MANCHESTER 1824

The University of Manchester

Simulating ALMA data:

The Observation Support Tool (OST) and other animals

Adam Avison

ALMA

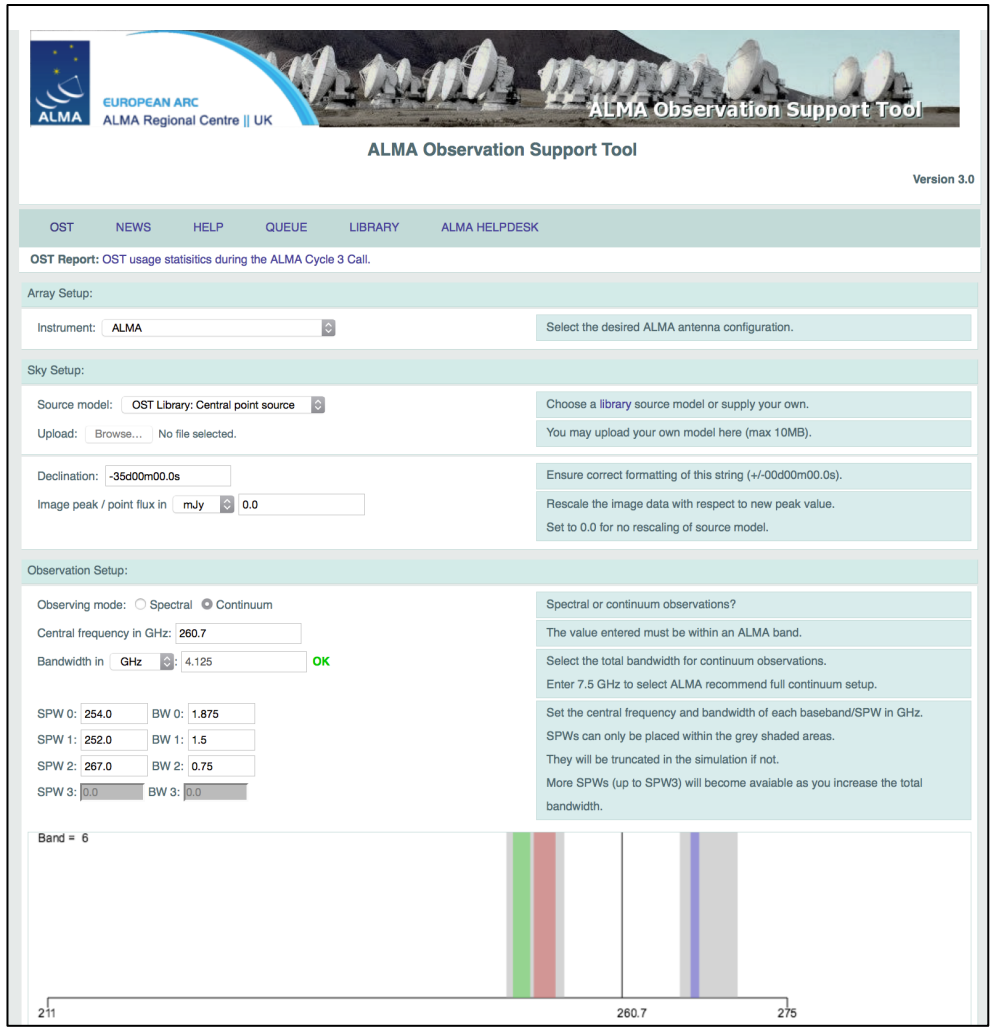
EUROPEAN ARC

ALMA Regional Centre || UK

Overview

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

The Observation Support Tool



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.

