Obit Development Memo 40
ALMA Continuum Scripts: Outline of Data Reduction and Heuristics

Bill Cotton

July 20, 2015

1 Introduction

1.1 Scope

The scope of the present version of the ALMA continuum scripts is to perform standard calibration and editing of ALMA data and produce continuum, possibly wide-band, images of target sources. Logs, reports and numerous diagnostic plots help evaluate the results of the processing. If default processing parameters are adequate, the scripts will start from ALMA archive ASDM/BDF files and result in FITS images, calibrated data, reports, plots etc. The scripting is also capable of being highly tuned to a particular project and can be rerun in whole or part with user specified parameters.

1.2 Software

The ALMA Obit scripts are:

- Written in python, and
- Use Obit and AIPS tasks to do the data processing, and
- Use AIPS data structures for intermediate data, and
- Write FITS images and (AIPS FITAB format) calibrated data-sets.

AIPS (http://www.aips.nrao.edu/index.shtml) and Obit (http://www.cv.nrao.edu/ bcotton/Obit.html) are installed on all NRAO Linux computers and available for installation via download to non-NRAO computers. A binary distribution is supported for Linux. The ALMA scripts are in the $OBIT/python directory with the template parameter script in $OBIT/share/scripts.

2 Execution

Several steps are needed to execute the ALMA scripts.

2.1 Generate parameter scripts

The processing is guided by values in python parameter scripts. These scripts can be created and initialized by information gleaned from the ALMA archive ALMA Science Data Model (ASDM) files using routine ALMACal.ALMAPrepare (see section 7). This will create one or more parameter files, each of which needs to be processed separately.

Alternatively, the parameter file can be derived manually using the template file $OBIT/share/scripts/ALMATemplateParm.py and making the substitutions described in the file.

2.2 Modify parameter scripts

If the default values in the automatically generated parameter script(s) are not appropriate, they can be changed, see Section 6. The details of each processing step and the parameters used are described in Section 7. Default parameters and control switches can be overridden in the parameter scripts. Additional calibrator model information can be entered as described in section 3. The end of the parameter script contains switches to turn on and off various stages.

2.3 AIPS and Obit setup scripts

A script needs to be created giving the details of the AIPS and Obit installations. This script is described in detail in Section 4.

2.4 Execute scripts

Each of the parameter scripts can be executed from the Unix shell by

```
> ObitTalk ALMAContPipe.py AIPSSetup.py \
     ALMAContParm_myProject_Cfg0\Nch64.py
```

where ALMAContParm_myProject_Cfg0_Nch64.py is the name of your parameter script.

This procedure can start from an archive data set and result in a set of calibrated data, images, reports, logs and various diagnostic plots, see Section 8 for details.

3 Calibrator models

All standard, constant calibrator models are weak and resolved at mm/submm wavelengths but may still be usable at lower frequencies. Some QSOs are monitored and point models may be adequate. Solar system objects are frequently used for flux calibrators; the calibration process will include self calibration which will derive a model, the parameter CalModelFlux can be used to give the total flux density.

The ALMA calibration scripts operate on arrays of calibrator dict structures with the following entries. These allow specifying parameterized models or images with CLEAN components (AIPS or FITS)

- `Source:` Source name as given in the SU table.
- `CalFile:` Calibrator model Cleaned FITS file name
- `CalName:` Calibrator model Cleaned AIPS map name
- `CalClass:` Calibrator model Cleaned AIPS map class
- `CalSeq:` Calibrator model Cleaned AIPS map seq
- `CalDisk:` Calibrator model Cleaned AIPS map disk
- `CalNfield:` Calibrator model No. maps to use for model
- `CalCCVer:` Calibrator model CC file version
- `CalBComp:` Calibrator model First CLEAN comp to use, 1/field
- `CalEComp:` Calibrator model Last CLEAN comp to use, 0=¿all
- `CalCmethod:` Calibrator model Modeling method, 'DFT','GRID',' '
- `CalCmodel:` Calibrator model Model type: 'COMP','IMAG'
- `CalFlux:` Calibrator model Lowest CC component used
- `CalModelSI:` Calibrator Spectral index
- `CalModelFlux:` Parameterized model flux density (Jy)
- `CalModelPos:` Parameterized model Model position offset (asec)
- `CalModelParm:` Parameterized model Model parameters (maj, min, pa, type)

These dicts are created in the parameter script by routine ALMACal.ALMACalModel for the various types of calibrators. ALMACal.ALMAStdModel is then used to fill in the details about calibrators it knows about and can find in the first FITS directory; unfortunately, this is not much for ALMA. Information not known to these scripts may be entered into the calibrator dict structure in the parameter script.

4 AIPS and Obit Setup

These scripts use data in AIPS format and some AIPS tasks; the location of the AIPS data directories and other details as well as the Obit initialization are given in the AIPSSetup.py file. The items that need to be specified are:

- `adirs`
  A list of the AIPS data directories as a tuple, the first element is the URL of the ObitTalkServer or None for local disk. The second element is the directory path.
- `fdirs`
  A list of the FITS data directories as a tuple, the first element is the URL of the ObitTalkServer or None for local disk. The second element is the directory path.
- `user`
  The AIPS user number to be used.
- `AIPS_ROOT`
  The root of the AIPS system directories. An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.
- `AIPS_VERSION`
  The AIPS version. An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.
- `DA00`
  The AIPS DA00 directory (TDD000004; file needed). An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.
- `OBIT_EXEC`
  The root directory of your Obit directories. Python None will default to your system installation on NRAO Linux machines.

