VLA Data Reduction: *Standard Calibration*

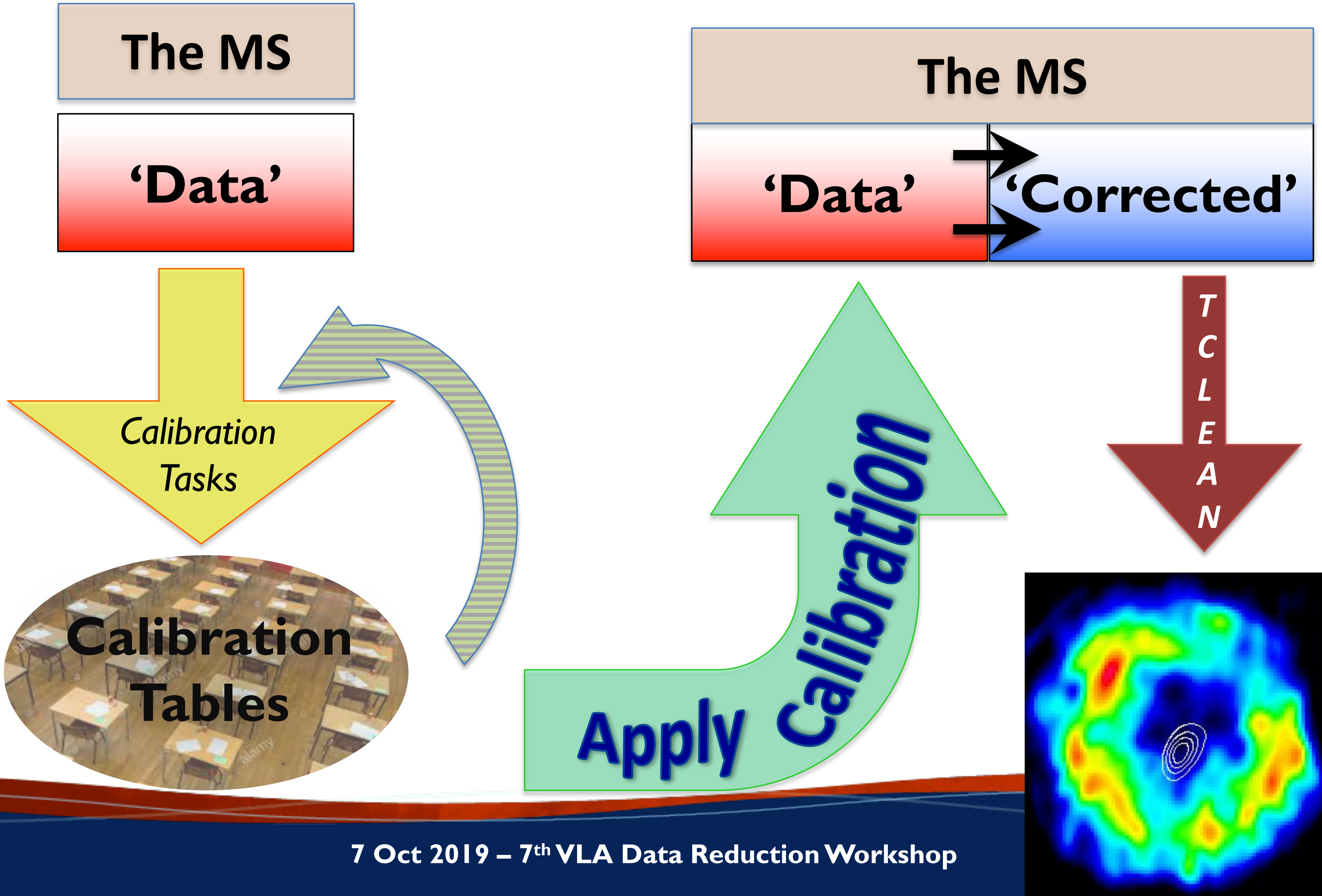
Amy Kimball (NRAO)

The measurement set (MS) structure

'Data' column	'Corrected' Column	'Model' Column (optional)
Raw Data	Calibrated Data	FT of source model

- A raw MS starts with only the 'Data' column.
- The other two columns can be created by various means.
- The creation of the other two columns → MS *triples* in size.
- The 'Model' Column is optional.
  - If not created → MS *doubles* in size.
  - "Virtual" models can be "attached" to the MS, FT-ed and used when needed (replacing the need for the 'Model' column).

Calibration & Imaging Flow



Start CASA in a terminal

INTERACTIVE

This is an interactive presentation.

Run the following in your own terminal.

Raise your hand anytime you need assistance.

```
# In a terminal window
cd ~/data/DRW/Lectures/L3_Standard_Calibration/
casa -r 5.4.2-5
```

Calibration

- Correcting antenna positions
- Gain Curves (high-freq)
- Opacity (high-) and Ionospheric (low-freq) corrections
- Re-quantizer gain calibration (mostly 3-bit data)

} *A priori* calibration

- Setting the flux density scale
- Delay calibration
- Initial phase-only calibration (high-freq)
- Bandpass calibration
- Complex gain calibration
- (Polarization Calibration)
- Setting the flux density scales of the complex gain calibrators

gencal: CASA task for various types of corrections

‘amp’ = amplitude correction

‘ph’ = phase correction

‘sbd’ = single-band delay

‘mbd’ = multi-band delay

‘antpos’ = ITRF antenna position corrections

‘antposvla’ = VLA-centric antenna pos. corrections

‘swpow’ = EVLA switched-power gains

‘rq’ = EVLA requantizer gains

‘swp/rq’ = EVLA switched power gains/req. gains

‘opac’ = Tropospheric opacity

‘gc’ = Gain curve (zenith-angle-dependent gain)

‘eff’ = Antenna efficiency (sqrt(K/Jy))

‘gceff’ = Gain curve and efficiency

‘tecim’ = Total electron content for ionospheric corrections

Antenna Positions: *gencal*

- Correct baselines after antenna moves
  - operator's log reports recent antenna moves
- Use the task *gencal* to produce a calibration table that will include the antenna position corrections
  - (check whether table was needed/created)

- Baseline correction related information is at: http://www.vla.nrao.edu/astro/archive/baselines/

Antenna position corrections

- CASA task *gencal*

```
# In CASA
default gencal
inp
vis = 'standard_calibration.ms'
caltable = 'antpos.cal'
caltype = 'antpos'
inp
go
```

Antenna position corrections reported in the casalogger:

offsets for antenna ea02 : -0.00060 0.00220 -0.00130
offsets for antenna ea04 : 0.00150 0.00190 -0.00150
offsets for antenna ea06 : 0.00120 0.00190 -0.00140
offsets for antenna ea13 : 0.00110 0.00120 -0.00140
offsets for antenna ea16 : 0.00110 0.00120 -0.00180
offsets for antenna ea25 : -0.00200 0.00000 -0.00120

INTERACTIVE

Gain Curves: *gencal*

- Large antennas have a forward gain that changes with elevation.
- Gain curves describe how each antenna behaves as a function of elevation, for each receiver band.
- The polynomial coefficients for the VLA are available directly from the CASA data repository.
- Important for *higher frequencies*.
- The VLA pipeline always performs this step.
- In *gencal*, set:

```
caltype  = 'gc'
caltable = 'gaincurve.cal'
```

Opacity Corrections (HF): *plotweather*

- Atmospheric optical depth, important for high frequencies (>15 GHz)
- CASA task *plotweather* uses weather statistics and/or seasonal models to estimate opacities and make weather plots

