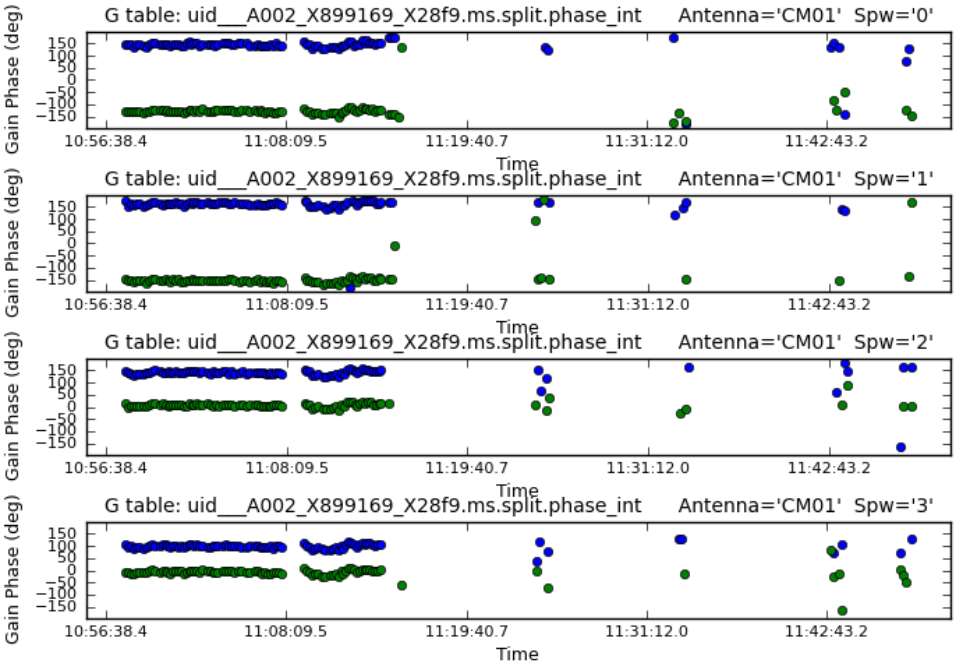
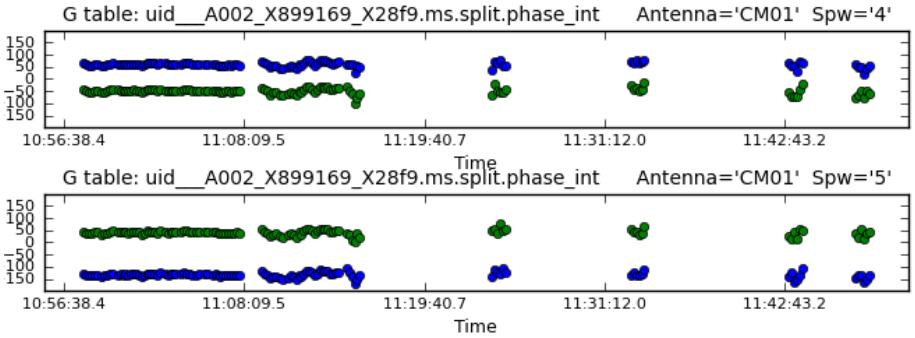
Phase transfer

In the case of a weak phase calibrator and small spws, it could happen that the phase calibration fails. This is because the s/n in the phase_int solutions is low and then, when applying to obtain the amplitude caltable the data is flagged and cannot compute solutions...

If we're lucky and have also a broad spw, where phase_int solutions are okay, we could use that one to transfer the phase to the small spws.





