ALMA Laboratory

CASA Introduction

The simple dataset ('calibrator.ms') contains three sources (Neptune, J1924-292, J1625-254) observed at GHz, in one spw. They have been already calibrated. In the column DATA you find the raw data, in CORRECTED the data already calibrated, and in MODEL the model used for the calibration.

In the following, we provide a list of tasks and interesting plots for data inspection.

- **listobs** You can simply run it writing:

  ```
  listobs('calibrators.ms')
  ```

  and the output will be written in the log. Or you can write the output in a file:

  ```
  listobs('calibrators.ms',listfile='prova.listobs')
  ```

- **plotants**

  You can simply run it writing:

  ```
  plotants('calibrators.ms')
  ```

  and the plot will be dispayed on the screen. Or you can save the plot in a file:

  ```
  listobs('calibrators.ms',figfile='plotants.png')
  ```

- **plotms**

  You can use it interactively by typing

  plotms

  and interactively changing the parameters, or you can give all the inputs:

  ```
  plotms(vis='calibrators.ms',xaxis='u', yaxis='v', field='', spw='0',
         ydatacolumn='', coloraxis='field', antenna='', correlation='',
         avgtime='', avgscan=F, avgchannel='')
  ```

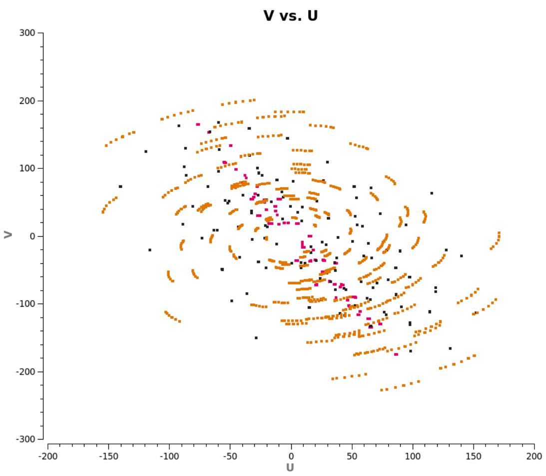

Figure 1: **UV coverage**

```
plotms(vis='calibrators.ms',xaxis='time', yaxis='amp', field='', spw='0',
       ydatacolumn='data', coloraxis='field', antenna='', correlation='',
       avgtime='', avgscan=F, avgchannel='')
```

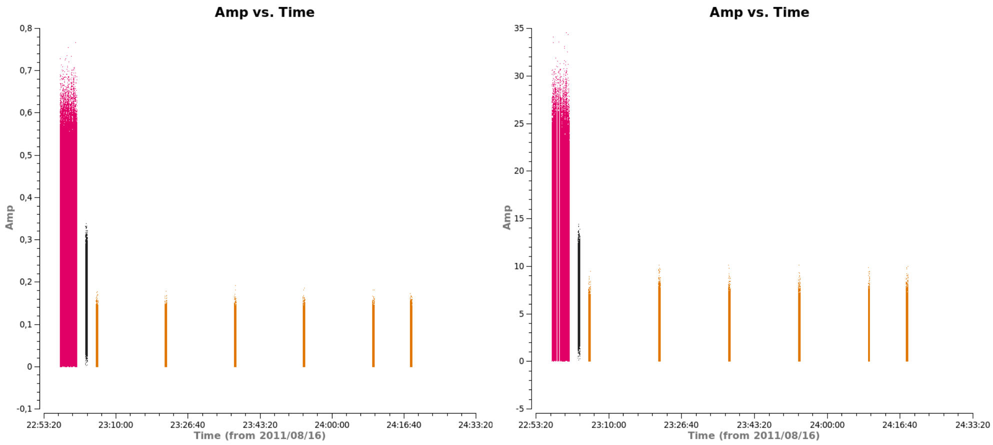

Figure 2: **Amp vs time** of raw data (right panel), and corrected data (left panel) of all observed sources, in different colours. The scientific target, which is not included in this dataset was observed during the large gaps between orange scans.

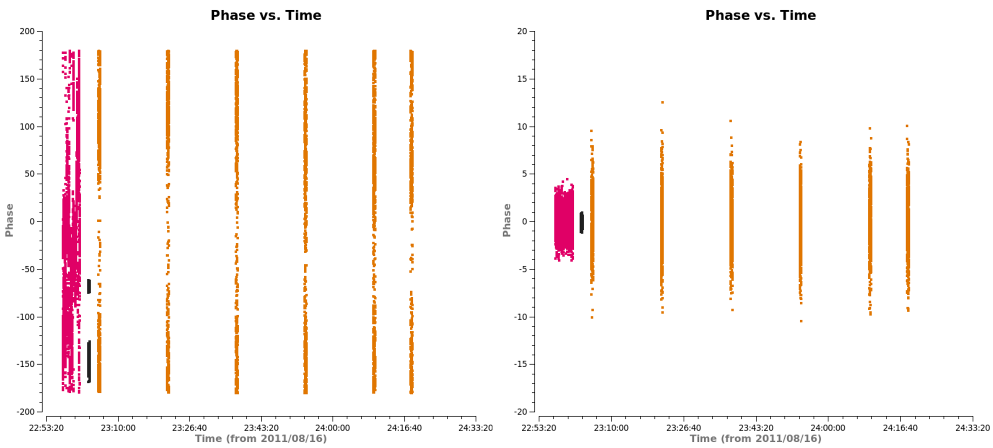

Figure 3: **Phase vs time** of raw data (right panel), and corrected data (left panel) of all observed sources, in different colours. Data are averaged by 10 channels.

