Basics of Interferometry

Logan Francis
PhD Student, Uvic
Sept 24th 2020

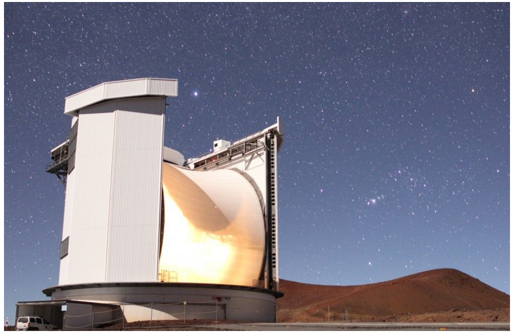
$$\xrightarrow{\mathcal{F}}$$

Outline

- Motivation and Overview
- The two-element interferometer
- Complex Correlators
- The Visibility Function
- Fourier Transform Relationship
- Sampling the visibility in the uv-plane
- Resources

Why observe with an Interferometer?

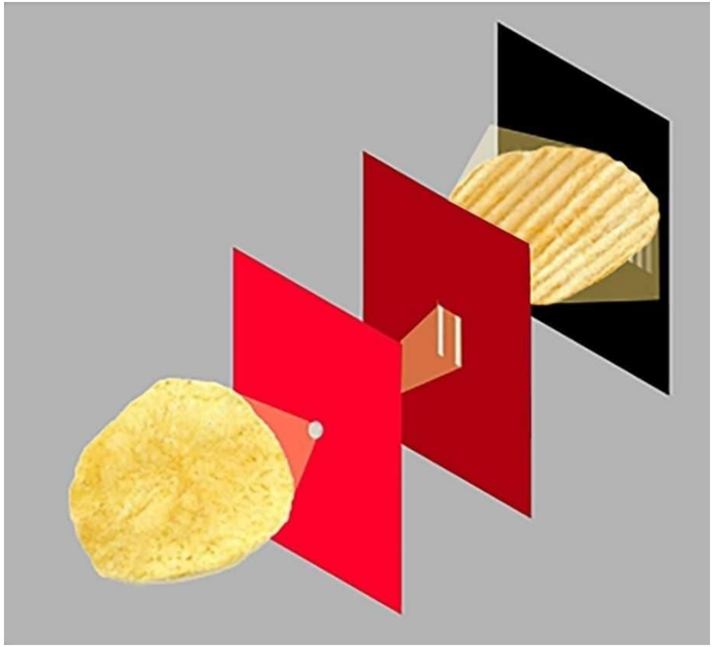
- Obtain **high resolution** at **long wavelengths**.
- Resolution (**θ**) determined by the **separation** between telescopes (**B**) rather than the dish diameter (**D**).
- JCMT @ λ = 850 μm, D = 15m → θ ~ 14”
- ALMA @ λ = 850 μm, B = 2.5km→ θ ~ 0.07”



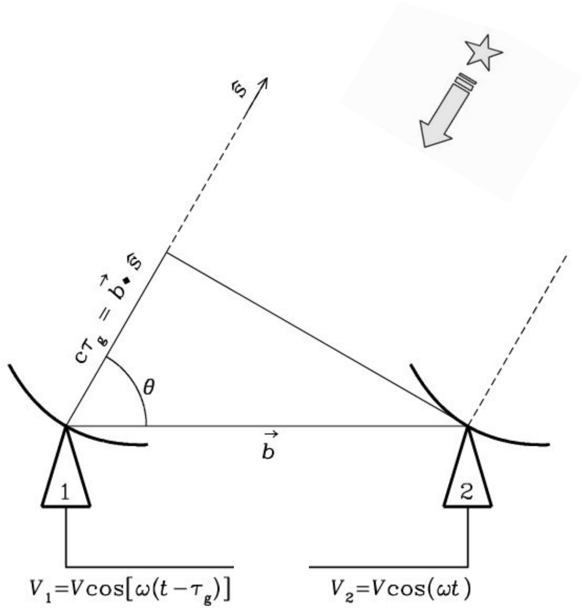
$$\theta = 1.22\frac{\lambda}{D} \Rightarrow \theta = \frac{\lambda}{B}$$

Radio Interferometry Overview:

- Interferometers multiply signals from pairs of antennas to obtain **samples** of a **complex** function called the **Visibility.**
- The Visibility is a **Fourier Transform** of the sky brightness.
- The **samples measured** depend on the antenna separation, orientation, and observing wavelength.
- Imaging require an ***inverse*** Fourier Transform of the samples and mitigation of sampling sparseness (CLEAN lecture)

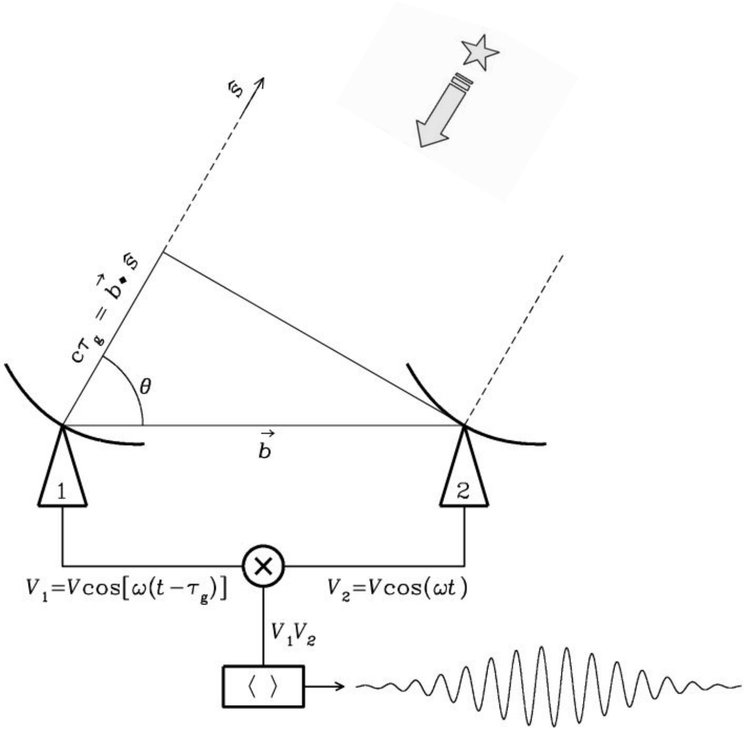


Two-element interferometer + cosine correlator



- Let's imagine a pair of antennas separated by baseline distance **b** observing a source in direction **s** at frequency $\boldsymbol{\nu}$.
- Each antenna collects and amplifies the signal to measure a voltage **V** oscillating at $\boldsymbol{\omega} = 2\pi\nu$.
- The same signal reaches both antennas, one sees a delay τ_g.

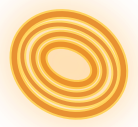
Two-element interferometer + cosine correlator



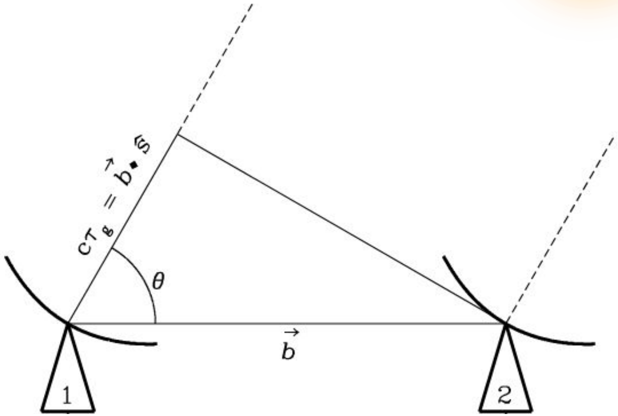
- The signals are digitized, then multiplied and averaged in a **correlator** to produce fringes with response $\mathbf{R_c}$.
- Fringe **amplitude** ∝ Source **brightness.**
- Fringe **phase** is sensitive to source **position**.

$$R_c = \left(\frac{V^2}{2}\right) cos(\omega\tau_g)$$

The complex correlator

$I(\mathbf{s}) =$ 

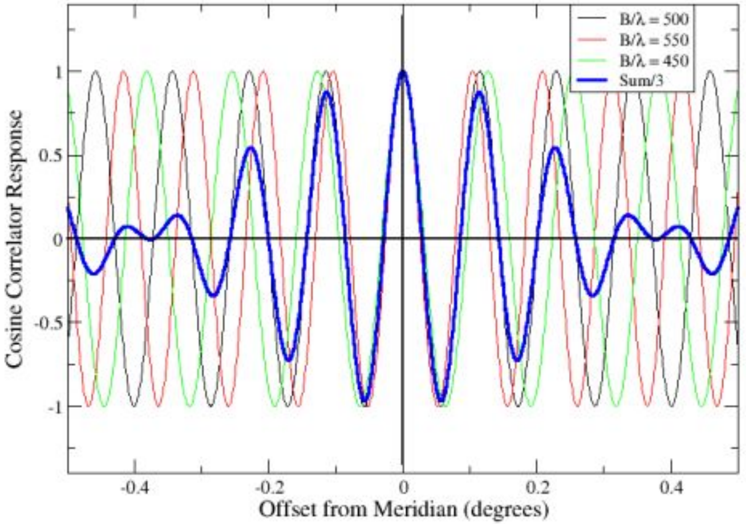
- What if we observe an extended source I(**s**)?
- Cosine correlator is only sensitive to **even** component of extended sources.
- Add 90º delay to make sine correlator and measure **odd** component too.

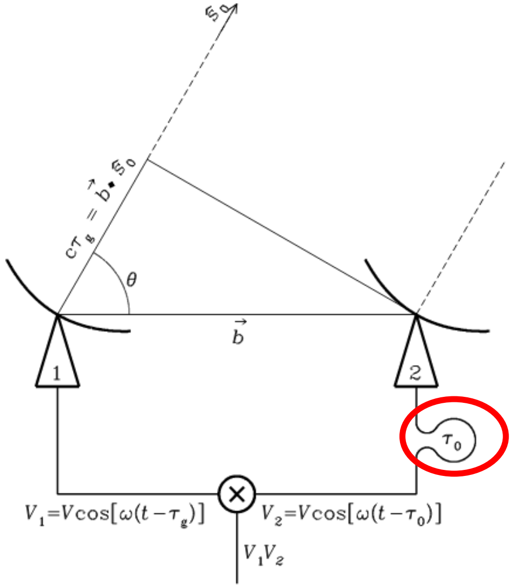


$$R_C = \iint I(\mathbf{s})\cos(2\pi\nu\mathbf{b}\cdot\mathbf{s}/c)\,d\Omega = \iint I_E(\mathbf{s})\cos(2\pi\nu\mathbf{b}\cdot\mathbf{s}/c)\,d\Omega$$

$$R_S = \iint I(\mathbf{s})\sin(2\pi\nu\mathbf{b}\cdot\mathbf{s}/c)\,d\Omega = \iint I_o(\mathrm{s})\sin(2\pi\nu\mathbf{b}\cdot\mathbf{s}/c)\,d\Omega$$

Other considerations - Bandwidth and Fringe Tracking



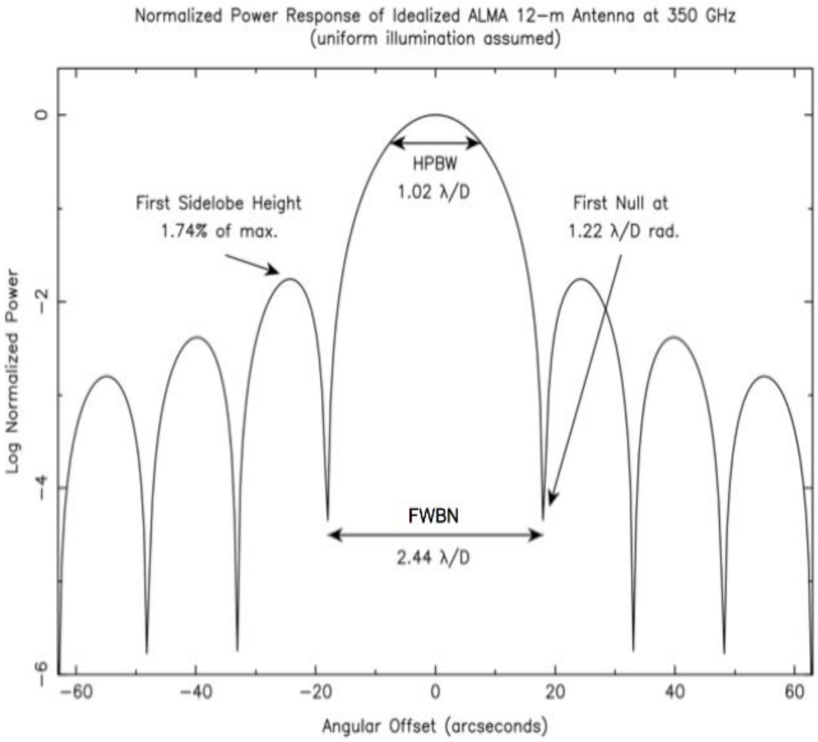


- Finite bandwidth attenuates fringe amplitude away from meridian.
- Earth's rotation moves sources through fringes rapidly.

- Solve both problems by adding extra delay to track a reference "phase center".

Other considerations - Primary Beam

- Parabolic antennas are not uniformly sensitive → emission away from primary beam center attenuated.
- Physically track the motion of the source to compensate.



The Visibility Function

$$V = R_C - iR_S = Ae^{-i\phi}$$

$$A = \sqrt{R_C^2 + R_S^2}$$

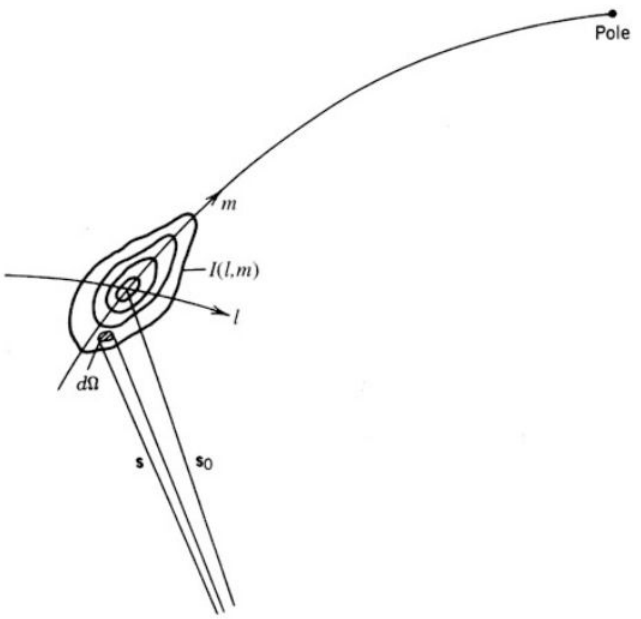
$$\phi = \tan^{-1}\left(\frac{R_S}{R_C}\right)$$

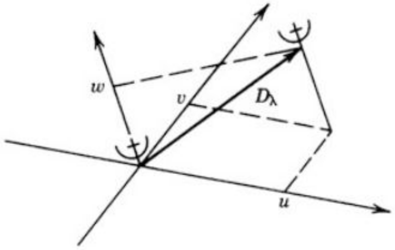
- The **complex Visibility** can now be defined.
- Under certain conditions, the Visibility is a 2D **Fourier Transform** of the sky brightness.

$$V_\upsilon(\mathbf{b}) = R_C - iR_S = \iint I_\nu(s)e^{-2\pi i\nu \mathbf{b}\cdot\mathbf{s}/c}\,d\Omega$$

uv-plane coordinates

- Image plane: **l** and **m** are angular separations on sky East/West and North/South, respectively.
- Visibility (uv) plane: **u** and **v** are **spatial frequencies** East/West and North/South, respectively.
- **u** and **v** are **wavenumbers**, i.e. the length of a baseline in **cycles of the observing wavelength.**





Fourier Transform relationship

- The Fourier transform relationship between the image and uv-plane is now:

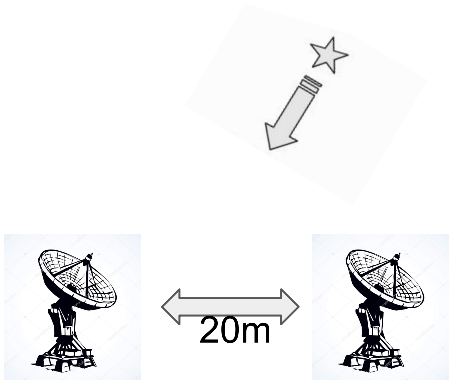
$$V(u,v) = \iint I_\nu(l,m) e^{-2\pi i(ul+vm)} dl dm$$

