VLA Data Reduction: *Standard Calibration*

Amy Kimball (NRAO)

Using these slides as a reference

This presentation is based on a 12.5-GB (raw) data set that can be downloaded from the new NRAO Data Archive:

data.nrao.edu

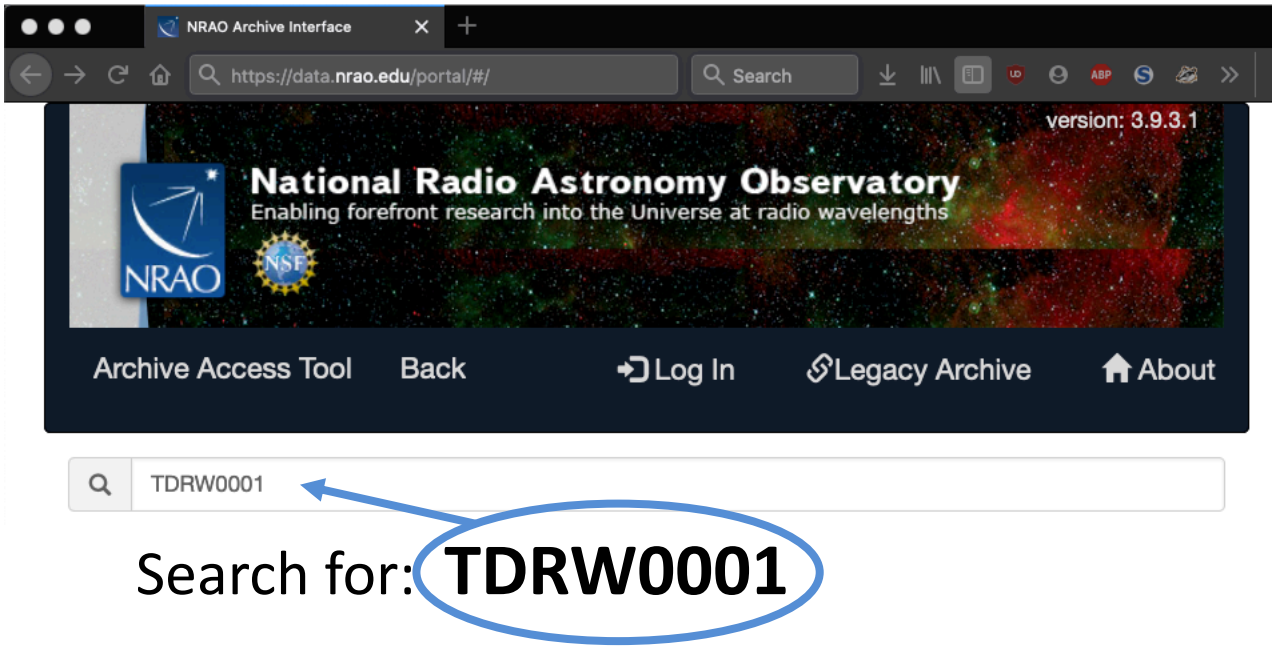
ON YOUR OWN

When you see text like this and the sidebar to the left:
```
CASA commands will appear like this
```

Accessing data used for this presentation

This presentation is based on a 12.5-GB (raw) data set that can be downloaded from the new NRAO Data Archive:

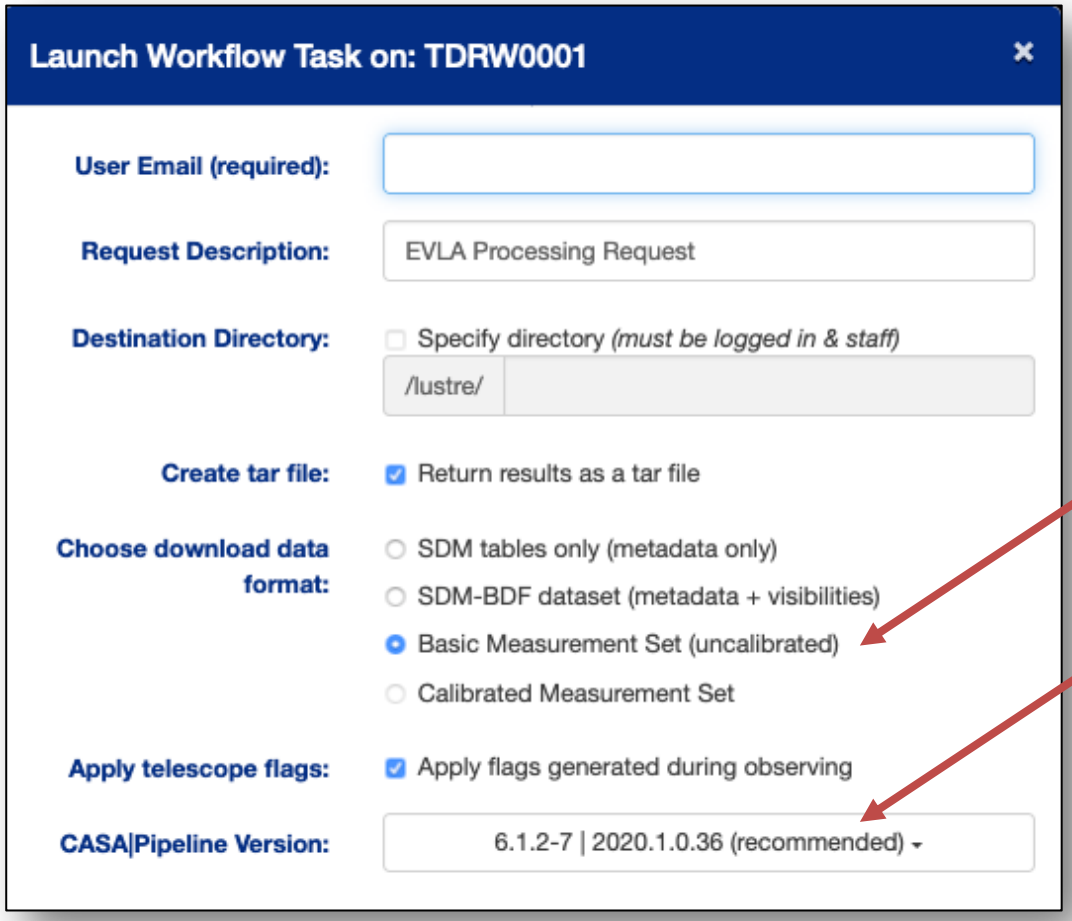
data.nrao.edu



Download ~12.5 GB data set: (TDRW0001.sb35624494.eb35628826.58395.23719237269)

ON YOUR OWN

Accessing data used for this presentation



When downloading a measurement set (MS), you must select the CASA version that will be used to convert the raw data (SDM-BDF) to MS.

CASA versions are usually (but not always) backwards compatible. CASA version shown here is the version used for this presentation.

ON YOUR OWN

Accessing data used for this presentation

ON YOUR OWN

Steps to prepare the data set for this presentation:

- Download from archive, save in working directory.
- Untar/unzip downloaded file.
- Untar/unzip measurement set (MS)
- Use mstransform to split out:
  - spw = '6~8'
  - scan = '5~11'
  - datacolumn = 'data'
  - keepflags = False
  - hanning = True
- Set name to something simple: "my_data.ms"

