

Simulating ALMA data:

The Observation Support Tool (OST) and other
animals

Adam Avison



Overview

- Introduction to the ALMA OST
- Using the OST
- Interlude with 'toy' interferometers
- Back to the OST
- Simulation in CASA

The Observation Support Tool

ALMA Observation Support Tool
Version 3.0

OST NEWS HELP QUEUE LIBRARY ALMA HELPDESK

OST Report: OST usage statistics during the ALMA Cycle 3 Call.

Array Setup:

Instrument: ALMA Select the desired ALMA antenna configuration.

Sky Setup:

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

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

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

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

Observation Setup:

Observing mode: Spectral Continuum Spectral or continuum observations?

Central frequency in GHz: 260.7 The value entered must be within an ALMA band.

Bandwidth in GHz 4.125 OK Select the total bandwidth for continuum observations. Enter 7.5 GHz to select ALMA recommend full continuum setup.

SPW 0: 254.0 BW 0: 1.875 Set the central frequency and bandwidth of each baseband/SPW in GHz. SPWs can only be placed within the grey shaded areas. They will be truncated in the simulation if not. More SPWs (up to SPW3) will become available as you increase the total bandwidth.

SPW 1: 252.0 BW 1: 1.5

SPW 2: 267.0 BW 2: 0.75

SPW 3: 0.0 BW 3: 0.0

Band = 6

211 260.7 275

A web-based ALMA simulator aimed at the non-interferometry expert user.

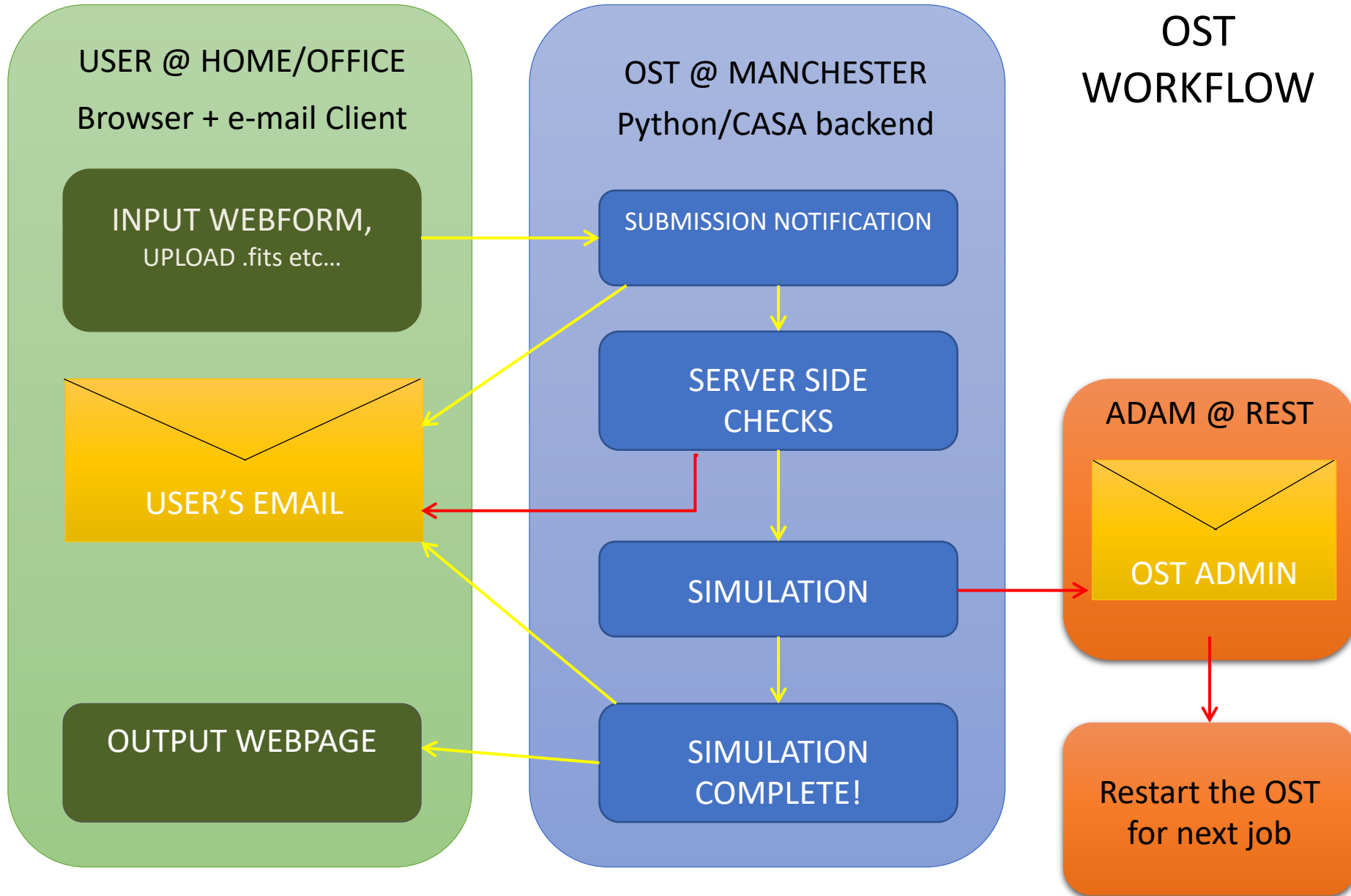
The primary version is for ALMA and has been available since ALMA Cycle 0 CfP. Since when it has been extensively used by the international community in all ALMA call for proposals.

