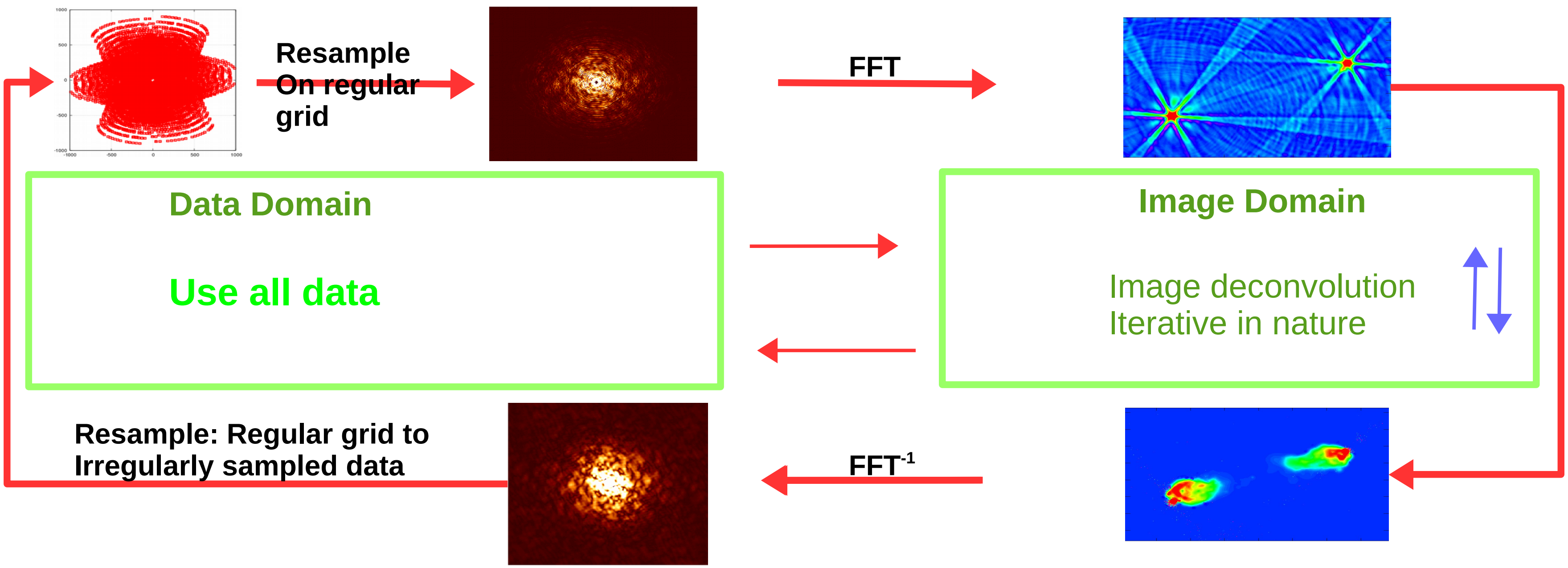
Using AW-Projection from CASA/ARDG branch

S. Bhatnagar

Sept. 7th, 2018

Imaging & Deconvolution: A recap

- Compute residuals using the original data
  - Needs Gridding and de-Gridding during major-cycle iterations



W-Projection

AW-Projection

WB AW-Projection

Standard gridding

CS-Clean

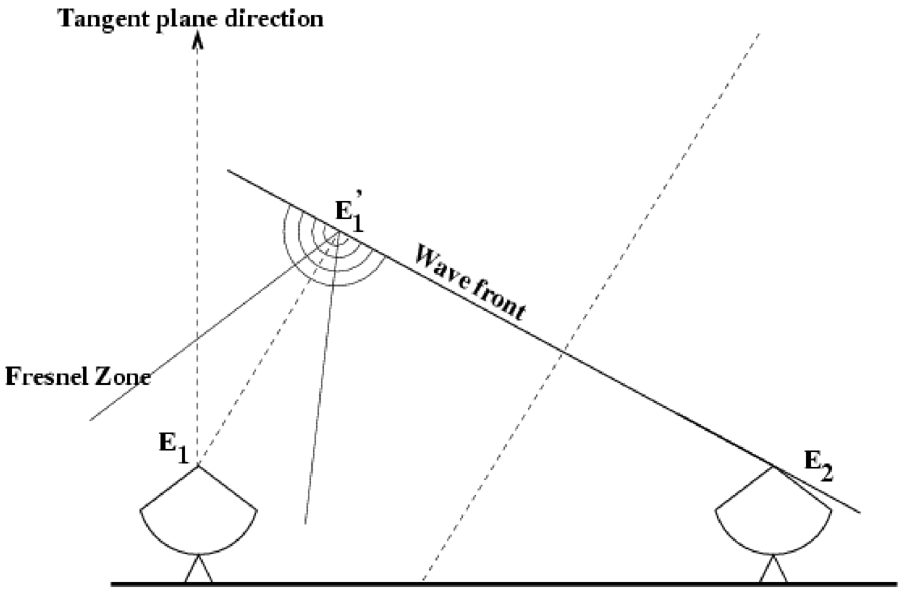
MS-Clean

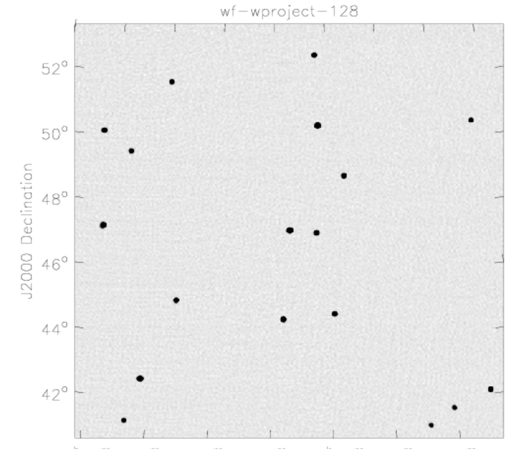
MT-MFS

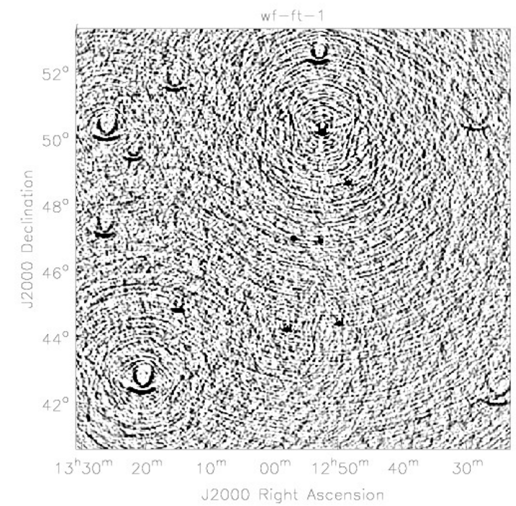
Hogbom Clean

Non co-planar baseline: The W-term

- 2D FT approximation of the Measurement Equation breaks down



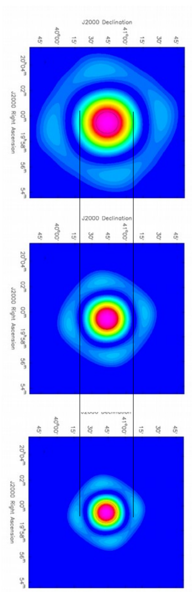


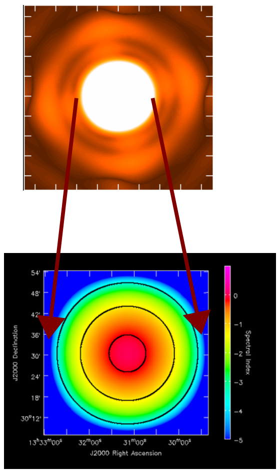


- We measure: $V_{12} = \langle \boldsymbol{E_1}(u,v,w=0)\, \boldsymbol{E_2}^*(0,0,0) \rangle$
- We interpret it as: $V^o_{12} = \langle \boldsymbol{E}'_{\boldsymbol{1}}(u,v,w \neq 0)\, \boldsymbol{E_2}^*(0,0,0) \rangle$
- We should interpret E_1 as [E_1' x Fresnel Propagator]

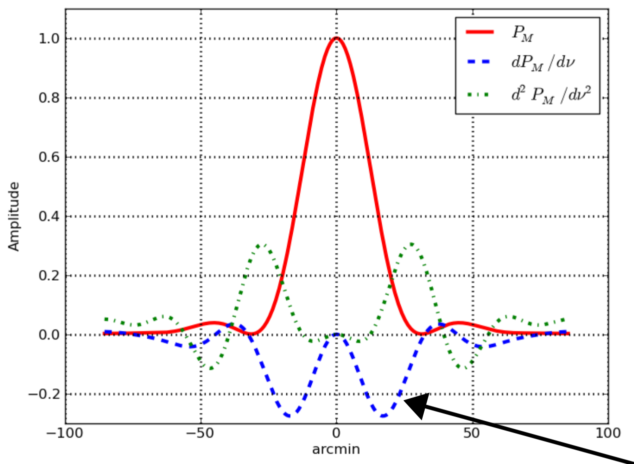
Wide-band Wide-field Imaging

- Wide band data to image beyond the ~50% point of the PB at a reference frequency
  - Bandwidth ratio > ~20%
  - FoV > ~HPBW @ reference frequency
  - Variable PB:
    - Long integration (rotation), Mosaicking (pointings at different PA), in-beam polarization is large (AA)





PB "Spectral Index"



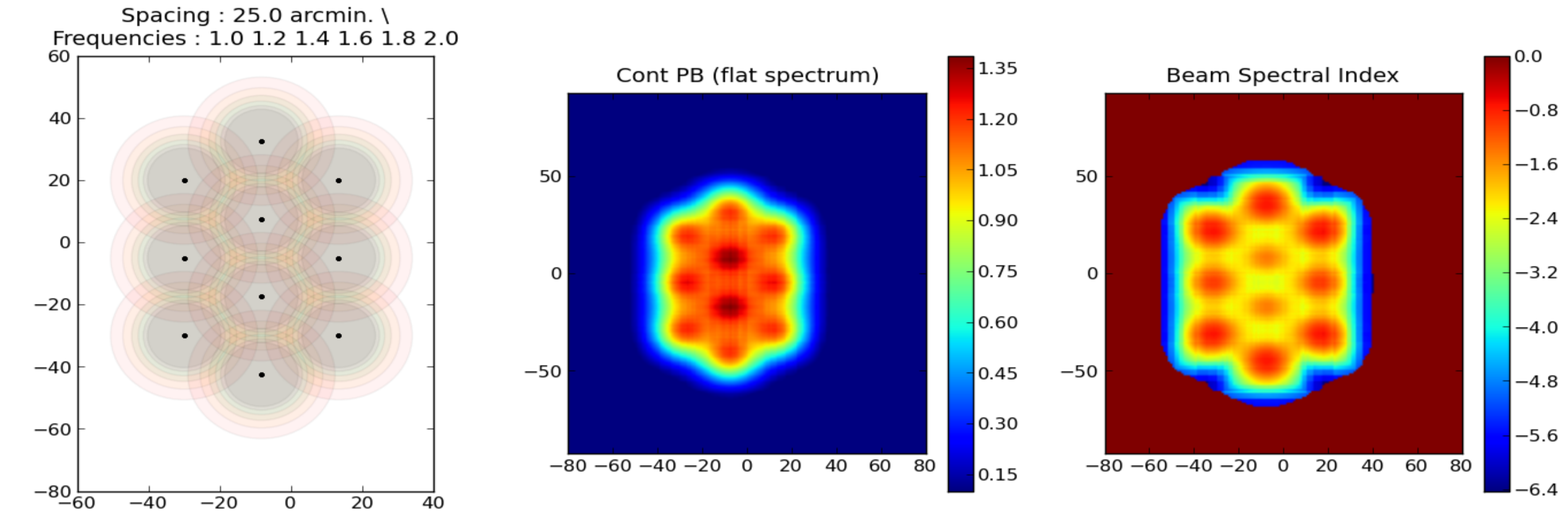
PB Frequency dependence (blue curve)