Plot data of Neptune:

```
plotms(vis='calibrators.ms', xaxis='uvwave', yaxis='amp', field='Neptune',
        spw='0', ydatacolumn='data',
        coloraxis='baseline', antenna='', correlation='', avgtime='1e8',
        plotfile='provaplotms.png')
```

by changing the xaxis, yaxis, and ydatacolumn parameters you get the plots shown in fig 1,2,3.

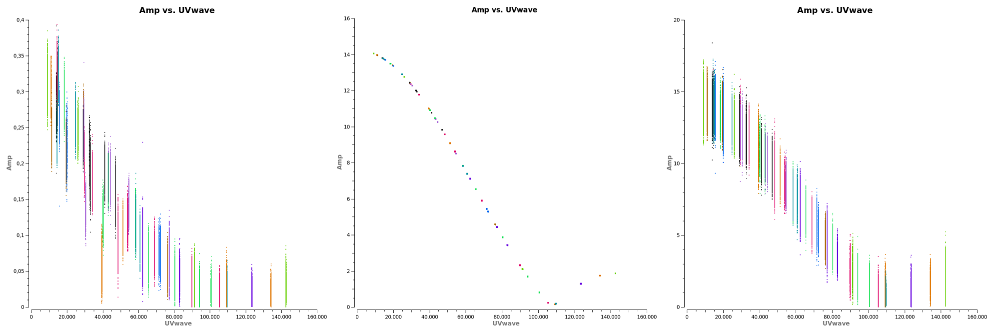

Figure 4: **Amplitude vs uvdistance (in wavelength)** of raw data (right panel), model (central panel) and corrected data (left panel) of Neptune

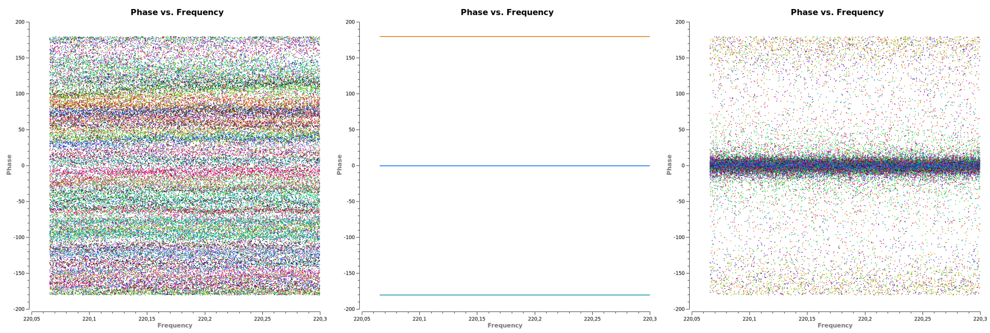

Figure 5: **Phase vs frequency (in wavelength)** of raw data (right panel), model (central panel) and corrected data (left panel) of Neptune

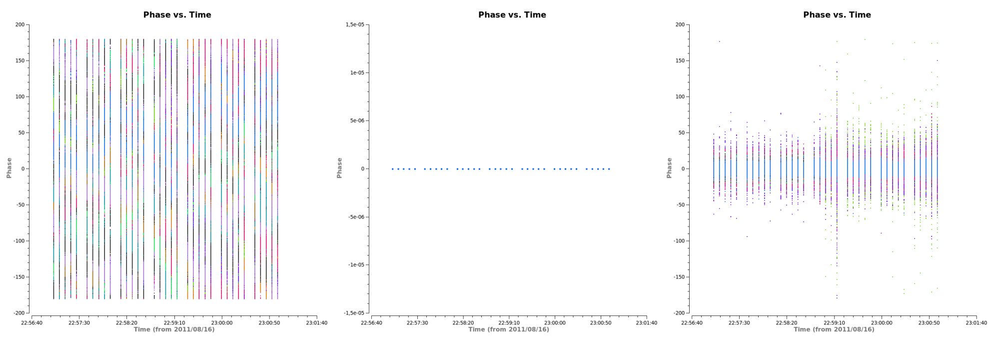

Figure 6: **Phase vs time** of raw data (right panel), model (central panel) and corrected data (left panel) of Neptune

Plot data of J1625-254:

```
plotms(vis='calibrators.ms',xaxis='imag', yaxis='real', field='J16*', spw='0',
       ydatacolumn='data', coloraxis='scan', antenna='', correlation='',
       avgtime='1e8', avgscan=F, avgchannel='1000')
```