Usual gain calibration

```
# Gain calibration
  os.system('rm -rf uid___A002_X899169_X28f9.ms.split.phase_int')
  gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
    caltable = 'uid___A002_X899169_X28f9.ms.split.phase_int',
    field = '0~2', # J0510+1800,J0510+180,J0604+2429
    solint = 'int',
    refant = 'CM06',
    gaintype = 'G',
    calmode = 'p',
    gaintable = 'uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch')

  os.system('rm -rf uid___A002_X899169_X28f9.ms.split.ampli_inf')
  gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
    caltable = 'uid___A002_X899169_X28f9.ms.split.ampli_inf',
    field = '0~2', # J0510+1800,J0510+180,J0604+2429
    solint = 'inf',
    refant = 'CM06',
    gaintype = 'T',
    calmode = 'a',
    gaintable = ['uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch', 'uid___A002_X899169_X28f9.ms.split.phase_int'])

  fluxscaleDict = fluxscale(vis = 'uid___A002_X899169_X28f9.ms.split',
    caltable = 'uid___A002_X899169_X28f9.ms.split.ampli_inf',
    fluxtable = 'uid___A002_X899169_X28f9.ms.split.flux_inf',
    reference = '1') # J0510+180

  os.system('rm -rf uid___A002_X899169_X28f9.ms.split.phase_inf')
  gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
    caltable = 'uid___A002_X899169_X28f9.ms.split.phase_inf',
    field = '0~2', # J0510+1800,J0510+180,J0604+2429
    solint = 'inf',
    refant = 'CM06',
    gaintype = 'G',
    calmode = 'p',
    gaintable = 'uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch')
```

Modifications for phase transfer (in phase cal step)

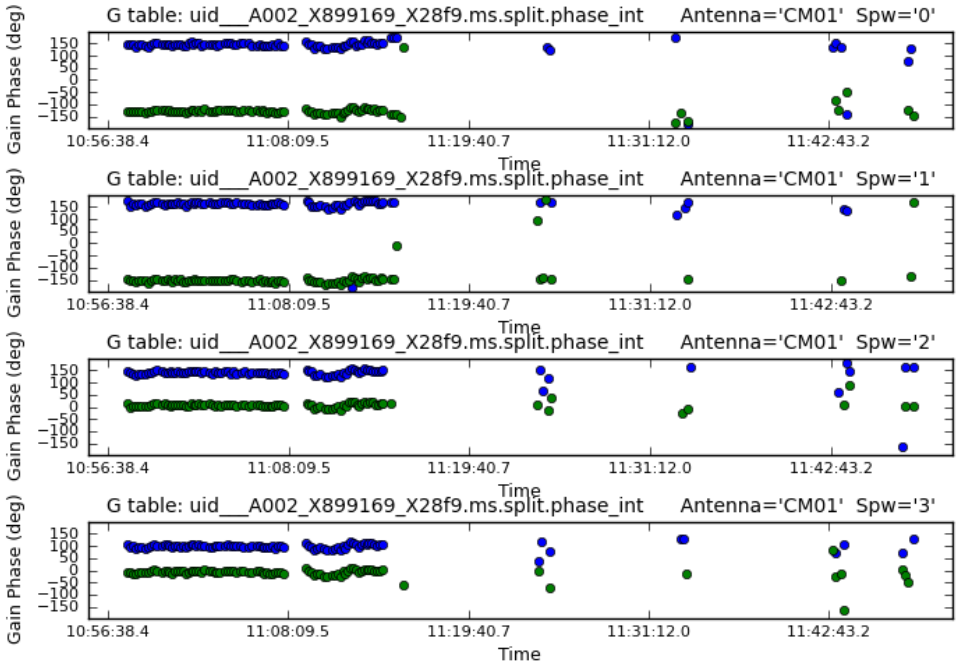
```
# NfR: first obtain phase variation between spws from the strong bandpass calibrator
os.system('rm -rf uid___A002_X899169_X28f9.ms.split.phase_pre_offsets_inf')
gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
  caltable = 'uid___A002_X899169_X28f9.ms.split.phase_pre_offsets_inf',
  field = '0', # J0510+1800
  solint = 'inf',
  refant = 'CM06',
  gaintype = 'G',
  calmode = 'p',
  gaintable = 'uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch')

os.system('rm -rf uid___A002_X899169_X28f9.ms.split.phase_offsets_inf')
gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
  caltable = 'uid___A002_X899169_X28f9.ms.split.phase_offsets_inf',
  field = '0', # J0510+1800
  solint = 'inf',
  refant = 'CM06',
  gaintype = 'G',
  calmode = 'p',
  gaintable = ['uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch',
'uid___A002_X899169_X28f9.ms.split.phase_pre_offsets_inf'],
  interp = ['','linearPD'],
  spwmap = [[],phasemap])
```

