Swinburne University of Technology

Centre for Astrophysics and Supercomputing

Reduction of the ATCA data with *MIRIAD*

by:
Robert Džudžar
email: rdzudzar@swin.edu.au

2017

Contents

1 Introduction **1**

2 Load and Inspect data **4**

3 Flagging **10**

4 Calibration **12**

5 Imaging **18**

6 Analysis **22**

References **30**

Chapter 1

Introduction

This reduction is performed on Swinburne supercomputer, to load programs type in terminal:

- `module load miriad` – will load miriad
- `module load karma` – will load `kvis` that will be used for visualization

Reduction of the data from the Australian Telescope Compact Array (ATCA) using the radio interferometry data reduction package, *MIRIAD* (Sault et al., 1995). This text describes reduction process for the pre-CABB data.

- Starting Miriad: `miriad`
- Input for specific task: `inp task_name` (e.g. `inp uvplt`)
- Once entered input for specific tasks, task can be re-called with the: `tget task_name`
- To run task, type: `go`
- Miriad will give explanation for any task, typing: **help 'task_name'** into Miriad shell.
- To clear input of certain parameter, type: `unset` parameter_name

Useful links

ATCA user guide:
https://www.narrabri.atnf.csiro.au/observing/users_guide/html/atug.html

ATCA archive:
https://atoa.atnf.csiro.au

NASA/IPAC Extragalactic Database (NED):
https://ned.ipac.caltech.edu

DS9 (useful tool, optional):
http://ds9.si.edu/site/Home.html

ATCA – Australia Telescope Compact Array

The ATCA is an array of six 22 m diameter antennas located 237 m above sea level. The array has a 3km east-west track with a 214m northern spur. Five antennas can be moved along these tracks, with the sixth antenna at a fixed position 3 km to the west of the east-west track. The longest possible baseline is, therefore, 6km. The array can be used for observations in five wavelength bands between 27 cm and (with five antennas only) 3 mm, between frequencies of approximately 1.1 GHz and 105 GHz.

The Australia Telescope Compact Array is an earth-rotation aperture synthesis radio interferometer. Earth-rotation aperture synthesis was first used in the 1950s for radio observations of the sun. The technique is comprehensively explained, with a historical perspective, in Interferometry and Synthesis in Radio Astronomy by Thompson, Moran & Swenson (Wiley, 1st edition 1986, 2nd edition 2001). Essentially, the array of antennas is comprised of a number of two-element interferometers. The visibility (i.e., the fraction of the signal common to both antennas of a pair) is derived by multiplying the (suitably delayed) signals together.

Figure 1.1. Five of the six ATCA antennas in the H75 array.

Figure 1.1: ATCA

Above information (and Figure 1.1) about ATCA is taken from: https://www.narrabri.atnf.csiro.au/observing/users_guide/html/atug.html

Check Archive and NED

Get familiar with your source, check whether all files are downloaded from the Archive and check source on the NED. You can search them by name, position on the sky. **From ATOA: Note their Project Code, Principle Investigator, Date and Baseline configuration – use then search using project code - to check whether all files are downloaded from that day - so we don't miss some calibration files.**

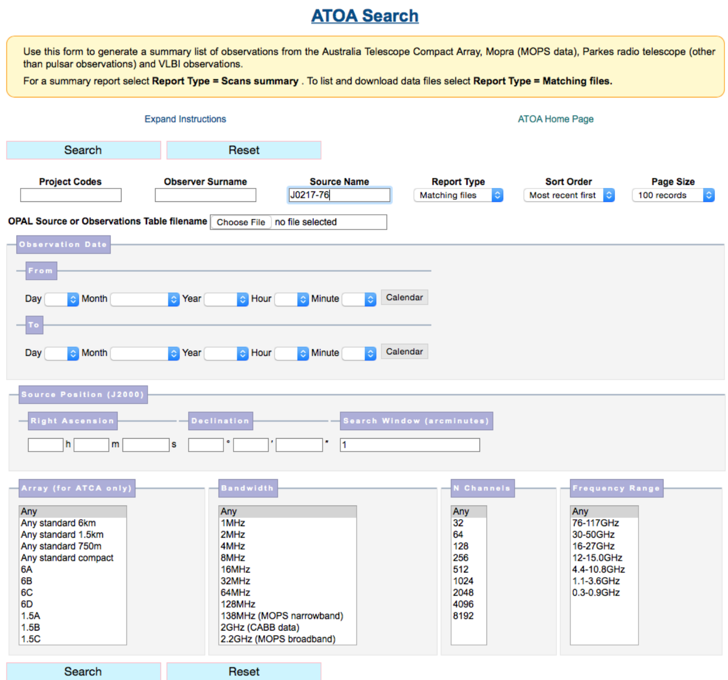


Figure 1.2: ATCA archive: https://atoa.atnf.csiro.au

From NED, note down the source velocity information. You can also check what source you have etc.

