length of the minor cycle are internally chosen such that clean converges well without giving up the speed improvement. In a final step, `tclean` derives a Gaussian fit to the inner part of the dirty beam, which defines the clean beam. The clean model is then convolved with the clean beam and added to the last residual image to create the final image.

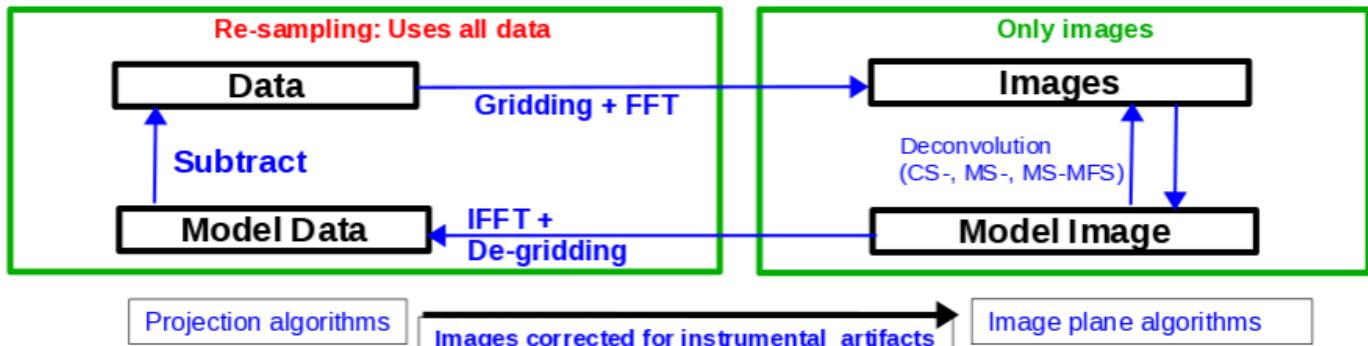


Figure 1: The CLEAN major and minor cycles, indicating the steps undertaken during gridding, projection algorithms, and creation of images.

For more details on imaging and deconvolution, we refer to the Astronomical Society of the Pacific Conference Series book entitled [Synthesis Imaging in Radio Astronomy II](#).

The chapter on [Deconvolution](#) may prove helpful.

In addition, imaging presentations are available on the [Synthesis Imaging Workshop](#) and [VLA Data Reduction Workshop](#) webpages.

The [CASA Documentation](#) chapter on [Synthesis Imaging](#) provides a wealth of information on the CASA implementation of `tclean` and related tasks.

Finally, we refer users to the [VLA Observational Status Summary](#) and the [Guide to Observing with the VLA](#) for information on the VLA capabilities and observing strategies.

## Weights and Tapering

'Weighting' amounts to giving more or less weight to certain visibilities in your data set, based on their location in the uv-plane. Emphasizing long-baseline visibilities improves the resolution of your image, whereas emphasizing shorter baselines improves surface brightness sensitivity. There are three main weighting schemes that are used in interferometry:

- **Natural** weighting: uv cells are weighted based on their rms. Data visibility weights are gridded onto a uv-cell and summed. More visibilities in a cell will thus increase the cell's weight, which will usually emphasize the shorter baselines. Natural weighting, therefore, results in a better surface brightness sensitivity, but also a larger PSF and therefore degraded resolution.
- **Uniform** weighting: The weights are first gridded as in natural weighting but then each cell is corrected such that the weights are independent of the number of visibilities inside. The 'uniform' weighting of the baselines is a better representation of the uv-coverage and sidelobes are more suppressed. Compared to natural weighting, uniform weighting usually emphasizes longer baselines. Consequently, the PSF is smaller, resulting in a better spatial resolution of the image. At the same time, however, the surface brightness sensitivity is reduced compared to natural weighting.
- **Briggs** weighting: This scheme provides a compromise between natural and uniform weighting. The transition can be controlled with the "robust" parameter where "robust=-2" is close to uniform and "robust=2" is close to natural weighting. Briggs weighting, therefore, offers a compromise between spatial resolution and surface brightness sensitivity. Typically, "robust" values near zero are used.

Details on the weighting schemes are given in [Daniel Brigg's dissertation \(Chapter 3\)](#). For a visual comparison between these three weighting schemes, please see the section on "CLEAN with Weights" in this guide.

	Robust/Uniform	Natural	Taper
resolution	higher	medium	lower
sidelobes	lower	higher	depends
point source sensitivity	lower	maximum	lower
extended source sensitivity	lower	medium	higher

Table 1: Table summarizing the effects of using weights and tapering.

**Tapering:** In conjunction with the above weighting schemes, one can specify the "uv taper" parameter within `tclean`, which will control the radial weighting of visibilities in the uv-plane. Figure 2 illustrates the uv-coverage during the observing session used in this guide. The taper in `tclean` is an elliptical Gaussian function which effectively removes long baselines and degrades the resolution. For extended structures, this may be desirable when the long baselines contribute a large fraction of the noise.