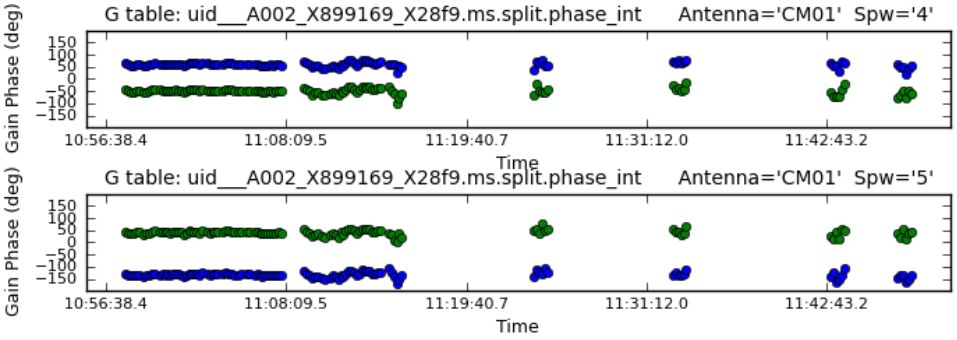
➔ This caltable will contain the offset in phase for each spw respect to spw5

where phasemap is defined at the beginning of the script as:

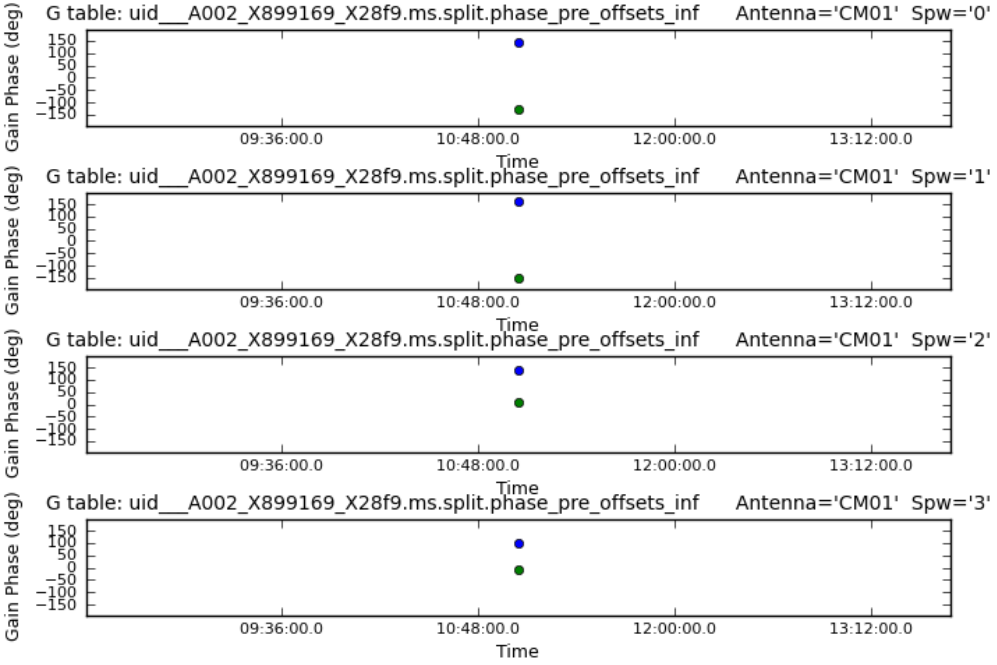
```
phasemap = range(6)
phasemap[0] = 5
phasemap[1] = 5
phasemap[2] = 5
phasemap[3] = 5
phasemap[4] = 4
phasemap[5] = 5
```

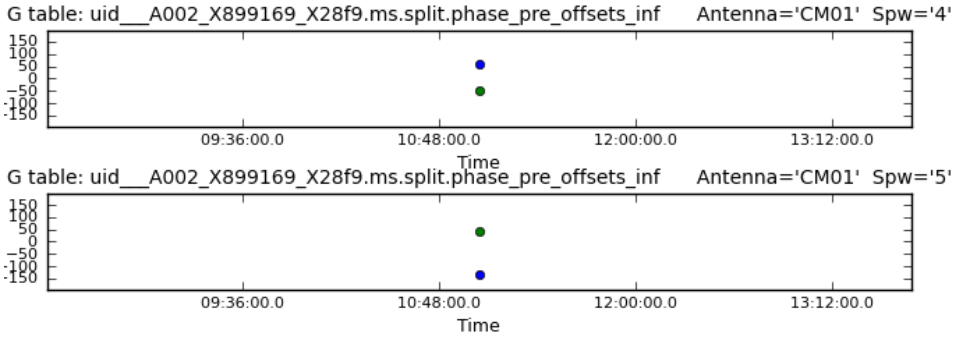
(I selected spw5 because is the closest in freq to spws 0~3)