- `noScrat`
  A list of AIPS disks to avoid for scratch files, max. 10.

- `nThreads`
  The maximum number of threads allowed to be used. This generally should not be more than the number of cores available.

- `disk`
  The AIPS disk number to use for temporary storage of the data and images.

An example AIPSSetup.py file follows, items which may need to be modified are marked by `<====`.:

```
#   <====  Define AIPS  and FITS disks
adirs = [(None, "/export/data_1/GOLLUM_1"),
         (None, "/export/data_1/GOLLUM_2"),
         (None, "/export/data_1/GOLLUM_3"),
         (None, "/export/data_1/GOLLUM_4"),
         (None, "/export/data_2/GOLLUM_5"),
         (None, "/export/data_2/GOLLUM_6"),
         (None, "/export/data_2/GOLLUM_7"),
         (None, "/export/data_2/GOLLUM_8")]
fdirs = [(None, "/export/users/aips/FITS")]

############################# Initialize OBIT #####################
err     = OErr.OErr()
user    = 104                      # <==== set user number
ObitSys = OSystem.OSystem ("Script", 1, user, 0, [" "], \
                         0, [" "], True, False, err)
OErr.printErrMsg(err, "Error with Obit startup")
# Setup AIPS
AIPS.userno = user
AIPS_ROOT    = "/home/AIPS/"        # <==== set root of AIPS
AIPS_VERSION = "31DEC15/"           # <==== set AIPS version
DA00         = "/home/AIPS/DA00/"   # <==== set AIPS DA00 directory
#  <====  Define OBIT_EXEC for access to Obit Software
OBIT_EXEC    = "/export/data_1/obit/ObitInstall/ObitSystem/Obit/"
# setup environment
ObitTalkUtil.SetEnviron(AIPS_ROOT=AIPS_ROOT, AIPS_VERSION=AIPS_VERSION, \
                        OBIT_EXEC=OBIT_EXEC, DA00=DA00, ARCH="LINUX", \
```

```
                          aipsdirs=adirs, fitsdirs=fdirs)
# List directories
ObitTalkUtil.ListAIPSDirs()
ObitTalkUtil.ListFITSDirs()
noScrat     = []          # <==== AIPS disks to avoid
nThreads    = 6           # <====  Number of threads allowed
disk        = 1           # <====  AIPS disk number
```

5 The Process Overview

The scripted processing uses the following processes. Many of the default processing parameters are frequency dependent and may be overridden and the various steps may be turned on or off.

The general approach to calibration and editing is to first apply editing steps which can be applied to uncalibrated data to remove the most seriously disturbed data. Then an initial pass at calibration is done and a pass at the editing needing calibrated data. Calibration aids in the editing as calibrator data with no detections are effectively removed and calibration with deviant amplitude solutions are also removed.

As part of the calibration process, calibrators are self calibrated which allows resolved sources, especially solar system objects to be used. Diagnostic plots at various stages of the processing are generated. These include plots of calibration results as well as sample spectra.

The calibrated target data are then imaged, using wide-band imaging if appropriate. Images, calibrated data and calibration tables are saved to FITS files and a number of source dependent diagnostic plots are generated. Finally, an HTML report is generated allowing easy examination and access to the various products. A processing log is kept containing most details of the processing.

Following is a summary of the processing. Details and parameters which may be modified are described in a section 7. Each of these steps if controlled by a switch which may be turned on and off.

1. Generation of parameter scripts from ASDM
2. Data converted to AIPS format
3. Hanning if necessary
4. Clear previous calibration

5. Copy initial FG table
6. Flag end channels
7. Apply Special Editing
8. Quack
9. Shadow Flagging
10. Apply online (Tsys) calibration
11. Initial Time domain flagging
12. Initial RMS flagging of calibrators
13. Find reference antenna if not specified
14. Plot raw sample spectra
15. Delay calibration
16. Bandpass calibration
17. Set X/Y gains and initial calibration
18. Self calibrate calibrators to get model
19. Phase calibration using calibrators
20. Amplitude calibration using calibrators
21. Flagging of calibrated data
22. Apply calibration and average data
23. Cross Pol clipping if XY, YX present in data
24. X-Y delay calibration if XY, YX present in data
25. Instrumental polarization calibration if XY, YX present in data
26. Plot final calibrated spectra
27. Image targets
28. Generate source report

29. Save images, calibrated data
30. Contour plots of images
31. source UV diagnostic plots
32. Generate HTML summary
33. Cleanup AIPS directories

