Simulations and Imaging with CASA!

09:15 - 09:50 -- ALMA Overview talk, ALMA Cycle 5 proposals

09:50 - 10:40 -- Local ALMA Science Highlights:
Brendan Bowler, Yao-Lun Yang,
Jackie Champagne, Sam Factor

10:40 - 11:00 -- Break

11:00 - 11:20 -- ALMA Observing Tool and ALMA data archive

11:20 - 12:30 -- Small group work on ALMA OT and/or archive

12:30 - 13:40 -- Lunch (provided for registered participants)

13:40 - 14:10 -- Introduction to data imaging and ALMA simulations

14:10 - 16:30 -- Small group work on Data Imaging or Simulations

NRAO

Goals:

- Simulate an ALMA image for your Cycle 5 proposals.
- Simulate ALMA observations for your paper.
- Image ALMA archival data or your own data.
- Learn how cleaning/imaging of interferometer data are done. Reproduce TW Hydra or other famous ALMA observations.
- Other goals?

CASA (Common Astronomy Software Application package):

https://casa.nrao.edu/casa_obtaining.shtml

Downloads

Please follow these links for downloading the code for your specific operating system. Installation instructions are provided, too.
CASA *Pre-releases* are code under development. Use with caution as the code is still undergoing changes, is less tested and not well documented.

Package sizes vary between 0.4-1.1GB.

	CASA		Pipelines	
CASA	**CASA Release 4.7.2**	**CASA Pre-releases**	**ALMA 4.7.0-1**	**VLA 4.7.2**
Linux Packages				Installation
Red Hat 7				
Red Hat 6				
Mac OS X Packages				Installation
OS X 10.11 (El Capitan)				
OS X 10.10 (Yosemite)				

Two ways to run CASA tasks

```
CASA <12>: help(importfits)
Help on importfits task:

Convert an image FITS file into a CASA image
```

- Scripting interface

```
importfits(fitsimage='ngc3256.fits', imagename='ngc3256.im', overwrite=True)
```

Two ways to run CASA tasks

- Tasking interface

```
CASA <16>: default(importfits)
```
or

```
CASA <14>: tget(importfits)
Restored parameters from file importfits.last
```

```
CASA <17>: inp
---------> inp()
#  importfits :: Convert an image FITS file into a CASA image
fitsimage           =        ''        #  Name of input image FITS file
imagename           =        ''        #  Name of output CASA image
whichrep            =         0        #  If fits image has multiple coordinate
                                       #   reps, choose one.
```

```
CASA <19>: fitsimage='test.fits'
```

```
CASA <20>: go()
Executing:  importfits()
```

Overview of ALMA simulation:

- **Simobserve**

Provide an input "model" and specify a telescope/configuration used for the observations. Provide parameters like source position, integration time, and noise. Simobserve can then generate mock visibility data (measurement set ".ms").

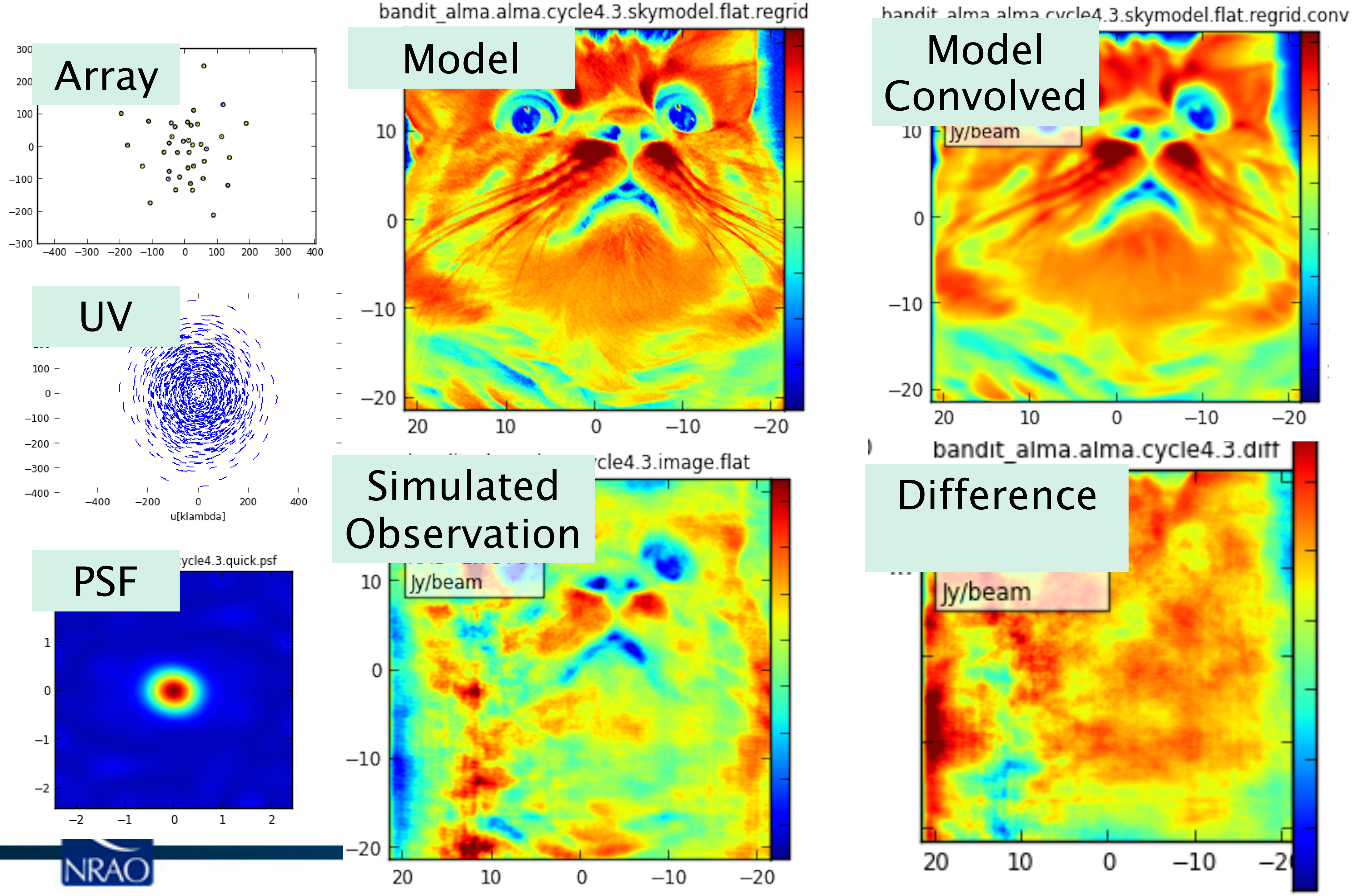
```
CASA <2>: help(simobserve)
Help on simobserve task:
```

- **Simanalyze**

"Clean" the visibility data generated from Simbobserve using CASA's clean task. It also calculates the differences between simulated observations and the original model data.

```
CASA <4>: help(simanalyze)
Help on simanalyze task:
```

Overview of ALMA simulation:

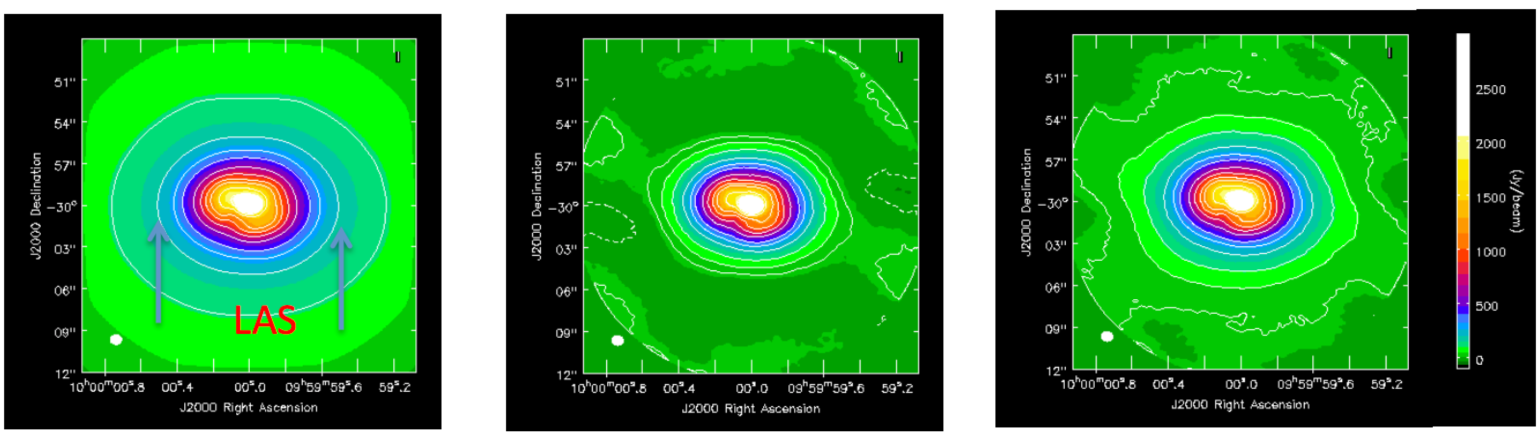


Purpose of simulation:

- Check the fidelity of image (more important in the earlier cycles).
- Demonstrate how well you can recover the flux. Help the reviewer to visualize the expected outcome.
- Properly fit models to your data (e.g., fitting in the uv plane).
- Others?

Images using 12-m C2 array with a resolution of 0.8”x0.7” in pa 80d

MODEL | 12-m image | 12m+7m Image



Restored flux 11000 Jy | 7000 Jy | 9000 Jy

Primary beam corrected: 20% cutoff: Contours: -20,20,50,100,200,300,400,600,800,1000,1200,1600,2000

ALMA simulation tutorial:

https://casaguides.nrao.edu/index.php/Simulating_Observations_in_CASA_4.3

page | discussion | view source | history

log in

Main Page

search

Search

Go | Search

tools

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information

Simulating Observations in CASA 4.3

- This guide describes steps used to simulate interferometric observations in CASA.
- The Guide to Simulating ALMA Data gives an introduction to simulations, with examples and a discussion of their relevance to ALMA observing proposals
- This guide is applicable to CASA version 4.3 and later. For older versions of CASA please see Simulating Observations in CASA 4.2.
- **Please note that the Cycle 4 OT pointing file format has changed such that it is not currently possible to directly simulate ALMA observations in CASA using this path.** We expect this issue to be fixed in the near-future CASA 4.6 release *provided users export pointings from the OT using absolute sexagesimal (not relative) positions*

Contents [hide]

Simulating Interferometric Observations in CASA

Simulating interferometric observations in CASA proceeds in the following steps:

1. Make a model image. The model image is a representation of the sky brightness distribution that you would like to simulate observing, stored initially as a FITS file. There are several paths to making the FITS file, discussed below.
2. Generate uv data with the `simobserve` task.
3. Image the simulated observation with the `simanalyze` task.

