VLA CASA Imaging-CASA5.4.0

From CASA Guides

Contents

Imaging

This guide has been prepared for CASA 5.4.0

This tutorial provides guidance on imaging procedures in CASA. The tutorial covers basic continuum cleaning and the influence of image weights, as well as wide-band and wide-field imaging techniques, multi-scale clean, an outlier field setup, and primary beam correction. Spectral line imaging procedures are explained but not covered in detail. For a more thorough example of spectral line imaging, refer to the VLA high frequency Spectral Line tutorial - IRC+10216 (https://casaguides.nrao.edu/index.php/VLA_high_frequency_Spectral_Line_tutorial_-_IRC%2B10216). This imaging tutorial concludes with basic image header calls and the conversion of Jy/beam to K surface brightness units through image arithmetic and header manipulation.

We will be utilizing data taken with the Karl G. Jansky Very Large Array, of a supernova remnant G055.7+3.4. The data were taken on August 23, 2010, in the first D-configuration for which the new wide-band capabilities of the WIDAR (Wideband Interferometric Digital ARchitecture) correlator were available. The 8-hour-long observation includes all available 1 GHz of bandwidth in L-band, from 1 - 2 GHz in frequency.

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 3C391 (https://casaguides.nrao.edu/index.php?title=VLA_Continuum_Tutorial_3C391) and VLA high frequency Spectral Line tutorial - IRC+10216 (https://casaguides.nrao.edu/index.php?title=VLA_high_frequency_Spectral_Line_tutorial_-_IRC%2B10216) guides.

```
A copy of the calibrated data (1.2GB) can be downloaded from http://casa.nrao.edu/Data/EVLA/SNRG55/SNR_G55_10s.calib.tar.gz
```

Your first step will be to unzip and untar the file in a terminal (before you start CASA):

```
tar -xzvf  SNR_G55_10s.calib.tar.gz
```

Then start CASA as usual via the **casa** command, which will bring up the ipython interface and launches the logger.

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 PSF (Point Spread Function), 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 found by performing a cross-correlation between the dirty image, and the PSF. The brightest parts (actually their PSFs) are subtracted, and the process is repeated again for the next brighter sources. Variants of CLEAN, such as multi-scale CLEAN, take into account extended kernels which may be better suited for extended objects.

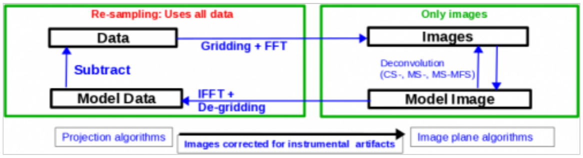


Figure 1
The CLEAN major and minor cycles, indicating the steps undertaken during gridding, projection algorithms, and creation of images.

For single pointings, CASA uses the Hogbom cleaning algorithm by default in the task tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) (*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 to create a (dirty) image. The minor cycles then operate in the image domain to find the clean components that are added to the clean model: repeatedly performing a PSF+image correlation 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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about), 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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) 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.

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 (http://www.aspbooks.org/a/volumes/table_of_contents/?book_id=292). The chapter on Deconvolution (http://www.aspbooks.org/a/volumes/article_details/?paper_id=17942) may prove helpful. In addition, imaging presentations are available on the Synthesis Imaging Workshop (http://www.cvent.com/events/15th-synthesis-imaging-workshop/agenda-e34ab08b2a854a48bae723c8fe86d2d4.aspx) and VLA Data Reduction Workshop (https://science.nrao.edu/science/meetings/2016/vla-data-reduction/program) webpages. The CASA Documentation (https://casa.nrao.edu/casadocs/casa-5.4.0) chapter on Synthesis Imaging (https://casa.nrao.edu/casadocs/casa-5.4.0/synthesis-imaging/imaging-overview) provides a wealth of information on the CASA implementation of tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) and related tasks.

Finally, we refer users to the VLA Observational Status Summary (https://science.nrao.edu/facilities/vla/docs/manuals/oss) and the Guide to Observing with the VLA (https://science.nrao.edu/facilities/vla/docs/manuals/obsguide) 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 the surface brightness sensitivity. There are three main weighting schemes that are used in interferometry:

1) **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.

2) **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 the 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.

3) **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 for 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) (http://www.aoc.nrao.edu/dissertations/dbriggs/).

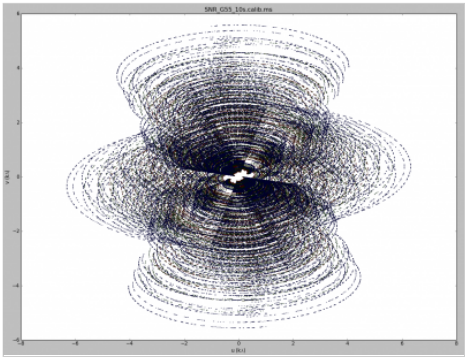
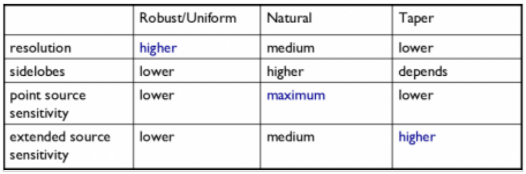


Figure 2
u,v coverage for the 8-hour observation of the supernova remnant G055.7+3.4

For a visual comparison between these three weighting schemes, please see the section on "CLEAN with Weights" in this guide.

Tapering: In conjunction with the above weighting schemes, one can specify the *uvtaper* parameter within tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about), 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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) is an elliptical Gaussian function which effectively removes long baselines and degrades the resolution. For extended structure 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 (https://casa.nrao.edu/casadocs/casa-5.4.0/synthesis-imaging/imaging-overview) section of the CASA Documentation for the details of the weighting implementation in CASA's tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about).



	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.

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 a FWHM equal to $42/\nu_{GHz}$ 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 below). 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 (https://library.nrao.edu/public/memos/evla/EVLAM_195.pdf). These newer beam corrections are the default in CASA 5.4.0.

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 (https://casaguides.nrao.edu/index.php?title=VLA_Continuum_Tutorial_3C391). In this guide, we discuss methods for imaging large areas from single-pointing data.

As these data were taken in the D-configuration, we can check the Observational Status Summary (https://science.nrao.edu/facilities/vla/docs/manuals/oss/performance/resolution)'s section on VLA resolution to find that the synthesized beam for uniform weighting will be approximately 46 arcsec. Variations in flagging, weighting scheme, and effective frequency may result in deviations from this value. The synthesized beam is effectively the angular resolution of the image. As we will see later, the synthesized beam of our data hovers around 29 arcsec or, for the extreme of uniform weighting, around 26"x25".

Cell (Pixel) Size and Image Size

For the most effective cleaning, we recommend using a pixel size such that there are 3 - 5 pixels across the synthesized beam. Based on the assumed synthesized beam size of 46 arcsec, we will use a cell (pixel) size of 8 arcsec.