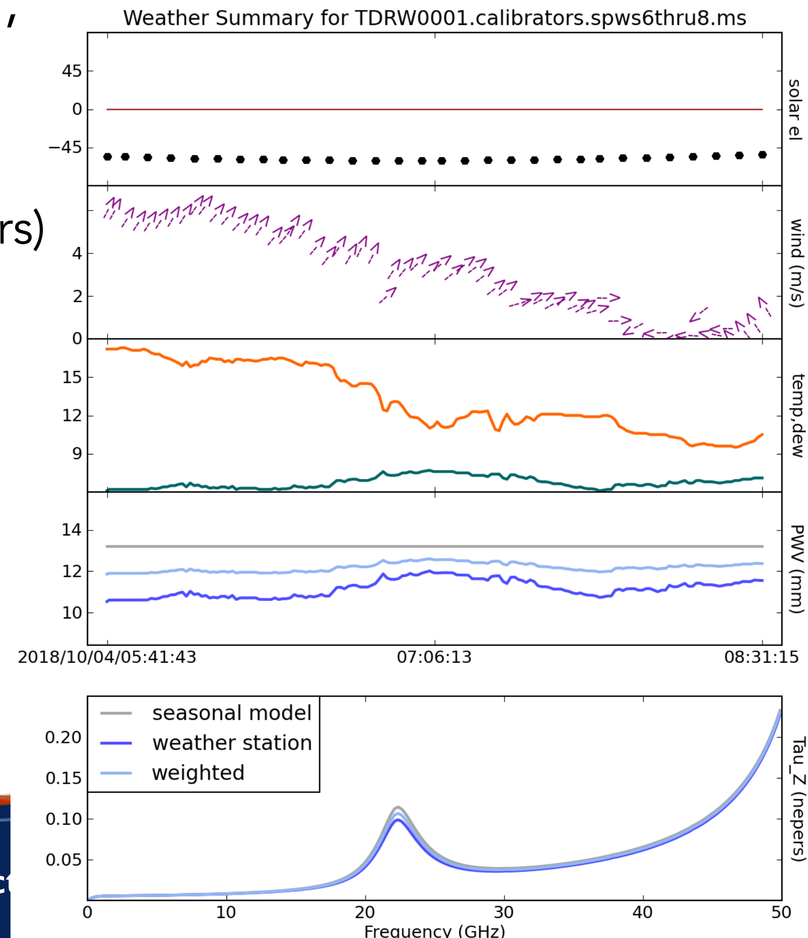
tau_val = *plotweather*(vis='<ms name>', doPlot=True, plotName='weather.png')

- Gives one value per spw:

SPW	Frequency (GHz)	Zenith opacity (nepers)
0	3.000	0.006
1	3.128	0.006
2	3.256	0.006

- Apply to data with *gencal* task:

caltype = 'opac'
caltable = 'opacity.cal'
parameter = tau_val
spw = '*<match to tau_val spws>*'



The Ionosphere: *Total Electron Content (TEC)*

Free electrons in the atmosphere cause a dispersive delay (phase errors). Effect goes as ν^{-2} but depends on ever-changing atmosphere:

- introduces Faraday rotation
- changes measured source position

TEC corrections are:

- Important for VLA low frequencies *(P, L, S bands; C and X if active Sun)*
- Important for large arrays (*baselines* $\gtrsim$ *5 km; VLA's A and B config*)
- Important for polarimetry
- Under commissioning

The VLA pipeline does* NOT *perform TEC corrections.

INTERACTIVE

Ionosphere correction (*Total Electron Content*)

- CASA “recipe” and CASA task *gencal*

`tec_maps` recipe retrieves TEC info from a NASA database.*

```
# In CASA
# import the TEC image
from recipes import tec_maps
tec_image, tec_rms_image, tec_graph = tec_maps.create(
    vis='standard_calibration.ms', doplot=True)

# run gencal
tget gencal
inp
caltype = 'tecim'
caltable = 'tecim.cal'
infile = tec_image
inp
go
```

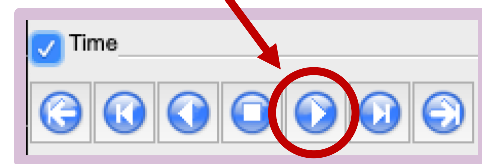
* https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/atmospheric_products.html

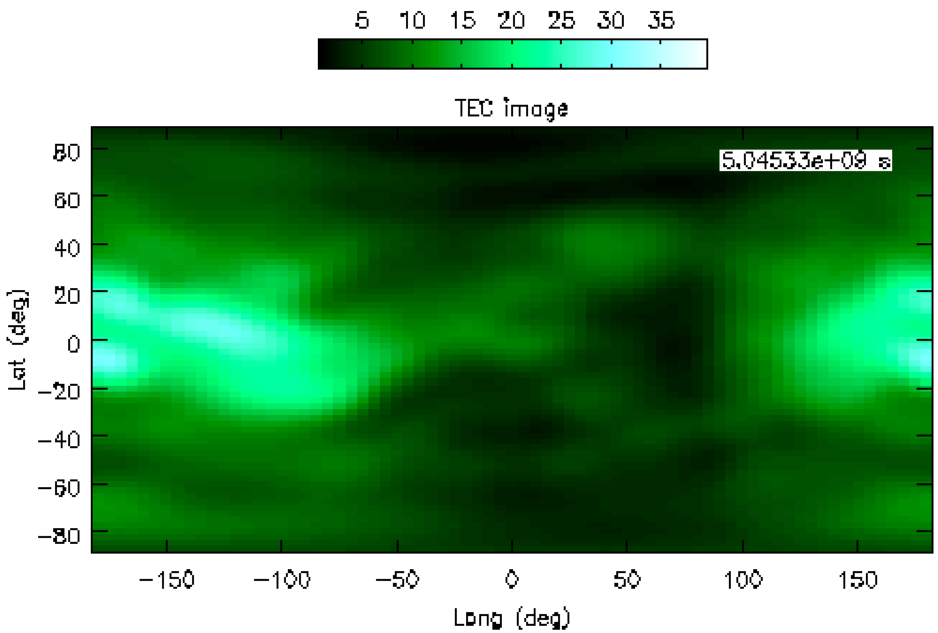
TEC image and rms image for this dataset

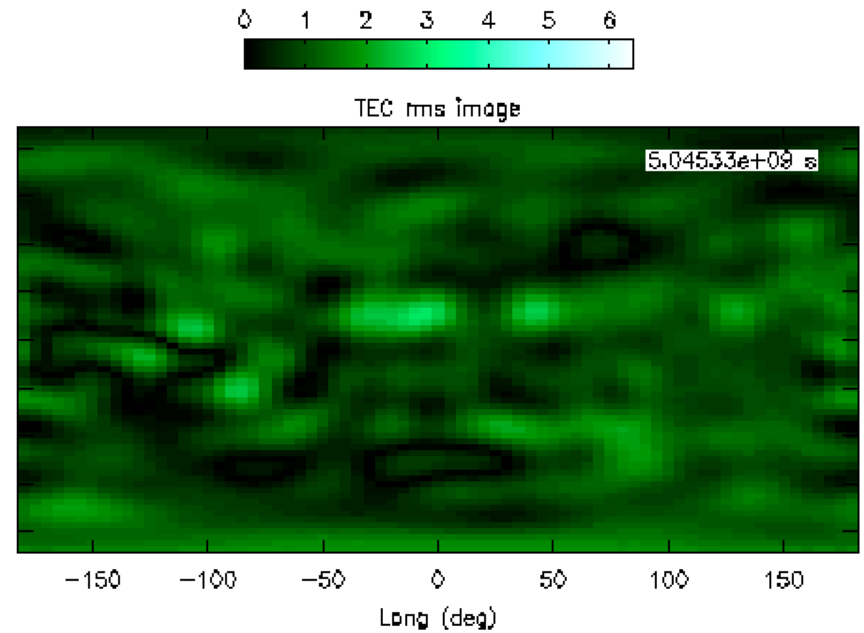
```
# in CASA
!casaviewer  standard_calibration.ms.IGS_TEC.im &
```

TEC is time dependent:

Menu bar → View → Animators







Requantizer gains: *gencal*

- Optimizes the digital power within each spectral window.
- Required for *3-bit data*.
- Strongly recommended for *all P-band data*.
- In *gencal*, set:

```
caltype  = 'rq'
caltable = 'requant_gains.cal'
```

Calibration

- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

} *A priori* calibration

- Setting the flux density scale
- Delay calibration
- Initial phase-only calibration (high-freq)
- Bandpass calibration
- Complex gain calibration
- (Polarization Calibration)
- Setting the flux density scales of the complex gain calibrators

Calibration: setting the flux density scale

- CASA task *setjy* calculates the absolute flux density as a function of frequency (and time):
  - for standard flux density calibrators (e.g., Perley-Butler 2017)
  - for Solar System objects (e.g., Butler-JPL-Horizons 2012)
- If provided, attaches a model record to the MS

```
field                  =  '<field name or #>'
standard               = 'Perley-Butler 2017'
     model             = '<source/band model name>'
     listmodels        =       False
usescratch             =       False
```

INTERACTIVE

Identifying available flux density models

- CASA task *setjy*

```
# In CASA
default setjy
standard = 'Perley-Butler 2017'  # default; other models available
listmodels = True
inp
go
```

`listmodels`

- For `True`, instead of calculating flux density, CASA will list the available primary calibrator models (3C138, 3C147, 3C286, 3C48; L, S, C, X, U, K, A, Q bands).