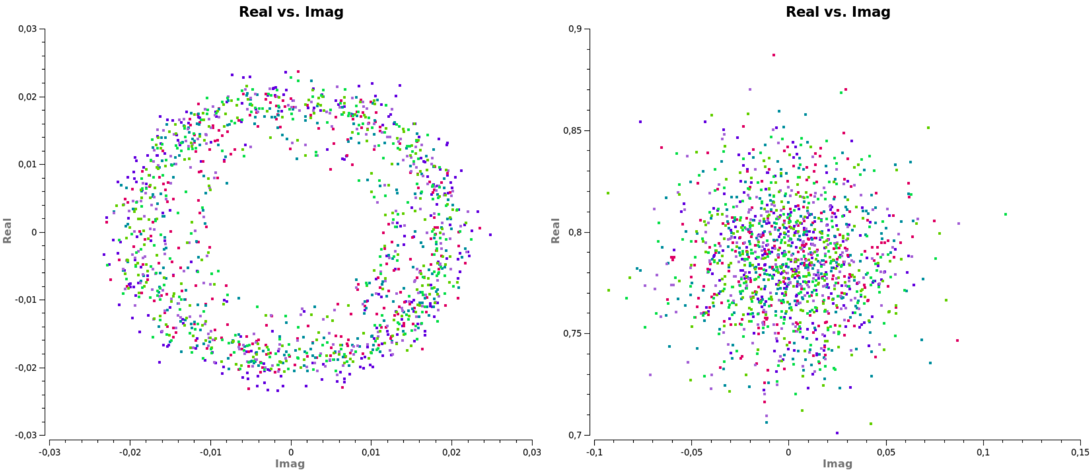

Figure 7: **Imag vs real** of raw data (right panel), model (central panel) and corrected data (left panel) of calibrator J1625

These plots show the real versus imaginary part of visibilities from the calibrator J1625-254, before and after the calibration. You can notice in the left plot that each scan describes a circle in the plane, centered around zero. The radius of this circle represents the amplitude of raw data. The calibrator is supposed to be a point source. For a point source the real part should give the point flux density, and the imaginary part should be zero. That's why in the right plot the visibilities concentrate in a region centered on zero in imag axes, and on the value of the amplitude in the real axes.

October 25

Tsys calibration

Dataset: **uid_A002_X1d54a1_X174.ms**
Tsys table: **cal-tsys_uid__A002_X1d54a1_X174.calnew**
Task **plotcal** to plot the Tsys table, antenna per antenna:

```
plotcal(caltable='cal-tsys_uid___A002_X1d54a1_X174.calnew', xaxis='chan', yaxis='amp',
                spw='7', subplot=111, overplot=False,
                plotrange=[0, 0, 0, 250], plotsymbol='.',iteration='antenna')
```

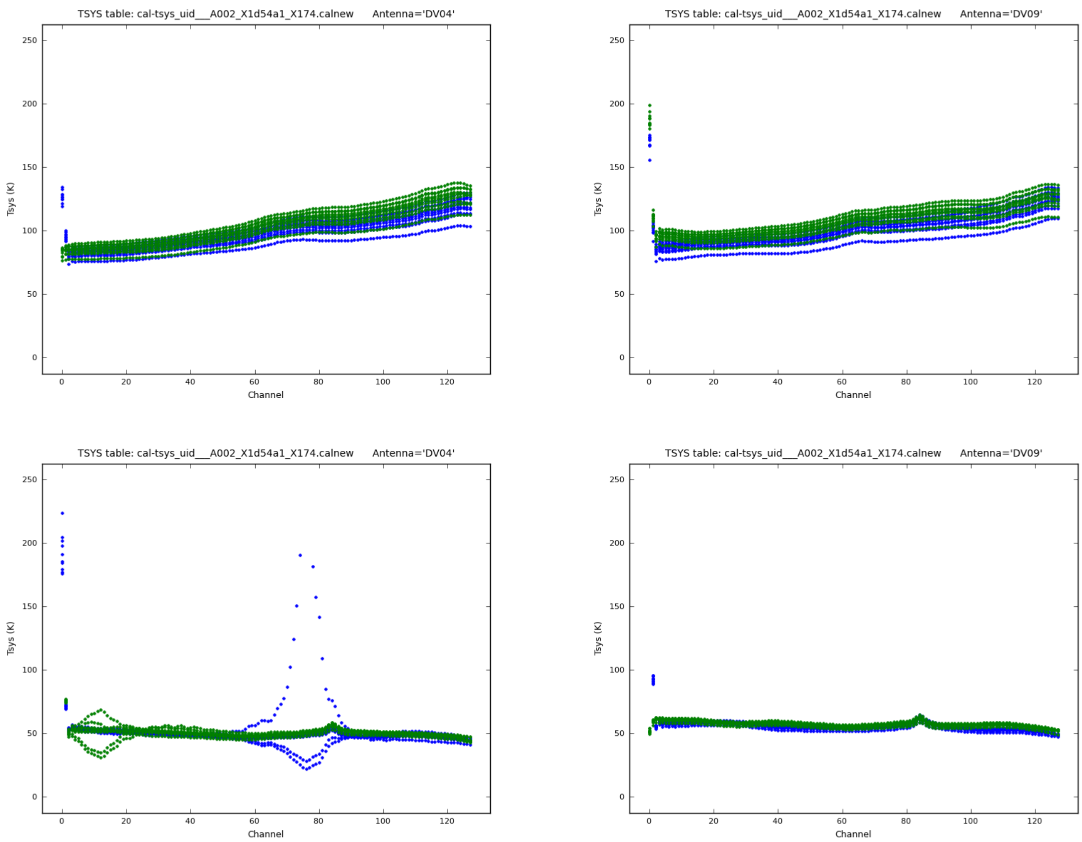

Figure 8: **Tsys gains vs channel** of antenna DV04 (right panel) and DV09 (left panel), in spw 1 (upper panels) and spw7 (lower panels)

Comparing the different spw is possible to notice an outlying feature in spectral window 7, antenna DV04. The feature can be identified using the locate button, and corresponds to the scan 8. This means that something wrong happend in the atmosphere during that period of time. To check the effect of this on the data, apply the solution to all sources, including the scientific target.
Task **applycal**

```
for field in ['Titan','1037*','NGC*']:
        applycal(vis='uid__A002_X1d54a1_X174.ms', spw='1,3,5,7', flagbackup=F, field=field,
        gainfield=field,interp='linear,spline',
        gaintable=['cal-tsys_uid__A002_X1d54a1_X174.calnew'])
```