In the tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) task, the image size is defined by the number of pixels in the RA and dec directions. The execution time of tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) depends on the image size. Large images generally take more computing time. There are some particular image sizes (by 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 $2^n * 10$ pixels, e.g. 160, 1280 pixels, etc. for improved processing speeds.

Our target field contains bright point sources that are well outside the primary beam. The VLA, in particular when using multi-frequency synthesis (see below), will have significant sensitivity outside the main lobe of the primary beam. Particularly at the lower VLA frequencies, sources that are located outside the primary beam may still be bright enough to be detected in the primary beam sidelobes, causing artifacts that interfere with the targeted field in the main part of the primary beam. Such sources need to be cleaned to remove the dirty beam response of these interfering sources from the entire image. This can be achieved either by creating a very large image that encompasses these interfering sources, or by using 'outlier' fields centered on the strongest sources (see section on outlier fields below). A large image has the added advantage of increasing the field of view for science (albeit at lower sensitivity beyond the primary beam). But other effects will start to become significant, like the non-coplanarity of the sky. Large image sizes will also slow down the deconvolution process. Details are provided in Sanjay Bhatnagar's presentation: "Advanced Imaging: Imaging in the Wide-band Wide-field era" (https://science.nrao.edu/science/meetings/2016/vla-data-reduction/copy_of_WF_WB.pdf) given at the 2016 VLA data reduction workshop (https://science.nrao.edu/science/meetings/2016/vla-data-reduction/program).

The calls to the tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) task within this guide will create images that are approximately 170 arcminutes on a side, or almost 6x the size of the primary beam, encompassing its first and second sidelobes. This is ideal for showcasing both the problems inherent in wide-band/wide-field imaging, as well as some of the solutions currently available in CASA to deal with these issues. We therefore set the image size to 1280 pixels on each side, for efficient processing speed. (1280 pixels x 8 arcsec/pixel = 170.67 arcmin)

Clean Output Images

As a result of the CLEAN algorithm, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will create a number of output images. For a parameter setting of *imagename='<imagename>'*, these image names will be:

<imagename>.residual: the residual after subtracting the clean model (unit: Jy/beam, where beam refers to the dirty beam)
<imagename>.model: the clean model, not convolved (unit: Jy/pixel)
<imagename>.psf: the point-spread function (dirty beam)
<imagename>.pb: the normalized sensitivity map, which corresponds to the primary beam in the case of a single-pointing image
<imagename>.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)
<imagename>.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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) 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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about), first remove all images of that name using 'rmtables()':

Important: By default, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) 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.

```
# In CASA
rmtables('<imagename>.*')
```

This method is preferable over 'rm -rf' as it also clears the cache.

Note that interrupting tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) by Ctrl+C may corrupt your visibilities - you may be better off choosing to let tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) 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.

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 (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) with niter=0, which will run the task without performing any CLEAN iterations.

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.dirty',
      imsize=1280, cell='8arcsec', pblimit=-0.01, niter=0,
      stokes='I', savemodel='modelcolumn')
```

- *imagename='SNR_G55_10s.dirty'*: the root filename used for the various tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) outputs.
- *imsize=1280*: the image size in number of pixels. A single value will result in a square image.
- *cell='8arcsec'*: the size of one pixel; again, entering a single value will result in a square pixel size.
- *niter=0*: this controls the number of iterations done in the minor cycle.
- *interactive=False*: For a tutorial that covers more of an interactive clean, please see the VLA high frequency Spectral Line tutorial - IRC+10216. (https://casaguides.nrao.edu/index.php?title=VLA_high_frequency_Spectral_Line_tutorial_-_IRC%2B10216)
- *savemodel='modelcolumn*: controls writing the model visibilities to the model data column. For self-calibration we currently recommend setting savemodel='modelcolumn'. The default value is "none": so, this must be changed for the model to be saved. The use of the virtual model with MTMFS in tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) is still under commissioning, so we recommend currently to set *savemodel='modelcolumn'.*
- *stokes='I'*: since we have not done any polarization calibration, we only create a total-intensity image. For using CLEAN while including various Stoke's parameters, please see the 3C391 CASA guide. (https://casaguides.nrao.edu/index.php?title=EVLA_Continuum_Tutorial_3C391-CASA4.5)

```
# In CASA
viewer('SNR_G55_10s.dirty.image')
```

```
# In CASA
viewer('SNR_G55_10s.dirty.psf')
```

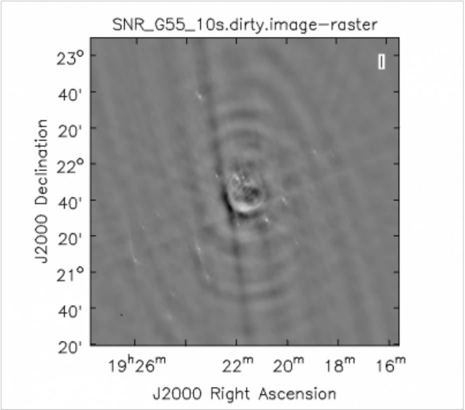


Figure 3a
A dirty image of the supernova remnant G55.7+3.4 in greyscale, with apparent sidelobes.

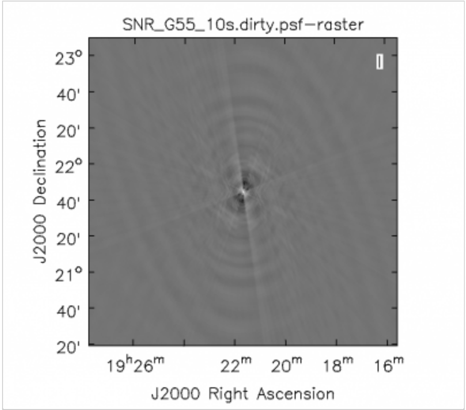


Figure 3b
The Point Spread Function (PSF) in greyscale. This is also known as the dirty beam.

The images may be easier to see in grey scale. To change the color scheme, click on "Data Display Options" (wrench icon, upper left corner) within the viewer, and change the color map to "Greyscale 1". You may also wish to change the scaling power options to your liking (e.g., -1.5). To change the brightness and contrast, assign a mouse button to this type of editing by clicking on the "Colormap fiddling" icon (black/white circle), and click/drag the mouse over the image to change the brightness (left-right mouse movement) and contrast (up-down mouse movement).

Note that the clean beam is only defined after some clean iterations. The dirty image has therefore no beam size specified in the header, and the PSF image (Figure 3b) is the representation of the response of the array to a point source. Even though it is empty because we set *niter=0*, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will still produce a model file. Thus we could progress into actual cleaning by simply restarting tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) with the same image root name (and *niter*>0).

Regular CLEAN & RMS Noise

Now, we will create a regular clean image using mostly default values to see how deconvolution improves the image quality. The first run of tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will use a fixed number of minor cycle iterations of *niter=1000* (default is 0); the second will use *niter=10000*. Note that you may have to play with the image color map and brightness/contrast to get a better view of the image details.