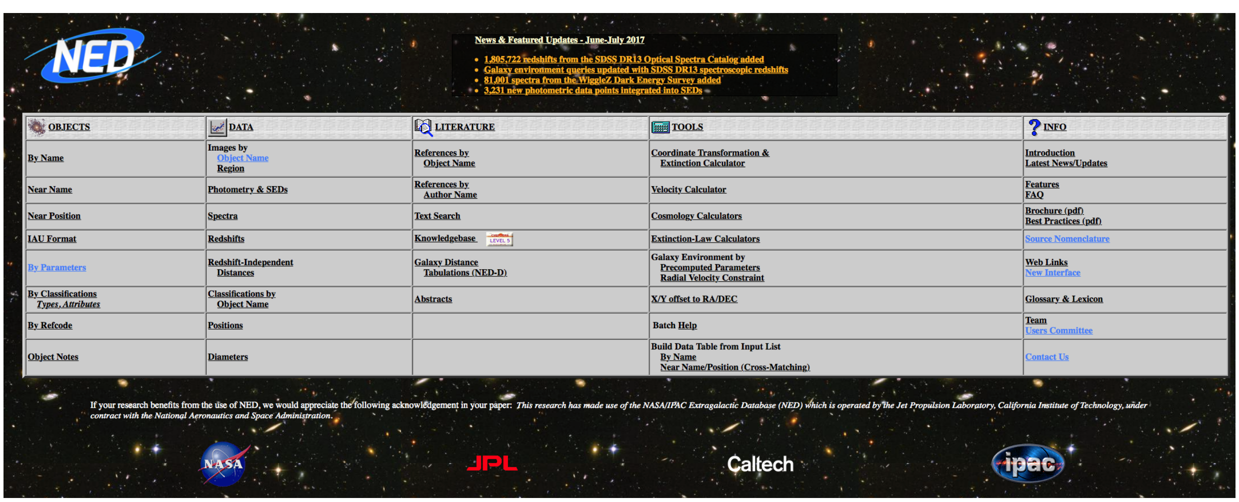


Figure 1.3: NED: https://ned.ipac.caltech.edu

Chapter 2

Load and Inspect data

The data was downloaded from the http://atoa.atnf.csiro.au/query.jsp archive, we will use task ***atlod*** to load them into $\mathcal{MIRIAD}$. FILENAME e.g. `94-06-12_1058.C339`: **94-06-12** date; **1058** time; **C339** project code. In case you have multiple days - load one day only - flag, calibrate; then load other day and do the same, we can combine them later in the `invert` step.

atlod%

```
in = 2011-03*
out = 1.5A.uv
ifsel =
restfreq = 1.420405752
options = birdie,compress,bary,noif
nfiles =
nscans =
nopcorr =
edge =
```

The ***atlod*** task is used to convert ATCA RPFITs data into a Miriad data-set. `in` – depends on your file names; with * you'll select all files that starts with this strings. The rest frequency is for the HI line. `options` will set some automatic flagging (removing bad data). CABB data: options = birdie,rfiflag,xycorr,bary,noauto

⋆ **birdie**: flags out channels where ATCA is affected by self-inferference.
⋆ **rfiflag**: flags known bad channels.
⋆ **bary**: Uses barycentre for velocity rest frame (Earth+Sun center of mass).

uvsplit%

```
vis = 1.5A.uv/
select =
options = nosource
maxwidth =
```

To work with single sources that are imaged on a specific frequency, multi-source (.uv) data is split with the ***uvsplit*** task, so we can separate the sources.