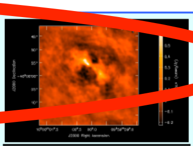
ALMA simulation tutorial:

https://casaguides.nrao.edu/index.php/Simulating_Observations_in_CASA_4.3

Tutorials

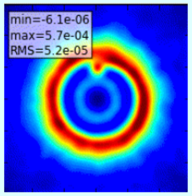
Simulation Guide for New Users (CASA 4.3)

A fully annotated tutorial that uses a Spitzer SAGE 8 micron continuum image of 30 Doradus and scales it to greater distance. A good place for new users to start



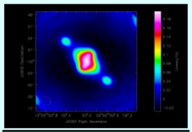
Protoplanetary Disk Simulation (CASA 4.3)

A sky model with a lightly annotated script that simulates a protoplanetary disk. Uses a theoretical model of dust continuum from Sebastian Wolff, scaled to the distance of a nearby star. This is another fairly generic simulation - if you're short on time, you probably don't need to go through this one and the New Users guide, but it can be useful to go through multiple examples.



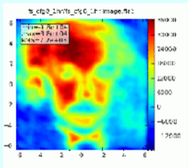
Simulation Guide Component Lists (CASA 4.3)

Tutorial for simulating data based on multiple sources (using both a FITS image and a component list). If you are interested in simulating from a list of simple sources (point, Gaussian, disk), rather than or in addition to a sky model image, then read the considerations here.



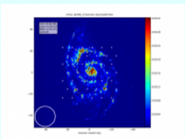
Einstein-Face (CASA 4.3)

A sky model and lightly annotated script that simulates the face of Einstein as seen by ALMA. This simulation is particularly useful for those who wish to better understand spatial filtering by an interferometer, but doesn't demonstrate new capabilities of the simulation tasks beyond those described above.



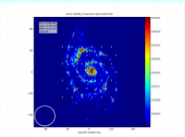
ACA Simulation (CASA 4.3)

A tutorial for simulating ALMA observations that use multiple configurations or use the 12-meter array in combination with the ALMA Compact Array. This tutorial demonstrates combining data from each ALMA component "by hand". This guide is of particular interest to those wishing to explore using the 12-m array in combination with the ACA, and those interested in combining data from multiple 12-m array configurations.



Simalma (CASA 4.3)

This tutorial demonstrates how to use **simalma**, a task that simplifies simulations that include the main 12-m array plus the ACA. Like the previous guide, this one is of particular interest to those wishing to explore multi-component ALMA observations.



ALMA Observation Support Tool:

A web-based simulation tool: http://almaost.jb.man.ac.uk/

ALMA Observation Support Tool

Version 5.0

OST NEWS HELP QUEUE LIBRARY ALMA HELPDESK

!!! OST User Notice: Version 5.0 released - 21/03/2017 !!! (more info). **OST Team**

Array Setup:

Instrument: ALMA — Select the desired ALMA antenna configuration.

Sky Setup:

Source model: OST Library: Central point source — Choose a library source model or supply your own.

Upload: Browse... No file selected. — You may upload your own model here (max 10MB).

Declination: -35d00m00.0s — Ensure correct formatting of this string (+/-00d00m00.0s).

Image peak / point flux in mJy 0.0 — Rescale the image data with respect to new peak value. Set to 0.0 for no rescaling of source model.

Observation Setup:

Observing mode: ○ Spectral ◉ Continuum — Spectral or continuum observations?

NRAO

Overview of Clean:

https://casaguides.nrao.edu/index.php/First_Look_at_Imaging

page | discussion | view source | history

- Main Page

search

Search

Go | Search

tools

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information

First Look at Imaging

Contents [hide]

About this Guide

The purpose of this tutorial is to provide a first look at imaging ALMA data for those new to CASA.

Data delivered by ALMA is pre-calibrated either by ARC staff or by the ALMA calibration pipeline. The delivered data is ready for imaging. This tutorial demonstrates the basic procedures that will help you complete the imaging steps.

This guide covers the same material used in hands-on training sessions at NRAO Community Days events and ALMA Data Reduction tutorials presented by NAASC staff.

The "Imaging Tutorials for CASA Beginners" guides work for CASA versions 4.2, 4.3, and 4.4, and possibly earlier versions.

Getting CASA

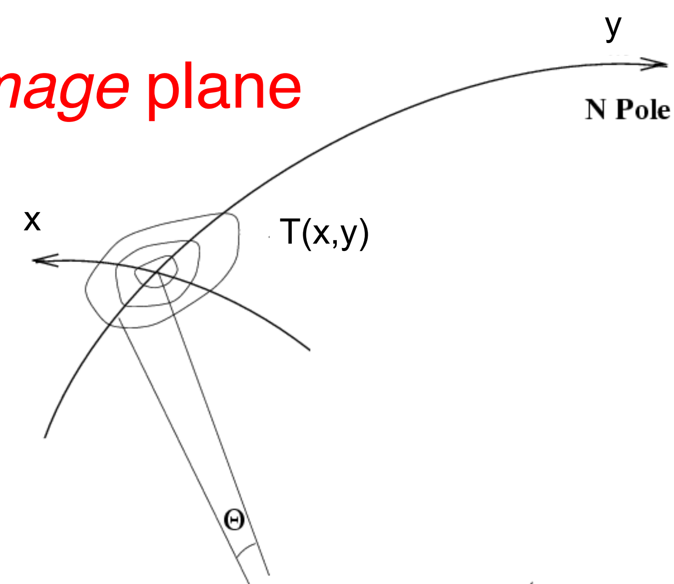
If you do not already have CASA installed on your machine, you will have to download and install it.

From Sky Brightness to Visibility

1. An interferometer measures the interference pattern observed by pairs of apertures
2. The interference pattern is directly related to the source brightness. In particular, for small fields of view the complex visibility, V(u,v), is the 2D Fourier transform of the brightness on the sky, T(x,y)

(van Cittert-Zernike theorem)

image plane



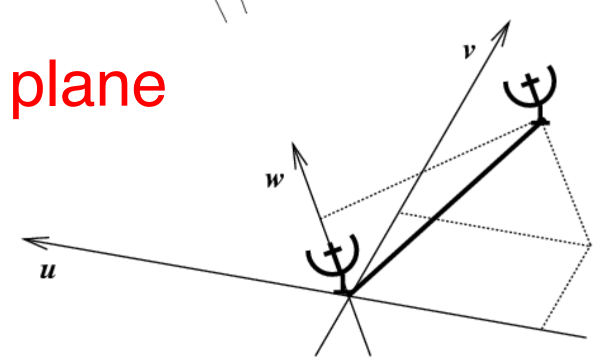
Fourier space/domain

$$V(u,v) = \int\int T(x,y) e^{2\pi i(ux+vy)} dxdy$$

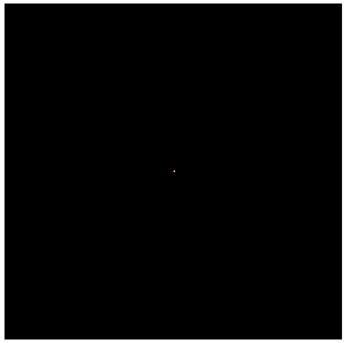
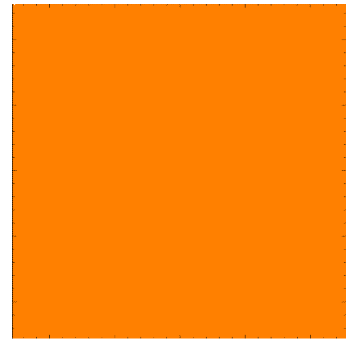
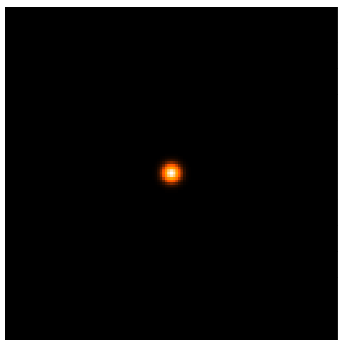
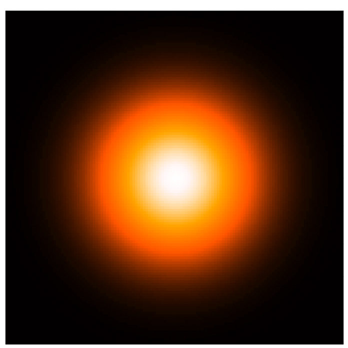
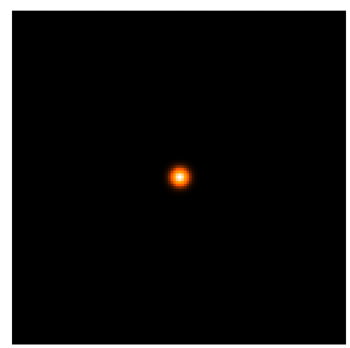
$$T(x,y) = \int\int V(u,v) e^{-2\pi i(ux+vy)} dudv$$

Image space/domain

uv plane



Some 2D Fourier Transform Pairs

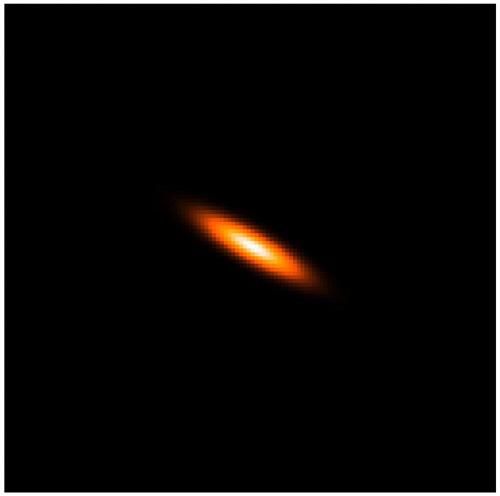
T(x,y)				Amp{V(u,v)}
δ Function		⇌		Constant
Gaussian		⇌		Gaussian
Gaussian		⇌		Gaussian

narrow features transform to wide features (and vice-versa)