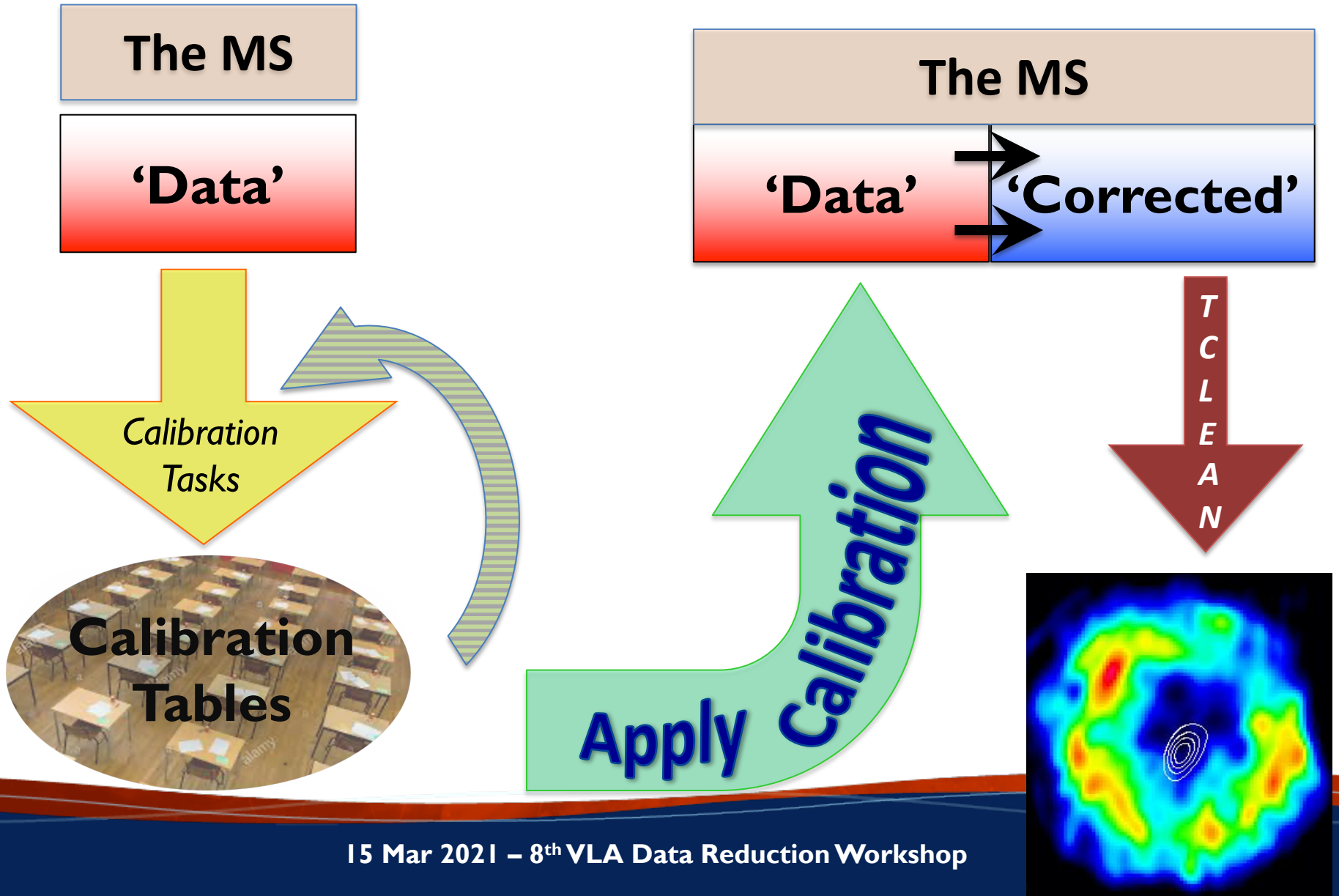
Final data set is ~1.2 GB.

The measurement set (MS) structure

'Data' column	'Corrected' Column	'Model' Column
Raw Data	Calibrated Data	(optional) 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



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' = for pre-upgrade VLA (*see documentation*)
'swpow' = EVLA switched-power gains
'rq' = EVLA requantizer gains
'swp/rq' = EVLA switched power gains/req. gains
'opac' = Tropospheric opacity
'gc' = VLA gain curve (zenith-angle-dependent gain)
'eff' = VLA antenna efficiency (sqrt(K/Jy))
'gceff' = VLA 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*

```
# CASA parameters for gencal
vis = 'my_data.ms'
caltable = 'antpos.cal'
caltype = 'antpos'
```

Antenna position corrections (if any) are 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 ea20 : -0.00190   0.00110  -0.00130
offsets for antenna ea25 : -0.00340   0.00190  -0.00280
```

ON YOUR OWN

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*** (>15 GHz).
- The VLA pipeline *always* performs this step.
- In *gencal*, set:

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

Opacity Corrections: *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' )
```

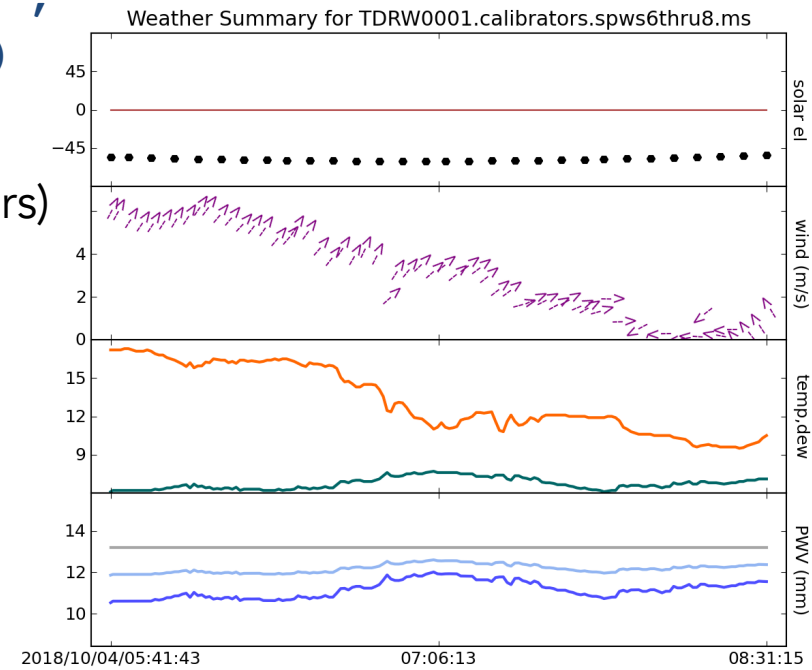
- Gives one value per spw:

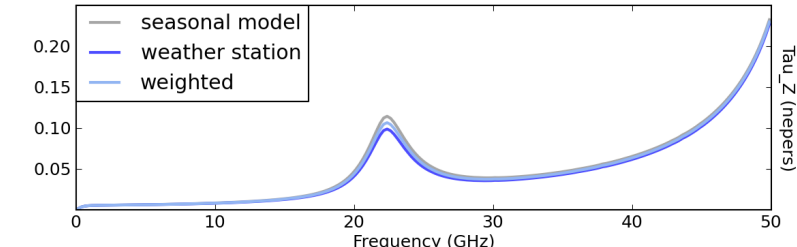
SPW : Frequency (GHz) : Zenith opacity (nepers)

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 = '0~2'  # (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
- **Still under commissioning**

The VLA pipeline does* NOT *perform TEC corrections.

Ionosphere correction (*Total Electron Content*)

- CASA "recipe" and CASA task *gencal*

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

```
# In CASA
# import the TEC image    (in CASA 5, import from "recipes")
from casatasks.private import tec_maps
tec_image, tec_rms_image, tec_graph = tec_maps.create(
    vis='my_data.ms', doplot=True)

# gencal parameters
caltype = 'tecim'
caltable = 'tecim.cal'
infile = tec_image
```