Tapering can therefore increase the surface brightness sensitivity of the data but will decrease the point source sensitivity. Too aggressive tapering, however, may also take its toll on the surface brightness sensitivity. Table 1 summarizes the main effects of the different weighting schemes. We refer to the [Synthesis Imaging](#) section of the CASA Documentation for the details of the weighting implementation in CASA's `tclean`.

## Primary and Synthesized Beam

---

The primary beam of a single antenna defines the sensitivity across the field of view. For the VLA antennas, the main part of the primary beam can be approximated by a Gaussian with an FWHM  $\sim \frac{\lambda}{D}$  (where D = 25m is the diameter of the VLA antennas), corresponding to  $\frac{42}{v_{\text{GHz}}}$  arcminutes (for frequencies in the range 1 - 50 GHz). But note that there are sidelobes beyond the Gaussian kernel that are sensitive to bright sources (see more [here](#)). Taking our observed frequency to be the middle of the L-band, 1.5 GHz, our primary beam (FWHM) will be about 28 arcmin in diameter.

Note: New beam measurements were made recently and are described in [EVLA memo 195](#). These newer beam corrections are the default in CASA 5.5.0.

If your science goal is to image a source or field of view that is significantly larger than the FWHM of the VLA primary beam, then creating a mosaic from a number of telescope pointings is usually the preferred method. For a tutorial on mosaicking, see the [3C391 tutorial](#). In this guide, we discuss methods for imaging single-pointing data. As the data for this tutorial were taken in the C-configuration, we can check the [Observational Status Summary](#)'s section on VLA resolution to find that **the synthesized beam will be approximately 14 arcsec at L band** (central frequency 1.5 GHz). Variations in flagging, weighting scheme, and effective frequency may result in deviations from this value. **We will compute the expected beam FWHM for this specific dataset from the AMP vs UVWAVE plot in plotsms** (see Figure 2). The synthesized beam is effectively the angular resolution of the image.

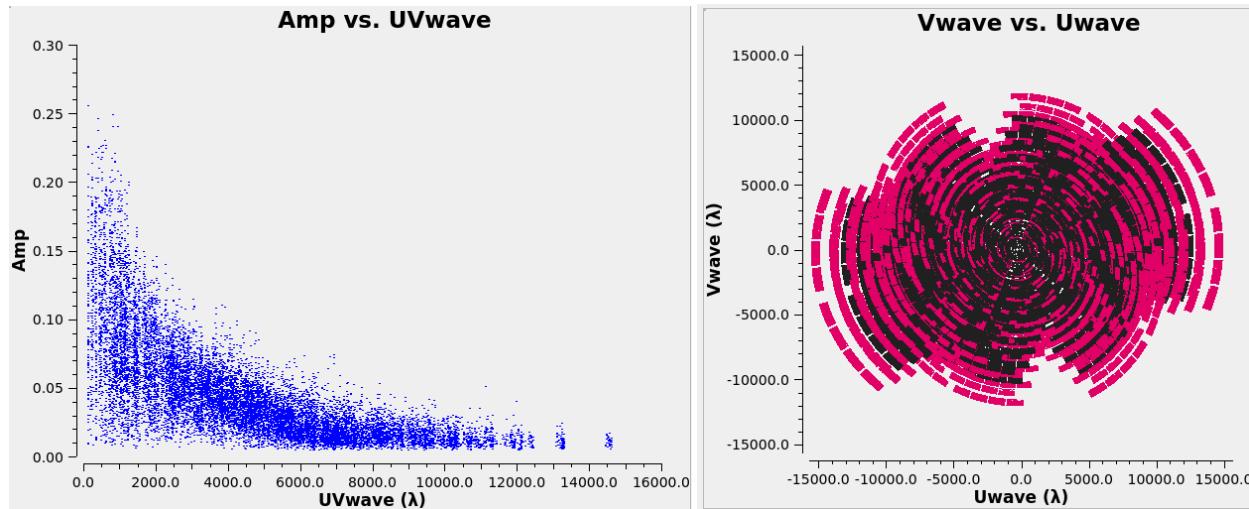


Figure 2: *u,v* coverage for the VLA observation of the galaxy cluster MS0735.6+7421. Left: amplitude vs *uvwave* plot obtained in `plotsms`, with 16 channels averaging, time averaging of 1000 seconds, and scan averaging. Right: *v*-wave vs *u*-wave (*uv*-plane) plot obtained in `plotsms`; colors reflect the two spectral windows.

## Cell (Pixel) Size and Image Size

---

For the most effective cleaning, we recommend using a pixel size such that there are at least 3-5 pixels across the synthesized beam, which satisfies the Nyquist sampling theorem for the minimum sampling rate. Based on the assumed synthesized beam size of 14 arcsec (as estimated from the maximum *uvwave* value in Fig. 2), we will use a cell (pixel) size of 2 arcsec.