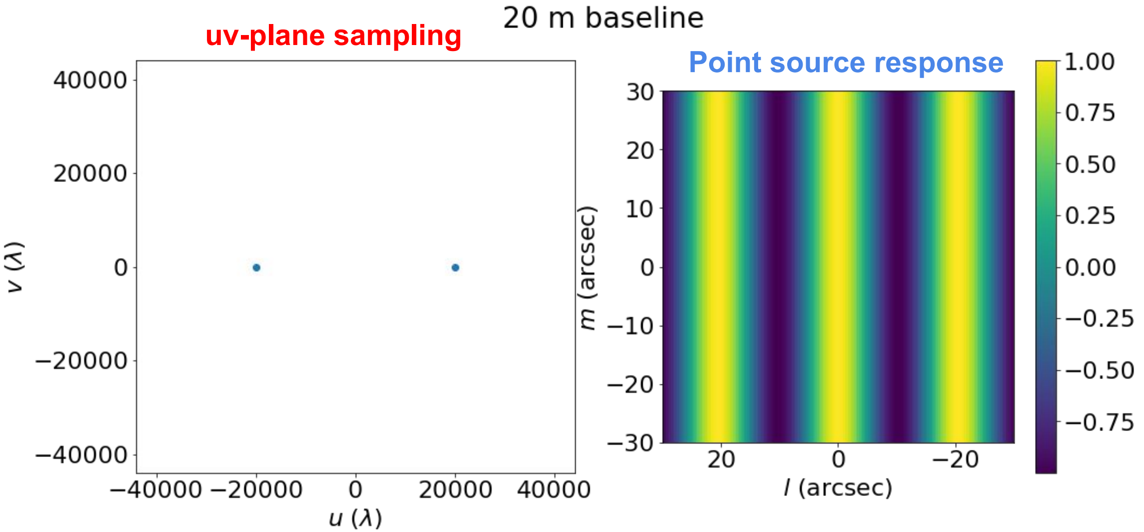
- **Our goal - measure I(l,m) from our samples of V(u,v).**
- With complete sampling of V(u,v) we could simply apply an inverse Fourier transform to recover I(l,m):

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

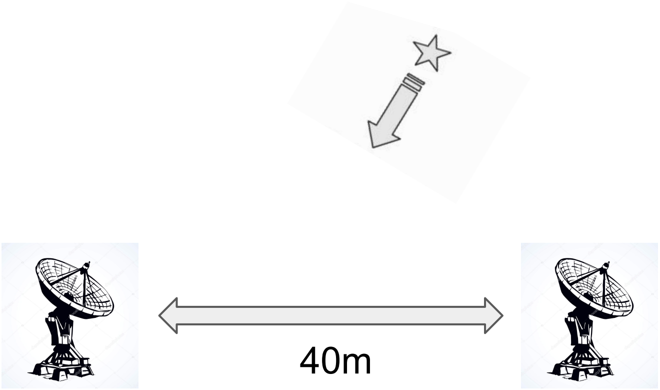
Sampling the visibility.

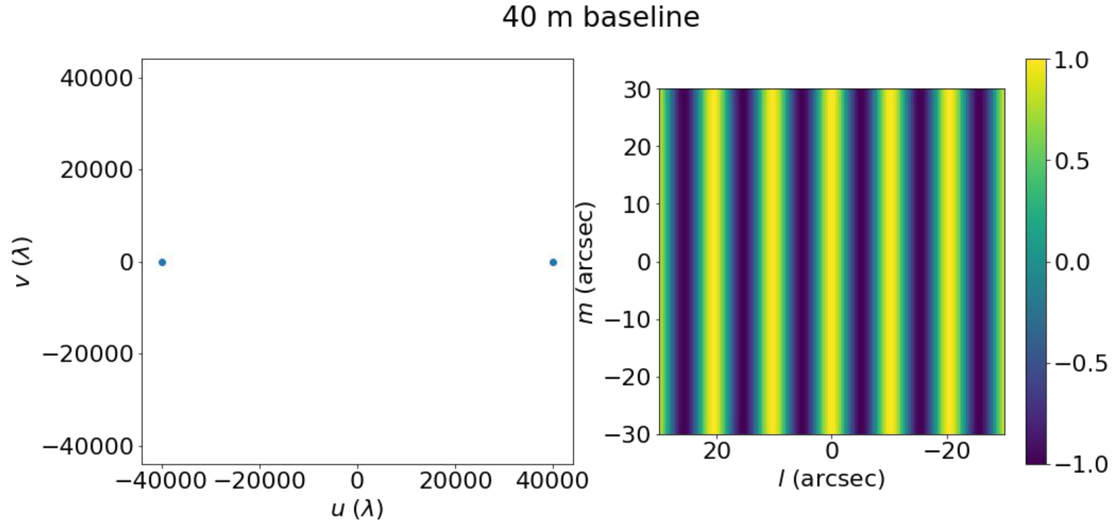
- Each unique baseline vector **b** measures a different **spatial frequency** determined by the **projected** baseline length in units of the observing wavelength.
- Assume we observe a point source at a wavelength of 1 mm with a single baseline:



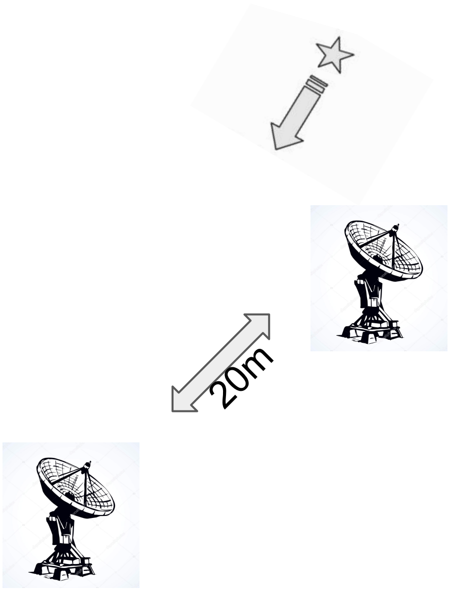


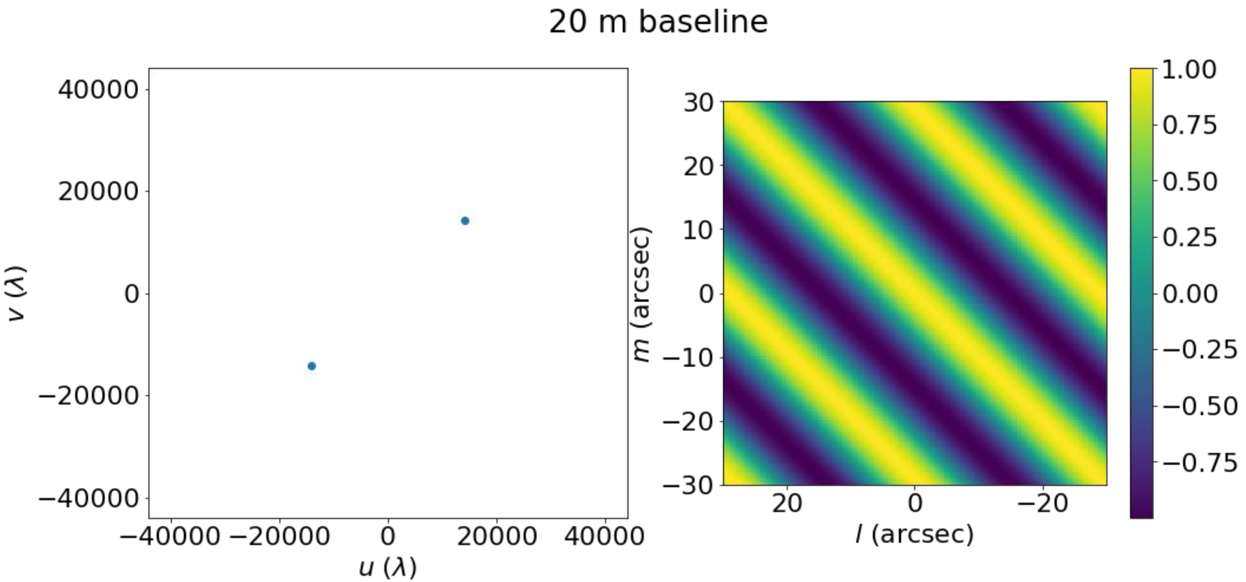
Sampling the visibility.





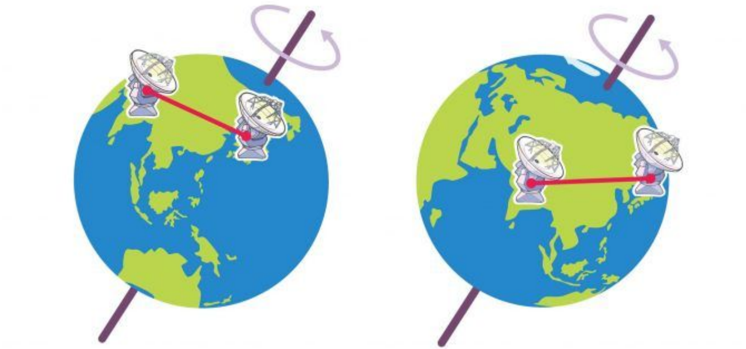
Sampling the visibility.

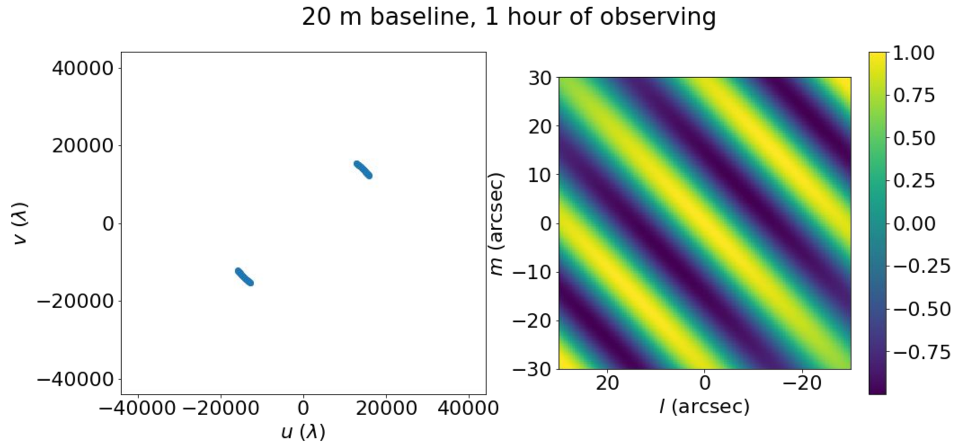




Earth Rotation Aperture Synthesis.

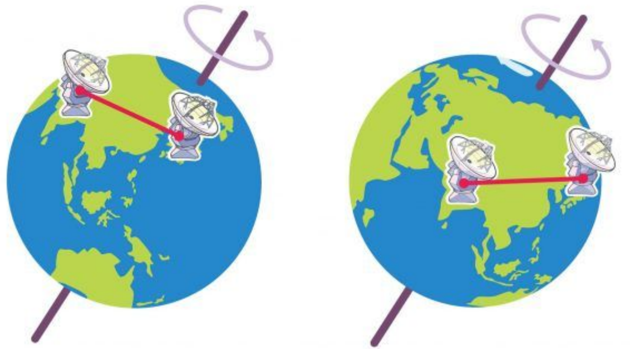
- Baseline projection **changes** as the Earth rotates!
- This allows us to fill in the uv-plane over time.

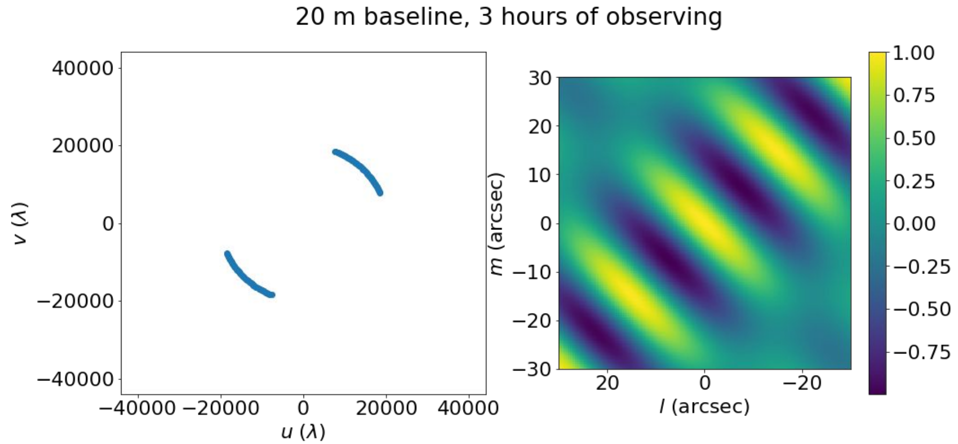




Earth Rotation Aperture Synthesis.

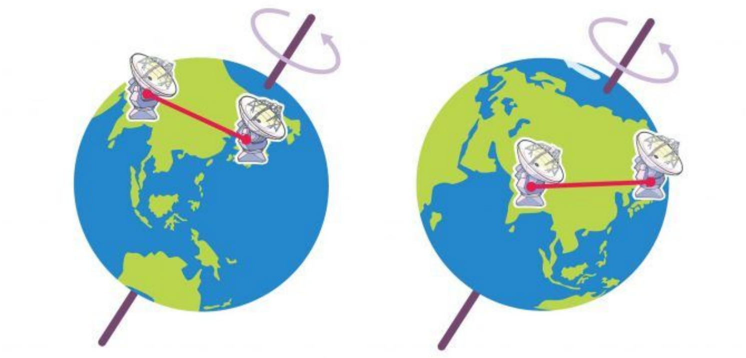
- Baseline projection **changes** as the Earth rotates!
- This allows us to fill in the uv-plane over time.

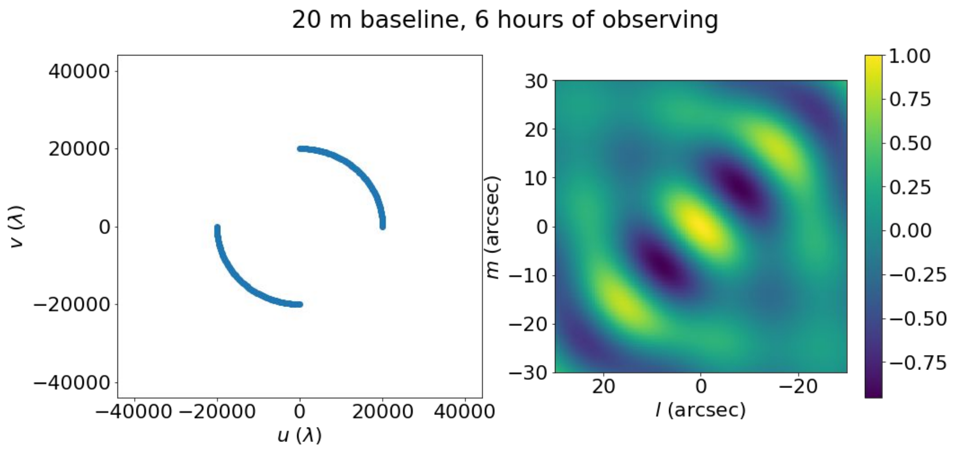




Earth Rotation Aperture Synthesis.

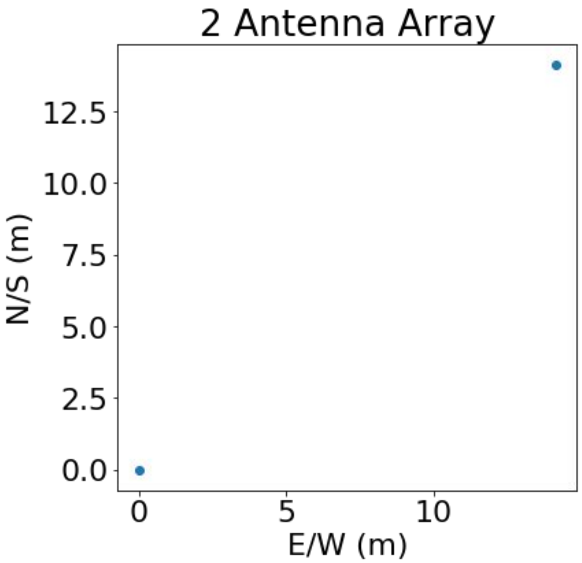
- Baseline projection **changes** as the Earth rotates!
- This allows us to fill in the uv-plane over time.