```
# In CASA. Create default clean image.
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.Reg.Clean.niter1K',
      imsize=1280, cell='8arcsec', pblimit=-0.01, niter=1000, savemodel='modelcolumn')
```

```
# In CASA.
viewer('SNR_G55_10s.Reg.Clean.niter1K.image')
```

The logger indicates that the image was obtained in two major cycles, and some improvements over the dirty image are visible. But clearly we have not cleaned deep enough yet; the image still has many sidelobes, and an inspection of the residual image shows that it still contains source flux and structure. So let's increase the *niter* value to 10,000 and compare the images.

```
# In CASA. Create default clean image with niter = 10000
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.Reg.Clean.niter10K',
      imsize=1280, cell='8arcsec', pblimit=-0.01, niter=10000, savemodel='modelcolumn')
```

```
# In CASA.
viewer('SNR_G55_10s.Reg.Clean.niter10K.image')
```

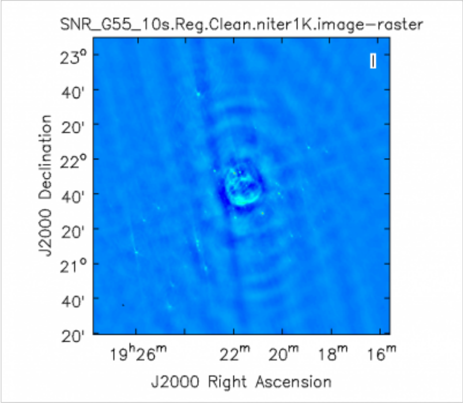


Figure 4a
Regular run of TCLEAN, with niter=1000.

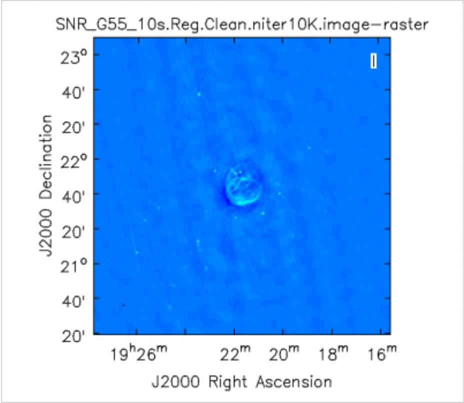


Figure 4b
Regular run of TCLEAN, with niter=10000.

As we can see from the resulting images, increasing the niter values (minor cycles) improves our image by reducing prominent sidelobes significantly (Figure 4b). One could now further increase the *niter* parameter until the residuals are down to an acceptable level. To determine the number of iterations, one needs to keep in mind that tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) 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. To avoid cleaning too deeply, we will set a *threshold* parameter that will stop minor cycle clean iterations once a peak residual value is being reached.

First, we will utilize the *SNR_G55_10s.Reg.Clean.niter1K.image* (Figure 4a) image to give us an idea of the rms noise (your sigma value). With the image open within the viewer, click on the 'Rectangle Drawing' button (Rectangle with R) and draw a square on the image at a position with little source or sidelobe contamination. Doing this should open up a "Regions" dock, which holds information about the selected region, including the pixel statistics in the aptly named "Statistics" tab. Take notice of the rms values as you click/drag the box around empty image locations, or by drawing additional boxes at suitable positions. If the "Regions" dock is not displayed, click on "View" in the menu bar at the top of the viewer and click on the "Regions" check-box.

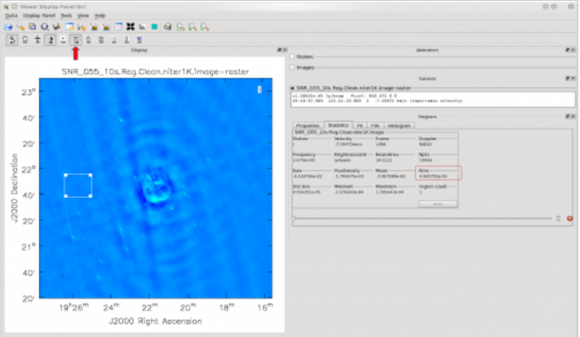


Figure 5
Attempting to find the lowest rms value within the CLEAN'ed image using niter=1000, in order to calculate our threshold.

The lowest rms value we found was in the order of about $5 \cdot 10^{-5}$ Jy/beam, which we will use to calculate our threshold. There really is no set standard, but fairly good threshold values can vary anywhere between 3.0 - 5.0*sigma; using clean boxes (see the section on interactive cleaning) allows one to go to lower thresholds. For our purposes, we will choose a threshold of 3*sigma. Doing the math results in a value of $15 \cdot 10^{-5}$ or equivalently 0.15mJy/beam. Therefore, for future calls to the tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) task, we will set *threshold='0.15mJy'*. The clean cycle will be stopped when the residual peak flux density is equals to or less than the *threshold* value, or when the maximum number of iterations *niter* is reached. To ensure that the stopping criterion is indeed *threshold*, *niter* should be set to a very high number. **In the following, we nevertheless will use *niter=1000* to keep the execution times of tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) on the low end as we focus on explaining different imaging methods.**

An alternative method to determine the approximate rms of an image is to use the VLA Exposure Calculator (https://science.nrao.edu/facilities/vla/docs/manuals/propvla/determining) and to enter the observing conditions. Make sure that the chosen bandwidth reflects the data after RFI excision.

CLEAN with Weights

To see the effects of using different weighting schemes to the image, let's change the *weighting* parameter within tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) and inspect the resulting images. We will be using the *Natural, Uniform, and Briggs* weighting algorithms. Here, we have chosen a smaller image size to mainly focus on our science target.

```
# In CASA. Natural weighting
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.natural', weighting='natural',
      imsize=540, cell='8arcsec', pblimit=-0.01, niter=1000, interactive=False, threshold='0.15mJy', savemodel='modelcolumn')
```

- *weighting*: specification of the weighting scheme. For Briggs weighting, the robust parameter will be used.
- *threshold='0.15mJy'*: threshold at which the cleaning process will halt.

```
# In CASA. Uniform weighting
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.uniform',weighting='uniform',
      imsize=540, cell='8arcsec', pblimit=-0.01, niter=1000, interactive=False, threshold='0.15mJy', savemodel='modelcolumn')
```

```
# In CASA. Briggs weighting, with robust set to 0.0
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.briggs', weighting='briggs', robust=0.0,
      imsize=540, cell='8arcsec', pblimit=-0.01, niter=1000, interactive=False, threshold='0.15mJy', savemodel='modelcolumn')
```

