CASA's simdata

- One command gets you
  - synthetic visibilities
  - a synthesized deconvolved image
  - analysis of differences between your input and the synthetic output.
- ALMA, (E)VLA, SMA, ATCA, you name it.
- All you need is a model of the sky

Outline

- inputs
  - array configuration
  - model image = “sky truth”
  - spatial and spectral parameters
- outputs
  - UV data
  - noninteractively cleaned image
  - difference from model and fidelity
- calculation of (noiseless) visibilities
- corruption of visibilities
  - thermal noise
  - atmospheric phase noise
  - X-pol and gain drift

Terminology

- colors
  - red = required
  - blue = optional
  - green = not yet fully implemented
- sm = Simulator tool
- simdata = task

Inputs: array configuration

Default

Default | Default | IPy casa/regress | Default

```
# observatory=ALMA
# coordsys=UTM
# datum=SAM56
# zone=19
# hemisphere=S
#UTM-X       UTM_Y       Z      Diam (m)    Pad #
627789.81   7453079.62  5029.4   12.0        1
627806.26   7453082.62  5028.7   12.0        2
627801.31   7453100.27  5029.4   12.0        3
627818.16   7453068.35  5028.3   12.0        4
627848.62   7453047.47  5027.3   12.0        5
627844.52   7453070.45  5027.7   12.0        6
627839.69   7453091.26  5028.1   12.0        7
627833.36   7453104.90  5028.5   12.0        8
627814.26   7453128.29  5029.4   12.0        9
627803.30   7453138.64  5029.8   12.0        10
627861.26   7453055.91  5027.3   12.0        12
627860.31   7453071.14  5027.2   12.0        13
627824.72   7453089.41  5028.6   12.0        15
627840.40   7453118.33  5028.6   12.0        16
627834.85   7453132.37  5028.8   12.0        17
627803.74   7453158.27  5029.8   12.0        19
627817.20   7453107.16  5028.9   12.0        20
627782.36   7453066.47  5029.6   12.0        21
627773.43   7453081.30  5029.9   12.0        22
627762.59   7453069.82  5029.9   12.0        23
627780.13   7453100.26  5029.9   12.0        24
```

Header: obs name & coordinate system

optional

Inputs: array configuration

Default

Default | Default | IPy casa/regress | Default

```
# observatory=ALMA
# coordsys=UTM
# datum=SAM56
# zone=19
# hemisphere=S
#UTM-X       UTM_Y       Z       Diam (m)    Pad #
627789.81   7453079.62  5029.4   12.0        1
627806.26   7453082.62  5028.7   12.0        2
627801.31   7453100.27  5029.4   12.0        3
627818.16   7453068.35  5028.3   12.0        4
627848.62   7453047.47  5027.3   12.0        5
627844.52   7453070.45  5027.7   12.0        6
627839.69   7453091.26  5028.1   12.0        7
627833.36   7453104.90  5028.5   12.0        8
627814.26   7453128.29  5029.4   12.0        9
627803.30   7453138.64  5029.8   12.0        10
627861.26   7453055.91  5027.3   12.0        12
627860.31   7453071.14  5027.2   12.0        13
627824.72   7453089.41  5028.6   12.0        15
627840.40   7453118.33  5028.6   12.0        16
627834.85   7453132.37  5028.8   12.0        17
627803.74   7453158.27  5029.8   12.0        19
627817.20   7453107.16  5028.9   12.0        20
627782.36   7453066.47  5029.6   12.0        21
627773.43   7453081.30  5029.9   12.0        22
627762.59   7453069.82  5029.9   12.0        23
627780.13   7453100.26  5029.9   12.0        24
```

Header: obs name & coordinate system
Obs name has to be in data repository
Need to better document how users can add

optional

Supported coordinate systems:
UTM
ITRF earth-centered
LOC local tangent plane offsets from COA
GEO geodetic lat/lon

Delivered with CASA:
ALMA, ACA, ALMA early science, ALMA CSV, EVLA, CARMA, SMA, ATNF, MeerKat

Set array configuration

IPy casa/regress (101,36)

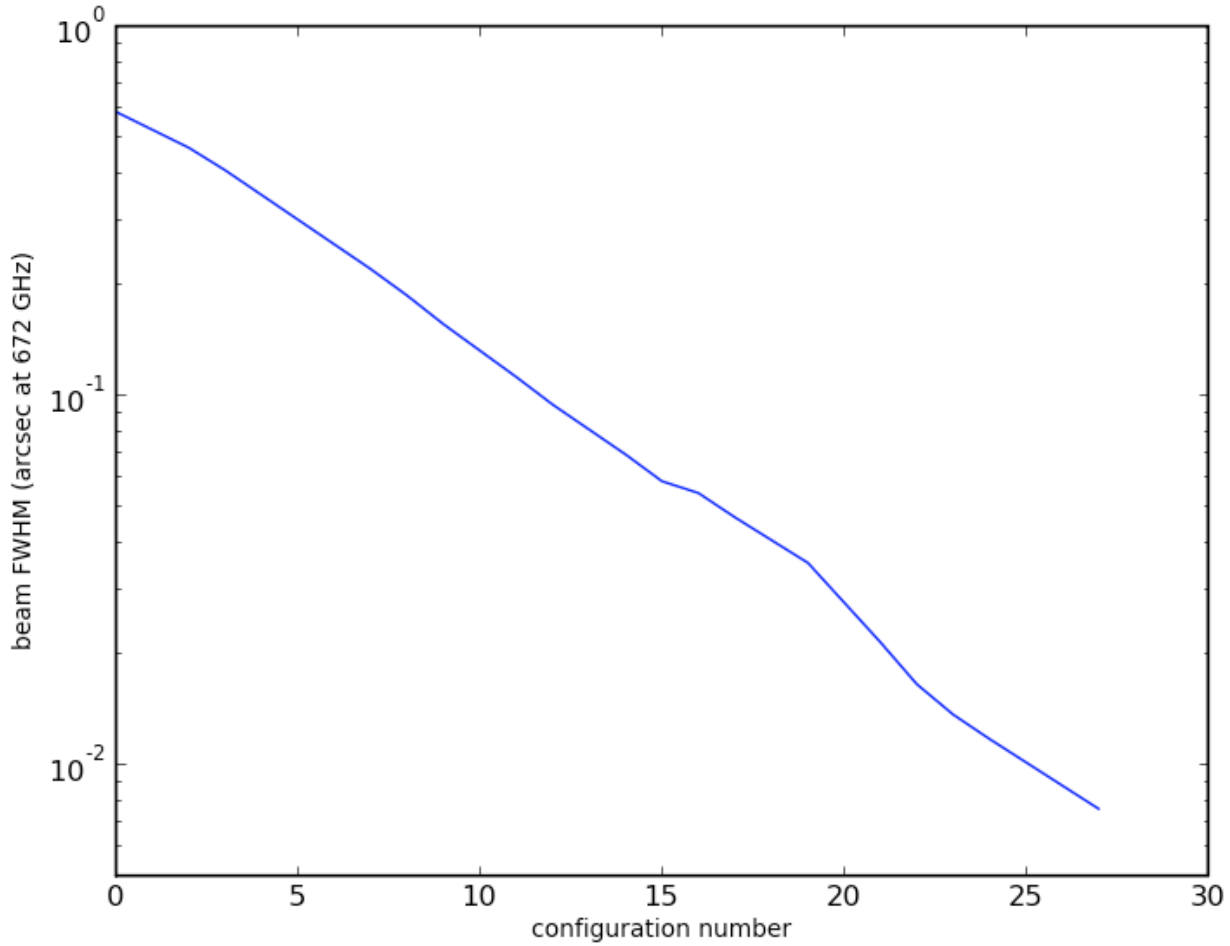
```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =      False        #  scale model coordinates to output parameters
inbright            = 'unchanged'       #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =         ''        #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =       'no'        #  graphically verify parameters [yes|no|only]
                                                                          *see help
totaltime           =     '1200s'       #  total time of observation
integration         =      '10s'        #  integration (sampling) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =          1        #  number of channels
                                                                          f pointings
                                                                          to pointingspacing
                                                                          minor cycles
                                                                          ities.
                                                                          ng simdata(...)
CASA <47>:
```

Many are delivered with CASA, probably in /opt/casa/data/alma/simmos.

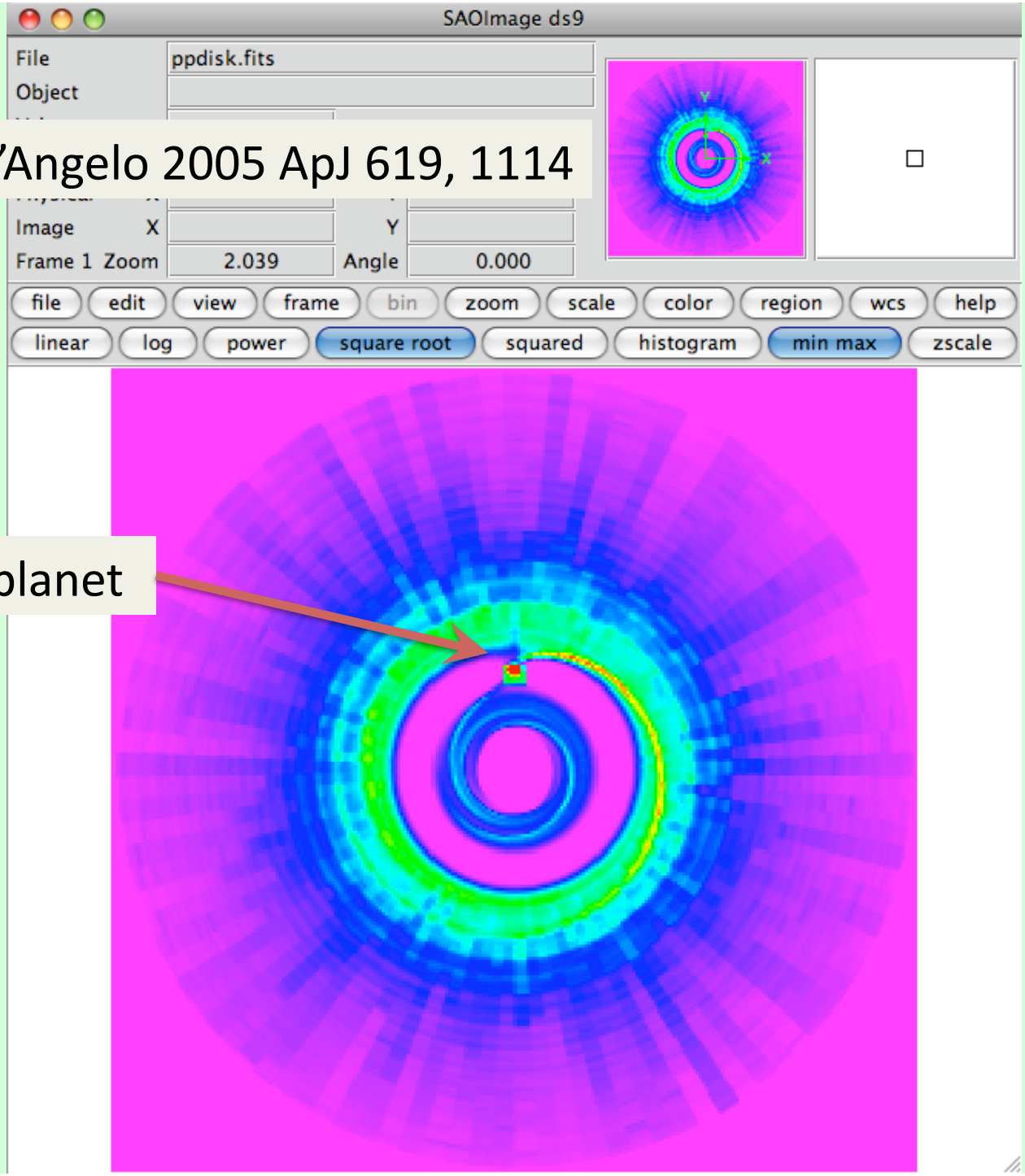
```
CASA <47>: !head /Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg
# observatory=ALMA
# coordsys=UTM
# datum=SAM56
# zone=19
# hemisphere=S
#UTM-X        UTM_Y         Z      Diam (m)    # Pad #
627801.31   7453100.27   5029.4   12.0        # 3
627762.59   7453069.82   5029.9   12.0        # 23
627808.00   7453045.89   5028.3   12.0        # 43
628103.00   7453218.00   5022.2   12.0        # 102
```

Set array configuration

On the Simulator wiki: Development planned for simdata to suggest a config given a synthesized beam size.



From Wolf & D'Angelo 2005 ApJ 619, 1114



Set up model

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =      False        #  scale model coordinates to output parameters
inbright            = 'unchanged'       #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =          ''       #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =       'no'        #  graphically verify parameters [yes|no|only]
                                                                    of observation *see help
                                                                    tion
                                                                    ) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =           1       #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'       #  spacing in between beams in mosaic
                                                                    e btw. pointings and edge, relative to pointingspacing
                                                                    ut cell/pixel size
                                                                    ut image size in pixels (x,y)
                                                                    mum number of iterations
threshold           =   '0.0mJy'        #  flux level (+units) to stop cleaning
psfmode             =    'clark'        #  method of PSF calculation to use during minor cycles
weighting           =  'natural'        #  weighting to apply to visibilities
uvtaper             =      False        #  apply additional uv tapering of  visibilities.
stokes              =        'I'        #  Stokes params to image
noise_thermal       =      False        #  add thermal noise
fidelity            =      False        #  Calculate fidelity images
display             =      False        #  Plot simulation result images,figures
verbose             =      False
async               =      False        #  If true the taskname must be started using simdata(...)