⋆ **nosource**: Doesn't make output data-set based on source name, just separates frequencies.

uvsplit%

```
vis = uvsplit.1393.2/
select =
options =
maxwidth =
```

To work with single sources that are imaged on a specific frequency (close to HI rest frequency in this case), we need to further split the data and it should contain: source, primary and secondary calibrator.

prthd%

```
in = j2027-51.1393/
log =
options =
```

The task ***prthd*** is used for the scientific summary of the dataset.

puthd%

```
in = j2027-51.1393/restfreq
value = 1.420405752
type =
```

In case **restfreq** is empty, this task can be used in order to input the value of the rest frequency.

uvlist%

```
vis = j2027-51.1393/
options = brief,spectral
select =
stokes =
line =
recnum =
log =
```

Gives information about UV file. For example of the output see Figure 6.2.

⋆ **brief**: Short list.

⋆ **spectral**: Gives spectral and velocity information. Note that there are both: radio and optical convection; this information will be useful for imaging part.

```
UV Listing for data-set j2027-51.1393/

Options: brief,spectral

----------------------------------------

Velocity rest frame        : Barycentric
Obs radial velocity (km/s):   -14.034
Rest frequency      (GHz) :   1.4204057
Start channel             :           1
Number of channels        :       17409
Start frequency     (GHz) :   1.3975000
Frequency increment (GHz) :  -0.0000005
Radio Velocities:
  Start velocity    (km/s):    4848.556
  End velocity      (km/s):    6642.575
  Velocity increment(km/s):       0.103
Optical Velocities:
  Start velocity    (km/s):    4927.796
  End velocity      (km/s):    6792.449
  Velocity increment(km/s):       0.106
```

Figure 2.1: Output of the **uvlist**.

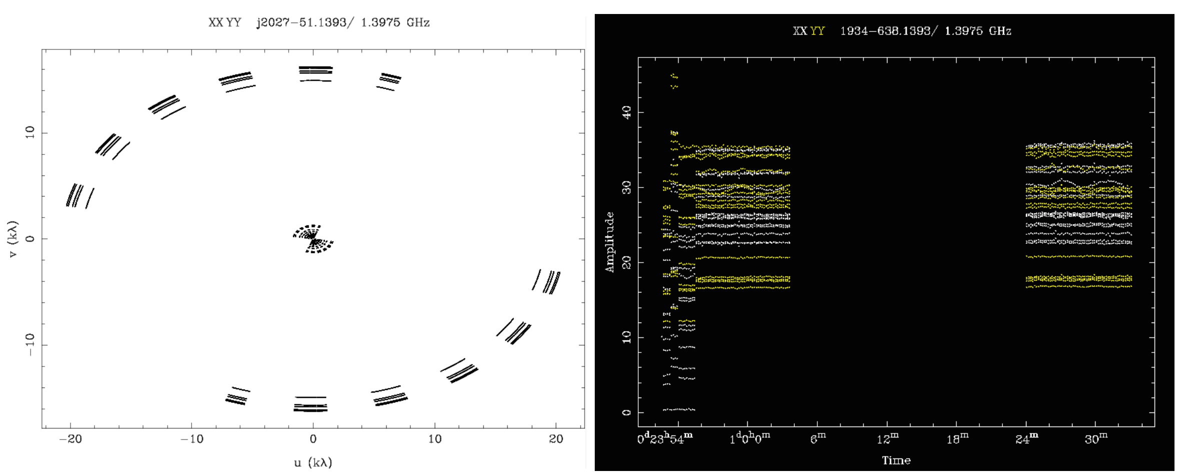
Plotting

uvplt%

```
vis = j2027-51.1393/
line =
select =
stokes = xx,yy
axis = uc,vc
xrange =
yrange =
average =
hann =
inc =
options = nobase
subtitle =
device = /xs
nxy = 1,1
size =
log =
comment =
```

To plot and inspect the visibilities **uvplt** can be used. `vis` is your visibility data; `nxy` is to set how many plots per window are shown. `nxy = 1,1` makes one plot, `nxy = 3,3` makes 3×3 plot, so nine of them.

- ⋆ **nobase**: Plot all baselines (baseline - measurement within two antennas) on one plot. With options unset it will plot baseline by baseline.
- ⋆ **uc,vc**: Axis for plot. Multiple variation can be used - time, amplitude; time, phase; uc,vc; and others.