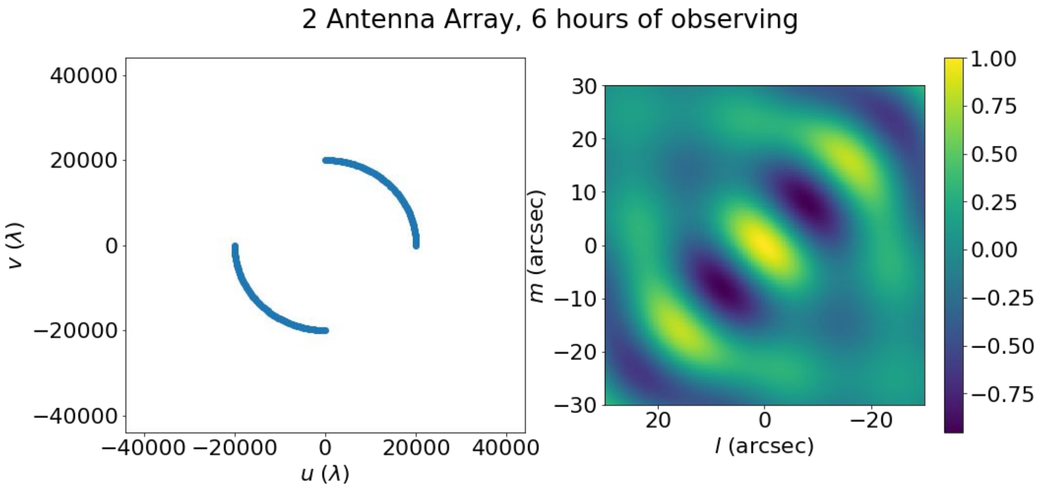




Multi-element arrays

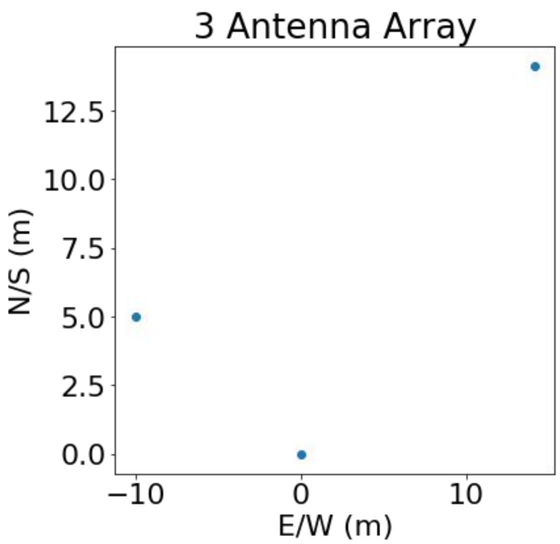
- We also can improve our uv-coverage by adding more antennas with different separations.

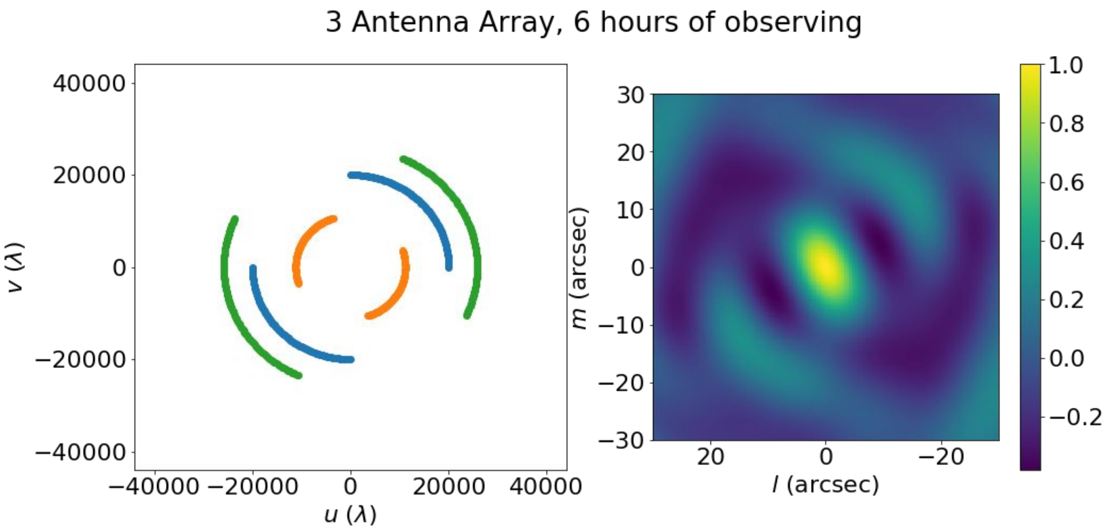




Multi-element arrays

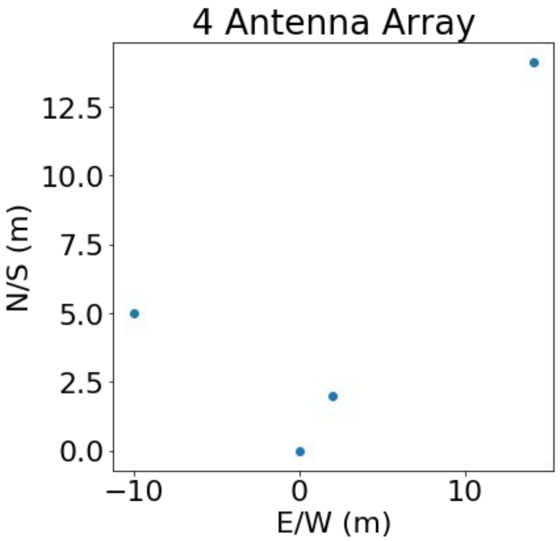
- We also can improve our uv-coverage by adding more antennas with different separations.

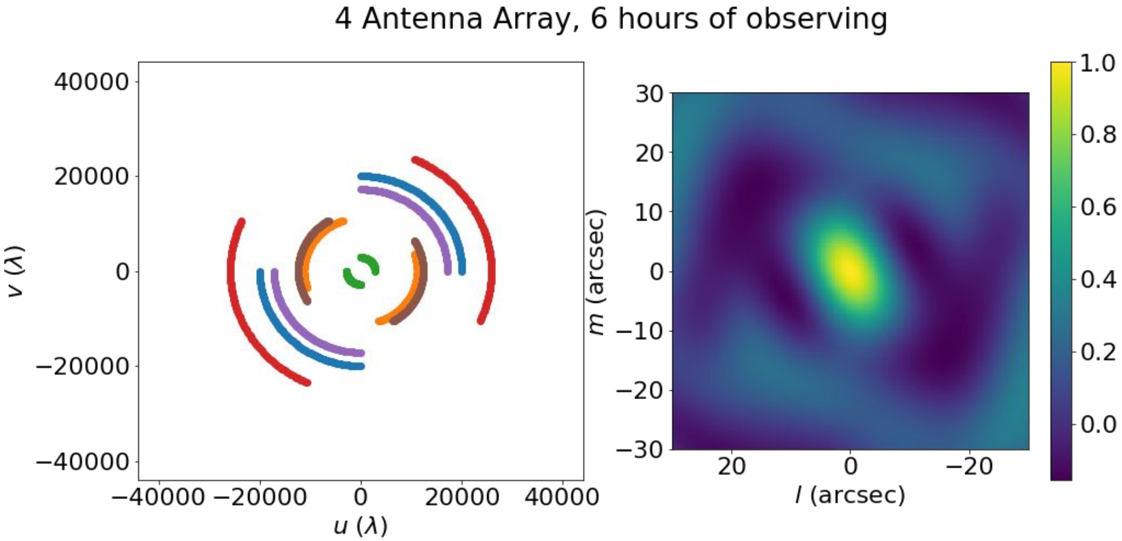




Multi-element arrays

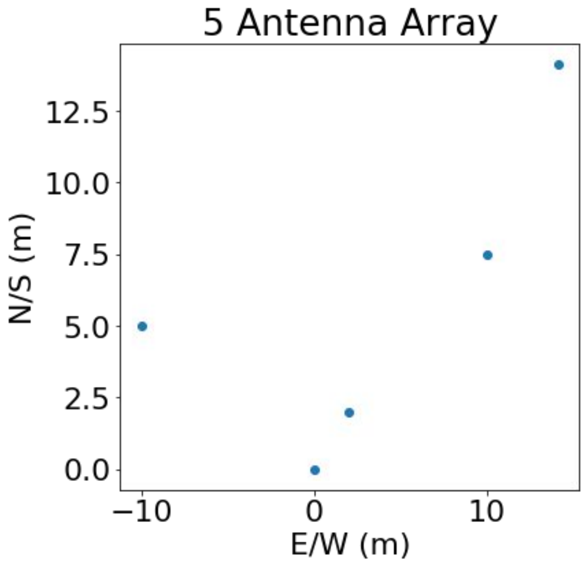
- We also can improve our uv-coverage by adding more antennas with different separations.

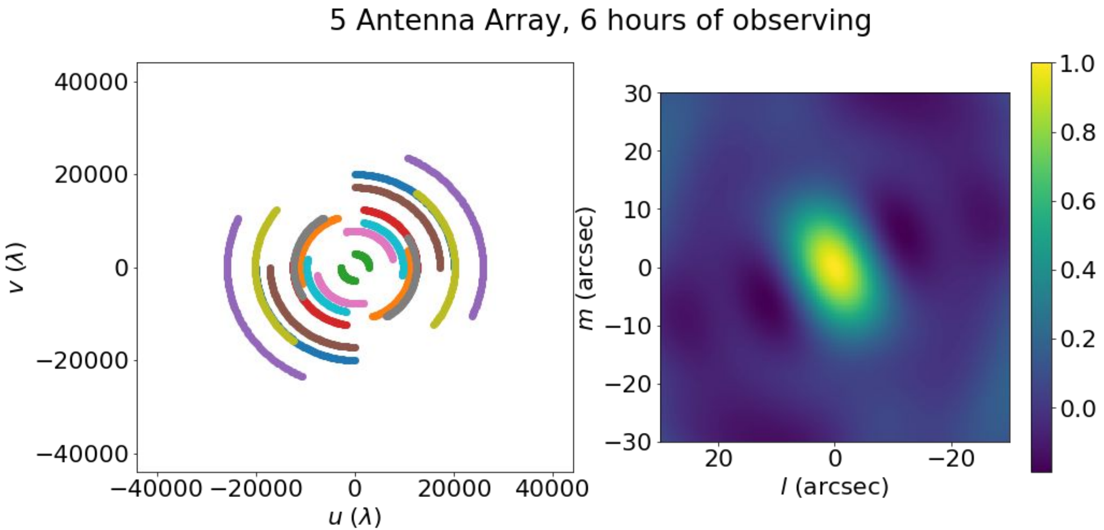




Multi-element arrays

- We also can improve our uv-coverage by adding more antennas with different separations.

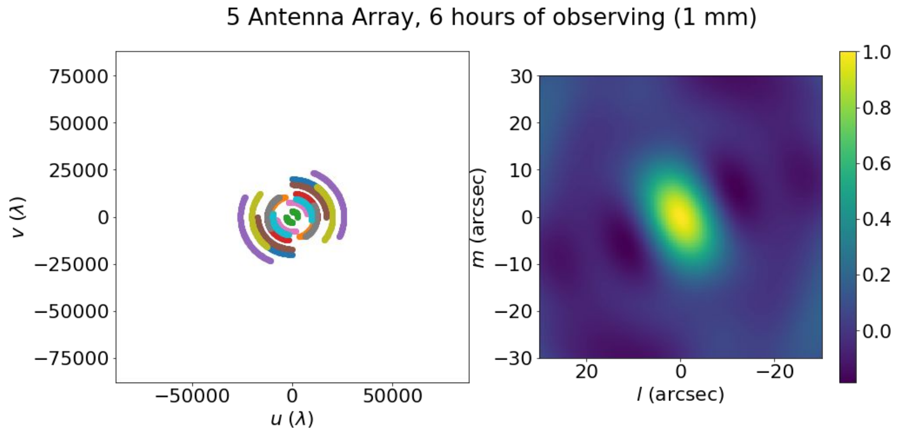


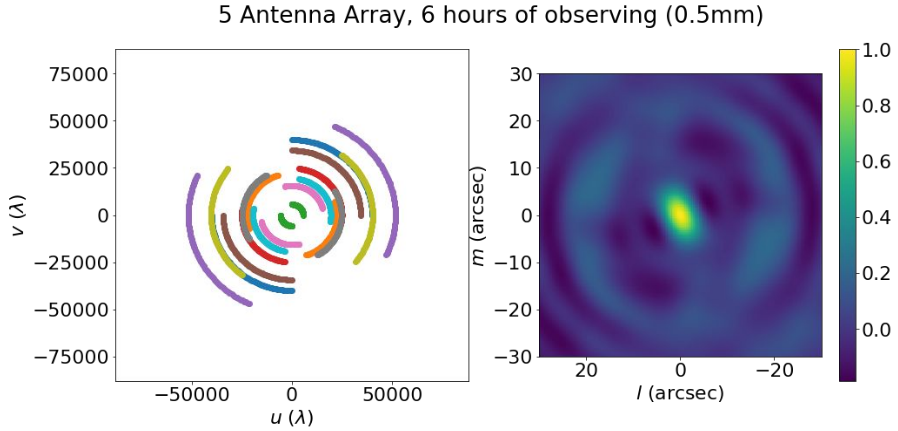


Frequency dependence

- The uv-coverage is also dependent on our observing wavelength.

- Better resolution can be obtained at shorter wavelengths (higher frequencies).





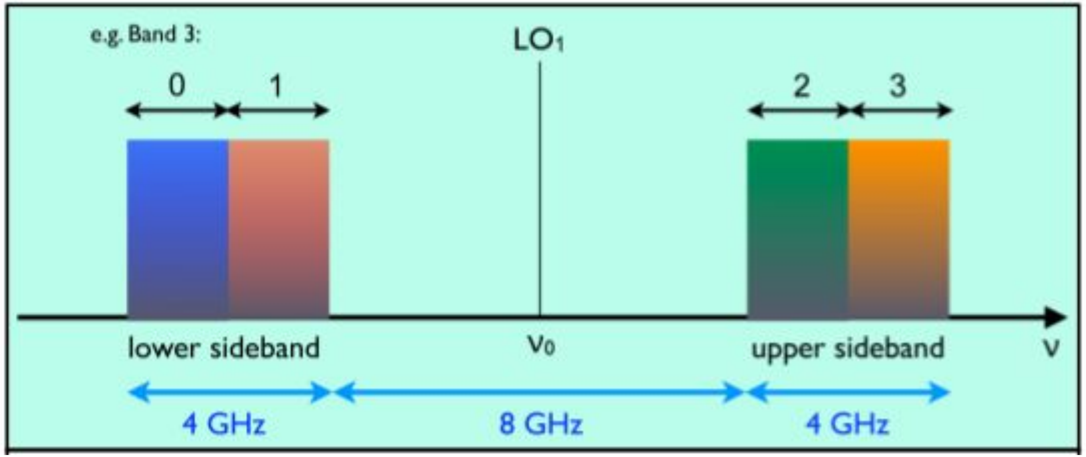
Resources

- Interferometry is a **deep** topic - these will help:
- NRAO Summer School Lectures - perfect for beginners, has recorded lectures and slides: https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/16th-synthesis-imaging-workshop-lectures
- Essential Radio Astronomy - beginner textbook https://www.cv.nrao.edu/~sransom/web/xxx.html
- Interferometry and Synthesis in Radio Astronomy - **extremely technical** but comprehensive textbook: https://link.springer.com/book/10.1007/978-3-319-44431-4
- ALMA Documentation - See the proposer's guide and primer for easy introduction, and the technical handbook for detailed inquiries. https://almascience.eso.org/documents-and-tools/cycle-8-documents

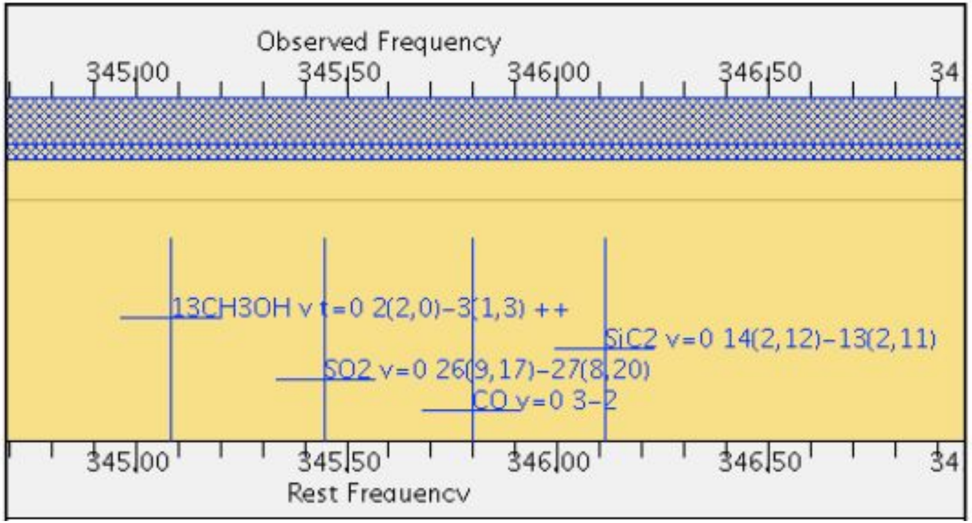
Questions?