Each of these steps if controlled by a switch which may be turned on and off; the following appear at the bottom of the parameter file.

```
# Control, mark items as F to disable
T   = True
F   = False
check                   = parms["check"]      # Only check script, don't execute tasks
debug                   = parms["debug"]      # run tasks debug
parms["doLoadArchive"] = T                    # Load from archive?
parms["doHann"]        = parms["doHann"]      # Apply Hanning?
parms["doClearTab"]    = T                    # Clear cal/edit tables
parms["doCopyFG"]      = T                    # Copy FG 1 to FG 2
parms["doEditList"]    = parms["doEditList"] # Edit using editList?
parms["doQuack"]       = T                    # Quack data?
parms["doShad"]        = parms["doShad"]      # Flag shadowed data?
parms["doOnlineCal"]   = T                    # Apply online calibration
parms["doMedn"]        = T                    # Median editing?
parms["doRMSAvg"]      = T                    # Do RMS/Mean editing for calibrators
parms["doDelayCal"]    = T                    # Group Delay calibration?
parms["doBPCal"]       = T                    # Determine Bandpass calibration
parms["doXYFixGain"]   = T                    # set X/Y gains and initial calibration
parms["doImgCal"]      = T                    # Self calibrate calibrators
parms["doPhaseCal"]    = T                    # Phase calibration
parms["doAmpPhaseCal"] = T                    # Amplitude/phase calibration
parms["doAutoFlag"]    = T                    # Autoflag editing after final calibrati
parms["doCalAvg"]      = T                    # calibrate and average data
parms["doXYDelay"]     = T                    # X/Y Delay calibration
parms["doPolCal"]      = parms["doPolCal"]    # Instrumental polarization calibration
parms["doImage"]       = T                    # Image targets
parms["doSaveImg"]     = T                    # Save results to FITS
parms["doSaveUV"]      = T                    # Save calibrated UV data to FITS
parms["doSaveTab"]     = T                    # Save UV tables to FITS
```

```
parms["doKntrPlots"]   = T                          # Contour plots
parms["doDiagPlots"]   = T                          # Source diagnostic plots
parms["doMetadata"]    = T                          # Generate metadata dictionaries
parms["doHTML"]        = T                          # Generate HTML report
parms["doVOTable"]     = T                          # VOTable report
parms["doCleanup"]     = T                          # Destroy AIPS files

# diagnostics
parms["doSNPlot"]      = T                          # Plot SN tables etc
parms["doBPPlot"]      = T                          # Plot BP tables etc
parms["doReport"]      = T                          # Individual source report
parms["doRawSpecPlot"] = @PLOTSRC@!='None'   # Plot Raw spectrum
parms["doSpecPlot"]    = @PLOTSRC@!='None'   # Plot spectrum at various stages
```

6 Tuning parameters

The ASDM includes intent information for sources and scans which are used by the ALMACal.ALMAPrepare to create initial parameter files; these may not be adequate for all purposes. The following discuss some parameters which may need adjustment. Several of these are marked as

```
parms['some_parameter'] = value   # ******************* Set this ****
```

in the derived parameter files.

1. doCleanup
   The default behavior of the script is to delete the AIPS data files on successful completion. If you want to try several strategies or do further processing on the AIPS data, turn this to False (or F)

2. CalModelFlux
   The "CalModelFlux" member of the calibrator source structure can be used to specify the flux density of a calibrator source, e.g.

   ```
   ACals[0]['CalModelFlux'] = 1.558   # ALMA 2013-12-21 1.558 +/- 0.06
   ```

   ALMA calibrator monitoring results are available at https://almascience.eso.org/sc/. Setting a model may be most useful for ACals, PCals and BPCals. A more general discussion of specifying calibrator information see Sect. 3.

3. IClip
   Absolute clipping level for any calibrator/target is given by IClip=[level_Jy,0.1]. This can be used to flag wild values

4. CAchAvg
   The number of channels to average when calibrating/averaging the data. This is useful for continuum observations or to reduce the spectral resolution of spectroscopic data.

5. CalAvgTime
   The time in minutes to average data when calibrating/averaging the data.

6. plotSource
   This is the name of the source to to be used for diagnostic bandpass plots, see Sect. 7 item 15. This defaults to the first scan on the first bandpass calibrator named.

7. plotTime
   Time range (days) for which to use plotSource.

8. XYGainSource
   This is the name of the source to set the X/Y gains (see Sect. 7 item 18). This defaults to the first scan on the first bandpass calibrator named.

9. XYGainTime
   Time range (days) for which to use XYGainSource.

10. XYDelaySource
    If the data is to be polarization calibrated, this item is the name of the polarized calibrator to be used to derive the cross–polarized delay. This is not specified in the ASDM and if needed must be manually added.

11. XYDelayTime
    If the data is to be polarization calibrated, this item is the time-range (days) of the data on XYDelaySource to be used.

12. PClist
    This is a list of sources to use in the instrumental polarization. The ASDM has a rather restricted view of this and more than one sources can be used (up to 10). See Sect. 7 item 26 for a description of how to specify models per calibrator in PCCalPoln.

7 Script Stage Details

Details of the various processing steps and the parameters are described in the following. Processing parameters are stored in a python dict named parms and may be specified in the parameter script as

```
parms["parameter"] = value
```

Tables in the following give the parameter name, default value and a description.

1. Generation of parameter scripts from ASDM
   This step is performed from ObitTalk to generate the parameter script(s). Multiple, compatible data-sets may be concatenated by giving a list of the base directories of the ASDM/BDFs.