More 2D Fourier Transform Pairs

T(x,y) Amp{V(u,v)}

elliptical Gaussian

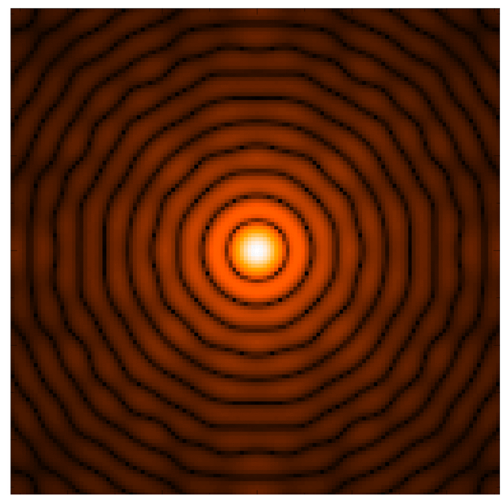


⇌

elliptical Gaussian

Disk

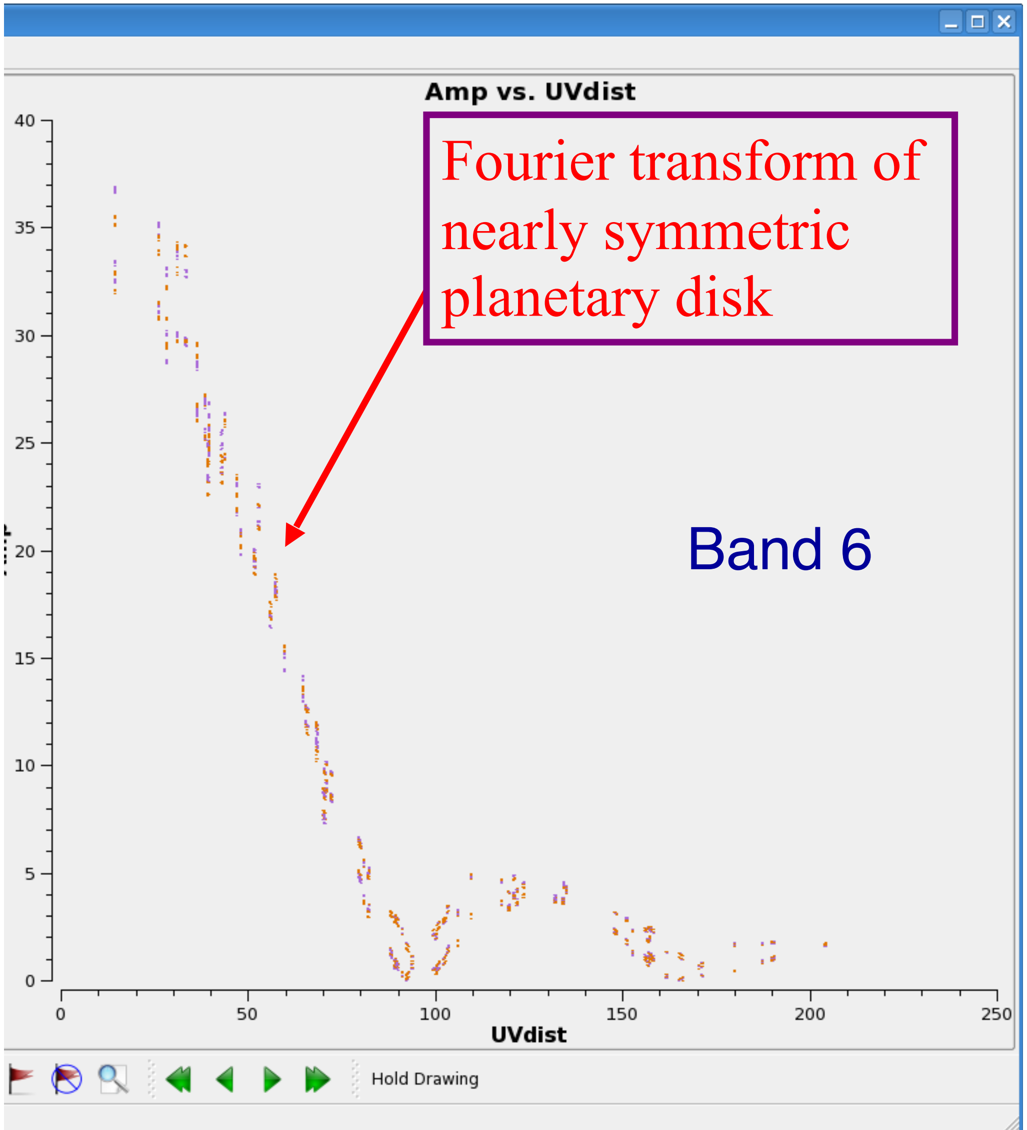
⇌

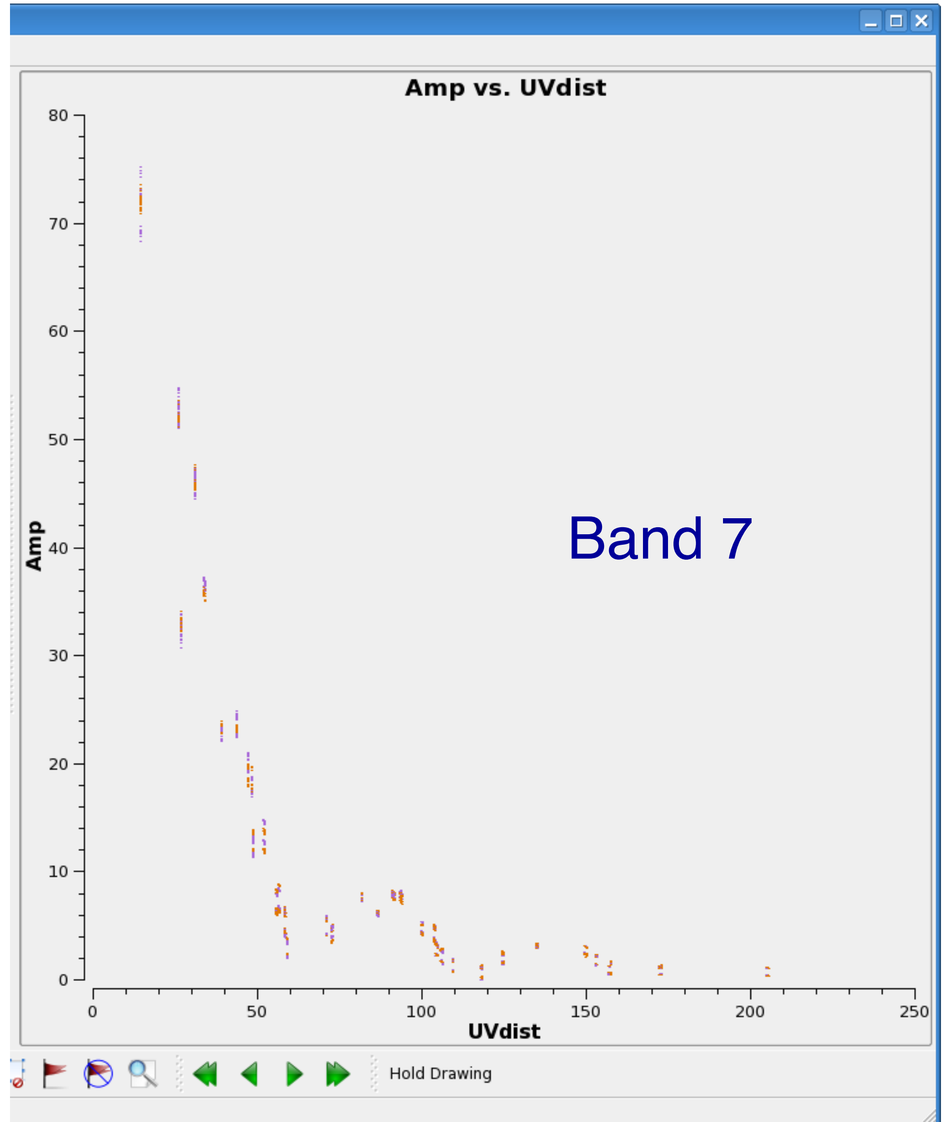


Bessel

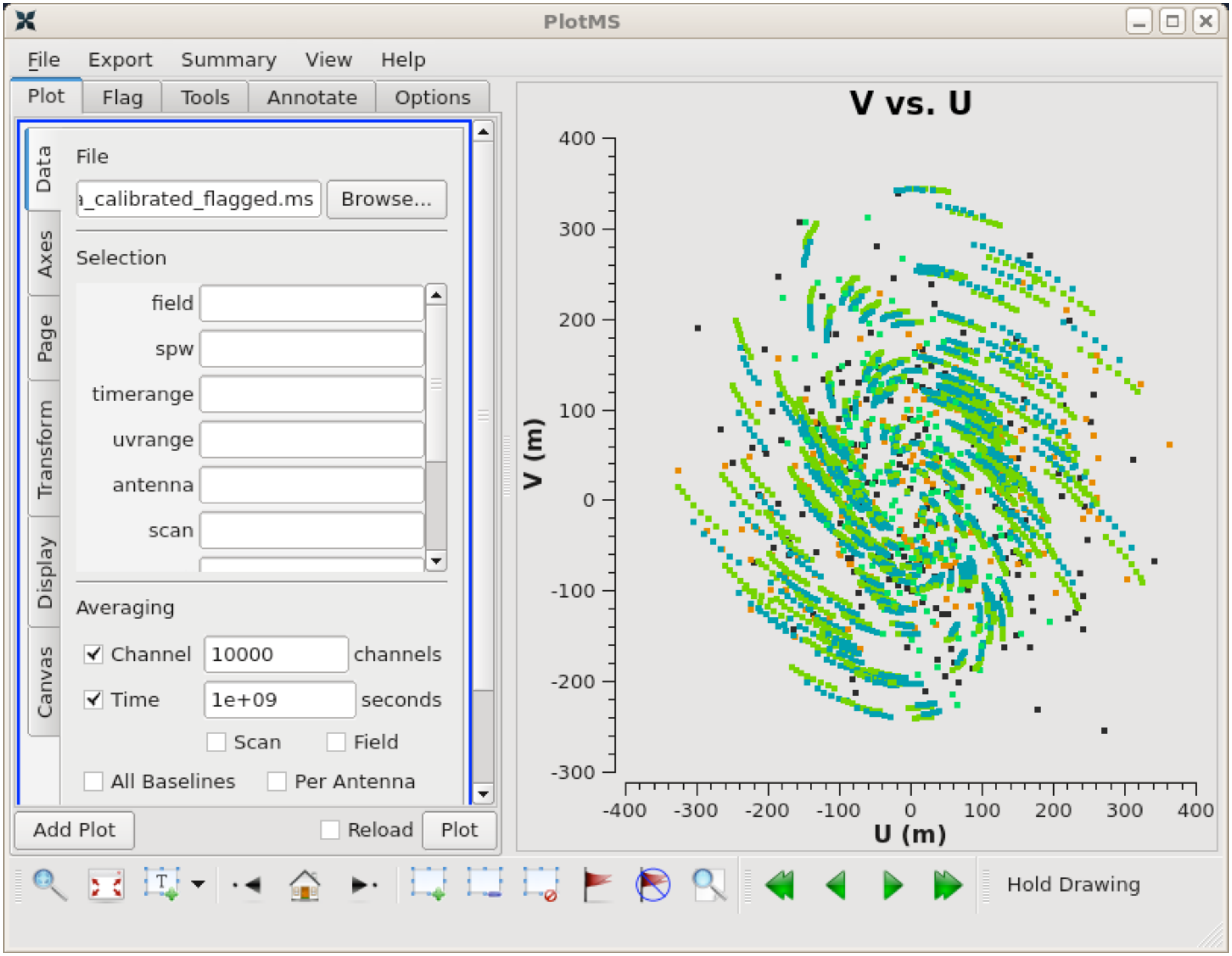
sharp edges result in many high spatial frequencies
(sinc function, “ringing”, Gibbs phenomenon)