In the `tclean` task, the image size is defined by the number of pixels in the RA and DEC directions. The execution time of `tclean` depends on the image size. Large images generally take more computing time. There are some particular image sizes (by a number of pixels) that are computationally inadvisable. For inputs corresponding to such image sizes, the logger will show a recommendation for an appropriate larger, but computationally faster, image size. As a general guideline, we recommend image sizes  $5 \cdot 2^n \cdot 3^m$  ( $n=1,2,\dots$ ,  $m=0,1,2,\dots$ ) pixel, e.g. 160, 1280 pixels, etc. for improved processing speeds.

We will set the image size to 540 pixels on each side, for efficient processing speed. Therefore, our calls to the `tclean` task within this guide will create images that are  $540 \times 540$  pixels = 1080 arcsec = 18 arcminutes on a side. This angular size corresponds to about 60% the size of the primary beam (which has a FWHM of  $\frac{42}{v_{\text{GHz}}}$  arcminutes = 28 arcminutes). This is enough to appreciate the effect of contamination from other sources in the field of our target, as well as to optimize the computing time.

## Clean Output Images

---

As a result of the CLEAN algorithm, `tclean` will create a number of output images. For a parameter setting of '`imagename='test'`', these image names will be:

`test.residual`: the residual after subtracting the clean model (unit: Jy/beam, where beam refers to the dirty beam)  
`test.model`: the clean model, not convolved (unit: Jy/pixel)

`test.psf`: the point-spread function (dirty beam)

`test.pb`: the normalized sensitivity map, which corresponds to the primary beam in the case of a single-pointing image

`test.image`: the residual + the model convolved with the clean beam; this is the final image (unit: Jy/beam, where beam refers to the clean beam). Note that if no cleaning is performed, this corresponds to the *dirty* image.

`test.sumwt`: a single pixel image containing sum-of-weights (for natural weighting, the sensitivity = 1/sqrt(sumwt)). Additional images will be created for specific algorithms like multi-term frequency synthesis or mosaicking.

! **Important** If an image file is present in the working directory and the same name is provided in "imagename", `tclean` will use that image (in particular the residual and model image) as a starting point for further cleaning. If you want a fresh run of `tclean`, first remove all images of that name using 'rmtables()':

```
# In CASA
rmtables('test.*')
```

This method is preferable over 'rm -rf' as it also clears the cache. *Note that interrupting tclean by Ctrl+C may corrupt your visibilities - you may be better off choosing to let tclean finish. We are currently implementing a command that will nicely exit to prevent this from happening, but for the moment try to avoid Ctrl+C..*

**Important:** By default, `tclean` sets "savemodel" to a value of 'none', meaning no model image is saved. Be sure to set this parameter to 'modelcolumn' for any model you wish to save. This is especially important for self-calibration. **We will not self-calibrate our data, so you can leave savemodel = 'none'.**

## Dirty Image

First, we will create a dirty image (Figure 3a) to see the improvements as we step through several cleaning algorithms and parameters. The dirty image is the true image on the sky, convolved with the dirty beam, also known as the Point Spread Function (PSF). We create a dirty image by running `tclean` with niter=0, which will run the task without performing any CLEAN iterations.

```
# In CASA
tclean(vis='MS0735-VLA_calibrated.ms', imagename='MS0735_test', cell='2arcsec', imsize=540,
niter=0, interactive=True, savemodel='none', stokes = 'I')
```

- `imagename='MS0735_test'`: the root filename used for the various `tclean` outputs.
- `imsize=540`: the image size in a number of pixels. A single value will result in a square image.
- `cell='2arcsec'`: the size of one pixel; again, entering a single value will result in a square pixel size. This should be set to **at least** 1/3 - 1/5 of the expected synthesized beam. **Important:** the angular resolution is given by the largest baseline; look at `plotms` AMP vs UVWAVE to determine your maximum uv distance and translate it into an angular resolution. Remember that  $\theta \sim \frac{1}{uv_{max}[\lambda]}$  radians, and that 1 rad = 206265 arcsec.
- `niter=0`: this controls the number of iterations done in the minor cycle.
- `interactive=True`: We will interactively clean our image.
- `savemodel='none'`: controls writing the model visibilities to the model data column. For self-calibration (not covered in this tutorial), we currently recommend setting `savemodel='modelcolumn'`. The default value is "none": so, this must be changed for the model to be saved.
- `stokes='I'`: since we have not done any polarization calibration, we only create a total-intensity image. For using `tclean` while including various Stoke's parameters, please see the [3C75 CASA guide](#).