Wideband Primary Beams – Mosaic

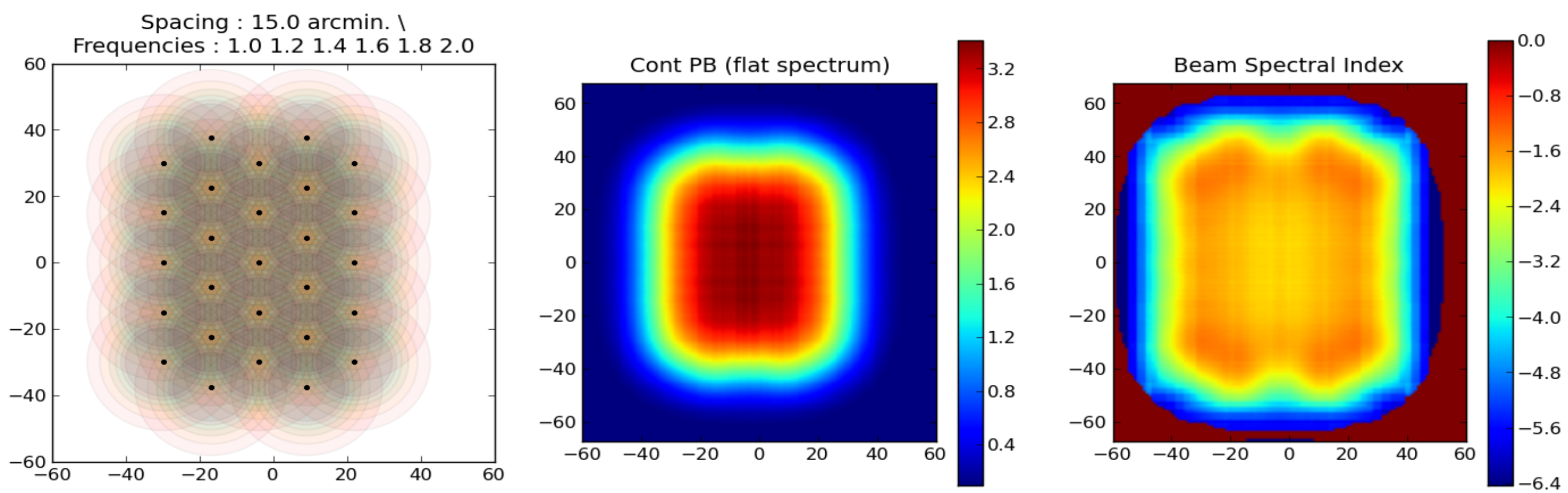
For single pointings, the wideband PB spectrum is relevant only away from the pointing center.

For mosaics, the wideband PB spectrum must be accounted-for <u>all over the mosaic field of view</u>

25 arcmin spacing (1-2 GHz)

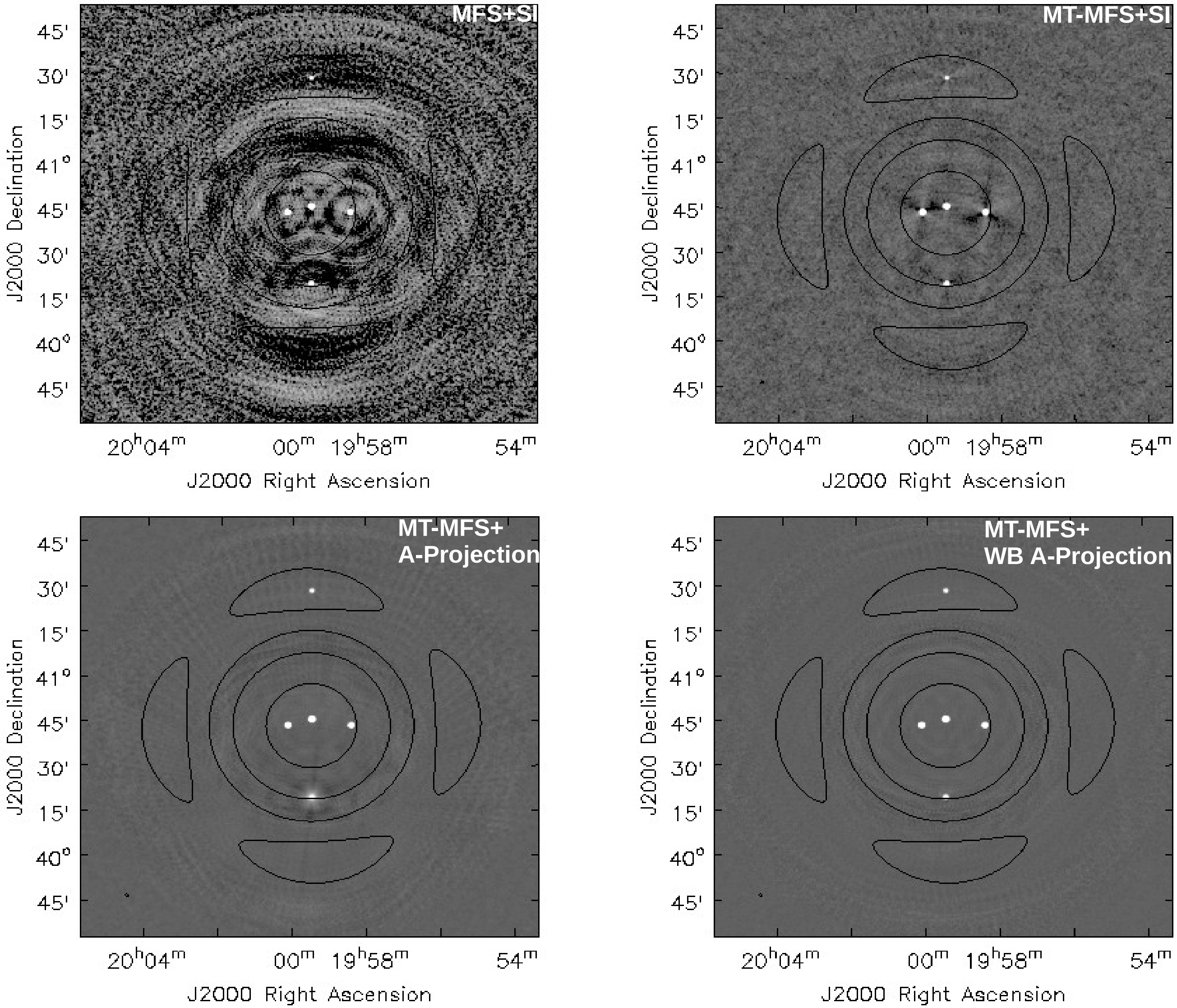


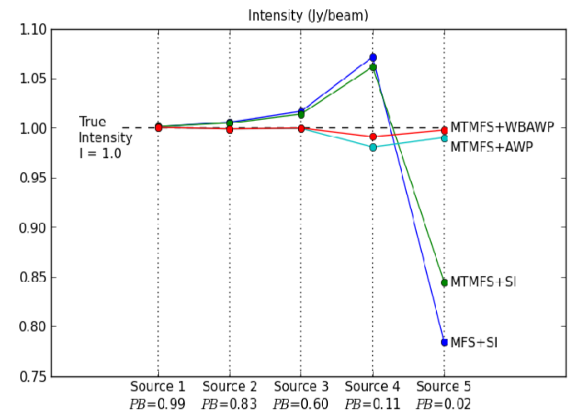
15 arcmin spacing (1-2 GHz)

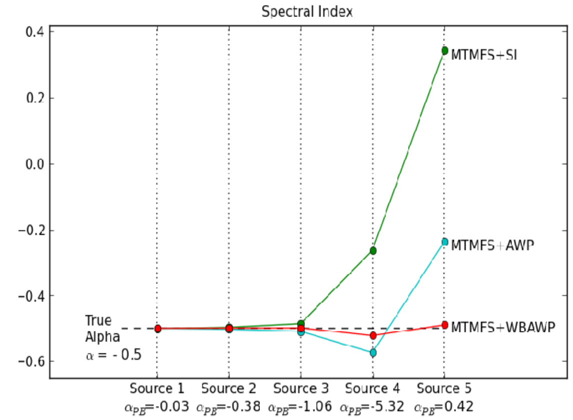


Wide-band Wide-field Imaging

- Characterization of the (WB) A-Projection + MT-MFS







Why new algorithms?

$$V_{ij}(\nu) = G^{DI}_{ij} \; R_{ij} \int \underbrace{P_{ij}(\mathbf{s},\nu,t) \; I(\mathbf{s},\nu) \; e^{\iota[u_{ij} l + v_{ij} m + w_{ij}(\sqrt{1-l^2-m^2}-1)]}}_{\text{Direction Dependent (DD) terms}} \; d\mathbf{s}$$

- Terms inside the integral cannot be accounted-for before imaging
  - Conventional imaging ignores DD terms
  - Also ignores time, frequency and polarization dependence
- Solutions: Project-out the effects during imaging + model frequency dependence of the sky during deconvolution
  - WB AW-Projection + MT-MFS
  - AWP with *conjbeams=True*
- Spectral cube imaging + image-plane corrections/averaging
  - AW-Projection for Cube Imaging + MT-MFS on collapsed cube
  - AWP with *conjbeams=False*