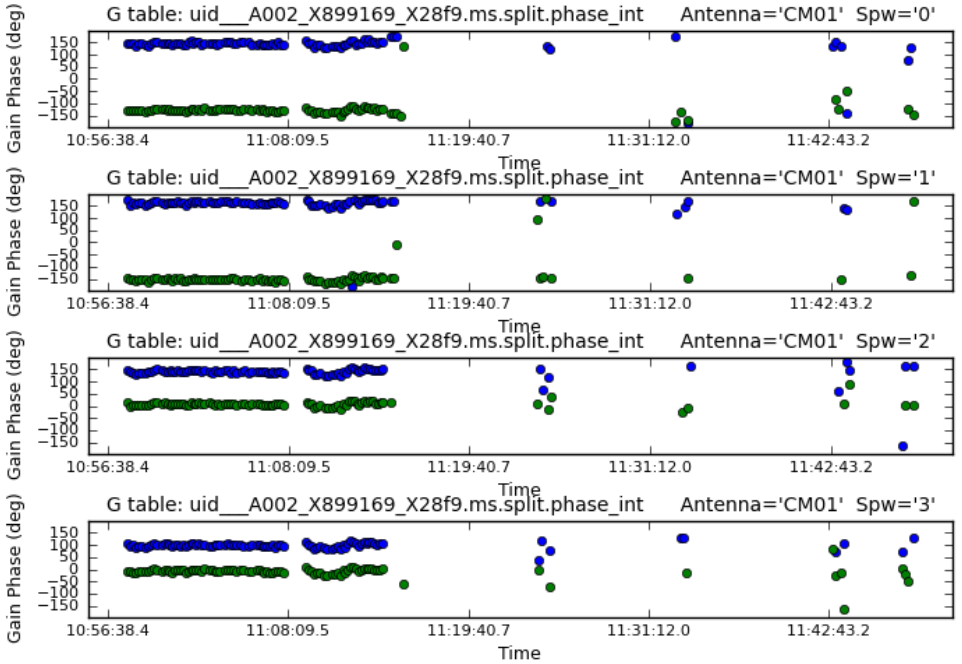


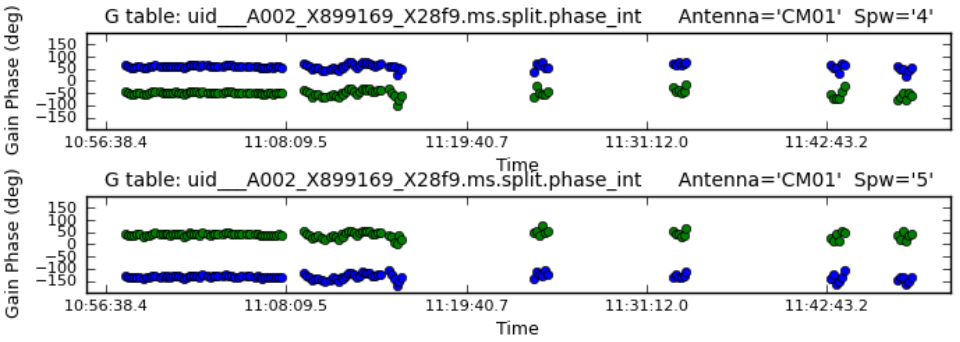
phase_int caltable



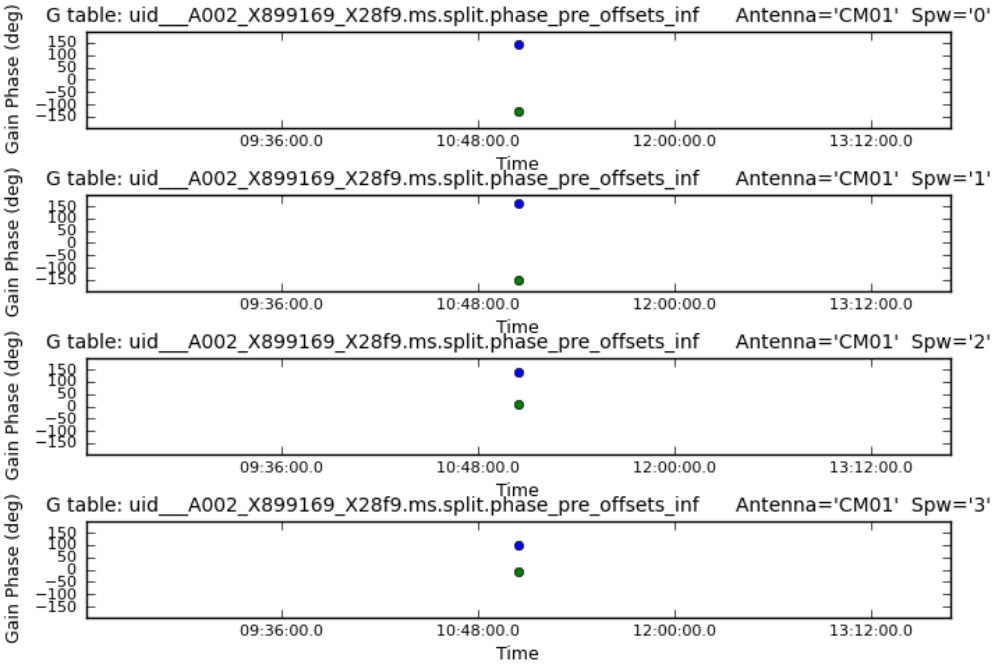


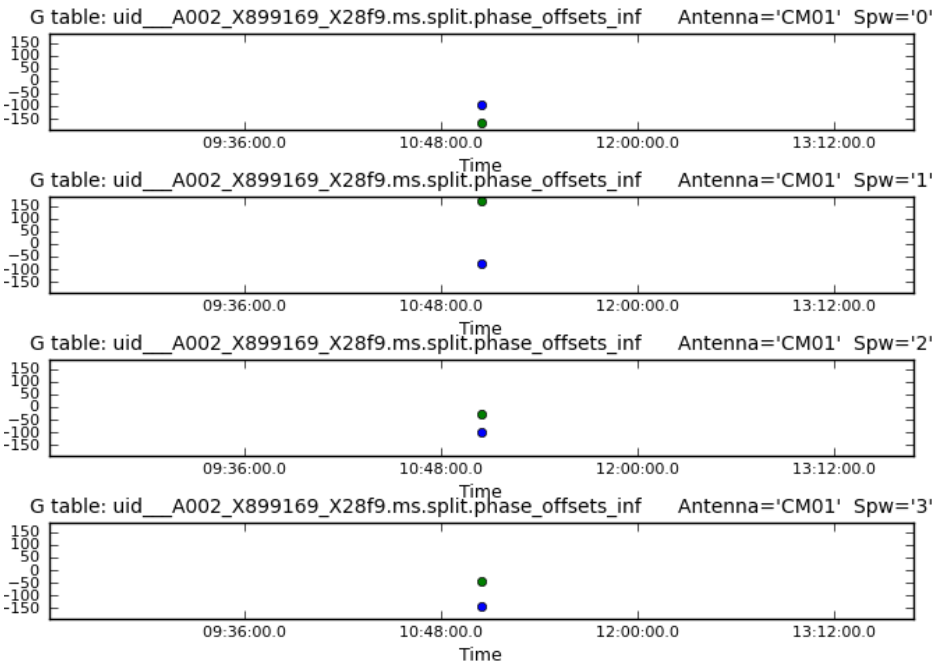
phase_pre_offsets_inf caltable





phase_offsets_inf caltable: Phase offset for each spw respect to spw5





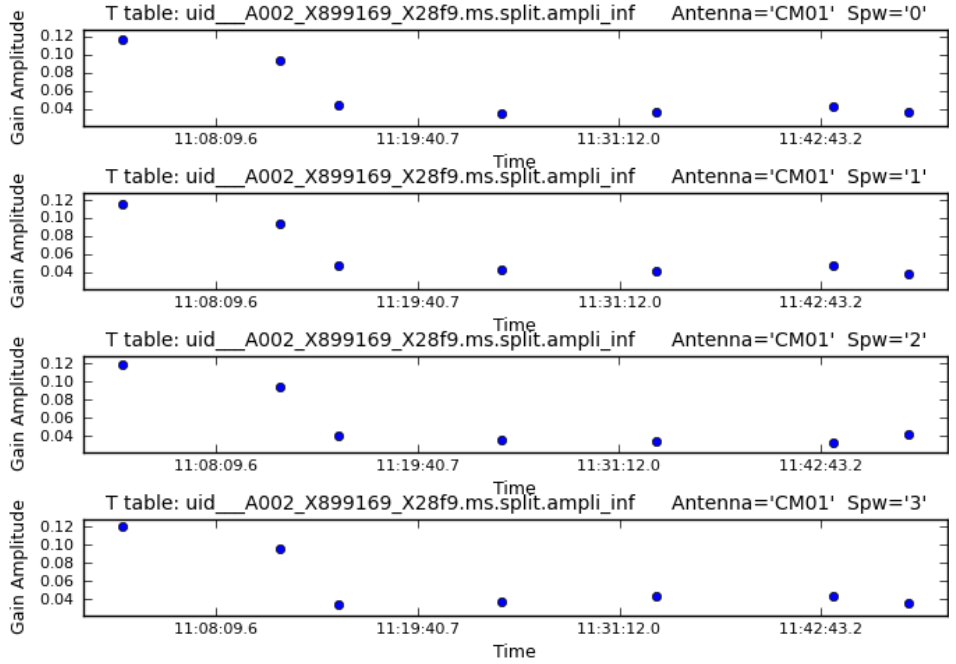