CASA <47>:
```

First, we need a model of the sky.
simdata can use a clean component list or an image (fits or CASA)

Next, you have to decide whether you want to use the World Coordinate System in the image or not.

Set up model

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =      False        #  scale model coordinates to output parameters
inbright            = 'unchanged'       #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =          ''       #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =       'no'        #  graphically verify parameters [yes|no|only]

                                                of observation *see help
                                                tion
                                                ) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =           1       #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'       #  spacing in between beams in mosaic
                                                and edge, relative to pointingspacing
                                                e
imsize              = [192, 192]        #  output image size in pixels (x,y)
niter               =         100       #  maximum number of iterations
threshold           =   '0.0mJy'        #  flux level (+units) to stop cleaning
psfmode             =    'clark'        #  method of PSF calculation to use during minor cycles
weighting           =  'natural'        #  weighting to apply to visibilities
uvtaper             =      False        #  apply additional uv tapering of  visibilities.
stokes              =        'I'        #  Stokes params to image
noise_thermal       =      False        #  add thermal noise
fidelity            =      False        #  Calculate fidelity images
display             =      False        #  Plot simulation result images,figures
verbose             =      False
async               =      False        #  If true the taskname must be started using simdata(...)

CASA <47>:
```

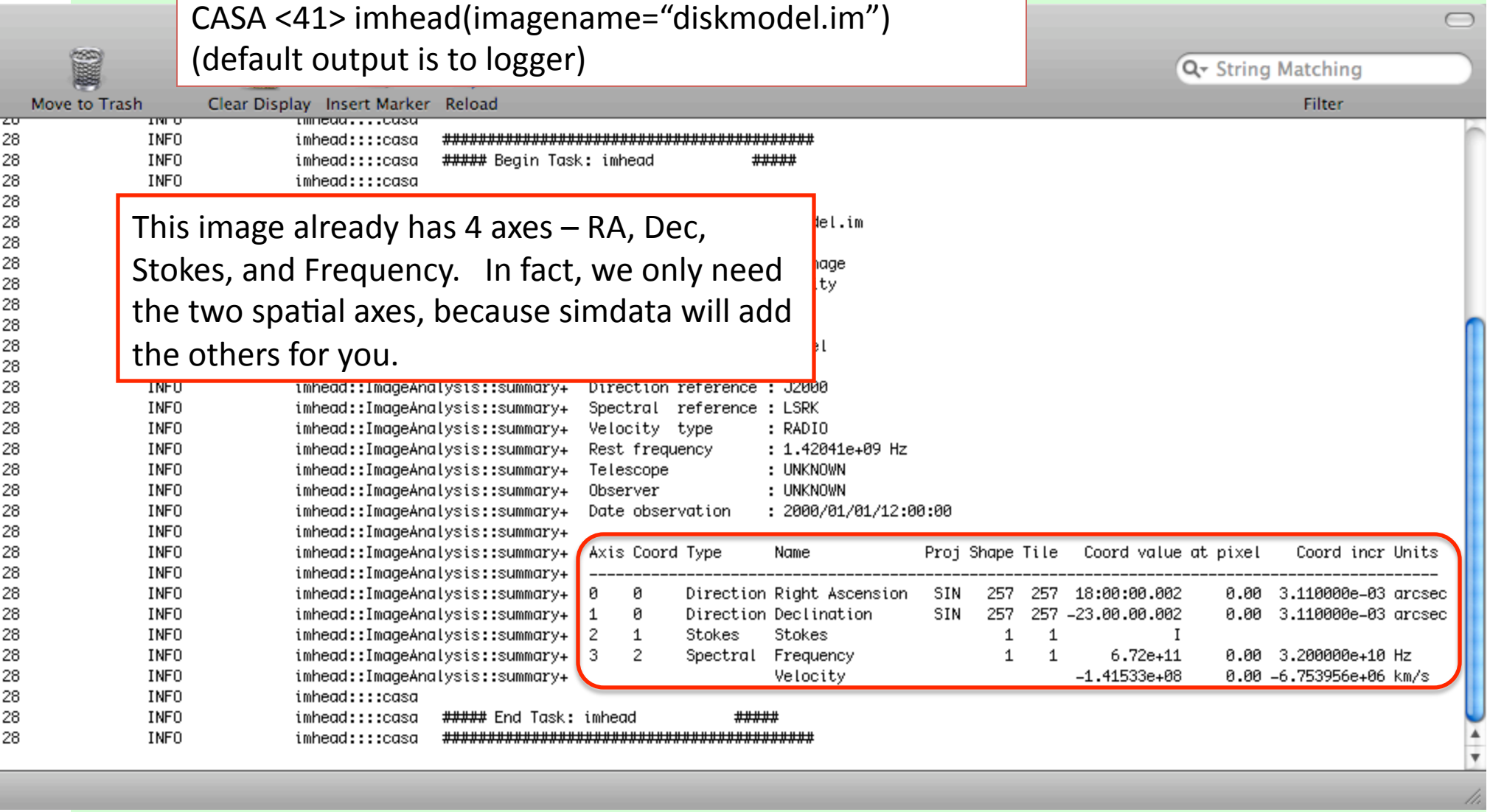
If you're happy with your input WCS, or you can edit it to your satisfaction, leave ignorecoord=False. Yes, it's a double negative.

If you just want to set the flux brightness scale, use "inbright"

What?

Imhead shows what you have:
CASA <41> imhead(imagename="diskmodel.im")
(default output is to logger)

This image already has 4 axes – RA, Dec, Stokes, and Frequency. In fact, we only need the two spatial axes, because simdata will add the others for you.



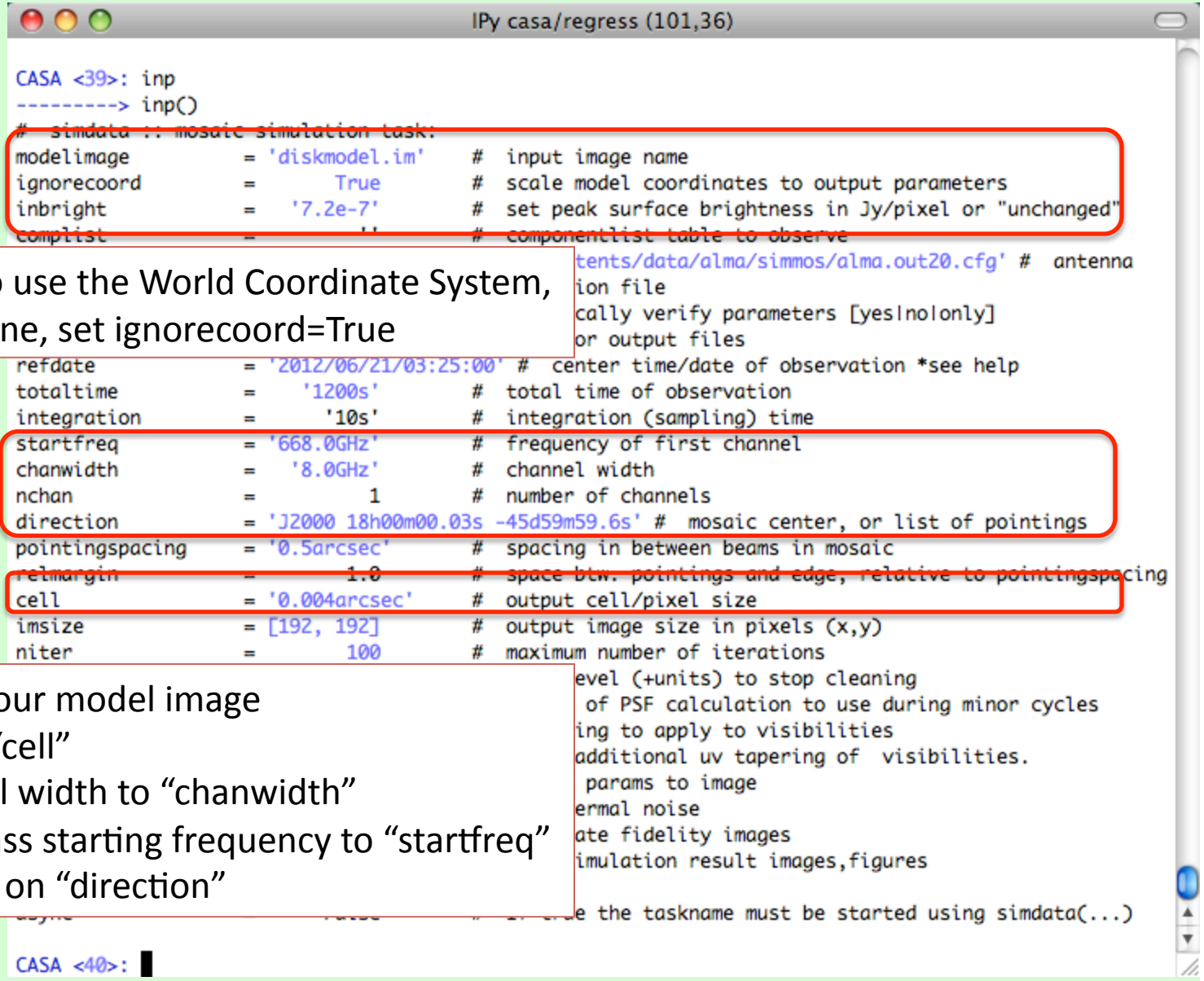

```
28    INFO    imhead::::casa   ##########################################
28    INFO    imhead::::casa   ##### Begin Task: imhead          #####
28    INFO    imhead::::casa
28
28                                                                  del.im
28
28                                                                  nage
28                                                                  ty
28
28
28                                                                  el
28
28    INFO    imhead::ImageAnalysis::summary+  Direction reference : J2000
28    INFO    imhead::ImageAnalysis::summary+  Spectral  reference : LSRK
28    INFO    imhead::ImageAnalysis::summary+  Velocity  type      : RADIO
28    INFO    imhead::ImageAnalysis::summary+  Rest frequency      : 1.42041e+09 Hz
28    INFO    imhead::ImageAnalysis::summary+  Telescope           : UNKNOWN
28    INFO    imhead::ImageAnalysis::summary+  Observer            : UNKNOWN
28    INFO    imhead::ImageAnalysis::summary+  Date observation    : 2000/01/01/12:00:00
28    INFO    imhead::ImageAnalysis::summary+
28    INFO    imhead::ImageAnalysis::summary+  Axis Coord Type      Name             Proj Shape Tile   Coord value at pixel    Coord incr Units
28    INFO    imhead::ImageAnalysis::summary+  ----------------------------------------------------------------------------------------------
28    INFO    imhead::ImageAnalysis::summary+  0    0     Direction Right Ascension   SIN   257  257  18:00:00.002     0.00  3.110000e-03 arcsec
28    INFO    imhead::ImageAnalysis::summary+  1    0     Direction Declination       SIN   257  257 -23.00.00.002     0.00  3.110000e-03 arcsec
28    INFO    imhead::ImageAnalysis::summary+  2    1     Stokes    Stokes                    1    1           I
28    INFO    imhead::ImageAnalysis::summary+  3    2     Spectral  Frequency               1    1     6.72e+11     0.00  3.200000e+10 Hz
28    INFO    imhead::ImageAnalysis::summary+                      Velocity                           -1.41533e+08     0.00 -6.753956e+06 km/s
28    INFO    imhead::::casa
28    INFO    imhead::::casa   ##### End Task: imhead          #####
28    INFO    imhead::::casa   ##########################################
```

What if I don't want my WCS?

If you don't want to use the World Coordinate System, or you don't have one, set ignorecoord=True

Simdata will set your model image

- spatial pixel to "cell"
- spectral channel width to "chanwidth"
- spectral bandpass starting frequency to "startfreq"
- image centered on "direction"



What if I don’t want my WCS?

Remember, you get $project.$modelimage.coord
with simdata’s idea of a CoordinateSystem

You can start a second simulation with that as modelimage.

Set up observation

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =      False        #  scale model coordinates to output parameters
inbright            = 'unchanged'       #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =          ''       #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =       'no'        #  graphically verify parameters [yes|no|only]
project             =     'psim'        #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'        #  total time of observation
integration         =      '10s'        #  integration (sampling) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =          1        #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'       #  spacing in between beams in mosaic
relmargin           =       1.0         #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'     #  output cell/pixel size
imsize              = [192, 192]        #  output image size in pixels (x,y)
                                                                  g minor cycles
                                                                  ilities.
async               =      False        #  If true the taskname must be started using simdata(...)

CASA <47>:
```

"direction" can be a list, if you know what pointings you want, or a center point, in which case simdata will try to fill your area with a mosaic of pointings.

* Source – cal – source sequence is possible in the tool only at the moment

Set up observation

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'     #  input image name
ignorecoord         =      False         #  scale model coordinates to output parameters
inbright            = 'unchanged'        #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =         ''         #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                         #   position file
checkinputs         =       'no'         #  graphically verify parameters [yes|no|only]
project             =     'psim'         #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'         #  total time of observation
integration         =      '10s'         #  integration (sampling) time
startfreq           = '668.0GHz'         #  frequency of first channel
chanwidth           =   '8.0GHz'         #  channel width
nchan               =          1         #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'        #  spacing in between beams in mosaic
relmargin           =       1.0          #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'      #  output cell/pixel size
imsize              = [192, 192]         #  output image size in pixels (x,y)
niter               =       100          #  maximum number of iterations
                                         ) to stop cleaning
                                         ulation to use during minor cycles
                                         to visibilities
                                         tapering of  visibilities.
                                         nage
fidelity            =      False         #  Calculate fidelity images
display             =      False         #  Plot simulation result images,figures
verbose             =      False
async               =      False         #  If true the taskname must be started using simdata(...)

CASA <47>:
```

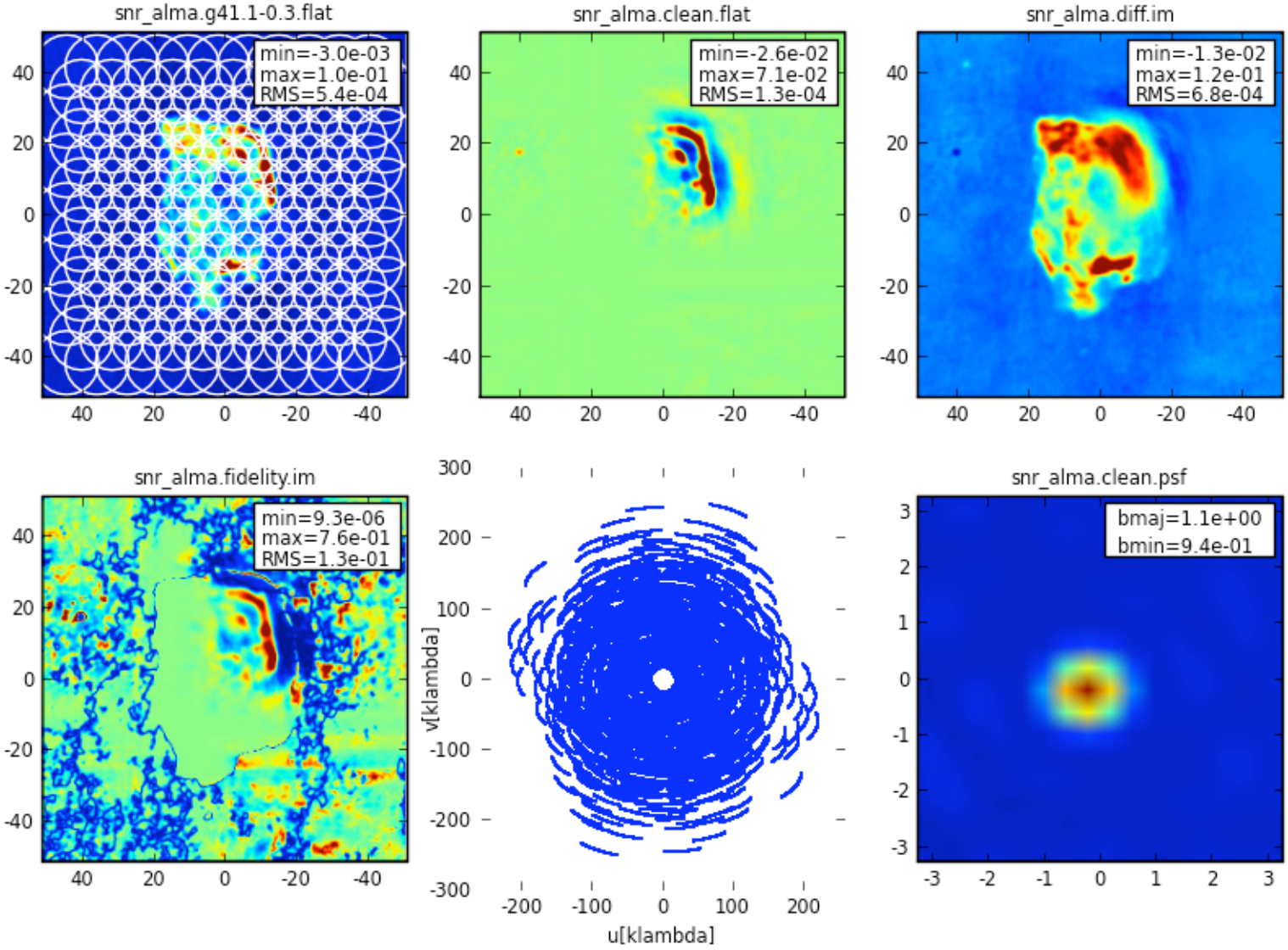
Mosaic positions will be in a hex pattern separated by “pointingspacing”, with a border around the edge of “relmargin” pointing spacings

Set up observation

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =      False        #  scale model coordinates to output parameters
inbright            = 'unchanged'       #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =         ''        #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =       'no'        #  graphically verify parameters [yes|no|only]
project             =     'psim'        #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'        #  total time of observation
integration         =      '10s'        #  integration (sampling) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =          1        #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'       #  spacing in between beams in mosaic
relmargin           =        1.0        #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'     #  output cell/pixel size
imsize              = [192, 192]        #  output image size in pixels (x,y)
niter               =        100        #  maximum number of iterations
threshold           =   '0.0mJy'        #  flux level (+units) to stop cleaning
                                        ulation to use during minor cycles
                                        to visibilities
                                        tapering of  visibilities.
                                        age

                                        images
                                        ult images,figures

async               =      False        #  If true the taskname must be started using simdata(...)