Note: for P-band models use `standard = 'Scaife-Heald 2012'`:

- 3C48, 3C147, 3C196, 3C286, 3C295, 3C380
- coefficients, *NOT* images

Setting the flux density scale

- CASA task *setjy*

```
# In CASA
result = setjy(vis='standard_calibration.ms',
         field='0', model='3C48_S.im')
```

output of `setjy` captured in variable "result":

```
{'0': {'0': {'fluxd': array([ 8.44827557,  0.        ,  0.        ,  0.        ])},
   '1': {'fluxd': array([ 8.13441944,  0.        ,  0.        ,  0.        ])},
   '2': {'fluxd': array([ 7.84281111,  0.        ,  0.        ,  0.        ])},
   'fieldName': '0137+331=3C48'},
  'format': "{field Id: {spw Id: {fluxd: [I,Q,U,V] in Jy}, 'fieldName':field name }}"}
```

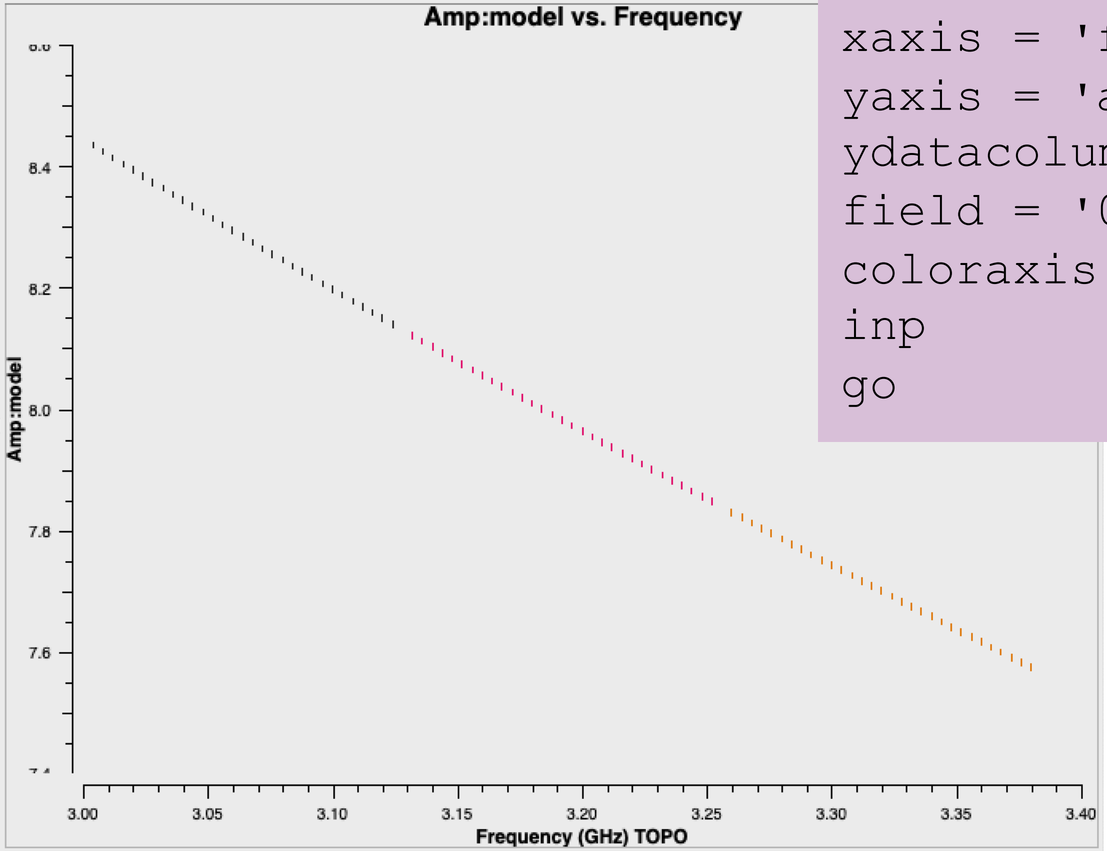
CASA reports in casalog:

```
Selected 54756 out of 97929 rows.
0137+331=3C48 (fld ind 0) spw 0  [I=8.4483, Q=0, U=0, V=0] Jy @ 3e+09Hz, (Perley-Butler 2017)
0137+331=3C48 (fld ind 0) spw 1  [I=8.1344, Q=0, U=0, V=0] Jy @ 3.128e+09Hz, (Perley-Butler 2017)
0137+331=3C48 (fld ind 0) spw 2  [I=7.8428, Q=0, U=0, V=0] Jy @ 3.256e+09Hz, (Perley-Butler 2017)
```

INTERACTIVE

Examine flux density scale calibrator model

- CASA task *plotms*



In CASA

```
default plotms
inp
vis = 'standard_calibration.ms'
xaxis = 'freq'
yaxis = 'amp'
ydatacolumn = 'model'
field = '0'
coloraxis = 'spw'
inp
go
```

Setting the flux density scale manually: *setjy*

- User can also provide flux density values instead of letting the task calculate them (manual mode)

```
standard        =     'manual'
    fluxdensity     = [8.446, 0, 0, 0]    # Stokes I, Q, U,V in Jy
    spix            = [-0.925,0]          # [alpha, curvature]
    reffreq         = '3 GHz'
```

Can also provide:

`polindex`: coefficients for frequency dependence of linear polarization fraction
`polangle`: coefficients for frequency dependence of polarization angle
`rotmeas`: rotation measure (rad/m^2)

* Polarization discussed later today in Frank Schinzel's talk

Calibration

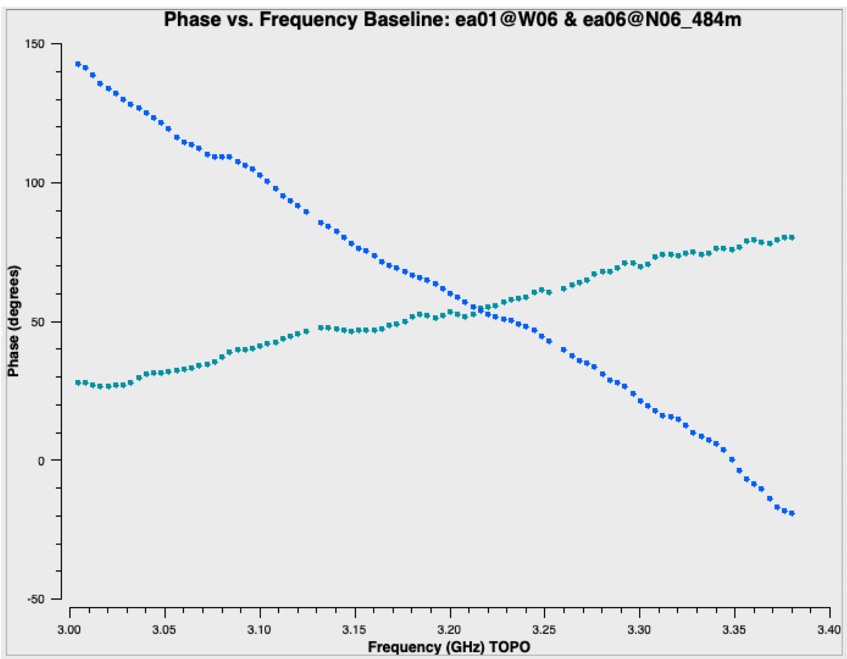
- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