(a) (b)

Figure 2.2: a) Plot of the UV coverage as seen with **uvplt** with axix = uc,vc; b) Amplitude vs. time – we can notice that there are some issues at the beginning of the observations.

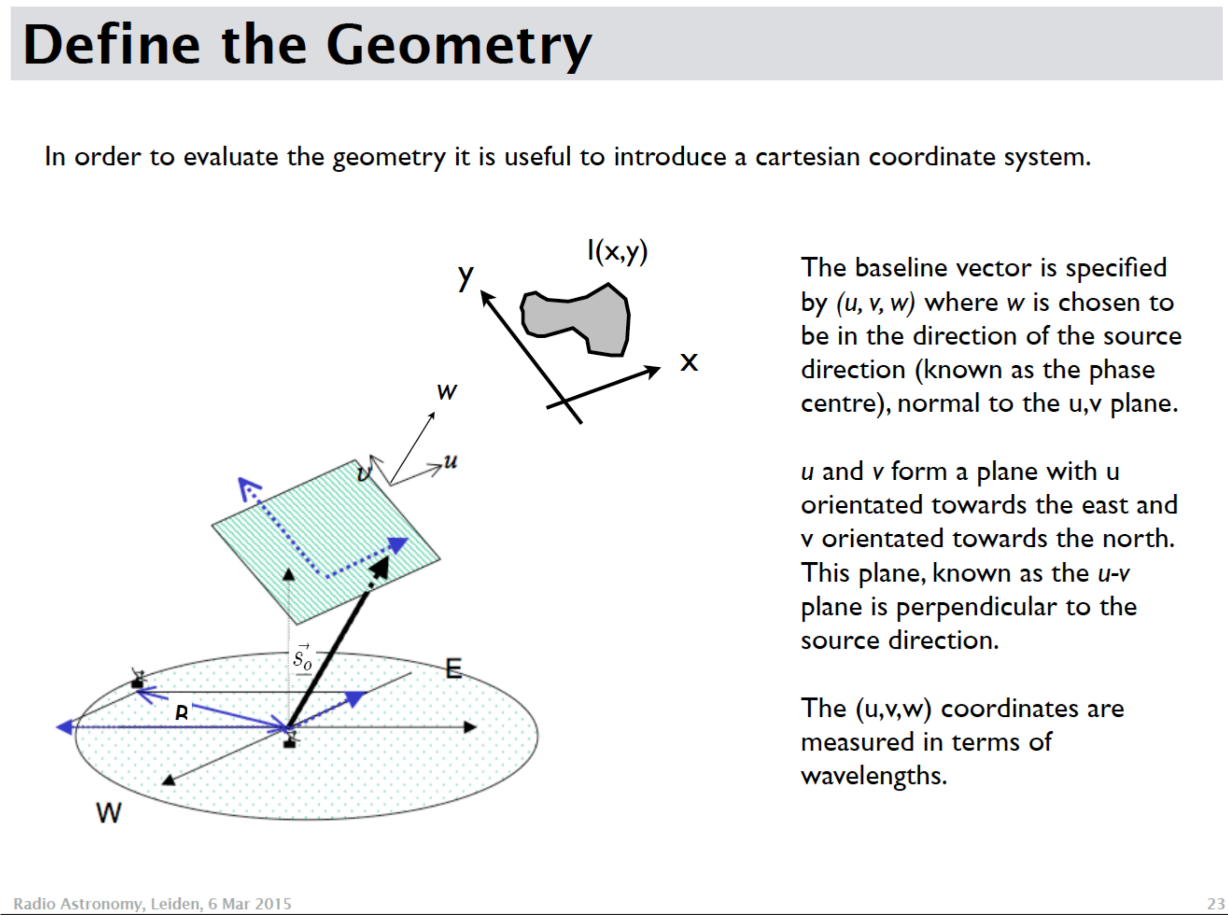


Figure 2.3: $u - -v$ plane. Slide taken from Huib Jan van Langevelde talk 'Advanced Interferometry'.