```
# In CASA. Open the viewer and select the created images.
viewer()
```

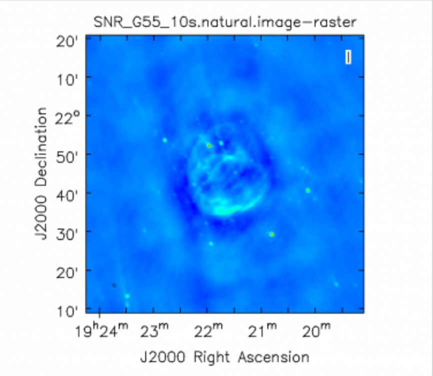


Figure 6a
Natural weighting.

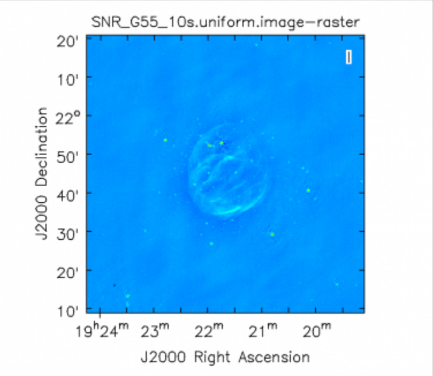


Figure 6b
Uniform weighting.

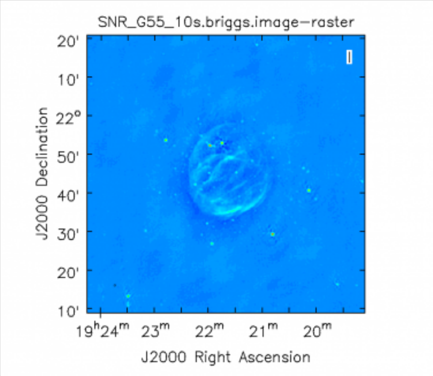


Figure 6c
Briggs weighting.

Figure 6a shows that the natural weighted image is most sensitive to extended emission (beam size of 46"x41"). The 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 6b) shows the highest resolution (26"x25") and Briggs (Figure 6c) 'robust=0' (default value is 0.5) is a compromise with a beam of 29"x29". To image more of the extended emission, the 'robust' parameter could be tweaked further toward more positive values.

Multi-Scale CLEAN

Since G55.7+3.4 is an extended source with many angular scales, a more advanced form of imaging involves the use of multiple scales. MS-CLEAN is an extension of the classical CLEAN algorithm for handling extended sources. Multi-Scale CLEAN works by assuming the sky is composed of emission at different angular scales and works on them simultaneously, thereby creating a linear combination of images at different angular scales. For a more detailed description of Multi Scale CLEAN, see T.J. Cornwell's paper Multi-Scale CLEAN deconvolution of radio synthesis images (http://arxiv.org/abs/0806.2228).

We will use a set of scales (expressed in units of the requested pixel or cell size) which are representative of the scales that are present in the data, including a zero-scale for point sources.

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.MultiScale', deconvolver='multiscale', scales=[0,6,10,30,60],
       smallscalebias=0.9, imsize=1280, cell='8arcsec', pblimit=-0.01, niter=1000,weighting='briggs', stokes='I',
       robust=0.0, interactive=False, threshold='0.12mJy', savemodel='modelcolumn')
```

- *multiscale=[0,6,10,30,60]*: a set of scales on which to clean. A general guideline when using multiscale is 0, 2xbeam, 5xbeam (where beam is the synthesized beam), and larger scales up to about half the minor axis maximum scale of the mapped structure. Since these are in units of the pixel size, our chosen values will be multiplied by the requested cell size. Thus, we are requesting scales of 0 (a point source), 48, 80, 240, and 480 arcseconds (8 arcminutes). Note that 16 arcminutes (960 arcseconds) roughly corresponds to the size of G55.7+3.4.
- *smallscalebias=0.9*: This parameter is known as the **small scale bias**, and helps with faint extended structure by balancing the weight given to smaller structures, which tend to have higher surface brightness but lower integrated flux density. Increasing this value gives more weight to smaller scales. A value of 1.0 weights the largest scale to zero, and a value of less than 0.2 weights all scales nearly equally. The default value is 0.6.

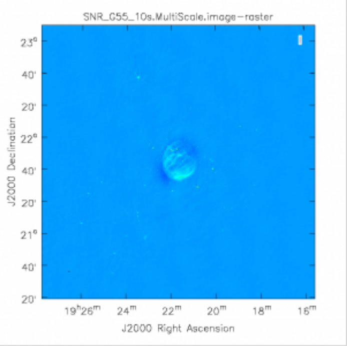


Figure 7
G55.7+3.4 Multi-Scale Clean

```
# In CASA
viewer('SNR_G55_10s.MultiScale.image')
```

The logger will show how much cleaning is performed on the individual scales.

This is the fastest of the advanced imaging techniques described here, but it's easy to see that there are artifacts in the resulting image (Figure 7). We can use the viewer (https://casa.nrao.edu/casadocs/latest/global-task-list/task_viewer/about) to explore the point sources near the edge of the field by zooming in on them (Figure 8). Click on the "Zooming" button on the upper left corner and highlight an area by making a square around the portion where you would like to zoom-in. Double-click within the square to zoom in on the selected area. The square can be resized by clicking/dragging the corners, or removed by pressing the "Esc" key. After zooming in on the area, we can see some radio sources that have prominent arcs, as well as spots with a six-pointed pattern surrounding them.

Next we will further explore advanced imaging techniques to mitigate the artifacts seen toward the edges of the image.

Multi-Scale, Wide-Field CLEAN (w-projection)

The next tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) algorithm we will employ is w-projection, which is a wide-field imaging technique that takes into account the non-coplanarity of the baselines as a function of distance from the phase center (Figure 9). For wide-field imaging, the sky curvature and non-coplanar baselines result in a non-zero w-term. The w-term, introduced by the sky and array non-coplanarity, results in a phase term that will limit the dynamic range of the final image. Applying 2-D imaging to such data will result in artifacts around sources away from the phase center, as we saw in running MS-CLEAN. Note that this affects mostly the lower frequency bands where the field of view is large.

The w-term can be corrected by faceting (which projects the sky curvature onto many smaller planes) in either the image or uv-plane. The latter is known as w-projection. A combination of the two can also be employed within tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) by setting the parameter *gridder='wproject* ***and*****deconvolver='multiscale** . If w-projection is employed, it will be done for each facet. Note that w-projection is an order of magnitude faster than the image-plane based faceting algorithm, but requires more memory.

For more details on w-projection, as well as the algorithm itself, see "The Noncoplanar Baselines Effect in Radio Interferometry: The W-Projection Algorithm" (http://adsabs.harvard.edu/abs/2008ISTSP...2..647C). Also, the chapter on Imaging with Non-Coplanar Arrays (http://www.aspbooks.org/a/volumes/article_details/?paper_id=17953) may be helpful.

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.wProj',
       gridder='wproject', wprojplanes=-1, deconvolver='multiscale', scales=[0,6,10,30,60],
       smallscalebias=0.9, imsize=1280, cell='8arcsec', pblimit=-0.01, niter=1000, weighting='briggs',
       stokes='I', robust=0.0, interactive=False, threshold='0.15mJy', savemodel='modelcolumn')
```

- *gridder='wproject'*: Use the w-projection algorithm.
- *wprojplanes=-1*: The number of w-projection planes to use for deconvolution. Setting to -1 allows CLEAN to automatically choose an acceptable number of planes for the given data.