Then, we obtain the usual phase_int solutions for all calibrators and spws (low s/n for 0~3)

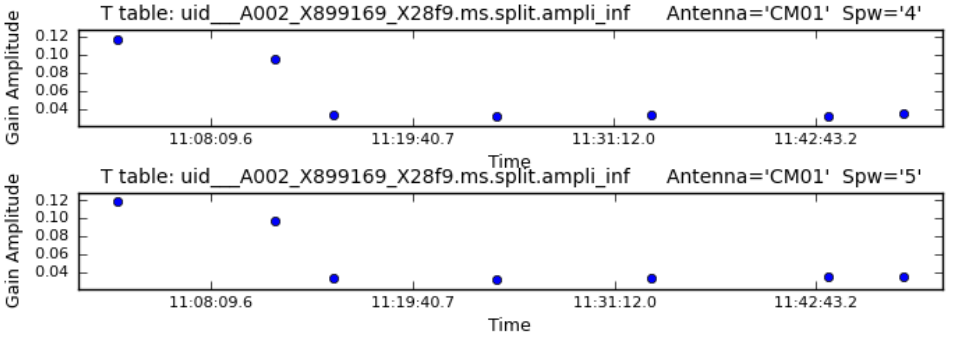
```
os.system('rm -rf uid___A002_X899169_X28f9.ms.split.phase_int')
gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
  caltable = 'uid___A002_X899169_X28f9.ms.split.phase_int',
  field = '0~2', # J0510+1800,J0510+180,J0604+2429
  solint = 'int',
  refant = 'CM06',
  gaintype = 'G',
  calmode = 'p',
  gaintable = 'uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch')
```

Which will be applied in the amplitude gaincal *but* using phase_int solutions from spw5, and taking into account the frequency dependence bw spws obtained above contained in phase_offsets_inf caltable