Correlator Stuff

- Signals are the ALMA antennas are combined (heterodyning) with a high frequency signal (local oscillator) to downconvert them to a lower (intermediate) frequency for easier processing.
- This produces an upper and lower sideband separated from
- Up to 4 ~2 GHz wide chunks of bandwidth (basebands) from either sideband can be sent to the correlator.
- Correlator resources are allocated to the basebands to divide them into one or more spectral windows with varying bandwidth/resolution.
- Example configurations of correlator.
- Note things are complex and vary between telescopes depending on receiver/correlator setup.



e.g. Band 3:
LO1
0
1
2
3
lower sideband
ν0
upper sideband
ν
4 GHz
8 GHz
4 GHz



Observed Frequency
345.00
345.50
346.00
346.50
13CH3OH v t=0 2(2,0)-3(1,3) ++
SiC2 v=0 14(2,12)-13(2,11)
SO2 v=0 26(9,17)-27(8,20)
CO v=0 3-2
Rest Frequency

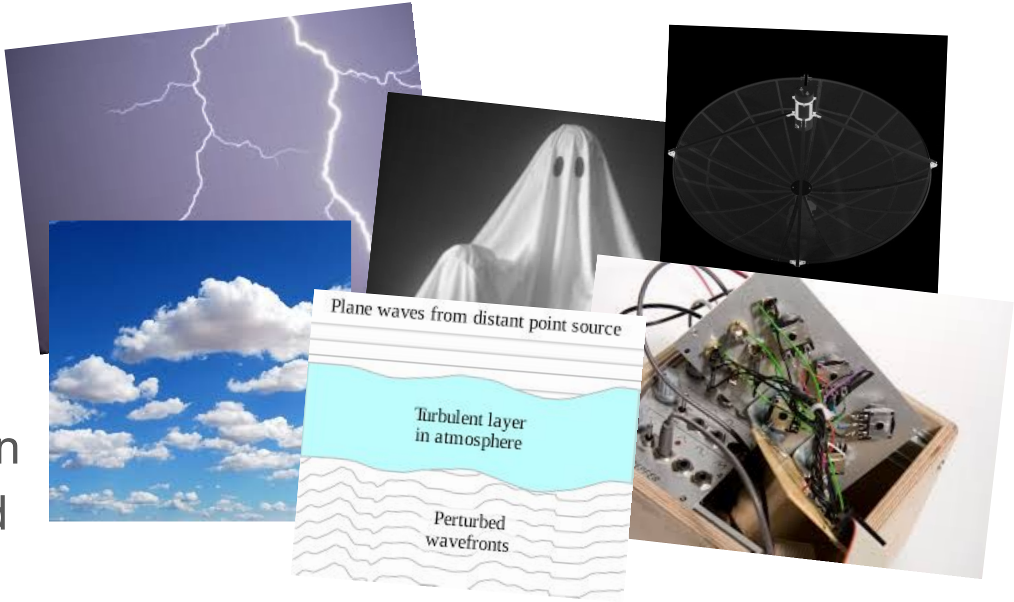
ALMA Calibration

Logan Francis
PhD Student, Uvic
Sept 24th 2020

Goal of Calibration

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

- Interferometers try to measure the visibility function…
- … but the atmosphere, wet weather, antenna errors, electronics problems, and other issues corrupt our measurements.
- Through calibration we want to eliminate these effects and obtain accurate visibility amplitudes and phases.



Antenna Calibration equation

- The measured visibility V_m on baseline between antennas i, j is corrupted by a **complex gain** factor which varies with time t_k and frequency v_f.

$$\mathcal{V}_o^{i,j}(t_k, \nu_f) = \mathcal{V}_m^{i,j}(t_k, \nu_f) * G^{i,j}(t_k, \nu_f)$$

- Amplitude and phase errors can be factored into **antenna-based** complex gains:

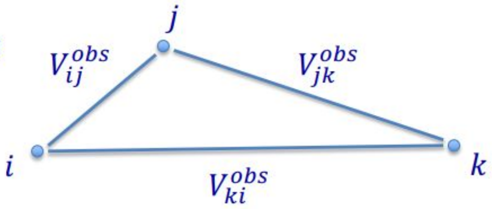
$$
\begin{aligned}
G^{i,j}(t_k, \nu_f) = G^i G^j = & \\
A^i(t_k, \nu_0) * A^j(t_k, \nu_0) & \text{ Temporal amplitude} \\
+A^i(t_0, \nu_f) * A^j(t_0, \nu_f) & \text{ Bandpass amplitude} \\
\phi^i(t_k, \nu_0) - \phi^j(t_k, \nu_0) & \text{ Temporal phase} \\
+\phi^i(t_0, \nu_f) - \phi^j(t_0, \nu_f) & \text{ Bandpass phase} \\
+\Delta G^{i,j}(t_k, \nu_f) & \text{ Any additional correction}
\end{aligned}
$$

Closure Relationships

$$G^{i,j} = G^i G^j = \mathcal{V}_m / \mathcal{V}_{\text{cal}}$$

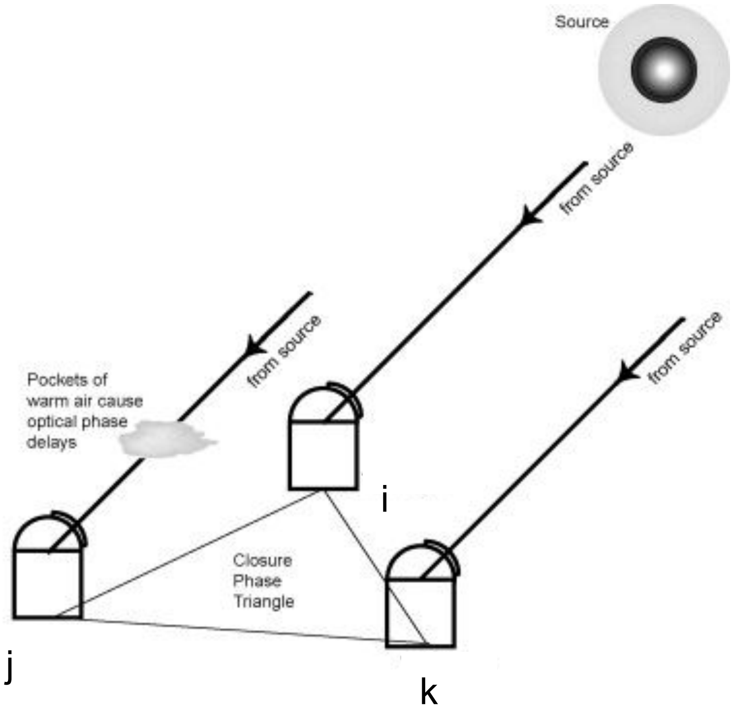
- Goal: recover the antenna complex gains by comparison of the measured visibilities with a known calibrator source:
- Useful closure phase (and amplitude relationships provide constraints for solving for antenna gains, e.g:

- Form total phase around three baselines:



$$\phi_{ij}^{obs} + \phi_{jk}^{obs} + \phi_{ki}^{obs} = (\phi_{ij}^{true} + \theta_i - \theta_j) + (\phi_{ij}^{true} + \theta_i - \theta_j) + (\phi_{ij}^{true} + \theta_i - \theta_j)$$

$$= \phi_{ij}^{true} + \phi_{jk}^{true} + \phi_{ki}^{true}$$



Observatory Calibrations

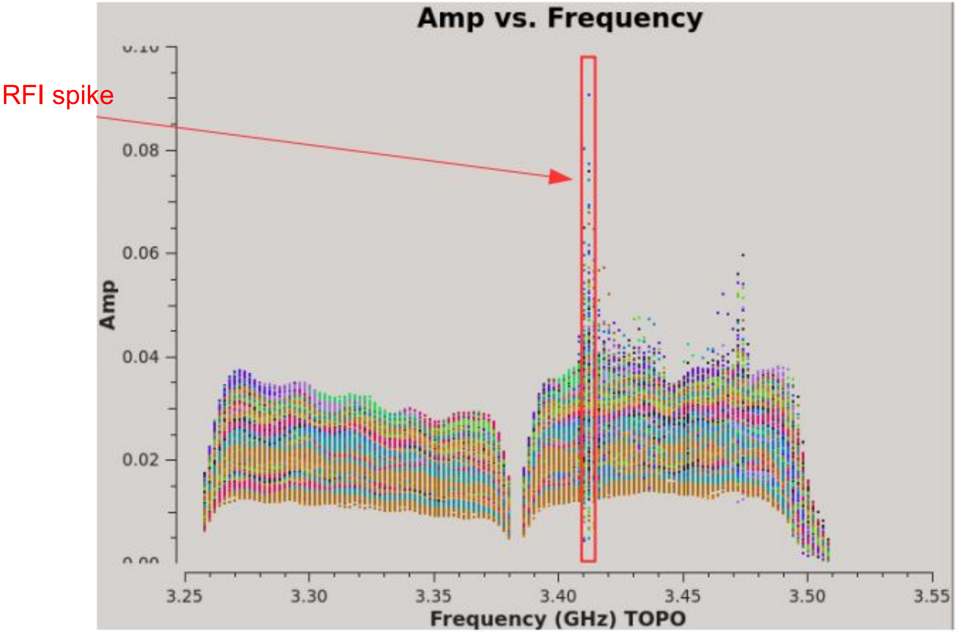
- These are done rarely, or when antennas are moved.
- For each antenna, calibration measurements are made for:
  - Array Position
  - Focus
  - Pointing/astrometry
  - Surface accuracy
  - Primary beam pattern

Science Observation Calibrations

- The following calibrations are done for every ALMA observation:
  - Data Flagging
  - System temperature Measurement
  - Water Vapor Radiometer Corrections
  - Bandpass Calibration
  - Flux Density Scale Calibration
  - Phase (gain) Calibration
  - Check source observations (long-baseline/high frequency only)
  - Polarization Calibrations (polarization observations only)
  - Self-Calibration (ALMA user applied! Friday Workshop :))
  -
- We'll go through (most of them) one at a time.

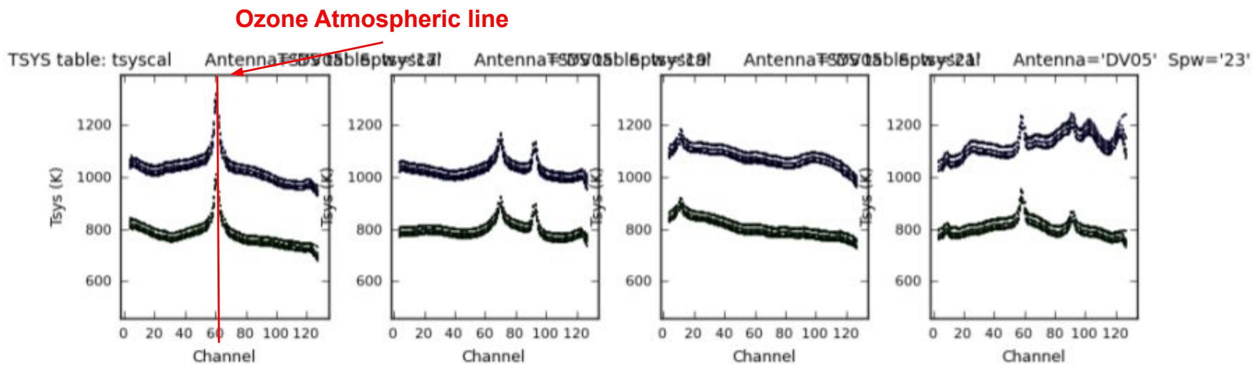
Data Flagging

- Some problems can completely ruin data, e.g:
  - Broken antennas
  - Correlator glitches
  - Telescope shadowing
  - Radio Frequency Interference
  - Bad calibration solutions
- When this happens, the bad data is removed or “flagged”.



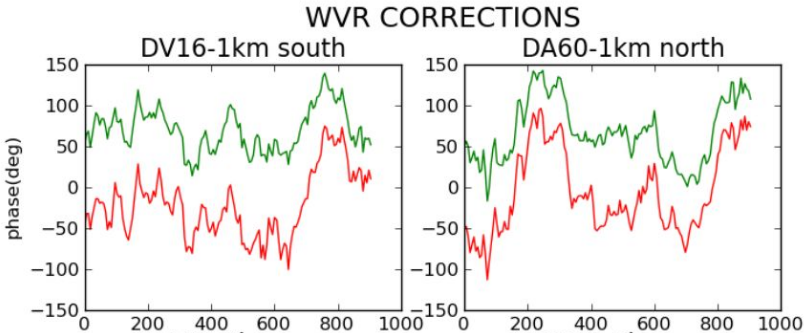
System Temperature Calibration

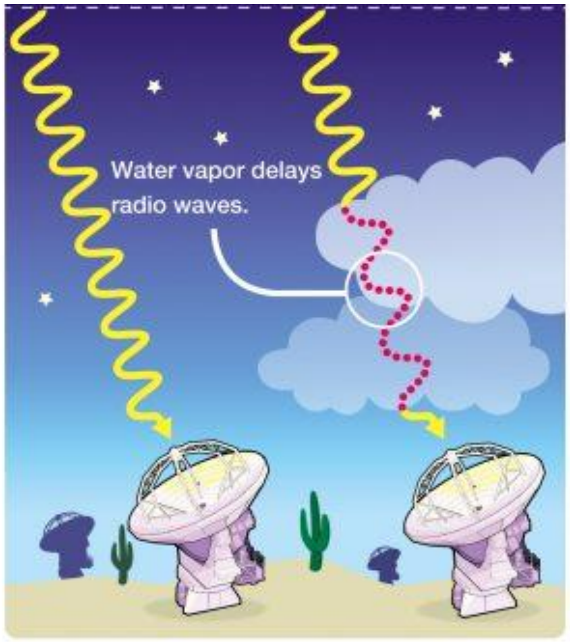
- System temperature = noise added to observations.
- $T_{sys} = T_{bg} + T_{sky} + T_{spill} + T_{loss} + T_{cal} + T_{rx}$ *, where:*
  - T_{bg} = noise contribution from microwave and galactic backgrounds
  - T_{sky} = noise contribution from atmospheric emission
  - T_{spill} = noise contribution due to ground radiation (spillover and scattering)
  - T_{loss} = noise contribution due to losses in feed
  - T_{cal} = noise contribution due to injected noise
  - T_{rx} = receiver noise temperature
- Need to measure T_{sys} **to get correct data weights!**
- Measure calibrator hot/cold “loads” (resistors) and sky contribution.



Water Vapor Radiometer Calibration

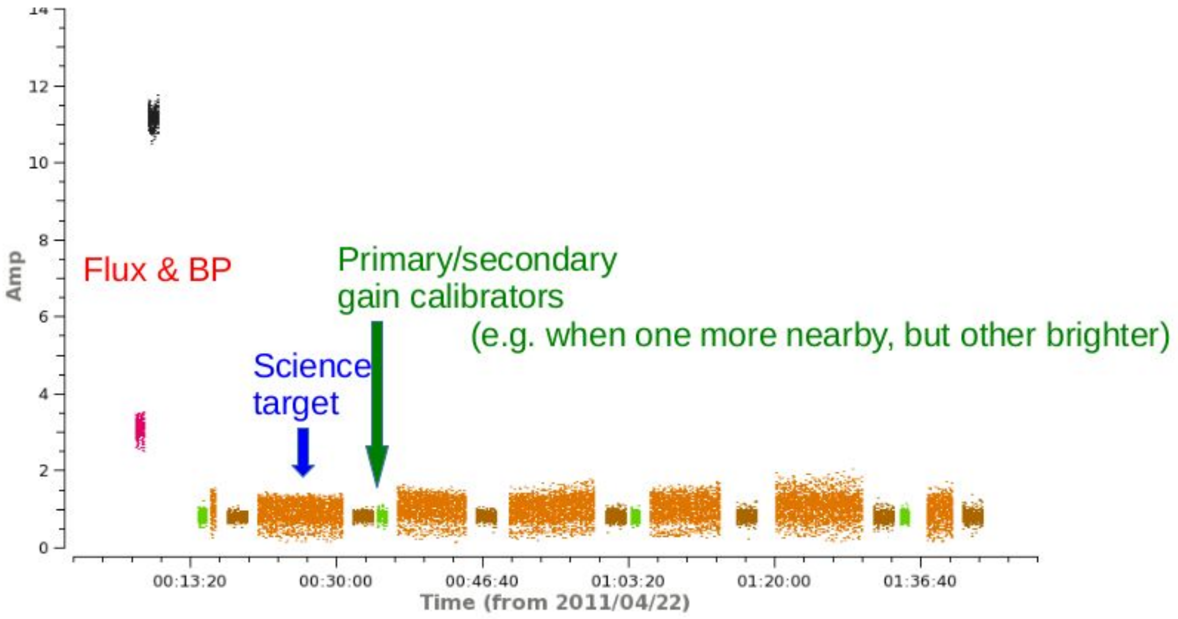
- Variations in water vapor in antenna LOS causes significant phase errors:
- ALMA 12m antennas have radiometers that measure water along LOS every 1.1s to correct for this:





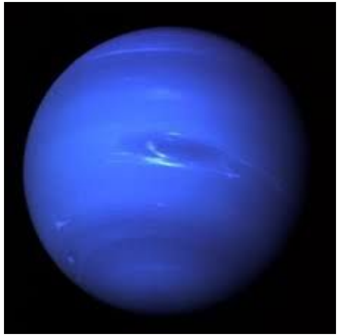
Calibrator Source Observations

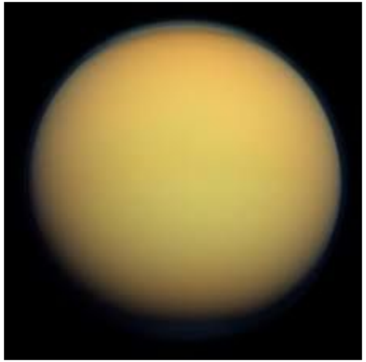
- Flux, bandpass, and phase (gain) calibration vary on short timescales and require observing additional calibrator targets:

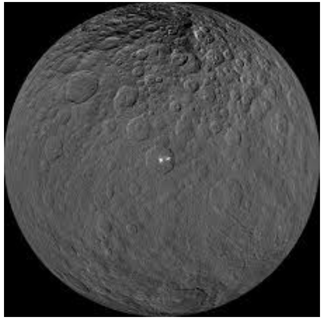


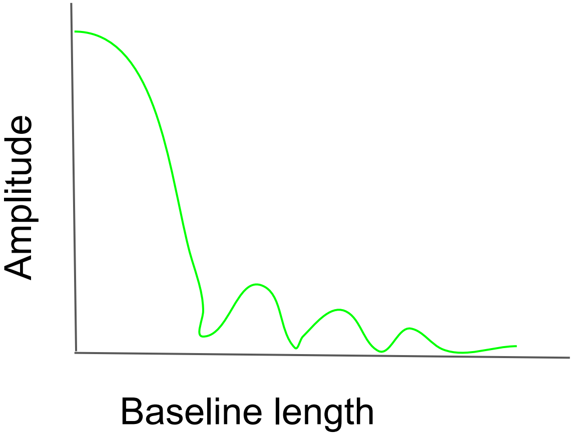
Flux Calibration

- Ideally: observe a mm source of known brightness and scale our visibility amplitudes.
- Problem: the mm sky has few suitable calibrators!
- Option 1: Solar System bodies
- Pros: bright, flux known to 5%
- Cons: not always observable, low flux on long baselines...