See <http://almaost.jb.man.ac.uk>

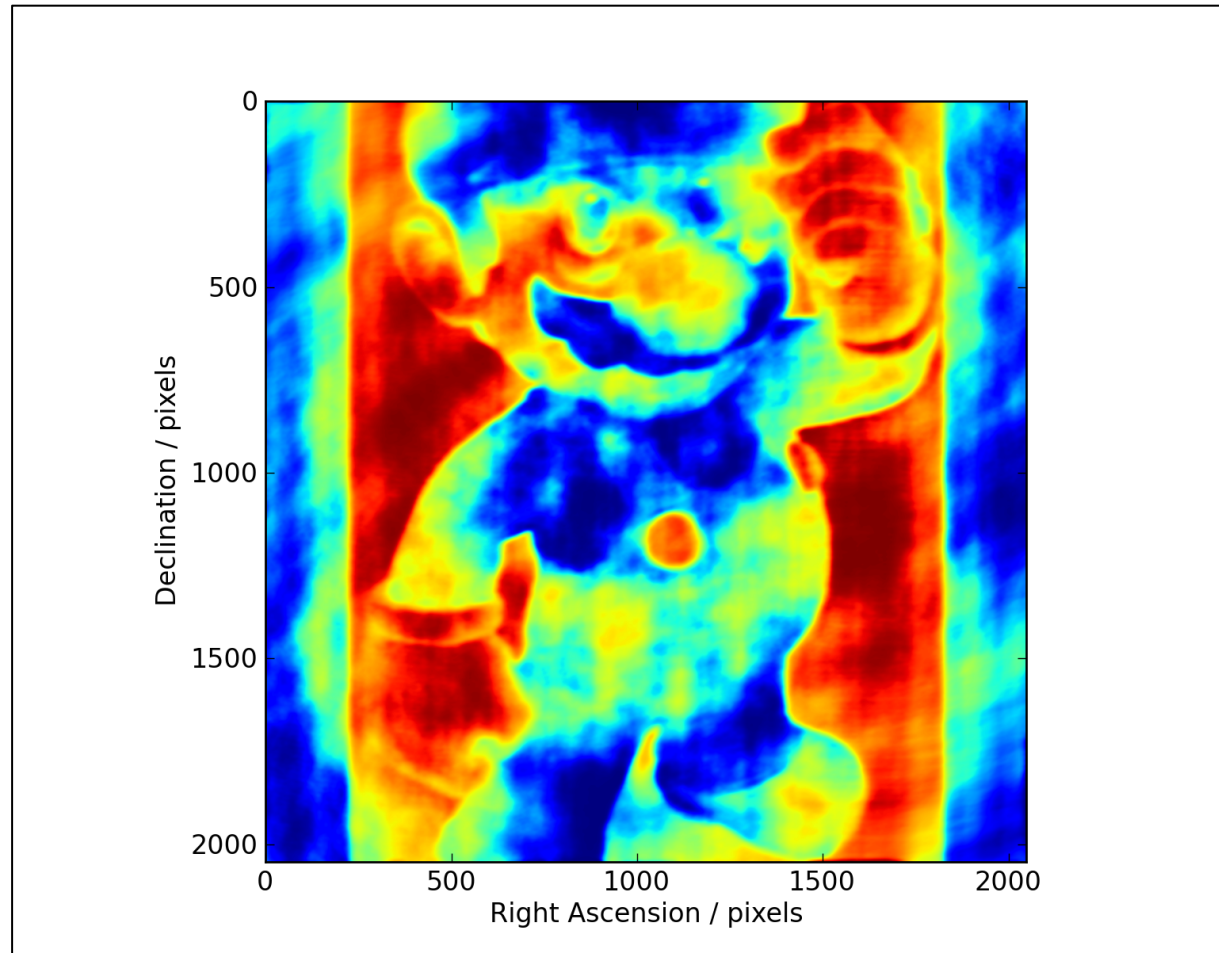
There are also eMerlin and an AVN* version.