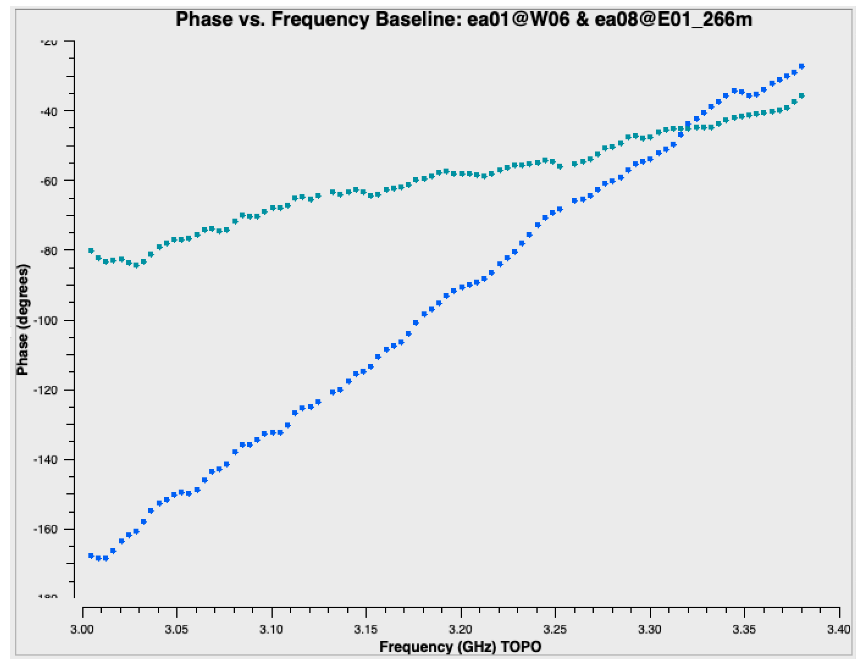
} *A priori* calibration

- ✓ Setting the flux density scale
- Delay calibration
- Pre-bandpass phase-only calibration (high-freq)
- Bandpass calibration
- Complex gain calibration
- (Polarization Calibration)
- Setting the flux density scales of the complex gain calibrators

Antenna-based residual delays

- Seen in UV data as linear phase-ramp vs frequency:
  - varying with baseline, correlation (RR, LL), baseband





Calibrating antenna-based delays

- CASA task *gaincal*

```
# In CASA
default gaincal
vis = 'standard_calibration.ms'
caltable = 'delays.cal'
solint = 'inf'              #  'inf' = infinite: combines all data within a scan
refant = 'ea10'
scan = '5'   #  can use one scan, or use all with e.g. "field = '0' "  "combine = 'scan' "
gaintype = 'K'
gaintable = ['antpos.cal', 'tecim.cal']
```

Use a strong (high signal-to-noise) source--- e.g. flux/bandpass calibrator.

`gaintype = 'K'`: solve for the residual delay solutions

`gaintable = [list]`: include all previous calibration tables

INTERACTIVE

A note on gaincal parameter *docallib*

In *current* version of CASA, make sure

```
docallib = False
```

- `docallib` refers to a "calibration library", a new portable interface for describing ensembles of calibration replacing `gaintable`, `gainfield`, etc… parameters.
- Enables on-the-fly calibration in CASA's calibration tasks.
- Provides increased capability and flexibility.
- Incorporated into the next CASA-pipeline release.

Calibration

- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

} *A priori* calibration

- ✓ Setting the flux density scale
- ✓ Delay calibration
- Pre-bandpass phase-only calibration (high-freq)
- Bandpass calibration

} Bandpass

- Complex gain calibration
- (Polarization Calibration)
- Setting the flux density scales of the complex gain calibrators

Before Bandpass Calibration

- Bandpass calibration is needed not just for spectral-line observations, but also for continuum.
- Before calibrating the bandpass, may opt to do phase-only calibration on the bandpass calibrator (to be applied when calibrating the bandpass).
  - Prevents decorrelation when vector averaging.
  - Critical for high frequency observations.
  - Can also be used in low frequency observations.

Bandpass Calibration

Needed for continuum observations too!

Uncalibrated bandpass:

- bandpass calibrator
- amp vs freq
- parallel-hands (RR, LL)
- avg in time
- iterate over baseline
- color by spw

Note: *sensitivity falls off in ~3 channels at each edge of a spw.* (Effect of digital filtering.)

Pre-bandpass phase-only calibration

CASA task *gaincal*

```
caltable = '<output cal table>'
solint = 'int'          # 'int' = integration
calmode = 'p'           # phase-only
spw = '0~2:13~18'       # a few RFI-free channels
gaintype = 'G'          # standard gain cal: one solution per pol, spw
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal']
```

Use *short solution interval* and a *few channels* per spw (RFI-free) to avoid de-correlation.

The resulting caltable must *only* be used for calibrating the bandpass.

INTERACTIVE

Bandpass calibration

- CASA task *bandpass*

```
# In CASA
default bandpass
inp
vis = 'standard_calibration.ms'
caltable = 'bandpass.cal'
field = '0'
solint = 'inf'
refant = 'ea10'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal']
inp
go
```

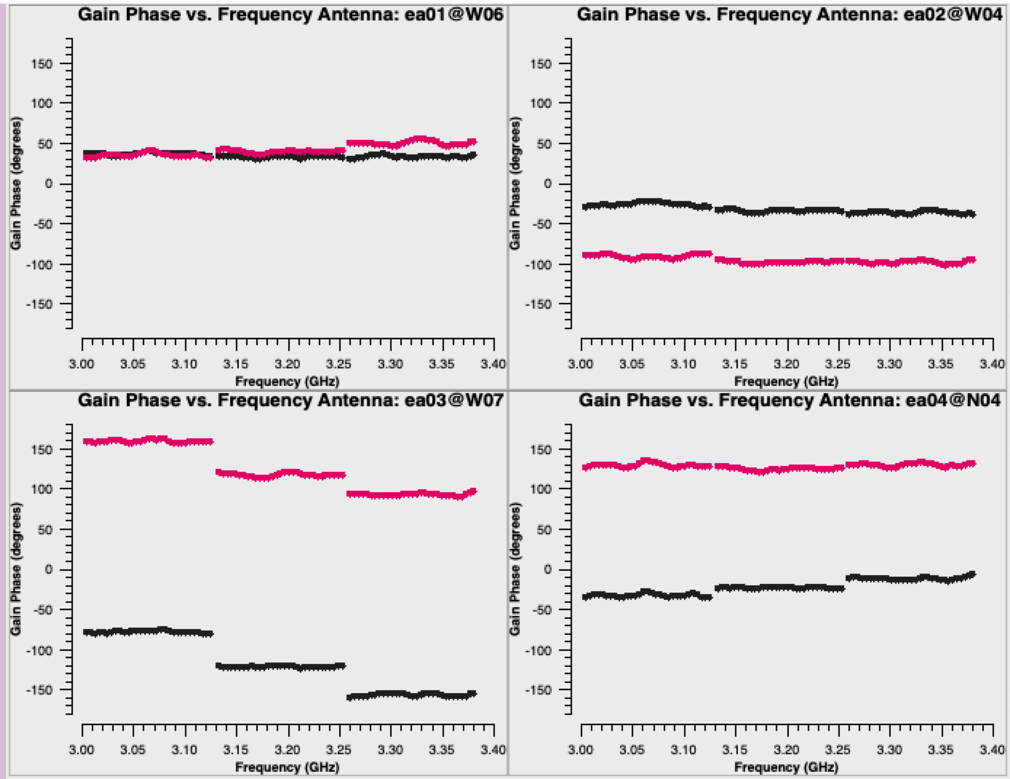
- `solint` can provide an interval in time and/or frequency

If bandpass cal ≠ flux cal, must account for spectral index/curvature.
See Topical CASAguide: "Correcting for a Spectral Index in Bandpass Calibration":
https://casaguides.nrao.edu/ → VLA