Now the dataset has two columns (DATA and CORRECTED). Using plotms we can focus on spw 7:

```
plotms(vis='uid__A002_X1d54a1_X174.ms', spw='7:20~100', field='0', antenna='!DV07',
        xaxis='freq', yaxis='amp', coloraxis='baseline', avgtime='1e8')
```

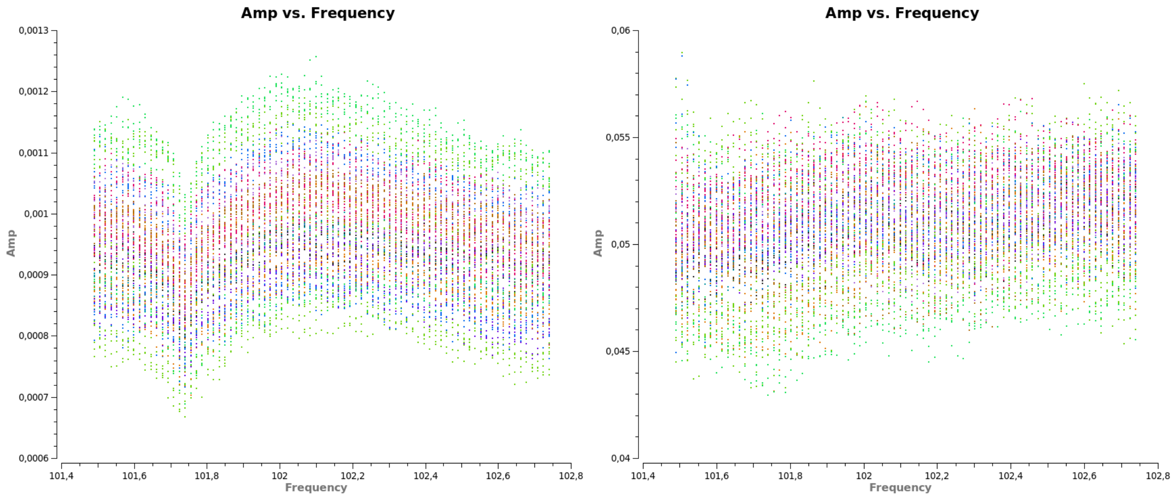

Figure 9: **Amp vs frequency** in spw 7 of data (right panel) and corrected (left panel) of field 0. We excluded antenna DV07 from the plotting. This antenna will be flagged in the following inspection.

It is possible to notice the atmospheric Ozone transition line at 101.7 GHz in the right panel. In the left panel the line disappears, due to the Tsys calibration. The same line is also visible in the lower-right panel of Fig. 7 around channel 70.

The issue with antenna DV04, in scan 8, can affect the scan 9 observation of the scientific target. Figure 9 shows the data of antenna DV04 in scan 9, before (right panel) and after (left panel) the application of the Tsys calibration.

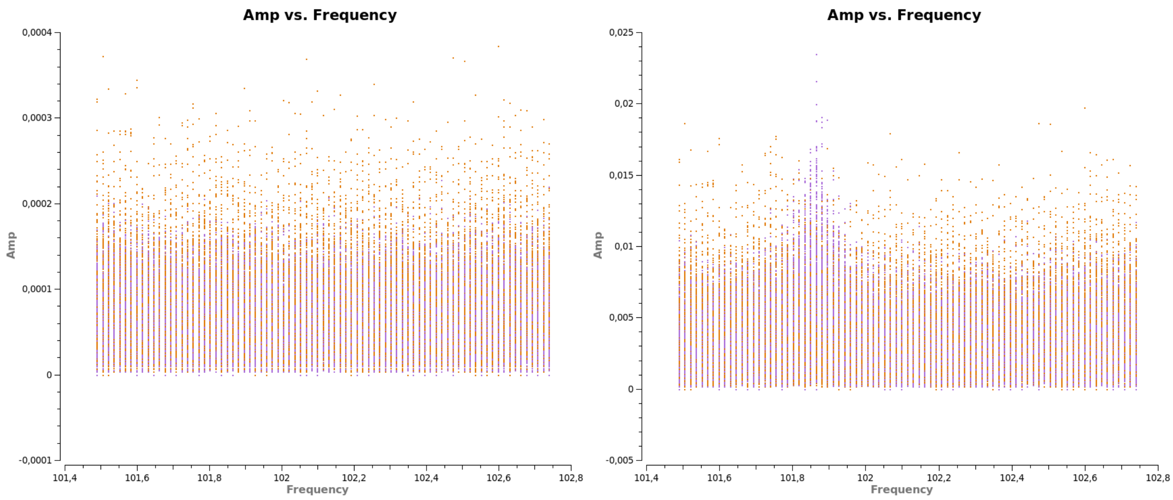

Figure 10: **Amp vs frequency** in scan 9, spw 7, of data (right panel) and corrected data (left panel) for antenna DV04

It is evident, that the application of the Tsys table for antenna DV04 produces in the corrected data a spectral feature, mimicking an emission, in correspondence of one of the correlation. Obviously, the introduction of non real spectral lines must be avoided, since it can lead to a misinterpretation of the results. DV04 in scan 9, must be flagged from the data, since it is not possible to properly model the atmospheric behaviour. A less evident effect is produced in scan 5, where the Tsys correction is taken from scan 3, where DV04 also presents problems.