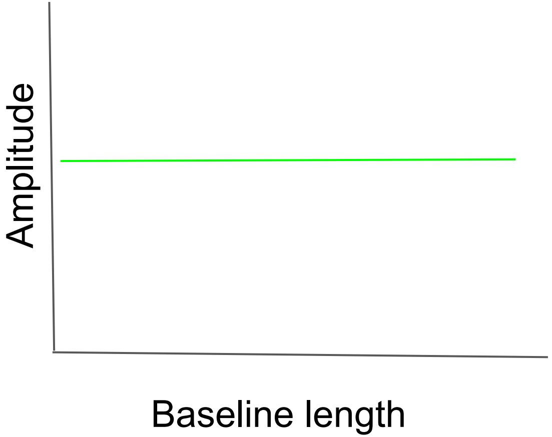


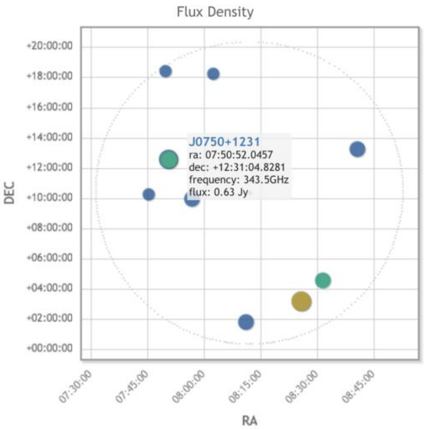


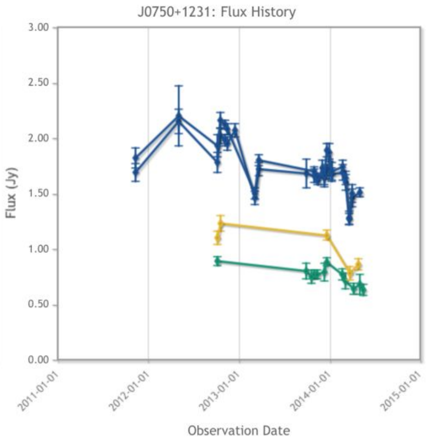


Flux Calibration

- Option 2: Quasars
- Pros: Bright, point-like, all over the sky
- Cons: **Time variable Flux!**
- ALMA's solution: Observe many quasars regularly with solar system objects in "grid" over the sky.
- Observing an ALMA grid quasar with science targets provides ~10% accuracy at 233 GHz (band 6)

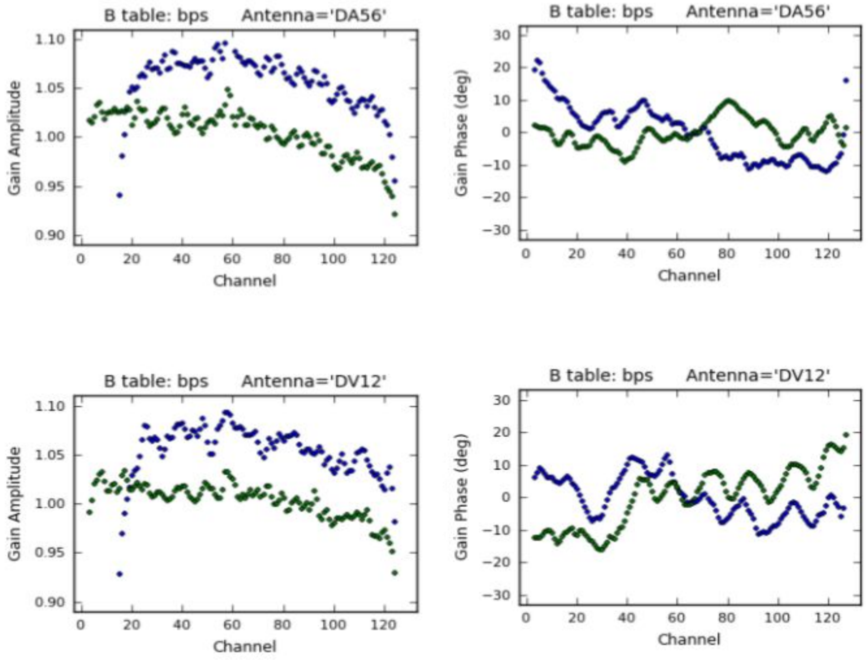






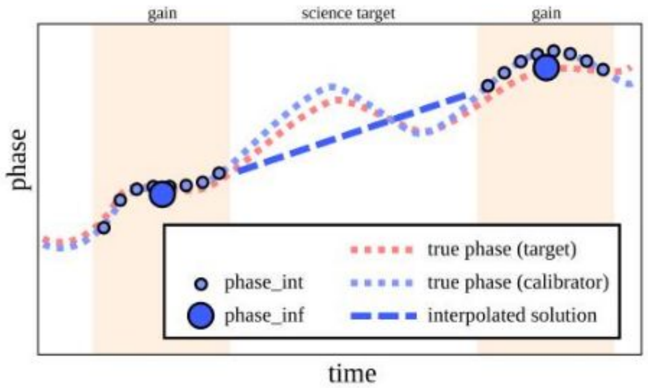
Bandpass Calibration

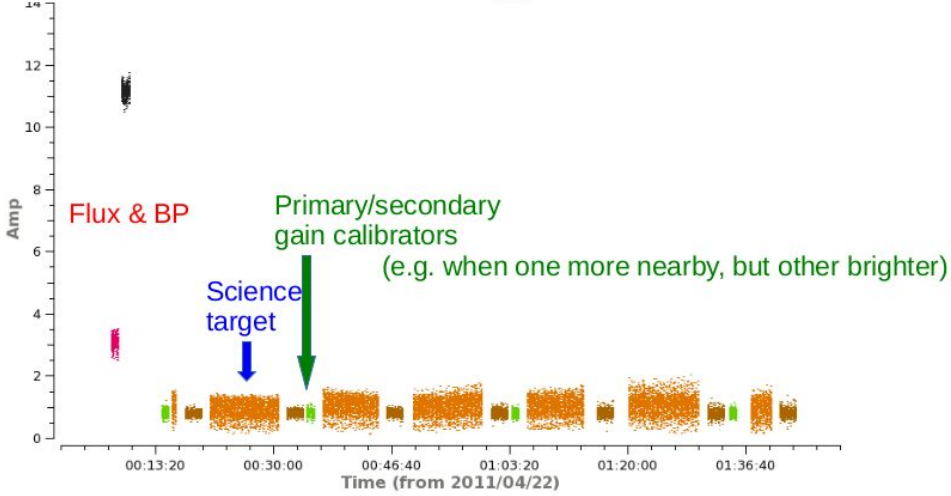
- Correct Amplitude and Phase across bandpass by observing a calibrator.
- Usually done once before/after science target observations.



Phase (gain) Calibration

- Atmospheric turbulence causes large, time variable phase observations **during** an observation.
- Short science target scans are bracketed by observations of a phase calibrator.
- Calibration solutions are interpolated in time to science observations.





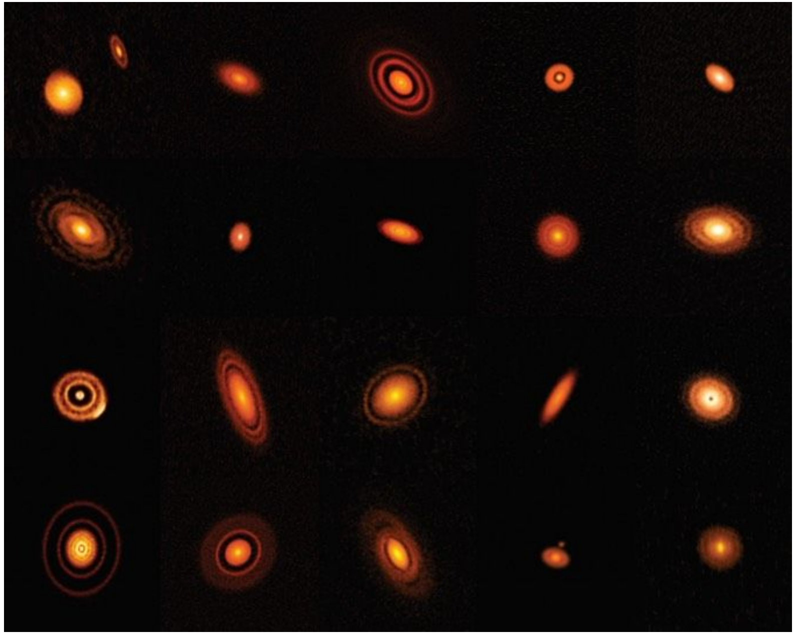
Questions?

Deconvolution and CLEAN

Logan Francis
PhD Student, Uvic
Sept 24th 2020

Outline

- The Fourier Transform Relationship
- Sampling and the Dirty Beam
- Dirty Images
- Radio Astronomy Image units
- Data Weighting Schemes
- The CLEAN algorithm
- CLEAN parameters
- CLEAN variations



Fourier Transform Relationship

- Recall that for interferometers the image and uv-plane are related by:

$$I_\nu(l,m) = \iint \mathrm{V}(u,v)e^{2\pi i(ul+vm)}dudv$$
$$V(u,v) = \iint I_\nu(l,m)e^{-2\pi i(ul+vm)}dldm$$

- Let's write things more compactly as follows:

$$I = \mathcal{F}^{-1}\{V\} \;\; V = \mathcal{F}\{I\}$$

Effects of Sampling

- Describe sampling of uv-plane with function S(u,v):

$$V_{\text{obs}} = S(u,v)V(u,v) \quad S(u,v) = \sum_i \delta(u-u_i, v-v_i) + \delta(u+u_i, v+v_i)$$

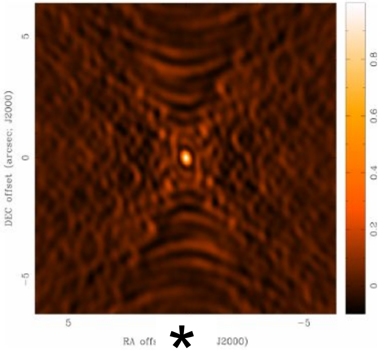
- If we take the Fourier transform of V_{obs}, we obtain the true sky brightness convolved with the Fourier transform of S:

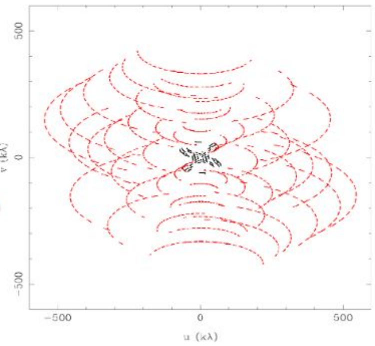
$$B(l,m) = \mathcal{F}\{S(u,v)\} \qquad I * B = \mathcal{F}^{-1}\{SV\}$$

- **B** is known as the "**dirty beam**" or "**synthesized beam**" and is how the array sees a point source. **I*B** is thus called the "**dirty image**".

Dirty Image Examples

"dirty beam"

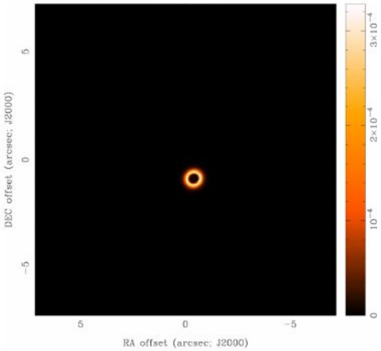


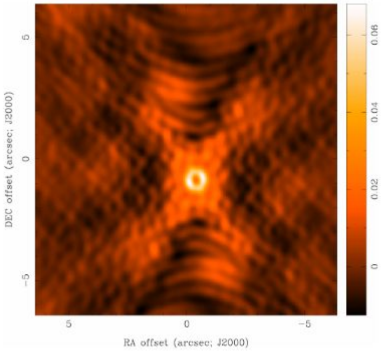


S(u,v)

*

"true image"





"dirty image"

Dirty Image Examples

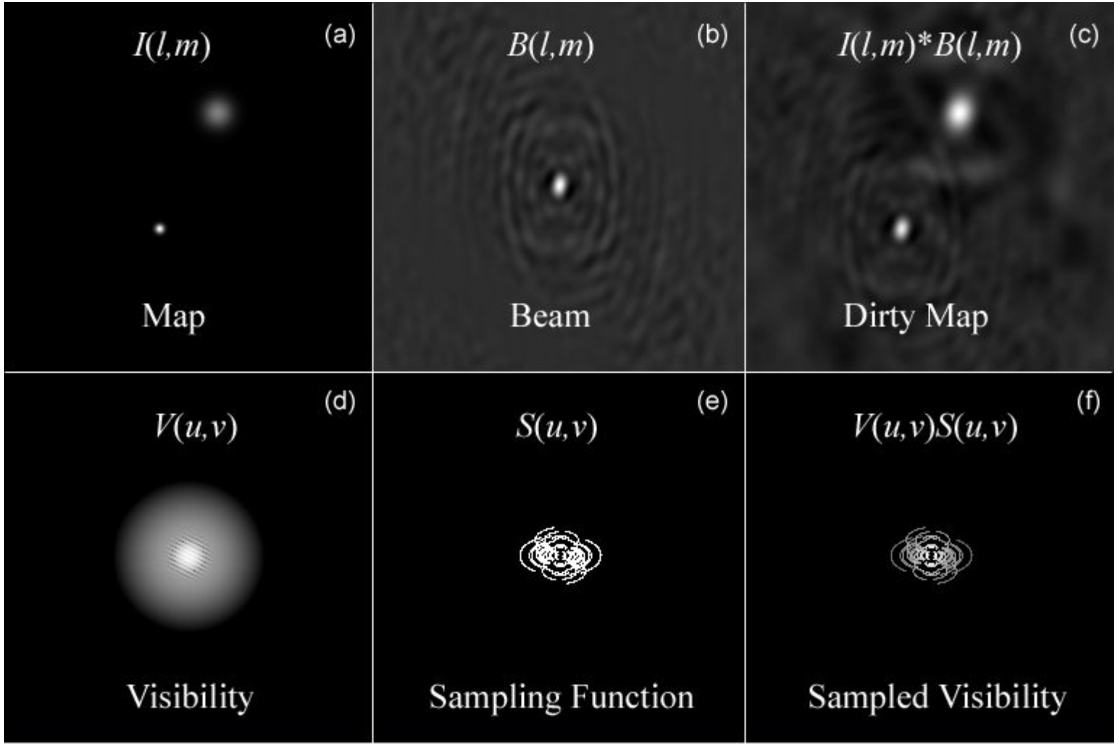
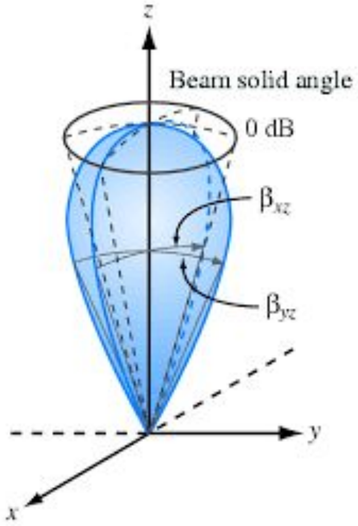