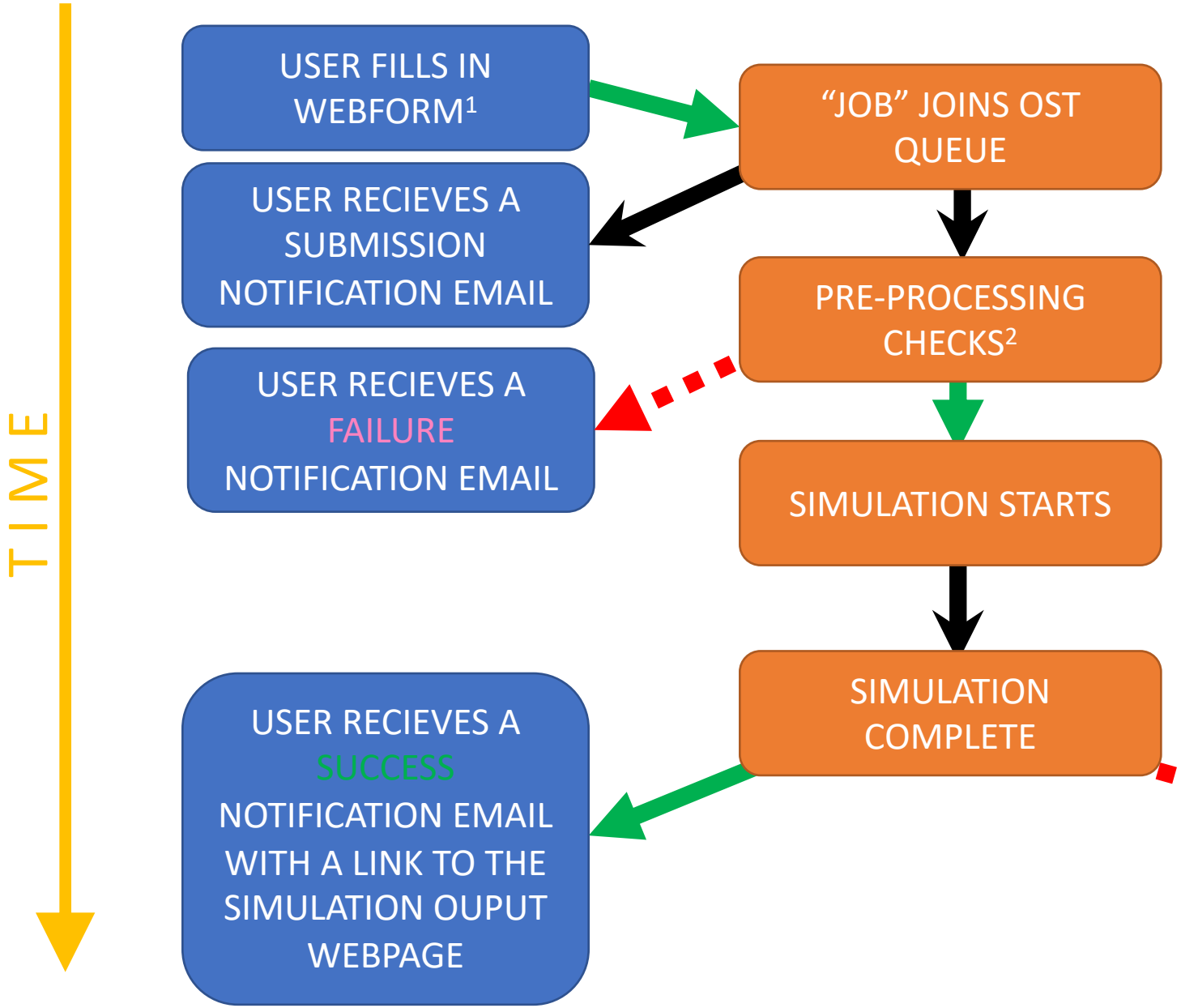
See *http://almaost.jb.man.ac.uk*

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 mode of operation



SIDENOTES:

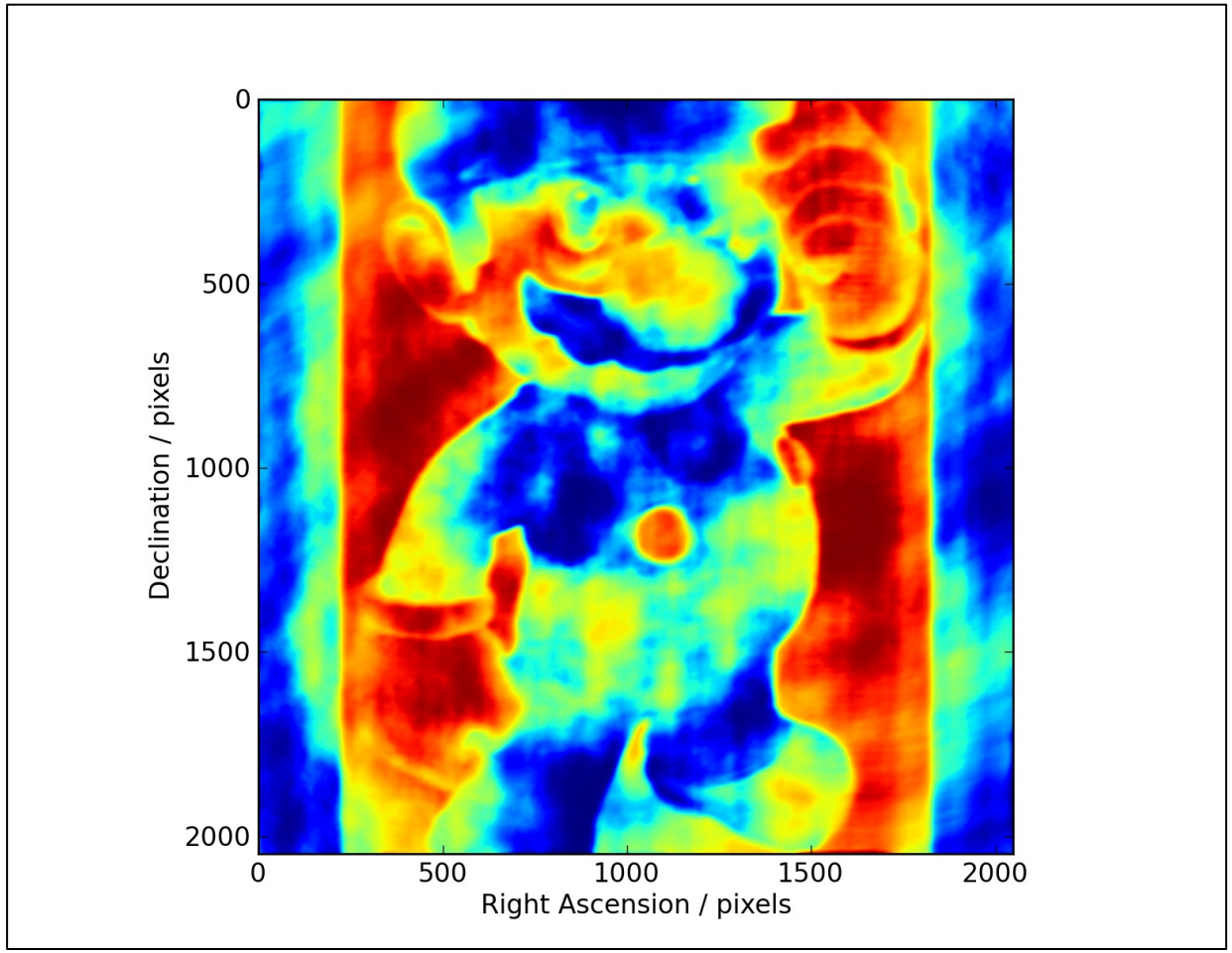
1. The webform contains a lot of "in browser" checks of user input, to prevent submitting badly formatted data.
2. Pre-processing checks catch most problem submission not caught in browser.

KEY: Automatic; On Pass; On Fail

THE OST:
1. EMAILS ITS ADMIN (me).
2. MOVES THE FAILED JOB OUT OF THE QUEUE.
3. RESTARTS ITSELF ON THE NEXT JOB

OST ADMIN EMAILS USER AN EXPLANATION OF FAILURE.