   ```
   >>> import ALMACal
   >>> ASDMRoots = ["/export/myData/12A-999.sb9332941.eb9360588.../",]
   >>> ALMACal.ALMAPrepare(ASDMRoots, err, project="myData")
   ```

   This will parse the ASDMs indicated and generate a parameter script for each configuration/number of channel combinations needed to process all data. Note: this will also include configurations used for calibration purposes only, such as offset pointing, so use Obit task ASDMList to see which configurations have useful data. For each configuration/number of channel, a parameter script with name of the form ALMAContParm_<project>_Cfg<config>_Nch<no channels>.py will be generated.

ASDMRoots		Root directories of ASDM/BDF data
project		Project name, 12 or fewer characters, used as AIPS file name
template	ALMATemplateParm.py	name of the parameter template file
parmFile		Name of desired parameter file, generated if not given

2. General control parameters
   These parameters control the naming of files, whether UV data is used in compressed form and script debugging control.

project	??	Project name, 12 or fewer characters, used as AIPS file name
session	??	session code, generated from configuration and no. channels
band	??	Observing band code, derived from ASDM frequencies
Compress	False	Use compressed (scaled 16 bit integers) UV data?
check	False	Only check script, don't execute tasks
debug	False	run tasks debug

3. Data converted to AIPS format
   The bulk of the processing uses AIPS format UV data and images. The ASDM/BDF data is first converted to an AIPS data file using Obit/BDFIn. The details of the AIPS configuration are given in the AIPSSetup.py file provided to the processing script.

doLoadArchive	True	Load AIPS data from archive?
archRoots	??	User specified list of ASDMs/BDFs
selConfig	??	Frequency configuration, generated from ASDM
seq	1	AIPS sequence number to use
selBand	??	Data band-code, derived from ASDM
selChan	??	Number of spectral channels, derived from ASDM
selNIF	??	Number of spectral windows (IFs), derived from ASDM
calInt	??	Calibration table interval (min), ALMA config. dependent

4. Hanning
   Very strong, narrow signals will produce "Gibbs" ringing due to the truncation of the lag spectra. Hanning smoothing can be used to suppress this effect.

doHann	False	Apply Hanning smoothing?
doDescm	True	If True, drop every other channel after smoothing

5. Clear previous calibration
   If the script is restarted it is frequently desirable that previous attempts at calibration and editing be removed.

doClearTab	True	Clear cal/edit tables?
doClearGain	True	Clear SN and CL tables > 1?
doClearFlag	True	Clear FG tables > 1?
doClearBP	True	Clear BP tables?

6. Copy initial FG table
   To allow restarting of the flagging, the on-line flags which are in FG table 1 are copied to table 2 and new flags added there. This should be turned off if the script is restarted except at the beginning. Note, this only affects the ASDM flags, binary flags are applied directly to the data weights.

doCopyFG	True	Copy FG 1 to FG 2?

7. Flag end channels
   The first and last few channels in each IF (SW) are flagged if FG table 2. This can be turned off by setting BChDrop and EChDrop to 0.

BChDrop	min (32,max(2, (nchan/64))	# of chan. to flag at the beginning
EChDrop	min (32,max(2, (nchan/64))	# of chan. to flag at the end

8. Apply Special Editing
   If some data are known to be bad, e.g. no receiver, then this information can be passed to the script. if doEditList is True, each entry is a python dict with the following:

   - `timer:` The affected time range as a pair of strings of the form day/hour:min:sec.
   - `Ant:`A baseline specification as a pair of antenna numbers, if the second is zero, then all baselines to the first antenna number is flagged. If the first is also zero, then all antennas are flagged
   - `IFs:` Range (1-rel) of IFs (spectral windows) to flag. If the second is zero then all IFs higher than the first are flagged.
   - `Chans:` Range (1-rel) of channels to flag. If the second is zero then all channels higher than the first are flagged.
   - `Stokes:` Array of flags, 1=>flag, 0 => not flag; in order XX, YY, XY, YX.
   - `"Reason:` Up to 24 characters giving reason.

   an example:

```
parms["doEditList"]  = True        # Edit using editList?
parms["editList"] = [
    {"timer":("0/00:00:0.0","5/00:00:0.0"),"Ant":[1,0],
     "IFs":[1,0],"Chans":[1,0],  "Stokes":'1111',"Reason":"No Rcvr"}
]
```

doEditList	False	Edit using editList?
editFG	2	Table to apply edit list to
editList	[ ]	List of data to flag

9. Quack
   Data at the beginning and end of each scan can be flagged using Obit/Quack.

doQuack	True	Quack data?
quackBegDrop	0.05	Time to drop from start of each scan (min)
quackEndDrop	0.0	Time to drop from end of each scan (min)
quackReason	"Quack"	Reason string

10. Shadow Flagging
    In the more compact ALMA configurations, some antennas may shadow others at times. The affected data may be flagged using task UVFlag.

doShad	False	Do shadow flagging?
shadBl	12.0	Minimum shadowing baseline (m)

11. Apply online (Tsys) calibration
    Online Tsys measurements are extracted from the ASDM when the data is filled into AIPS format and written in the form of an AIPS SN version 1 table. This table can be applied to the initial Calibration (AIPS CL) table to produce AIPS CL table version 2.

doOnlineCal	True	Apply TSys calibration?