Water Vapour Radiometer (WVR) calibration

With the aim of correcting for the shot-term variation in the atmosphere, the mm antennas mount a radiometer recording, typically for each second of observation, the atmosphere H_2O 183GHz line emission. This is a direct measure of the water vapor amount in front of the antenna. Here we stress that only the gas component of the water content produces a path length variation, directly linked to a signal phase variation. This variation depends on the wvr amount, on the frequency of observation according to:

$$\Delta\phi_e \sim 12.6\pi\Delta Wvr/\lambda \tag{1}$$

In the ALMA data this information is stored in spw 0. If we make a plot phase vs. time of a calibrated dataset comparing the impact of wvr correction we can roughly say that the the data dispersion is reduced when the wvr correction is applied.

```
plotms(vis='calibrated_wvr_on.ms',xdatacolumn='corrected',xaxis= 'time',
       ydatacolumn= 'corrected',yaxis= 'phase',avgchannel= '2000',
       avgtime= '20',plotrange=[0,0,-10, 10],field= '0',spw='1:20~100',
       coloraxis='baseline')

plotms(vis='calibrated_wvr_off.ms',xdatacolumn='corrected',xaxis= 'time',
       ydatacolumn= 'corrected',yaxis= 'phase',avgchannel= '2000',
       avgtime= '20',plotrange=[0,0,-10, 10],field= '0',spw='1:20~100',
       coloraxis='baseline')
```

It is possible to quantify this effect by using the task "visstat", that displays on the logger the statistical information related to the specified visibilities. Pay particular attention to rms value: in "wvr_on.ms" is lower than in "wvr_off.ms".

```
visstat(vis="wvr_on.ms",axis="phase",datacolumn="corrected",useflags=True,spw="1:20~100",field="0",
selectdata=True,antenna="",uvrange="",timerange="",correlation="",scan="",array="",observation="")

visstat(vis="wvr_off.ms",axis="phase",datacolumn="corrected",useflags=True,spw="1:20~100",field="0",
selectdata=True,antenna="",uvrange="",timerange="",correlation="",scan="",array="",observation="")
```

This effect has in impact also on the amplitude values. According to:

$$V = V_0 \times exp^{\phi_{rms}^2/2} \tag{2}$$

we expect an improvement in the flux estimation.

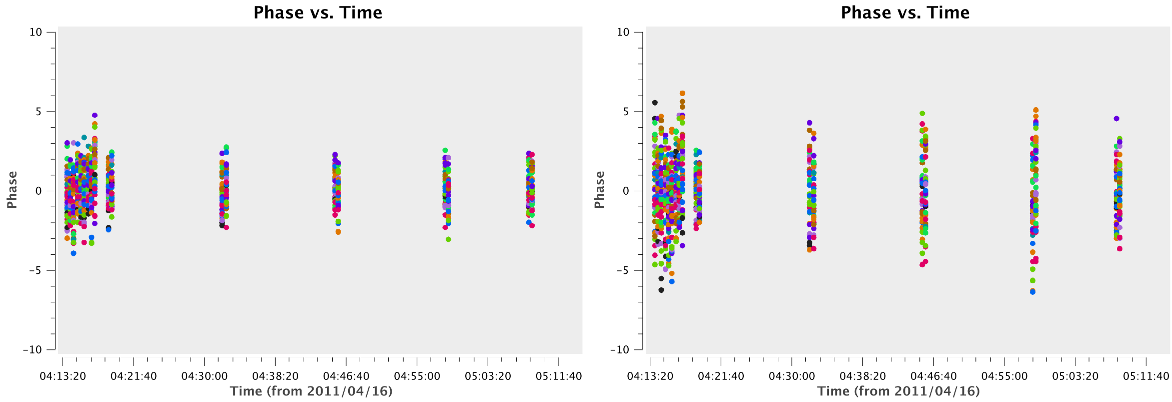

Figure 11: **Phase vs time** in spw 1. The spread is reduced when wvr correction is on (left panel). Different colors per different baselines.

NGC3256 data reduction

Usually ALMA datasets consist of several sets of observations contained in different files .ms. This is because every ALMA observing project is broken down into scheduling blocks, to facilitate the observation of projects in the required weather conditions (dynamic scheduling). This SV dataset is made of 6 datasets, which need to be combined. Before the combination, some preliminary steps must be followed on each dataset (preliminary flags, and application of Tsys and WVR tables, provided by the observatory). It is convenient to write scripts to loop over the different datasets:

```
basename=['X1d54a1_X174_tutorial','X1d54a1_X5_tutorial', 'X1d54a1_X2e3_tutorial',
'X1d5a20_X174_tutorial', 'X1d5a20_X330_tutorial', 'X1d5a20_X5_tutorial']

for name in basename:
    os.system('ls -d '+name+'.ms')
```

Preliminary flags

- Shadow: depending on the elevation of the observed source, in compact configuration it can happen that some antennas cast their shadow on the closest one, so reducing their effective area. It is possible to correct for this effect by using the task flagdata in 'shadow' mode.
- Autocorrelation: the combination of the signals of each antenna with itself are registered in the dataset, but are not needed once the Tsys and WVR tables are computed. They can be flagged since we are interested in cross-correlation.
- Pointing: several scans are used to check the precision on the pointing. These scans are not used in the offline calibration, so we can flag them.
- Atmosphere: for the same reason we can flag the scans used for the atmospere calibration.

```
for name in basename:
        flagdata(vis=name+'.ms', flagbackup = F, mode = 'shadow')
        flagdata(vis=name+'.ms',mode='manual', autocorr=True)
        flagdata(vis=name+'.ms', mode='manual', flagbackup = F, intent='*POINTING*')
        flagdata(vis=name+'.ms', mode='manual', flagbackup = F, intent='*ATMOSPHERE*')
        flagmanager(vis = name+'.ms', mode = 'save', versionname = 'Apriori')
```

The last task used 'flagmanager' saves the informations about the performed flags so if needed it is possible to restore the data before those saved flags.