Now, to visualize the image, we will use the CASA task `imview`:

```
# In CASA
imview
```

Now select the '.image' and '.psf' files to be displayed by clicking on the 'folder' icon, and open the files as *raster* images. Note that you may have to play with the image color map and brightness/contrast in (using the "wrench" symbol in the viewer panel) to get a better view of the image details. **For more information on how to visualize images in CASA task imview, please see the brief guide for plotms and viewer on Virtuale.**

Note that the clean beam (the Gaussian fit of the main lobe of the dirty beam) is only defined after some clean iterations. The dirty image (Figure 3, left) has therefore no beam size specified in the header, and the PSF image (Figure 3, right) is the representation of the response of the array to a point source. Even though it is empty because we set `niter=0`, `tclean` will still produce a model file. Thus we could progress into actual cleaning by simply restarting `tclean` with the same image root name (and "`niter">>0`"). Starting in CASA 6.6, `tclean` will return a dictionary with information about the image, such as the peak residual, for niter=0 calls to `tclean`. With previous versions, such as 6.5.4, only `tclean` calls using `niter >0` will return this dictionary.

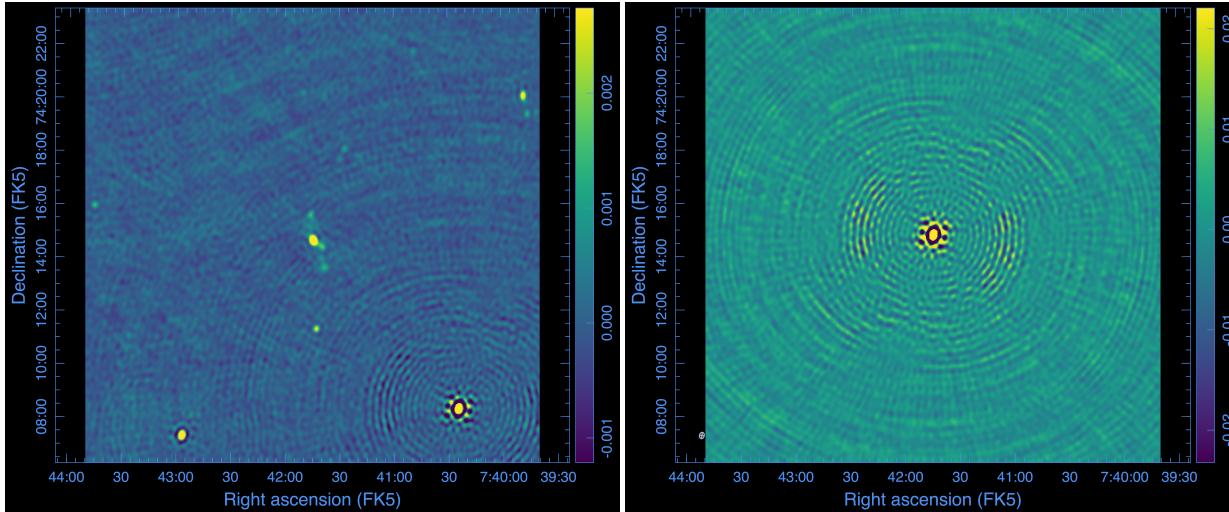


Figure 3. Left: a dirty image of the MS 0735.6+7421 field in viridis scale, with apparent sidelobes. Right: the Point Spread Function (PSF) in viridis scale. This corresponds to the **Dirty Beam**.

## Regular CLEAN & RMS Noise

Now, we will create a regular clean image to see how deconvolution improves the image quality. In our call of `tclean` we will use a fixed number of iterations of "niter=10000" (default is 0). We will proceed with *interactive* cleaning, that is, we will help the software in finding *clean components* to deconvolve. Specifically, remember that the CLEAN algorithm assumes that the image consists of a number of point sources. It will iteratively find the highest value in the image and subtract a small gain of this point source convolved with the dirty beam of the observation, until the highest value is smaller than some threshold or the maximum number of iterations is reached. We will interact with the software by selecting regions where to find these *point source-like components* to deconvolve. This is achieved with the following call of `tclean`:

```
# In CASA. Create a clean image with interactive cleaning.
tclean(vis='MS0735-VLA_calibrated.ms', imagename='MS0735', cell='2arcsec', imsize=540,
niter=10000, threshold='0.03mJy', interactive=True, savemodel='none', stokes='I')
```

- *imagename='MS0735'*: the root filename used for the various `tclean` outputs.
- *niter=10000*: this controls the number of iterations done in the minor cycle.
- *interactive=True*: this will allow us to select the regions where to find the components to be cleaned.
- *threshold = '0.03mJy'*: To avoid cleaning too deeply, you can set a *threshold* parameter that will stop minor cycle clean iterations once a peak residual value is reached. This *threshold* is usually set to three times the expected rms noise levels of the images. A method to determine the approximate rms of an image is to use the [VLA Exposure Calculator](#) and to enter the observing conditions. **We will follow this approach**. In the [VLA Exposure Calculator](#), first select a representative (central) frequency and bandwidth. From the inspection of the data with the task `listobs`, we know that we have two spectral windows of 128 MHz each, centered at frequencies of 1.1 GHz and 1.3 GHz, respectively. We can thus select 1.2 GHz as frequency and 256 MHz of bandwidth. Next, select the array configuration (C), leave the number of antennas and polarization setup as default, and select *Type of image weighting = Robust*. Then leave *number of frequencies*, *session includes pointing, digital samplers, elevation, and average weather* as default. Next, change *calculation type* to *Noise/Tb*, leave *number of sources* = 1, and insert the time spent on the target MS 0735.6+7421. From the call to `listobs`, we know that the source has been observed in 32 scans of 7.5 minutes each, for a total of 4h of on-source-time. Insert this final information in the [VLA Exposure Calculator](#) and note the expected *RMS Noise* (1e-5 Jy/beam). The *threshold* parameter should be set to 3 - 5× the expected *RMS Noise*.