12. Initial Time domain flagging
    Obit task MednFlag can be used to flag data by amplitudes deviant from a running median by more than a specified amount. This is performed independently on each data stream (baseline, channel, IF, poln). At this point the data are uncalibrated.

doMedn	True	Median editing?
mednSigma	10.0	Sigma clipping level
mednTimeWind	1.0	Window width (min) for median flagging
mednAvgTime	10.0/60.	Averaging time (min)
mednAvgFreq	0	1=>avg mednChAvg chans, 2=>avg all chan, 3=>avg chan and IFs
mednChAvg	1	Number of channels to average

13. Initial RMS flagging of calibrators
    Calibrators are expected to be simple and have significant SNR so can be edited by having an RMS/average amplitude of less than some amount. Discrepant calibrator data can be flagged in this step using Obit task AutoFlag.

doRMSAvg	True	Edit calibrators by RMS/Avg?
RMSAvg	3.0	Max RMS/Avg for time domain RMS filtering
RMSTimeAvg	1.0	Time averaging (min)

14. Reference antenna
    The choice of reference antenna is of some importance but nothing in the ASDM helps in this choice. If the reference antenna (parms["refAnt"]) is unspecified (0), this step runs Obit task Calib on the bandpass calibrator(s) (assumed to give good fringes). The resultant SN table is then examined for the antenna with the maximal amount of valid solutions and with the highest average SNR; this antenna is used as the reference antenna. Values found from a previous run will be stored in a python pickle file.

refAnt		Reference antenna, if <= 0 then determine
BPCals		Determined from the ASDM
bpsolint1		Bandpass first solution interval,

15. Plot Raw spectra
    At this point, plots of sample spectra can be made to display problematic data; baselines to refAnt for the specified source and timerange are plotted for the parallel correlations.

doRawSpecPlot	True	Plot diagnostic raw spectra?
plotSource		Default is first bandpass calibrator
plotTime		List of start and end time in days.
		Default is first bandpass calibrator scan
refAnt		Reference ant., baselines to refAnt are plotted

16. Delay calibration
    Parallel hand group delays are solved for using the list of calibrator models in DCals. Obit task Calib solves for the delays which are then smoothed and applied to all sources in a new CL table using Obit task CLCal. Solutions can be plotted as well as sample spectra applying the delay calibration. For ALMA data the delay residuals are generally very small.

doDelayCal	True	Determine/apply delays?
DCals		The list of delay calibrators are determined from the ASDM, all amplitude, phase and bandpass calibrator. The list of models is determined from the parameter script using standard calibrator models.
delayBChan		first channel to use in delay solutions max(2, 0.05*nchan)
delayEChan		highest channel to use in delay solutions min(nchan-2, nchan-0.05*nchan
solInt		Solution interval (min), config. dependent A:2 sec, B: 5 sec, C:10 sec, D:15 sec.
refAnts	[refAnt]	Delay reference ant., baselines to refAnt are plotted
doTwo	True	Use two baseline combinations in delay cal
delayZeroPhs	True	Zero phase in Delay solutions?
doSNPlot	True	Plot calibration solutions?
doSpecPlot	True	Plot diagnostic calibrated spectra?
plotSource		Source to plot spectra.
plotTime		List of start and end time in days.

17. Bandpass calibration
    Bandpass calibration uses Obit task BPass and calibrator model list BPCals. BPass does a two pass calibration, the first doing a phase only calibration to straighten out the phases followed by a longer amplitude and phase calibration using blocks of channels. The resultant solutions are then combined into a BP table

doBPCal	True	Determine/apply bandpass calibration?
BPCals		The list of bandpass calibrators is determined from the ASDM, The list of models is determined from the parameter script using standard calibrator models.
bpsolint1		BPass phase correction solution
bpsolint2	10.0	BPass bandpass solution interval (min)
bpsolMode	'A&P'	Bandpass type 'A&P', 'P', 'P!A'
bpBChan1	1	Low freq. channel, initial cal
bpEChan1	0	Highest freq channel, initial cal, 0=>all
bpBChan2	1	Low freq. channel for BP cal
bpEChan2	0	Highest freq channel for BP cal, 0=>all
bpChWid2	3	Number of channels in running mean BP soln
bpDoCenter1	None	Fraction of channels in 1st, overrides bpBChan1, bpEChan1
bpUVRange	[0.0,0.0]	UV range for bandpass cal zeroes=> all
refAnt		BP reference ant., baselines to refAnt are plotted
doSpecPlot	True	Plot diagnostic calibrated spectra?
plotSource		Source to plot spectra.
plotTime		List of start and end time in days.

