

James Miller-Jones
2009 June 29

The data in question are from project code AT166, and these instructions aim to deal with the 5-GHz C-band data. I assume that a calibrated, single-source uv-dataset is available, denoted as 3C129.SPLIT.1

#####

First of all, plot the data
Assume *i* is the catalog number of this SPLIT file
Find the relevant number using *ucat*

UVPLT

```
default uvplt
getn i
dotv 1
docal -1
gainuse 0
sources "
stokes 'i'
go
```

This plots amplitude against distance from the centre of the uv-plane. Note the maximum baseline out to which the plot goes, *b_max* (in λ). Use this to calculate the required pixel size (3-5 pixels across the resolution element).

Also calculate the required image size to image the full primary beam. FWHP beamsize for a 25m dish ~ 45 arcmin/frequency(GHz)

```
cellsi = 3600*180/(5*pi*uvmax*1000)
imsi = 45*60/(freq(GHz)*cellsi)
```

If there is any bad data, get rid of it with TVFLG

#####

```
TVFLG
default tvflg
getn i
dparm(3) 1
outfgver 1
go
```

#####

Now image the source IMAGR
default imagr
getn i

```
flagver 1
cellsi 0.22
imsi 2048
uvwtn 'n'
dotv 1
niter 2000
boxfile 'PWD:3c129.box'
oboxfile boxfile
go
```

Image size must be a power of 2.

Now interactively CLEAN the image until you don't trust any of the remaining flux. Then stop CLEANing.

```
#####
```

Inspect the final image.

j is the catalog number of the resulting ICL001 file

```
getn j ; tvin ; tvlo ; tvps
```

Find the rms noise

```
tvwin ; imstat
```

imh or qh: print the max, min flux in the CLEANed image

```
#####
```

Plot up the pixel distribution

```
IMEAN
```

```
default imean
```

```
getn j
```

```
doinver 1
```

```
dohis 2
```

```
dotv 1
```

```
pixavg 0
```

```
pixstd 0
```

```
tvin
```

```
tvwin
```

```
go
```

This places the rms noise in the image header, as ACTNOISE

```
#####
```

Now self-calibrate the data, using your image as a model

```
CALIB
```

```
default calib
```

```
getn i
```

```
get2n j
```

```
cmethod 'dft'
```

```
cmodel 'comp'
```

```
refant 22
```

```
solint 2
```

```
solmo 'p'
```

```
go
```

This writes an SN table and a CALIB file (catalog number k)

#####

Inspect the solutions

```
SNPLT
default snplt
getn j
inext 'sn'
inver 1
dotv 1
nplot 9
opty 'phas'
go
```

Ensure the phases vary smoothly with time

#####

Re-image the data

```
IMAGR
tget imagr
getn k
niter 5000
go
```

Produces a file ICL001.2, which has catalog number m

#####

Assess the new rms noise

```
IMEAN
tget imean
getn m
go
```

#####

Try an amplitude and phase selfcal

```
CALIB
getn k
get2n m
solint 20
solmo 'a&p'
cparm(2) 1
go
```

Produces new CALIB file (catalog number n) and new SN table

#####

Now you MUST inspect the data to check it did something sensible

```
SNPLT
tget snplt
getn k
inext 'sn'
inver 1
dotv 1
nplot 9
opty 'amp'
```

```
go
  Ensure the amplitudes are relatively uniform
UVPLT
tget uvplt
getn n
go
#####
Now make the final image
  IMAGR
tget imagr
getn n
go
#####
```