ALMA observes planetary disk



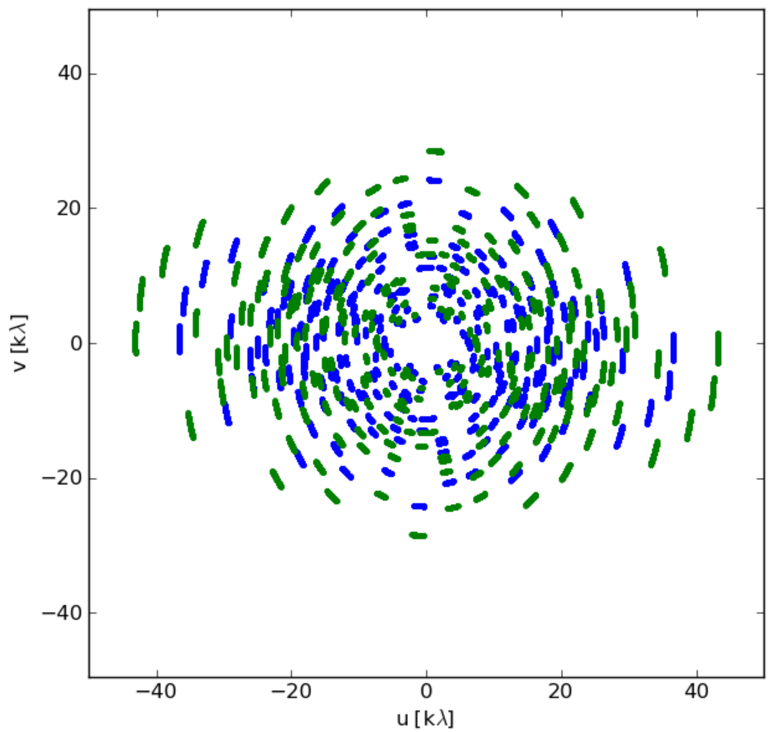


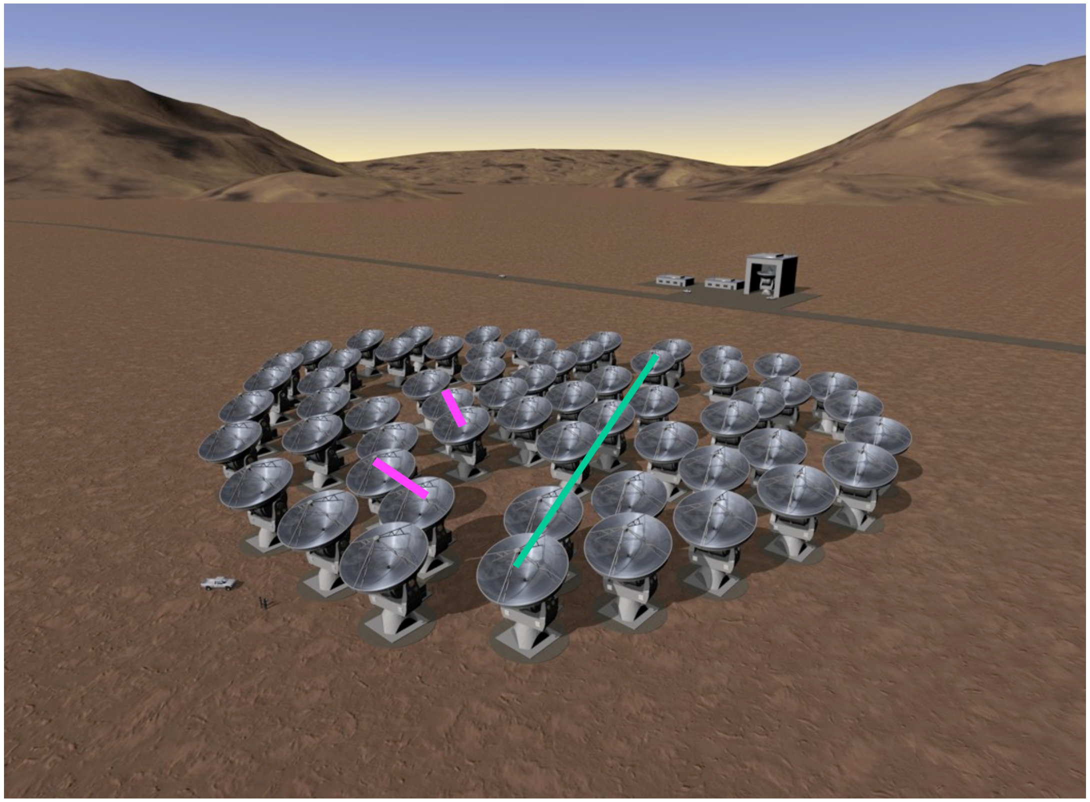
Plotms: Versatile examination of UV data



Sampling Function

Interferometers cannot see the entire Fourier/uv domain. But each antenna pair samples one spot: ➔ **imperfect image**





Small uv-distance: **short baselines** (measure extended emission)
Long uv-distance: **long baselines** (measure small scale emission)
Orientation of baseline also determines orientation in the uv-plane
Each visibility has a phase and an amplitude

NRAO

Dirty Images from a Dirty Beam

- We sample the Fourier domain at discrete points

$$B(u,v) = \sum_k (u_k, v_k)$$

- The inverse Fourier Transform is

$$T^D(x,y) = FT^{-1}\{B(u,v) \times V(u,v)\}$$

- The convolution theorem tells us

$$T^D(x,y) = b(x,y) \otimes T(x,y)$$

- Where the point spread function is

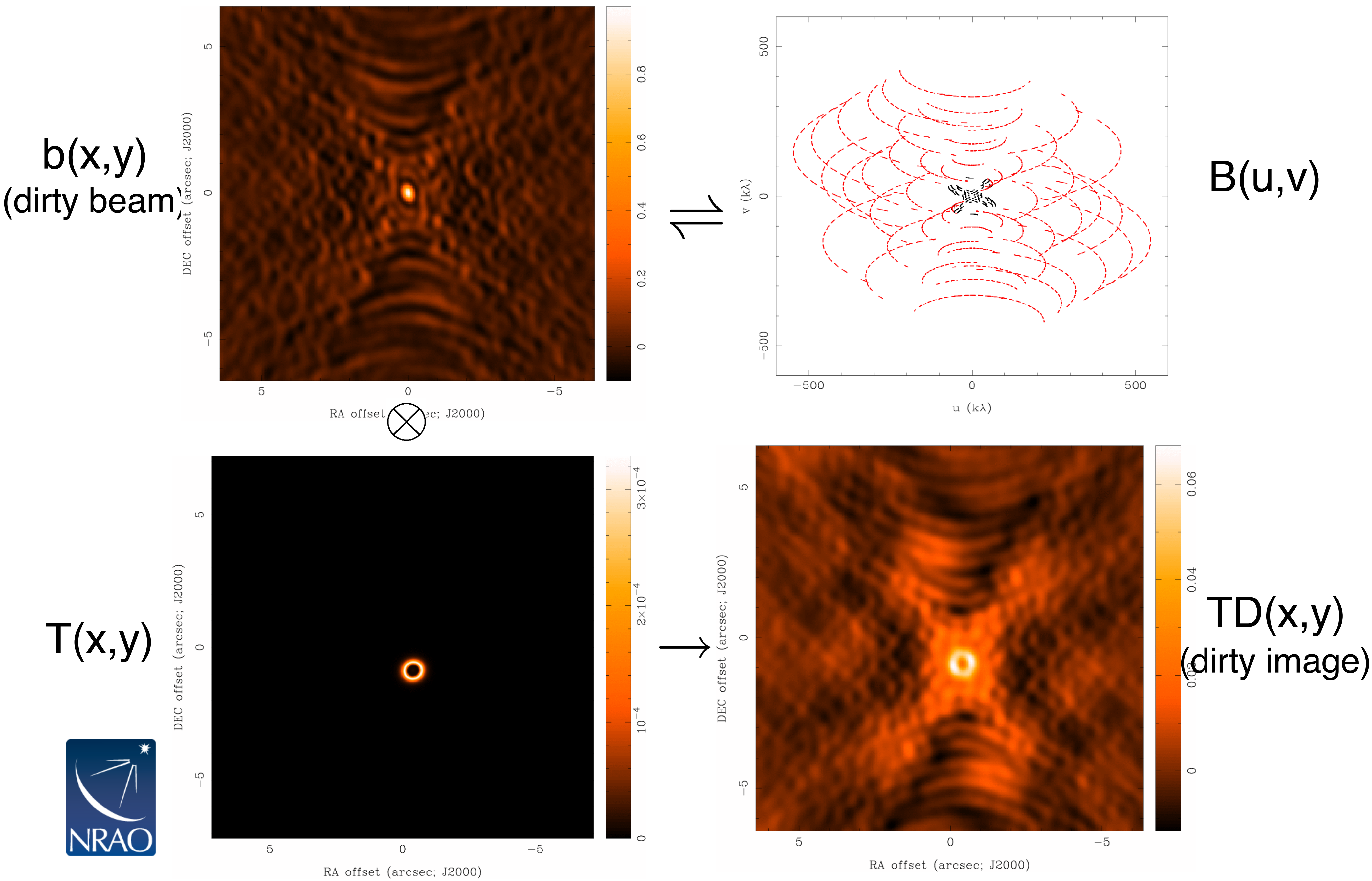
$$b(x,y) = FT^{-1}\{B(u,v)\}$$

- Fourier transform of sampled visibilities yields the true sky brightness convolved with the point spread function ("dirty beam")