As noted in the Tsys section, in one of the dataset it is possible to notice that the antenna DV04 has problem in the scans 5 and 8. We flag this antenna from the closest scans in order to avoid introducing errors in the scientific data.

```
flagdata(vis='X1d54a1_X174_tutorial.ms', mode='manual',
         antenna='DV04', flagbackup = T, scan='4,5,9', spw='7')
```

A priori calibrations and data split

We apply WVR and Tsys calibration, provided with the raw data. To apply Tsys, in particular, we need to specify the field from which we want to take the solutions from the gaintable. That's why we specify the gainfield parameter. The interpolation option is set to 'linear' in time, since the Tsys changes smoothly with time, and it is recommended 'spline' in frequency.

```
for name in basename:
for field in ['Titan','1037*','NGC*']:
applycal(vis=name+'_WVR.ms', spw='1,3,5,7', flagbackup=F, field=field, gainfield=['',field],
interp=['nearest','linear,spline'], gaintable=['cal-'+name+'.Wnew','cal-tsys_'+name+'.calnew'])
```

At this point we can save the corrected data, isolating just those we are interested in (spw:1,3,5,7) producing new files '_line.ms':

```
for name in basename:
os.system('rm -rf '+name+'_line.ms*')
split(vis=name+'.ms', outputvis=name+'_line.ms',
datacolumn='corrected', spw='1,3,5,7')
```

And finally combine the datasets:

```
comvis=[]
for name in basename:
        comvis.append(name+'_line.ms')

os.system('rm -rf ngc3256_line.ms*')
concat(vis=comvis, concatvis='ngc3256_line.ms')
```

the first part of this script simply define a string contaning the names of the datasets to combine, which will be the input parameter vis in the CASA task 'concat'.

Data inspection and editing

First of all we inspect the content of the file:

```
listobs(vis='ngc3256_line.ms', listfile='ngc3256_line.listobs' )
```

A careful inspection of the data is needed to identify possibly 'wrong' data, to be removed before the calibration. It is useful to look at the phase calibrator data, where it easier to identify outliers in time or frequency.

Edge channels have always higher amplitude than the central channels of the bandpass, so we flag them in all spw (see an example in left panel of Figure 12).

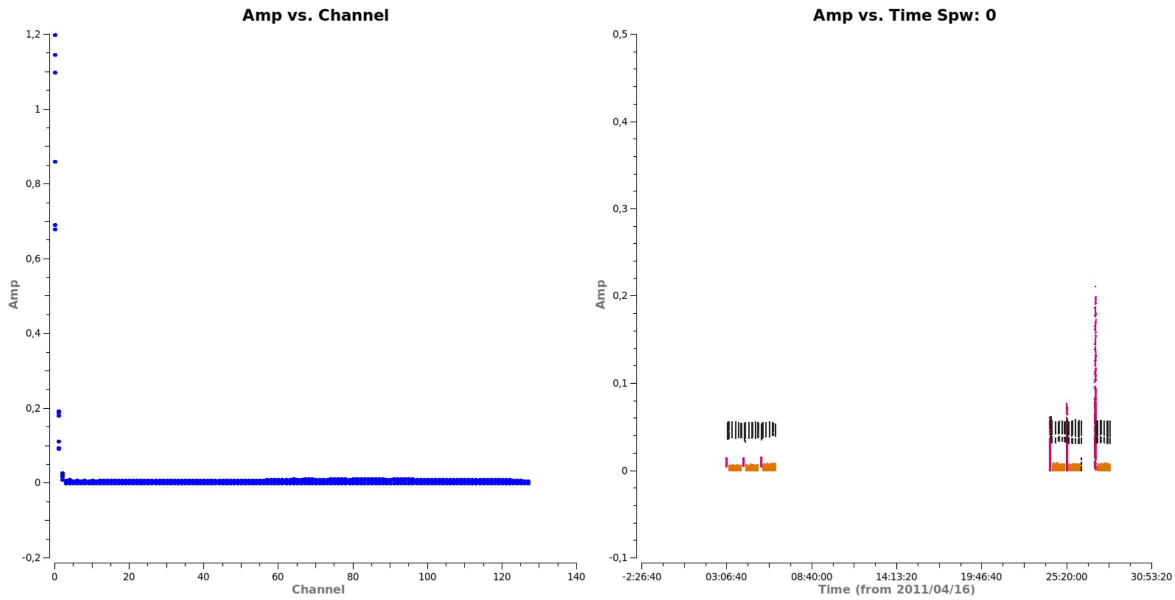

Figure 12: Left panel: amp vs time of the source 1037*, to notice the edge channels. Right panel: amp vs time of all field, to notice Titan higher amplitude during the second day observations.