```
# In CASA
viewer('SNR_G55_10s.wProj.image')
```

This will take slightly longer than the previous imaging round; however, the resulting image which employs Multi-Scale and w-projection (Figure 10b) has noticeably fewer artifacts than the image which just employed Multi-Scale (Figure 10a). In particular, compare the same outlier source in the Multi-Scale w-projected image with the Multi-Scale-only image: note that the swept-back arcs have disappeared. There are still some obvious imaging artifacts remaining, though.

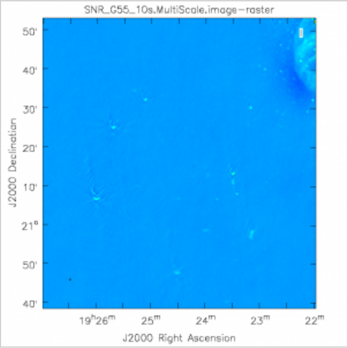


Figure 8
Artifacts around point sources

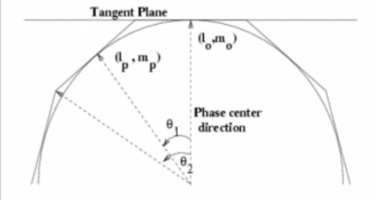


Figure 9
Facting when using wide-field gridder, which can be used in conjunction with w-projection.

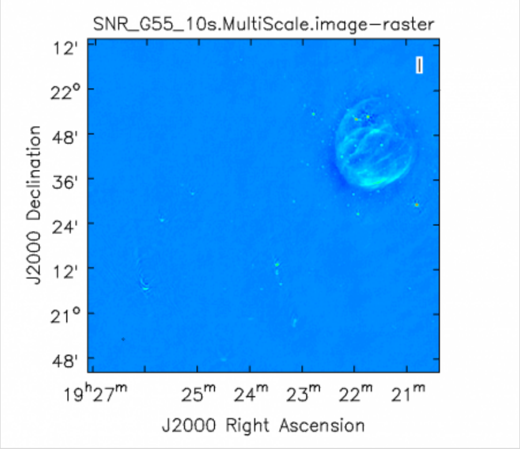


Figure 10a
Multi-Scale image which shows arcs around point sources far from the phase center.

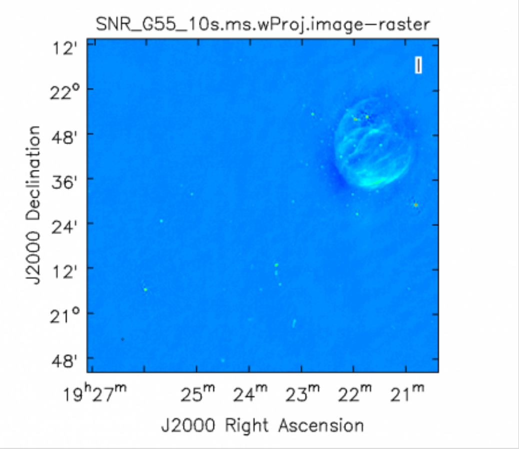


Figure 10b
Multi-Scale and w-projection image shows improvements in the image by removing prominent artifacts.

Multi-Scale, Multi-Term Multi-Frequency Synthesis

A consequence of simultaneously imaging all the frequencies across the wide fractional bandwidth of the VLA is that the primary and synthesized beams have substantial frequency-dependent variation over the observed band. If this variation is not accounted for, it will lead to imaging artifacts and will compromise the achievable image rms.

The coordinates of the (u,v) plane are measured in wavelengths. Observing at several frequencies results means that a single baseline will sample several ellipses in the (u,v) plane, each with different sizes. We can therefore fill in the gaps in the single frequency (u,v) coverage (Figure 11) to achieve a much better image fidelity. This method is called Multi-Frequency Synthesis (MFS), which is the default cleaning mode in CLEAN. For a more detailed explanation of the MS-MFS deconvolution algorithm, please see the paper A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry (http://arxiv.org/abs/1106.2745).

The Multi-Scale, Multi-Term Multi-Frequency Synthesis (MS-MT-MFS) algorithm provides the ability to simultaneously image and fit for the intrinsic source spectrum in each pixel. The spectrum is approximated using a polynomial Taylor-term expansion in frequency, with the degree of the polynomial as a user-controlled parameter. A least-squares approach is used, along with the standard clean-type iterations.

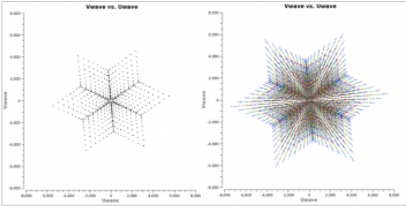


Figure 11
Left: (u,v) coverage snapshot at a single frequency. *Right:* Multi-Frequency Synthesis snapshot of (u,v) coverage. Using this algorithm can greatly improve (u,v) coverage, thereby improving image fidelity.

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.ms.MTMFS',
       gridder='standard', imsize=1280, cell='8arcsec', specmode='mfs',
       deconvolver='mtmfs', nterms=2, scales=[0,6,10,30,60], smallscalebias=0.9,
       interactive=False, niter=1000,  weighting='briggs', pblimit=-0.01,
       stokes='I', threshold='0.15mJy', savemodel='modelcolumn')
```

- *nterms=2*: the number of Taylor-terms to be used to model the frequency dependence of the sky emission. Note that the speed of the algorithm will depend on the value used here (more terms will be slower). nterms=2 will fit a spectral index, while nterms=3 will fit a spectral index and spectral curvature.

For Multi-Term MFS, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) produces images with extension tt*, where the number in place of the asterisk indicates the Taylor term: <imagename>.image.tt0 is the total intensity image (Figure 12) and <imagename>.image.alpha will contain an image of the spectral index in regions where there is sufficient signal-to-noise (Figure 13). For Figure 12, we have included a color wedge (use the data display option icon to have it displayed) to give an idea of the flux within the image. The spectral index image can help convey information about emission mechanisms as well as optical depth of the source.

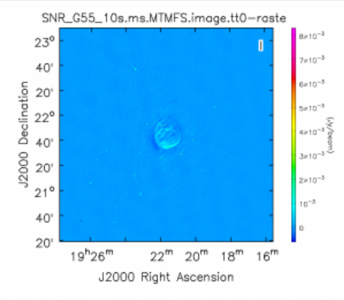


Figure 12
Multi-Scale imaged with MT-MFS and nterms=2. Artifacts around point sources close to phase center are reduced while point sources away from the phase center still show artifacts.

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.image.tt0')
```

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.alpha')
```

Note: To replicate Figure 13, open the image within the viewer, click on "Panel Display Options" (wrench with a small P), change "background color" to white, and adjust your margins under the "Margins" button. To view the color wedge, click on the "Data Display Options" (wrench without the P, next to the folder icon), click on the "Color Wedge" button, click on "Yes" under "display color wedge", and adjust the various parameters to your liking.

For more information on the multi-term, multi-frequency synthesis mode and its outputs, see the Deconvolution Algorithyms section (https://casa.nrao.edu/casadocs/casa-5.4.0/synthesis-imaging/deconvolution-algorithms#cit5) in the CASA documentation.

