Using CASA to Simulate Interferometer Observations

Nuria Marcelino
North American ALMA Science Center
Simulating Interferometer Data

• Take a model image and simulate how it would look if observed by ALMA or the EVLA.
  o Other arrays (e.g., SMA, CARMA, etc.) also included

• Explore the effects of:
  o Number of antennas
  o Antenna configuration
  o Length of observation
  o Thermal noise
  o Phase noise

• Functionality included in CASA via tasks `simobserve` and `simanalyzer` (nee `simdata`).

• CASAguides includes several walkthroughs:
Basic Simulation Workflow

In CASA...

1. **Model Sky Distribution** (FITS, image, components)
2. **simobserve**
3. **Simulated Measurement Set** (calibrated u-v data)
4. **simanalyze**
5. **Simulated Image & Analysis Plots Comparing “Observed”/original image**
Simulation Tasks

- **simobserve** simulates interferometric (and single dish) observations of a source.

- **simanalyze** images and analyzes these simulations.

<table>
<thead>
<tr>
<th>Visualization</th>
<th>Simulation</th>
<th>Single dish</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearplot</td>
<td>simobserve</td>
<td>asap_init</td>
<td>browsetable</td>
</tr>
<tr>
<td>imview</td>
<td>simdata</td>
<td>sdbaseline</td>
<td>caltabconvert</td>
</tr>
<tr>
<td>msview</td>
<td>simobserve</td>
<td>sdcal</td>
<td>clearplot</td>
</tr>
<tr>
<td>plotants</td>
<td></td>
<td>sdcoadd</td>
<td>clearstat</td>
</tr>
<tr>
<td>plotcal</td>
<td></td>
<td>sdfit</td>
<td>concat</td>
</tr>
<tr>
<td>plotms</td>
<td></td>
<td>sflag</td>
<td>conjugatevis</td>
</tr>
<tr>
<td>plotuv</td>
<td></td>
<td>sflagmanager</td>
<td>find</td>
</tr>
<tr>
<td>plotxy</td>
<td></td>
<td>sdgrid</td>
<td>help_par.parameter</td>
</tr>
<tr>
<td>viewer</td>
<td></td>
<td>sdimaging</td>
<td>help_taskname</td>
</tr>
<tr>
<td>(plotweather)</td>
<td></td>
<td>sdimprocess</td>
<td>imview</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdlist</td>
<td>msview</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdmath</td>
<td>plotms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdplot</td>
<td>rmtables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sreduce</td>
<td>startup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdsave</td>
<td>taskhelp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdprocess</td>
<td>tasklist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdspikes</td>
<td>testconcat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdscale</td>
<td>toolhelp</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdsmooth</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sdpimaging</td>
<td></td>
</tr>
</tbody>
</table>

“tasklist” output
simobserve

- simulates interferometer observations of a source.

```
# simobserve :: mosaic simulation task:
project = 'sim'
# root prefix for output file names
skymodel = ''
# model image to observe
complist = ''
# componentlist to observe
setpointings = True
# integration (sampling) time
integration = '10s'
direction = ''
# "J2000 19h00m00 -40d00m00" or "" to center on model
mmapsizE = [' ', ' ']
# angular size of map or "" to cover model
mtype = 'ALMA'
# hexagonal, square, etc
pointingspac = ''
# spacing in between pointings or "0.25PB" or "" for 0.5 PB

obsnode = 'int'
# observation mode to simulate [int(interferometer)|sd(singledish)|""(none)]
antennalist = 'alma.out10.cfg'
# interferometer antenna position file
refdate = '2012/05/21'
# date of observation - not critical unless concatting simulations
hourangle = 'transit'
# hour angle of observation center e.g. -3:00:00, or "transit"
totaltime = '7200s'
# total time of observation or number of repetitions
caldirection = ''
# pt source calibrator [experimental]
calflux = '1Jy'

thermalnoise = 'Sopapilla'
# add thermal noise: [tsys-atm|tsys-manual|""]
leakage = 0.0
# cross polarization (interferometer only)
graphics = 'both'
# display graphics at each stage to [screen|file|both|none]
verbose = False
# overwrite files starting with $project
overwrite = True
# If true the taskname must be started using simobserve(...)
async = False
```