CASA <47>:
```

Go ahead and set the size and pixel size of your output image even if you only want to create a measurement set. Simdata will try to build you a mosaic based on the sky you want out, not the sky you put in


snr_alma.g41.1-0.3.flat
min=-3.0e-03
max=1.0e-01
RMS=5.4e-04
snr_alma.clean.flat
min=-2.6e-02
max=7.1e-02
RMS=1.3e-04
snr_alma.diff.im
min=-1.3e-02
max=1.2e-01
RMS=6.8e-04
snr_alma.fidelity.im
min=9.3e-06
max=7.6e-01
RMS=1.3e-01
v[klambda]
u[klambda]
snr_alma.clean.psf
bmaj=1.1e+00
bmin=9.4e-01

Set up observation

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'     #  input image name
ignorecoord         =      False         #  scale model coordinates to output parameters
inbright            = 'unchanged'        #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =         ''         #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                         #   position file
checkinputs         =       'no'         #  graphically verify parameters [yes|no|only]
project             =     'psim'         #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'         #  total time of observation
integration         =      '10s'         #  integration (sampling) time
startfreq           = '668.0GHz'         #  frequency of first channel
chanwidth           =   '8.0GHz'         #  channel width
nchan               =          1         #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'        #  spacing in between beams in mosaic
relmargin           =       1.0          #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'      #  output cell/pixel size
imsize              = [192, 192]         #  output image size in pixels (x,y)
niter               =       100          #  maximum number of iterations
threshold           =   '0.0mJy'         #  flux level (+units) to stop cleaning
```

Set up spectral information. If you are interested in a 2d (continuum) simulation, set chanwidth to e.g. 4GHz, and nchan=1

* Multiple SPW: some aspects, like corruption, can be handled at the tool level. Simdata task can do one.

```
CASA <47>:
```

Set up what you want to calculate

IPy casa/regress (101,36)

```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'     #  input image name
ignorecoord         =      False         #  scale model coordinates to output parameters
inbright            = 'unchanged'        #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =          ''        #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                         #   position file
checkinputs         =       'no'         #  graphically verify parameters [yes|no|only]
project             =     'psim'         #  root for output files
                                                                    ation *see help
nchan               =          1         #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'        #  spacing in between beams in mosaic
relmargin           =        1.0         #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'      #  output cell/pixel size
imsize              = [192, 192]         #  output image size in pixels (x,y)
niter               =        100         #  maximum number of iterations
threshold           =   '0.0mJy'         #  flux level (+units) to stop cleaning
psfmode             =    'clark'         #  method of PSF calculation to use during minor cycles
weighting           =  'natural'         #  weighting to apply to visibilities
uvtaper             =      False         #  apply additional uv tapering of  visibilities.
stokes              =        'I'         #  Stokes params to image
noise_thermal       =      False         #  add thermal noise
fidelity            =      False         #  Calculate fidelity images
display             =      False         #  Plot simulation result images,figures
verbose             =      False
async               =      False         #  If true the taskname must be started using simdata(...)

CASA <47>:
```

Parameters that control imaging – this just calls clean().
If you don't want to deconvolve at all, set psfmode="none"

simdata doesn't try very hard with clean, but it does tell you what inputs it used

IPy casa/regress

Default | Default | IPy casa/regress | rindebet@protostar:~/casa...

```
sec'}, stokes='I', phasecenter='J2000 18:00:00.03 -045.59.59.60', nchan=1, mode="FREQ-LSRK",start='668.0GHz',step={'value': 8.0, 'unit': 'GHz'}, restf
req='668.0GHz')
[setup model] swapping input axes 2 with 3
[setup model] model array minmax= 0.000000e+00 6.524700e-05
[setup model] scaling model brightness by a factor of 1.000000
[simutil] predicting from psim.diskmodel.im.coord

0%....10....20....30....40....50....60....70....80....90....100%
[simutil] generation of measurement set psim.ms complete
[noise] adding thermal noise to psim.noisy.ms
[noisetemp] subreflector diameter=0.75
[noisetemp] ruze phase efficiency for surface accuracy of 25.0um = 0.60902060889
[noisetemp] using ALMA/ACA Rx specs
[noise] interpolated receiver temp=175.236842105
[noise] antenna efficiency    = 0.500843323236
[noise] spillover efficiency  = 0.96
[noise] correlator efficiency = 0.95
SVC::Simulate :...9...8...7...6...5...4...3...2...1...0   MM::setSimulate()
AtmosCorruptor::init Tsys = -0.415 + exp(0.500843) * 438.297 => 219.103 [freq dep=0]
SVC::Simulate :...9...8...7...6...5...4...3...2...1...0[noise] done corrupting with thermal noise
[deconvolve] cleaning to psim.clean
[deconvolve] clean inputs:
[deconvolve] clean(vis='psim.noisy.ms',imagename='psim.clean',niter=1000,threshold='1e-7Jy',ftmachine='ft',imsize=[192, 192],cell='0.004arcsec',phasec
enter='J2000 18h00m00.03s -45d59m59.6s')

0%....10....20....30....40....50....60....70....80....90....100%

0%....10....20....30....40....50....60....70....80....90....100%

0%....10....20....30....40....50....60....70....80....90....100%
[analysis] removing degenerate input image axes
[analysis] removing degenerate output image axes
2009-11-18 15:15:00     WARN    ImageRegrid::regrid(...) (file /Users/remy/casa/svn/include/casacore/images/Images/ImageRegrid.tcc, line 1993)  The St
okes axis cannot be regridded - removing from list
[simutil] done inverting and cleaning
[analysis] getting beam from psim.clean.image
[analysis] Simulation rms: [  7.24876113e-07]
[analysis] Simulation max: [  8.47508396e-06]

CASA <16>:
```

Set up what you want to calculate

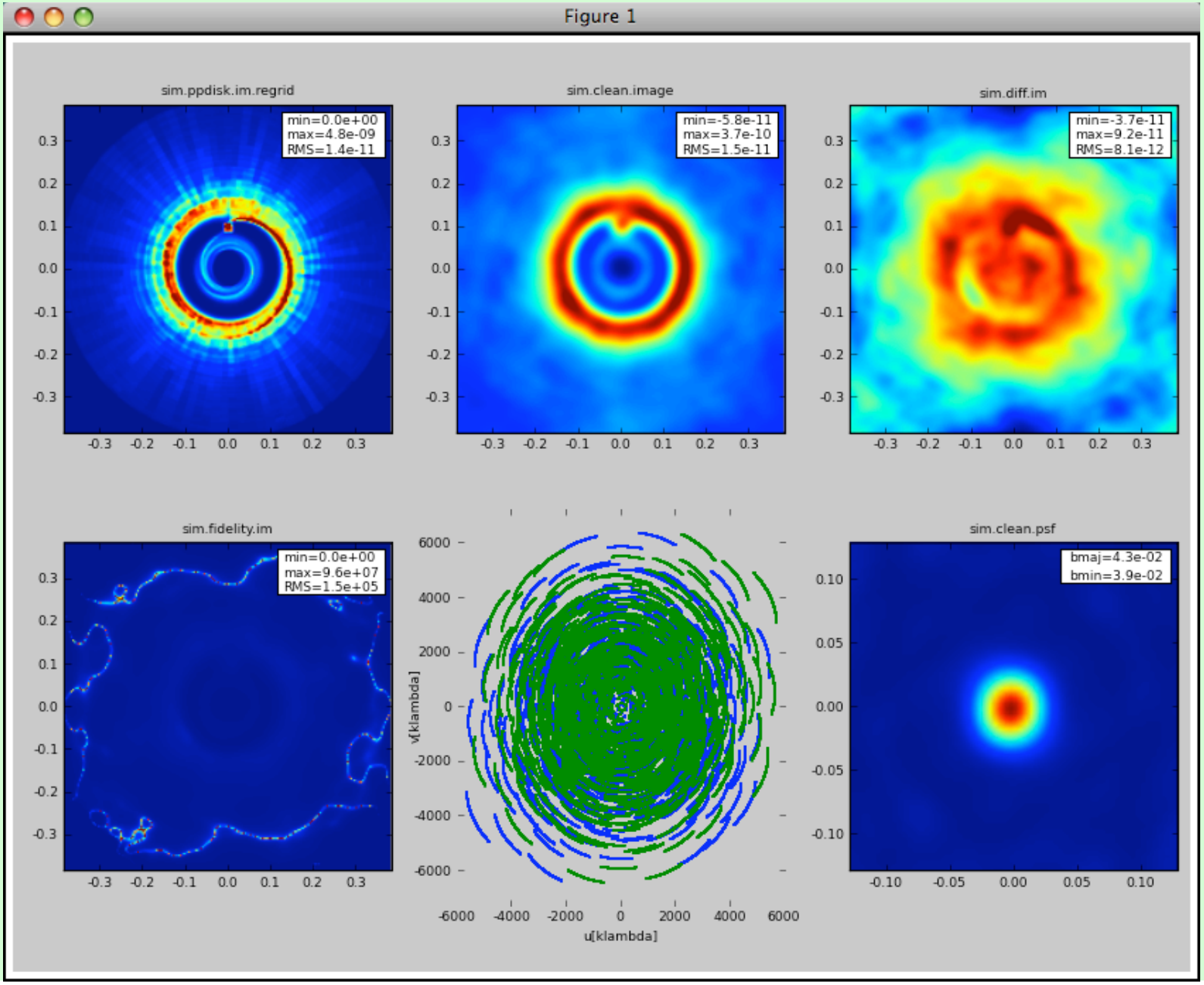
IPy casa/regress (101,36)

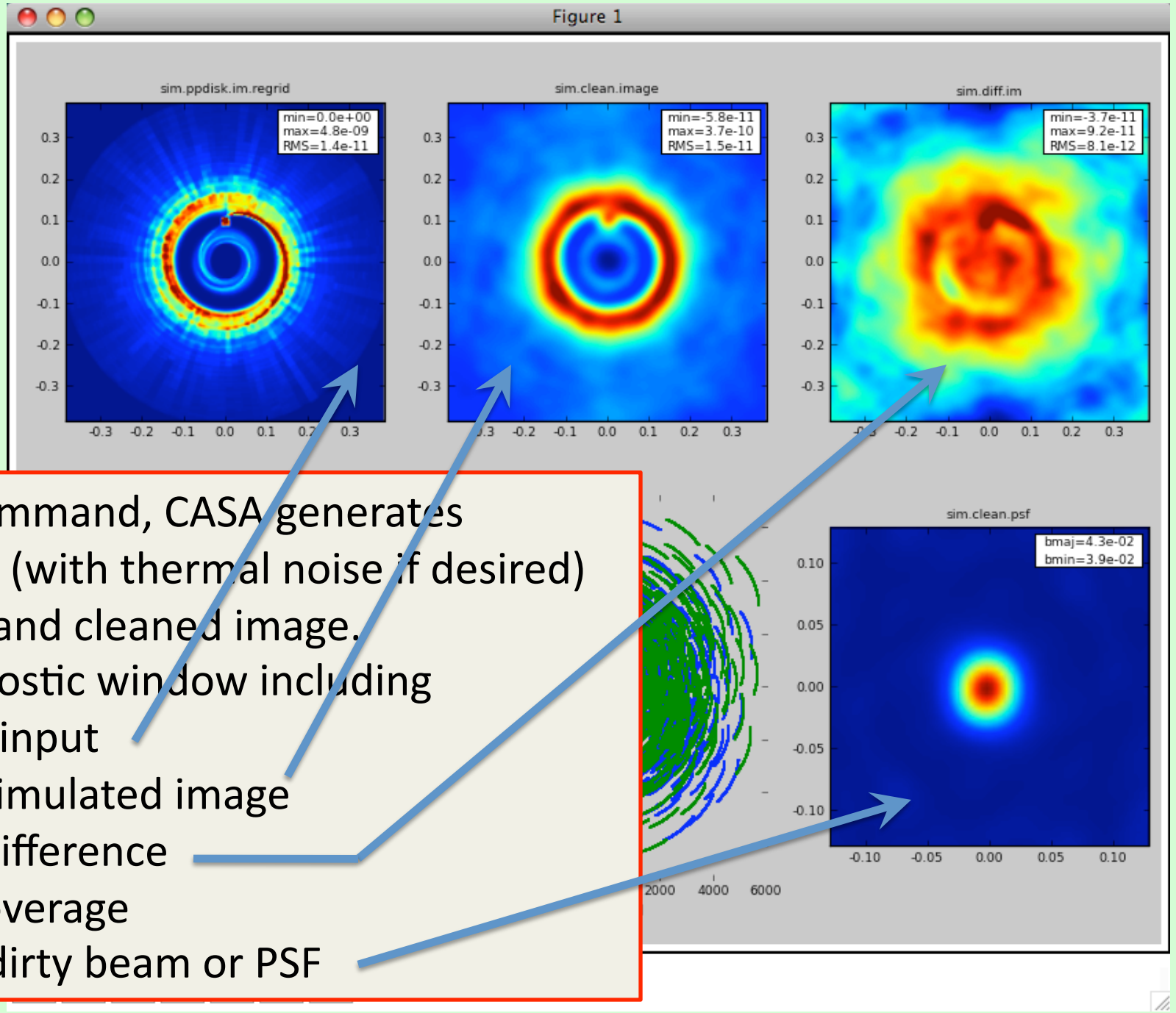
```
CASA <46>: inp
---------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'     #  input image name
ignorecoord         =      False         #  scale model coordinates to output parameters
inbright            = 'unchanged'        #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =          ''        #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                         #   position file
checkinputs         =       'no'         #  graphically verify parameters [yes|no|only]
project             =     'psim'         #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'         #  total time of observation
integration         =      '10s'         #  integration (sampling) time
startfreq           = '668.0GHz'         #  frequency of first channel
chanwidth           =   '8.0GHz'         #  channel width
nchan               =          1         #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'        #  spacing in between beams in mosaic
                                                                    tive to pointingspacing
                                                                    ing minor cycles
                                                                    ibilities.
stokes              =          'I'       #  Stokes params to image
noise_thermal       =      False         #  add thermal noise
fidelity            =      False         #  Calculate fidelity images
display             =      False         #  Plot simulation result images,figures
verbose             =      False
async               =      False         #  If true the taskname must be started using simdata(...)

CASA <47>:
```

Display=True to get the nice graphical output

Fidelity=True to calculate a fidelity image (a measure of how well your synthesis image represents the sky)

Run it already!