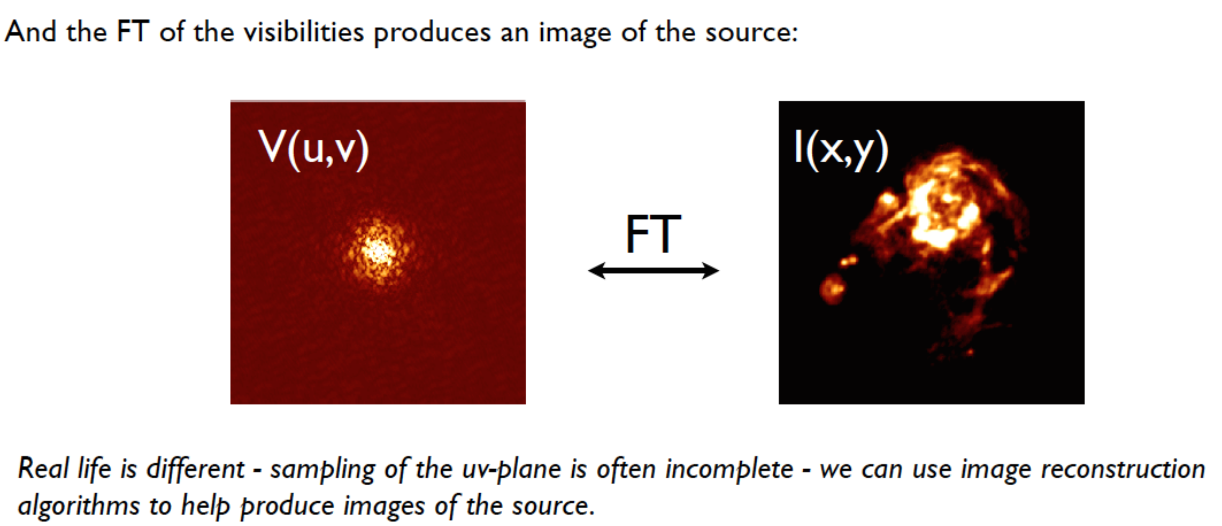


Figure 2.4: Fourier Transform of the visibilities gives an image of the source. Image taken from Prof. Mike Garrett: Basics of radio interferometry & Aperture Synthesis

uvspec%

```
vis = 1933-587.1393/
select =
line =
stokes = xx,yy
interval =
hann =
offset =
options =
axis = channel,amplitude
yrange =
device = /xs
nxy = 1,1
log =
```

Besides **uvplt** for plotting and inspecting the data, **uvspec** can be also used. `Stokes=i` will be use to inspect intensity.

Ignore cgdisp at the moment, will be back to it when we start analysing the data.

cgdisp%

```
in = j2027-51.icln/
type = c
region = arcsec,kms,box(-700,-700,700,900)(5520,6140)
xybin =
chan =
slev =
levs1 =
levs2 =
levs3 =
cols1 =
cols2 =
cols3 =
range = 0,0.5,lin,2
vecfac =
boxfac =
device = /xs
nxy = 3,3
labtyp = arcsec
beamtyp = b,l,4
options = full,wedge,3val
3format =
lines =
break =
csize =
scale =
olay =
```

Task for displaying image, it can be used for contour plots, regular images, combination of images and contours. It can take multiple inputs, and then they are defined in type whether plot will be contour or image. Example can be seen in the Figure 2.5. More details will be in the Analysis section.

- ⋆ **type**: c – contour; g – image
- ⋆ **region**: Region to plot - image size dimensions and velocity space.
- ⋆ **levs1**: Values for the contours to be plotted
- ⋆ **range**: Plot range: minimum intensity, maximum intensity, transfer function type, color table.
- ⋆ **beamtyp**: For plotting beam on the plot. With (b,l,4), beam will be cross-hatched in the bottom left corner.
- ⋆ **options**: **full** – to plot annotation with contour levels, display range, eference values, etc.;

wedge – plot map of intensity to color; **3val** – label each sub-plot with the appropriate value of the third axis.