"inp simobserve" output
**CASA Refresher**

- **In casapy** type
  
default simobserve

```python
# simobserve :: mosaic simulation task:
project = 'sim'  # root prefix for output file names
skymodel = ''    # model image to observe
complist = ''    # componentlist to observe
setpointings = True  # integration (sampling) time
integration = '10s'  # "J2000 19h00m00 -40d00m00" or "" to center on model
direction = ''  # angular size of map or "" to cover model
mapsize = ['.', '.']  # hexagonal, square, etc
maptype = 'ALMA'  # spacing in between pointings or "0.25PB" or "" for 0.5 PB
pointingsspacing = ''

obsmode = 'int'  # observation mode to simulate [int(interferometer)|lsd(singledish)|""(none)]
anennalist = 'alma.out10.cfg'  # interferometer antenna position file
refdate = '2012/05/21'  # date of observation - not critical unless concating simulations
hourangle = 'transit'  # hour angle of observation center e.g. -3:00:00, or "transit"
totalltime = '7200s'  # total time of observation or number of repetitions
caldirection = ''  # pt source calibrator [experimental]
calflux = '1Jy'
thermalnoise = 'Sopapilla'  # add thermal noise: [tsys-atm|tsys-manual|"
leakage = 0.0  # cross polarization (interferometer only)
grafics = 'both'  # display graphics at each stage to [screen|file|both|none]
verbose = False  # overwrite files starting with $project
overwrite = True  # If true the taskname must be started using simobserve(...)""
async = False
```

---

**NRAO**
CASA Refresher

- inp shows parameter names

```plaintext
# simobserve :: mosaicsimulation task:
project = 'sim' # root prefix for output file names
skymodel = ' ' # model image to observe
complist = ' ' # componentlist to observe
setpointings =
  integration = '10s' # integration (sampling) time
  direction = ['J2000 19h00m00 -40d00m00' or ''] to center on model
  mapsize = [','] # angular size of map or '' to cover model
  maptypes = ['ALMA'] # hexagonal, square, etc
  pointingspacing = '' # spacing in between pointings or '0.25PB' or '' for 0.5 PB

obsmode =
  antennalist = 'alma.out10.cfg' # interferometer antenna position file
  reffdate = 2012/05/21 # date of observation - not critical unless concatting simulations
  hourangle = 'transit' # hour angle of observation center e.g. -3:00:00, or "transit"
  totaltime = '7200s' # total time of observation or number of repetitions
  calldirection = ' ' # pt source calibrator [experimental]

thermalnoise = 'lopapilla' # add thermal noise: [tsys-aml|tsys-manuall'"
  leakage = 0.0 # cross polarization (interferometer only)
  graphics = 'both' # display graphics at each stage to [screen|file|both|none]
  verbose = False # overwrite files starting with $project
  overwrite = True # If true the taskname must be started using simobserve(...)"}
```
CASA Refresher

- `inp` shows current value (change, e.g., by `project = "myproj"`)

```python
# simobserve :: simobserve simulation task:
project = 'sim'  # root prefix for output file names
skymodel = ''    # model image to observe
complist = ''    # componentlist to observe
setpointings = True  # 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, etc
pointingspacings = ''  # spacing in between pointings or "0.25PB" or "" for 0.5 PB

# obsnode
antennalist = 'alma.out10.cfg'  # interferometer antenna position file
refdate = '2012/05/21'  # date of observation - not critical unless concating simulations
hourangle = 'transit'  # hour angle of observation center e.g. -3:00:00, or "transit"
totaltime = '72000s'  # total time of observation or number of repetitions
caldirection = ''  # not source calibrator [experimental]
calflux = '1Jy'

# thermalnoise
leakage = 0.0  # cross polarization (interferometer only)

# graphics
verbatim = False  # display graphics at each stage to [screen|file|both|none]
overwrite = True  # overwrite files starting with $project
async = False  # If true the taskname must be started using simobserve(...)"
```