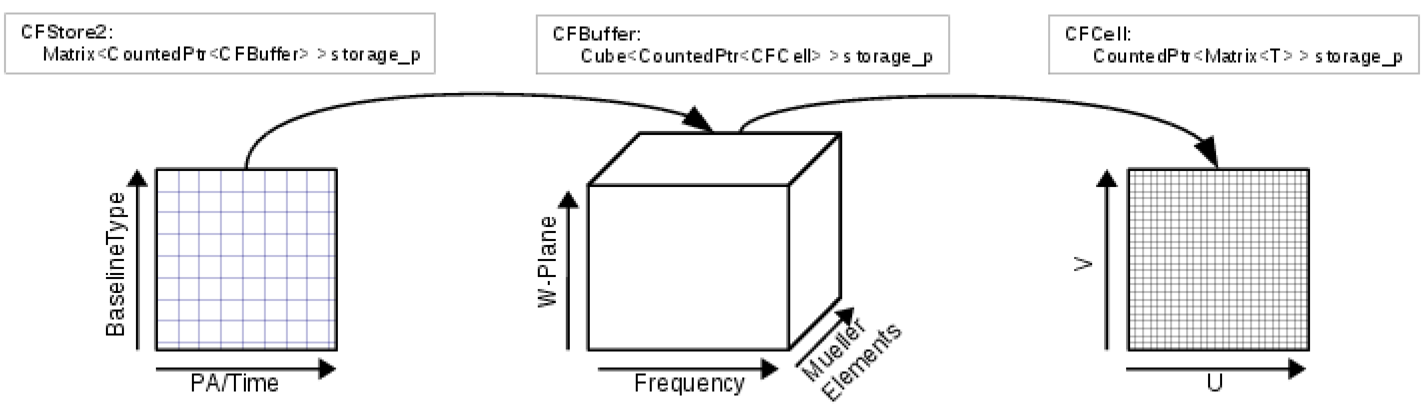
$$V_{ij}(\nu) = G^{DI}_{ij} \; R_{ij} \int P_{ij}(\mathbf{s},\nu,t) \; I(\mathbf{s},\nu) \; e^{\iota[u_{ij} l + v_{ij} m]} e^{\iota w_{ij}(\sqrt{1-l^2-m^2}-1)} \; d\mathbf{s}$$

Why new algorithms?

$$V(\nu,u_{ij},v_{ij},w_{ij}) = R_{ij}\int\left[P_{ij}(\mathbf{s},\nu,t)\ I^{M}(\mathbf{s},\nu)\ e^{\iota w_{ij}\left(\sqrt{1-l^2-m^2}-1\right)}\right]\ e^{\iota[u_{ij}l+v_{ij}m]}\ d\mathbf{s}$$

$$V(\nu,u_{ij},v_{ij},w_{ij}) = R_{ij}\left[\mathbf{A}(\mathbf{u}_{ij},\mathbf{v}_{ij},\boldsymbol{\nu},\mathbf{t}) * \mathbf{W}(\mathbf{u}_{ij},\mathbf{v}_{ij}) * \mathbf{V}^{o}(\mathbf{u}_{ij},\mathbf{v}_{ij})\right] = R_{ij}\ CF_{ij} * V^{o}_{ij}$$

- **A** : A-term/Aperture term. Fourier transform of the antenna PB
- **W** : W-term/Non-coplanar array. Fourier transform of the w-term (Fresnel propagator)
- CF is the Convolution Function. A 2D function that varies with frequency, time, polarization, w-value and antenna pairs (baseline)



Projection algorithms

- Direction-dependent effects in the image domain are convolutional terms in the data domain

- Projection algorithms for DD corrections:
  - Project-out various DD effects as part of the gridding operator
    - ME:

$$V_{ij}^{Obs} = A_{ij} * V^o + N_{ij}$$

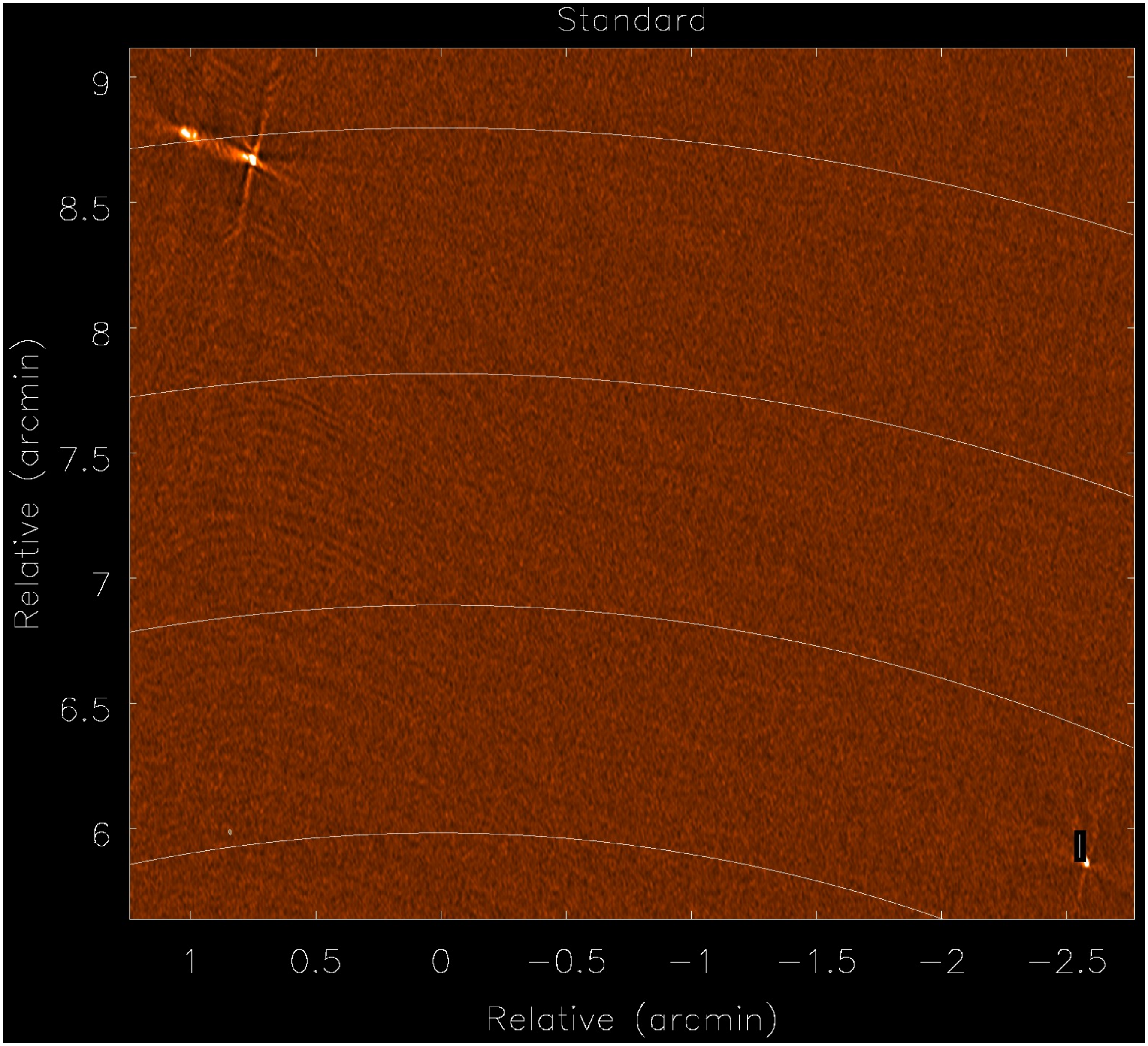
    - Construct D, such that

$$D_{ij}^T * A_{ij} \approx \text{Time/Freq./Pol. indep.}$$

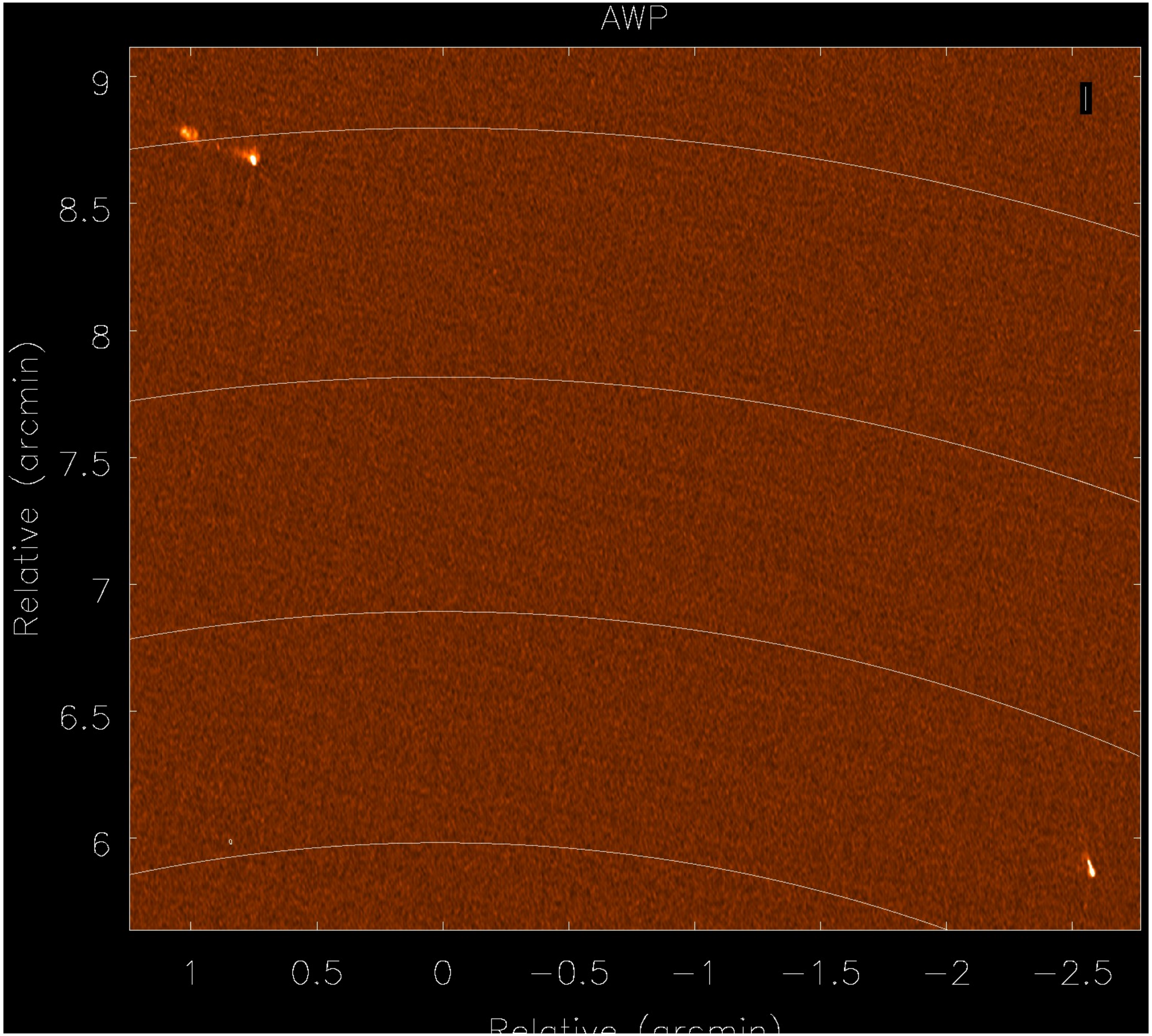
    - Imaging:

$$I = F^{-1} \sum_{ij} D_{ij}^T * V_{ij}^{Obs} = F^{-1} \frac{\sum_{ij} D_{ij}^T * A_{ij} * V_{ij}^o + D_{ij}^T * N_{ij}}{Normalization}$$