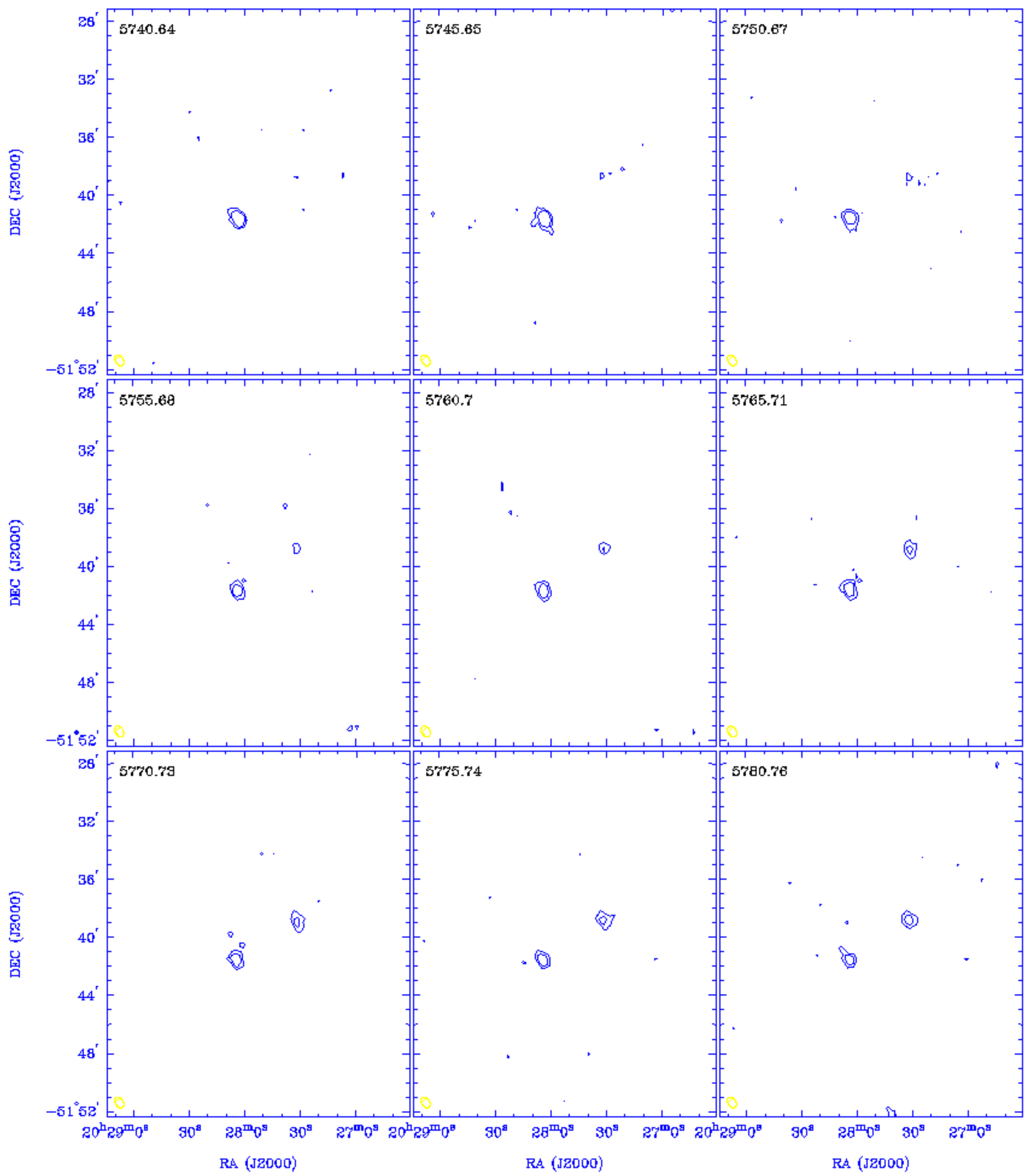


Figure 2.5: Example of contour plot in the **cgdisp** of the galaxy group. We can see position on the x and y axis, while stepping through velocity channels - velocity is marked in the upper left corner. We can vary range, levs1. Placing `type = g` will give us image. Also we can plot here our Moment maps, specifying either flux (Moment 0) or velocity (Moment 1, 2).

Chapter 3

Flagging

During inspection of the data with **uvplt** or **uvspec**, one can notice bad data that needs to be flagged (removed). Here are some examples of the tasks for flagging and their brief description.

uvflag%

```
vis = j2027-51.uvlin/
select = amplitude(500)
line =
edge =
flagval = flag
options =
log =
```

Setting flagval to be *flag*, the uv data will be flagged as bad. **Be careful not to flag everything** – in case that happens, no worries because it can be undone, however, it will mean that all of your data is un-flagged.