**Invalid Value**
CASA Refresher

- inp shows brief description

```python
# simobserve :: mosaic simulation task:
project = 'sim'
skymodel = '
complist = '
setpointings = True
integration = '10s'
direction = '
mapsiz = [', ']
maptype = 'ALMA'
pointingspacing =

obsmode = 'int'
antennalist = 'alma.out10.cfg'
refdate = '2012/05/21'
hourangle = 'transit'
totalltime = '7200s'
caldirection =
calflux = '1Jy'

thermalnoise = 'Sopapilla'
leakage = 0.0

graphics = 'both'
verbose = False
overwrite = True
async = False

root prefix for output file names
# model image to observe
# componentlist to observe
# integration (sampling) time
# "J2000 19h00m00.40d00m00" or "" to center on model
# angular size of map or "" to cover model
# hexagonal, square, etc
# spacing in between pointings or "0.25PB" or "" for 0.5 PB
# observation mode to simulate [int(interferometer)|lsd(singledish)]""(none)]
# interferometer antenna position file
# date of observation - not critical unless concatting simulations
# hour angle of observation center e.g. -3:00:00, or "transit"
# total time of observation or number of repetitions
# pt source calibrator [experimental]
# add thermal noise: [tsys-atm|tsys-manual]"
# cross polarization (interferometer only)
# display graphics at each stage to [screen|file|both|none]

overwrite files starting with $project
# If true the taskname must be started using simobserve(...)```
CASA Refresher

• Change values by
  
  \texttt{project = "myproj"}
  
  \texttt{inp}

```python
# simobserve :: mosaic simulation task:
project = 'myproj'  # root prefix for output file names

# complist
# setpointings
integration = '10s'  # integration (sampling) time
direction = ''  # "J2000 15h00m00 -40d00m00" or "" to center on model
mapsize = ['', '']  # angular size of map or "" to cover model
maptype = 'ALMA'  # hexagonal, square, etc
pointingspacing = ''  # spacing in between pointings or "0.25PB" or "" for 0.5 PB

# obsmode
antennalist = 'alma.out10.cfg'  # interferometer antenna position file
refdate = '2012/05/21'  # date of observation - not critical unless concatting simulations
hourangle = 'transit'  # hour angle of observation center e.g. -3:00:00, or "transit"
totalltime = '7200s'  # total time of observation or number of repetitions
caldirection = ''  # pt source calibrator [experimental]
calflux = '1Jy'

# thermalnoise
leakage = 0.0  # add thermal noise; [tsys-atm|itsys-manual|""]
graphics = 'both'  # cross polarization (interferometer only)
verbose = False  # display graphics at each stage to [screen|file|both|none]
overwrite = True  # overwrite files starting with $project
async = False  # If true the taskname must be started using simobserve(...)```
When all parameters are set, execute with “go simobserve”

If you get stuck:

- Type “tasklist” to see all tasks
- Type “help taskname” to get help on taskname
- Type “default taskname” to set the default inputs
- Type “inp” to review the inputs of the current task
- Ask!
Basic Simulation Workflow

In CASA...

- Model Sky Distribution (FITS, image, components)
- Simulated Measurement Set (calibrated u-v data)
- Simulated Image & Analysis Plots Comparing “Observed”/original image
What Defines a Simulation?