S-Band A-array imaging: Standard gridder

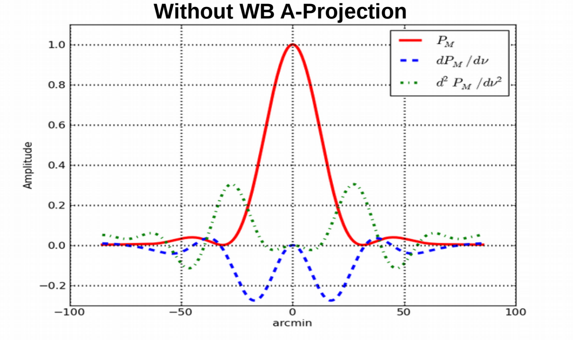


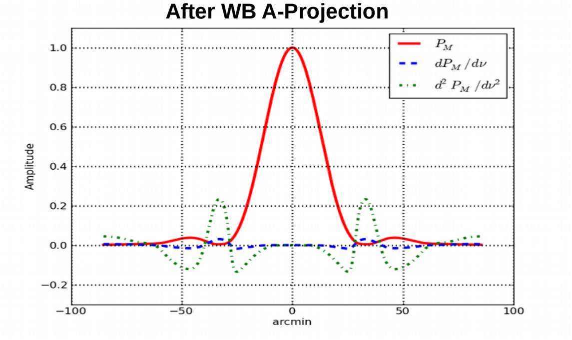
S-Band A-array imaging: AWP gridder



Wide-band Wide-field Imaging

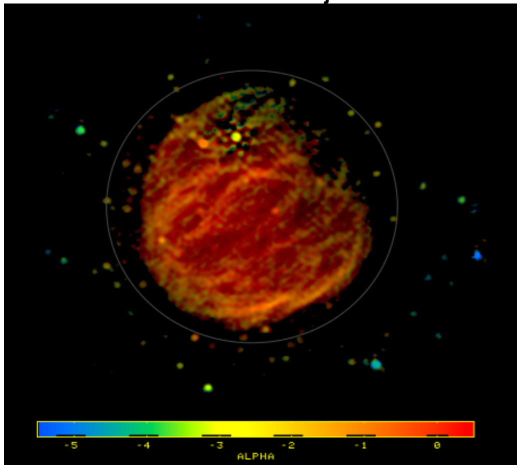
- WB A-Projection + MT-MFS
  - WB A-Projection for PB



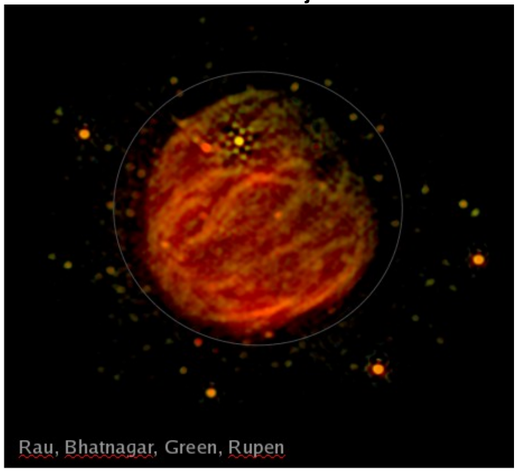


  - MT-MFS for sky
    - **Reconstructed spectral index increases with distance from the center without PB correction**