The [viewer](#) will open when it's ready to start an interactive CLEAN.

**How to use the interactive CLEAN window in `imview`:** First, let's change the color map and color scale: this can be done under the Data Display Options by clicking on the wrench icon or in the top menu and then changing the *Scaling power cycles* to some values (e.g., -1 is recommended). Now let's draw regions around components that we believe are real and that we want to deconvolve. Single clicks can be used to adjust the region and small squares will show anchor points that can be dragged to new positions. For the polygon region tool, single clicks will add new segments with anchor points on the vertices (shown as small green squares). A double click will connect the first and last vertices to create a polygon. Double click within the polygon to add the region to the CLEAN mask, which will be shown in white. To use the circular region tool instead, click and drag the bounding box of the circle over the source and adjust the anchor points on the corners to cover the emission. If you happen to mis-click, set the radio button to "Erase" instead of "Add", and the region will be subtracted from the CLEAN mask. Create as many regions as you see fit to describe the number of components in the image (see Figure 4, left, as an example).

**Note:** If an interactive clean is started, and a mask is not created, then `tclean` will stop immediately, because in interactive sessions only emission within a mask is deconvolved. This is the opposite of a call to `tclean` with *interactive=False*, where without a mask the entire field of view will be deconvolved in an unrestricted fashion.

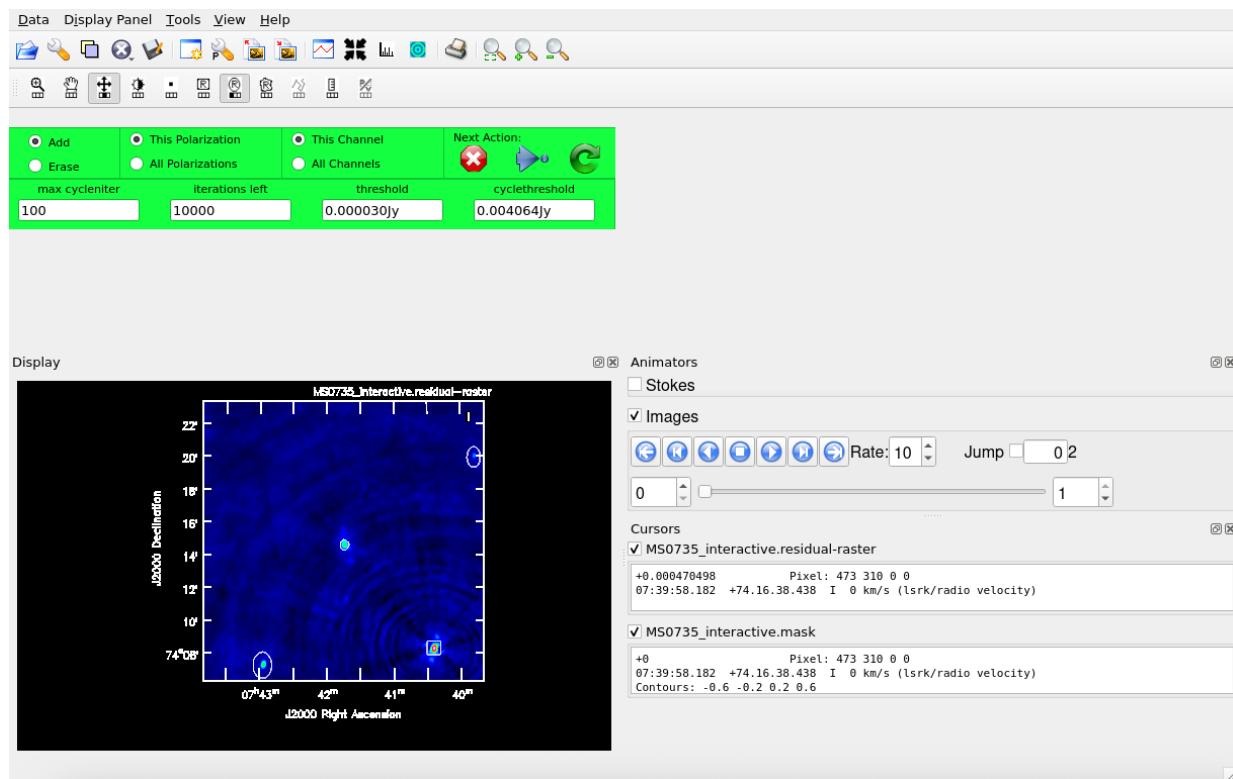


Figure 4. First round of cleaning in the interactive viewer, after adjusting the color scale using the wrench icon and adding white circles and rectangles around significant sources of emission. These regions represent the regions where the CLEAN algorithm will look for components to deconvolve.