Examine bandpass calibration *phase* solutions

- CASA task *plotms*
  - Parameter "`vis`" can be a caltable
  - `coloraxis` = `'corr'` → actually polarization

```
# In CASA
default plotms
inp
vis = 'bandpass.cal'
gridrows = 2
gridcols = 2
xaxis = 'freq'
yaxis = 'phase'
iteraxis = 'antenna'
coloraxis = 'corr'
plotrange = [-1,-1,-180,180]
inp
go
```

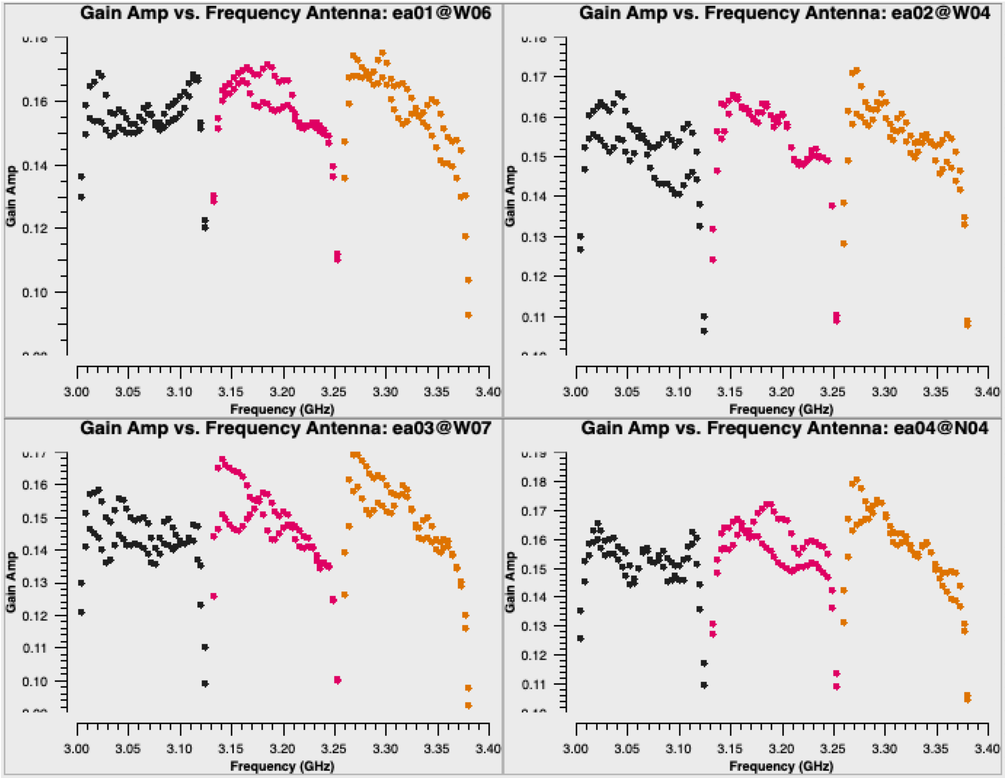


Move forward with ▶ and see that reference antenna (ea10) has phases = 0°.

Examine bandpass calibration *amp* solutions

- CASA task *plotms*

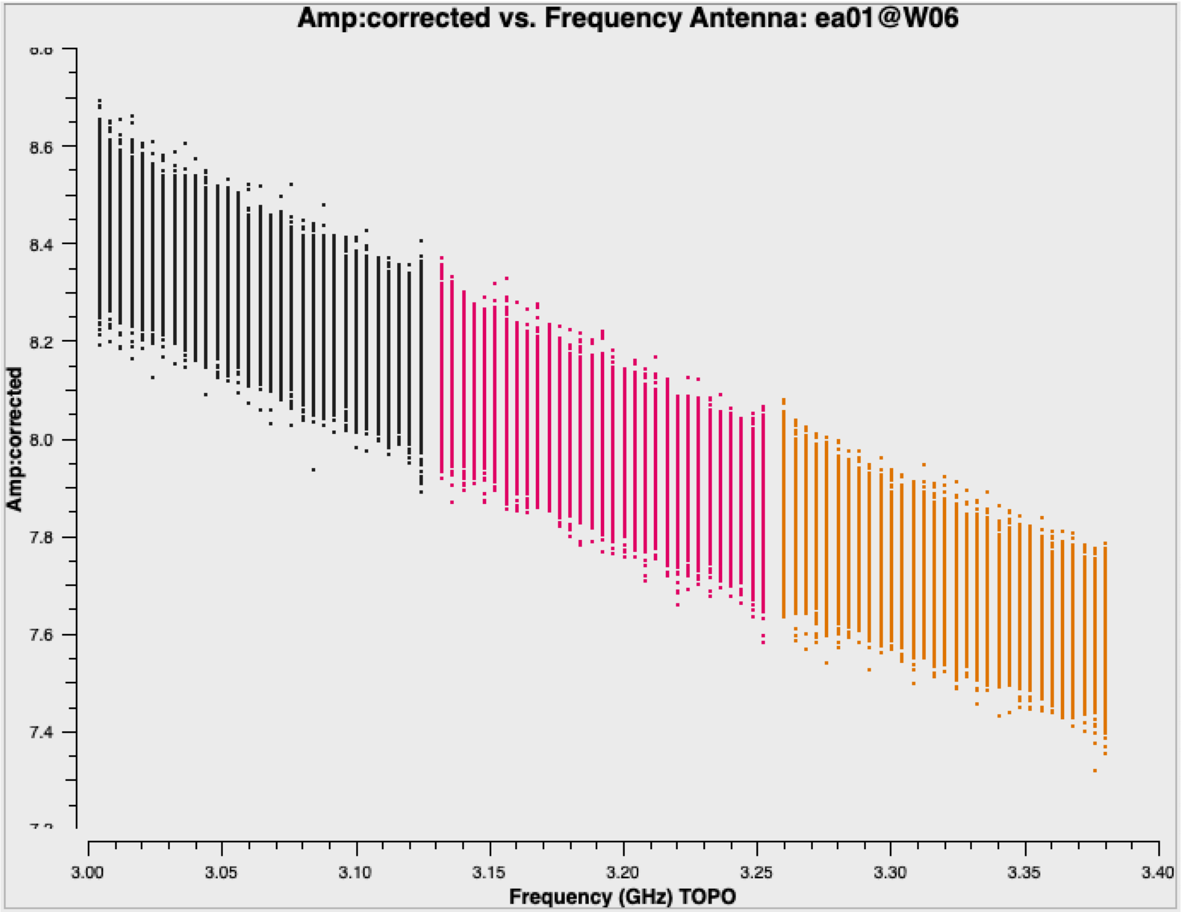
```
# In CASA
tget plotms
inp
yaxis = 'amp'
coloraxis = 'spw'
plotrange = []
inp
go
```



INTERACTIVE

Bandpass-corrected 3C48 data

(CASA tasks *applycal* and *plotms*)