**Model Sky Distribution**
(Required)
What does the sky really look like in your field?

**Telescope**
(Required)
Number of Antennas, Configuration, Diameter

**Observation**
(Required)
Integration time, scan length, pointing centers

**Corruption**
(Optional)
Thermal noise, phase noise, polarization leakage
• Model sky distribution as FITS file or “component list”

```
# simobserve :: mosaic simulation task:

# root prefix for output file names
Project = Sim

# model image to observe
skymodel =

# component list to observe
complist =

integration = '10s'
direction =

mapsize = ['','']
maptype = 'ALMA'
pointingspacing =


# integration (sampling) time
# "J2000 19h00m00 -40d00m00" or "" to center (model)
# angular size of map or "" to cover (model)
# hexagonal, square, etc
# spacing in between pointings or "0.25PB" or "" for 0.5 PB

# observation mode to simulate [int(interferometer)|lsd(singledish)|""(none)]
# interferometer antenna position file
# date of observation - not critical unless concatting simulations
# hour angle of observation center e.g. -3:00:00, or "transit"
# total time of observation or number of repetitions
# pt source calibrator [experimental]

# add thermal noise: [tsys-atm|tsys-manual|""]
# cross polarization (interferometer only)
# display graphics at each stage to [screen|file|both|none]
# overwrite files starting with $project
# If true the taskname must be started using simobserve(...)```
simobserve

- Telescope via configuration file.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>project</td>
<td>'sim'</td>
</tr>
<tr>
<td>skymodel</td>
<td>''</td>
</tr>
<tr>
<td>complist</td>
<td>''</td>
</tr>
<tr>
<td>antennalist</td>
<td>'alma.out10.cfg'</td>
</tr>
<tr>
<td>refdate</td>
<td>'2012/05/21'</td>
</tr>
<tr>
<td>hourangle</td>
<td>'transit'</td>
</tr>
<tr>
<td>totaltime</td>
<td>'7200s'</td>
</tr>
<tr>
<td>caldirection</td>
<td>''</td>
</tr>
<tr>
<td>calflux</td>
<td>'1Jy'</td>
</tr>
<tr>
<td>thermalnoise</td>
<td>'Sopapilla'</td>
</tr>
<tr>
<td>leakage</td>
<td>0.0</td>
</tr>
<tr>
<td>graphics</td>
<td>'both'</td>
</tr>
<tr>
<td>verbose</td>
<td>False</td>
</tr>
<tr>
<td>overwrite</td>
<td>True</td>
</tr>
<tr>
<td>async</td>
<td>False</td>
</tr>
</tbody>
</table>

Telescope (Required)
Number of Antennas, Configuration, Diameters
**Observations defined via** setpointings **and** obsmode

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>project</td>
<td>'sim'</td>
</tr>
<tr>
<td>skymodel</td>
<td></td>
</tr>
<tr>
<td>compelit</td>
<td></td>
</tr>
<tr>
<td>setpointings</td>
<td>True</td>
</tr>
<tr>
<td>integration</td>
<td>'10s'</td>
</tr>
<tr>
<td>direction</td>
<td></td>
</tr>
<tr>
<td>mapsize</td>
<td>['', '', '']</td>
</tr>
<tr>
<td>maptype</td>
<td>'ALMA'</td>
</tr>
<tr>
<td>pointingspacing</td>
<td></td>
</tr>
<tr>
<td>obsmode</td>
<td>'int'</td>
</tr>
<tr>
<td>antennalist</td>
<td>'alma.out10.cfg'</td>
</tr>
<tr>
<td>refdate</td>
<td>'2012/05/21'</td>
</tr>
<tr>
<td>hourangle</td>
<td>'transit'</td>
</tr>
<tr>
<td>totaltime</td>
<td>'7200s'</td>
</tr>
<tr>
<td>caldirection</td>
<td></td>
</tr>
<tr>
<td>calflux</td>
<td>'1Jy'</td>
</tr>
<tr>
<td>thermalnoise</td>
<td>'Sopapilla'</td>
</tr>
<tr>
<td>leakage</td>
<td>0.0</td>
</tr>
<tr>
<td>graphics</td>
<td>'both'</td>
</tr>
<tr>
<td>verbose</td>
<td>False</td>
</tr>
<tr>
<td>overwrite</td>
<td>True</td>
</tr>
<tr>
<td>async</td>
<td>False</td>
</tr>
</tbody>
</table>

**Observation**

(Required)

Integration time, scan length, pointing centers
• Corruption with thermalnoise & toolkit

```plaintext
# simobserve :: mosaic simulation task:
project = 'sim'
skymodel = '
complist = '
setpointings = True
integration = '10s'
direction = '
mapsize = ['','']
maptype = 'ALMA'
pointingspacing =

obsmode = 'int'
antennalist = 'alma.out10.cfg'
refdate = '2012/05/21'
hourangle = 'transit'
toaltime = '7200s'
caldirection =
calflux = '1Jy'

thermalnoise = 'Sopapilla'
leakage = 0.0
graphics = 'both'
verbose = False
overwrite = True
async = False

# root prefix for output file names
# model image to observe
# componentlist to observe
# integration (sampling) time
# "J2000 19h00m00 -40d00m00" or "" to center on model
# angular size of map or "" to cover model
# hexagonal, square, etc
# spacing in between pointings or "0.25PB" or "" for 0.5 PB
# observation mode to simulate [int(interferometer)|sd(singledish)|""(none)]
# interferometer antenna position file
# date of observation - not critical unless concatting simulations
# hour angle of observation center e.g. -3:00:00, or "transit"
# total time of observation or number of repetitions
# pt source calibrator [experimental]
# add thermal noise: [tsys-atm|tsys-manual|"
# cross polarization (interferometer only)
# display graphics at each stage to [screen|file|both|none]
# overwrite files starting with $project
# If true the taskname must be started using -python
```

Corruption (Optional)

Thermal noise, phase noise, polarization leakage
• Model sky distribution as FITS file or “component list”