Without WB A-Projection



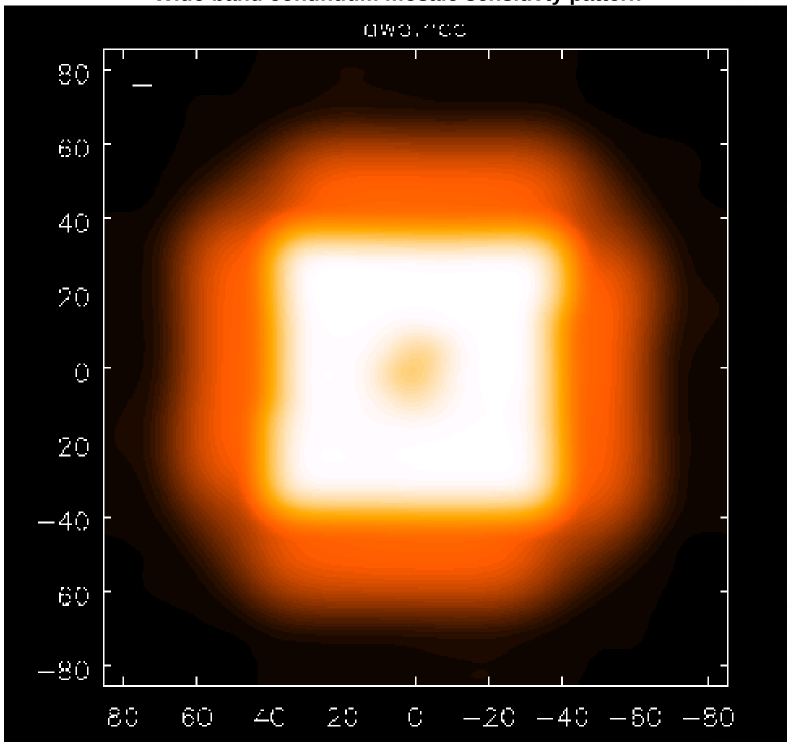
After WB A-Projection



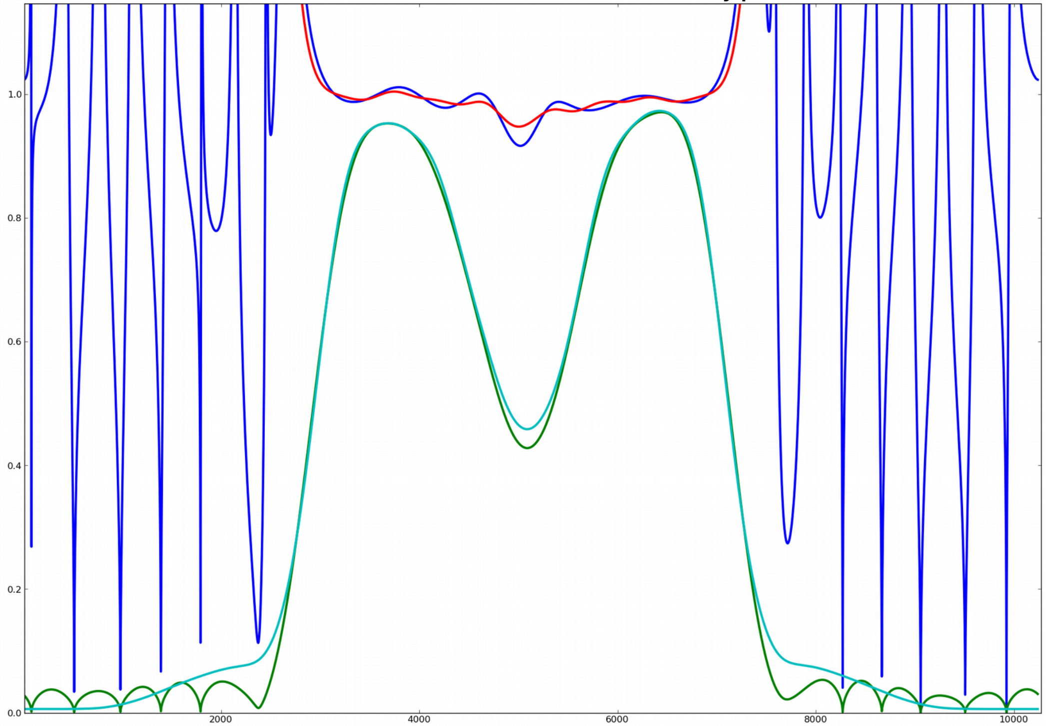
Wide-band sensitivity pattern

- Wide-band mosaic continuum sensitivity pattern

Wide-band continuum mosaic sensitivty pattern



Slice across wide-band continuum mosaic sensitivty pattern

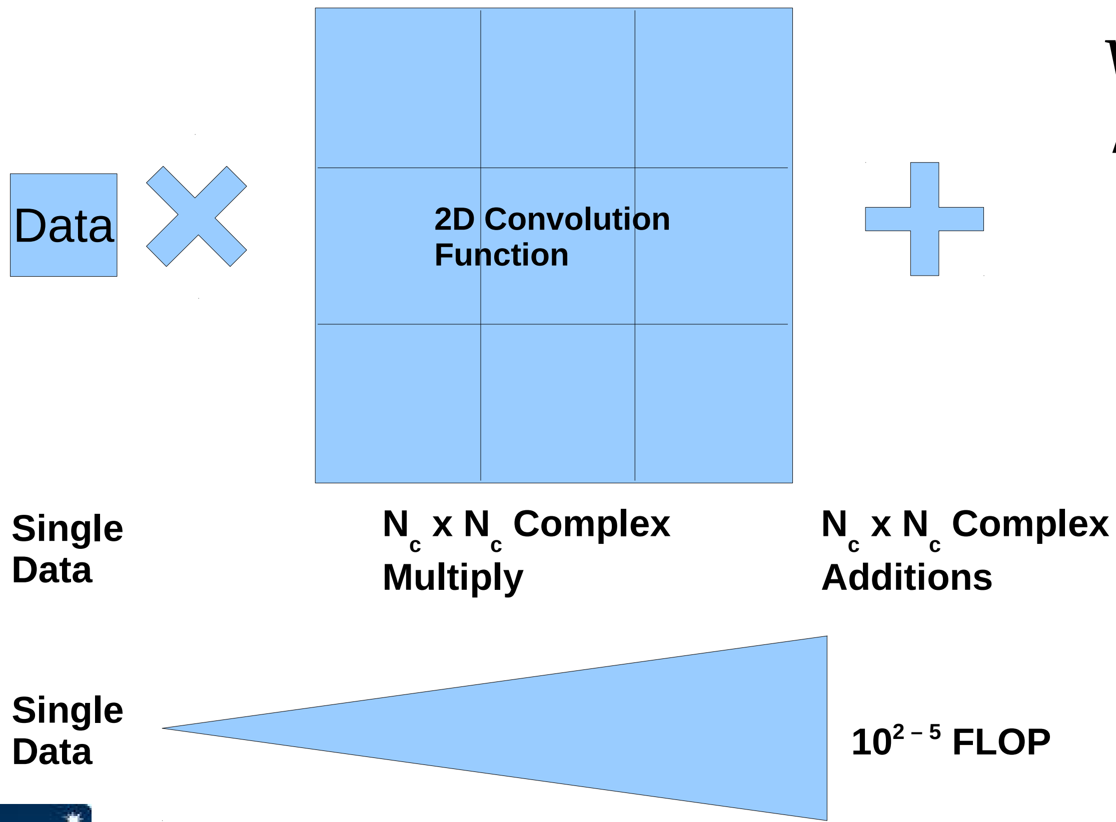


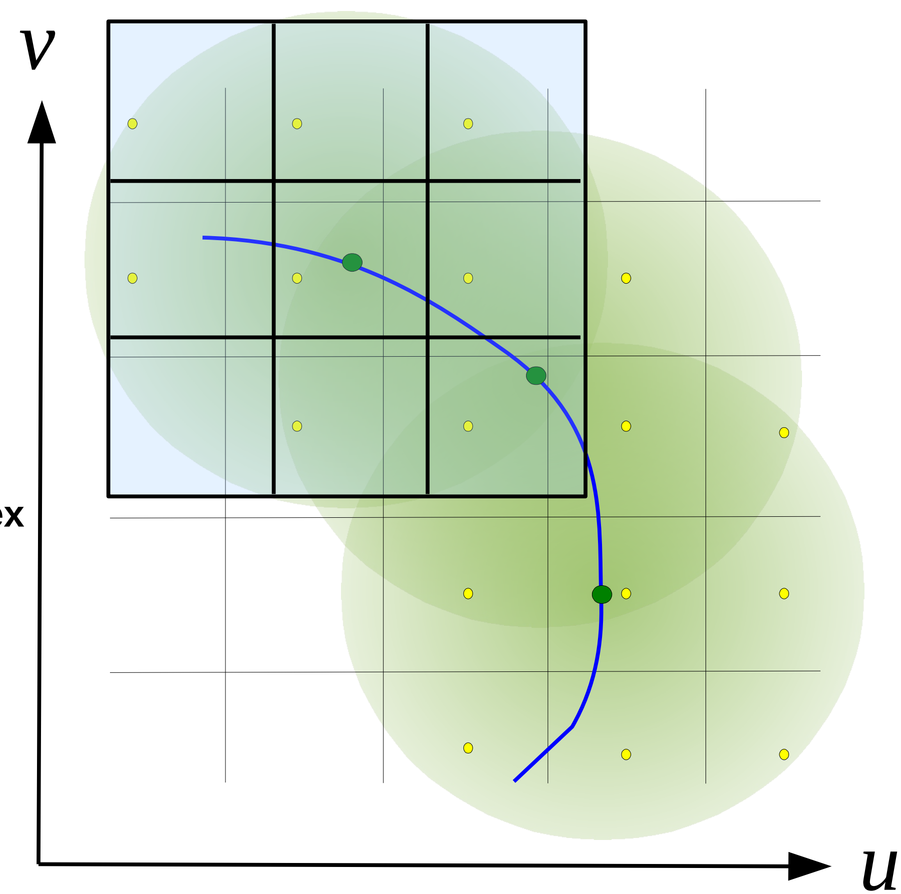
- Red and light green curves are from AWP gridder
- Blue and dark green curvers are from mosaic gridder

- Red and blue curves are ratio of PB with conjbeams=True and conjbeams=False

Gridding: Computations

- Gridding/de-gridding: 2D interpolation via convolutional resampling
- 2D convolution functions ← → 2D weighting functions

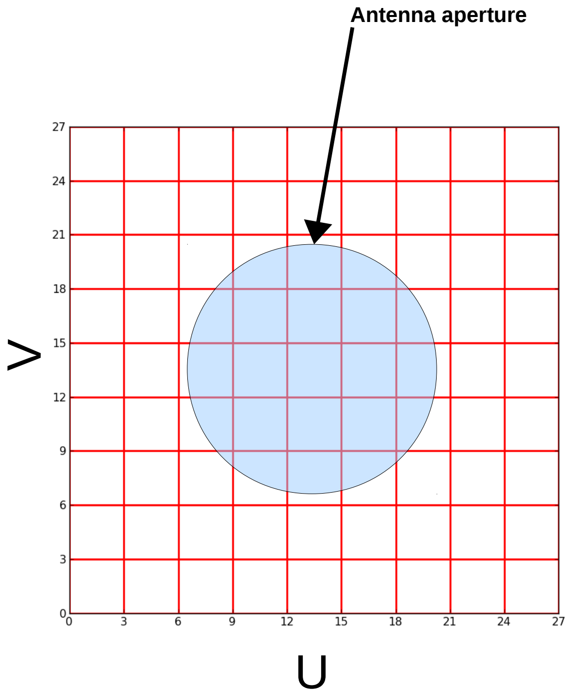




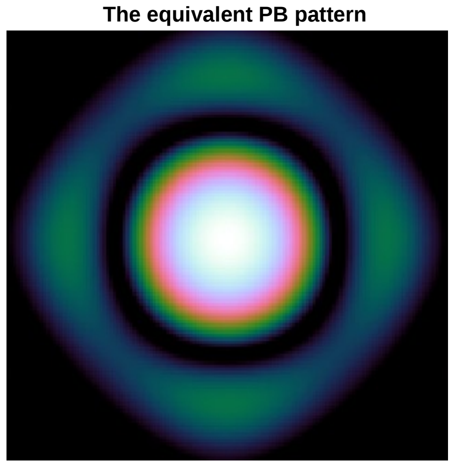
N_{data} x 12 N_C^2 N_{data} = $O(10^{10 -- 12})$

WF imaging: A-Projection

- WF imaging needs larger convolution functions (CF)



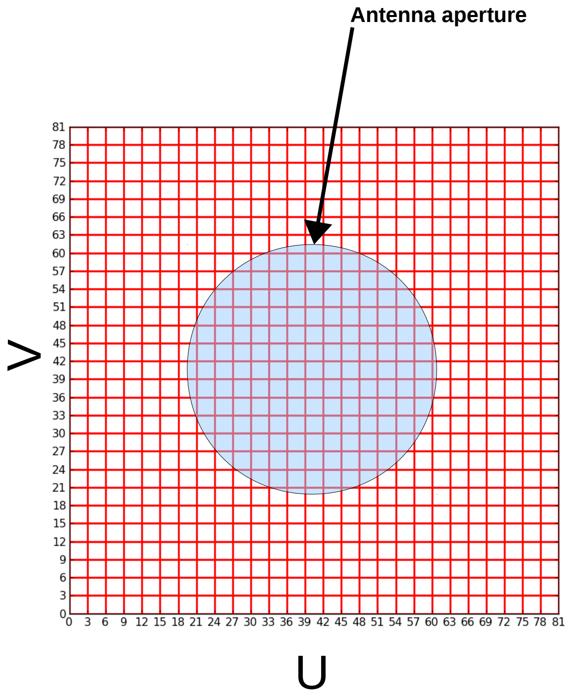
Number of uv-pixel
across antenna aperture



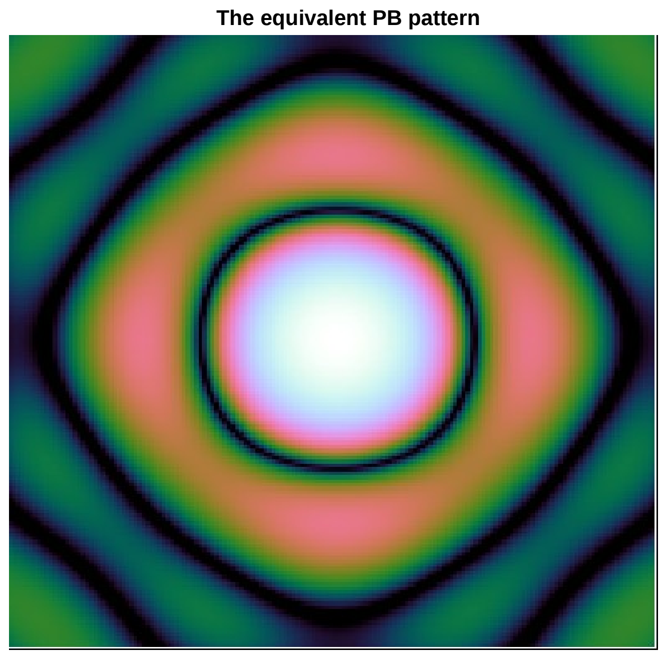
Include the first sidelobe (few%)

WF imaging: A-Projection

- WF imaging needs larger convolution functions (CF)



Number of uv-pixel
across antenna aperture



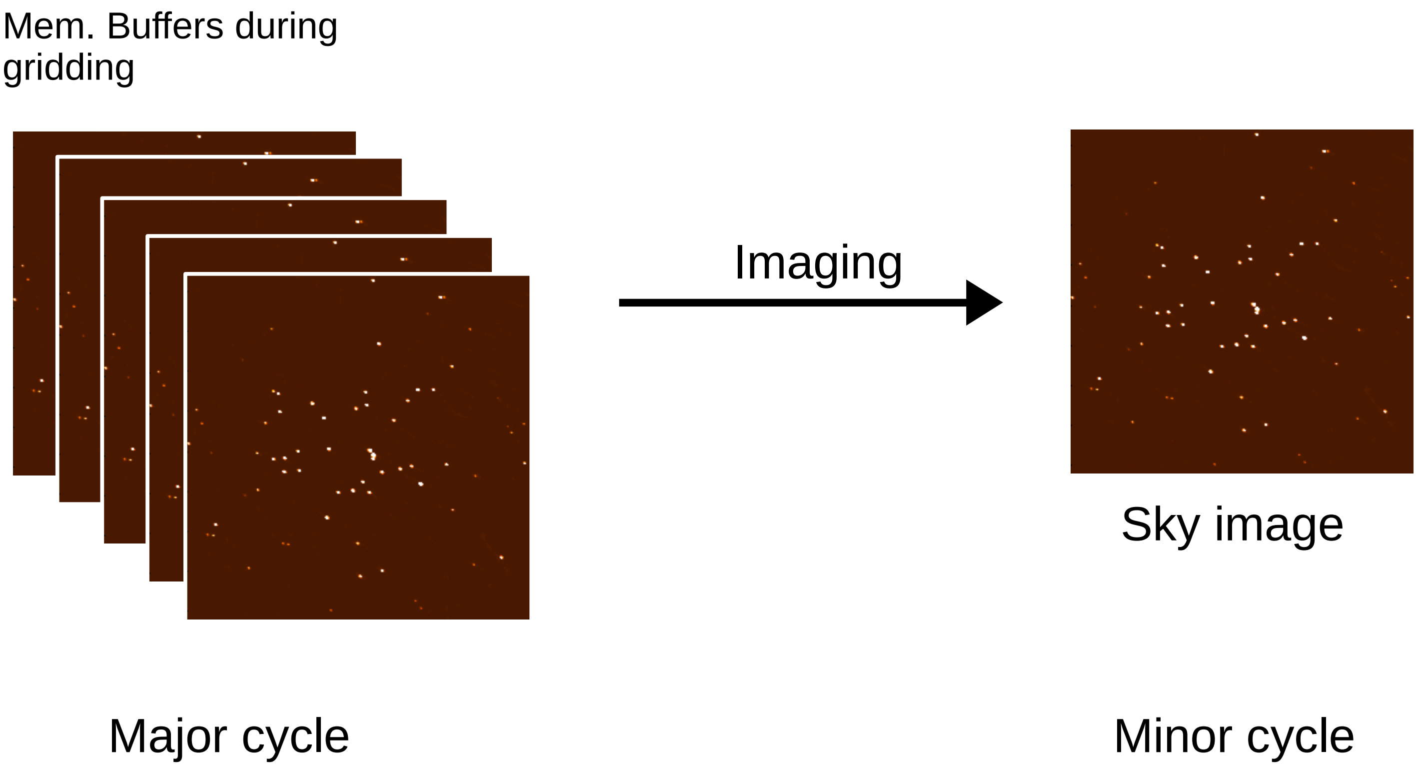
..beyond the first sidelobe

Imaging Memory footprint

- Each sky-image of size N_x x N_y requires
  - 2 x Complex x (N_x x N_y)+ (N_x x N_y) = **5 x (N_x x N_y) floats**