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

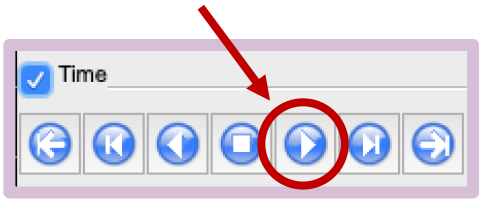
ON YOUR OWN

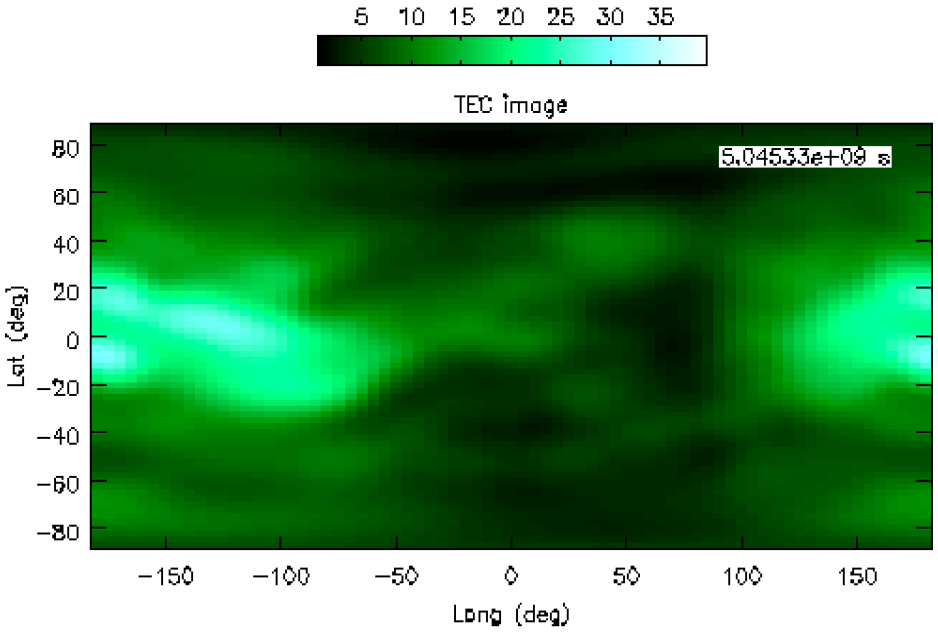
TEC image and rms image for this dataset

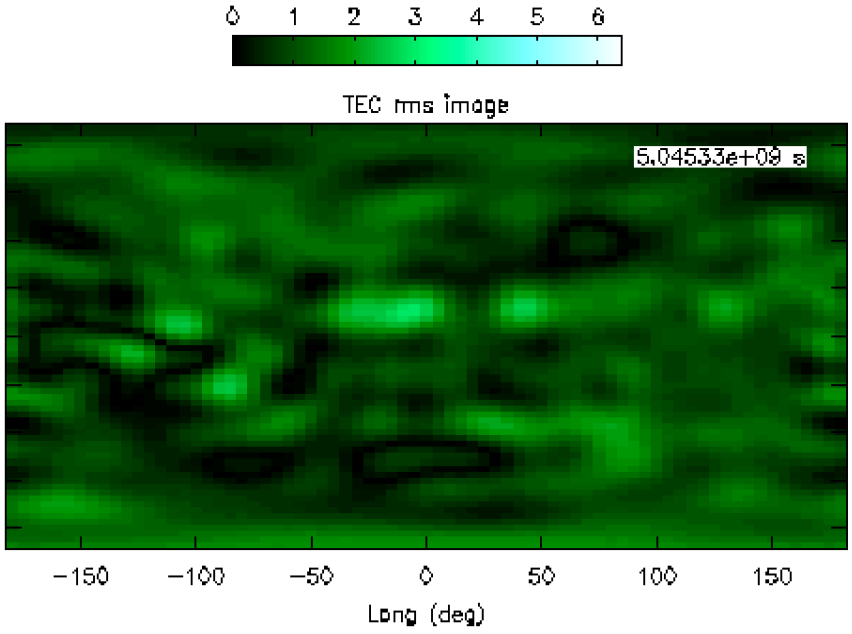
```
# in CASA
viewer('my_data.ms.IGS_TEC.im')
```

TEC is time dependent:

Menu bar → View → Animators







ON YOUR OWN

Requantizer gains: *gencal*

- Optimizes the digital power within each spectral window.
- Required for *3-bit data*. (This example data set is 8-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     =   True or False
usescratch           =      False
```

Identifying available flux density models

- CASA task *setjy*

```
# CASA parameters for setjy
standard = 'Perley-Butler 2017'  # default; other models available
listmodels = True
```

`listmodels`

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

P-band models have another standard available:

`standard = 'Scaife-Heald 2012'`:

3C48, 3C147, 3C196, 3C286, 3C295, 3C380

- 3C123 and 3C138 available for P-band only with "Perley-Butler 2017"

ON YOUR OWN

Setting the flux density scale

- CASA task *setjy*

```
# In CASA
result = setjy(vis='my_data.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:

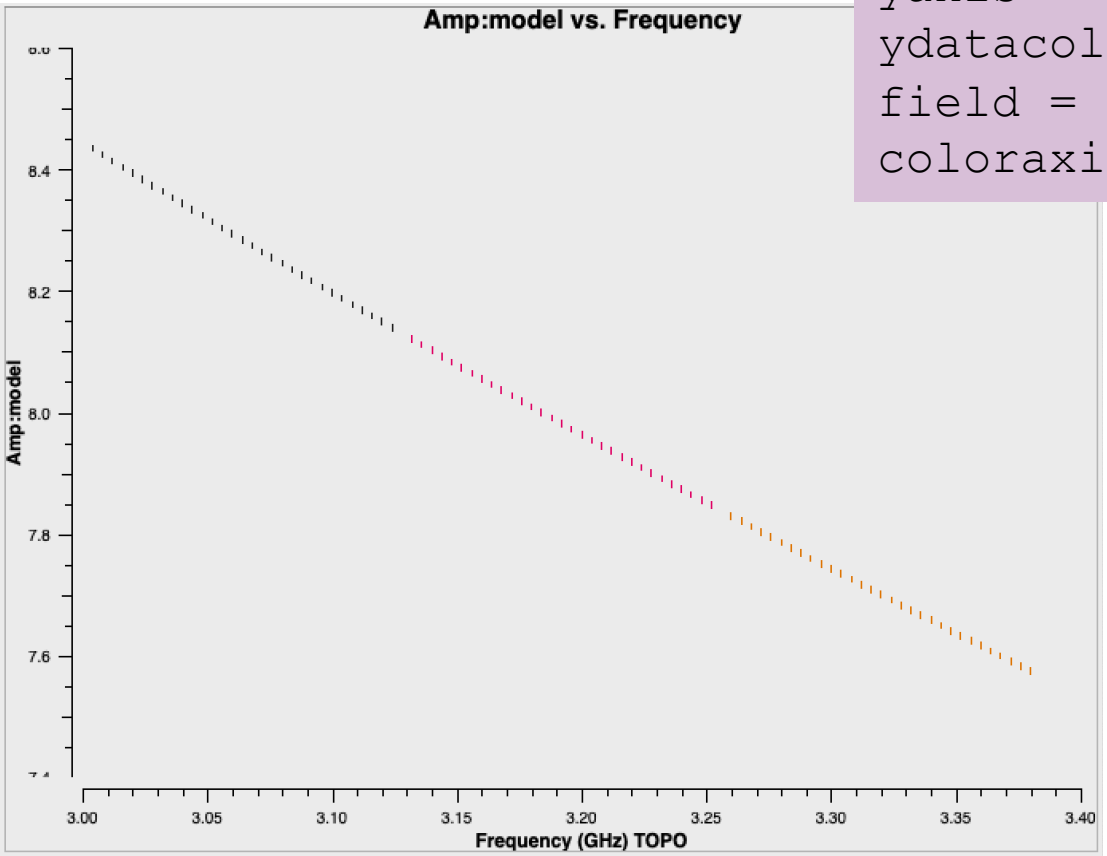
```
Selected 54756 out of 97929 rows.
0137+331=3C48 (fld ind 0) spw 0  [I=8.4483, Q=0, U=0, V=0] Jy @ 3e+09Hz, (Perley-Butler 2017)
0137+331=3C48 (fld ind 0) spw 1  [I=8.1344, Q=0, U=0, V=0] Jy @ 3.128e+09Hz, (Perley-Butler 2017)
0137+331=3C48 (fld ind 0) spw 2  [I=7.8428, Q=0, U=0, V=0] Jy @ 3.256e+09Hz,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

ON YOUR OWN

Examine flux density scale calibrator model

- CASA task *plotms*