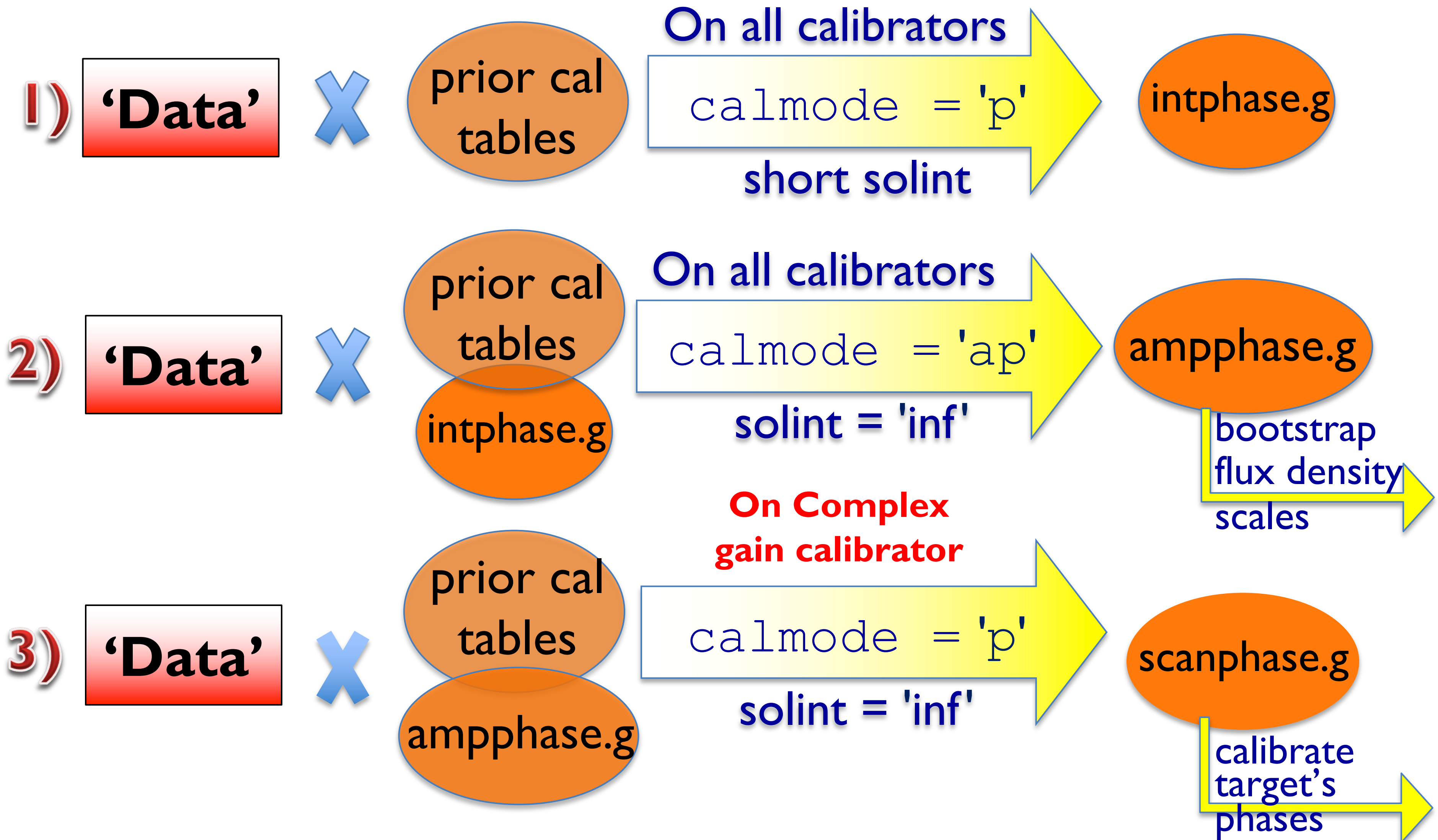
Calibration

- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

} *A priori* calibration

- ✓ Setting the flux density scale
- ✓ Delay calibration
- ✓ Pre-bandpass phase-only calibration (high-freq)
- ✓ Bandpass calibration
- Complex gain calibration
- (Polarization Calibration)
- Setting the flux density scales of the complex gain calibrators

Complex Gain Calibration: *gaincal*, High Freq



Complex Gain Calibration: *gaincal*, Low Freq

1) 'Data' X prior cal tables → On all calibrators `calmode = 'ap'` solint = 'inf' → scanphase.g → bootstrap flux density scales

- Examine the resulting solutions (plotms)
- If the phases show rapid variations (e.g., due to ionosphere), use the method outlined for high frequencies.
- The VLA calibration pipeline uses the high-freq approach.

Complex gain calibration

- CASA task *gaincal*

INTERACTIVE

`field`: include fluxcal *and* gain calibrator in order to (later) transfer flux scaling

`spw`: could choose to avoid low-sensitivity channels at each spw edge

```
# In CASA
default gaincal
inp
vis = 'standard_calibration.ms'
caltable = 'scanphase.gcal'
field = '0,1'
refant = 'ea10'
gaintable = ['antpos.cal', 'tecim.cal',
             'delays.cal', 'bandpass.cal']
inp
go
```

Important defaults

`solint = 'inf'`: yields one solution per scan on complex gain calibrator

`minsnr = 3.0`: reject solutions at lower signal to noise than this value

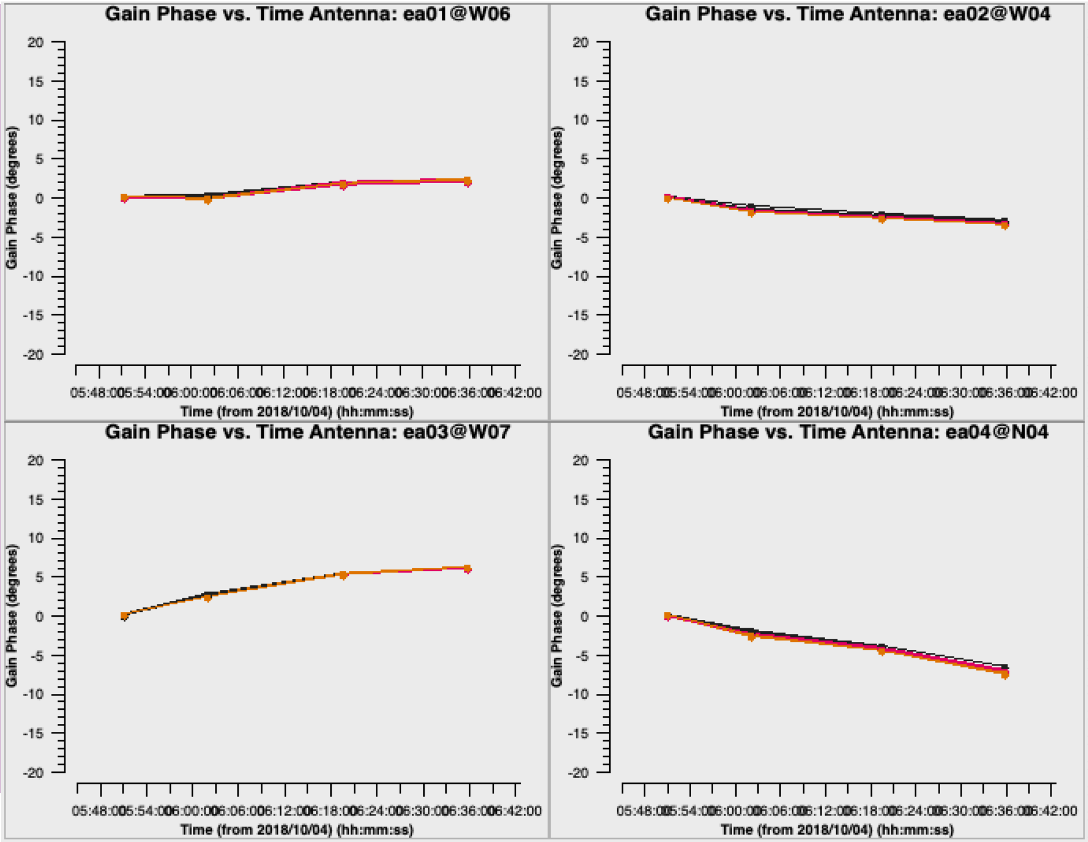
`calmode = 'ap'`: perform both amplitude (`a`) and phase (`p`) calibration

`solnorm = False`: no need to normalize because we're doing amplitude calibration

Examine complex gain cal *phase* solutions

- CASA task *plotms*

In CASA
```
default plotms
inp
vis = 'scanphase.gcal'
gridrows = 2
gridcols = 2
xaxis = 'time'
yaxis = 'phase'
iteraxis = 'antenna'
coloraxis = 'spw'
xconnector = 'line'
plotrange = [-1,-1,-20,20]
inp
go
```