Inspect the brighter point sources in the field near the supernova remnant. You will notice that some of the artifacts, which had been symmetric around the sources, are gone; however, as we did not use w-projection, there are still strong features related to the non-coplanar baseline effects still apparent for sources further away.

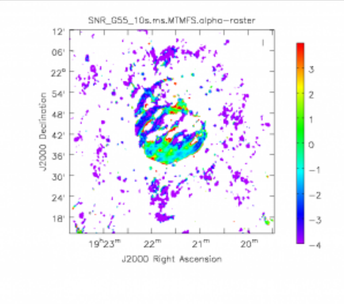


Figure 13
Spectral Index image

At this point, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) takes into account the frequency variation of the synthesized beam but not the frequency variation of the primary beam. For low frequencies and large bandwidths, this can be substantial; e.g., 1 - 2 GHz L-band observations result in a variation of a factor of 2. One effect of imaging with MFS across such a large fractional bandwidth is that primary beam nulls will be blurred; the interferometer is sensitive everywhere in the field of view. However, if there is no correction made to account for the variation in primary beam with frequency, sources away from the phase center appear to have a steeply false spectral slope. A correction for this effect should be made with the task widebandpbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_widebandpbcor/about) or set *pbcor=True* (see section on Primary Beam Correction below).

Multi-Scale, Multi-Term Multi-Frequency Synthesis with W-Projection

We will now combine the w-projection and MS-MT-MFS algorithms. Be forewarned -- these imaging runs will take a while, and it's best to start them running and then move on to other things.

Using the same parameters for the individual-algorithm images above, but combined into a single tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) run, we have:

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.ms.MTMFS.wProj',
       gridder='wproject',wprojplanes=-1, pblimit=-1, imsize=1280, cell='8arcsec', specmode='mfs',
       deconvolver='mtmfs', nterms=2, scales=[0,6,10,30,60], smallscalebias=0.9,
       interactive=False, niter=1000,  weighting='briggs',
       stokes='I', threshold='0.15mJy', savemodel='modelcolumn')
```

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.wProj.image.tt0')
```

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.wProj.alpha')
```

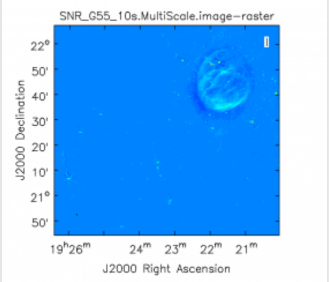


Figure 14a
Multi-Scale.

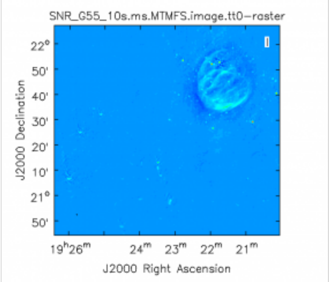


Figure 14b
Multi-Scale, MTMFS.

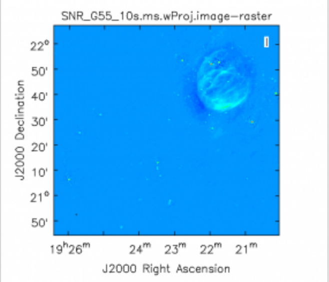


Figure 14c
Multi-Scale, w-projection.

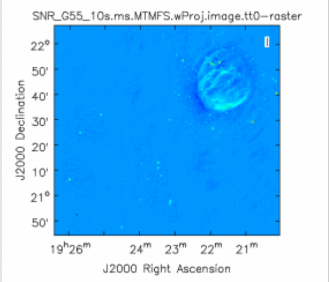


Figure 14d
Multi-Scale, MTMFS, w-projection.

Again, looking at the same outlier source, we can see that the major sources of error have been removed, although there are still some residual artifacts. One possible source of error is the time-dependent variation of the primary beam; another is the fact that we have only used *nterms=2*, which may not be sufficient to model the spectra of some of the point sources. Some weak RFI may also show up that may need additional flagging.

Imaging Outlier Fields

When strong sources are far from the main target but still strong enough to produce sidelobes in the main image, they should be cleaned. Sometimes, however, it is not practical to image very large fields for this purpose. An alternative is to add outlier fields. This mode will allow the user to specify a few locations that are then cleaned along with the main image. The outlier fields should be centered on strong sources that, e.g., are known from sky catalogs or are identified by other means.

In order to find outlying sources, it will help to first image a very large field with lower resolution, and to identify bright outliers using the CASA viewer. By moving the mouse cursor over the sources, we can grab their positions and use the information within an outlier file. In this example, we've called our file 'outliers.txt'. It contains the names, sizes, and positions of the outliers we want to image (see section 5.3.18.1 (https://casa.nrao.edu/docs/cookbook/casa_cookbook006.html#sec356) of the CASA cookbook).

Open your favorite text editor and input the following:

```
#content of outliers.txt
#
#outlier field1
imagename=Outlier1
imsize=[320,320]
phasecenter = J2000 19:23:27.693 22.37.37.180
#
#outlier field2
imagename=Outlier2
imsize=[320,320]
phasecenter = J2000 19:25:46.888 21.22.03.365
```

tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will then be executed as follows:

```
# In CASA
tclean(vis='SNR_G55_10s.calib.ms', imagename='SNR.MS.MFS-Main', outlierfile='outliers.txt',
      imsize=[640,640], cell='8arcsec',deconvolver='multiscale', scales=[0,6,10,30,60], smallscalebias=0.9,
      interactive=False, niter=1000,  weighting='briggs', robust=0, stokes='I', threshold='0.15mJy',
      savemodel='modelcolumn',pblimit=-1)
```

- *outlierfile='outliers.txt'*: the name of the outlier file.

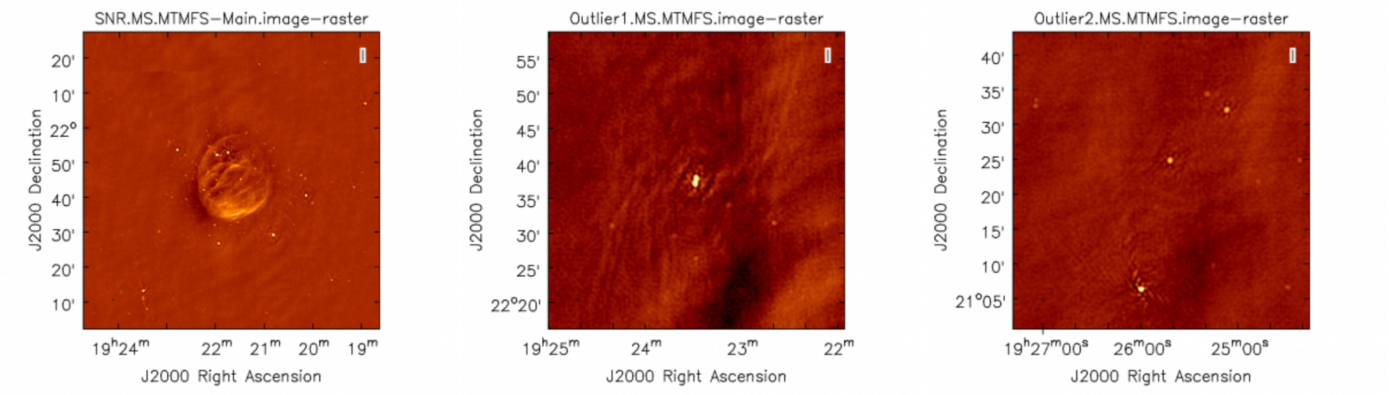
We can view the images with viewer (https://casa.nrao.edu/casadocs/latest/global-task-list/task_viewer/about) one at a time. Due to the difference in sky coordinates, they cannot be viewed on the same window display.