<http://almaost.jb.man.ac.uk/emerlin>
<http://almaost.jb.man.ac.uk/AVN>

* For a hypothetical AVN telescope



OST Walkthrough



- OST Simulation of the 'Super' M-4R10 Galaxy

Whilst we let the OST run for a bit, here are some toy interferometers you can play with:

Pynterferometer

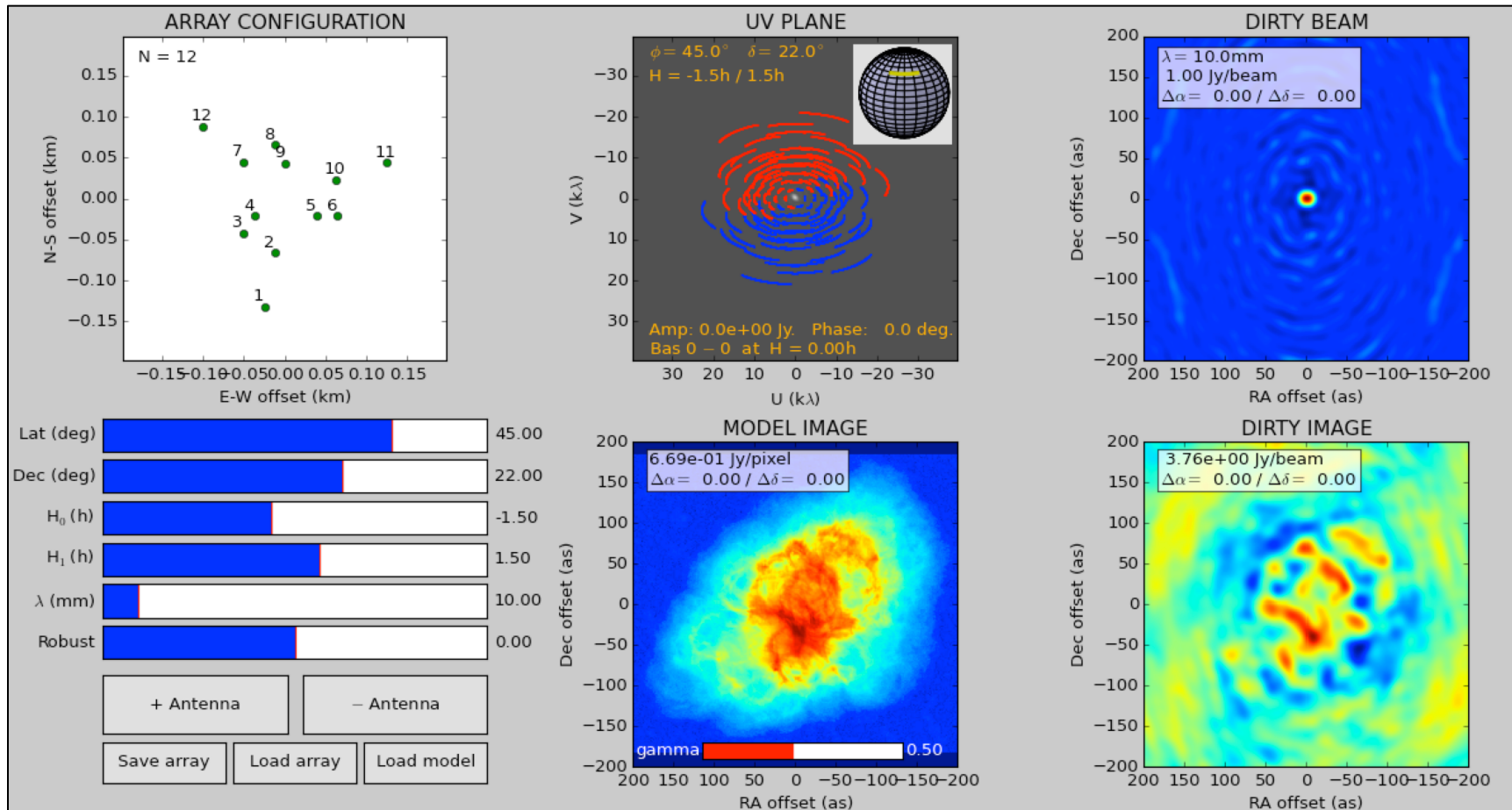
- <http://www.jb.man.ac.uk/pynterferometer/>