18. Fix X/Y gain ratio and initial calibration
    ALMA uses linearly polarized feeds and even parallel correlations respond to a combination of Stokes I, Q and U. This is a combination of I and plus or minus a function of Q and U for the XX and YY correlations. As the calibrators, generally QSOs, have significant linear polarization, gains derived from the parallel polarized will be in error and differently in error for the XX and YY correlations. The average of XX and YY will, to first order, give Stokes I but if polarimetric calibration is needed, this effect must be corrected. This step is to fix the X/Y gain ratio to the value for a specific calibrator for a specific time. If the calibrator is unpolarized, the gain ratio will be 1.0 at all times and all data can be used. For a polarized source, the time range should be restricted to a range over which the parallactic angle has a small change. Calibration after this step prior to instrumental polarization calibration will average XX and YY and will not modify the X/Y gain ratio. The instrumental polarization calibration can include determination of the X/Y gain ratio.

doXYFixGain	True	Fix X/Y gain ratio
XYGainSource		Calibrator source to be used
XYGainTime		start and end times for calibration in days
refAnt		Reference anttenna

19. Self calibrate calibrators
    Calibrators, especially solar system objects, may be resolved. Self calibration is used to derive a source model to be used is subsequent amplitude and phase calibration.

doImgCal	True	Image/self calibrate calibrators
CalFOV	??	Field of view (deg) to image
		Frequency and maximum baseline dependent.
outCClass	'ISfCal'	Image AIPS class
refAnt		Reference antenna.
maxPSCLoop	1	Max. number of phase self cal loops
minFluxPSC	0.05	Min flux density peak for phase self cal
solPInt	??	Phase self cal solution interval (min), ν dependent
maxASCLoop	1	Max. number of Amp+phase self cal loops
minFluxASC	0.5	Min flux density peak for amp+phase self cal
solAInt	3	amp+phase self cal solution interval (min),
avgPol	True	Average poln in self cal?
avgIF	False	Average IF in self cal?
minSNR	4.0	Minimum Allowed SNR in self cal

20. Phase Calibration
An initial, short term phase calibration is done on the calibrators to remove the phase variations. This uses the calibrator models derived from the self calibration stage. Both "phase" and "amplitude" calibrators are used. Solutions are applied to the previous CL table to create a new CL table.

doPhaseCal	True	Do phase calibration?
PCals		The list of phase calibrator models
ACals		The list of amplitude calibrator models
refAnt		Reference antenna
solPInt		Solution interval (min), ν dependent:
ampBChan	max(2, 0.05*nchan)	first channel to use in solutions
ampEChan	min(nchan-2, nchan-0.05*nchan)	highest channel to use in solutions
ampScalar	False	Ampscalar solutions?

21. Amp & phase Calibration
Standard flux density calibrators have their flux densities entered into the SU table using Obit task SetJy, other calibrators have their flux density entries set to the value of CalModelFlux, if given, else 1.0. All the amplitude and phase calibrators have Obit/Calib run using their models and doing amplitude and phase solutions. Solutions are then median window smoothed using Obit/SNSmo to time solSmo clipping

really wild points. Obit task GetJy then solves for the flux densities for non flux density calibrators and adjusts the SU and SN tables. If doAmpEdit is True, solutions in each IF (spectral window) more than ampSigma from the mean are flagged both in the SN table and in FG table ampEditFG. Finally solutions are applied to the previous CL table to create a new CL table. Solution plots are written into file parms["project"]+"_"+parms["session"]+"_"+parms["band"]+"APCal.ps".

doAmpPhaseCal	True	Do amplitude and phase calibration?
ACals		The list of amplitude calibrators are determined from the ASDM,
		The list of models is determined from the parameter script
		using standard calibrator models.
PCals		The list of phase calibrators are determined from the ASDM
refAnt		Reference antenna
solInt		Solution interval (min), config. dependent:
ampBChan		first channel to use in A&P solutions
		max(2, 0.05*nchan)
ampEChan		highest channel to use in A&P solutions
		min(nchan-2, nchan-0.05*nchan)
solSmo	0.0	Smoothing interval for Amps (min)
ampScalar	False	Ampscalar solutions?
doAmpEdit	True	Edit/flag on the basis of amplitude solutions
ampSigma	20.0	Multiple of median RMS about median gain to clip/flag
ampEditFG	2	FG table for editing
doSNPlot	True	Plot calibration solutions?

22. Flagging of calibrated data
    Calibrated data are then edited using Obit/AutoFlag. Data with amplitudes outside of a given range are flagged and data overly discrepant from a running median in frequency is flagged.

doAutoFlag	True	Autoflag editing after first pass calibration?
IClip	[200.,0.1]	AutoFlag Stokes I clipping
minAmp	1.0e-5	Minimum allowable amplitude
timeAvg	0.33	AutoFlag time averaging in min.
doAFFD	False	do AutoFlag frequency domain flag
FDmaxAmp	IClip[0]	Maximum average amplitude (Jy)
FDmaxV	VClip[0]	Maximum average VPol amp (Jy)
FDwidMW	31	Width of the median window
FDmaxRMS	[5.0,.1]	FDmaxRMS
FDmaxRes	6.0	Max. residual flux in sigma
FDmaxResBL	6.0	Max. baseline residual
FDbaseSel	[0,0,0,0]	Channels for baseline fit

23. Calibrate and average data
The calibration and editing files are then applied with possible averaging in time and/or frequency. This uses Obit/Splat which writes a multi-source file.

doCalAvg	True	Calibrate and average?
avgClass	"UVAv?"	AIPS class of calibrated/averaged UV data
seq	1	AIPS sequence
CalAvgTime		Time for averaging calibrated UV data (min)
avgFreq	0	0 => no averaging, 1 =>avg chAvg chans,
		2 =>avg all, 3 =>avg chan and IFs
chAvg	1	Number of channels to average
CABChan	1	First channel to copy
CAEChan	0	Highest channel to copy, 0 => all higher than CABChan
CABIF	1	First IF to copy
CAEIF	0	Highest IF to copy, 0 => all higher than CABIF
Compress	False	Write compressed UV data?