```
# CASA parameters for plotms
vis = 'my_data.ms'
xaxis = 'freq'
yaxis = 'amp'
ydatacolumn = 'model'
field = '0'
coloraxis = 'spw'
```



ON YOUR OWN

Setting the flux density scale manually: *setjy*

- User can choose to provide flux density values rather than 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 use *setjy* to 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 tomorrow in Frank Schinzel's talk

Calibration

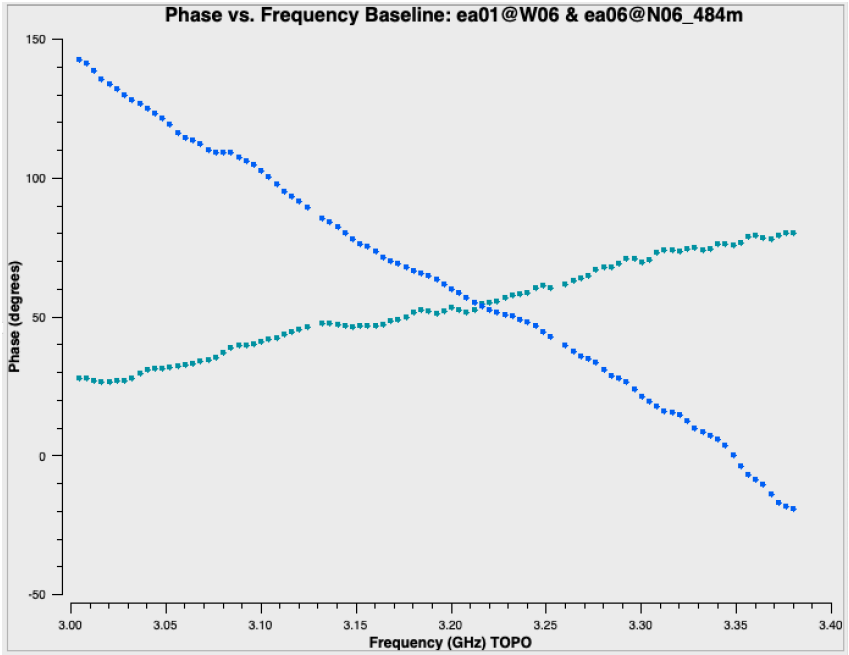
- ✓ Correcting antenna positions
- ✓ Gain Curves (high-freq)
- ✓ Opacity (high-) and Ionospheric (low-freq) corrections
- ✓ Re-quantizer gain calibration (mostly 3-bit data)

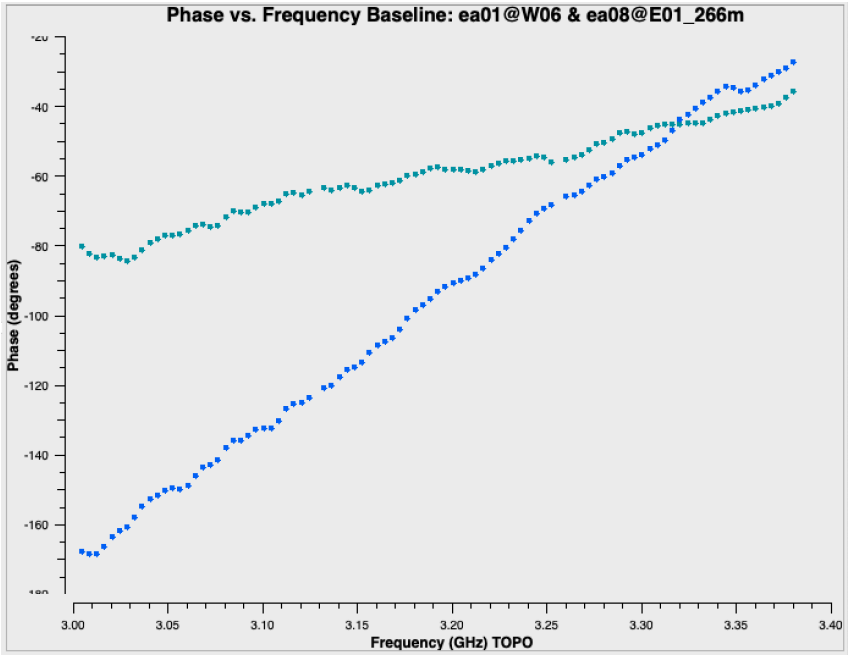
A priori calibration

- ✓ 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*

```
# CASA parameters for gaincal
vis = 'my_data.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

*Warning! * Data with "failed" solutions will be flagged later, during *applycal* stage

ON YOUR OWN

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 choose to do a phase-only calibration on the bandpass calibrator (to be applied *only* when calibrating the bandpass).
  - Prevents de-correlation when vector averaging.
  - Critical for *high frequency* observations.
  - Can also be used in low frequency observations.

Bandpass Calibration

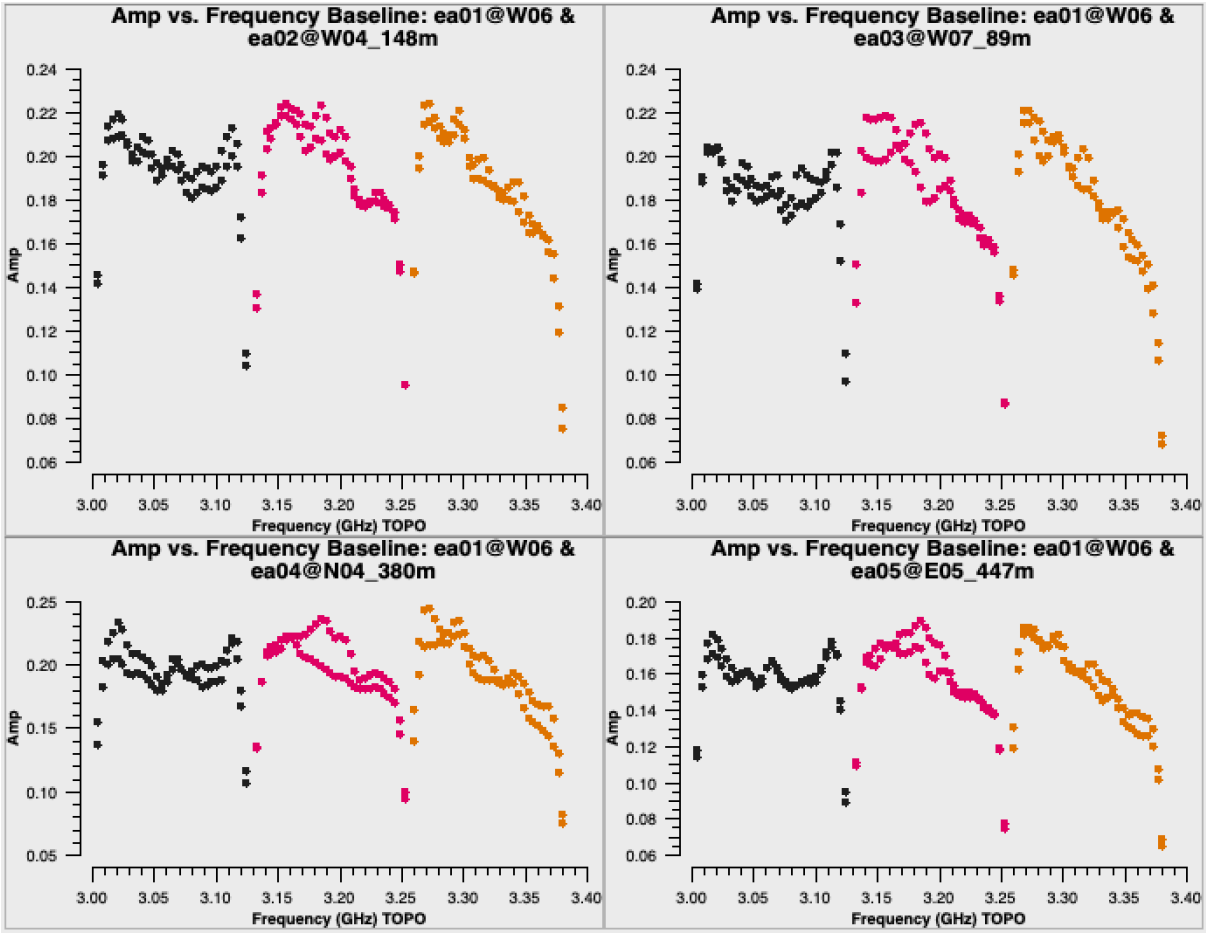
Needed for continuum observations too!