- The "dirty image" is the true image convolved with the "dirty beam"

NRAO

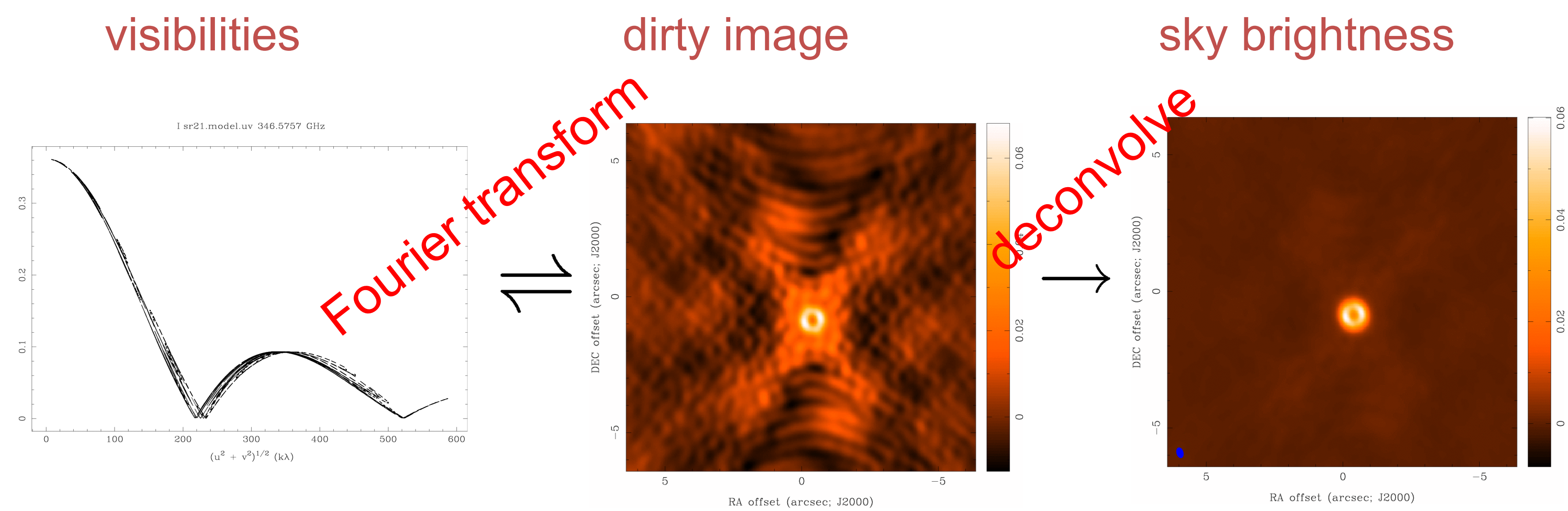
Dirty Beam and Dirty Image



How to analyze (imperfect) interferometer data

Image plane analysis

- dirty image TD(x,y) = Fourier transform { V(u,v) }
- deconvolve b(x,y) from TD(x,y) to determine (model of) T(x,y)

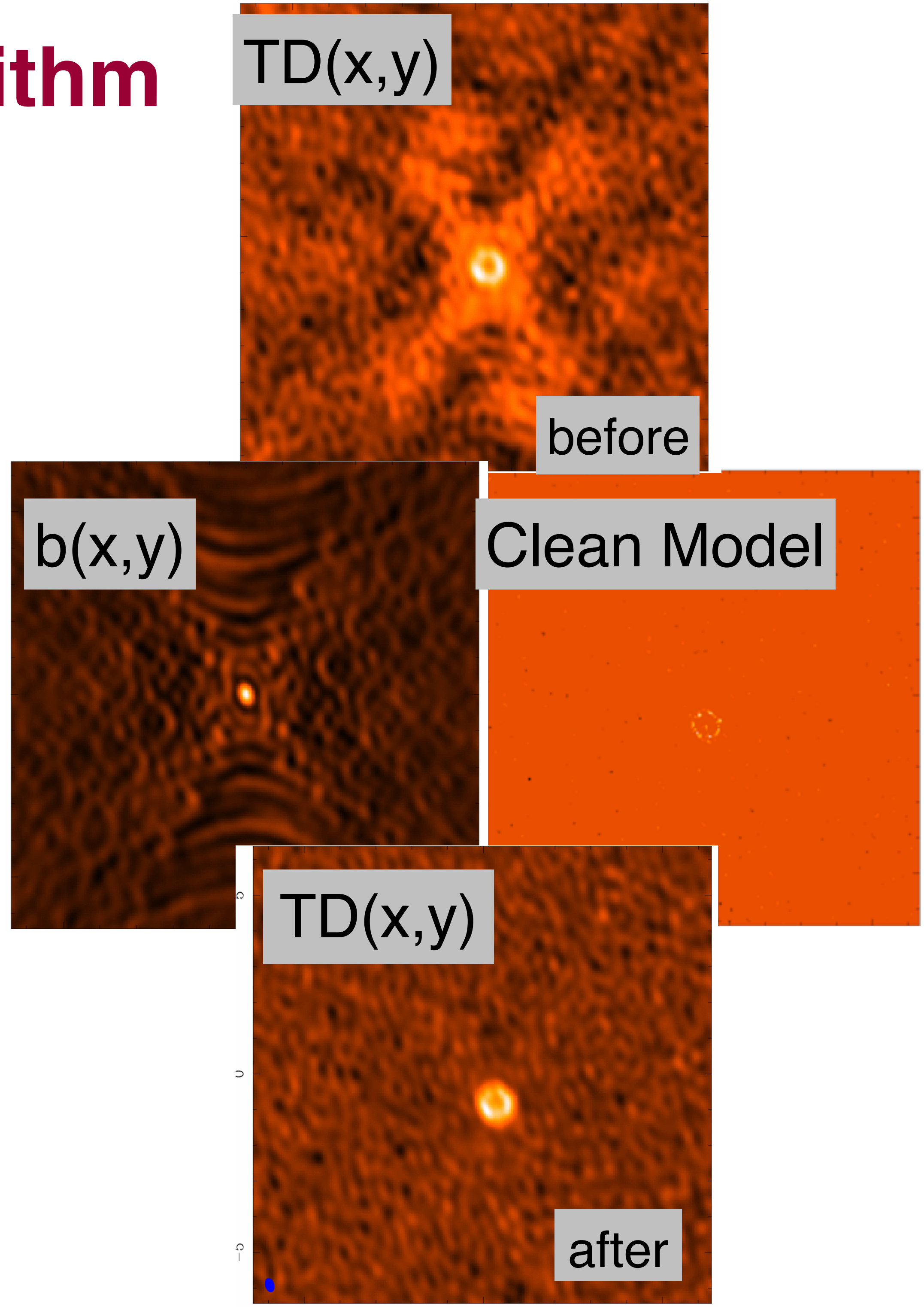


Basic CLEAN Algorithm

A. Initialize a *residual* map to the dirty map

1. Start loop
2. Identify strongest feature in *residual* map as a point source
3. Add this point source to the clean component list
4. Convolve the point source with b(x,y) and subtract a fraction g (the loop gain) of that from *residual* map
5. If stopping criteria not reached, do next iteration

B. Convolve *Clean component* (cc) list by an estimate of the main lobe of the dirty beam (the “Clean beam”) and add *residual* map to make the final “restored” image



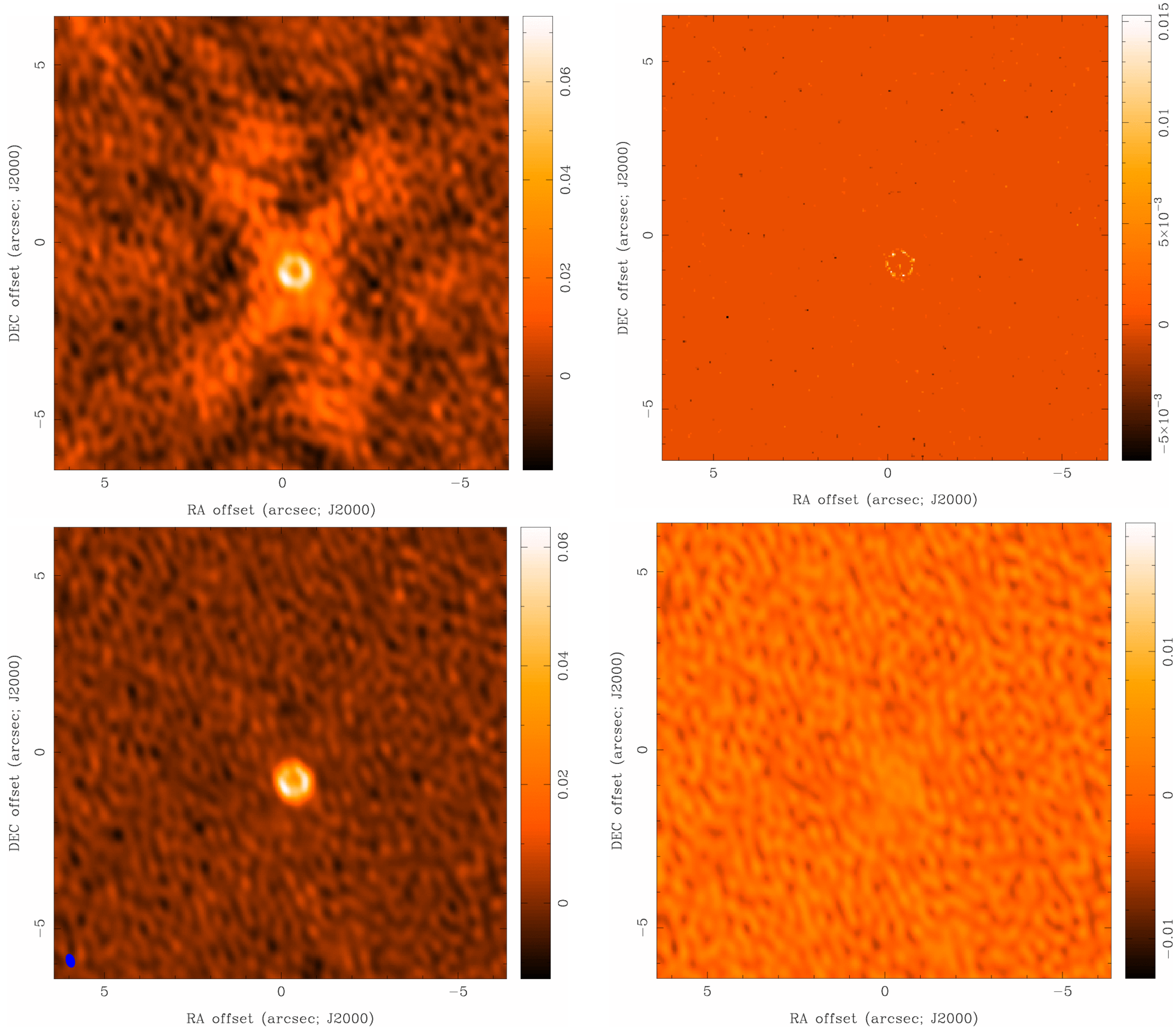
CLEAN

TD(x,y)