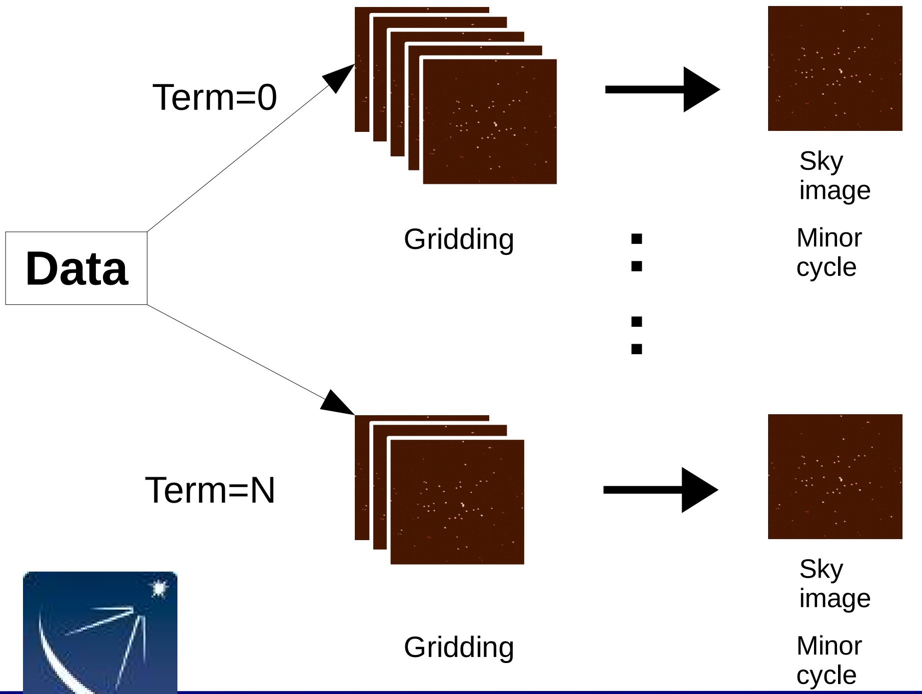
Imaging Memory footprint

- Each sky-image of size N_x x N_y requires
  - 2 x Complex x (N_x x N_y)+ (N_x x N_y) = **5 x (N_x x N_y) floats**



MT-MFS: Higher memory footprint

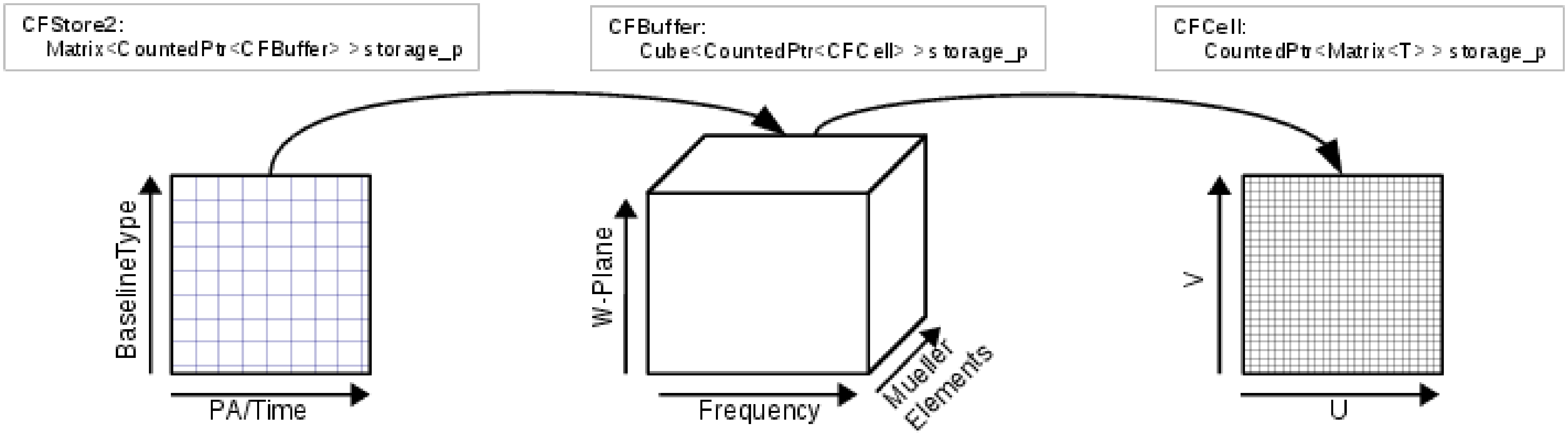
- WB A-Projection: N_A x N_{SPW} (order 10x increase in CF memory footprint)
- MS-MFS
  - Compute load: Gridding for N_{terms} images+ Convolution of large images
  - Memory: Multiple minor-cycle images (N_{scales})
  - Total images (each of size N_x x N_y) : N^2_{terms} x N^2_{scales}



Memory storage for: N^2_{terms} x N^2_{scales}

Compute convolutions of images

Convolution Functions



- WB AW-Projection needs 2x 2 x N_w x N_{SPW} complex-valued functions
- N_{SPW} = 16, N_w = 128. Total CFs = 8K.
- Saved in a CFCache on the disk.
- Size: $(\text{SupportSize})^2$ x Oversampling x 2 x SizeOfFloat
  - $[4 – O(100)]^2$ x 20 x 2 x 4 x N_{CF} bytes = O(15-20GB) per SPW

Convolution Functions

```
CASA <6>: execfile("/users/sbhatnag/Scripts/checkcfc.py")

CASA <7>: checkcfc('cfcache.prediction.im10000conjbeams_True', 16, 64)
0| 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6 6 6 6 7 7 8 8 9 9 10 10 11 11 12 12 12 12 12 13 13 14 12 14 14 15 16 17 17 18 19 19 20 21 22 23 23 24 25 26 27 28 28 29 30 Total lazy-fill size:  439982836.0 B 13516
1| 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 6 6 6 7 7 7 8 8 9 10 10 10 11 11 12 12 13 13 12 13 13 14 14 15 15 15 16 16 17 18 18 19 20 21 21 22 23 24 24 25 26 27 28 29 Total lazy-fill size:  411898444.0 B 12467
2| 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 6 6 6 6 7 7 7 8 8 9 9 10 10 11 11 11 12 13 13 13 13 13 13 14 14 15 15 16 15 16 17 17 18 19 19 20 21 22 22 23 24 25 25 26 27 Total lazy-fill size:  383527380.0 B 11441
3| 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 7 7 7 7 8 8 9 9 10 10 11 11 11 12 12 13 14 13 13 13 14 14 15 15 16 16 16 17 17 18 19 19 20 21 21 22 23 24 24 25 26 Total lazy-fill size:  361747036.0 B 10711
4| 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 7 7 7 8 8 8 9 9 10 10 11 11 12 12 13 13 14 14 15 14 14 14 15 15 16 16 17 17 17 18 18 19 20 20 21 22 22 23 24 25 Total lazy-fill size:  346504332.0 B 10125
5| 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 8 8 8 8 9 10 10 11 11 12 12 12 13 13 14 14 15 14 14 14 15 15 16 16 17 18 18 16 18 19 19 20 21 21 22 23 23 Total lazy-fill size:  328226964.0 B 9529
6| 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 8 8 8 8 9 9 10 10 11 11 12 12 13 13 14 15 15 15 16 16 16 16 15 16 17 17 18 18 19 19 20 19 20 21 21 22 23 Total lazy-fill size:  333836556.0 B 9667
7| 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 7 7 7 7 7 8 8 8 9 9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 17 16 17 17 18 18 19 19 20 20 20 20 21 22 Total lazy-fill size:  324689972.0 B 9350
8| 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 7 7 7 7 7 8 8 8 9 9 9 9 10 11 11 12 12 13 13 14 14 14 15 15 16 16 17 17 17 17 17 17 18 18 19 19 20 20 21 21 21 Total lazy-fill size:  319928364.0 B 9169
9| 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8 9 9 9 10 10 11 11 12 12 13 13 14 14 15 15 15 16 17 17 18 18 19 18 17 18 18 19 19 20 20 21 21 Total lazy-fill size:  316565412.0 B 9074
10| 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 8 8 8 8 8 9 9 9 10 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 18 19 18 18 19 19 20 20 21 Total lazy-fill size:  311490780.0 B 8873
11| 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 9 9 9 10 10 10 10 11 11 12 13 13 14 14 15 15 15 16 16 17 17 18 18 19 19 20 19 18 19 19 20 20 Total lazy-fill size:  308449676.0 B 8785
12| 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 9 9 9 9 10 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20 21 20 20 19 20 Total lazy-fill size:  313597148.0 B 8926
13| 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 8 9 9 9 9 10 10 10 11 11 11 12 12 13 13 14 14 15 15 16 16 17 18 18 19 20 20 20 20 21 21 22 21 Total lazy-fill size:  318544796.0 B 9096
14| 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 9 9 9 9 9 10 10 10 11 11 11 12 12 12 13 13 14 15 15 15 16 17 17 17 18 19 20 19 20 20 21 21 22 Total lazy-fill size:  309371412.0 B 8804
15| 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 9 9 9 9 9 10 10 10 10 11 11 12 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 20 21 21 20 21 21 Total lazy-fill size:  306764828.0 B 8708
Total CFC size: 5435125936.0 B

CASA <8>:
```

- Peak CFC memory footprint: Max. of the "lazy-fill size"
- Compute load scaling $\sum_{w,\nu} N_{vis}(w,\nu)\times N^2_{support}(w,\nu)$
    - The second number is $\sum_{w} N^2_{support}(w,\nu)$

tclean interface

```
gridder          = 'awproject'     #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     wprojplanes =          1      #  Number of distinct w-values for convolution functions
     normtype    = 'flatnoise'     #  Normalization type (flatnoise, flatsky,pbsquare)
     psterm      =      False      #  Use prolate spheroidal during gridding
     aterm       =       True      #  Use aperture illumination functions during gridding
     cfcache     = 'cube_8K.cf'    #  >Convolution function cache directory name
     computepastep =    360.0      #  At what parallactic angle interval to recompute AIFs (deg)
     rotatepastep  =    360.0      #  At what parallactic angle interval to rotate nearest AIF (deg)
     wbawp       =       True      #  Use wideband A-terms
     conjbeams   =      False      #  Use conjugate frequency for wideband A-terms
     pblimit     =      0.001      #  >PB gain level at which to cut off normalizations
```

- ***cfcache***: Name of the disk CFCache. CF computation is triggered if CFC is not found (or is empty?). w and freq quantization, *aterm, psterm, computepastep, conjbeams* settings determined from existing CFC.
  - Can be reused for the same i*msize, cellsize,* and the above parameters are unchanged.
    - **Code will not detect an invalid CFC.**

- ***aterm, psterm, wprojplanes***: In general, CF = PS * A * W

Operation	aterm	psterm	wprojplanes	CF
AW-Projection	True	T or F	>1	PS*A*W or A*W
A-Projection	True	T or F	1	PS*A or A
W-Projection	False	True	>1	PS*W
Standard	False	True	1	PS