To continue with `tclean` use one of the Next action buttons in the green area on the Viewer Display GUI: The red "X" will stop `tclean` where it is; the blue arrow will stop the interactive part of `tclean`, but continue to clean non-interactively until reaching the stopping niter ('iterations x cycles') or threshold, whichever comes first. The green circular arrow will clean until it reaches the iterations parameter on the left side of the green area. When the interactive viewer comes back use the tape deck to recheck that the mask encompasses what you think is real emission. If your mouse has a middle button, then by default it controls the image stretch.

For this example, the threshold has been set to three times the expected RMS via the `threshold` parameter to avoid CLEANing too deeply. With a careful CLEAN mask you can CLEAN too close to the thermal noise limit (remember: the actual observed RMS noise limit may not match the theoretical one calculated for the data, as flagging, weather, etc., can affect what you actually get). As you proceed with cleaning, enlarge existing regions or add new regions to include more emission that becomes significant and that needs cleaning.

Continue cleaning by using the green Next Action arrow until the residual displayed in the viewer looks noise-like (you should not see the 'shade' of the source anymore). To speed things up, you might change the '`max_cycleniter`' parameter in the viewer. When the residuals look like random noise, use the red X button to stop cleaning. The resulting restored image is shown in Figure 5 (right).

We now use the `imview` task of CASA to visualize the image that we have just obtained.

```
# In CASA
imview
```

Looking at Figure 5, many improvements over the dirty image are clearly visible. We could clean more by increasing the number of iterations, which might improve our image by reducing prominent sidelobes significantly. To determine the number of iterations, one needs to keep in mind that `tclean` will fail once it starts cleaning too deeply into the noise. At that point, the cleaned flux and the peak residual flux values will start to oscillate as the number of iterations increase. This effect can be monitored on the CASA logger output.

We will now utilize the `MS0735.image` (Figure 5, right) image to estimate the rms noise (your sigma value). With the image open within the `imview`, click on the 'Rectangle' button in the region bar at the top of the GUI and draw a square on the image at a position with little source or sidelobe contamination. After drawing the square in an empty region, click the Statistic panel from the region panel at the top of the GUI. This will open another window of the name "Statistic: Region 1" (Figure 5) which holds information about the selected region, including the pixel statistics. Take notice of the RMS values. User can draw multiple regions (of any shape which draw over enough amount of pixels) like this at different empty regions, and take the average of the individual RMS values to obtain a better RMS estimation.

**For more information on how to visualize images in CASA task `imview`, please see the brief guide for `plotms` and `viewer` on `Virtuale`.**

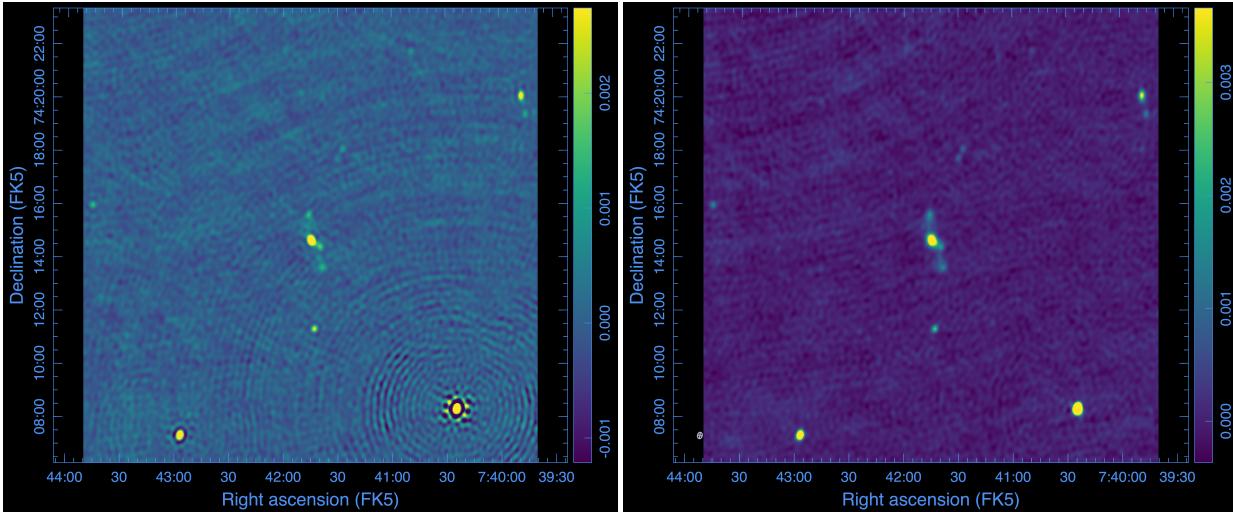


Figure 5. Left: a dirty image of the MS 0735.6+7421 field in viridis scale, with apparent sidelobes. Right: CLEANed image of the MS 0735.6+7421 field in viridis scale. The improvement after the cleaning procedure is evident.