By plotting amplitude versus time for all the fields:

```
plotms(vis='ngc3256_line.ms', xaxis='time', yaxis='amp',
       averagedata=T, avgchannel='128', coloraxis='field',
       iteraxis='spw')
```

we can notice that Titan during the second day has high amplitude. This is due to the proximity of Saturn rings, which enter the field of view of ALMA. We flag Titan data of the second day, since the amplitude calibration including these data would be wrong.

```
flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='*:0~16,*:125~127')
flagdata(vis = 'ngc3256_line.ms', flagbackup = T,
        timerange='>2011/04/16/12:00:00', field='Titan')
```

In figure 13 the plots used to identify some of the data to be flagged, are shown. These are the necessary flags:

```
flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='3',
         correlation='YY', mode='manual',
         antenna='DV07', timerange='')

flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='3',
         correlation='YY', mode='manual',
         antenna='DV08', timerange='>2011/04/17/03:00:00')

flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='',
         correlation='', mode='manual',
         antenna='PM03&DV10', timerange='>2011/04/16/15:00:00')


flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='0',
         correlation='', mode='manual',
         antenna='PM03', timerange='2011/04/17/02:15:00~02:15:50')

flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='2,3',
         correlation='', mode='manual',
         antenna='PM03', timerange='2011/04/16/04:13:50~04:18:00')

flagdata(vis='ngc3256_line.ms', flagbackup=T, spw='0',
         correlation='', mode='manual',
         antenna='PM03', timerange='2011/04/16/04:13:50~04:18:00')
```

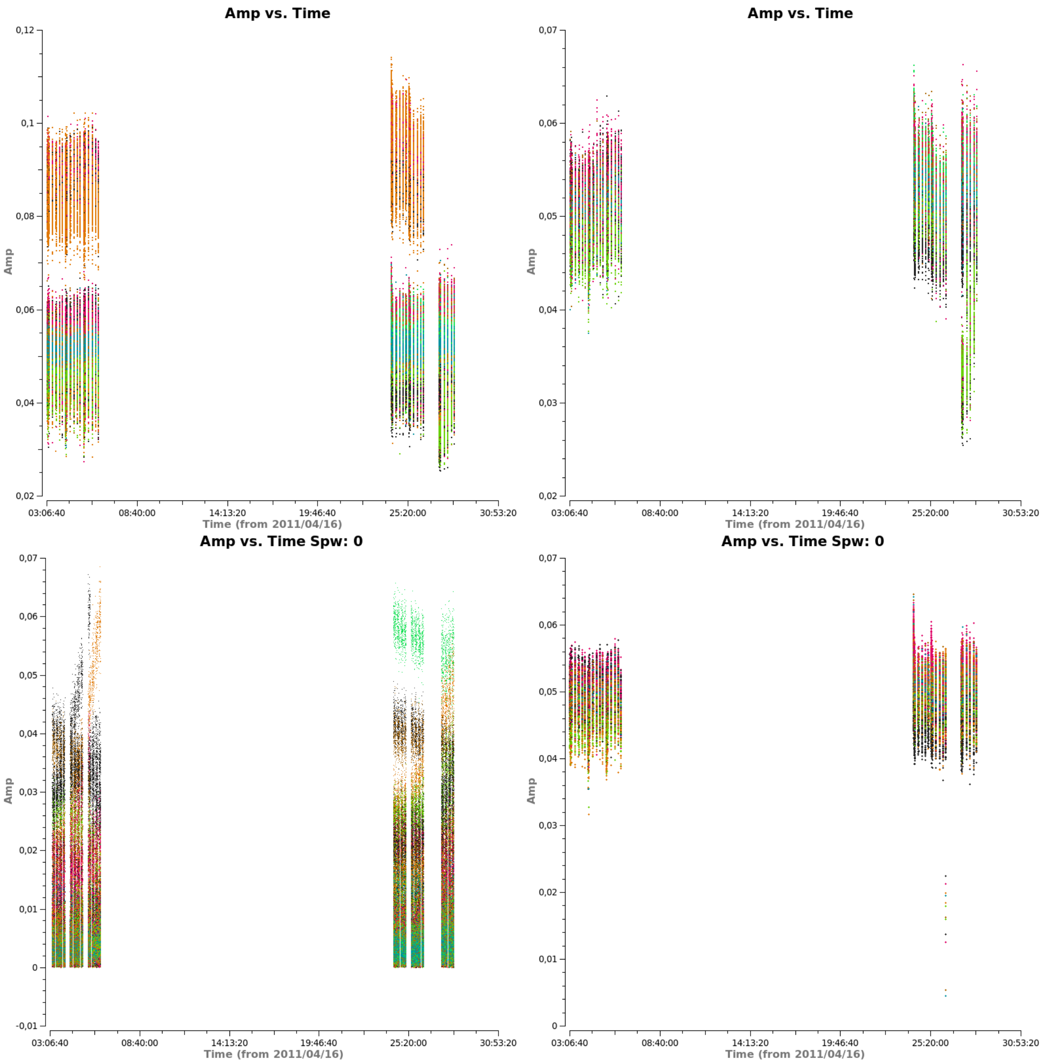

Figure 13: Upper panel. Left: identify DV07 in spw 3; right: DV08 in spw 3 at ¿03:00:00. Lower panel. Left: baseline PM03&DV10 (green points); right: PM03 in a timerange

Calibration

- **Modeling the calibrator**

  The first step is to set the flux density for Titan using the task setjy. We use the Butler-JPL-Horizons 2012 model:

```
setjy(vis='ngc3256_line.ms', field='Titan', standard='Butler-JPL-Horizons 2012',
spw='0,1,2,3', usescratch=False)
```

  Setjy task fills the "MODEL" column. It is nice to verify with plotms the calibrator's model. When the calibration step is over, if the calibration curves are properly applied to the calibrator itself, "CORRECTED" and "MODEL" columns should be similar (See **??**).