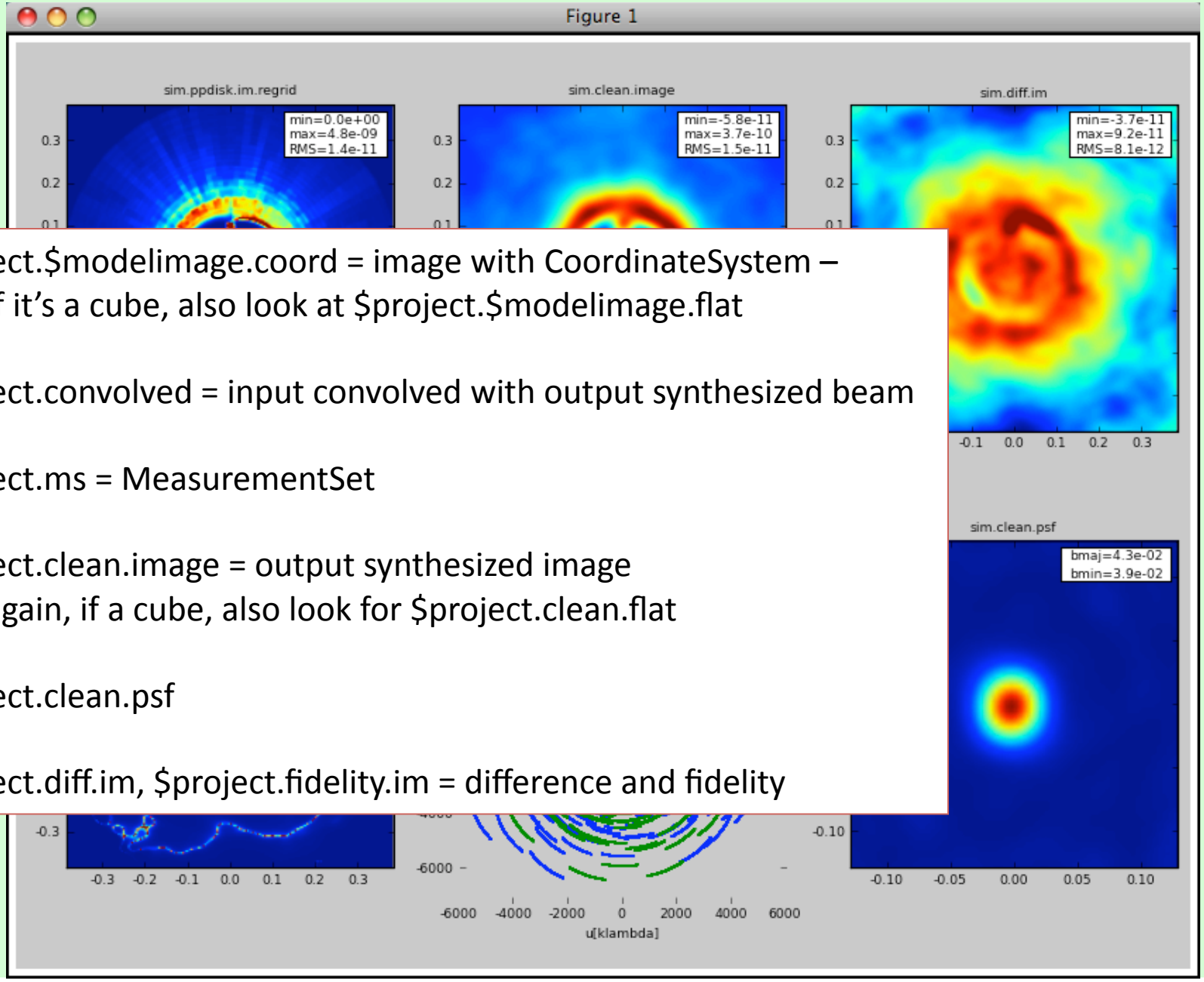




In one command, CASA generates

- uv data (with thermal noise if desired)
- a dirty and cleaned image.
- a diagnostic window including
  - your input
  - the simulated image
  - the difference
  - uv coverage
  - and dirty beam or PSF

What else do we get?



$project.$modelimage.coord = image with CoordinateSystem –
If it's a cube, also look at $project.$modelimage.flat

$project.convolved = input convolved with output synthesized beam

$project.ms = MeasurementSet

$project.clean.image = output synthesized image
again, if a cube, also look for $project.clean.flat

$project.clean.psf

$project.diff.im, $project.fidelity.im = difference and fidelity

What is this "fidelity" of which you speak?

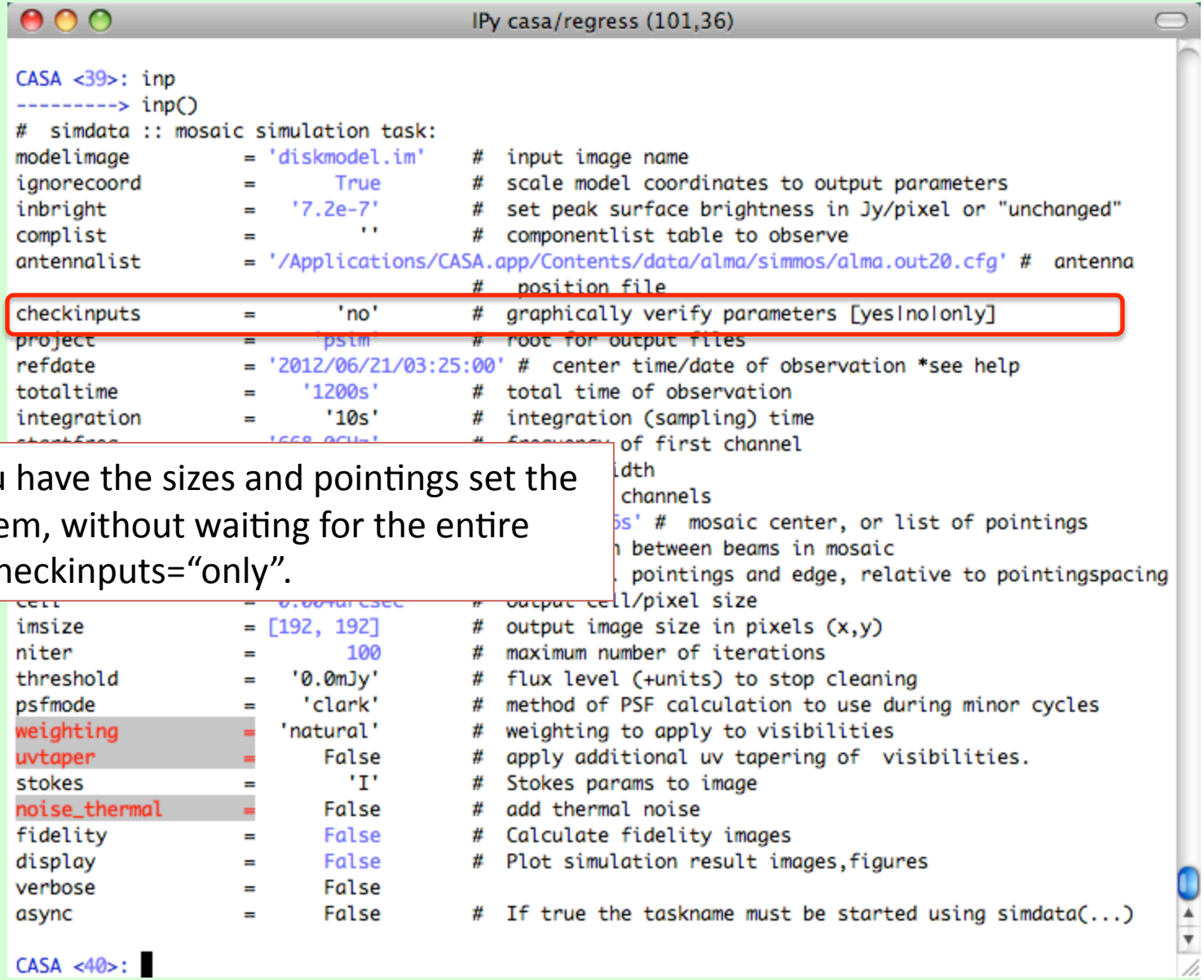
http://www.alma.nrao.edu/memos/

These deal with simulation: 488, 398, 387, **386**

$$fidelity = \frac{input}{abs(input - output)} = \frac{input}{difference}$$

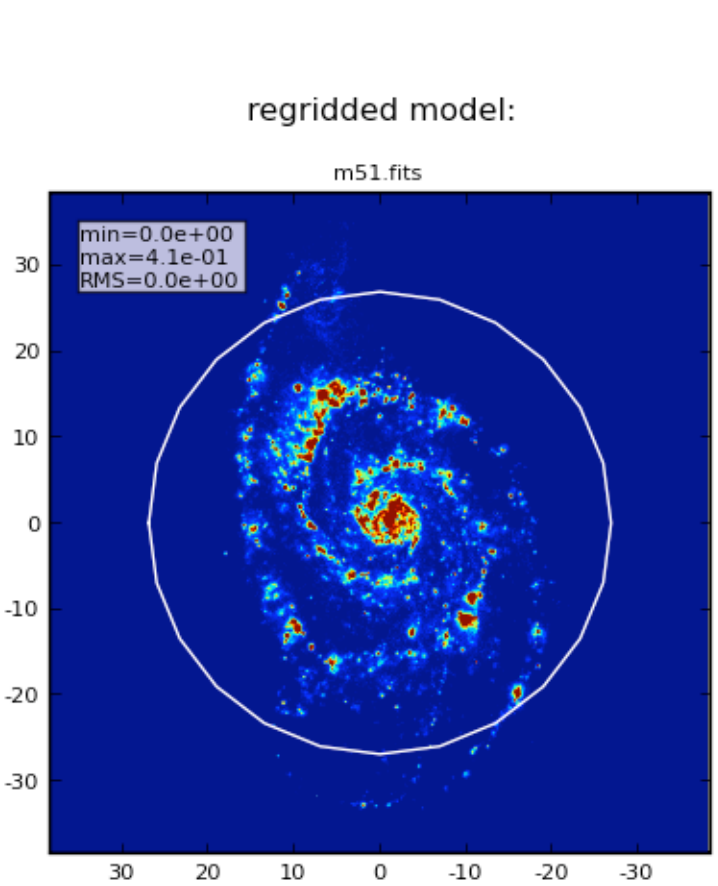
$$fidelity = \frac{input}{\max(difference, 0.7 * rms(difference))}$$

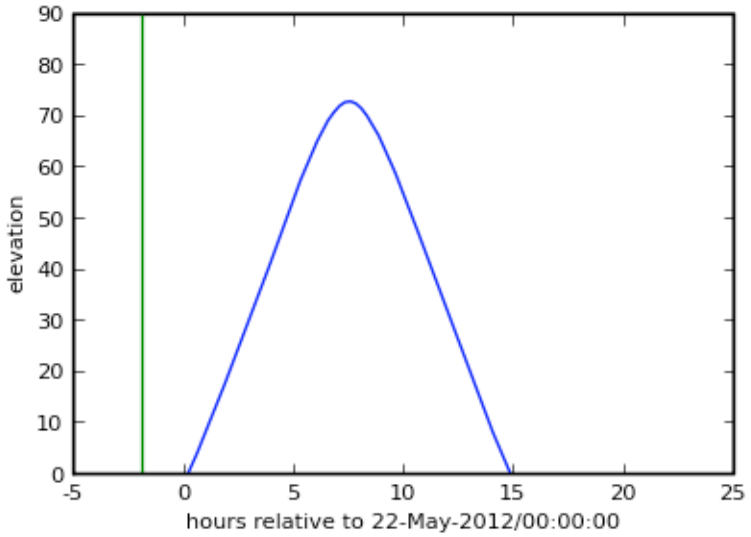
What if I don't like to wait?

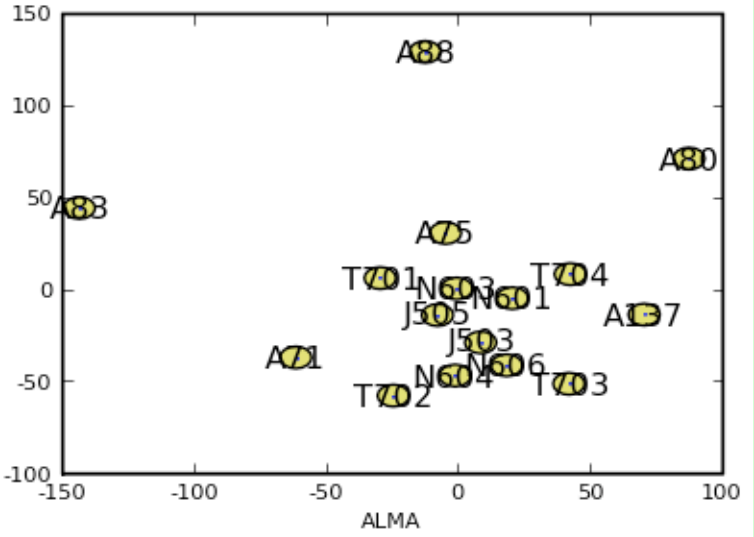


To check that you have the sizes and pointings set the way you want them, without waiting for the entire calculation, set checkinputs="only".

checkinputs







I want noise!

IPy casa/tutorial (101,38)

IPy casa/tutorial | Default

```
CASA <5>: inp
--------> inp()
#  simdata :: mosaic simulation task:
modelimage          = 'diskmodel.im'    #  input image name
ignorecoord         =       True        #  scale model coordinates to output parameters
inbright            =   '7.2e-7'        #  set peak surface brightness in Jy/pixel or "unchanged"
complist            =         ''        #  componentlist table to observe
antennalist         = '/Applications/CASA.app/Contents/data/alma/simmos/alma.out20.cfg' #  antenna
                                        #   position file
checkinputs         =     'only'        #  graphically verify parameters [yes|no|only]
project             =     'psim'        #  root for output files
refdate             = '2012/06/21/03:25:00' #  center time/date of observation *see help
totaltime           =    '1200s'        #  total time of observation
integration         =      '10s'        #  integration (sampling) time
startfreq           = '668.0GHz'        #  frequency of first channel
chanwidth           =   '8.0GHz'        #  channel width
nchan               =          1        #  number of channels
direction           = 'J2000 18h00m00.03s -45d59m59.6s' #  mosaic center, or list of pointings
pointingspacing     = '0.5arcsec'       #  spacing in between beams in mosaic
relmargin           =        1.0        #  space btw. pointings and edge, relative to pointingspacing
cell                = '0.004arcsec'     #  output cell/pixel size
imsize              = [192, 192]        #  output image size in pixels (x,y)
                                        ions
                                        top cleaning
                                        n to use during minor cycles
                                        sibilities
uvtaper             =       False       #  apply additional uv tapering of  visibilities.
stokes              =         'I'       #  Stokes params to image
noise_thermal       =        True       #  add thermal noise
     t_amb          =       265.0       #  ambient temperature
     tau0           =         0.1       #  zenith opacity

fidelity            =       False       #  Calculate fidelity images
display             =       False       #  Plot simulation result images,figures
verbose             =       False
async               =       False       #  If true the taskname must be started using simdata(...)

CASA <6>:
```

In simdata, we have thermal noise (rest in sm tool)

* Interface likely to change slightly this week

Thermal noise

System temperature referenced above the atmosphere – usually the source signal T_A^* is neglected

$$T_{sys} = \eta_{spill} T_A^* + T_{CMB} + \eta_{spill} T_{atm} (e^{\tau} - 1) + (1 - \eta_{spill}) T_{amb} e^{\tau} + T_{RX} e^{\tau}$$

$$\tau = \tau(\nu, airmass)$$

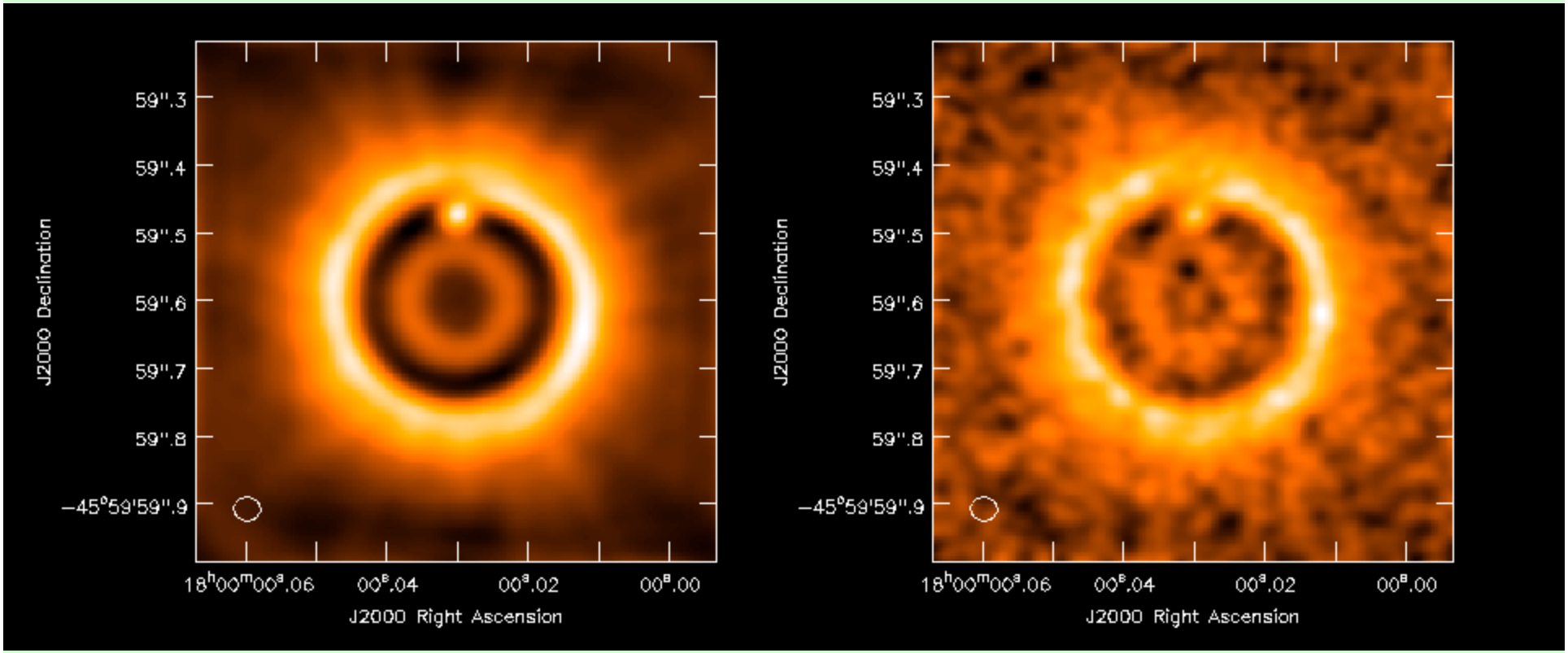
Random noise is added to the real and imaginary parts of the visibilities according to the average of the airmasses seen by the two antennas making up each given baseline.