24. Cross Pol clipping
if XClip[0] is not None, cross polarized data with amplitudes > XClip[0] are flagged.

XClip	[5.0,0.05]	AutoFlag cross-pol clipping, None=> no flagging

25. X-Y delay calibration
If the data contain XY and YX correlations and polarimetric results

are desired, the delay difference between the X and Y systems is needed. This step determines the x-y delay from a single calibrator. These calibrators are not specified in the ASDM and will need to be inserted manually as parms['XYDelaySource'] and parms['XYDelayTime']. If the first source in this list is not None, and the data contains the XY and YX correlations, the x-Y delay calibration is performed.

doXYDelay	??	Do X-Y delay calibration?
XYDelaySource	??	Calibrator, skip if None
XYDelayTime	??	timerange for calibration
xyBChan	1	First (1-rel) channel number
xyEChan	0	Highest channel number. 0 => high in data.
xyUVRange	[0.0,0.0]	Range of baseline used in kilo wavelengths, zeros=all
xyDoCal	2	Apply calibration table? positive=>calibrate
xygainUse	0	CL/SN table to apply, 0 =>highest
xynumIFs	1	Number of IFs per solution
xyflagVer	2	FG table version to apply
refAnt		Reference antenna

26. Instrumental polarization calibration
Determine instrumental polarization from a list of calibrators if the data contains XY and YX correlations. The function ALMACal.ALMAPrepare sets this list to those with an intent "CALIBRATE_POLARIZATION". Parameter parms["doPolCal"] is set True if PCInsCals is not empty and the calibration performed if the data contains the XY and YX correlations. Calibration uses Obit task PCal which determines antenna and source polarization parameters on blocks of channels in a running window. The antenna parameters are the ellipticity and orientation of the feed; see Obit Development Memo 32 (in preparation) for details. Model parameters can be specified in PCCalPoln as a list of tuples corresponding to sources in PCInsCals, each tuple is:

    (a) PPol
    Fractional poln, <0 => fit
    (b) EVPA
    Polarization angle at reference frequency in deg
    (c) RM
    Rotation measure (rad/m**2)

doPolCal	??	Determine instrumental polarization?
PCInsCals	??	Instrumental poln cals, name or list of names
PCCalPoln	None	Models for PCInsCals, None=>fit all
PCSolInt	2.0	Instrumental solution interval (min), 0 => scan average(?)
PCRefAnt	0	Reference antenna, defaults to refAnt
PCSolType	" "	Solution type, "LM " (better), " " (faster)
PCChInc	5	Channel step in spectrum
PCChWid	5	Number of channels to average
doFitOri	False	Fit (linear feed) orientations?
doFitXY	True	Fit X-Y gain phase
doPol	False	Apply polarization cal in subsequent calibration?
PDVer	1	Apply PD table in subsequent polarization cal?

27. Plot final calibrated spectra
    At this point, plots of sample spectra can be made to display calibrated data.

doSpecPlot	True	Plot diagnostic spectra?
plotSource		Source to plot
plotTime		List of start and end time in days.
refAnt		Reference ant., baselines to refAnt are plotted

28. Image targets
    All targets are imaged and deconvolved using Obit/Imager or Obit/MFImage if wideband imaging needed (fractional spanned bandwidth $\geq$ MB-maxFBW). Phase only and amp and phase self calibration may be applied if sources exceed given thresholds. If wideband imaging is used, then the resultant images are cubes having planes:

    (a) Total intensity at reference frequency.
    (b) Spectral index at reference frequency
    (c) any higher order planes
    (d) One plane for each of the coarse frequency samples.

doImage	True	Image targets?
targets	[?]	Target list set from ASDM, empty=>all
seq	1	AIPS sequence for images
doPol	True	Apply polarization cal?
PDVer	1	Apply PD table?
outIclass	"IClean"	Image AIPS class
Stokes	"I"	Stokes to image
Robust	0.0	Weighting robust parameter
FOV		Field of view radius in deg, average ν dependent:
Niter	500	Max number of CLEAN iterations
minFlux	0.0	Minimum CLEAN flux density (Jy)
minSNR	4.0	Minimum Allowed SNR in self cal
maxPSCLoop	1	Max. number of phase self cal loops
minFluxPSC	0.05	Min flux density peak for phase self cal
solPInt		Phase self cal solution interval (min), ν dependent
		$\nu < 1$ GHz, L,C,X,Ku,K,Ka: 0.25, Q band:0.10
solPMode	"P"	Solution mode for phase self cal
solPType"	"L1"	Solution type for phase self cal
maxASCLoop	1	Max. number of Amp+phase self cal loops
minFluxASC	0.5	Min flux density peak for amp+phase self cal
solAInt	3	amp+phase self cal solution interval (min), ν dependent
solAMode	"A&P"	Amp and phase self cal
solAType	"L1"	Solution type for Amp and phase self cal
avgPol	True	Average poln in self cal?
avgIF	False	Average IF in self cal?
nTaper	0	Number of additional imaging multi-resolution tapers
Tapers	[20.0,0.0]	List of tapers in pixels
do3D	False	Make ref. pixel tangent to celest. sphere for each facet
noNeg	False	Allow negative components in self cal model?
BLFact	1.01	Baseline dependent time averaging for > 1.0?
BLchAvg	True	Baseline dependent frequency averaging?
doMB	??	Set in parameter script depending on spanned bandwidth
MBnorder	1	Order of wideband imaging
MBmaxFBW	0.05	max. MB fractional bandwidth
CleanRad	None	CLEAN radius about center or None=autoWin