  - **At least one out of *aterm* and *psterm* needs to be set to True**

tclean interface

```
gridder          = 'awproject'     #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     wprojplanes =           1     #  Number of distinct w-values for convolution functions
     normtype    = 'flatnoise'     #  Normalization type (flatnoise, flatsky,pbsquare)
     psterm      =       False     #  Use prolate spheroidal during gridding
     aterm       =        True     #  Use aperture illumination functions during gridding
     cfcache     = 'cube_8K.cf'    #  >Convolution function cache directory name
     computepastep =     360.0     #  At what parallactic angle interval to recompute AIFs (deg)
     rotatepastep  =     360.0     #  At what parallactic angle interval to rotate nearest AIF (deg)
     wbawp       =        True     #  Use wideband A-terms
     conjbeams   =       False     #  Use conjugate frequency for wideband A-terms
     pblimit     =       0.001     #  >PB gain level at which to cut off normalizations
```

- ***conjbeams***: Correct for frequency dependence of the PB
  - Use the A-term at $\nu^*=\sqrt{2\nu_{ref}^2-\nu^2}$ when imaging data at ν

    $I(\nu) = F\sum_\nu A^{M^T}(\nu^*) * V^{obs}(\nu) = F \sum_\nu A^{M^T}(\nu^*) * A^o(\nu) * V^o(\nu)$

- ***wbawp***: Computes one CF per SPW if set to True. Else one CF for the entire band.
  - **When *wbawp*=False, conjbeams setting is irrelevant.**

- ***computepastep***: PA-step to trigger computation of a new CF-cube (w,freq,Pol).
  - Value of 360 ==> Compute for only the first PA value

- ***rotatepastep***: PA-step to trigger in-memory rotation of the PA-cube.
  - Value of 360 ==> never rotate
  - Trade-off between computation vs CFC size

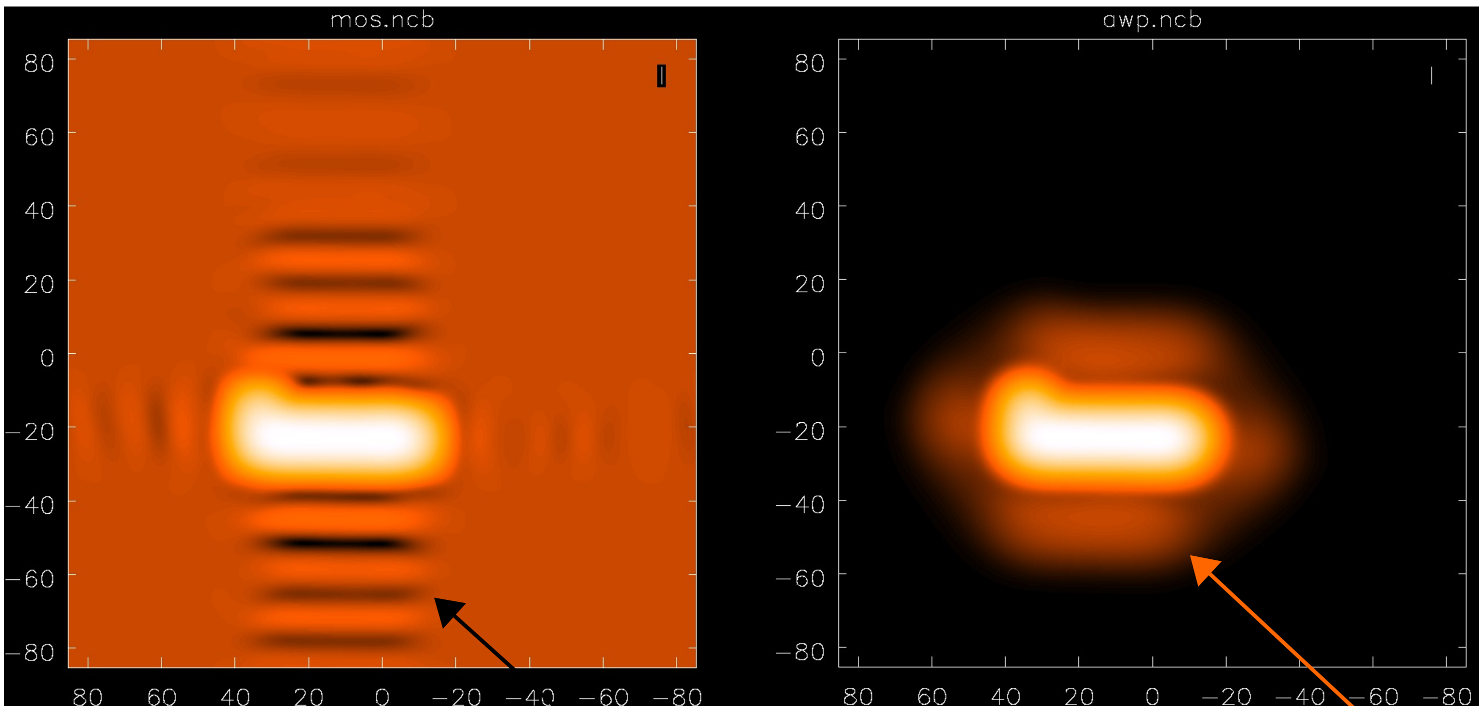
tclean interface

```
gridder             = 'awproject'       #  Gridding options (standard, wproject, widefield, mosaic, awproject)
     wprojplanes    =          1        #  Number of distinct w-values for convolution functions
     normtype       = 'flatnoise'       #  Normalization type (flatnoise, flatsky,pbsquare)
     psterm         =      False        #  Use prolate spheroidal during gridding
     aterm          =       True        #  Use aperture illumination functions during gridding
     cfcache        = 'cube_8K.cf'      #  >Convolution function cache directory name
     computepastep  =      360.0        #  At what parallactic angle interval to recompute AIFs (deg)
     rotatepastep   =      360.0        #  At what parallactic angle interval to rotate nearest AIF (deg)
     wbawp          =       True        #  Use wideband A-terms
     conjbeams      =      False        #  Use conjugate frequency for wideband A-terms
     pblimit        =      0.001        #  >PB gain level at which to cut off normalizations
```

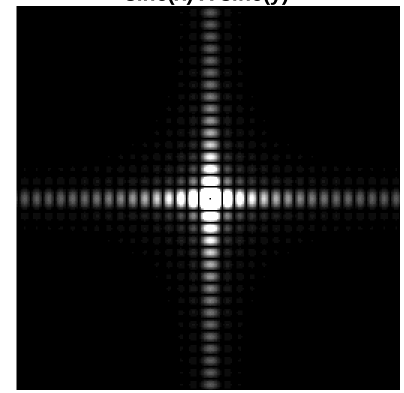
- ***pblimit***: PB model (the *.pb.tt0 image*) considered to be consistent with zero when the gain < *pblimit*. Sky images masked where PB is "zero".
    - PB model = FT [CF * CF]
    - With only PS term, a circular footprint is *not* indicative of antenna PB mask
- ***normtype***: $I = \frac{\sum_\nu \left[P^M(\nu^*)\, P^o(\nu)\ \ I^{sky}(\nu) \ + \ P^M(\nu)\ n \right]}{Norm}$ where $P^M(\nu)$ is the frequency dependent PB model
    - "flatsky": $Norm = \sum_\nu P^M(\nu^*)\, P^M(\nu)$ **[Norm is saved in the *.weight* image in a *tclean* run]**
    - "flatnoise": $Norm = \sqrt{\sum_\nu P^M(\nu^*)\, P^M(\nu)}$

Aliasing and ringing

- Sharp cut-off of the CF leads to ringing in the image domain
  - (True PB) * [sinc(x) X sinc(y)]



sinc(x) X sinc(y)

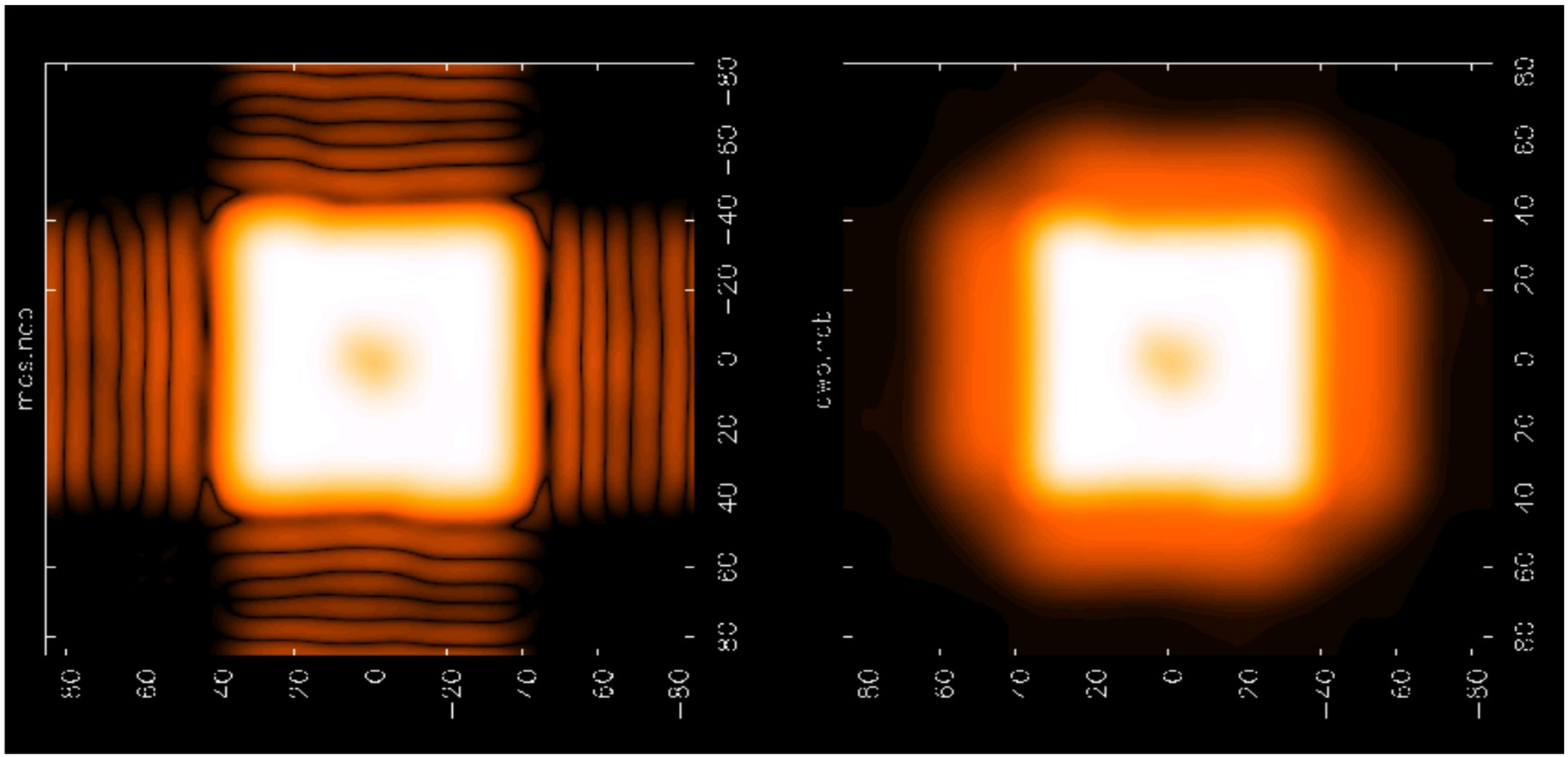


Not PB sidelobes!