Uncalibrated bandpass!

Plotms settings:

- 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 gaincal: 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.

Bandpass calibration

- CASA task *bandpass*

```
# CASA parameters for bandpass
vis = 'my_data.ms'
caltable = 'bandpass.cal'
field = '0'
solint = 'inf'
refant = 'ea10'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal']
```

("pre-bandpass phase-only" caltable would also go in this list)

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

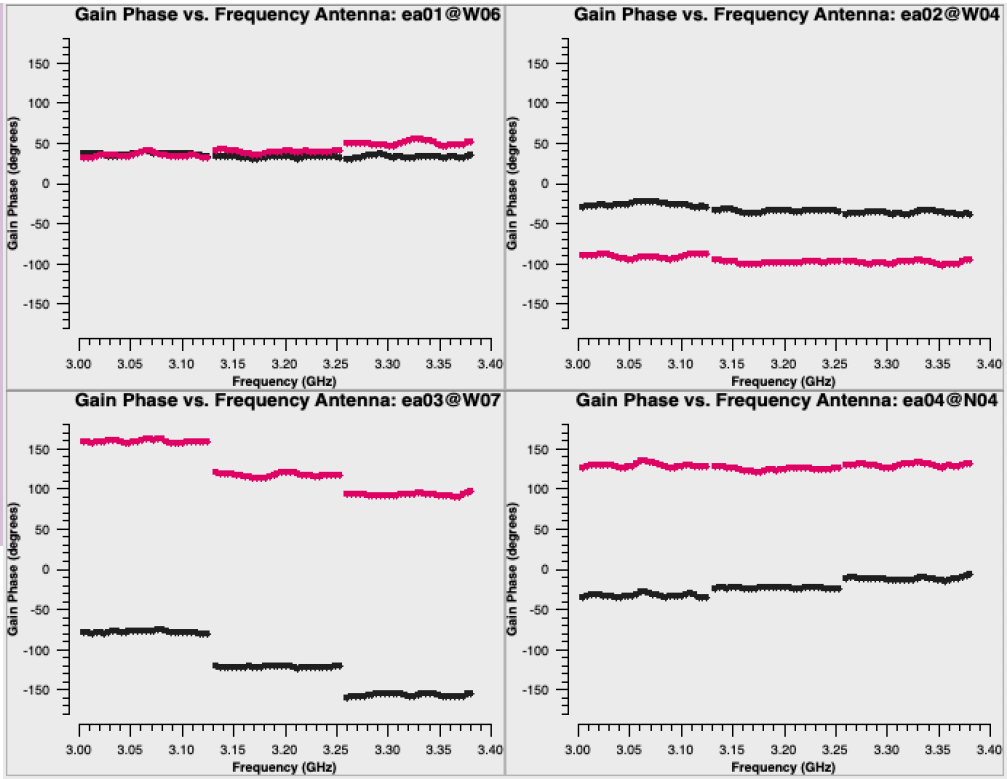
If bandpass cal ≠ flux cal, must account for spectral index/curvature.
Topical CASAguide: *"Correcting for a Spectral Index in Bandpass Calibration"*:
https://casaguides.nrao.edu/ → VLA

ON YOUR OWN

Examine bandpass calibration *phase* solutions

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

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



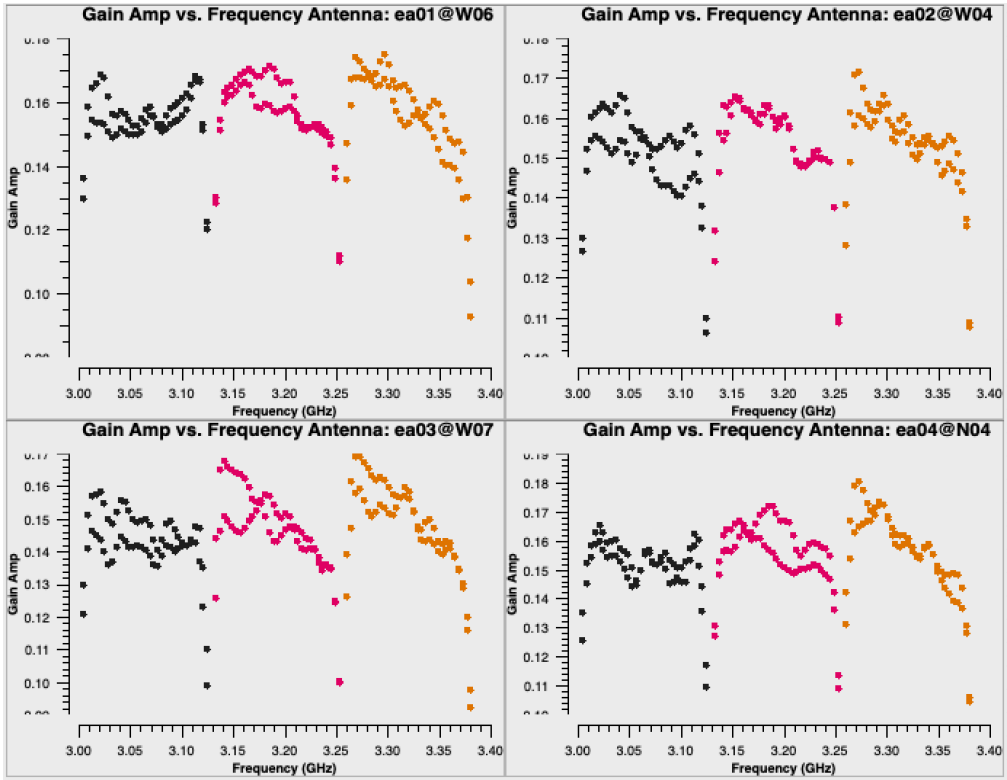
Move forward with  and see that reference antenna (ea10) has phases = 0°.

ON YOUR OWN

Examine bandpass calibration *amp* solutions

- CASA task *plotms*

```
# CASA parameters for plotms
vis = 'bandpass.cal'
gridrows = 2
gridcols = 2
xaxis = 'freq'
yaxis = 'amp'
iteraxis = 'antenna'
coloraxis = 'spw'
plotrange = []
```