OST Walkthrough

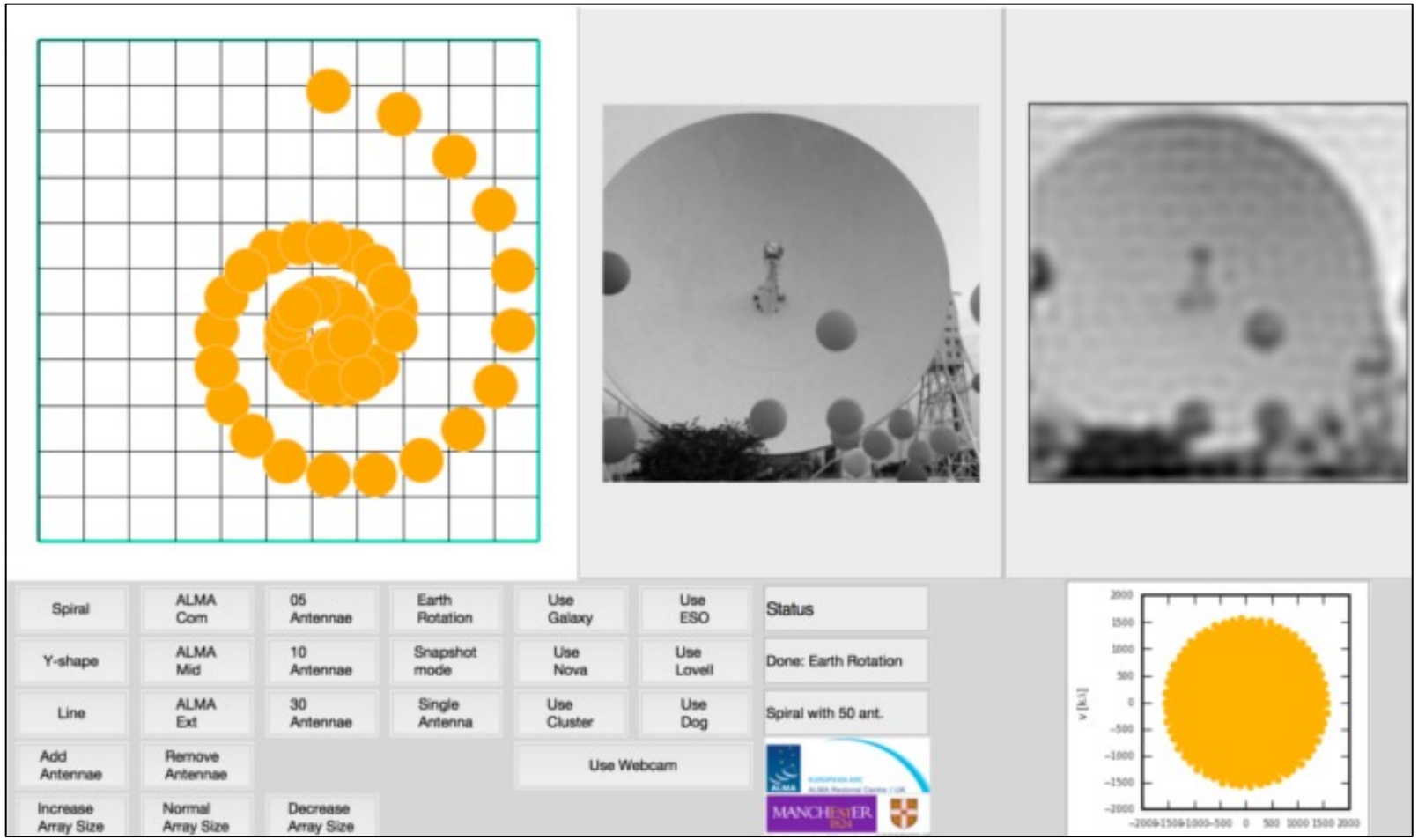


- OST Simulation of the ‘Super’ M-4R10 Galaxy

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

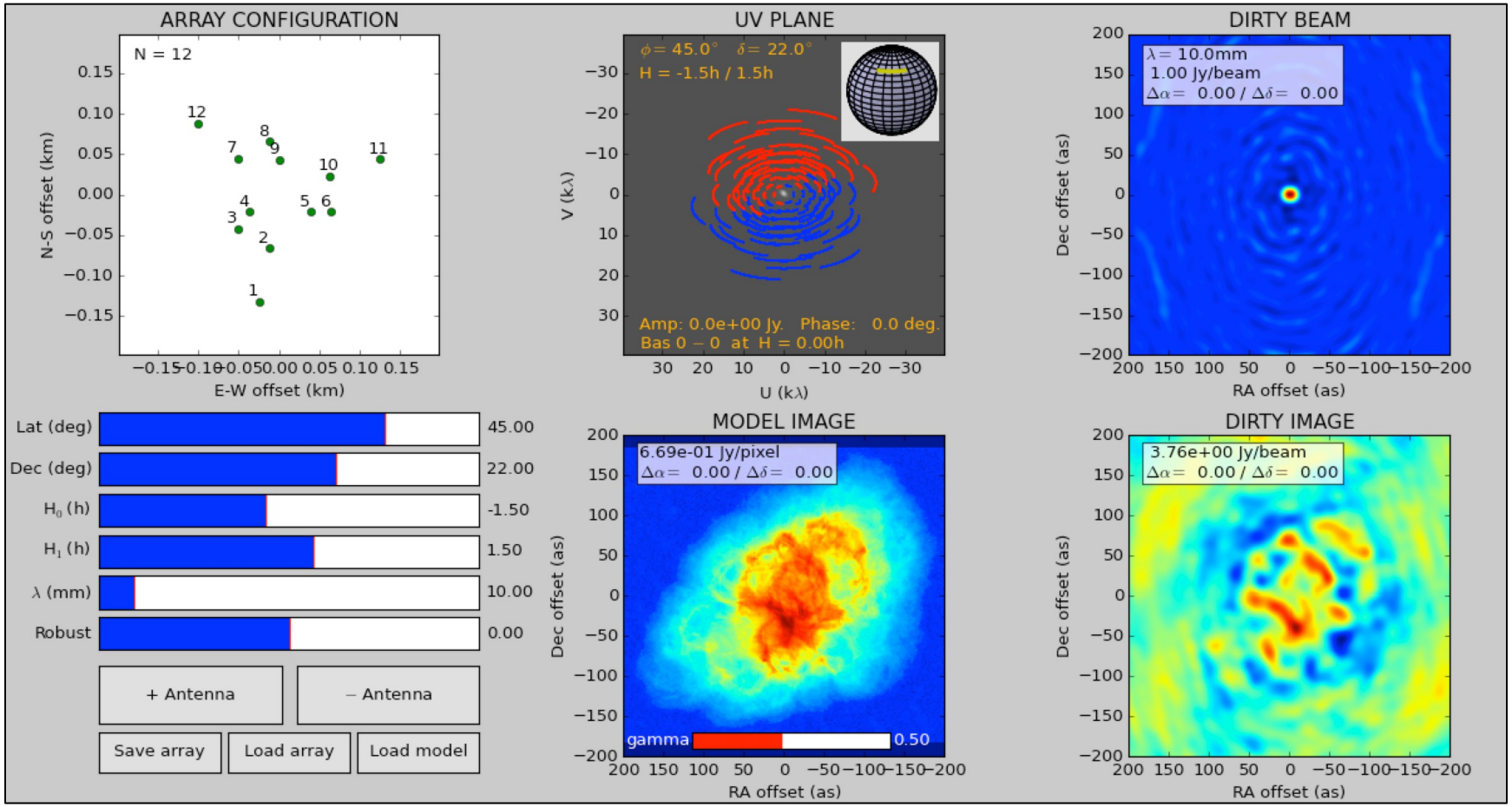
Pynterferometer