⋆ **select**: There are few options how the data can be selected in order to be flagged. Example:*select = time(hh:mm:ss,hh:mm:ss); select = amplitude(120)* etc. so type 'help select' to examine other options.

blflag%

```
vis = 1933-587.1393/
line =
device = /xs
stokes = i
select =
axis = time,amp
xrange =
yrange =
options =
```

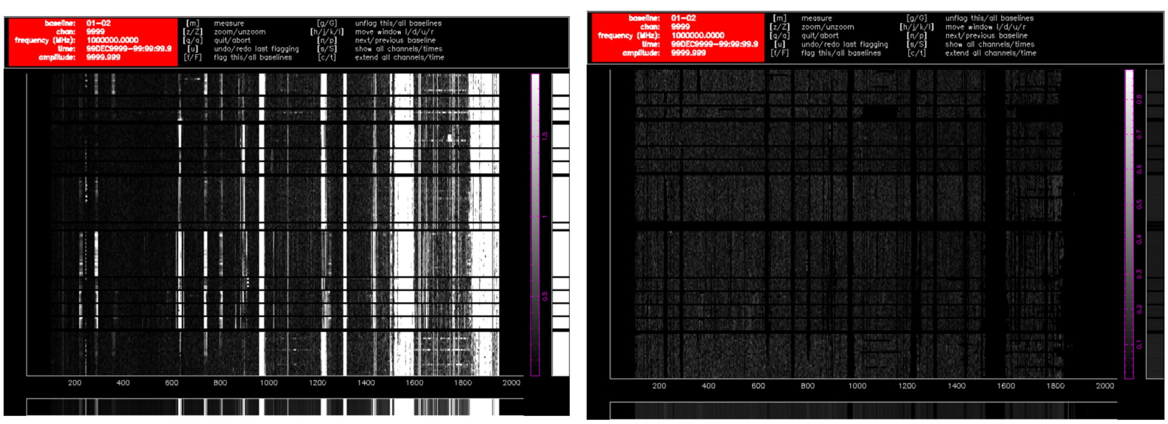
This task is useful for manually removing bad points: such as large amplitude peaks that are most often result of an RFI (Radio interference) from airplanes, mobile phones, satellites, sun, moon, weather stations etc. See example in Figure 3.1. With left mouse click you can select bad points; pressing `r` will refresh plot – while removing selected bad point; with right mouse click you can go to next baseline.

pgflag%

```
vis = 1933-587.1393/
line =
select =
stokes =
device = /xs
device2 =
mode =
flagpar =
patch =
log =
command =
options =
```

This task can be used for automatic flagging, examining baseline by baseline, we can select certain data and flag them by pressing `f`; Selection of the bad 'times' or 'channels' with mouse click: left click – right click, their separation will select box size region.

- See the intensity value of the pixels (colorbar); white - high intensity ones are the ones that we tend to minimize/remove;
- Selecting box shaped regions with mouse click;
- Pressing **'z'** you can zoom into that region; Capital **'Z'** for un-zoom;
- If you want to flag whole row or column with specific width: select that bad data and by pressing **'t'** or **'c'** it will select whole row or column.
- Pressing '<' – algorithm will be invoked to do automatic removal of the high intensity data.
- To go to the next baseline, press **'n'**; to return to the previous baseline, press **'p'**.
- To exit this task press **'q'** button, and then type **'Y'** in the terminal – this will save performed flagging.



(a) (b)

Figure 3.1: How data looks like in `pgflag`; a) Unflagged; b) After flagging. Image taken from http://www.narrabri.atnf.csiro.au/observing/users_guide/html/chunked/ch04s03.html

Chapter 4

Calibration

Once we are happy with the flagging process, we can proceed and calibrate our data in order to obtain accurate data since there are many factors (atmospheric and instrumental factors) that distort our observations. During observations of our science targets, we observe also sources with known fluxes, thus we call them: calibrators e.g:

- **1934-638**: The most usual primary and bandpass calibrator;
- **e.g. 0252-712 ; 0407-658** Secondary (phase) calibrator; *vary from source to source;

With the primary calibrator we will determine our flux density scale. We will first use Miriad task `mfcal` (determines gains and bandpass); since flux density of the calibrator is well known.

mfcal%

```
vis = 1934-638.1393/
line =
stokes =
edge =
select =
flux =
refant = 2
minants =
interval = 1
options =
tol =
```

Refant will set up reference antenna; Default is `refant=3`. Interval used to determine the solution in minutes. Pay attention to the output in terminal, if the data are good - the solution will converge within 2-3 iterations.

gpcopy%

```
vis = 1934-638.1393/
out = 1933-587.1393/
mode =
options =
```

Copy solutions from primary calibrator onto secondary.