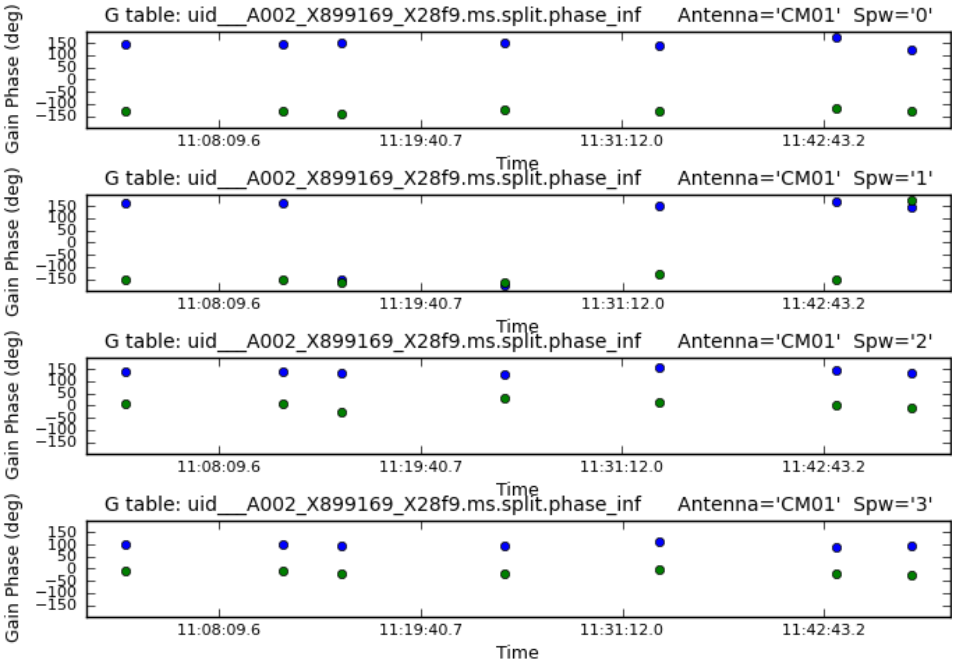
```
os.system('rm -rf uid___A002_X899169_X28f9.ms.split.ampli_inf')
gaincal(vis = 'uid___A002_X899169_X28f9.ms.split',
  caltable = 'uid___A002_X899169_X28f9.ms.split.ampli_inf',
  field = '0~2', # J0510+1800,J0510+180,J0604+2429
  solint = 'inf',
  refant = 'CM06',
  gaintype = 'T',
  calmode = 'a',
  interp = ['','linearPD',''],
  spwmap = [[],phasemap,[]],
  gaintable = ['uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch',
'uid___A002_X899169_X28f9.ms.split.phase_int','uid___A002_X899169_X28f9.ms.split.phase_offsets_inf'])
```

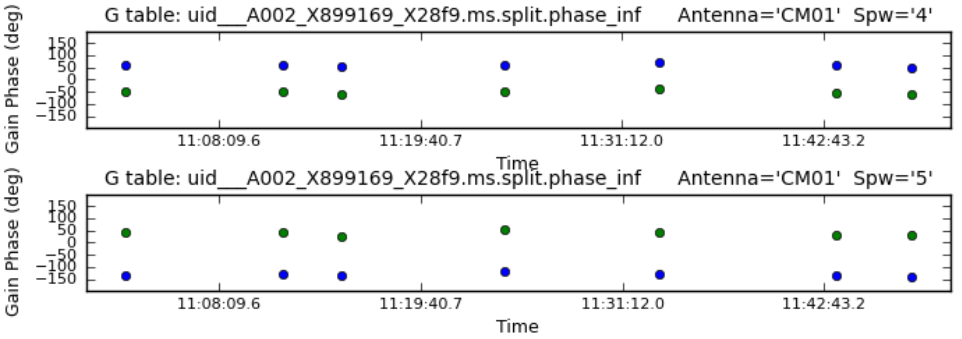
Then do usual fluxscale and phase_inf solutions





ampli_inf caltable





phase_inf caltable

Modifications for phase transfer (in applycal step)

```
# NfR: applycal for phase calibrator and target fields, using phase transfer:
  applycal(vis = 'uid___A002_X899169_X28f9.ms.split',
    field = '2,4~25', # G191.51-0.76
    gaintable = ['uid___A002_X899169_X28f9.ms.split.bandpass_smooth20ch',
'uid___A002_X899169_X28f9.ms.split.phase_inf', 'uid___A002_X899169_X28f9.ms.split.flux_inf',
'uid___A002_X899169_X28f9.ms.split.phase_offsets_inf'],
    gainfield = ['', '2', '2',''], # J0604+2429
    interp = ['', 'linearPD', '',''],
    spwmap = [[], phasemap, phasemap,[]],
    calwt = T,
    flagbackup = F)
```

➔ Note that we apply the phase_inf and flux_inf caltables using phase transfer from spw5