CLEAN model

restored image

residual map



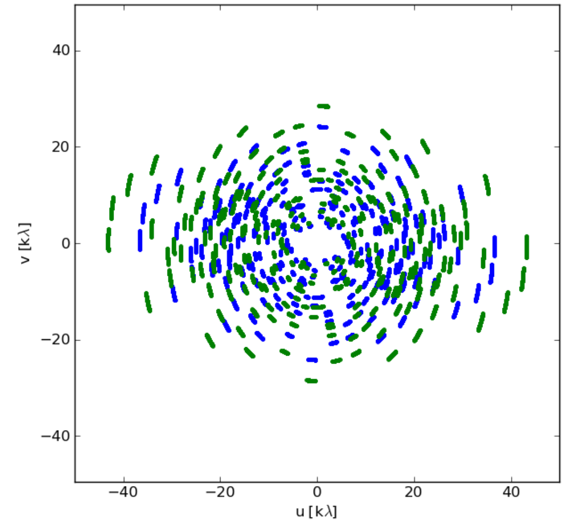
CLEAN in CASA:

```
CASA <3>: inp clean
--------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
vis                 =         ''        #  Name of input visibility file
imagename           =         ''        #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for
                                        #   outliers
field               =         ''        #  Field Name or id
spw                 =         ''        #  Spectral windows e.g. '0~3', '' is all
selectdata          =       True        #  Other data selection parameters
     timerange      =         ''        #  Range of time to select from data
     uvrange        =         ''        #  Select data within uvrange
     antenna        =         ''        #  Select data based on antenna/baseline
     scan           =         ''        #  Scan number range
     observation    =         ''        #  Observation ID range
     intent         =         ''        #  Scan Intent(s)

mode                =      'mfs'        #  Spectral gridding type (mfs, channel, velocity,
                                        #   frequency)
gridmode            =         ''        #  Gridding kernel for FFT-based transforms,
                                        #   default='' None
niter               =        500        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =   '0.0mJy'        #  Flux level to stop cleaning, must include
                                        #   units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor
                                        #   cycles
imagermode          =  'csclean'        #  Options: 'csclean' or 'mosaic', '', uses
                                        #   psfmode
     cyclefactor    =        1.5        #  Controls how often major cycles are done. (e.g.
                                        #   5 for frequently)
     cyclespeedup   =         -1        #  Cycle threshold doubles in this number of
                                        #   iterations

multiscale          =         []        #  Deconvolution scales (pixels); [] = standard
                                        #   clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                =         []        #  Cleanbox(es), mask image(s), region(s), or a
                                        #   level
imsize              = [256, 256]        #  x and y image size in pixels.  Single value:
                                        #   same for both
cell                = ['1.0arcsec']     #  x and y cell size(s). Default unit arcsec.
phasecenter         =         ''        #  Image center: direction or field index
restfreq            =         ''        #  Rest frequency to assign to image (see help)
stokes              =        'I'        #  Stokes params to image (eg I,IV,IQ,IQUV)
weighting           =  'natural'        #  Weighting of uv (natural, uniform, briggs, ...)
uvtaper             =      False        #  Apply additional uv tapering of visibilities
modelimage          =         ''        #  Name of model image(s) to initialize cleaning
restoringbeam       =       ['']        #  Output Gaussian restoring beam for CLEAN image
pbcor               =      False        #  Output primary beam-corrected image
minpb               =        0.2        #  Minimum PB level to use
usescratch          =      False        #  True if to save model visibilities in
                                        #   MODEL_DATA column
allowchunk          =      False        #  Divide large image cubes into channel chunks
                                        #   for deconvolution
```

Dirty Beam Shape and Weighting



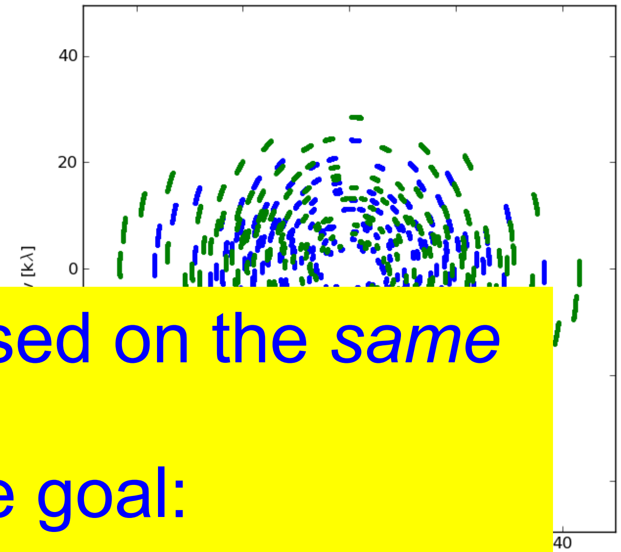
- Each visibility point is given a weight in the imaging step
- First piece: weight given by Tsys, integration time, etc.
- **Natural**
  - Each sample is given the same weight
  - There are many samples at short baselines, so natural weighting will give the largest beam and the best surface brightness sensitivity (and sometimes pronounced wings in the dirty beam)
- **Uniform**
  - each visibility is given a weight inversely proportional to the sample density
  - Weighs down short baselines, long baselines are more pronounced. Best resolution; poorer noise characteristics
- **Briggs (Robust)**
  - A graduated scheme using the parameter *robust;* compromise of noise and resolution
  - In CASA, set *robust* from -2 (~ uniform) to +2 (~ natural)
  - *robust* = 0 often a good choice

- **Taper:** additional weight function to be applied (typically a Gaussian to suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

NRAO

Dirty Beam Shape and Weighting

- Each visibility point is given a weight in the imaging step
- First piece: weight given by Tsys, integration time, etc.
- **Natural**
  - Each sample is given the same weight



It is possible to create different images based on the *same* visibility datasets.
Adjust the weighting to match your science goal:

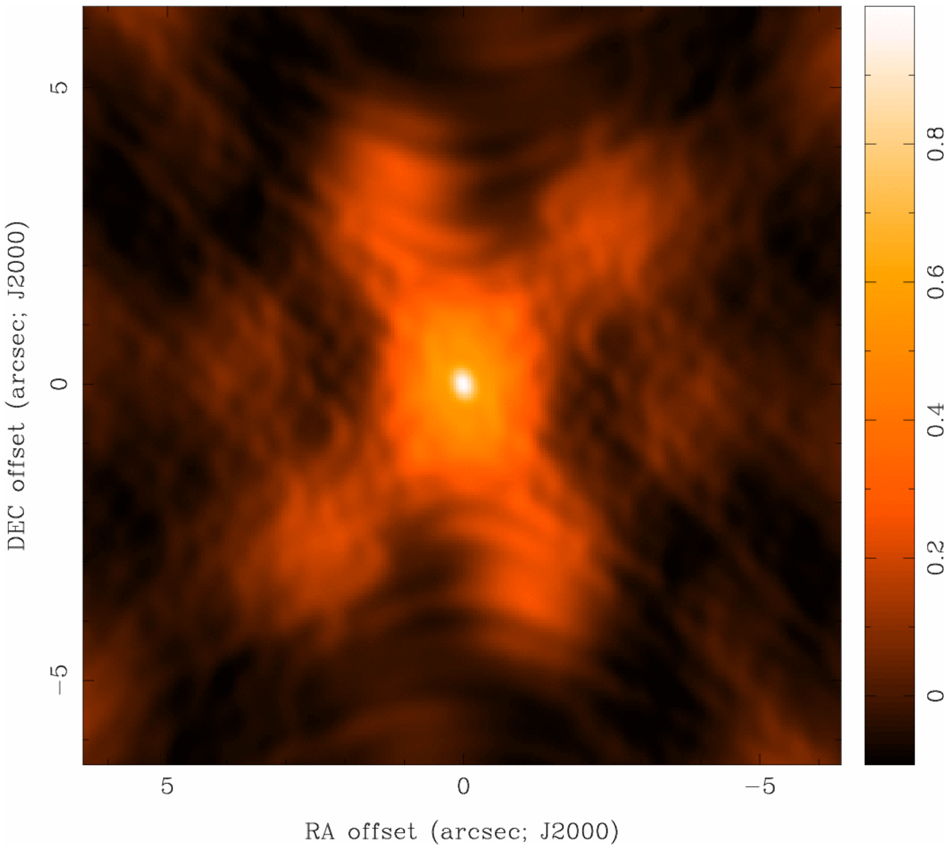
* Detection experiment/weak extended source: **natural** (maybe even with a taper)