Image units for Radio telescopes

$$1\,\mathrm{Jy} = 10^{-26}\,\frac{\mathrm{W}}{\mathrm{Hz}\cdot\mathrm{m}^2}$$

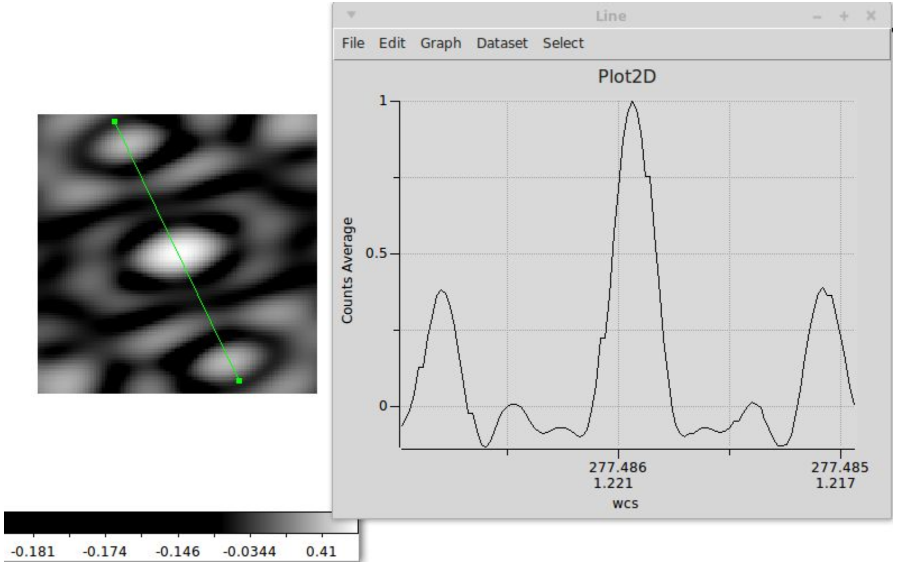
- A Jansky (Jy) is a unit of spectral **flux** density for radio telescopes.
- Image units are often in Jy/**beam**, a unit of spectral **intensity**.
- The **beam** is the solid angle of the telescope point spread function.
- Flux is calculated by integrating the intensity over the angular extent of the source.



$$F_\nu = \iint_{source} I_\nu d\Omega$$

Image units for interferometers

- For interferometers, the beam size of the **synthesized beam** main lobe.
- The Full-width at half-maximum (FWHM) of a Gaussian fit to the main lobe is typically used.
- For interferometers, the single dish response is called the **primary beam**.



Weighting Schemes - Natural

- Not all data is created equally - each visibility sample has an uncertainty determined during calibration.
- A “Natural” weighting scheme weights the **i**’th according to its uncertainty:

$$w_i = \frac{1}{\sigma_i^2}$$

- Natural weighting provides optimal sensitivity for point source detection.

Weighting Schemes - Uniform

- The dirty beam has large sidelobes due to the sparseness of uv-plane sampling.
- This can be mitigated by increasing the weight of sparsely sampled regions of the uv-plane:

$$w_i = \left(\frac{1}{\sigma_i^2}\right) / W_k$$

- W_k is the **local density** of data points in the uv-plane, which has been binned into **k** cells of size 2/FOV.
- Uniformly weighting data **increases the image noise.**

Weighting Schemes - Briggs Robust

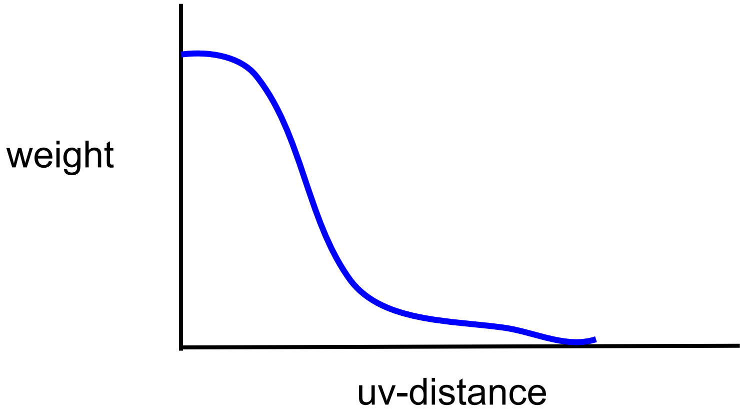
- Brigg's weighting scheme attempts to find a good compromise between uniform and natural weightings.
- The "robustness" parameter **R** takes values between -2 (essentially uniform) and 2 (essentially natural).

$$\omega_i = \frac{1}{\sigma_k^2} \quad w_i = \frac{\omega_i}{1 + W_k f^2} \quad f^2 = \frac{(5 * 10^{-R})^2}{\frac{\sum_k W_k^2}{\sum_i \omega_i}}$$

- R=0.5 provides a good compromise for most ALMA data, but you should experiment and choose what's best for your data!

Weighting Schemes - uv tapering

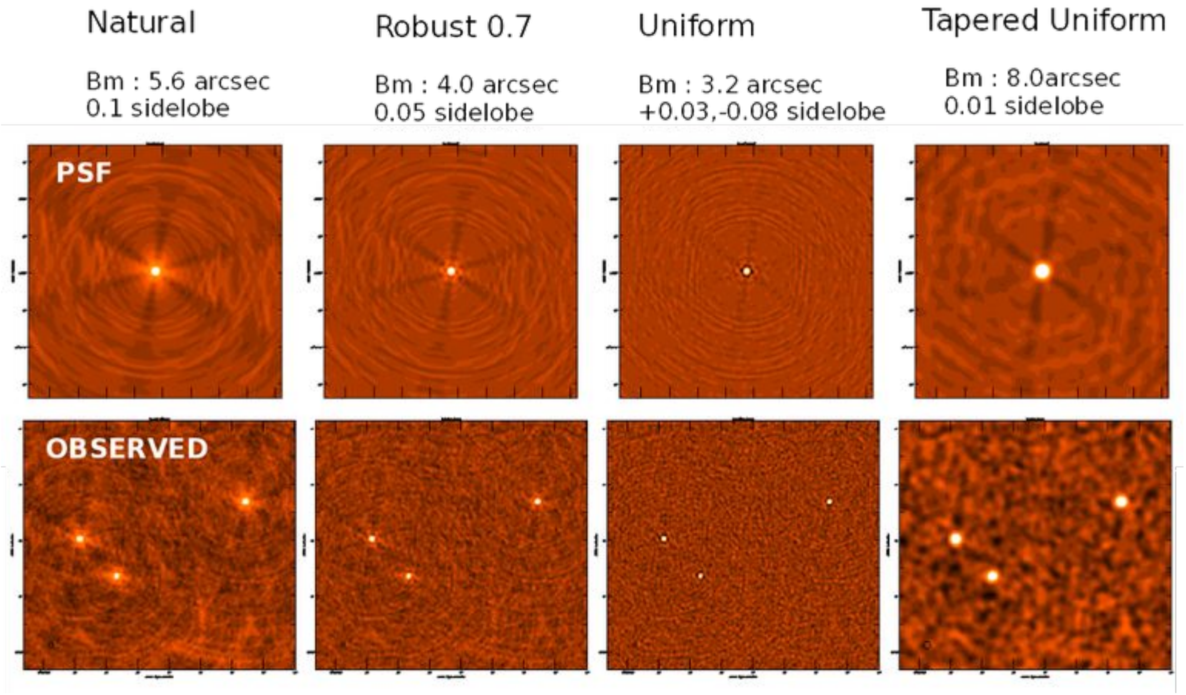
- Long baselines may contribute significantly to high sidelobes in the dirty beam.
- Short baselines recover extended structures which may be of interest.
- We can optimize for this by multiplying our data weights in the uv-plane by a Gaussian, thus “tapering” out the effect of long baselines.
- This **throws away data** and **increases noise!**



Effect of Weighting on the Dirty Beam

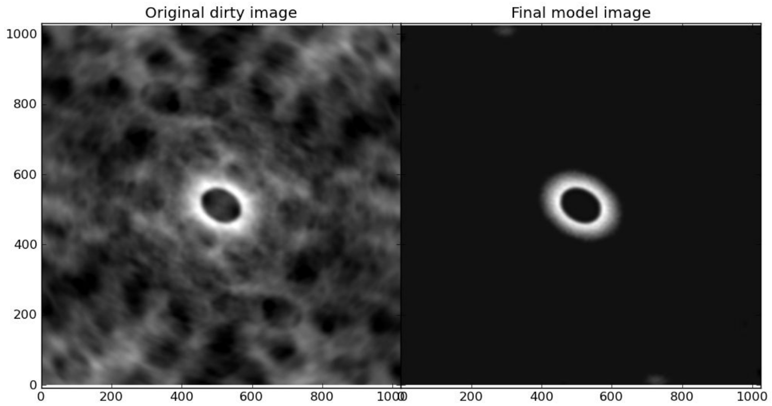
- We typically include the weights in our expression for the sampling function:

$$S(u,v) = \sum_i w_i \left(\delta(u - u_i, v - v_i) + \delta(u + u_i, v + v_i)\right)$$



The CLEAN Algorithm

- The CLEAN algorithm (Hogbom 1974) attempts to **deconvolve** the image and remove the effects of the dirty beam.



See http://nesanders.github.io/gICLEAN/examples.html

1. Produce an initial dirty image I_D

2. Find brightest point I_0 in I_D

3. Subtract from the image the dirty beam B scaled and shifted to peak at γI_0, where $\gamma<1$.

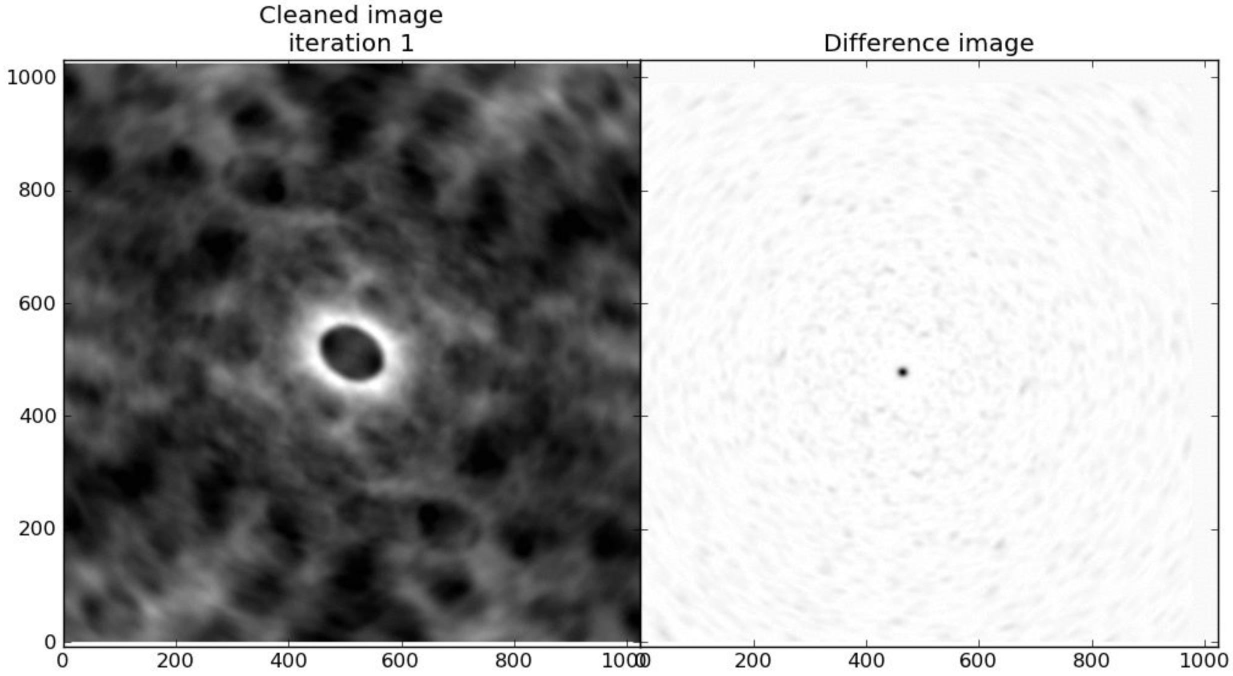
4. Add a point source of brightness γI_0 to "clean" model image.

5. Repeat Steps 2-4, replacing I_D with the subtracted map from the last iteration. Stop when I_0 is comparable to the noise level in the map.

6. Convolve the "clean" model with an ideal "clean" beam, typically a Gaussian fit to the main lobe of the dirty beam.

7. Add dirty image residuals to clean image.

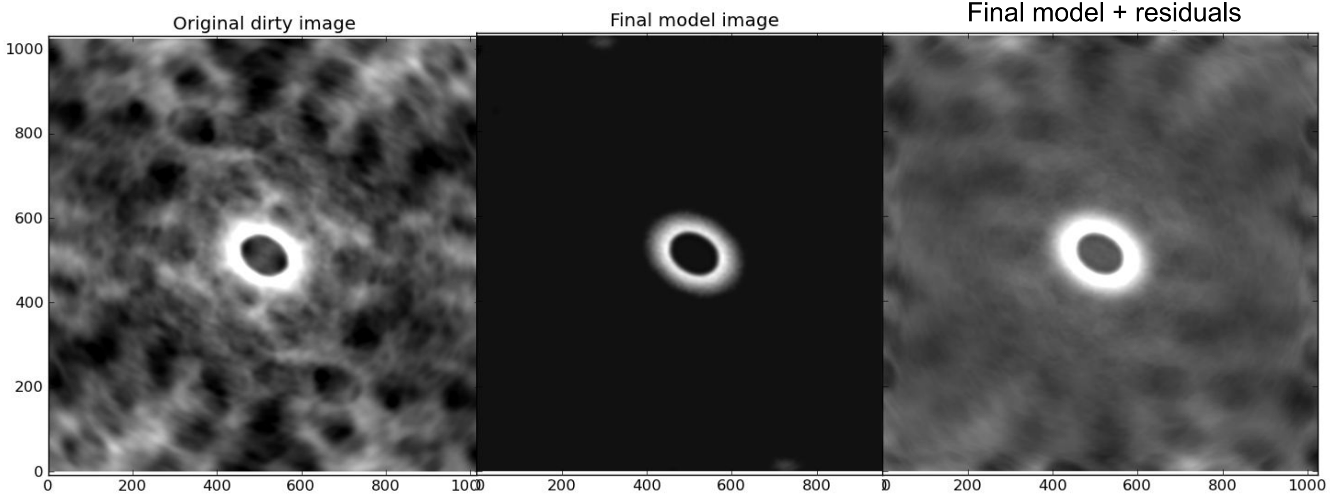
CLEAN in action



- Left: residual image. Right: Removed model x Dirty beam

CLEAN in action

- CLEAN provides an impressive increase in image quality!
- CLEAN works well because we know our images are sparse and mostly empty of emission.