## CLEAN with Weights

To see the effects of using different weighting schemes on the image, let's change the `weighting` parameter within `tclean` and inspect the resulting images. We will test the 'Briggs' weighting algorithm, by varying the `robust` parameter from 2 (natural weighting), to 0 (intermediate between natural and uniform), and finally to -2 (uniform weighting).

```
# In CASA. Briggs R=2 (natural) weighting
tclean(vis='MS0735-VLA_calibrated.ms', imagename='MS0735_R2', cell='2arcsec',
weighting='briggs', robust = 2, imsize=540, niter=10000, threshold='0.03mJy', interactive=True,
savemodel='none', stokes='I')

# In CASA. Briggs R=0 (intermediate) weighting
tclean(vis='MS0735-VLA_calibrated.ms', imagename='MS0735_R0', cell='2arcsec',
weighting='briggs', robust = 0, imsize=540, niter=10000, threshold='0.03mJy', interactive=True,
savemodel='none', stokes='I')

# In CASA. Briggs R=-2 (uniform) weighting
tclean(vis='MS0735-VLA_calibrated.ms', imagename='MS0735_R-2', cell='2arcsec',
weighting='briggs', robust = -2, imsize=540, niter=10000, threshold='0.03mJy', interactive=True,
savemodel='none', stokes='I')
```

- `weighting`: specification of the weighting scheme.
- `robust`: robustness parameter for Briggs weighting.

Figure 6 (left) shows that the natural weighted image ( $R=2$ ) is most sensitive to extended emission (beam size of  $21'' \times 15''$ ). Possible negative values around the extended emission (often referred to as a negative 'bowl') are a typical signature of unsampled visibilities near the origin of the uv-plane. That is, the flux density present at shorter baselines (or larger angular scales) than measured in this observation is not represented well. Uniform weighted data (Figure 6, right;  $R=-2$ ) shows the highest resolution ( $13'' \times 10''$ ) and intermediate weighted data (Figure 6, center;  $R=0$ ) is a compromise with a beam of  $14'' \times 11''$ .

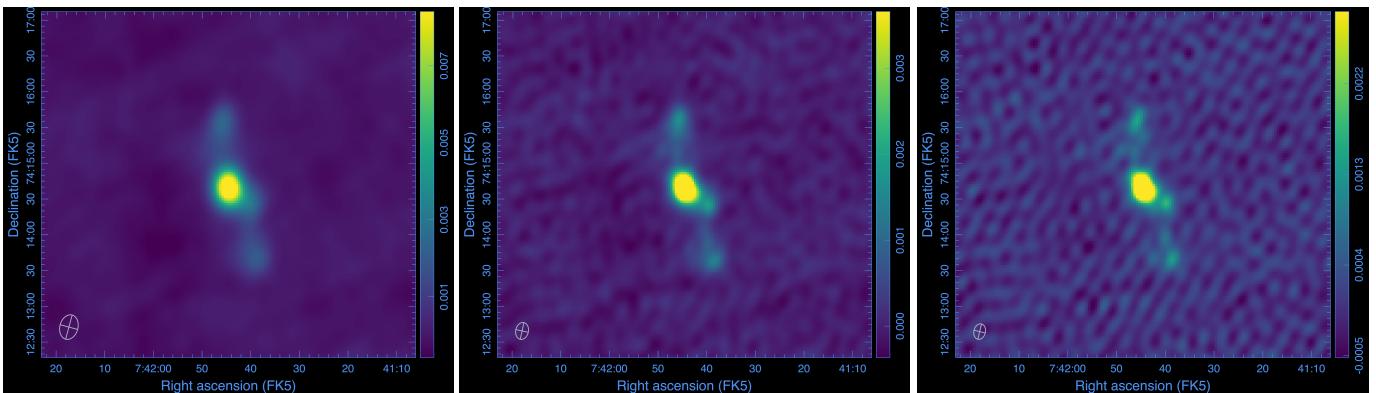


Figure 6. Left: Briggs  $R=2$  (natural) weighting. Center: Briggs  $R=0$  (intermediate) weighting; Right: Briggs  $R=-2$  (uniform) weighting.

## Scientific Analysis

Please see the files with the instructions for your own project: "info-VLAGroups.pdf" & "info-VLAGroups-optional.pdf" on Virtuale. Please also see see the brief guide for plotms and viewer on Virtuale.