```python
# simobserve :: mosaic simulation task:
# root prefix for output file names
skymodel = ''
model image to observe
complist = ''
component list to observe
integration = '10s'
direction = ''
mapsizex = ['', ']
gap = ['ALMA']
pointingspacing = ''

obsmode = 'int'
antennalist = 'alma.out10.cfg'
reftype = '2012/05/21'
hourangle = 'transit'
totalltime = '7200s'
calcdirection = ''
calflux = '1Jy'

thermalnoise = 'Sopapilla'
leakage = 0.0
gradients = 'both'
verbose = False
overwrite = True
async = False
```

What does the sky really look like in your field?
Input Sky Model

• Model sky distribution as FITS file. *simobserve* needs:

  o Coordinates
  o Brightness units
  o Pixel scale (angular and spectral)
  o Polarization*

• These may be specified in your FITS header or supplied/over-written by *simobserve*.

```python
skymodel = '30dor.fits'

inbright = ...
indirection = ...
incell = ...
incenter = ...
inwidth = ...

complist = ...
```

# model image to observe
# scale surface brightness of brightest pixel e.g. "1.2Jy/pixel"
# set new direction e.g. "J2000 19h00m00 -40d00m00"
# set new cell/pixel size e.g. "0.1arcsec"
# set new frequency of center channel e.g. "89GHz" (required even for 2D model)
# set new channel width e.g. "10MHz" (required even for 2D model)

# componentlist to observe
Input Sky Model

• Alternatively, supply a Gaussian “component list.” Example at:


```
skymodel = '30dor.fits'
inbright =
indirection =
incell =
incenter =
inwidth =
```

```
modelimage = '30dor.fits'
# model image to observe
# scale surface brightness of brightest pixel e.g. "1.2Jy/pixel"
# set new direction e.g. "J2000 19h00m00 -40d00m00"
# set new cell/pixel size e.g. "0.1arcsec"
# set new frequency of center channel e.g. "89GHz" (required even for 2D model)
# set new channel width e.g. "10MHz" (required even for 2D model)
# compoentlist to observe
```

Simulation Guide Component Lists (CASA 3.3)

```
# Simulating Observations in CASA
This guide is applicable to CASA version 3.3.
To create a script of the Python code on this page see Extracting scripts from these tutorials.

Contents
1 Explanation of the guide
2 Getting Started
3 CASA Basics
4 Making a Simple FITS Image
5 Simulating Observations with a FITS Image and a Component List
6 Simulating Observations with Just a Component List
```
Simple Example

- Simulate observing 1mm dust continuum in a 30-Doradus (LMC)-like region at the distance of M31/M33 (800 kpc).

- We have a near-IR image of 30 Doradus, will need to:
  
  o Scale the brightness and observing frequency
  
  o Adjust the pixel scale (move it from 50-800 kpc)
  
  o Set a new position
  
  o Define the observations
    INTEGRATION TIME, TELESCOPE, ETC.
Simple Example

- **inbright** = “0.6mJy/pixel”
  
  \textbf{REQUIRES SPECTRAL MODEL/OTHER KNOWLEDGE TO ESTIMATE (SCIENCE!)}

- **Indirection** = “J2000 10h00m00s -40d00m00s”

- **incell** = “0.15arcsec”
  
  \textbf{NATIVE CELL SIZE = 2.3”, MOVING FROM 50 KPC TO 800 KPC SCALE BY 50/800}

- **incenter** = “230GHz”, **inwidth** = “2GHz”
  
  \textbf{NEED TO SUPPLY OBSERVING FREQUENCY & BANDWIDTH (HERE 1MM DUST CONTINUUM)}
Simple Example

- **inbright** = “0.6mJy/pixel”
  
  **REQUIRES SPECTRAL MODEL/OTHER KNOWLEDGE TO ESTIMATE (SCIENCE!)**

- **Indirection** = “J2000 10h00m00s -40d00m00s”

- **incell** = “0.15arcsec”
  
  **NATIVE CELL SIZE = 2.3”, MOVING FROM 50 KPC TO 800 KPC SCALE BY 50/800**

- **incenter** = “230GHz”, **inwidth** = “2GHz”
  
  **NEED TO SUPPLY OBSERVING FREQUENCY & BANDWIDTH (HERE 1MM DUST CONTINUUM)**
• Telescope via configuration file.