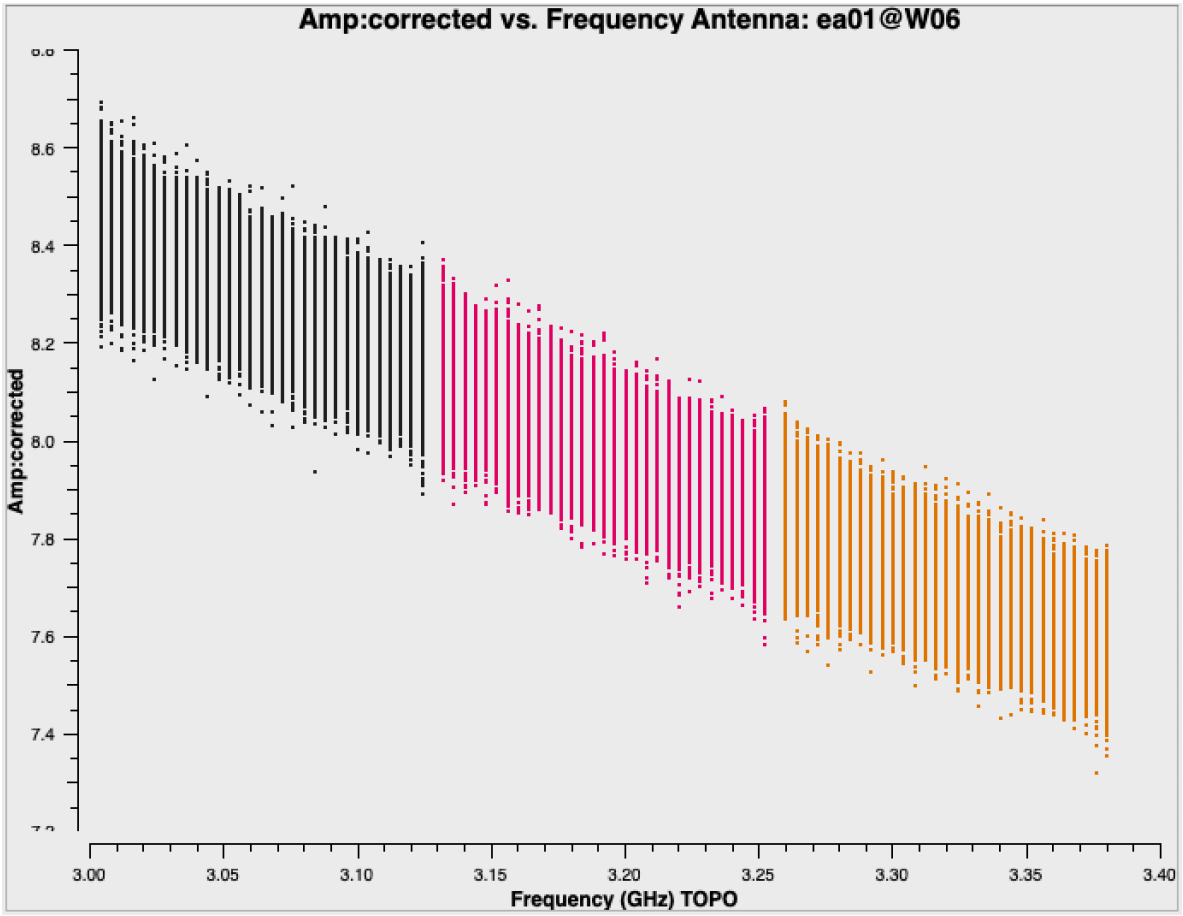


Note similarity to bandpass amp shapes (slide 29). These are the values that will be *applied* to the data in order to "correct" the bandpasses.

ON YOUR OWN

Bandpass-corrected 3C48 data

(CASA tasks *applycal* and *plotms)*



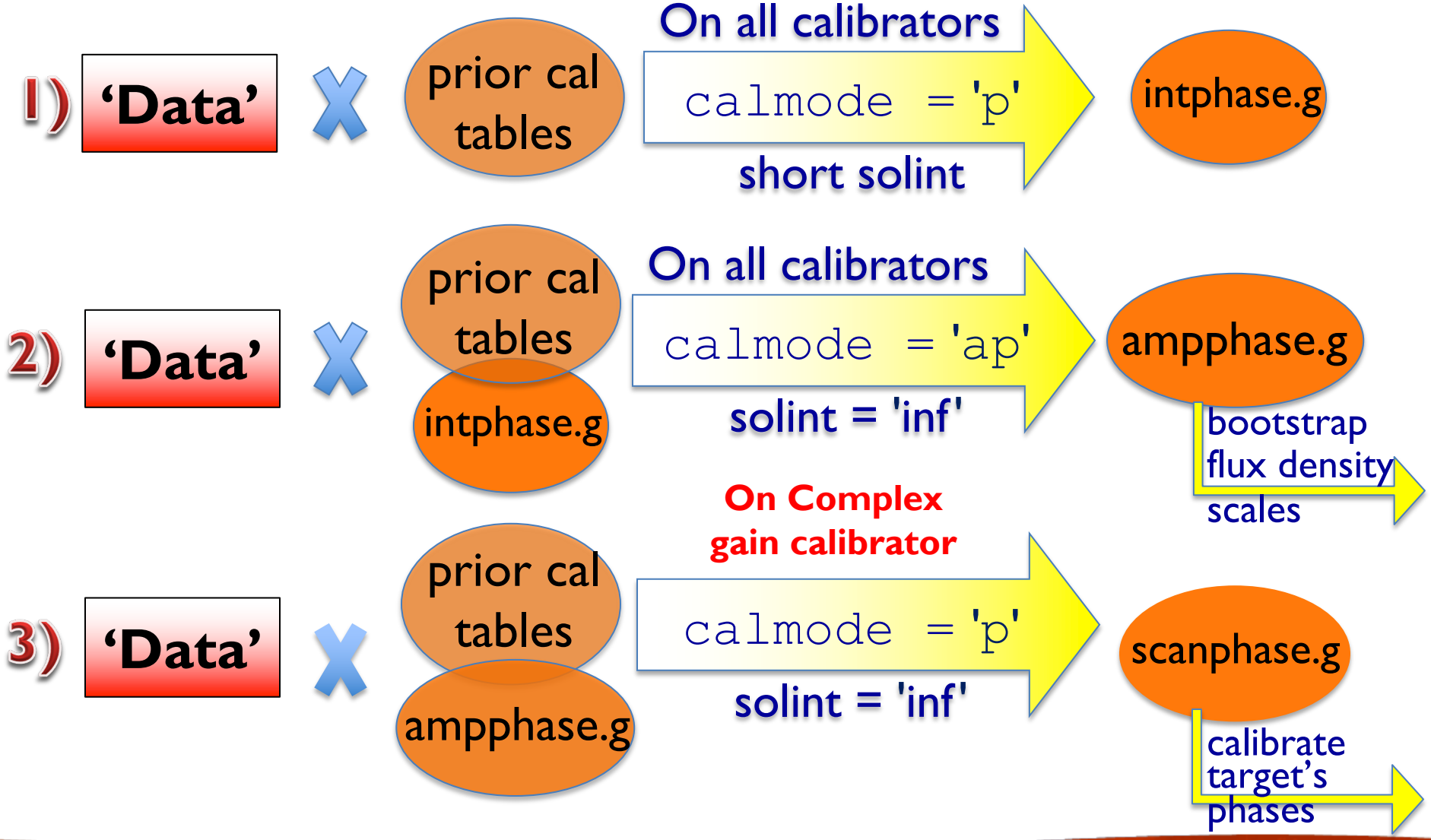
Calibration

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

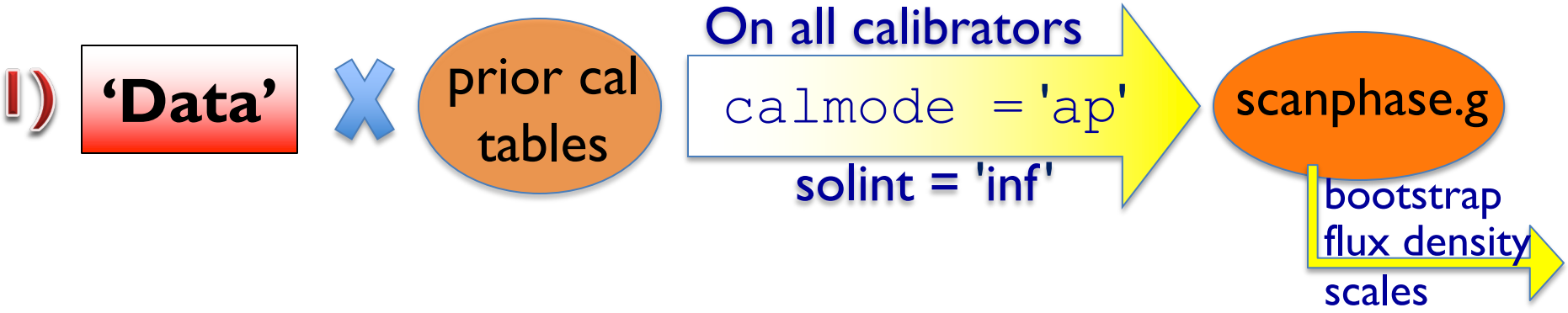
} *A priori* calibration

- ✓ 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



- 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 always uses the high-frequency approach.