The noise is actually applied in Janskies with the following conversion:

$$F_{\nu} = \frac{4\sqrt{2} k 10^{-23}}{\eta_{ant} \eta_{correl} \pi d_1 d_2 \sqrt{2 \Delta\nu \Delta t}} T_{sys}$$

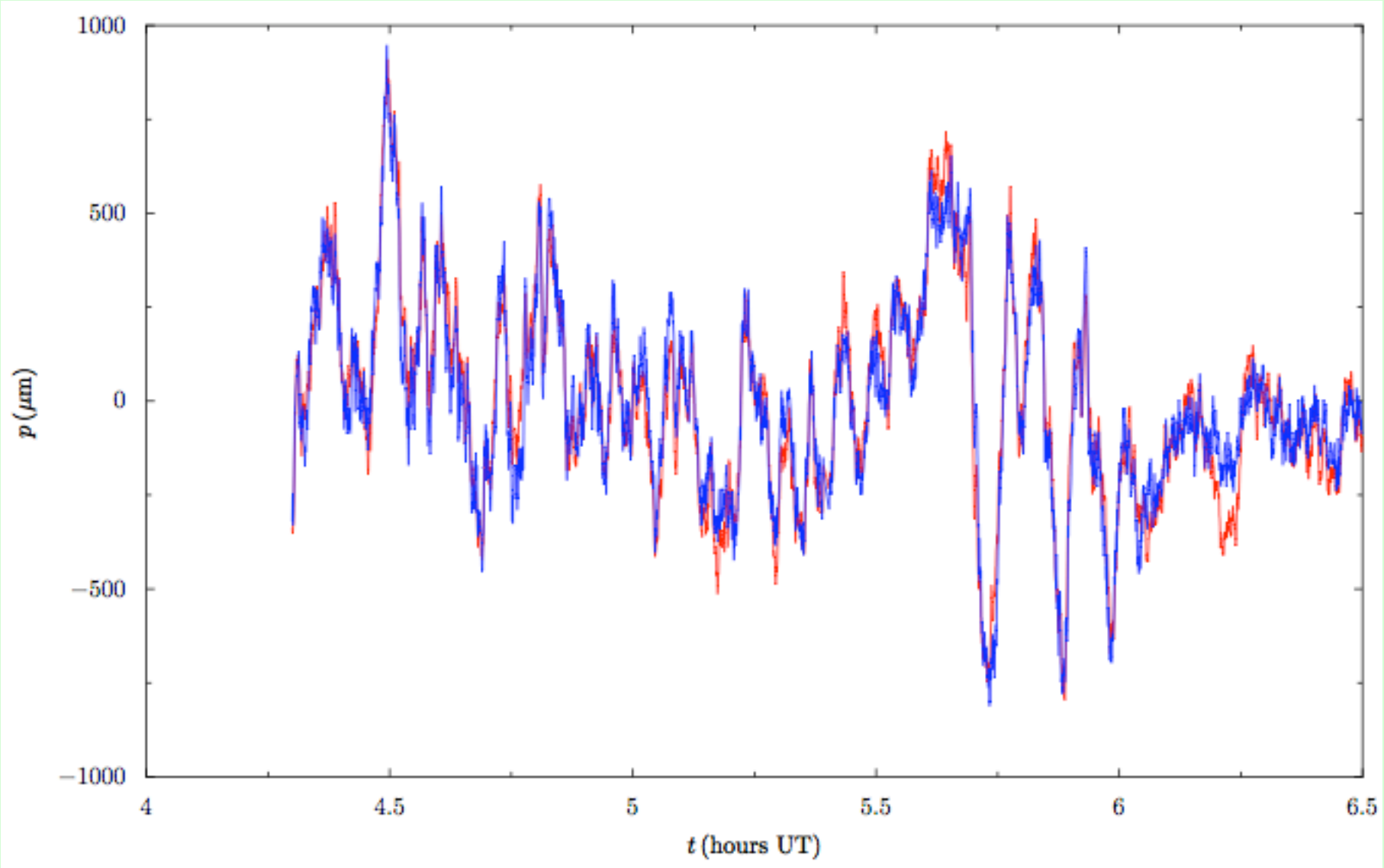
A new noisy MS is created, and optionally a cal table containing the added noise

Thermal noise



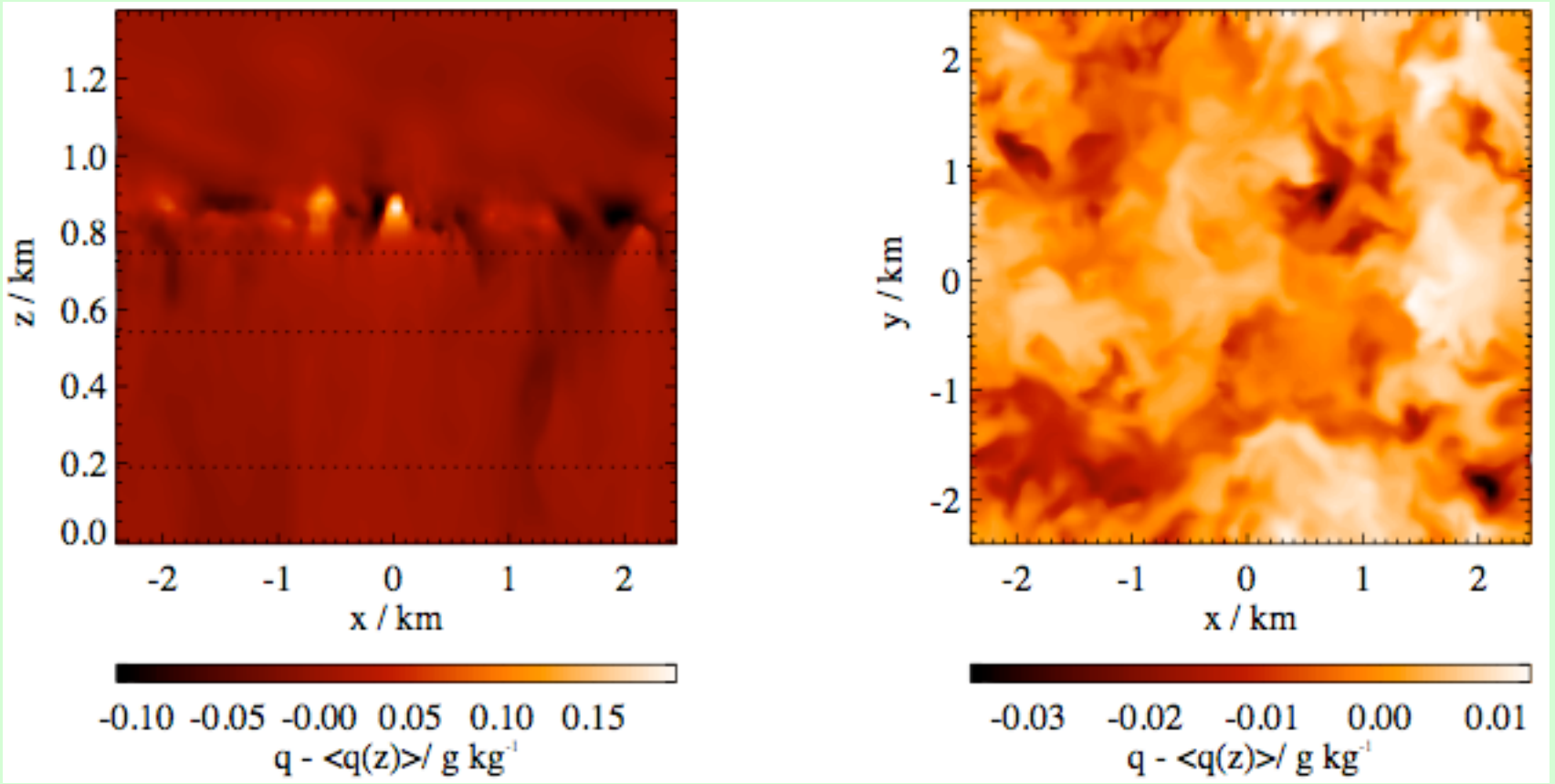
Atmospheric phase delay

From Bojan Nikolic's presentation of real path fluctuation measured at the SMA, and the calculated path (blue) from WVR data



Atmospheric phase delay

Stirling memo 517: convective simulations reveal PWV fluctuations concentrated in a layer, with approximately fractal structure in the horizontal plane:



Atmospheric phase delay

Bojan Nikolic memo 588

approximate real hydro simulation with a static phase screen, blown over the array.

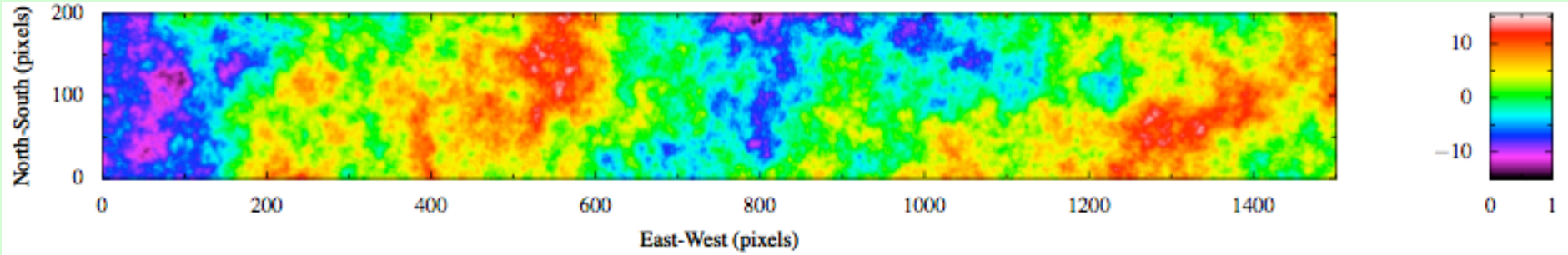


Figure 2. A one-eight subsection of the turbulent phase screen used in the simulation of empirically determining the correlation between temperature brightness and phase fluctuations. The scale is in arbitrary units, as the screen is later re-scaled so that the fluctuation on a 300 m baseline is 200 μm.

my implementation in CASA: fractional Brownian motion or generalized 1/f noise

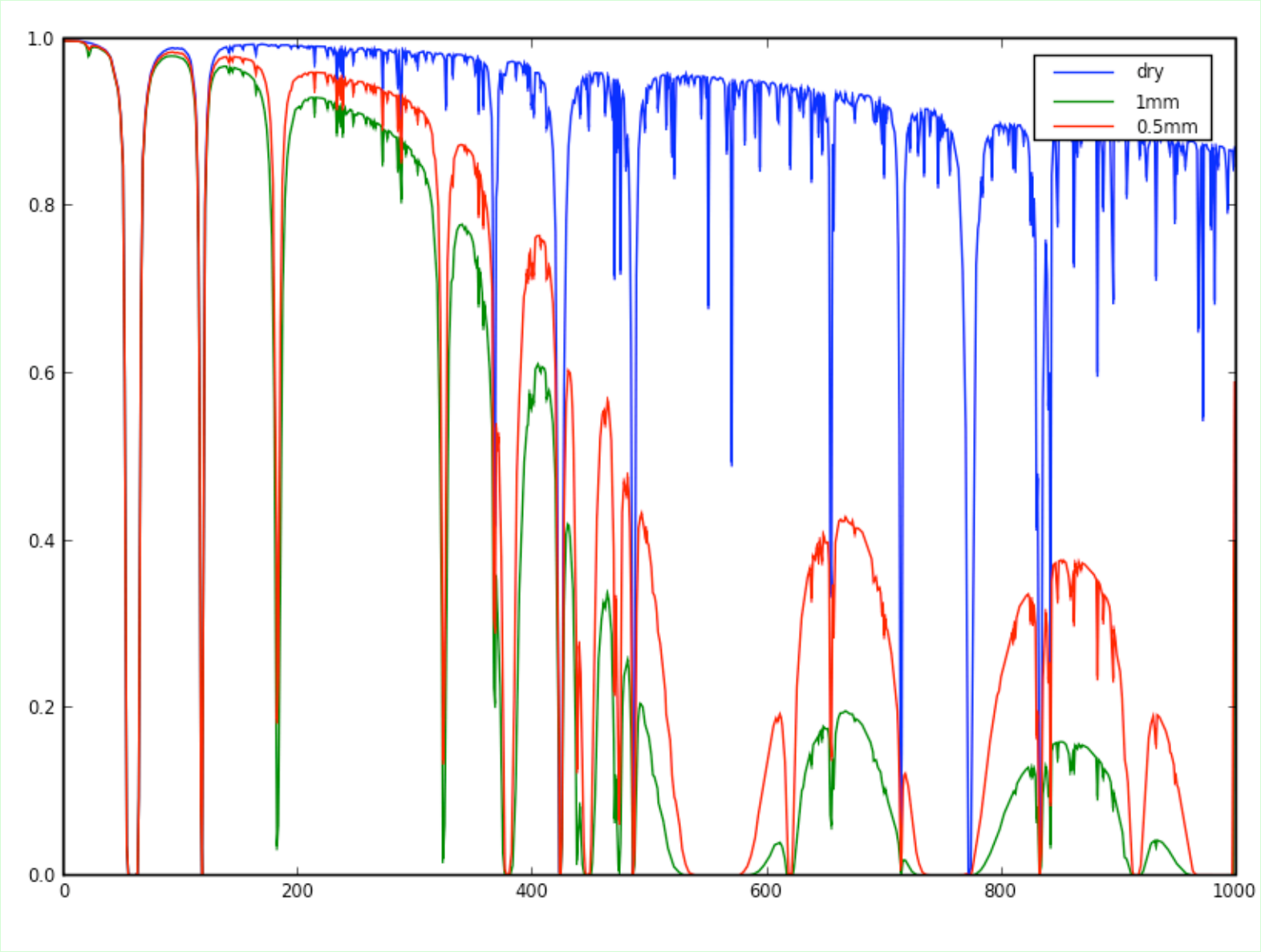
- a statistically good representation of many stochastic natural processes
- this phase screen approximation has been verified with on-site measurements e.g. Ishizaki memo 529.
- Bojan's work has quantified the errors from using a 2-d screen v. a 3d screen (which can take into account the different WVR and science receiver beams)
- 2d should be sufficient for current implementation in CASA, enhancement possible