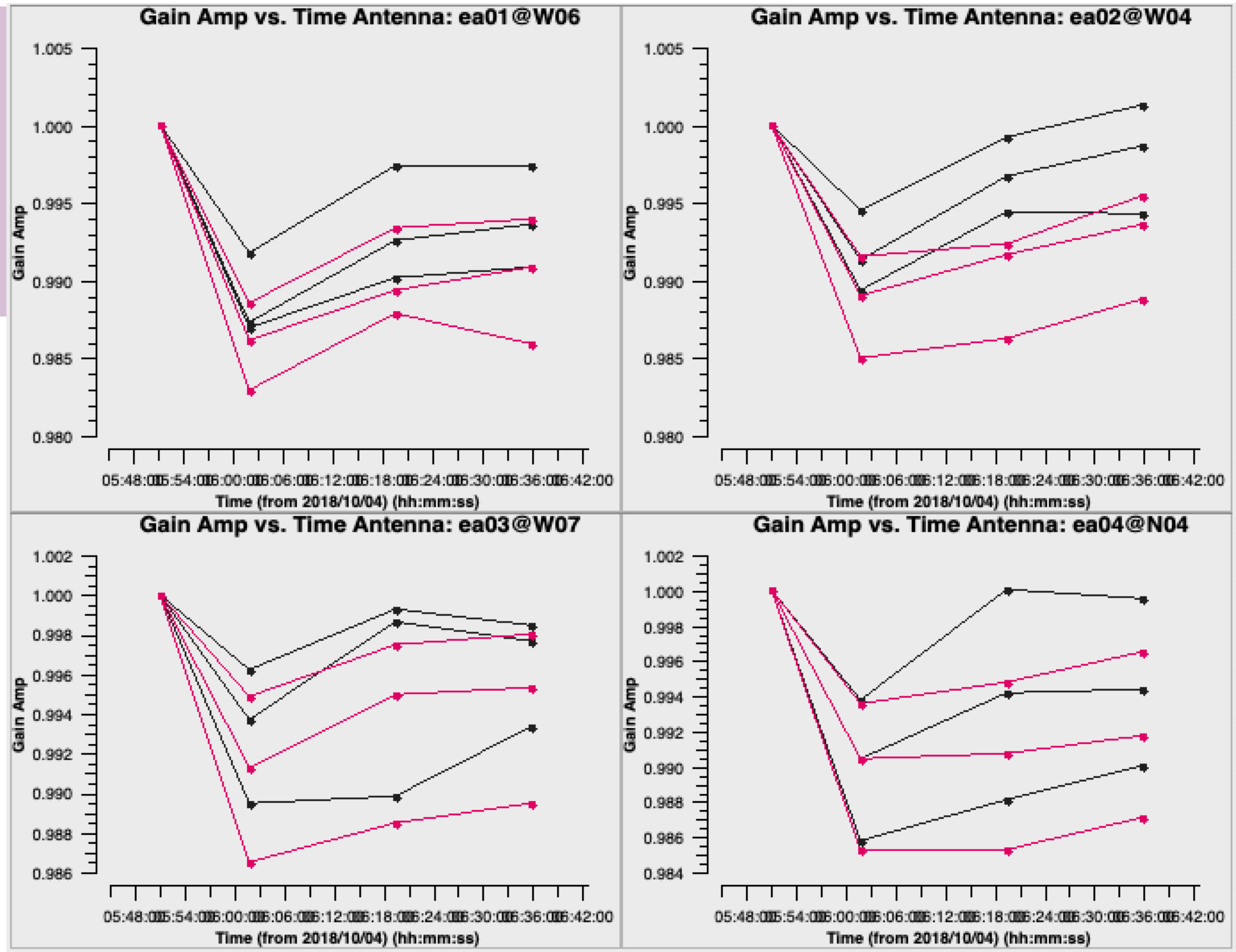
INTERACTIVE

INTERACTIVE

Examine complex gain cal *amp* solutions

- CASA task *plotms*

```
# In CASA
tget plotms
yaxis = 'amp'
coloraxis = 'corr'
plotrange = []
```



The first data point is 3C48: the flux calibrator.

(=1 because we applied bandpass calibration, also based on 3C48)

Calibration

- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

A priori calibration

- ✓ Setting the flux density scale
- ✓ Delay calibration
- ✓ Pre-bandpass phase-only calibration (high-freq)
- ✓ Bandpass calibration
- ✓ Complex gain calibration
- (Polarization Calibration) ← **Frank Schinzel's talk**
- Setting the flux density scales of the complex gain calibrators

Scaling the flux densities: CASA task *fluxscale*

Bootstrapping the flux density scales:

- We earlier used `setjy` to set the flux density values for the flux calibrator, and `gaincal` to solve for the antenna gains (`'scanphase.gcal'`) based on those values. In `fluxscale` task, those gains are used to determine flux density of the complex gain calibrators.
- Fits a 1st- (linear) or 2nd-order curve to each spectrum to report spectral index and curvature. Choice of fit order may depend on amount of curvature, signal-to-noise of calibrator.
- Results can be stored in a variable or written to a file.

`reference`	= name of flux density calibrator
`transfer`	= name or fields of complex gain calibrators
`fitorder`	= `1` or `2` (1st or 2nd order curve fit to spectrum; default is `1`)
`listfile`	= name of output file to store results (optional)

Scaling the flux densities: CASA task *fluxscale*

Fluxscale produces a new calibration table but there are two options:

```
fluxtable = '<output cal table>'
incremental = True or False
```

If `incremental = False`:

The `<output cal table>` *replaces* the input '`ap`' table.

If `incremental = True`:

The `<output cal table>` contains *only* the scaling factors, and must be used alongside the input '`ap`' table when applying calibration.

Which approach to use is a matter of personal preference.

Bootstrap the flux density, fit spectrum

- CASA task *fluxscale*

```
# In CASA
default fluxscale
inp
vis = 'standard_calibration.ms'
caltable = 'scanphase.gcal'
fluxtable = 'fluxscale.cal'
reference = '0137+331=3C48'
transfer = ['J0259+0747']
fitorder = 1
incremental = False
inp
go
```

Results reported in casalogger:

```
Flux density for J0259+0747 in SpW=0 (freq=3.062e+09 Hz) is: 0.977186 +/- 0.000949625 (SNR = 1029.02, N = 54)
Flux density for J0259+0747 in SpW=1 (freq=3.19e+09 Hz) is: 0.985137 +/- 0.000945082 (SNR = 1042.38, N = 54)
Flux density for J0259+0747 in SpW=2 (freq=3.318e+09 Hz) is: 0.992095 +/- 0.000994838 (SNR = 997.242, N = 54)
Fitted spectrum for J0259+0747 with fitorder=1: Flux density = 0.984792 +/- 0.000178186 (freq=3.18829 GHz) spidx=0.188748 +/- 0.00557161
Storing result in fluxscale.cal
```