* Finer detail of strong sources: **robust** or even **uniform**

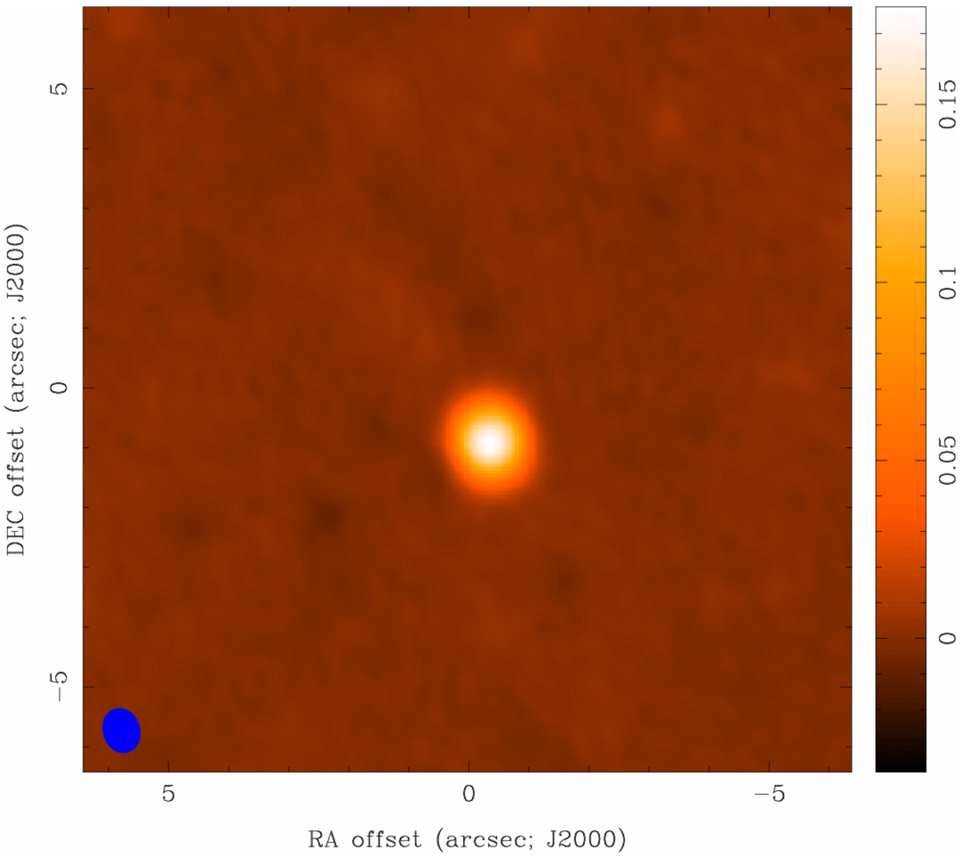
Taper: additional weight function to be applied (typically a Gaussian to suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