Generated in Fourier space, with X a random variable in [0,1],

i^{th} phase = $2\pi X$, i^{th} amplitude = $X\, i^{-\beta/2}$

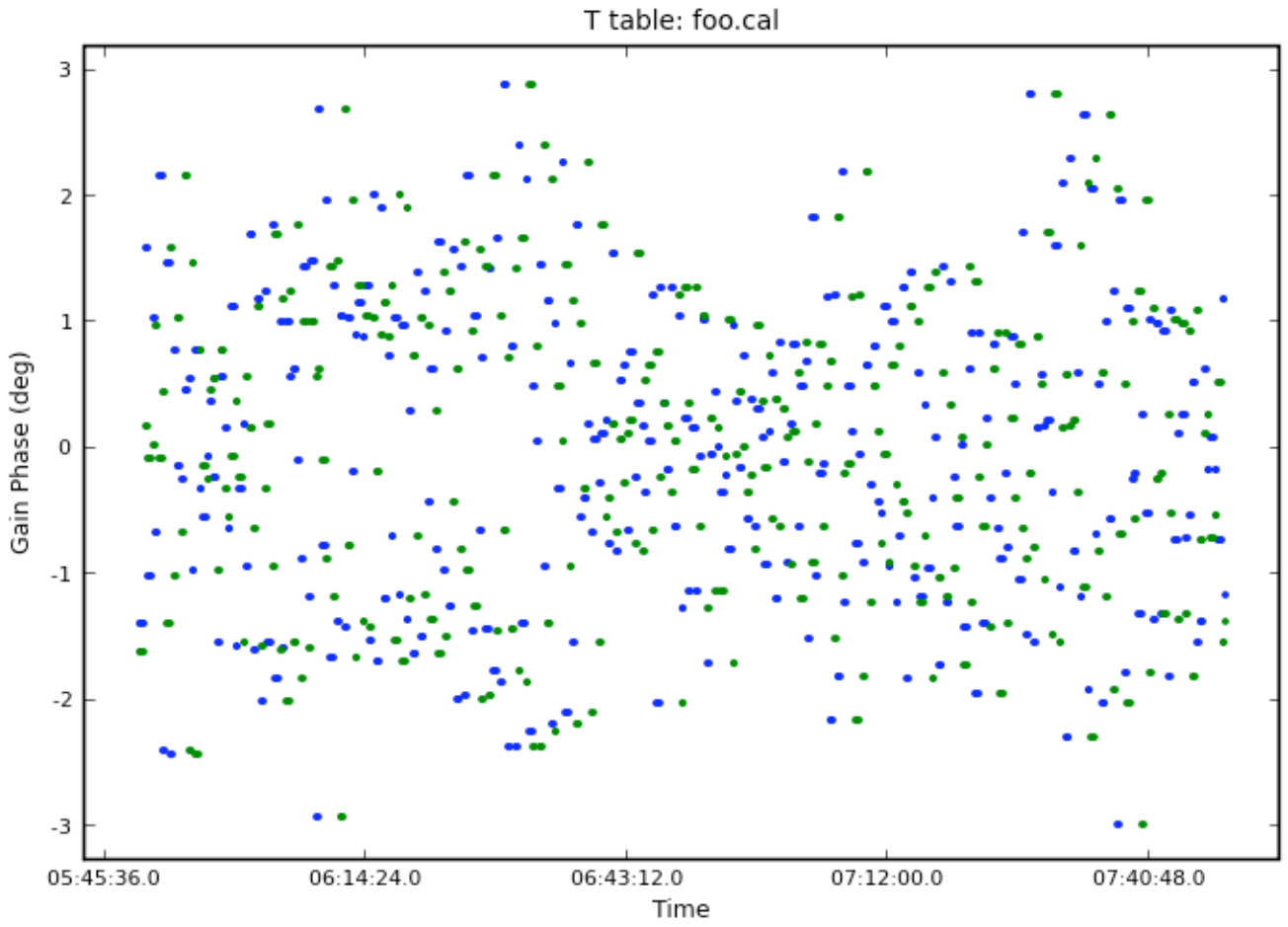
Atmospheric phase delay

What is generated is a PWV screen, then Pardo's ATM library is used to calculate phase



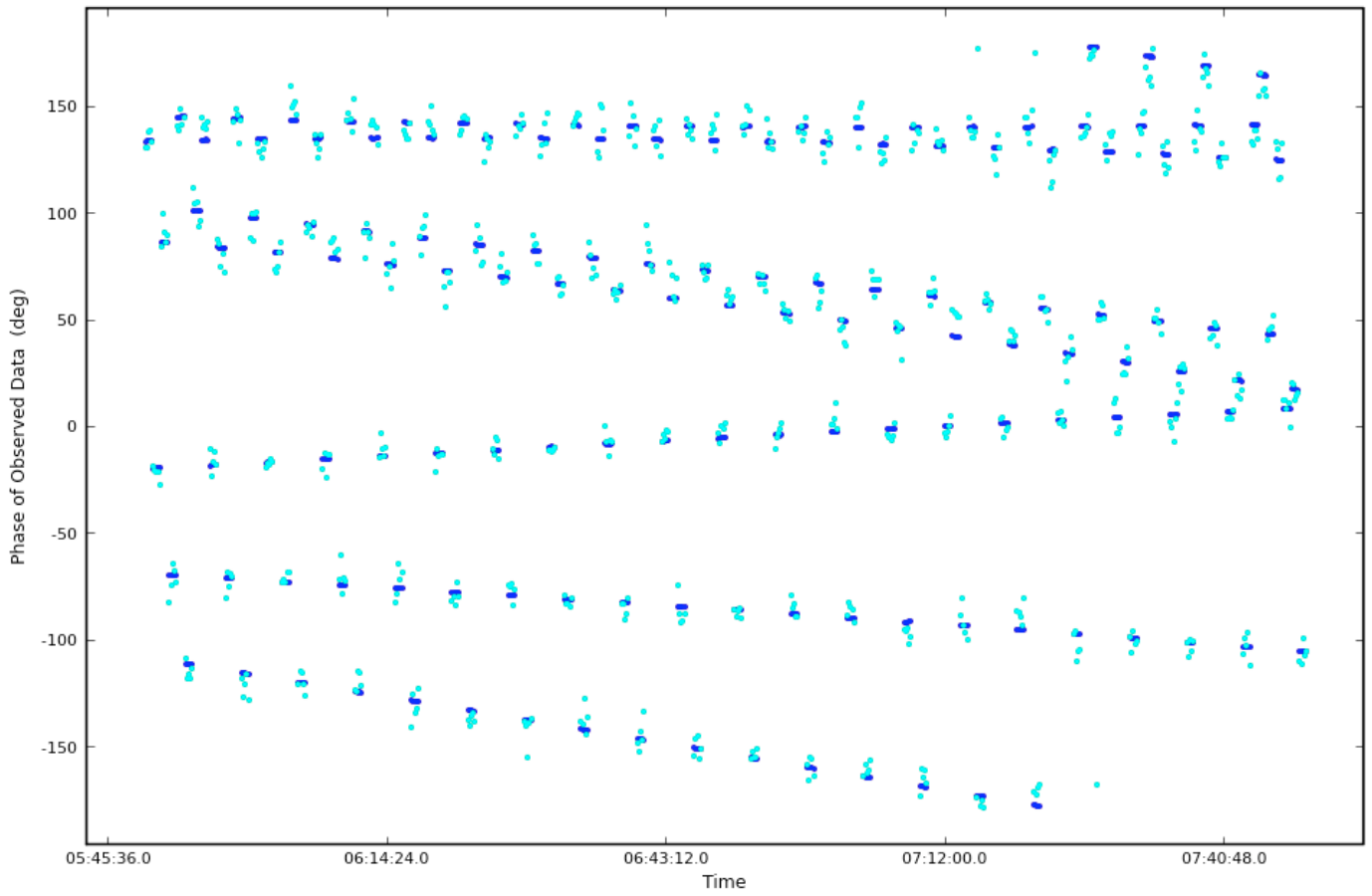
Atmospheric phase delay

Actual cal table generated, for 2 antennas that lie close to the wind direction:

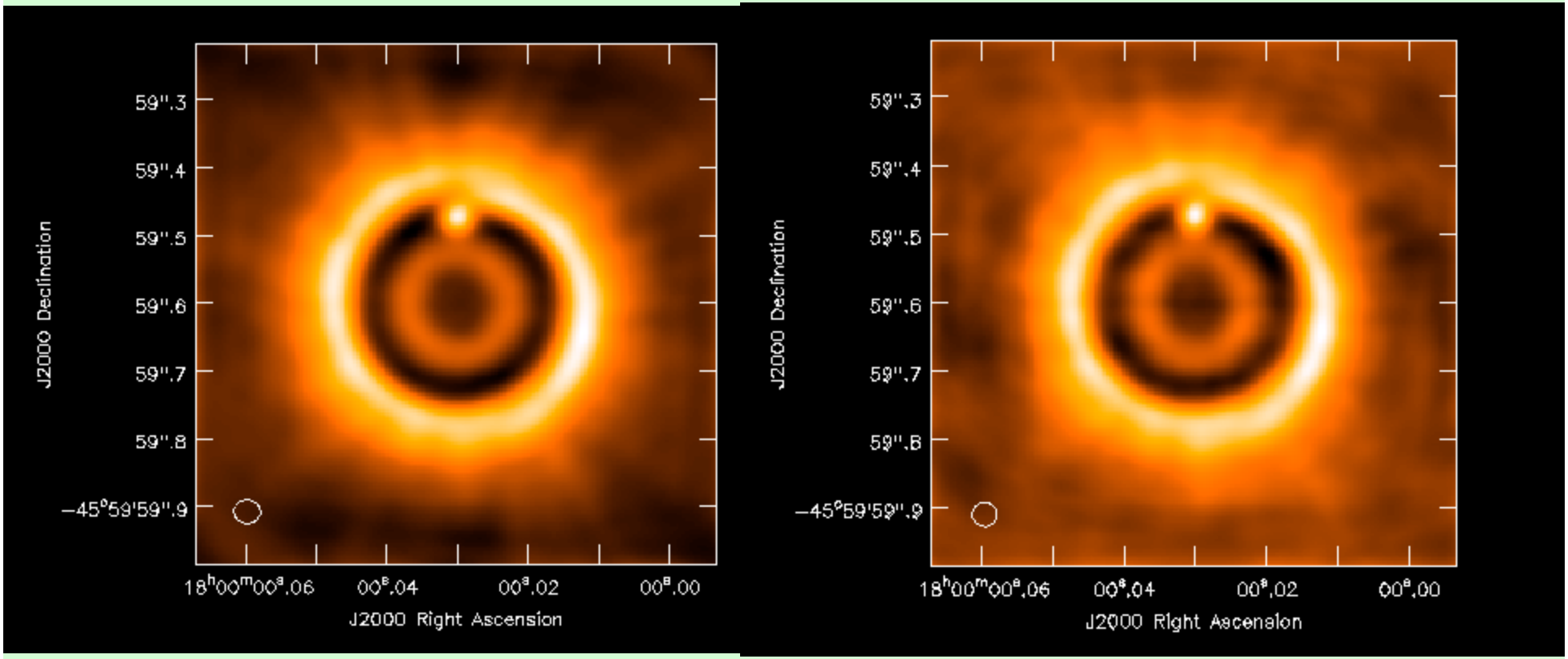


Atmospheric phase delay

Applied to a MS:



Atmospheric phase delay



Atmospheric phase delay

Today, use tool. Hope to interface to simdata task for 3.0 release

CASA> sm.openfromms("my.ms")

CASA> sm.settrop(pwv=1.)

CASA> sm.corrupt()

CASA> sm.done()

```
CASA <4>: help sm.settrop
--------> help(sm.settrop)
Help on built-in function settrop:

settrop(...)
    Set tropospheric gain corruptions
    --- --- --- --- --- --- Parameters  --- --- --- --- --- ---
      mode:  Mode of operation - screen or individual antennas screen 'screen'
      table:  Name of cal table ''
      pwv:  total precipitable water vapour in mm 3.0
      deltapwv:  RMS PWV fluctuations in mm (typically about 15% of total PWV) 0.5
      beta:  exponent of fractional brownian motion 1.1
      windspeed:  wind speed for screen type corruption (m/s) 7.
    --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
          mode      = screen
          table
          pwv       = 3.0
          deltapwv  = 0.5
          beta      = 1.1
          windspeed = 7.
         ----------------------------------------
(END)
```

Corruption summary

Thermal noise : sm.setnoise2

- user sets T_{RX}, T_{CMB}, T_{atmos}, and efficiencies
- tau(v) calculated using ATM or user specifies a value
- tau and atmospheric T_b could be calculated more self-consistently, from user-specified altitude, PWV, etc

Thermal noise: simdata

- user sets T_{atmos} and tau, simdata looks up sensible atmosphere for the site, and looks up receiver specs for the given observatory (could do with a bit more flexibility and more complete site info)

Atmospheric phase screen: sm.settrop

- individual (uncorrelated) fluctuations for each antenna
- 2d screen blows across the array

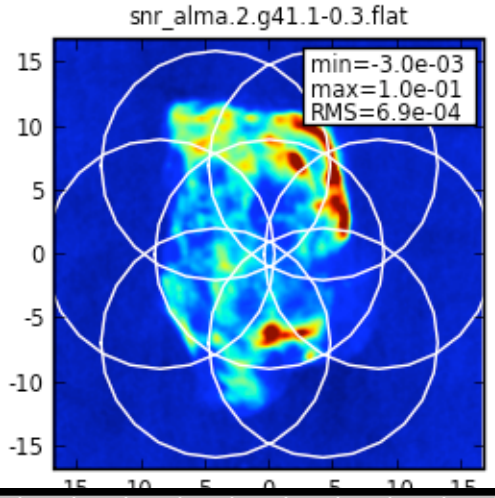
Cross-polarization: sm.setleakage – only constant at the moment

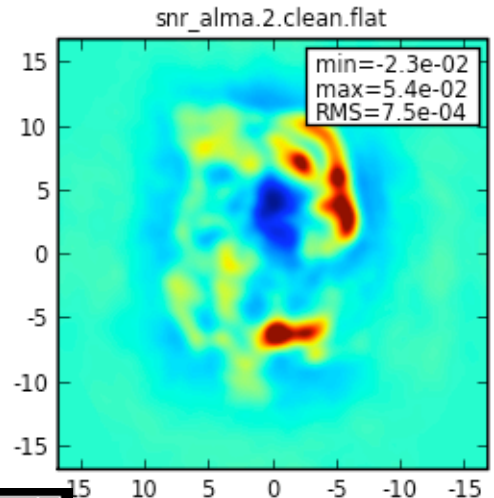
Gain fluctuation: sm.setgain – constant or fractional brownian motion fluctuations

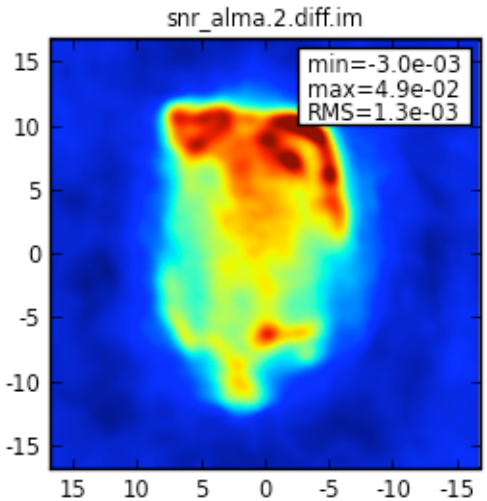
Bandpass and pointing errors – BP soon, pointing = Sanjay.