Complex gain calibration

- CASA task *gaincal*

`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

```
# CASA parameters for gaincal
vis = 'my_data.ms'
caltable = 'scanphase.gcal'
field = '0,J0259+0747'
refant = 'ea10'
gaintable = ['antpos.cal', 'tecim.cal',
             'delays.cal', 'bandpass.cal']
```

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

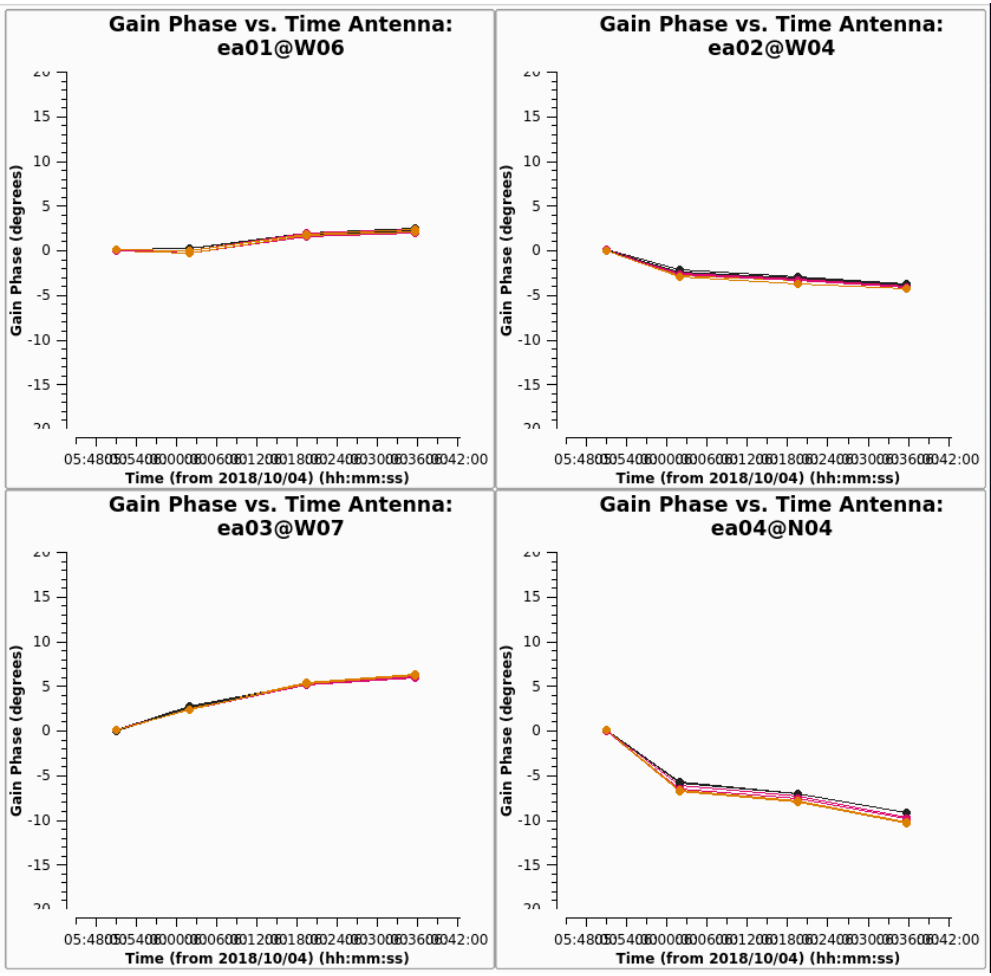
`solnorm = False`: no need to normalize, because we're doing amplitude calibration

ON YOUR OWN

Examine complex gain cal *phase* solutions

- CASA task *plotms*

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

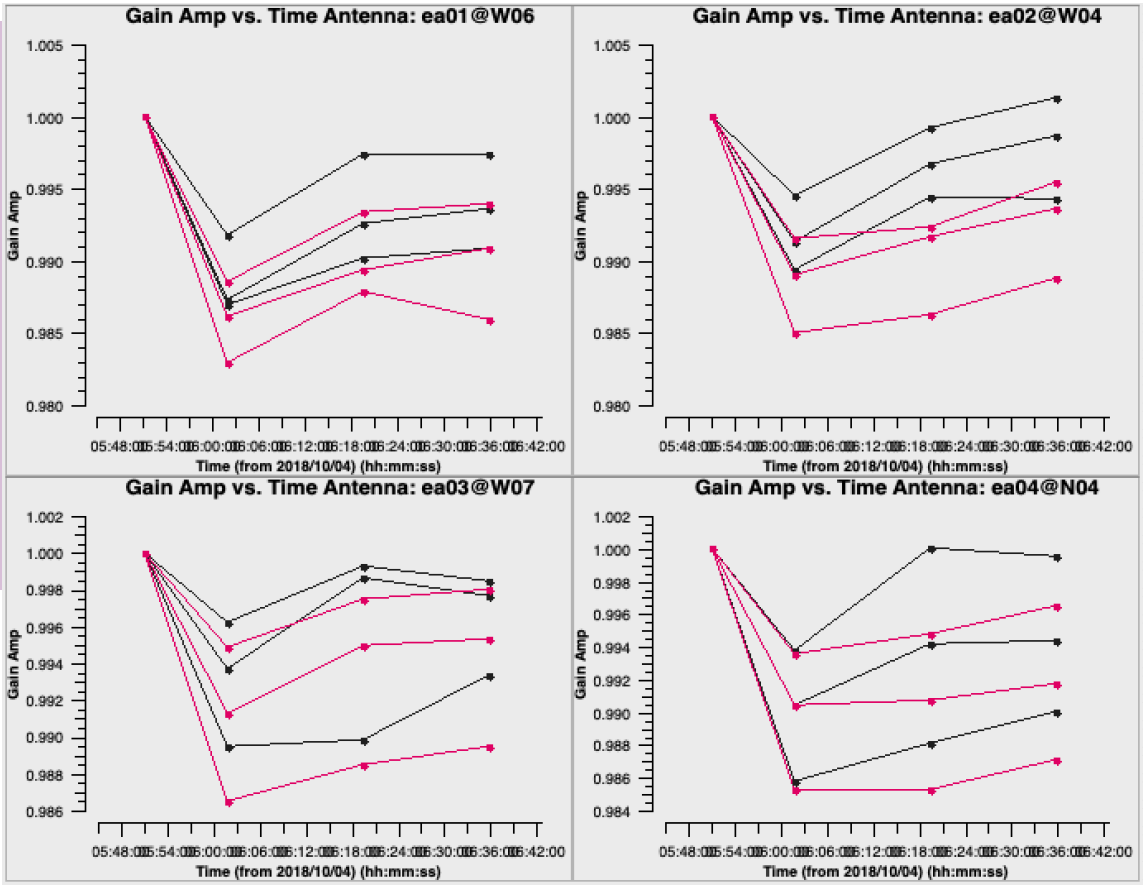


ON YOUR OWN

Examine complex gain cal *amp* solutions

- CASA task *plotms*

```
# CASA parameters for plotms
vis = 'scanphase.gcal'
gridrows = 2
gridcols = 2
xaxis = 'time'
yaxis = 'amp'
iteraxis = 'antenna'
coloraxis = 'corr'
xconnector = 'line'
plotrange = []
```



Data point with value of 1 is the flux calibrator (3C48): in this observation, bandpass cal and flux cal are the same source. (This is common).

ON YOUR OWN

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 tomorrow**
- 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 the `fluxscale` task, those gains are used to determine flux densities 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 purely a matter of personal preference.