```
# In CASA
viewer('SNR.MS.MFS-Main.image')
```

```
# In CASA
viewer('Outlier1.image')
```

```
# In CASA
viewer('Outlier2.image')
```

In the resulting images, we have changed the color map to "Hot Metal 2" in order to show the different colors that can be used within the viewer. Feel free to play with other color maps that may be better suited for your screen.



Primary Beam Correction

In interferometry, the images formed via deconvolution are representations of the sky, multiplied by the primary beam response of the antenna. The primary beam can be described as Gaussian with a size depending on the observing frequency. Images produced via tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) are by default not corrected for the primary beam pattern (important for mosaics), and therefore do not have the correct flux density away from the phase center.

Correcting for the primary beam, however, can be done during tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) by using the *pbcor* parameter. Alternatively, it can be done after imaging using the task impbcor (https://casa.nrao.edu/docs/TaskRef/impbcor-task.html) for regular data sets, and widebandpbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_widebandpbcor/about) for those that use Taylor-term expansion (*nterms > 1*). A third alternative is utilizing the task immath (https://casa.nrao.edu/casadocs/latest/global-task-list/task_immath/about) to manually divide the *<imagename>.image* by the *<imagename>.flux* image (*<imagename>.flux.pbcoverage* for mosaics).

Flux corrected images usually don't look pretty due to the noise at the edges being increased. Flux densities, however, should only be calculated from primary beam corrected images. Let's run the impbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_impbcor/about) task to correct our multiscale image.

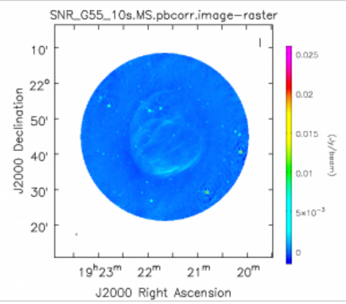


Figure 16
Primary beam corrected image using the widebandpbcor task for the MS.MTMFS.wProj image created during the CLEAN process.

```
# In CASA
impbcor(imagename='SNR_G55_10s.MultiScale.image', pbimage='SNR_G55_10s.MultiScale.pb',
        outfile='SNR_G55_10s.MS.pbcorr.image')
```

- *imagename*: the image to be corrected
- *pbimage*: the <imagename>.pb image as a representation of the primary beam

```
# In CASA
viewer('SNR_G55_10s.MS.pbcorr.image')
```

Let us now use the widebandpbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_widebandpbcor/about) task for wide-band (*nterms*>1) images. Note that for this task, we will be supplying the image name that is the prefix for the Taylor expansion images, tt0 and tt1, which must be on disk. Such files were created during the last Multi-Scale, Multi-Term, Widefield run of CLEAN. widebandpbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_widebandpbcor/about) will generate a set of images with a "pbcor.image.*" extension.

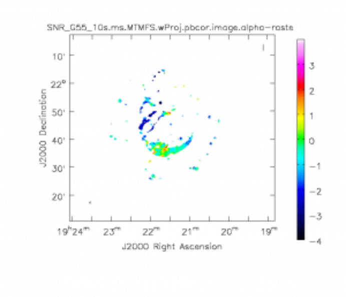


Figure 17
Primary beam corrected spectral index image using the widebandpbcor task.

```
# In CASA
widebandpbcor(vis='SNR_G55_10s.calib.ms', imagename='SNR_G55_10s.ms.MTMFS.wProj',
              nterms=2, action='pbcor', pbmin=0.2, spwlist=[0,1,2,3],
              weightlist=[0.5,1.0,0,1.0], chanlist=[25,25,25,25], threshold='0.6mJy')
```

- *spwlist=[0,1,2,3]*: We want to apply this correction to all spectral windows in our calibrated measurement set.
- *weightlist=[0.5,1.0,0,1.0]*: Since we did not specify reference frequencies during tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about), the widebandpbcor (https://casa.nrao.edu/casadocs/latest/global-task-list/task_widebandpbcor/about) task will pick them from the provided image. Running the task, the logger reports the multiple frequencies used for the primary beam, which are 1.305, 1.479, 1.652, and 1.825 GHz. We selected weights using information external to this guide: The first chosen frequency lies within spectral window 0, which had lots of data flagged due to RFI. The *weightlist* parameter allows us to give this chosen frequency less weight. The primary beam at 1.6 GHz lies in an area with no data, therefore we give a weight value of zero for this frequency. The remaining frequencies 1.479 and 1.825 GHz lie within spectral windows which contained less RFI, therefore we provide a larger weight percentage. (Exercise for the reader: use an amplitude vs. frequency plot in plotms (https://casa.nrao.edu/casadocs/latest/global-task-list/task_plotms/about) to verify these claims.)
- *pbmin=0.2*: Gain level below which not to compute Taylor-coefficients or apply a primary beam correction.
- *chanlist=[25,25,25,25]*: Make primary beams at frequencies corresponding to channel 25 in each spectral window.
- *threshold='0.6mJy'*: Threshold in the intensity map, below which not to recalculate the spectral index. We adjusted the threshold to get good results, as too high of a threshold resulted in errors and no spectral index image, and too low of a threshold resulted in bogus values for the spectral index image.

Note: CASA may produce a warning stating that *Images are not contiguous along the concatenation axis.* This is usually an indication of missing channels due to flagging (or unevenly spaced basebands) and is not a cause for worry.

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.wProj.pbcor.image.tt0')
```

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.wProj.pbcor.image.alpha')
```

It's important to note that the image will cut off at about 20% of the HPBW, as we are confident of the accuracy within this percentage. Anything outside becomes less accurate, thus there is a mask associated with the creation of the corrected primary beam image (Figure 16). Spectral indices may still be unreliable below approximately 70% of the primary beam HPBW (Figure 17).

It would be a good exercise to use the viewer (https://casa.nrao.edu/casadocs/latest/global-task-list/task_viewer/about) to plot both the primary beam corrected image, and the original cleaned image and compare the intensity (Jy/beam) values, which should differ slightly.

Imaging Spectral Cubes

For spectral line imaging using CASA tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about), set *specmode='cube', and the* width *parameter accepts wither frequency or velocity values. Both options will create a spectral axis in frequency, but entering a velocity width will add an additional velocity label.*

The following keywords are important for spectral modes (velocity in this example):