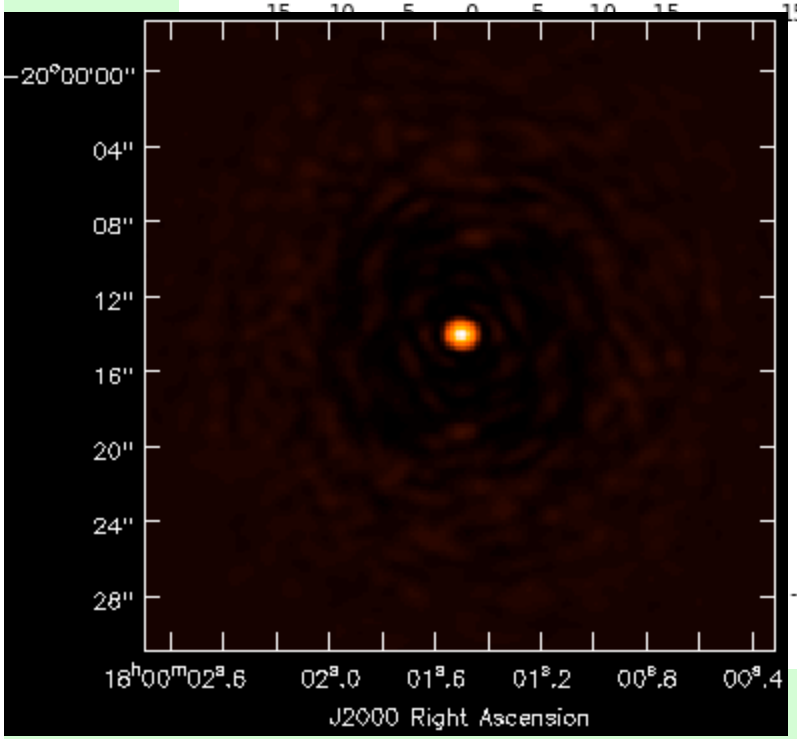
Variable feed angles on different antennas – Spring 2010

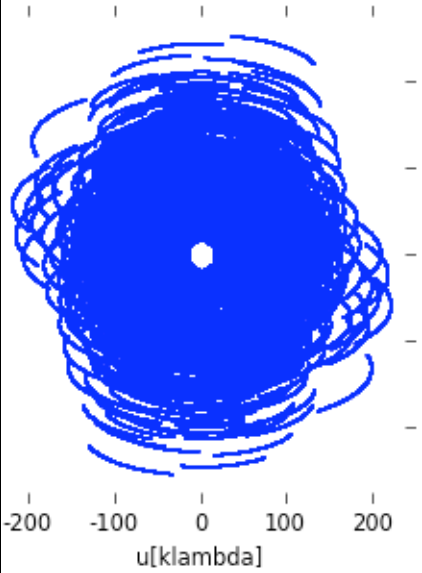
ALMA

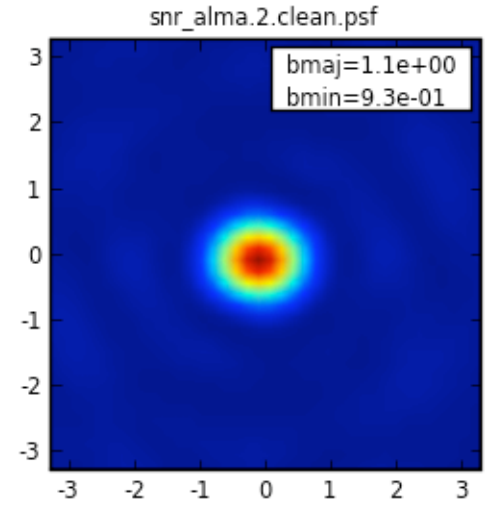

snr_alma.2.g41.1-0.3.flat
min=-3.0e-03
max=1.0e-01
RMS=6.9e-04


snr_alma.2.clean.flat
min=-2.3e-02
max=5.4e-02
RMS=7.5e-04

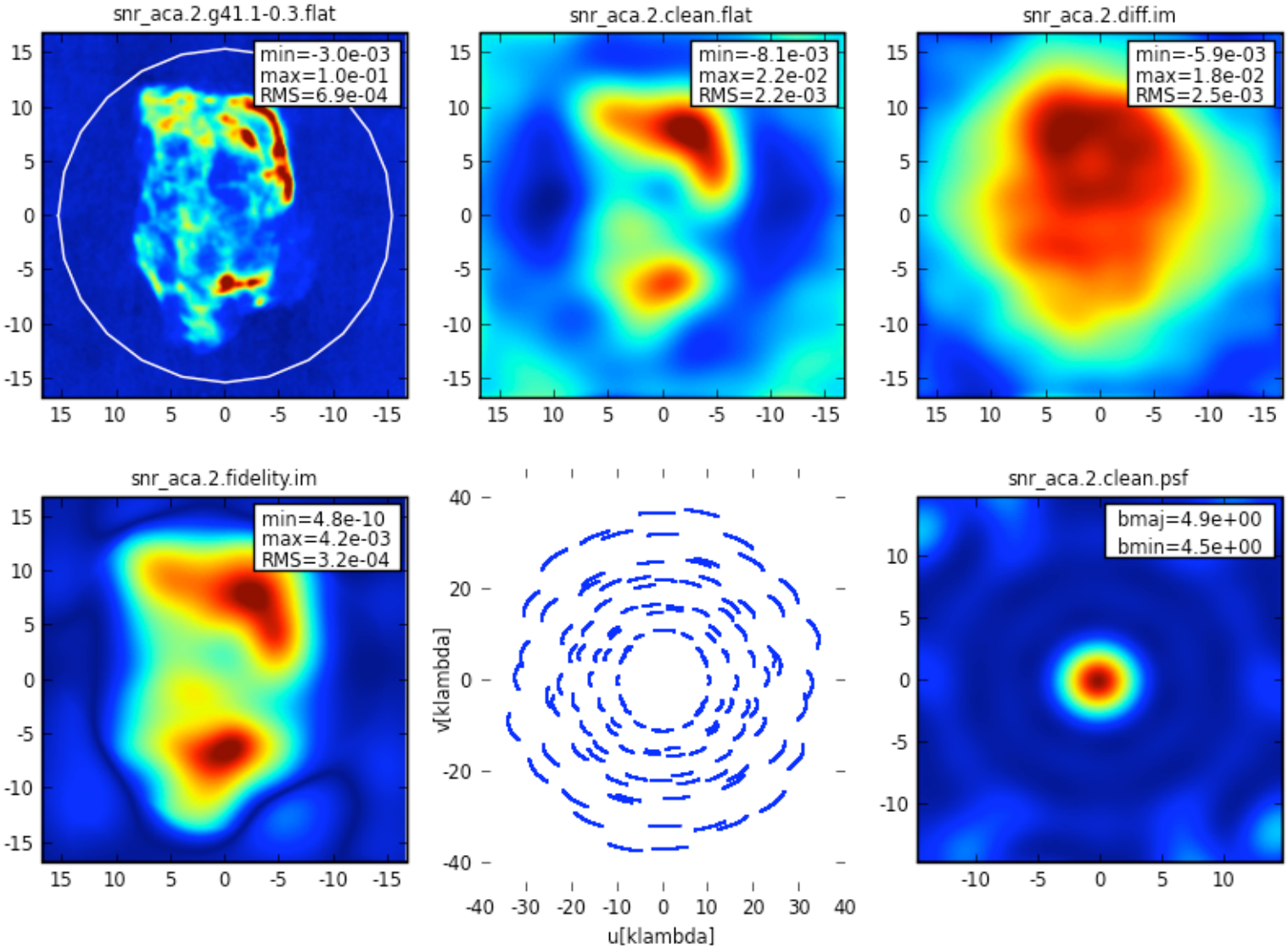

snr_alma.2.diff.im
min=-3.0e-03
max=4.9e-02
RMS=1.3e-03


-20°00'00"
04"
08"
12"
16"
20"
24"
28"
18h00m02s.6
02s.0
01s.6
01s.2
00s.8
00s.4
J2000 Right Ascension

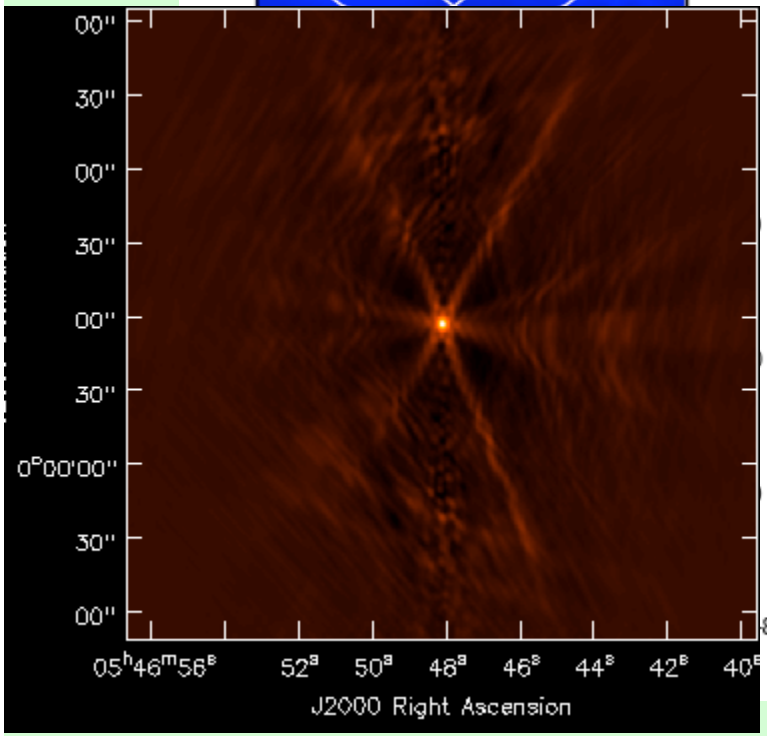

-200
-100
0
100
200
u[klambda]

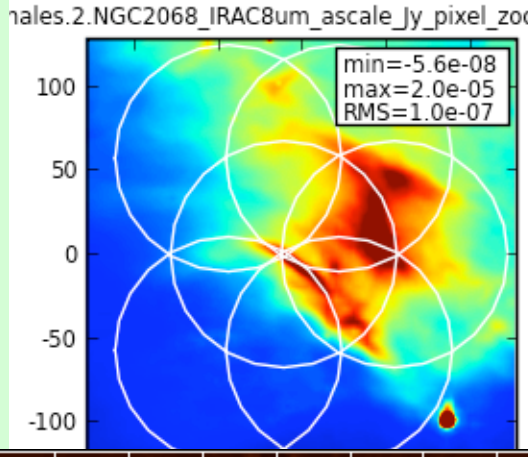

snr_alma.2.clean.psf
bmaj=1.1e+00
bmin=9.3e-01

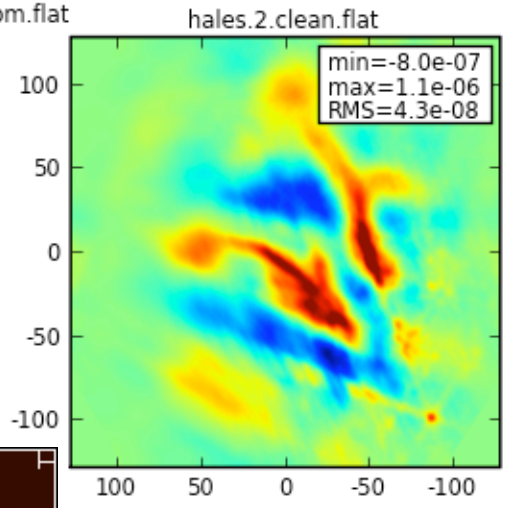
ACA

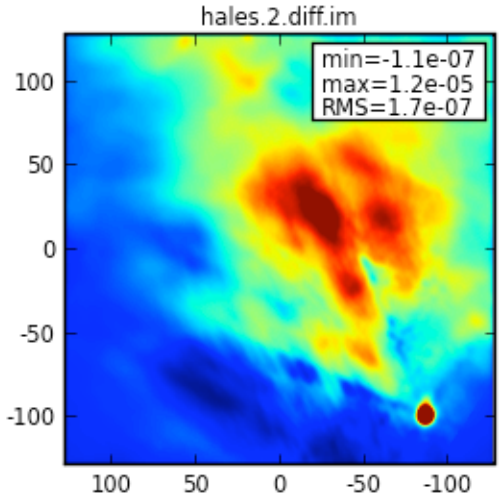

snr_aca.2.g41.1-0.3.flat
min=-3.0e-03
max=1.0e-01
RMS=6.9e-04
snr_aca.2.clean.flat
min=-8.1e-03
max=2.2e-02
RMS=2.2e-03
snr_aca.2.diff.im
min=-5.9e-03
max=1.8e-02
RMS=2.5e-03
snr_aca.2.fidelity.im
min=4.8e-10
max=4.2e-03
RMS=3.2e-04
v[klambda]
u[klambda]
snr_aca.2.clean.psf
bmaj=4.9e+00
bmin=4.5e+00

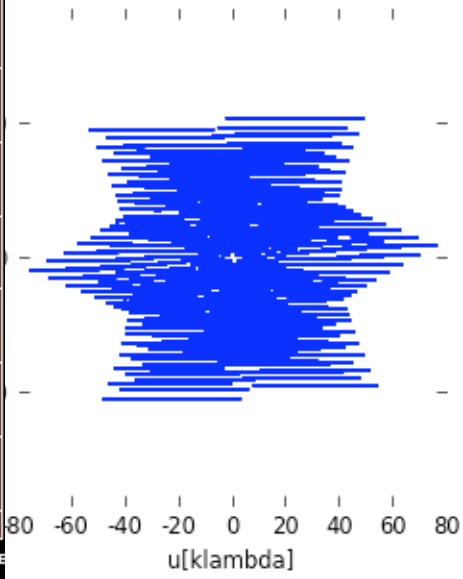
VLA

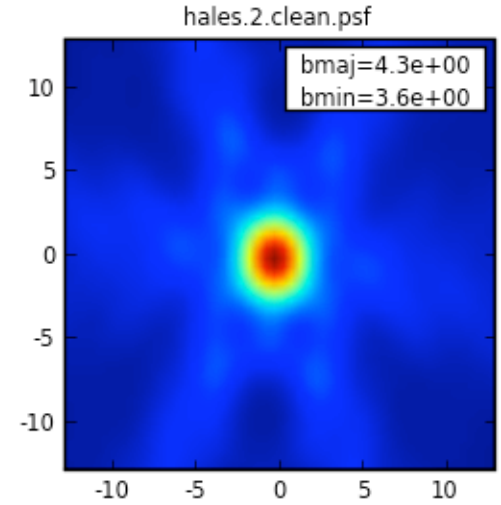

00''
30''
00''
30''
00''
30''
0°00'00''
30''
00''
05h46m56s
52s
50s
48s
46s
44s
42s
40s
J2000 Right Ascension


hales.2.NGC2068_IRAC8um_ascale_Jy_pixel_zoom.flat
min=-5.6e-08
max=2.0e-05
RMS=1.0e-07
100
50
0
-50
-100