The screenshot displays the Pynterferometer web application interface, which is divided into several sections:

- Top Left:** A 2D grid showing a spiral arrangement of orange circles representing antenna positions.
- Top Middle:** A photograph of a large white radio telescope dish.
- Top Right:** A grayscale image of a celestial object, likely a galaxy, showing a bright central region and a diffuse outer structure.
- Bottom Left:** A control panel with various buttons for configuration, including:
 - Shape selection: Spiral, Y-shape, Line.
 - Antenna configuration: Add Antennae, Remove Antennae, Increase Array Size, Normal Array Size, Decrease Array Size.
 - Observation mode: Earth Rotation, Snapshot mode, Single Antenna.
 - Telescope selection: Use Galaxy, Use ESO, Use Nova, Use Lovell, Use Cluster, Use Dog.
 - Other: Use Webcam.
- Bottom Middle:** A status area showing:
 - Status: Done: Earth Rotation
 - Current configuration: Spiral with 50 ant.
 - Logos for ALMA (Atacama Large Millimeter/submillimeter Array) and the University of Manchester.
- Bottom Right:** A 2D plot of the antenna array in the v [k] plane. The x-axis ranges from -2000 to 2000 k, and the y-axis ranges from -2000 to 2000 k. The plot shows a dense, circular distribution of orange dots representing the antenna baselines.