```
# In CASA
specmode       =     'cube'        #  Spectral definition type (mfs, cube, cubedata, cubesource)
nchan          =         -1        #  Number of channels (planes) in output image; -1 = all data specified by spw, start, and width parameters.
start          =         ''        #  Velocity of first channel: e.g '0.0km/s'(''=first channel in first SpW of MS)
width          =         ''        #  Channel width e.g '-1.0km/s' (''=width of first channel in first SpW of MS)
outframe       =         ''        #  spectral reference frame of output image; '' =input
veltype        =    'radio'        #  Velocity definition of output image
restfreq       =         ''        #  Rest frequency to assign to image (see help)
```

The spectral dimension of the output cube will be defined by these parameters and tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will regrid the visibilities to it. Note that invoking cvel (https://casa.nrao.edu/casadocs/latest/global-task-list/task_cvel/about) before imaging is in most cases not necessary, even when two or more measurement sets are being provided at the same time.

The cube is specified by a *start* velocity in km/s or MHz, the *nchan* number of channels and a channel *width* (where the latter can also be negative for decreasing velocity cubes).

To correct for Doppler effects, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) also requires a *outframe* velocity frame, where the default *LSRK* and *BARY* are the most popular Local Standard of Rest (kinematic) and sun-earth Barycenter references. *veltype* defines whether the data will be gridded via the optical or radio velocity definition. A description of the available options and definitions can be found in the VLA observing guide (https://science.nrao.edu/facilities/vla/docs/manuals/obsguide/modes/line) and the CASA cookbook (https://casa.nrao.edu/docs/cookbook/casa_cookbook014.html). By default, tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) will produce a cube using the LRSK and radio definitions.

An example of spectral line imaging procedures is provided in the VLA high frequency Spectral Line tutorial - IRC+10216 (https://casaguides.nrao.edu/index.php?title=VLA_high_frequency_Spectral_Line_tutorial_-_IRC%2B10216).

Beam per Plane

For spectral cubes spanning relatively wide ranges in frequency, the synthesized beam can vary substantially across the channels. To account for this, CASA will calculate separate beams for each channel when the difference in beams is more than half a pixel across the cube. All CASA image analysis tasks are capable of handling such cubes.

If it is desired to have a cube with a single synthesized beam, two options are available. It is possible for tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) to do the smoothing, by setting the parameter *smonothfactor*. The second option is to use the task imsmooth (https://casa.nrao.edu/casadocs/latest/global-task-list/task_imsmooth/about), after cleaning, with *kernel='commonbeam'.* This task will clean up all header variables such that only a single beam appears in the data. It may be wise to go with this second option, which will allow you to first verify the clean results before smoothing the image.

Image Header

The image header holds meta data associated with your CASA image. The task imhead (https://casa.nrao.edu/casadocs/latest/global-task-list/task_imhead/about) will display this data within the casalog, and also output the result as a Python dictionary in the CASA terminal. We will first run imhead (https://casa.nrao.edu/casadocs/latest/global-task-list/task_imhead/about) with *mode='summary'* :

```
# In CASA
imhead(imagename='SNR_G55_10s.ms.MTMFS.wProj.image.tt0', mode='summary')
```

- *mode='summary'* : gives general information about the image, including the object name, sky coordinates, image units, the telescope the data was taken with, and more.

For further information, run *mode='list'* . We will assign a variable 'header' to also capture the output dictionary:

```
# In CASA
header=imhead(imagename='SNR_G55_10s.ms.MTMFS.wProj.image.tt0', mode='list')
```

- *mode='list'* : gives more detailed information, including beam major/minor axes, beam primary angle, the location of the max/min intensity, and lots more. Essentially this mode displays the FITS header variables.

Image Conversion

Finally, as an example, we convert our image from intensity to main beam brightness temperature.

We will use the standard equation

$$T = 1.222 \times 10^6 \frac{S}{\nu^2 \theta_{maj} \theta_{min}}$$

where the main beam brightness temperature T is given in units of Kelvin, the intensity S in Jy/beam, the reference frequency ν in GHz, and the major and minor beam sizes θ in arcseconds.

For a beam of 29.30"x29.03" and a reference frequency of 1.579 GHz (as taken from the previous imhead (https://casa.nrao.edu/casadocs/latest/global-task-list/task_imhead/about) run) we calculate the brightness temperature using immath (https://casa.nrao.edu/casadocs/latest/global-task-list/task_immath/about):

```
# In CASA
immath(imagename='SNR_G55_10s.ms.MTMFS.wProj.image.tt0', mode='evalexpr',
       expr='1.222e6*IM0/1.579^2/(29.30*29.03)',
       outfile='SNR_G55_10s.ms.MTMFS.wProj.image.tt0-Tb')
```

- *mode='evalexpr'* : immath (https://casa.nrao.edu/casadocs/latest/global-task-list/task_immath/about) is used to calcuate with images

- *expr*: the mathematical expression to be evaluated. The images are abbreviated as IM0, IM1, ... in the sequence of their appearance in the *imagename* parameter.

Since immath (https://casa.nrao.edu/casadocs/latest/global-task-list/task_immath/about) only changes the values but not the unit of the image, we will now change the new image header 'bunit' key to 'K'. To do so, we will run imhead (https://casa.nrao.edu/casadocs/latest/global-task-list/task_imhead/about) with *mode='put'* :

```
# In CASA
imhead(imagename='SNR_G55_10s.ms.MTMFS.wProj.image.tt0-Tb', mode='put', hdkey='bunit', hdvalue='K')
```

- *hdkey*: the header keyword that will be changed

- *hdvalue*: the new value for header keyword

Launching the viewer will now show our image in brightness temperature units:

```
# In CASA
viewer('SNR_G55_10s.ms.MTMFS.wProj.image.tt0-Tb')
```

A few particular issues to look out for:

- The default value of *robust* is now 0.5; this produces the same behavior as *robust=0.0* in AIPS.
- Parameter *pblimit* is used to ensure that the image is only produced out to a distance from the phasecenter where the sensitivity of the Primary Beam is equal to *pblimit*. The default value of *pblimit* is 0.2. For this example, where we are making an image that is much, much larger than the Primary Beam, it is necessary to set *pblimit=-1* so that no limit is applied.
- Parameter *savemodel* is used rather than clean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_clean/about)'s *usescratch* to indicate whether the model should be saved in a new column in the ms, as a virtual model, or not at all. The default value is "none": so, this must be changed for the image to be saved. The use of the virtual model with MTMFS in tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) is still under commissioning, so we recommend currently to set *savemodel='modelcolumn'.*
- Some parameter default values in tclean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_tclean/about) are different than the equivalent default values in clean (https://casa.nrao.edu/casadocs/latest/global-task-list/task_clean/about).

Last checked on CASA Version 5.4.0.

↲ **CASAguides**

Retrieved from "https://casaguides.nrao.edu/index.php?title=VLA_CASA_Imaging-CASA5.4.0&oldid=26261"

- This page was last modified on 5 December 2018, at 13:37.