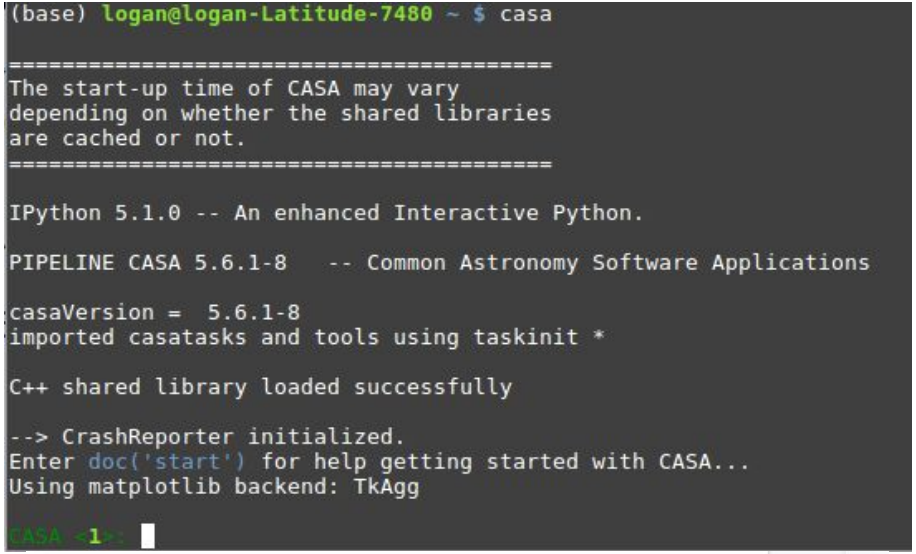


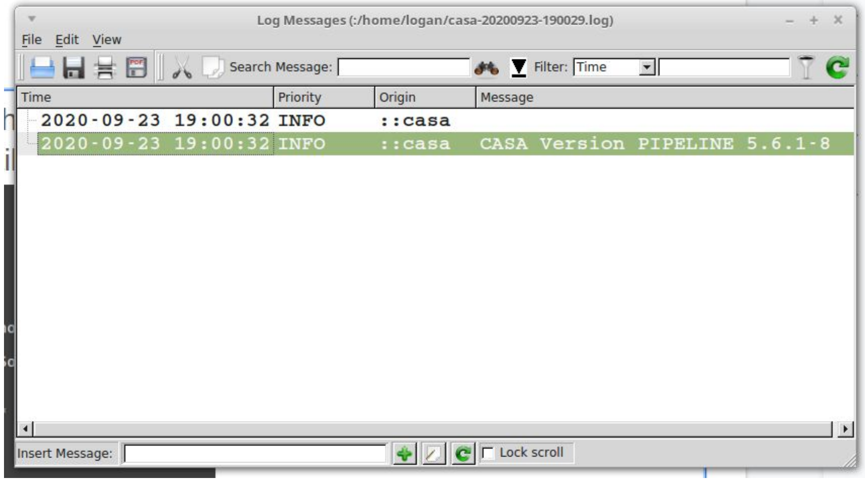
CLEAN and CASA

- Using CLEAN with ALMA data is usually done with the CASA package.
- Set of C++ tools for radio data reduction with python bindings.
- Includes it's own python installation.
- Formerly python 2, new versions will use python 3.
- Documentation:
- https://casa.nrao.edu/casadocs/casa-6.1.0
- CASA Tutorials:
  https://casaguides.nrao.edu/index.php?title=Main_Page

CASA basics

- Interface provides an ipython interpreter and GUI output logger.
- Radio reduction tools available as “tasks” i.e. python functions.
- Useful commands:
  - **execfile**(*scriptname)* - execute commands in a script
  - **inp** *taskname* - check task inputs
  - **help**(*taskname)* - get task documentation in terminal
  - **doc**(*‘taskname’*) - get task documentation in web browser
  - **go** *taskname* - run task with current inputs
  - **tget** *taskname* - get inputs last used to run task





CLEAN in CASA

- CLEAN can be used through task *tclean* in CASA.
- **Huge** number of inputs - but only a handful at a time are needed.

```
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''        #  Name of input visibility file(s)
selectdata          =       True        #  Enable data selection parameters
     field          =         ''        #  field(s) to select
     spw            =         ''        #  spw(s)/channels to select
     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)

datacolumn          = 'corrected'       #  Data column to image(data,corrected)
imagename           =         ''        #  Pre-name of output images
imsize              =      [100]        #  Number of pixels
cell                = ['1arcsec']       #  Cell size
phasecenter         =         ''        #  Phase center of the image
stokes              =        'I'        #  Stokes Planes to make
projection          =      'SIN'        #  Coordinate projection
startmodel          =         ''        #  Name of starting model image
specmode            =      'mfs'        #  Spectral definition mode
                                        #   (mfs,cube,cubedata, cubesource)
     reffreq        =         ''        #  Reference frequency

gridder             = 'standard'        #  Gridding options (standard, wproject,
                                        #   widefield, mosaic, awproject)
     vptable        =         ''        #  Name of Voltage Pattern table
     pblimit        =        0.2        #  PB gain level at which to cut off
                                        #   normalizations

deconvolver         =   'hogbom'        #  Minor cycle algorithm (hogbom,clark,m
                                        #   ultiscale,mtmfs,mem,clarkstokes)
restoration         =       True        #  Do restoration steps (or not)
     restoringbeam  =         []        #  Restoring beam shape to use. Default
                                        #   is the PSF main lobe
     pbcor          =      False        #  Apply PB correction on the output
                                        #   restored image

outlierfile         =         ''        #  Name of outlier-field image
                                        #   definitions
weighting           =  'natural'        #  Weighting scheme
                                        #   (natural,uniform,briggs,
                                        #   briggsabs[experimental])
     uvtaper        =         []        #  uv-taper on outer baselines in uv-
                                        #   plane

niter               =          0        #  Maximum number of iterations
usemask             =     'user'        #  Type of mask(s) for deconvolution:
                                        #   user, pb, or auto-multithresh
     mask           =         ''        #  Mask (a list of image name(s) or
                                        #   region file(s) or region string(s) )
     pbmask         =        0.0        #  primary beam mask

fastnoise           =       True        #  True: use the faster (old) noise
                                        #   calculation. False: use the new
                                        #   improved noise calculations
restart             =       True        #  True : Re-use existing images. False
                                        #   : Increment imagename
savemodel           =     'none'        #  Options to save model visibilities
                                        #   (none, virtual, modelcolumn)
calcres             =       True        #  Calculate initial residual image
calcpsf             =       True        #  Calculate PSF
parallel            =      False        #  Run major cycles in parallel
```

Basic tclean parameters

- Most parameters for *tclean* can be left as default.
- Pay attention to datacolumn - ‘corrected’ contains calibrated data, if present.
- *field* - defaults to select every observed source (a bad idea)
- *spw* - selects all channels/windows by default - fine if there are no spectral lines

```
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =         ''          #  Name of input visibility file(s)
selectdata          =       True          #  Enable data selection parameters
     field          =         ''          #  field(s) to select
     spw            =         ''          #  spw(s)/channels to select
     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)

datacolumn          = 'corrected'         #  Data column to image(data,corrected)
imagename           =         ''          #  Pre-name of output images
imsize              =      [100]          #  Number of pixels
cell                = ['1arcsec']         #  Cell size
phasecenter         =         ''          #  Phase center of the image
stokes              =        'I'          #  Stokes Planes to make
projection          =      'SIN'          #  Coordinate projection
startmodel          =         ''          #  Name of starting model image
```

Basic tclean parameters

- Specmode:
  - 'mfs' = multifrequency synthesis - used for continuum
  - 'Cube' used to image spectral lines
- *reffreq* - reference frequency of image - defaults to average across selected spw
- *pbcor* - correct for antenna primary beam - important for sources on field edge.

```
specmode            =      'mfs'        #  Spectral definition mode
                                        #   (mfs,cube,cubedata, cubesource)
     reffreq        =         ''        #  Reference frequency

gridder             = 'standard'        #  Gridding options (standard, wproject,
                                        #   widefield, mosaic, awproject)
     vptable        =         ''        #  Name of Voltage Pattern table
     pblimit        =        0.2        #  PB gain level at which to cut off
                                        #   normalizations

deconvolver         =    'hogbom'       #  Minor cycle algorithm (hogbom,clark,m
                                        #   ultiscale,mtmfs,mem,clarkstokes)
restoration         =       True        #  Do restoration steps (or not)
     restoringbeam  =         []        #  Restoring beam shape to use. Default
                                        #   is the PSF main lobe
     pbcor          =      False        #  Apply PB correction on the output
                                        #   restored image
```

Basic tclean parameters

- *Weighting* - ‘natural’, ‘uniform’ and ‘briggs’ available (and others…).

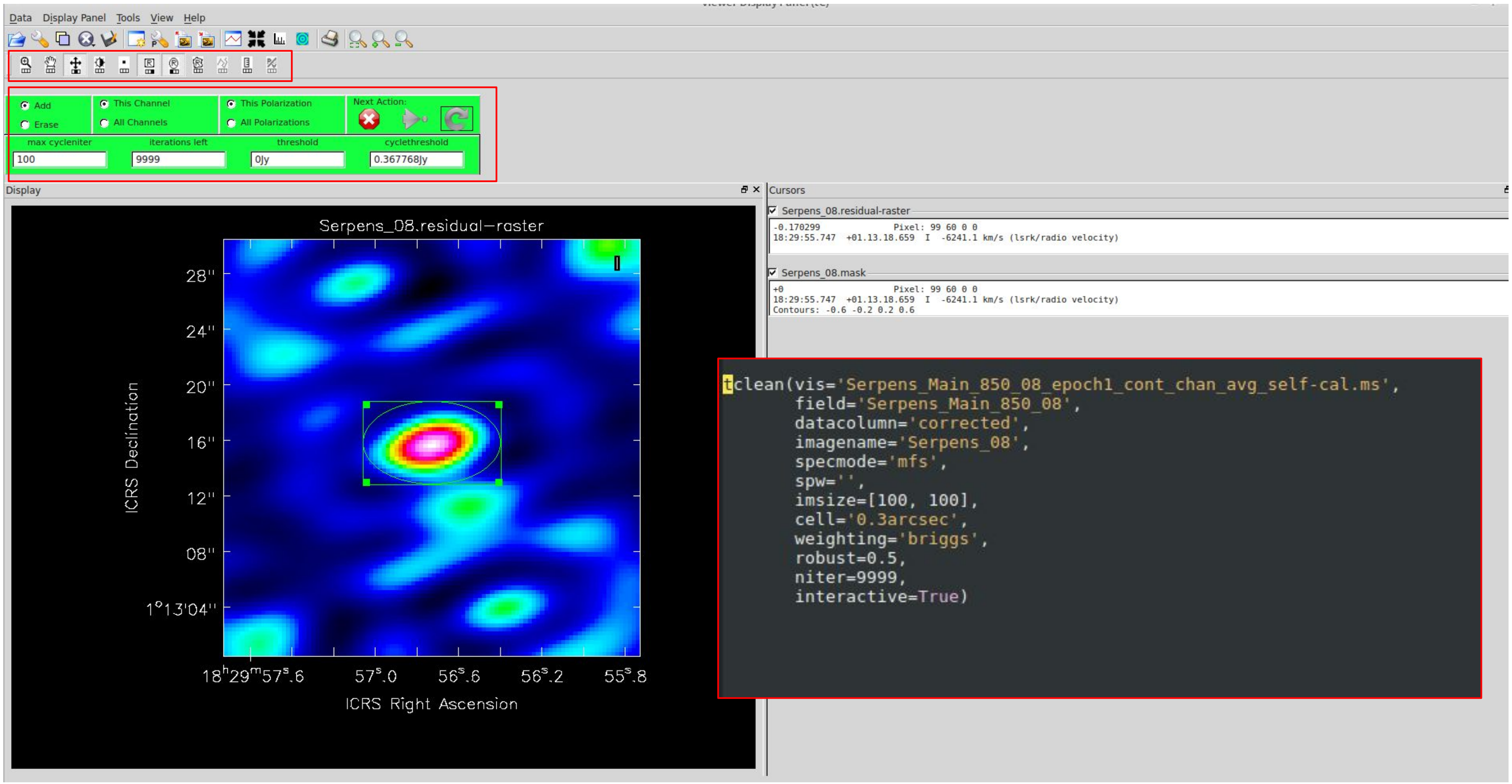
```
weighting           =     'briggs'        #  Weighting scheme
                                          #   (natural,uniform,briggs,
                                          #   briggsabs[experimental])
     robust         =          0.5        #  Robustness parameter
     npixels        =            0        #  Number of pixels to determine uv-cell
                                          #   size
     uvtaper        =           []        #  uv-taper on outer baselines in uv-
                                          #   plane
```

Basic tclean parameters

- *niter* - just set to some huge number if you don't know what to use
- *threshold* - can be used to instead of *niter*, e.g '3 mJy', set to expected noise level in image.
- *Interactive* - Set to "True" to show GUI interface

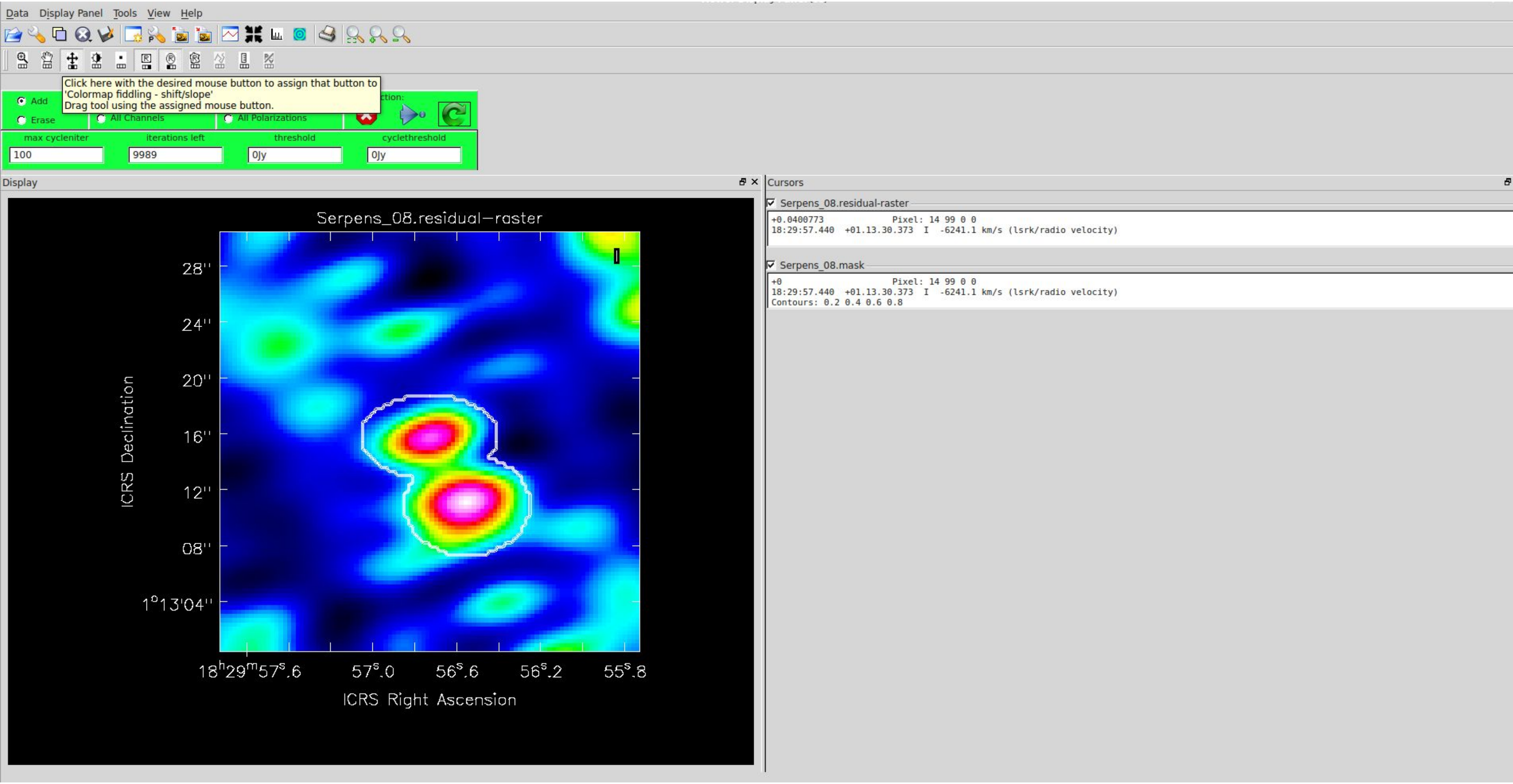
```
niter            =       9999     #   Maximum number of iterations
    gain             =        0.1     #   Loop gain
    threshold        =        0.0     #   Stopping threshold
    nsigma           =        0.0     #   Multiplicative factor for rms-based
                                      #    threshold stopping
    cycleniter       =         -1     #   Maximum number of minor-cycle
                                      #    iterations
    cyclefactor      =        1.0     #   Scaling on PSF sidelobe level to
                                      #    compute the minor-cycle stopping
                                      #    threshold.
    minpsffraction   =       0.05     #   PSF fraction that marks the max depth
                                      #    of cleaning in the minor cycle
    maxpsffraction   =        0.8     #   PSF fraction that marks the minimum
                                      #    depth of cleaning in the minor cycle
    interactive      =      False     #   Modify masks and parameters at
                                      #    runtime
```