APSYNSIM

- <http://www.nordic-alma.se/support/software-tools>

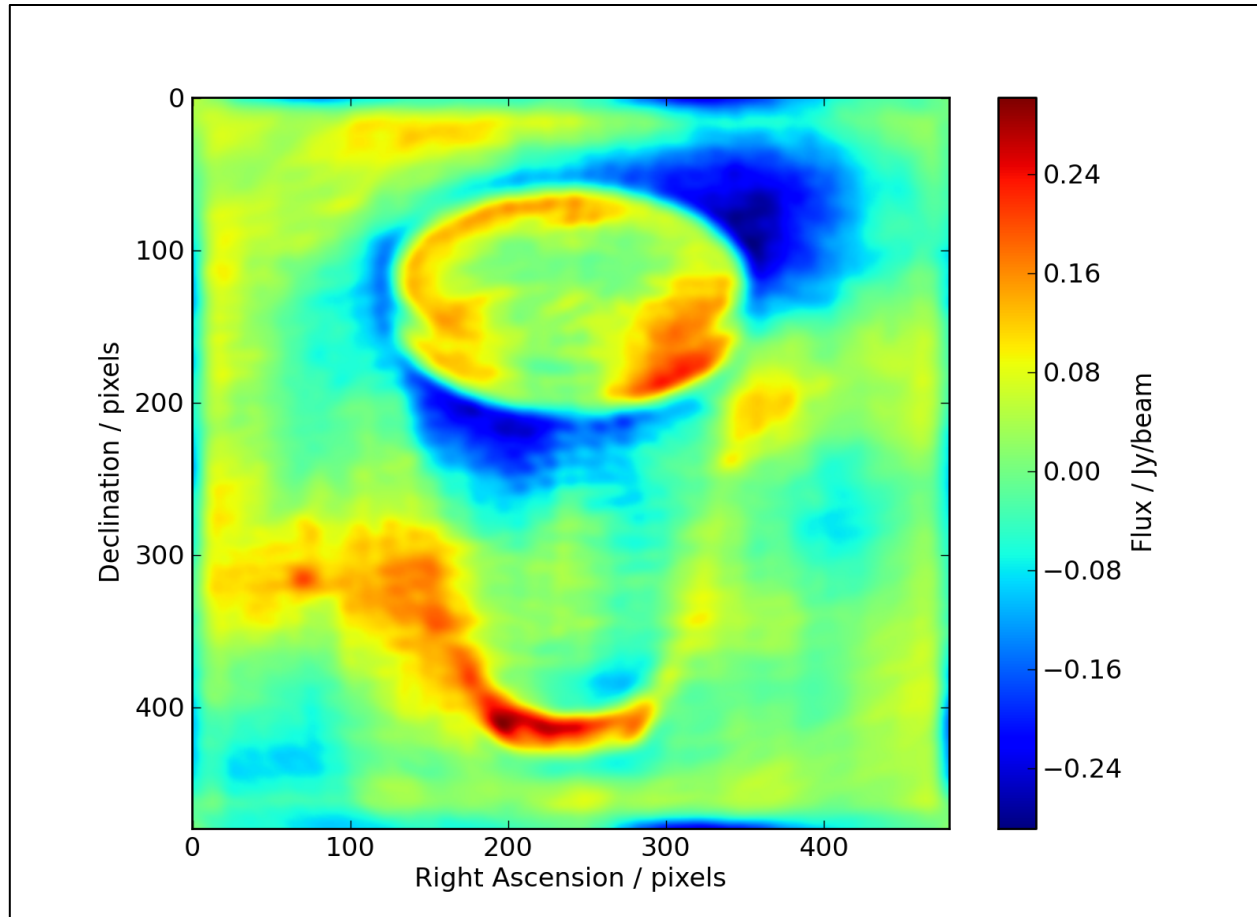


Work in progress web-based interferometer demo

http://www.jb.man.ac.uk/~aavison/ESO_javascript/gridIndex.html

Useful for considering effects of multiple arrays

Back to OST output



Simulating within CASA

(The following slides will make more sense tomorrow after Intro to CASA!
This is just to show it can be done for the sake of completeness.)

Simobserve

simobserve is used to create the simulated interferometric observations within CASA and **simanalyze** is used to analyze the output of **simobserve** (*suprise surprise*).

Creating simulations in CASA, a recipe:

- First select an existing image of the region or type of object you want to simulate, for use as your **skymodel**.
- Your input sky model can then be rescaled in pixel size, brightness, sky position, central frequency and channel width with e.g. **incell**
- The **setpointings** and **observe** parameters are then used to define the simulated observation, with similar parameters to defining real observations.
- The corruption due to the atmosphere can then be set in **thermalnoise**.
- With **antennalist** you can define which telescope you are using. CASA knows a lot... eMerlin, ALMA, VLA, ATCA, PdBI, EVN, etc etc.

```
[CASA <1>: inp simobserve
-> inp(simobserve)
# simobserve :: visibility simulation task
project          = 'sim'          # root prefix for output file names
skymodel         = ''            # model image to observe
complist        = ''            # componentlist to observe
setpointings    = True          #
  integration     = '10s'        # integration (sampling) time
  direction      = ''           # "J2000 19h00m00 -40d00m00" or "" to
                                # center on model
  mapsize        = ['', '']     # angular size of map or "" to cover
                                # model
  maptype        = 'ALMA'       # hexagonal, square (raster), ALMA, etc
  pointingspacing = ''          # spacing in between pointings or
                                # "0.25PB" or "" for ALMA default
                                # INT=lambda/D/sqrt(3), SD=lambda/D/3
obsmode         = 'int'         # observation mode to simulate [int(int
                                # erferometer)|sd(singledish)|""(none)
                                # ]
  antennalist    = 'alma.out10.cfg' # interferometer antenna position file
  refdate       = '2014/05/21'    # date of observation - not critical
                                # unless concatting simulations
  hourangle     = 'transit'      # hour angle of observation center e.g.
                                # "-3:00:00", "5h", "-4.5" (a number
                                # without units will be interpreted as
                                # hours), or "transit"
  totaltime     = '7200s'        # total time of observation or number
                                # of repetitions
  caldirection  = ''            # pt source calibrator [experimental]
  calflux       = '1Jy'         #
outframe        = 'LSRK'        # spectral frame of MS to create
thermalnoise    = 'tsys-atm'    # add thermal noise: [tsys-atm|tsys-
                                # manual|""]
  user_pvw      = 0.5           # Precipitable Water Vapor in mm
  t_ground      = 269.0         # ambient temperature
  seed          = 11111         # random number seed
leakage         = 0.0           # cross polarization (interferometer
                                # only)
graphics        = 'both'       # display graphics at each stage to
                                # [screen|file|both|none]
verbose         = False        #
overwrite       = True         # overwrite files starting with
                                # $project
```

Simanalyze

Here we convert the CASA MS into an image file.

- The **image** parameter effectively acts like CLEANing a real dataset with iteration, weighting etc
- Next the **analyze** parameter defines which output images you would like from your analysis. Such as Clean image, UV coverage and image

```
[CASA <2>: inp simanalyze
-----> inp(simanalyze)
# simanalyze :: image and analyze measurement sets created with simobserve
project      = 'sim'      # root prefix for output file names
image       = True       # (re)image $project.*.ms to
                        # $project.image
vis         = 'default'  # Measurement Set(s) to image
modelimage  = ''        # lower resolution prior image to use
                        # in clean e.g. existing total power
                        # image
imsize      = 0         # output image size in pixels (x,y) or
                        # 0 to match model
imdirection = ''        # set output image direction,
                        # (otherwise center on the model)
cell        = ''        # cell size with units e.g. "10arcsec"
                        # or "" to equal model
interactive = False     # interactive clean? (make sure to set
                        # niter>0 also)
niter       = 0         # maximum number of iterations (0 for
                        # dirty image)
threshold   = '0.1mJy'  # flux level (+units) to stop cleaning
weighting   = 'natural' # weighting to apply to visibilities.
                        # briggs will use robust=0.5
mask        = []        # Cleanbox(es), mask image(s),
                        # region(s), or a level
outertaper  = []        # uv-taper on outer baselines in uv-
                        # plane
pbcor       = True      # correct the output of synthesis
                        # images for primary beam response?
stokes      = 'I'       # Stokes params to image
featherimage = ''      # image (e.g. total power) to feather
                        # with new image

analyze     = False     # (only first 6 selected outputs will
                        # be displayed)
graphics    = 'both'   # display graphics at each stage to
                        # [screen|file|both|none]
verbose     = False
overwrite   = True     # overwrite files starting with
                        # $project
dryrun      = False    # only print information [experimental;
                        # only for interferometric data]
logfile     = ''
```

Simalma

- A wrapper of simobserve and simanalyze which has some of these tasks parameters set to typical ALMA values.