```
# simobserve :: mosaic simulation task:
project = 'sim'
skymodel = ''
complist = ''
setpointings = True
integration = '10s'
direction = ''
mapsizex = [' ', ' ']
maptype = 'ALMA'
pointraster = ''
pointingspacing =

dsource = Inc
antennalist = 'alma.out10.cfg'
refdate = '2012/05/21'
hourangle = 'transit'
totaltime = '7200s'
calcdirection =
calflux = '1Jy'

thermalnoise = 'Sopapilla'
leakage = 0.0
graphics = 'both'
verbose = False
overwrite = True
async = False

# root prefix for output file names
# model image to observe
# componentlist to observe
# integration (sampling) time
# "J2000 19h00m00.0-40d00m00.0" or "" to center on model
# angular size of map or "" to cover model
# hexagonal, square, etc
# spacing in between pointings or "0.25PB" or "" for 0.5 PB
# observation mode to simulate [int(interferometer)lsd(singledish)|""(none)]
# interferometer antenna position file
# date of observation - not critical unless concating simulations
# hour angle of observation center e.g. -3:00:00, or "transit"
# total time of observation or number of repetitions
# pt source calibrator [experimental]
# add thermal noise: [tsys-atm|tsys-manual|""]
# cross polarization (interferometer only)
# display graphics at each stage to [screen|file|both|none]
# overwrite files starting with $project
# If true the taskname must be started using simobserve(...)```
Configuration Files

- Define telescope array for `simobserve`.

Config Files in CASA Already
ALMA, EVLA, CARMA, SMA, etc.

Example Config File: ALMA Cycle 1 ACA

```
# observatory=ACA
# coordsys=LOC (local tangent plane)
# ACA-9-02
# x y z diam pad#
-47.99531371 -564.8585951 -2.318302577 7. J501
-35.89239576 -569.6206755 -2.318648465 7. J504
-65.31846157 -560.7014943 -2.320087842 7. J505
-63.03702802 -574.7165969 -2.320317857 7. J506
-56.9451361 -560.096901 -2.312796311 7. J507
-49.2177138 -555.3091122 -2.314469317 7. J508
```

```
x y z diameter name
```
Configuration Files

• Pick an intermediate-extent full-ALMA configuration
Observations defined via **setpointings** and **obsmode**

```
# simobserve: mosaic simulation task:
project = 'sim'
skymodel = ''
comlist = ''

setpointings = True
  integration = '10s'
  direction = ['','']
  mapsize = ['ALMA']
  mpttype = ['ALMA']
  pointingspacing = ''

obsmode = 'int'
  antennalist = 'alma.out10.cfg'
  refdate = '2012/05/21'
  hourangle = 'transit'
  totaltime = '7200s'
  caldirection = ''
  calflux = '1Jy'

thermalnoise = 'Sopapilla'
  leakage = 0.0
  graphics = 'both'
  verbose = False
  overwrite = True
  async = False
```

*Observation (Required)*

Integration time, scan length, pointing centers
**setpointings**

- **setpointings** dictates field, integration time, mosaic

```python
setpointings = True
integration = '600s'
direction = ''
mapsiz = ['', ']
maptype = 'ALMA'
pointingspacing = ''
```

# integration (sampling) time
# "J2000 19h00m00 -40d00m00" or "" to center on model
# angular size of map or "" to cover model
# hexagonal, square, etc
# spacing in between pointings or "0.25PB" or "" for 0.5 PB

- **integration** sets data averaging (and field visit) time
HERE AVERAGING 600S (10M) ENSURES A QUICK INITIAL EXECUTION

- **direction** sets field or map center

- **mapsizemaptypepointingspacing** define a mosaic
BY DEFAULT IT WILL COVER THE MODEL, HERE THAT MEANS A 9-POINT MOSAIC
obsmode

- obsmode sets total time, date, observing sequence

```python
obsmode =
    'int'
    'alma.out10.cfg'
    '2012/05/21'
    'transit'
    '7200s'
    ''
    '1Jy'
```

- totaltime sets total observation direction
  HERE 7200s IS A TYPICAL ALMA OBSERVATION DURATION

- Optionally specify the date, LST, and a calibrator sequence.

  go simobserve
  SIMOBSERVE CREATES A MEASUREMENT SET (MS) IN
  projectname/projectname.ms
• **`simobserve` outputs diagnostic plots to project directory**