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

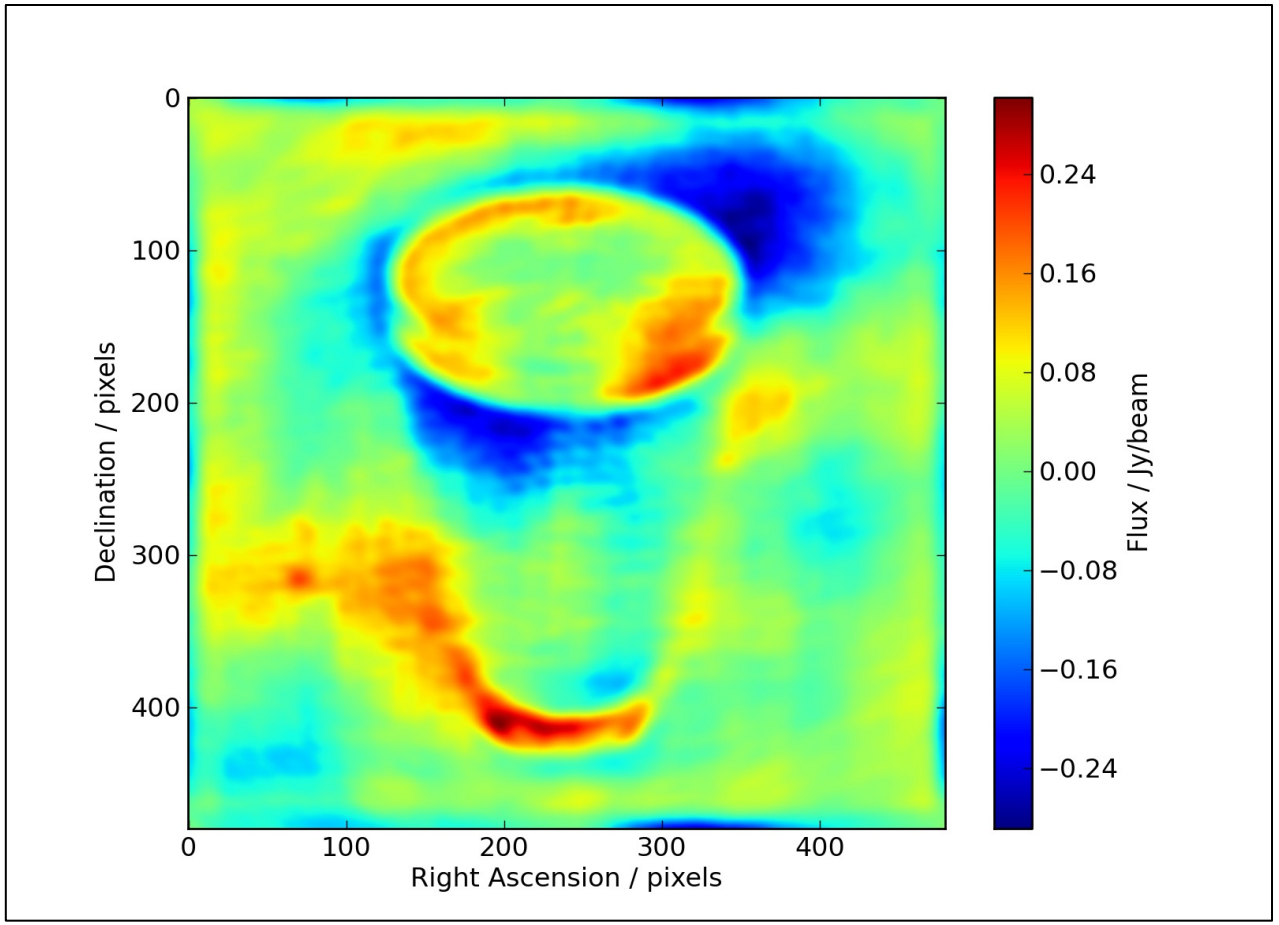


APSYNSIM

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



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 suprise).*

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_pwv       =        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 interfermetric data]
logfile             =         ''
```

Simalma

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