Bootstrap the flux density, fit spectrum

- CASA task *fluxscale*

```
# CASA parameters for fluxscale
vis = 'my_data.ms'
caltable = 'scanphase.gcal'
fluxtable = 'fluxscale.cal'
reference = '0137+331=3C48'
transfer = ['J0259+0747']
fitorder = 1
incremental = False
```

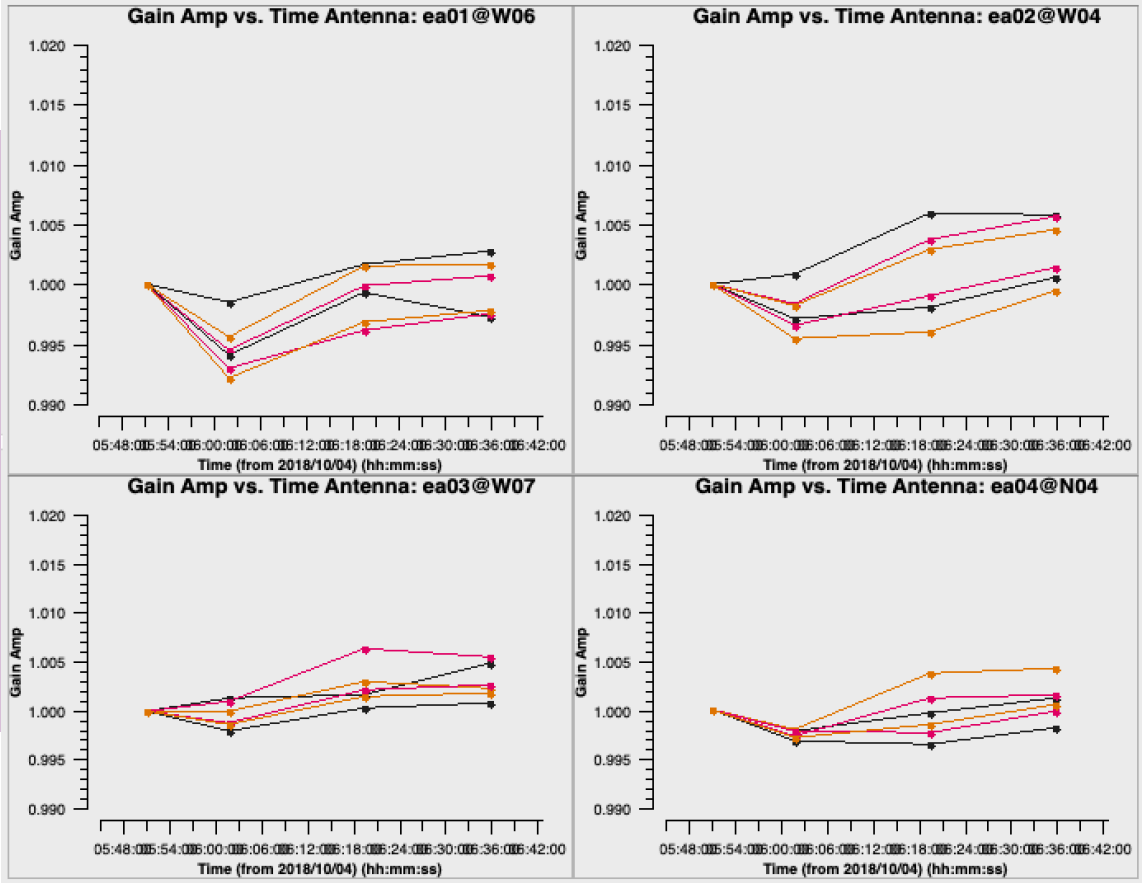
Results reported in casalogger:

```
Flux density for J0259+0747 in SpW=0 (freq=3.063e+09 Hz) is: 0.977184 +/- 0.000951237 (SNR = 1027.28, N = 54)
Flux density for J0259+0747 in SpW=1 (freq=3.191e+09 Hz) is: 0.985051 +/- 0.000940935 (SNR = 1046.89, N = 54)
Flux density for J0259+0747 in SpW=2 (freq=3.319e+09 Hz) is: 0.992086 +/- 0.000997213 (SNR = 994.859, N = 54)
Fitted spectrum for J0259+0747 with fitorder=1: Flux density = 0.98476 +/- 0.000140244 (freq=3.18929 GHz) spidx:
Storing result in fluxscale.cal
```

Examine rescaled amplitude solutions

- CASA task *plotms*

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



ON YOUR OWN

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 approach is to use one run of the *applycal* task for each calibrator

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`)

```
# CASA parameters for applycal: apply calibration to bandpass/flux cal
vis = 'my_data.ms'
field = '0137+331=3C48'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', '0137+331=3C48']
interp = ['', '', '', '', 'nearest']
calwt = False
```

ON YOUR OWN

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...

```
# CASA parameters for applycal: apply calibration to gain calibrator
vis = 'my_data.ms'
field = 'J0259+0747'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', 'J0259+0747']
interp = ['', '', '', '', 'nearest']
calwt = False
```

ON YOUR OWN

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

```
# CASA parameters for applycal: apply calibration to target
vis = 'my_data.ms'
field = '3C75'
gaintable = ['antpos.cal', 'tecim.cal', 'delays.cal',
             'bandpass.cal', 'fluxscale.cal']
gainfield = ['', '', '', '', 'J0259+0747']
interp = ['', '', '', '', 'linear']
calwt = False
```

ON YOUR OWN

Split target source(s) into their own MS

- CASA task *mstransform*

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

```
# CASA parameters for mstransform or split
vis = 'my_data.ms'
outputvis = '3C75.ms'
field = '3C75'
correlation = 'RR,LL'
datacolumn = 'corrected'
```

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

ON YOUR OWN

Re-weight the visibilities

- CASA task *statwt*

```
# CASA parameters for statwt
vis = '3C75.ms'
datacolumn = 'data'
```

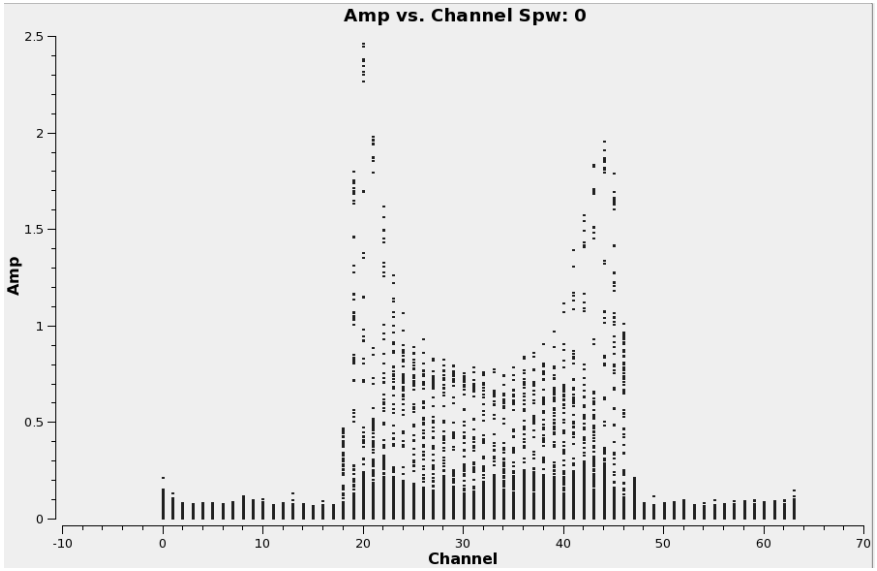
- Data weights are initialized to be based on channel bandwidth and integration-time ($2\Delta\nu\Delta\tau$).
- *statwt* re-weights the visibilities according to their scatter:
  - down-weight underperforming antennas
  - down-weight frequency ranges affected by RFI
- Use on *fully-calibrated* data!

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

ON YOUR OWN

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: *mstransform*

- 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

- However, if *self-calibrating* on a strong/narrow spectral line, must first correct in the visibilities *before* running *tclean*/*gaincal*:
  - use *mstransform* to Doppler-correct multiple MSs to same frame
  - run *tclean*/*gaincal* as usual
    - Preshanth Jagannathan talk this afternoon (imaging)
    - Josh Marvil talk tomorrow (self-calibration)

The VLA Calibration Pipeline

- Performs basic flagging and calibration using CASA.
- Primarily designed for Stokes `I` continuum data.
- To run successfully, the scan intents in the observation scheduling block *must* be set correctly.
- Information are at:

https://science.nrao.edu/facilities/vla/data-processing/pipeline

- More details later ⟵ **John Tobin's talk tomorrow**

Science-Ready Data Products (SRDP)

Now available! (for some observing modes)

- Provide calibrated (and eventually imaged) PI observations

  ⇾ users can focus on their science!
- Intended ultimately for all NRAO instruments

- SRDP project will cover increasing functionality over the next several years. Currently available for:
  - C-band and higher frequencies (≥ 4 GHz)
  - continuum science only

Observing setups *must* follow all VLA observing guidelines and be pipeline compatible (e.g. scan intents, calibrator cycles)

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.