Imaging Results

Natural Weight Beam



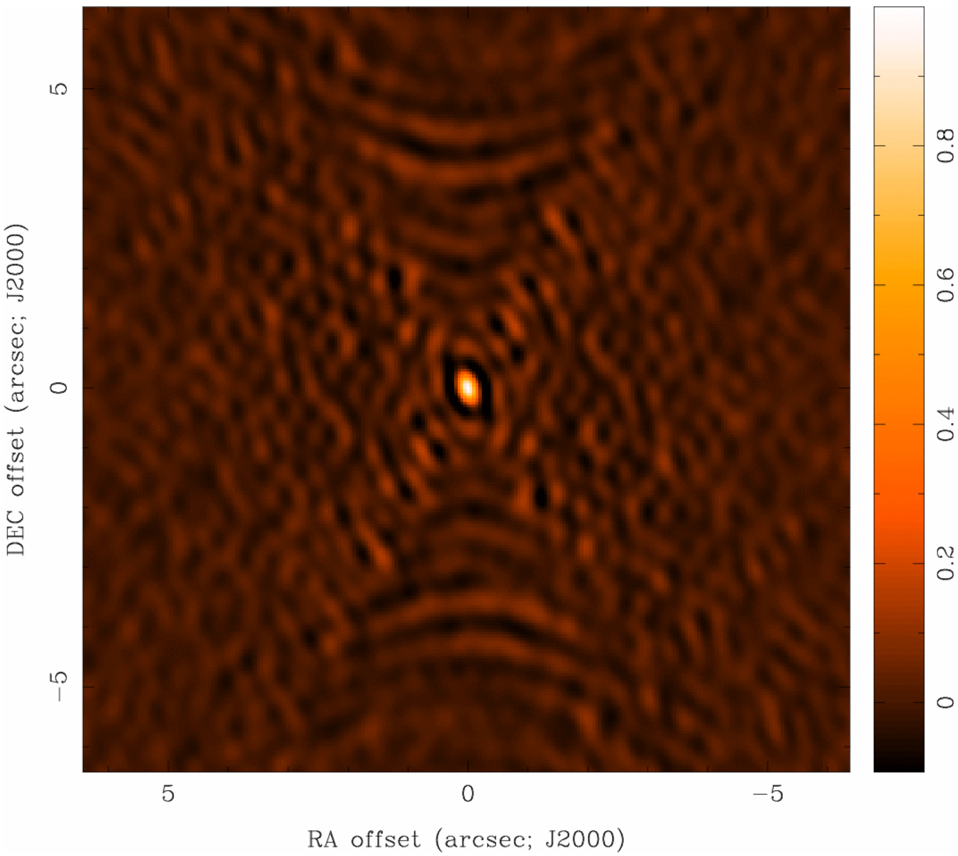
CLEAN image



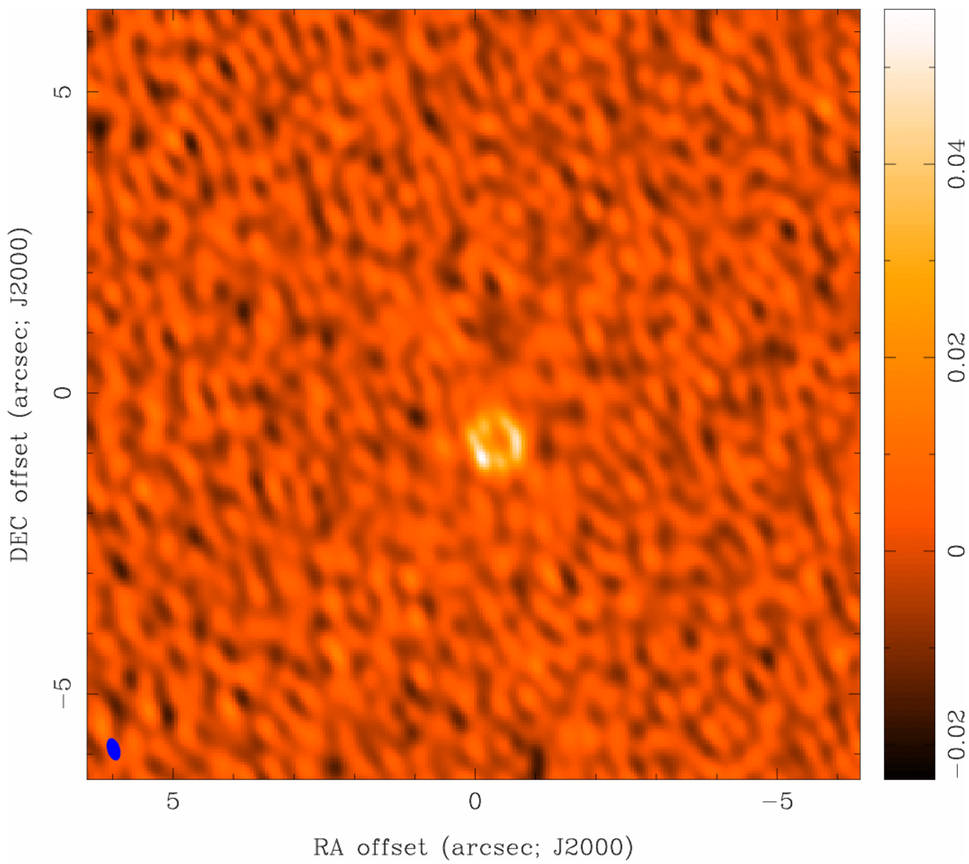
NRAO

Imaging Results

Uniform Weight Beam

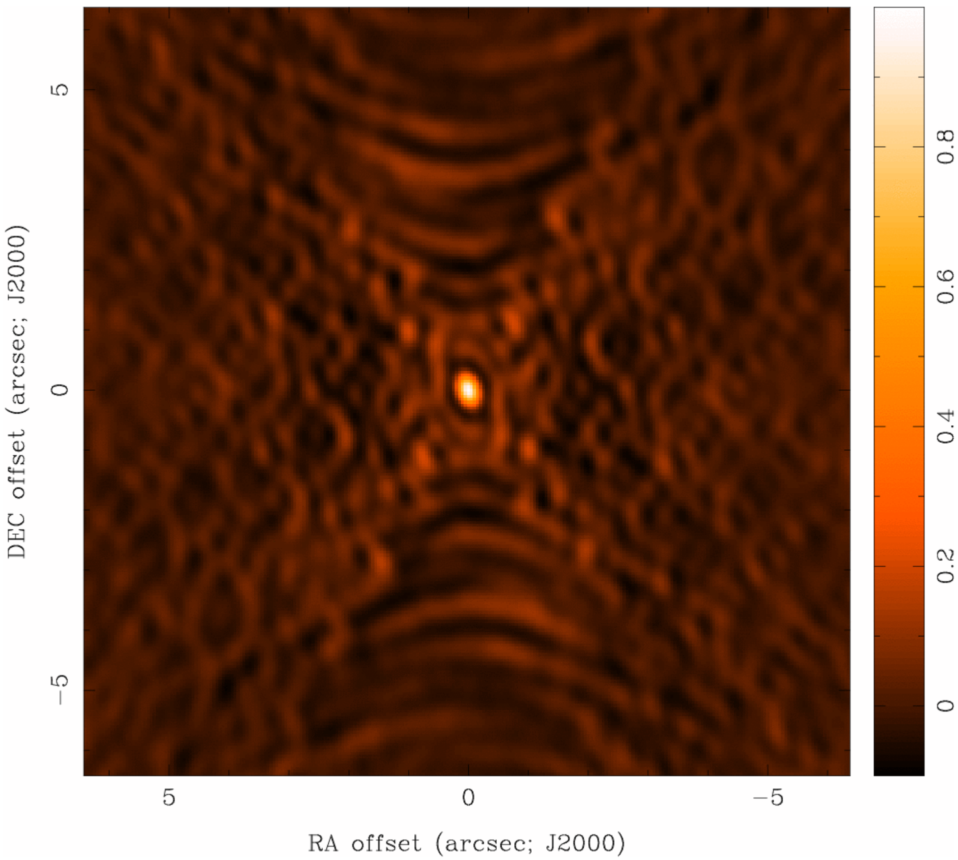


CLEAN image

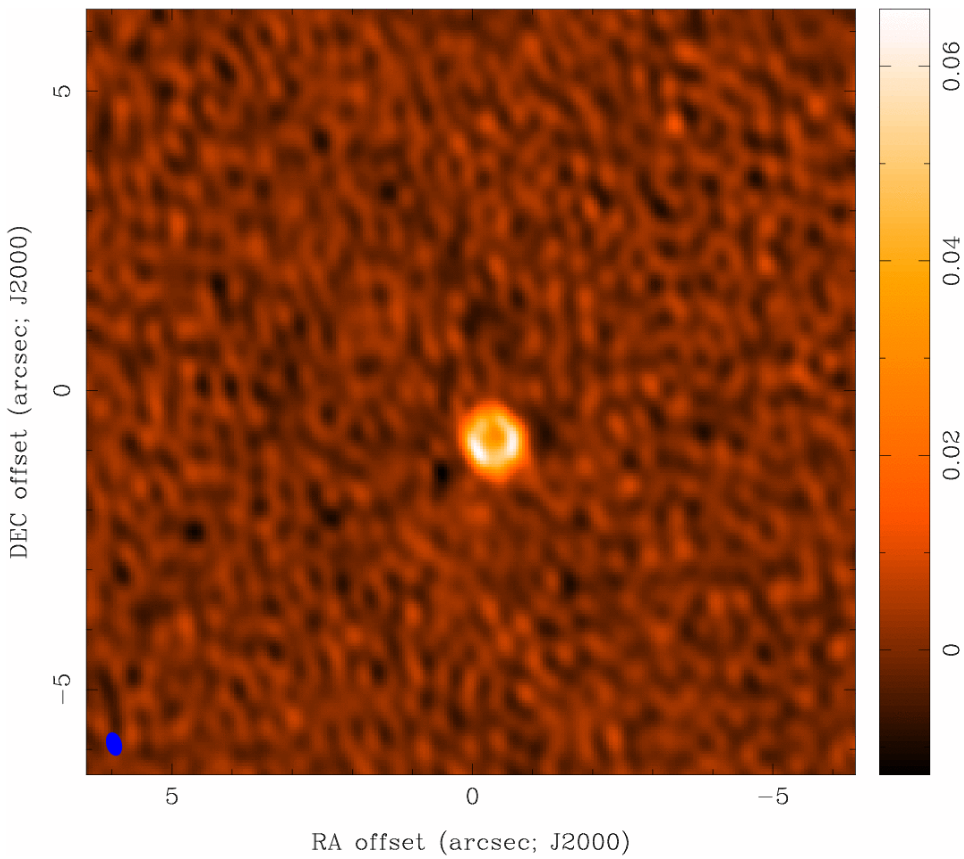


Imaging Results

Robust=0 Beam



CLEAN image



NRAO

Output of CLEAN

Minimally:

- my_image.flux — Relative sky sensitivity
- my_image.image — Cleaned and restored image (Jy/clean beam)
- my_image.mask — Clean "boxes"
- my_image.model — Clean components (Jy/pixel)
- my_image.psf — Dirty beam
- my_image.residual — Residual (Jy/dirty beam)

If CLEAN is started again with same image name, it will try to continue deconvolution from where it left off. Make sure this is what you want. If not, give a new name or remove existing files with rmtables('my_image.*')

Also: try NOT to do CTRL+C as it could corrupt your MS when it touches the visibilities in a major cycle.

Coming in Cycle 5: TCLEAN will replace CLEAN

Why replace clean? Clean has become difficult to maintain, trace down issues, and most importantly to extend to new capabilities, like combining different algorithms together and parallelizing the code efficiently.

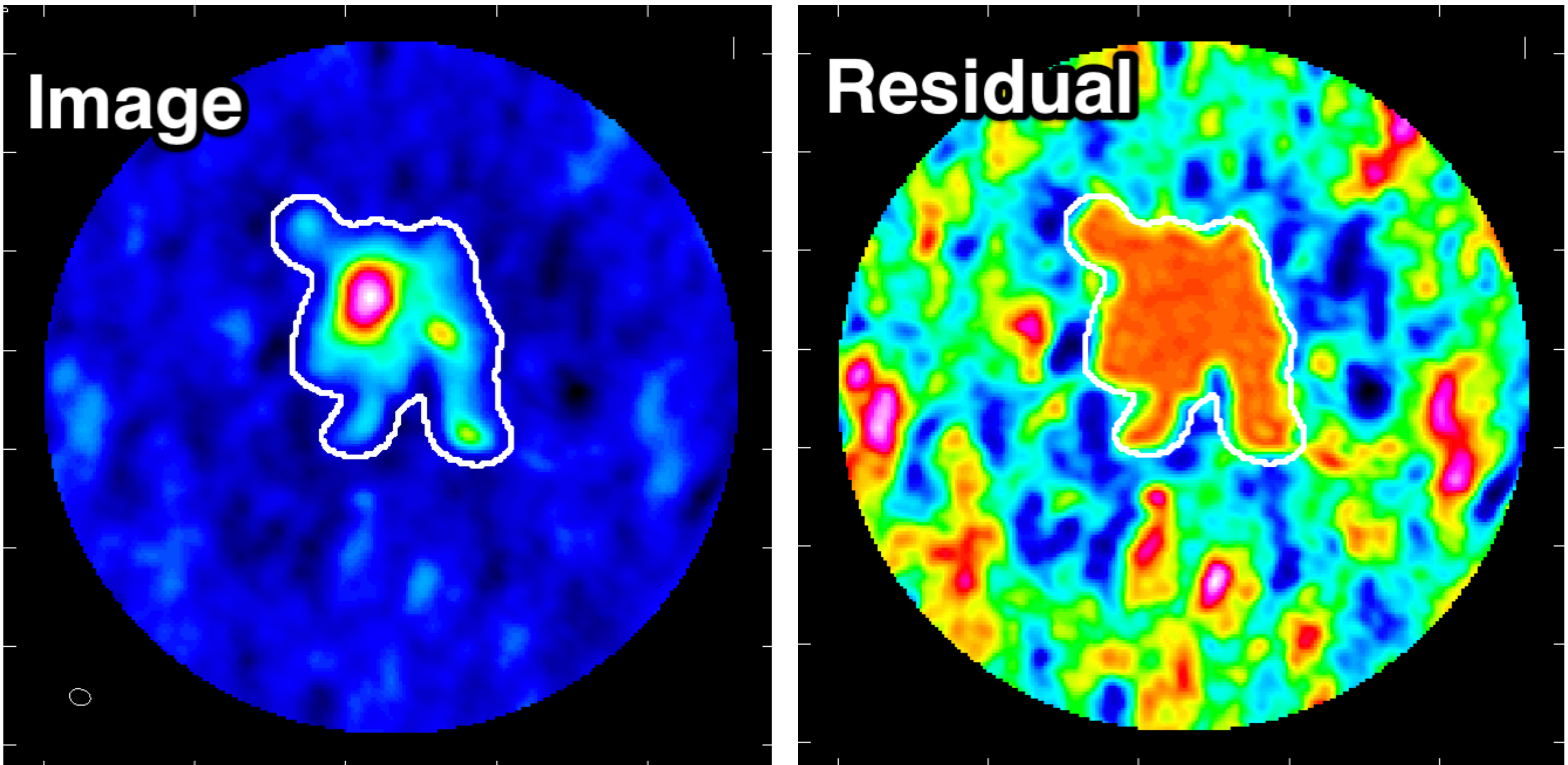
Benefits of TCLEAN:

- A more straightforward interface that is more logical, usable and reliable to the users including significant improvements to the logging output and has been made much more homogeneous across algorithms.
- more combinations of imaging algorithms - most important of these for ALMA is the ability to account for spatial spectral index variations (nterms >1) for mosaics.
- Includes algorithms for autoboxing - refactored code has simplified development of a fully integrated autoboxing capability (available for the first time in CASA 5.0, and for deployment in the pipeline for CASA 5.1, Cycle 5)
- Add significant improvements in divergence checks for safeguarding against divergence in the cleaning process

NRAO

Coming in Cycle 5: TCLEAN

Also included in TCLEAN: Robust Autoboxing algorithms!



Coming in Cycle 5: TCLEAN

Should I use CLEAN or TCLEAN?

As we transition from clean to tclean, we will be updating all the relevant documentation including the CASA Guides, Scripts and Tutorials... Etc.

The ALMA Imaging Pipeline uses tclean and calls to tclean are in all the pipeline “hif_” routines.

So we strongly advise people that tclean should start to be used especially as we are getting closer to Cycle 5. There will be no further development or bug fixes in clean!

Coming in Cycle 5: TCLEAN

TCLEAN also offers a more straightforward user interface inside CASA and clearer logging output

Example Syntax Changes:

clean	tclean
mode	specmode
resmooth=True	*restoringbeam='common'*
minpb	*pblimit*
gridmode, imagermode	gridder
psfmode	deconvolver
modelimage	startmodel
.flux (output)	.pb (output)

Major syntax and usage changes from clean → tclean are summarized here:
https://casaguides.nrao.edu/index.php/TCLEAN_and_ALMA

NRAO

Overview of Clean:

https://casaguides.nrao.edu/index.php/First_Look_at_Imaging

page | discussion | view source | history

log in

- Main Page

search

Search

Go | Search

tools

- What links here
- Related changes
- Special pages
- Printable version
- Permanent link
- Page information

First Look at Imaging

Contents [hide]

About this Guide

The purpose of this tutorial is to provide a first look at imaging ALMA data for those new to CASA.

Data delivered by ALMA is pre-calibrated either by ARC staff or by the ALMA calibration pipeline. The delivered data is ready for imaging. This tutorial demonstrates the basic procedures that will help you complete the imaging steps.

This guide covers the same material used in hands-on training sessions at NRAO Community Days events and ALMA Data Reduction tutorials presented by NAASC staff.

The "Imaging Tutorials for CASA Beginners" guides work for CASA versions 4.2, 4.3, and 4.4, and possibly earlier versions.

Getting CASA

If you do not already have CASA installed on your machine, you will have to download and install it.

Small group work:

- **Suggested timeline:**
  (5 mins) Share your working goals with your teammates.
  (until ~4:20 pm) work individually, discuss with teammates.
  (10 mins) Wrap up, share your achievements with teammates.

Group 1:	Group 2:	Group 3:	Group 4:
Brendan	Andrew	Kristy	Caprice
Laurence	Jackie	Nathan	Peter
Patrick	Meghana	Rachael	Richard
Rebecca	Intae	Sam	Yao-Lun
	Taylor	Sydney	

ALMA simulation tutorial:

https://casaguides.nrao.edu/index.php/Simulating_Observations_in_CASA_4.3

ALMA imaging tutorial:

https://casaguides.nrao.edu/index.php/First_Look_at_Imaging

TCLEAN syntax:

https://casaguides.nrao.edu/index.php/TCLEAN_and_ALMA

For more info:

http://www.almaobservatory.org

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organisation for Astronomical Research in the Southern Hemisphere (ESO), the U.S. National Science Foundation (NSF) and the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Republic of Chile. ALMA is funded by ESO on behalf of its Member States, by NSF in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and by NINS in cooperation with the Academia Sinica (AS) in Taiwan and the Korea Astronomy and Space Science Institute (KASI). ALMA construction and operations are led by ESO on behalf of its Member States; by the National Radio Astronomy Observatory (NRAO), managed by Associated Universities, Inc. (AUI), on behalf of North America; and by the National Astronomical Observatory of Japan (NAOJ) on behalf of East Asia. The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.