- **Short-time phase gains**

  We run gaincal on the bandpass calibrator to determine phase-only gain solutions. We use solint='int' for the solution interval, which means that one gain solution will be determined for every integration time. This short integration time is possible because the bandpass calibrator is a very bright point source, so we have very high signal-to-noise and a perfect model. This will correct for any phase variations in the bandpass calibrator as a function of time, a step which will prevent decorrelation of the vector-averaged bandpass solutions. These solutions are applied on-the-fly when we run bandpass.

```
gaincal(vis='ngc3256_line.ms', caltable='cal-ngc3256.G1', spw='*:40~80', field='1037*',
        selectdata=T, solint='int', refant='DV04', calmode='p')
```

- **Bandpass**

  When the variation in phase (vs. time) for the bandpass calibrator are calculated, we can determine the bandpass solutions with bandpass. We apply the phase calibration table on-the-fly with the parameter "gaintable". Now that the phases are corrected, the data can be time-averaged over longer intervals to maximise SNR in each individual channel. This is possibile because the frequency dependent gain variation, due to instrumental effect in most of the cases, are very stable in time, therefore they can be considered time-independent (with the caveat that no modification in the telescope spectral setup has been applied!!!). We determine bandpass solutions for both days separately. We first calculate the bandpass solution for the first day. In order to achieve that only a single solution is created for the day, we set parameter "combine" to 'scan,obs' meaning that combination of solutions should neither halt at scan nor at observation boundaries. The message "Insufficient unflagged antennas" is related to the flagged edge channels.

```
bandpass(vis = 'ngc3256_line.ms', caltable = 'cal-ngc3256.B1',
        gaintable = 'cal-ngc3256.G1', timerange='<2011/04/16/15:00:00',
field = '1037*', minblperant=3, minsnr=2, solint='inf', combine='scan,obs',
bandtype='B', fillgaps=1, refant = 'DV04', solnorm = T)
```

```
bandpass(vis = 'ngc3256_line.ms', caltable = 'cal-ngc3256.B1',
        gaintable = 'cal-ngc3256.G1', timerange='>2011/04/16/15:00:00',
field = '1037*', minblperant=3, minsnr=2, solint='inf', combine='scan,obs',
bandtype='B', fillgaps=1, refant = 'DV04', solnorm = T, append=True)
```

- **Phase and amplitude gains** Next step is to correct for time dependent gain variation. These variation are frequency independent, thus we can average over the frequency to obtain a more statistically significant solution.Therefore we have to apply (this time) the bandpass calibration solutions on-the-fly (specified in gaintable input parameter). We solve for amplitude and phase simultaneously and determine average solutions per scan:

```
gaincal(vis = 'ngc3256_line.ms', caltable = 'cal-ngc3256.G2', spw =
'*:16~112', field = '1037*,Titan', minsnr=1.0,
solint= 'inf', selectdata=T, solnorm=False, refant = 'DV04',
gaintable = 'cal-ngc3256.B1', calmode = 'ap')
```

- **Fluxscale**

  Finally, we will bootstrap the flux density of the secondary calibrator from that of Titan using the task fluxscale. The new flux table cal-ngc3256.G2.flux replaces the previous cal-ngc3256.G2 table in future application of the calibration to the data, i.e. the new flux table contains both cal-ngc3256.G2 and the newly acquired flux scaling. Unlike the gain calibration steps, this is not an incremental table.

  ```
  fluxscale( vis="ngc3256_line.ms", caltable="cal-ngc3256.G2",fluxtable="cal-ngc3256.G2.flux",
  reference="Titan",transfer="1037*", refspwmap=[0,1,2,3])
  ```

To check whether or not, the application procedure was correct is a good practice to apply the calibration solutions to the calibrator itself. Thus, we use applycal to apply the bandpass and gaincal tables that we have generated in the previous sections. Here we apply calibration corrections both to scientific target and calibrator.

```
applycal( vis='ngc3256_line.ms', flagbackup=F, field='NGC*,1037*',
interp=['linear','nearest'], gainfield = ['1037*', '1037*'],
gaintable=['cal-ngc3256.G2.flux', 'cal-ngc3256.B1'])
```

At the end, we use plotms to verify the calibration (and application to data) procedure was ok. We plot amp, phase vs time, amp, phase vs. frequency. We stress that for a point-like source amplitude should be flat (and without any ripples) and phase flat and centered at 0.

```
plotms(vis='ngc3256_line.ms', xaxis='freq', yaxis='phase', selectdata=True,
        field='1037*', avgtime='1E6', avgscan=T, coloraxis='baseline')
```

```
plotms(vis='ngc3256_line.ms', xaxis='time', yaxis='phase', selectdata=True, spw='0',
coloraxis='baseline',field='1037*', avgchannel='100', avgscan=T, iteraxis='antenna',
timerange='<2011/04/16/08:00:00')
```

The first plotms plots the phases vs frequency for the different baselines, in the different spws, averaging all scans (avgtime='1e6', avgscan=T). While the second plots the phase vs time, in spw 0, for each antenna only for the first day observation.