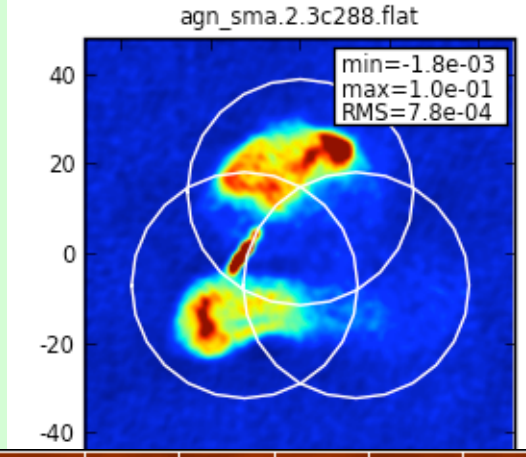

hales.2.clean.flat
min=-8.0e-07
max=1.1e-06
RMS=4.3e-08
100
50
0
-50
-100

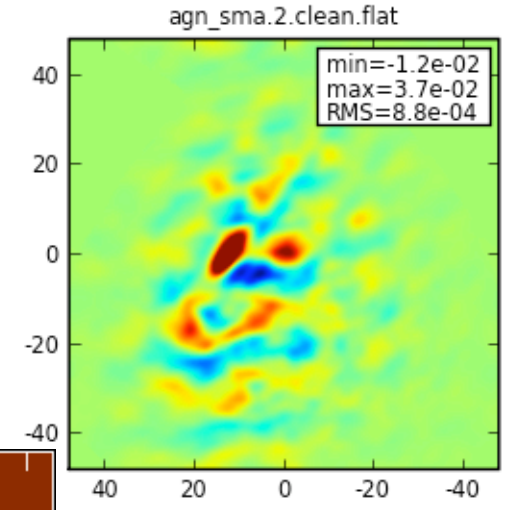

hales.2.diff.im
min=-1.1e-07
max=1.2e-05
RMS=1.7e-07
100
50
0
-50
-100

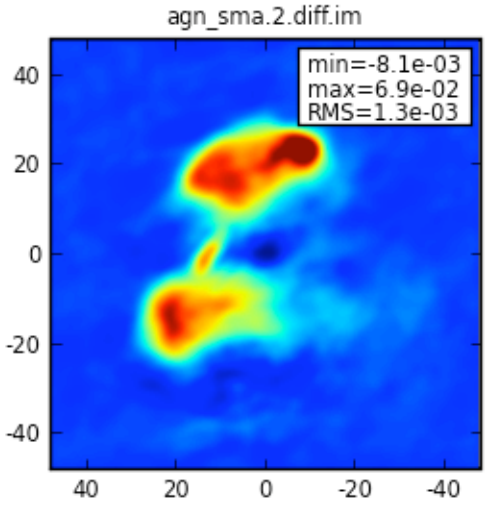

-80
-60
-40
-20
0
20
40
60
80
u[klambda]

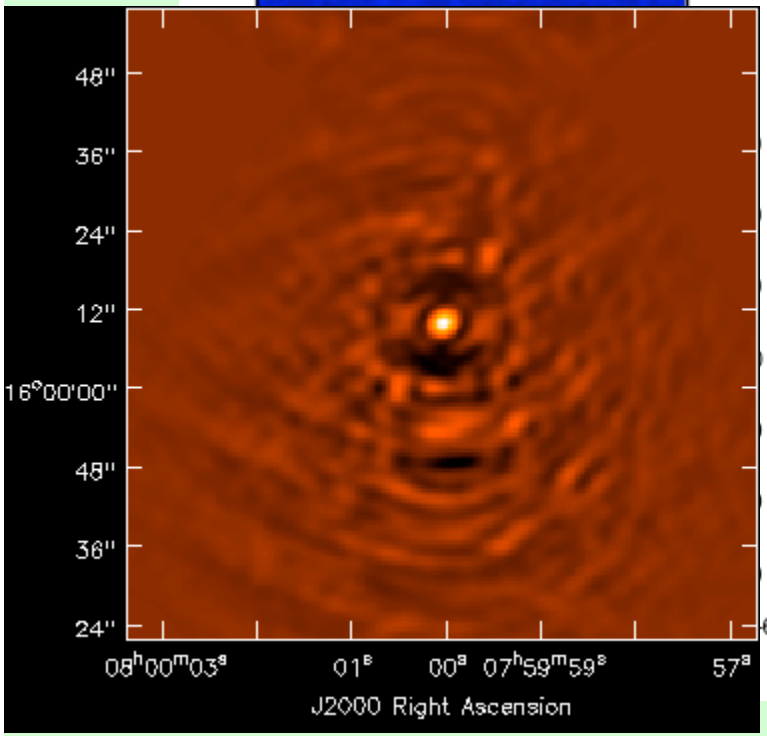

hales.2.clean.psf
bmaj=4.3e+00
bmin=3.6e+00
10
5
0
-5
-10

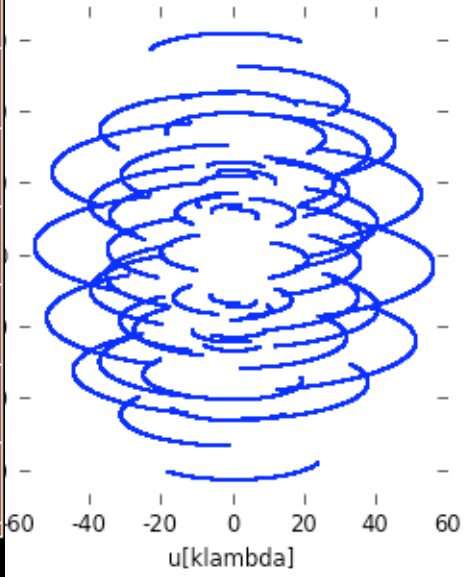
SMA

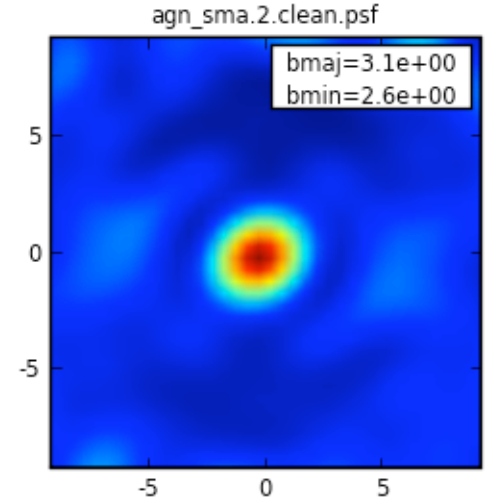

agn_sma.2.3c288.flat
min=-1.8e-03
max=1.0e-01
RMS=7.8e-04


agn_sma.2.clean.flat
min=-1.2e-02
max=3.7e-02
RMS=8.8e-04

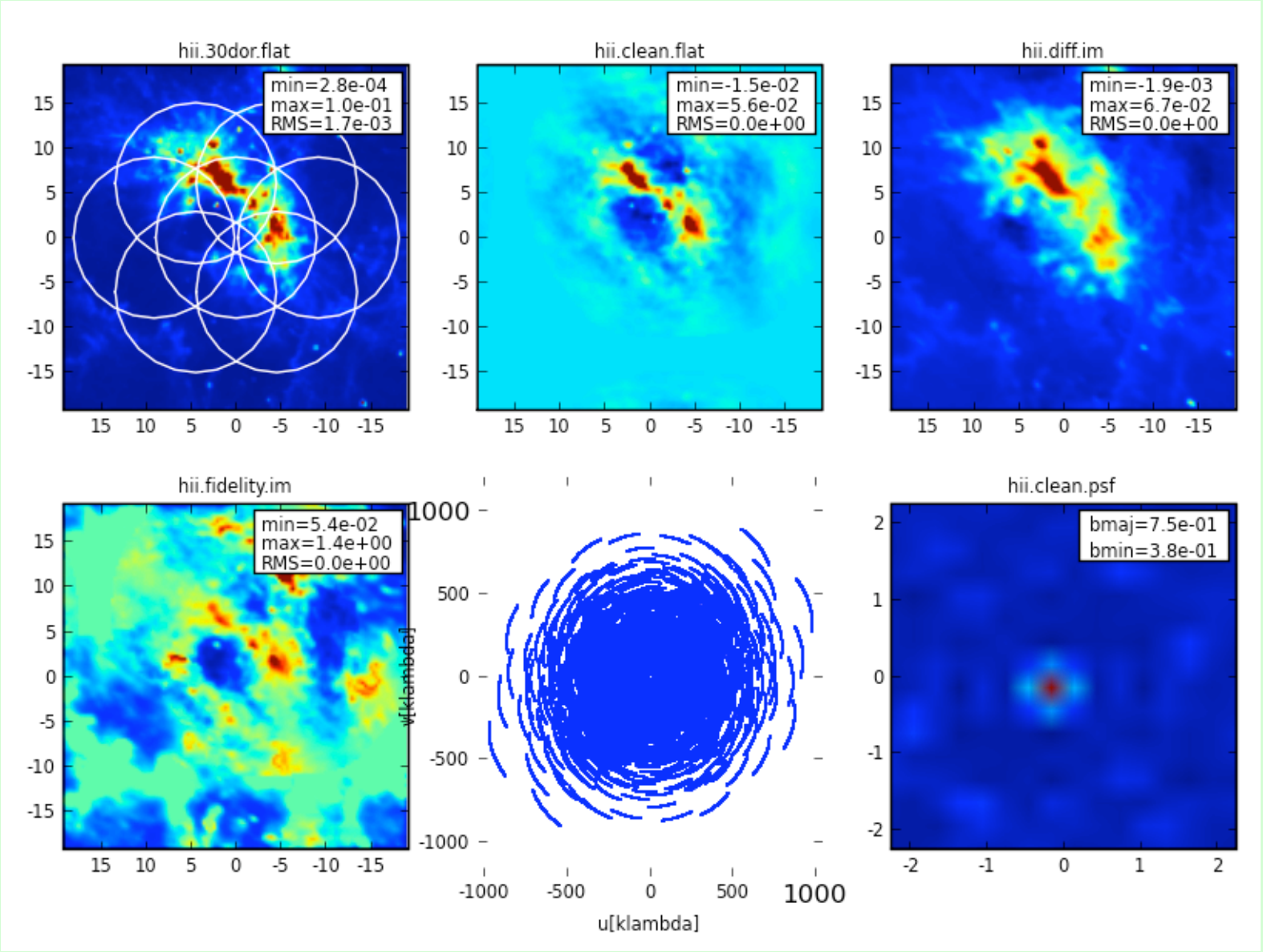

agn_sma.2.diff.im
min=-8.1e-03
max=6.9e-02
RMS=1.3e-03

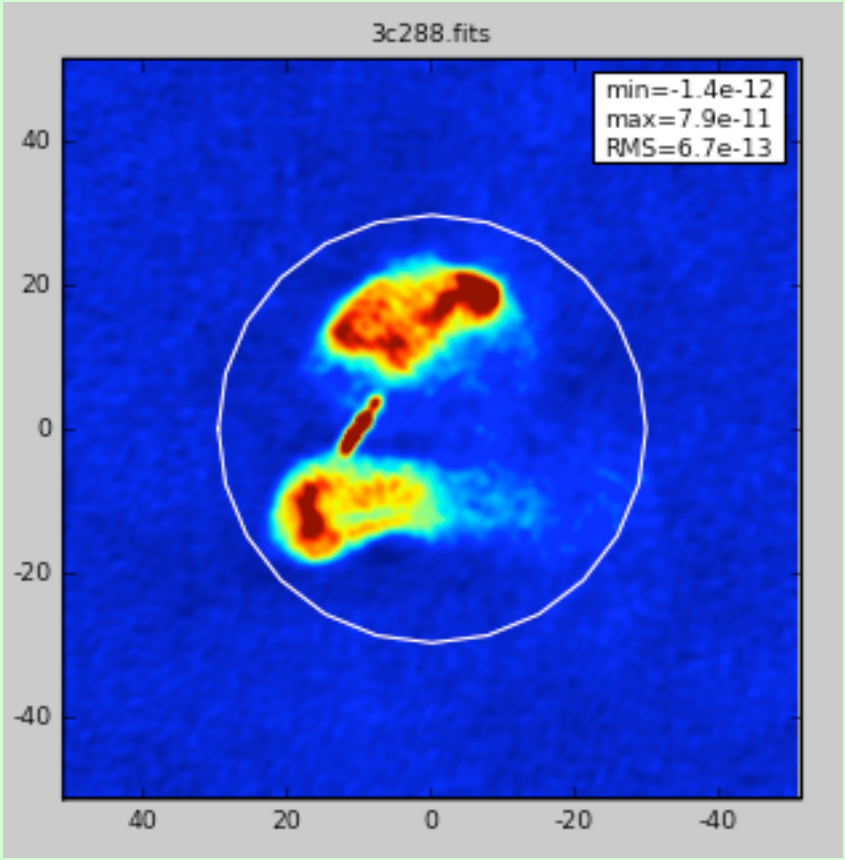

48"
36"
24"
12"
16°00'00"
48"
36"
24"
08h00m03s
01s
00s
07h59m59s
57s
J2000 Right Ascension

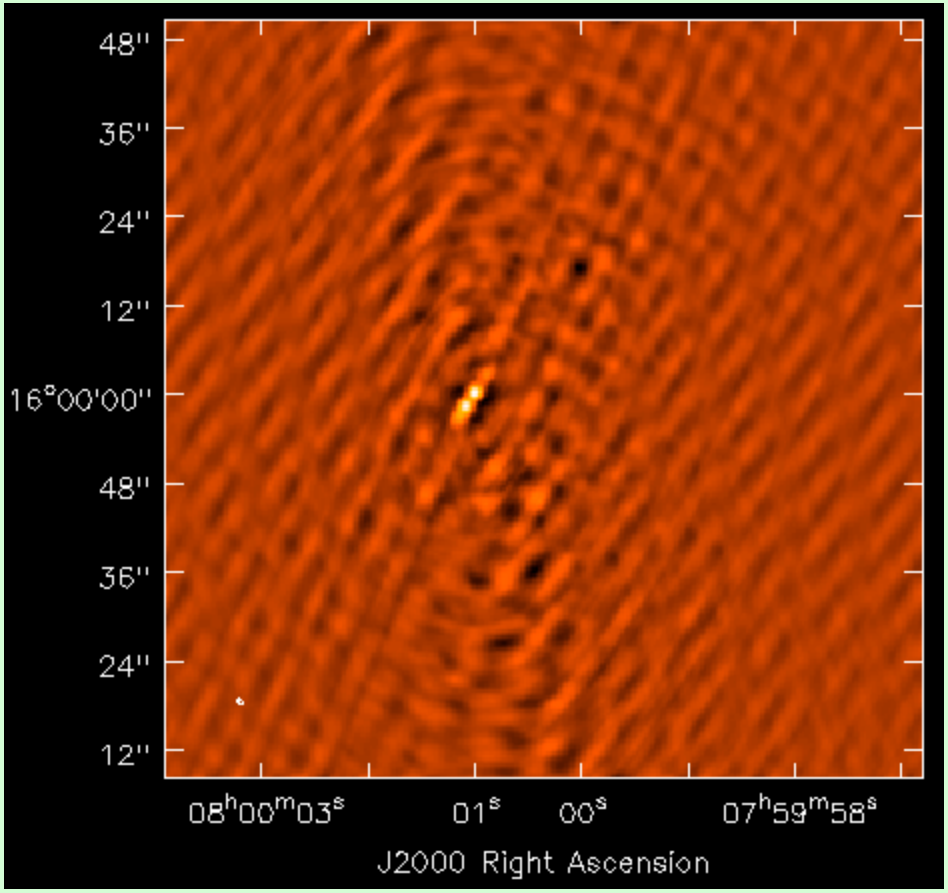

u[klambda]


agn_sma.2.clean.psf
bmaj=3.1e+00
bmin=2.6e+00

ALMA


hii.30dor.flat
min=2.8e-04
max=1.0e-01
RMS=1.7e-03
hii.clean.flat
min=-1.5e-02
max=5.6e-02
RMS=0.0e+00
hii.diff.im
min=-1.9e-03
max=6.7e-02
RMS=0.0e+00
hii.fidelity.im
min=5.4e-02
max=1.4e+00
RMS=0.0e+00
v[klambda]
u[klambda]
hii.clean.psf
bmaj=7.5e-01
bmin=3.8e-01


3c288.fits
min=-1.4e-12
max=7.9e-11
RMS=6.7e-13
40
20
0
-20
-40
40
20
0
-20
-40


48''
36''
24''
12''
16°00'00''
48''
36''
24''
12''
08h00m03s
01s
00s
07h59m58s
J2000 Right Ascension