The central radio galaxy in MS 0735.6+7421 is a famous radio galaxy, being one of the most powerful Active Galactic Nucleus (AGN) ever observed in a cluster of galaxies. Previous works ([McNamara et al. 2005](#); [Gitti et al. 2007](#); [Vantyghem et al. 2014](#)) have investigated the interaction of its jets and lobes with the hot gas in the cluster, as well as the intrinsic properties of the radio galaxy. A recent work that presented radio observations of MS 0735.6+7421 taken with different instruments and at different frequencies is [Biava et al. 2021](#). If you look at Table 5 of this paper, you will see that the total flux density of the central radio galaxy, measured at 1.42 GHz, is  $20 \pm 2$  mJy. Section 3.3.1 also reports the average *spectral index* of this radio source  $\alpha = -2.5$ . Remember that the synchrotron flux density,  $S$ , scales with frequency as:

$$S(v) \propto v^\alpha$$

where  $\alpha$  is the spectral index and represents the slope of the flux density vs frequency in the log-log space:

$$\alpha = \frac{\log_{10}\left(\frac{S_{v_1}}{S_{v_2}}\right)}{\log_{10}\left(\frac{v_1}{v_2}\right)}$$

With the information on the flux density of the radio source at 1.42 GHz,  $S_{1.42} = 20 \pm 2$  mJy, and its spectral index  $\alpha = -2.5$ , **use the above equation to predict the expected flux density at your central frequency of 1.2 GHz,  $S_{1.2}^{\text{exp}}$ .**

Compare the resulting expected flux density  $S_{1.2}^{\text{exp}}$  with the flux density measured in your image,  $S_{1.2}^{\text{obs}}$ . To do so, let's open the image we created in [tclean](#) as both *raster image* and *contour map*. Let's first change the color scaling (use the wrench, then change the *Scaling Power Cycles* to some negative values, e.g., -2). Now, while remaining in the wrench tool, move to the next panel, which controls the appearance of iso-brightness contours. Make sure that *base contour level* is set to 0, and then set the *Relative Contour Levels* to an array that samples multiples of the rms noise up to the peak in the image, e.g., [-3, 3, 6, 12, 24, 48, 96 ...]. Finally, set the *Unit Contour Level* to the rms noise of the image: draw a circle/rectangle on a portion of the image where you do not see any sources; then, from the Statistics panel, read the *Rms* value, e.g., 0.00001 Jy/beam. Set the *Unit Contour Level* to 0.00001.

To measure the flux density  $S_{1.2}^{\text{obs}}$  of the radio galaxy in MS 0735.6+7421 from the image, use the *Polygon* region to draw lines that enclose the  $3 \times \text{Rms}$  contour. Then, from the Statistics panel, read the *FluxDensity* value. This corresponds to  $S_{1.2}^{\text{obs}}$ . You can also calculate an uncertainty on the flux density,  $\delta S_v$ , that takes into account the RMS noise of the image,  $\sigma_{\text{rms}}$ , and the uncertainty in the absolute flux density scale, usually assumed to be 5% for the VLA:

$$\delta S_v = [(0.05 \cdot S_v)^2 + (\sigma_{\text{rms}})^2 \cdot N_{\text{beam}}]^{1/2},$$

where  $N_{\text{beam}}$  is the ratio between the area of the region you used to measure the flux density and the area of your clean beam, and can be calculated as:

$$N_{\text{beam}} = \frac{\text{Area}}{\text{BeamArea}}$$

In practice, to estimate  $N_{\text{beam}}$ , note the *BeamArea* (area of the gaussian beam in pixels<sup>2</sup>) and the *Npts* (number of pixels within the selected region) values reported in the Statistic panel of the viewer. The area of the region in pixels<sup>2</sup>, *Area*, is equal to *Npts*. Therefore  $N_{\text{beam}} = \text{Npts}/\text{BeamArea}$ .

At this point, you should now have the measurement of the flux density with its associated uncertainty, that is  $S_{1.2}^{\text{obs}} \pm \delta S_{1.2}^{\text{obs}}$ .

You can now proceed with the astrophysical discussion, for example:

- How does  $S_{1.2}^{\text{obs}}$  compare with the expected flux density that we have calculated above?
- Given the redshift of MS 0735.6+7421 ( $z = 0.216$ ), use a Cosmological Calculator such as the [NED Cosmology Calculator](#) to compute the luminosity distance of this source. Then, convert the flux density  $S_{1.2}$  [mJy] to monochromatic radio luminosity  $L_v$  [W/Hz], accounting for the spectral index  $\alpha = -2.5$  for the K-correction (see Equation 80 from [Condon & Matthews 2018](#)).

Finally, using the CASA task [exportfits](#), export the '.image' file to a FITS format that can be read by other softwares:

```
# In CASA. Export images to FITS format.
exportfits(imagename = 'MS0735.image', fitsimage = 'MS0735-VLA.fits')
```

This image can then be displayed also by [ds9](#) for a comparison with the Chandra X-ray image "MS0735-Chandra.fits" available on Virtuale (see instructions on "info-VLAGroups-optional.pdf").