  **TEXT FILES SHOW THE LOCATION OF POINTING CENTERS (OR `LISTOBS`)**
- Corruption with **thermal noise & toolkit**

```plaintext
# simobserve :: mosaic simulation task:
project = 'sim'
skymodel = ''
complist = ''
setpointings = True
    integration = '10s'
    direction = ''
    mapsize = [',', ']
    maptype = 'ALMA'
    pointingsspacing =
obsmode = 'int'
    antennalist = 'alma.out10.cfg'
    refdate = '2012/05/21'
    hourangle = 'transit'
    totaltime = '7200s'
    caldirection = ''
    calflux = '1Jy'

thermalnoise = 'Sopapilla'
    leakage = 0.0
    graphics = 'both'
    verbose = False
    overwrite = True
    async = False
```

**Corruption (Optional)**
- Thermal noise, phase noise, polarization leakage
Multiple sets of observations

- One can simulate multiple sets of observations with multiple calls to simobserve
  - Simulate combining data from compact and extended arrays
  - Simulate combining data from interferometers and single dish telescopes
- The CLEAN task can take multiple measurement sets to combine interferometric observations
- The FEATHER task can combine single dish and interferometric observations
thermal noise

- Set observing conditions to add random noise to image

- See CASA guides and toolkit for other ways to corrupt data. E.G., PHASE NOISE

- We will make a noisy and a not-noisy version to compare.

  - MAKE SURE TO SHOW THIS, OR NOT SAY IT

```bash
go simobserve
SIMOBSERVE CREATES A MEASUREMENT SET (MS) IN projectname/projectname.ms```

NRAO
Basic Simulation Workflow

In CASA...

1. **Model Sky Distribution** (FITS, image, components)
2. **simobserve**
3. **Simulated Measurement Set** (calibrated u-v data)
4. **simanalyze**
5. **Simulated Image & Analysis Plots Comparing “Observed”/original image**
simanalyze

- Image and analyze simobserve output

CASA <0>: inp simanalyze
--------> inp(simanalyze)
# simanalyze :: image and analyze simulated datasets
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 = ''  # prior image to use in clean e.g. existing single dish 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 or '' to equal model
    niter = 500  # maximum number of iterations (0 for dirty image)
    threshold = '0.1mJy'  # flux level (+units) to stop cleaning
    weighting = 'natural'  # weighting to apply to visibilities
    mask = []  # Cleanbox(es), mask image(s), region(s), or a level
    outertaper = []  # uv-taper on outer baselines in uv-plane
    stokes = 'I'  # Stokes params to image

analyze = False  # (only first 6 selected outputs will be displayed)
geraphics = 'both'  # display graphics at each stage to [screen\file\both\none]
verbose = False  # overwrite files starting with $project
overwrite = True  # If true the taskname must be started using simanalyze(...)
• Grid, invert, and CLEAN the simulated data set.

• Similar but reduced options compared to CLEAN. **Defaults are “smart”, informed by the model.**

• You can also image the simulated observations with CLEAN. **They are a normal CASA measurement set for all purposes.**
Output files can be examined with the CASA viewer. In CASA 3.4 these live in `projectname/projectname.image`.
analyze

- Create diagnostic plots based on `simobserve` and `image`

<table>
<thead>
<tr>
<th>analyze</th>
<th>=</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>showuv</td>
<td>=</td>
<td>True</td>
</tr>
<tr>
<td>showpsf</td>
<td>=</td>
<td>True</td>
</tr>
<tr>
<td>showmodel</td>
<td>=</td>
<td>True</td>
</tr>
<tr>
<td>showconvolved</td>
<td>=</td>
<td>False</td>
</tr>
<tr>
<td>showclean</td>
<td>=</td>
<td>True</td>
</tr>
<tr>
<td>showresidual</td>
<td>=</td>
<td>False</td>
</tr>
<tr>
<td>showdifference</td>
<td>=</td>
<td>True</td>
</tr>
<tr>
<td>showfidelity</td>
<td>=</td>
<td>True</td>
</tr>
</tbody>
</table>

# (only first 6 selected outputs will be displayed)
# display uv coverage
# display synthesized (dirty) beam (ignored in single dish simulation)
# display sky model at original resolution
# display sky model convolved with output beam
# display the synthesized image
# display the clean residual image (ignored in single dish simulation)
# display difference image
# display fidelity

- Pick up to 6 of these.
analyze

• Create diagnostic plots based on `simobserve` and `image`
• Create diagnostic plots based on `simobserve` and `image`
Try It Yourself!

- Simulate one of the suite of model images at http://casaguides.nrao.edu/index.php?title=Sim_Inputs