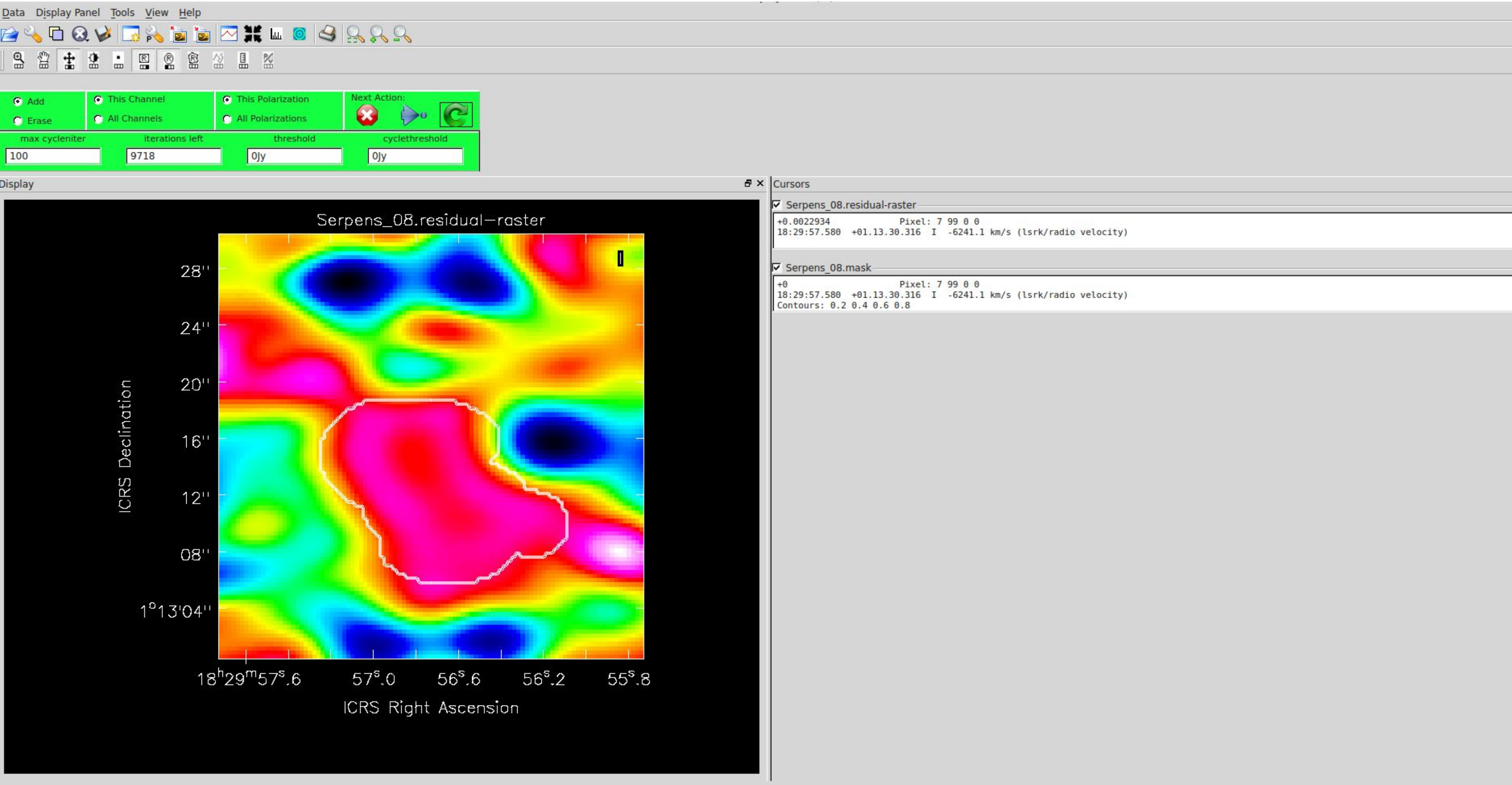
Tclean example



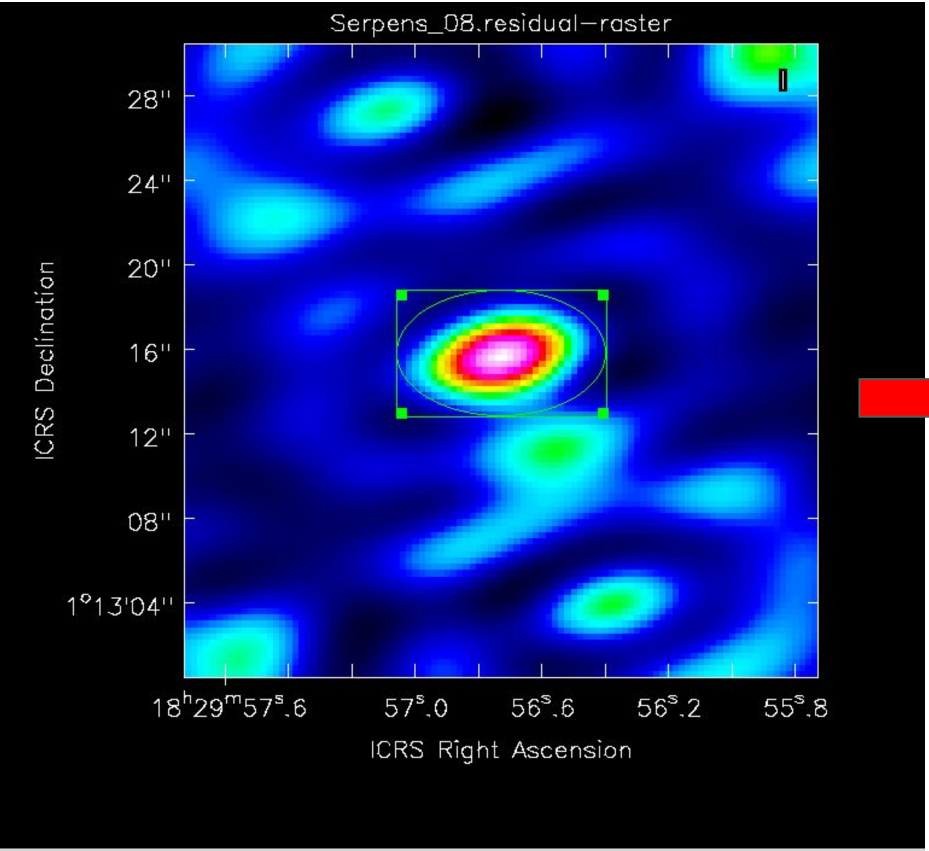
Tclean in logger

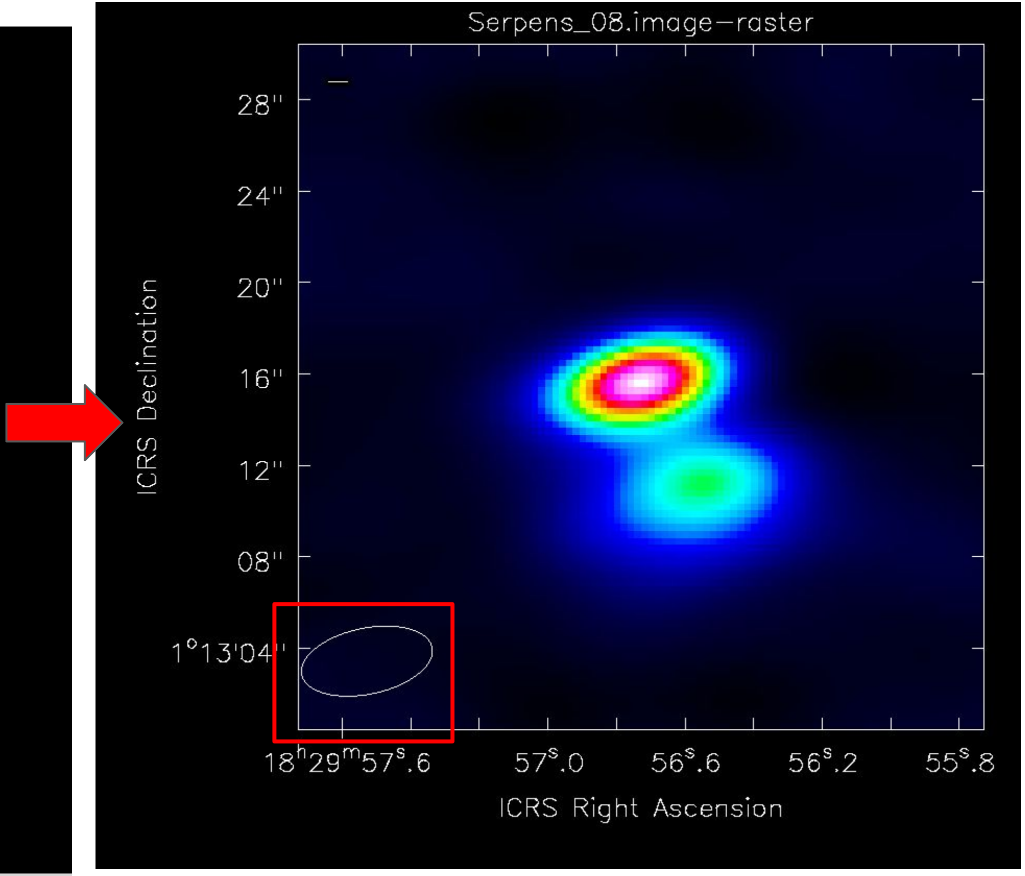
INFO	…fineImage	Impars : start
INFO	…fineImage	Shape : [100, 100, 1, 1]Spectral : [3.43505e+11] at [0] with increment [1.58443e+10]
INFO	…fineImage	Set Gridding options for [Serpens_08] with ftmachine : gridft
INFO	…:weight()	Set imaging weights : Briggs weighting: sidelobes will be suppressed over full image
INFO	…gWeight()	Normal robustness, robust = 0.5
INFO	…nvolution	Set Deconvolution Options for [Serpens_08] : hogbom
INFO	…Iteration	Set Iteration Control Options
INFO	…::makePSF	--- Make PSF ---
INFO	…nsitivity	[Serpens_08] Theoretical sensitivity (Jy/bm):0.000533952
INFO	…ntBeamSet	Beam : 5.82657 arcsec, 2.88889 arcsec, -78.1369 deg
INFO	…imaryBeam	vi2 : Evaluating Primary Beam model onto image grid(s)
INFO	…ajorCycle	--- Run Major Cycle 1 -------------------------------------
INFO	…mageStats	[Serpens_08] Peak residual (max,min) over full image : (0.94841,-0.264522)
INFO	…mageStats	[Serpens_08] Total Model Flux : 0
INFO	…setupMask	[Serpens_08] Initializing new mask to 0.0 for interactive drawing
INFO	…setupMask	[Serpens_08] Number of pixels in the clean mask : 0 out of a total of 10000 pixels. [0 %]
INFO	…ctivemask	[Serpens_08.mask] Mask modified from 0 pixels to 510 pixels
INFO	…teraction	[Serpens_08] Mask changed interactively.
INFO	…mageStats	[Serpens_08] Peak residual (max,min) within mask : (0.94841,-0.118535) over full image : (0.94841,-0.264522)
INFO	…mageStats	[Serpens_08] Total Model Flux : 0
INFO	…inorCycle	--- Run Minor Cycle Iterations ---
INFO	…econvolve	[Serpens_08] Run Hogbom minor-cycle | CycleThreshold=0.367768, CycleNiter=100, Gain=0.1
INFO	…econvolve	[Serpens_08] iters=0->10 [10], model=0->0.61882, peakres=0.94841->0.333689, Reached cyclethreshold.
INFO	…ionRecord	Completed 10 iterations.


Data Display Panel Tools View Help
Click here with the desired mouse button to assign that button to
'Colormap fiddling - shift/slope'
Drag tool using the assigned mouse button.
Add
Erase
All Channels
All Polarizations
max cycleniter
100
iterations left
9989
threshold
0Jy
cyclethreshold
0Jy
Display
Serpens_08.residual−raster
ICRS Declination
28''
24''
20''
16''
12''
08''
1°13'04''
18h29m57s.6
57s.0
56s.6
56s.2
55s.8
ICRS Right Ascension
Cursors
Serpens_08.residual-raster
+0.0400773 Pixel: 14 99 0 0
18:29:57.440 +01.13.30.373 I -6241.1 km/s (lsrk/radio velocity)
Serpens_08.mask
+0 Pixel: 14 99 0 0
18:29:57.440 +01.13.30.373 I -6241.1 km/s (lsrk/radio velocity)
Contours: 0.2 0.4 0.6 0.8


Data Display Panel Tools View Help
Add
Erase
This Channel
All Channels
This Polarization
All Polarizations
Next Action:
max cycleniter
100
iterations left
9718
threshold
0Jy
cyclethreshold
0Jy
Display
Cursors
Serpens_08.residual-raster
+0.0022934 Pixel: 7 99 0 0
18:29:57.580 +01.13.30.316 I -6241.1 km/s (lsrk/radio velocity)
Serpens_08.mask
+0 Pixel: 7 99 0 0
18:29:57.580 +01.13.30.316 I -6241.1 km/s (lsrk/radio velocity)
Contours: 0.2 0.4 0.6 0.8
Serpens_08.residual−raster
28''
24''
20''
16''
12''
08''
1°13'04''
ICRS Declination
18h29m57s.6
57s.0
56s.6
56s.2
55s.8
ICRS Right Ascension

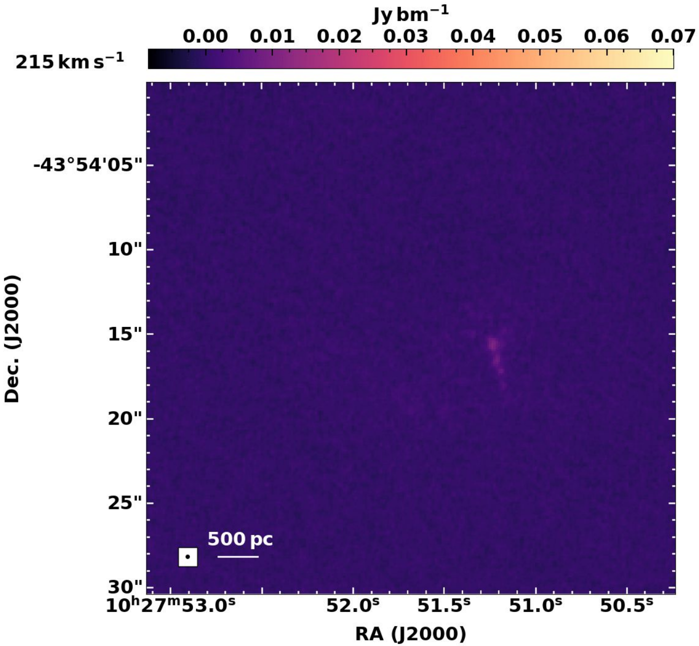
Tclean results





Other tclean options - Cube cleaning

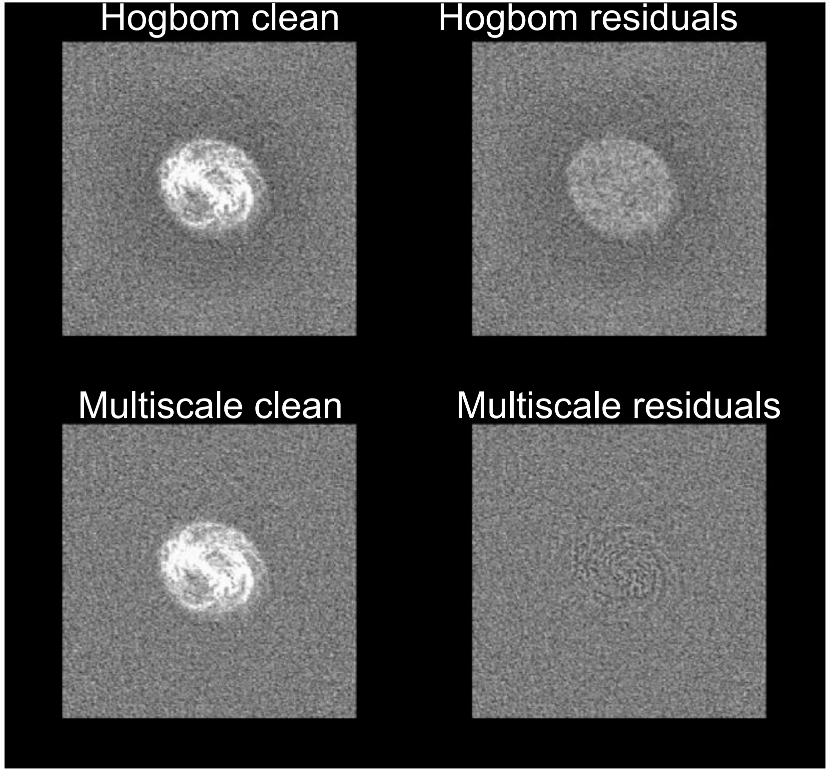
- Use *specmode*='cube' and subparameters to image spectral data




```
specmode            =     'cube'        #  Spectral definition mode
                                        #   (mfs,cube,cubedata, cubesource)
     nchan          =         20        #  Number of channels in the output
                                        #   image
     start          =  '-20km/s'        #  First channel (e.g. start=3,start='1.
                                        #   1GHz',start='15343km/s')
     width          =    '2km/s'        #  Channel width (e.g. width=2,width='0.
                                        #   1MHz',width='10km/s')
     outframe       =     'LSRK'        #  Spectral reference frame in which to
                                        #   interpret 'start' and 'width'
     veltype        =    'radio'        #  Velocity type (radio, z, ratio, beta,
                                        #   gamma, optical)
     restfreq       = '230.538GHz'      #  List of rest frequencies
     interpolation  =   'linear'        #  Spectral interpolation
                                        #   (nearest,linear,cubic)
     chanchunks     =          1        #  Number of channel chunks
     perchanweightdensity =       True  #  whether to calculate weight density
                                        #   per channel in Briggs style
                                        #   weighting or not
```

Other tclean options - multiscale cleaning

- CLEAN creates models of the sky using collections of point sources.
- This is not optimal for very extended sources.
- Multiscale cleaning allows CLEAN to use extended Gaussian-like functions for creating a model.



Questions?