29. Generate report

doReport	True	Generate source report?
targets	[?]	Target list set from ASDM, empty=>all
seq	1	AIPS sequence for images
outIclass	”IClean”	Image AIPS class
Stokes	”I”	Stokes imaged

30. Save images, calibrated data
Images and calibrated/averaged data and calibration tables are written to FITS files. File names begin with
parms[”project”]+parms[”session”]+parms[”band”] followed by <source_name>+<Stokes>+”Clean.fits” for images and ”Cal.uvtab” for calibrated data and ”CalTab.uvtab” for calibration tables from the original data.

doSaveImg	True	Save target images to FITS?
targets	[?]	Target list set from ASDM, empty=>all
doSaveUV	True	Save calibrated UV data for AIPS/FITAB format?
doSaveTab	True	Save calibration tables for AIPS/FITAB format?

31. Contour plots of images
Contour plots are generated for target images. Plot names are
parms[”project”]+”_”+parms[”session”]+”_”+parms[”band”] followed by the source name and ”.cntr.ps” which are also converted to jpeg with the suffix ”jpg”.

doKntrPlots	True	Generate contour plots?
targets	[?]	Target list set from ASDM, empty=>all

32. UV diagnostic plots
Plots of amplitude vs. baseline length, real vs. imaginary and UV coverage are generated. Plot names are
parms[”project”]+”_”+parms[”session”]+”_”+parms[”band”] followed by the source name and ”.amp.ps”, ”.ri.ps”, or ”.uv.ps” which are also converted to jpeg with the suffix ”jpg”.

doDiagPlots	True	Make UV diagnostic plots per source?
targets	[?]	Target list set from ASDM, empty=>all

33. Generate HTML Summary
    Generate an HTML page with source statistics and links to the various plots.

doHTML	True	Generate HTML reports?

34. Cleanup
    AIPS data and image files are zapped.

doCleanup	True	Clean out AIPS directories?

8 The Products

- Calibrated (u,v) dataset with calibration and flagging tables in AIPS FITAB format – Tables from initial data and averaged visibilities per input dataset. These files are
  parms["project"]+parms["session"]+parms["band"]+"Cal.uvtab" and
  parms["project"]+parms["session"]+parms["band"]+"CalTab.uvtab".

- FITS Images – for each target object in files
  parms["project"]+"_"+parms["session"]+"_"+parms["band"]+
  source_name+".IClean.fits".
  If wideband imaging is used, then the resultant images are cubes having planes:

  1. Total intensity at reference frequency.
  2. Spectral index at reference frequency
  3. any higher order planes
  4. One plane for each of the coarse frequency samples.

- Diagnostic plots – calibration and several per source. The project plots have prefix parms["project"]+"_"+parms["session"]+"_"+parms["band"] and are

  – `RawSpec.ps:` AIPS/POSSM plots of sample spectra with initial editing but no calibration applied.
  – `DelaySpec.ps:` AIPS/POSSM plots of sample spectra with initial editing and delay calibration applied. One set per pass through the calibration.

- `BPSpec.ps:` AIPS/POSSM plots of sample spectra with initial editing and delay and bandpass calibration applied. One set per pass through the calibration.
- `Spec.ps:` AIPS/POSSM plots of sample spectra with final editing and calibration applied.
- `XYSpec2.ps:` AIPS/POSSM plots of sample XY and YX spectra with final editing and calibration applied.
- `DelayCal.ps:` AIPS/SNPLT plots of delay calibration.
- `APCal.ps:` AIPS/SNPLT plots of amplitude and phase calibration.

The source plots have prefix parms["project"]+"_"+parms["session"]+"_"+parms["band"] and are

- `source_name.cntr.jpg:` Source image contour plot as jpeg
- `source_name.cntr.ps:` Source image contour plot as postscript
- `source_name.amp.jpg:` Source amp. vs baseline plot as jpeg
- `source_name.amp.pdf:` Source amp. vs baseline plot as pdf
- `source_name.amp.ps:` Source amp. vs baseline plot as postscript
- `source_name.ri.jpg:` Source real vs imaginary plot as jpeg
- `source_name.ri.pdf:` Source real vs imaginary plot as pdf
- `source_name.ri.ps:` Source real vs imaginary plot as postscript
- `source_name.uv.jpg:` Source uv coverage plot as jpeg
- `source_name.uv.pdf:` Source uv coverage plot as pdf
- `source_name.uv.ps:` Source uv coverage plot as postscript

- Reports and logs created during the process
The logfile is
parms["project"]+"_"+parms["session"]+"_"+parms["band"]+".log", and
the HTML report is
parms["project"]+"_"+parms["session"]+"_"+parms["band"]+".report.html".

The file set comprising all files and the meta-data are stored in a single directory.