PB sidelobes

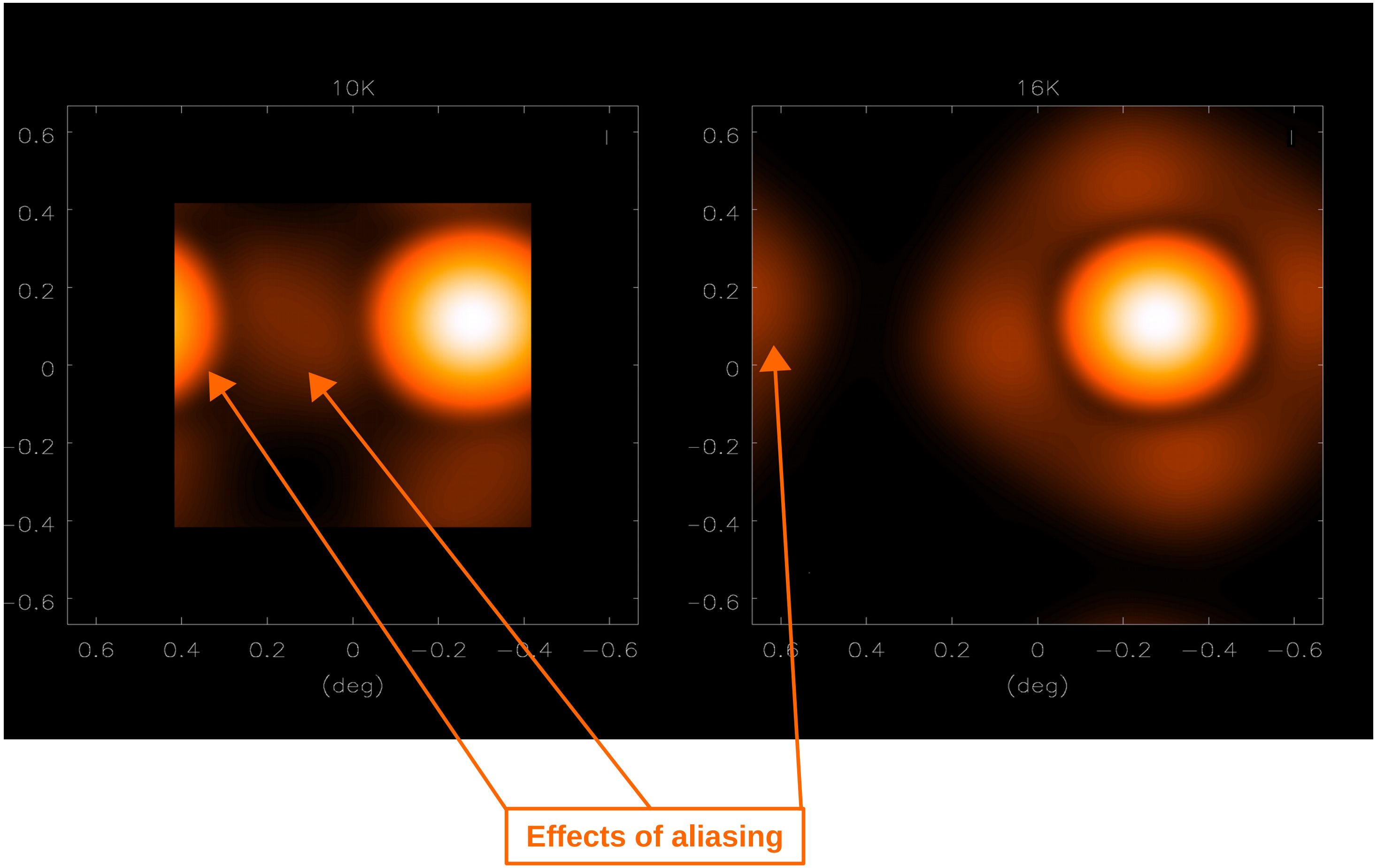
Aliasing and ringing

- Sharp cut-off of the CF leads to ringing in the image domain
  - (True PB) * [sinc(x) X sinc(y)]



Aliasing and ringing

- Sidelobes of the WB PB alias back on the opposite side from an edge pointing if the image size is not large enough.
    - Sources in the aliased regions will lead to deconvolution divergence



tclean interface

- Hidden/internal parameters. Controlled via *~/.aipsrc* or *~/.casarc*
  - Can be controlled via similarly named environment variables (but there is a bug!)

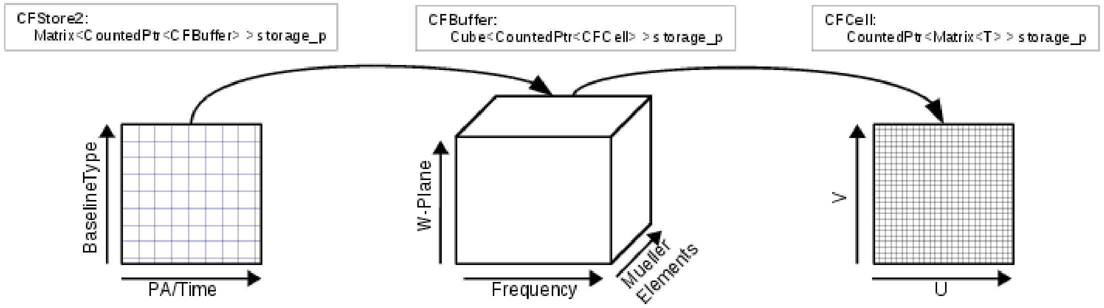
- *Aterm.CONVSIZE*:   Default = 2048
  - Size of the internal buffer used to compute the CFs
  - Increasing it will increase CFC computation time

- *Aterm.OVERSAMPLING*:   Default = 20
  - Oversampling used for computing CFs
  - Larger value improves accuracy, run-time for building CFC and CFC memory footprint
  - When used as W-Projection-only, OS=4 to match *gridder='wproject'*

- *CFCach.LAZYFILL: Default = 1*
  - Activate CF garbage collection/paging
  - Empty the in-memory CFC when data from a new SPW is encountered

The CFCache

- Hold 2D complex-valued CF for all W, Freq., Polarizations, PA and Baseline type. Also holds CFs for computing the weighted sensitivity pattern



- Naming convention
  - CFS_B_P_CF_F_W_Pol.im
    - B: Index for baseline type
    - P: Index for PA slot
    - F: Index for frequency
    - W: Index for w-value
    - Pol: Index for Mueller element

```
tableinfo>go
2018-09-07 16:42:08     INFO            Image data type  : Complex
2018-09-07 16:42:08     INFO    +       Image name       : CFS_0_0_CF_3_288_0.im
2018-09-07 16:42:08     INFO    +       Object name      :
2018-09-07 16:42:08     INFO    +       Image type       : PagedImage
2018-09-07 16:42:08     INFO    +       Image quantity   : Intensity
2018-09-07 16:42:08     INFO    +       Pixel mask(s)    : None
2018-09-07 16:42:08     INFO    +       Region(s)        : None
2018-09-07 16:42:08     INFO
2018-09-07 16:42:08     INFO    +       Spectral  reference : LSRK
2018-09-07 16:42:08     INFO    +       Velocity  type      : RADIO
2018-09-07 16:42:08     INFO    +       Rest frequency      : 3.00218e+09 Hz
2018-09-07 16:42:08     INFO    +       Telescope           : UNKNOWN
2018-09-07 16:42:08     INFO    +       Observer            : UNKNOWN
2018-09-07 16:42:08     INFO    +       Date observation    : UNKNOWN
2018-09-07 16:42:08     INFO    +
2018-09-07 16:42:08     INFO    +       Axis Coord Type     Name       Proj Shape Tile Coord value at pixel        Coord incr Units
2018-09-07 16:42:08     INFO    +       ------------------------------------------------------------------------------------------
2018-09-07 16:42:08     INFO    +       0    0     Linear   UU               2048  256          0  1024.00     -5.729578e+00 lambda
2018-09-07 16:42:08     INFO    +       1    0     Linear   VV               2048  128          0  1024.00      5.729578e+00 lambda
2018-09-07 16:42:08     INFO    +       2    1     Stokes   Stokes              1    1         RR
2018-09-07 16:42:08     INFO    +       3    2     Spectral Frequency           1    1   2.438e+09     0.00 1.898112696840e+09 Hz
2018-09-07 16:42:08     INFO    +                           Velocity                      56337.7     0.00     -1.895425e+05 km/s
2018-09-07 16:42:08     INFO
2018-09-07 16:42:08     INFO    +       Attached miscellaneous Information :
2018-09-07 16:42:08     INFO    +        MiscInfo : Xsupport: Int 50
2018-09-07 16:42:08     INFO    +        MiscInfo : Ysupport: Int 50
2018-09-07 16:42:08     INFO    +        MiscInfo : Sampling: Float 20
2018-09-07 16:42:08     INFO    +        MiscInfo : ParallacticAngle: Double -7.75756
2018-09-07 16:42:08     INFO    +        MiscInfo : MuellerElement: Int 0
2018-09-07 16:42:08     INFO    +        MiscInfo : WValue: Double 194384
2018-09-07 16:42:08     INFO    +        MiscInfo : WIncr: Double 0.426701
2018-09-07 16:42:08     INFO    +        MiscInfo : Name: String "CFS_0_0_CF_3_288_0.im"
2018-09-07 16:42:08     INFO    +        MiscInfo : ConjFreq: Double 3.438e+09
2018-09-07 16:42:08     INFO    +        MiscInfo : ConjPoln: Int 8
2018-09-07 16:42:08     INFO    +        MiscInfo : TelescopeName: String "EVLA"
2018-09-07 16:42:08     INFO    +        MiscInfo : BandName: String "EVLA_S"
2018-09-07 16:42:08     INFO    +        MiscInfo : Diameter: Float 25
2018-09-07 16:42:08     INFO    +        MiscInfo : OpCode: Bool 0
```

- Actual values written in the image header and *MiscInfo* database in the images

The CFCache

- CFC is re-usable. For VLASS-style imaging, can be computed once.
- Can be computed with *parallel=True*
  - Serial:
    - Does a dry-run to determine (a) number of CFs and (b) their parameters
    - Writes them as 2x2 pixel images in the on-disk CFC
  - Parallel:
    - Divide the total CFs equally among the parallel processes
    - Each process computes its share of CFs
    - Currently each process holds its share of CFs in memory till it finishes.
  - Example:
    - Computed 32K CFs in ~1hr using 98 processes on the cluster

- To-Do
  - Fix the bug for *specmode='cube'* and *chanchunks >1*
  - Allow using smaller wprojplanes than what's in the CFC
  - Allow use of existing <u>read-only</u> CFC
  - More aggressive CFC garbage collection
  - Automatically adjust Aterm.CONVSIZE parameter
  - Separate PS from the .pb image. The latter is used for making PB-corrected final images.