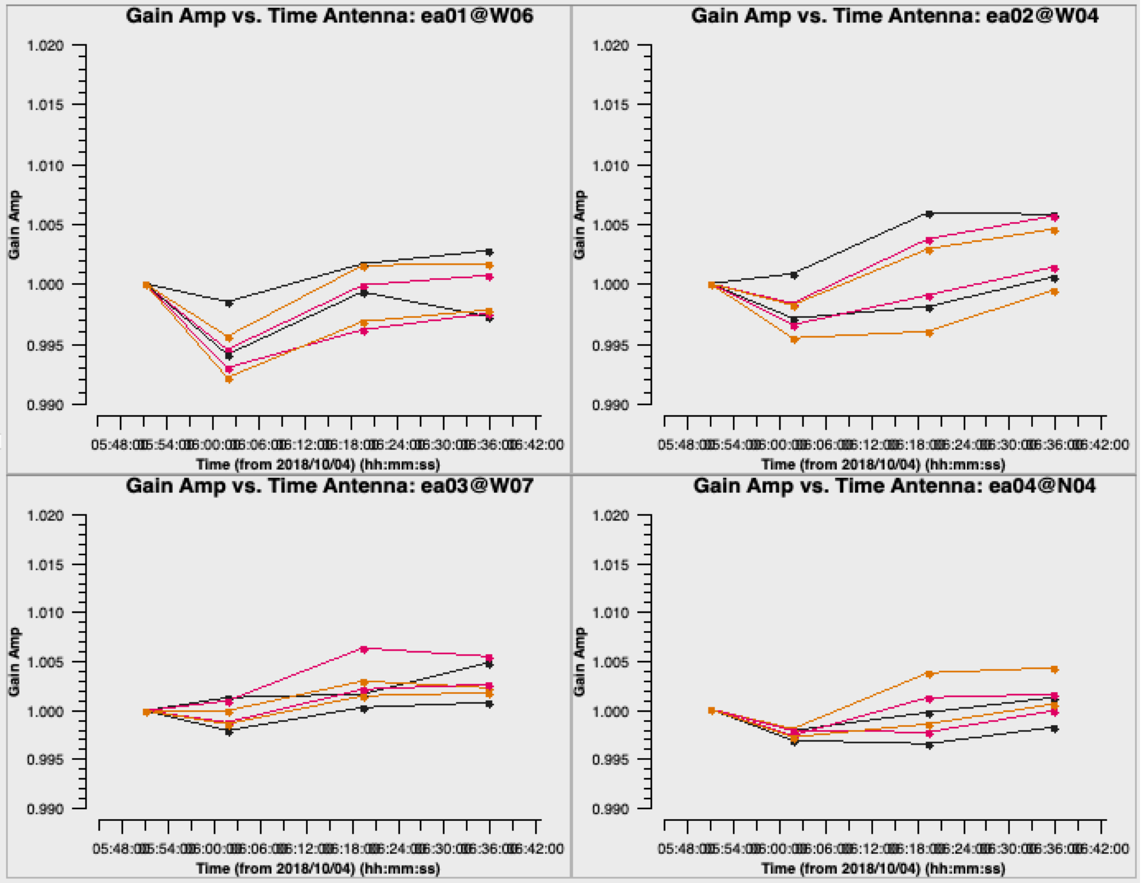
INTERACTIVE

INTERACTIVE

Examine rescaled amplitude solutions

- CASA task *plotms*

```
# In CASA
default plotms
vis = 'fluxscale.cal'
xaxis = 'time'
yaxis = 'amp'
gridrows = 2
gridcols = 2
coloraxis = 'spw'
iteraxis = 'antenna'
yselfscale = True
xconnector = 'line'
```



Calibration

- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

} *A priori* calibration

- ✓ Setting the flux density scale
- ✓ Delay calibration
- ✓ Initial phase-only calibration (high-freq)
- ✓ Bandpass calibration
- ✓ Complex gain calibration
- ✓ (Polarization Calibration)
- ✓ Setting the flux density scales of the complex gain calibrators

Applying the calibration

- First apply calibration not to the targets, but to the calibrators themselves.
  - Looking at calibrated visibilities for the calibrators is a good way to confirm that the calibration is good and to identify bad data that may have been missed before (e.g. RFI).
- Multiple calibrators in the data?
  - Simplest to use one run of *applycal* per calibrator

INTERACTIVE

Apply the calibration: *flux/bandpass calibrator*

- CASA task *applycal*

`gaintable`: the calibration tables

`gainfield`: fields from the above tables (if table has solutions from >1 source)

`interp`: apply *`nearest`* solution? or interpolate between solutions (*`linear`*)?

`calwt`: use system calibration to weight the data? not yet for VLA data! (`False`)

```
# In CASA
default applycal
vis = 'standard_calibration.ms'
field = '0137+331=3C48'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', '0137+331=3C48']
interp = ['', '', '', '', 'nearest']
calwt = False
```

Apply the calibration: *gain calibrators*

- CASA task *applycal*

Now apply the calibration to each of the phase calibrators. Most of the input parameters remain the same...

```
# In CASA
inp
field = 'J0259+0747'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', 'J0259+0747']
interp = ['', '', '', '', 'nearest']
inp
go
```

INTERACTIVE

Examine the calibrated data (the corrected column) with *plotms*.

Flag, if needed, and re-calibrate.

Apply the calibration: *targets*

- CASA task *applycal*

`gainfield`: apply the solution from the appropriate complex gain calibrator

`interp`: use *`linear`* interpolation to interpolate in time between the complex gain calibration solutions

```
# In CASA
default applycal
vis = 'standard_calibration.ms'
field = '3C75'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', 'J0259+0747']
interp = ['', '', '', '', 'linear']
calwt = False
```

INTERACTIVE

Split target source(s) into their own MS

- CASA task *split*

- Split target source(s) using corrected column.
- Optionally:
  - apply time and/or frequency averaging
  - choose certain antennas and/or spws/channels

```
# In CASA
default split
vis = 'standard_calibration.ms'
outputvis = '3C75.ms'
field = '3C75'      # or field='2'
correlation = 'RR,LL'
```

- The *split* out data will occupy the 'data' column in the output MS.
- This step strongly recommended before re-weighting the data (`statwt`) and before imaging.
- Note: if using full-polarization data that is not polarization-calibrated, use: `correlation = 'RR,LL'`

Re-weight the visibilities

- CASA task *statwt*

INTERACTIVE

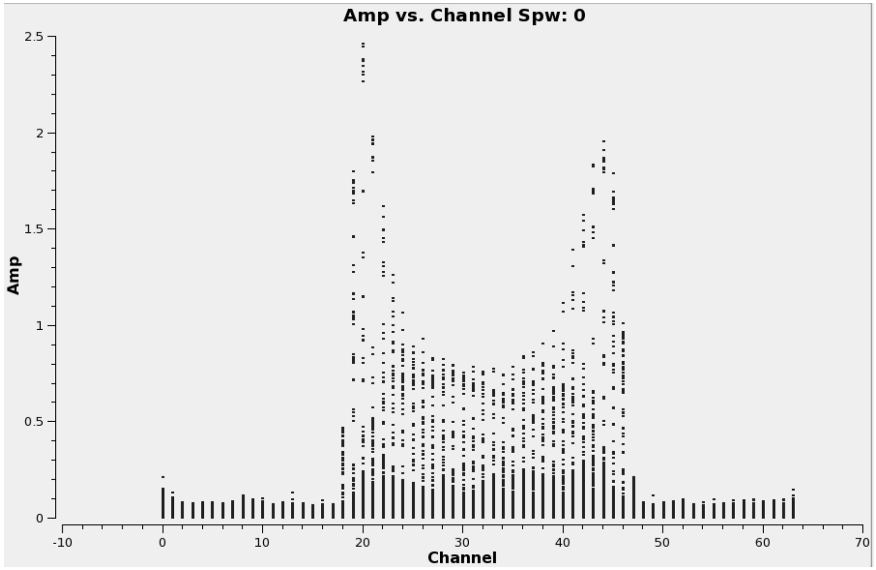
- Data weights are initialized to be based on channel bandwidth and integration-time ($2\Delta\nu\Delta\tau$).
- *statwt* re-weights the visibilities according to their scatter:
  - down-weight underperforming antennas
  - down-weight frequency ranges affected by RFI
- Use on *fully-calibrated* data!

```
# In CASA
default statwt
inp
vis = '3C75.ms'
datacolumn = 'data'
inp
go
```

- *Note for spectral lines*:
  - use `fitspw` parameter to exclude strong lines

Continuum Subtraction: *uvcontsub*

```
vis           = 'my.ms'
fitspw        = '0:4~13;52~60'   # can use multiple spws
excludechans  =  False   or   True
want_cont     =        False
```



* See online CASAguide for spectral line data reduction (Carbon Star IRC+10216)

Doppler Correction: *cvel2*

- The VLA offers Doppler *setting*, NOT Doppler *Tracking*
- Line of interest may shift (by channel) in different observations
- *tclean* with `specmode='cube'` will do Doppler corrections on-the-fly! No need to Doppler-correct for general imaging cases
  - provide *tclean* a list of MSs (do not concat!) and `restfreq`
  - if multiple lines in data, run *tclean* separately for each line
- <u>However</u>, if *self-calibrating* on a strong/narrow spectral line, must first correct in the visibilities *before* running *tclean/gaincal*:
  - use *cvel2* task to Doppler-correct multiple MSs to same frame
  - run *tclean/gaincal* as usual (imaging talks tomorrow)

www.nrao.edu
science.nrao.edu
public.nrao.edu

The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.