Check solutions of the calibrations using `uvplt` & `uvspec`

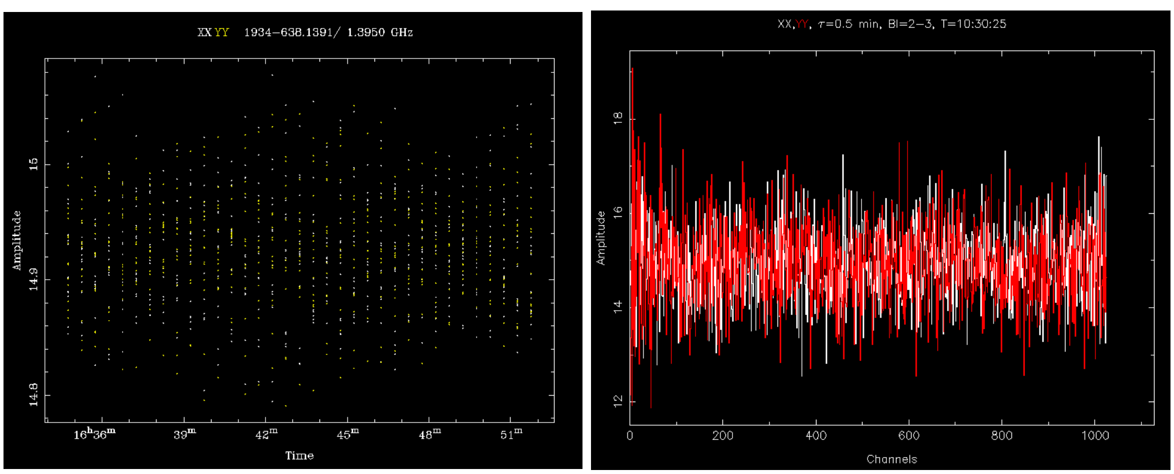
Time vs. amplitude using **uvplt** for plotting and inspecting the data, see Figure 4.1a; Channel vs. amplitude using **uvspec**. `Stokes=i` will be use to inspect intensity, see Figure 4.1b.

uvplt%

```
vis = j2027-51.1393/
line =
select =
stokes = i
axis = time,amplitude
xrange =
yrange =
average =
hann =
inc =
options = nobase
subtitle =
device = /xs
nxy = 1,1
size =
log =
comment =
```

uvspec%

```
vis = 1933-587.1393/
select =
line =
stokes = xx,yy
interval =
hann =
offset =
options =
axis = channel,amplitude
yrange =
device = /xs
nxy = 1,1
log =
```



(a) **(b)**

Figure 4.1: a) Time vs. Amplitude for the primary calibrator; b) Channel vs. Amplitude

Skip `gpcal` if you have measured only the XX and YY correlations.

gpcal%

```
vis = 1934-638.1393/
select =
line =
flux =
refant = 2
minants =
interval = 0.1
nfbin =
tol =
xyphase =
options = nopol,noxy
```

First for primary (1934-638.1393/) and once for secondary (1933-587.1393/) calibrator.
Usually done with up to 5 iterations.

- CABB data: `options = xyvary`; then repeat task with the secondary calibrator and `options = qusolve,xyvary`.

gpboot%

```
vis = 1933-587.1393/
cal = 1934-638.1393/
select =
```

In order to correct the flux of a visibility data-set's gain table. We are correcting secondary calibrator using the primary one.

mfboot%

```
vis = 1934-638.1393/,1933-587.1393/
line =
select = source(1934-638)
flux =
mode =
clip =
device = /xs
options =
```

In order to correct the bandpass slope.
`in = primary calibrator, secondary calibrator`, `select=source(1934-638)` only in case this is our primary calibrator - if not, then name of the one that we are using.

At this point, both bandpass calibrator and flux calibrator should be calibrated, so we can copy it onto a source.

gpcopy%

```
vis = 1933-587.1393/
out = j2027-51.1393/
mode =
options =
```

Copy calibration solutions onto a source (out = source file).

Check solutions of the calibrations using `uvplt`

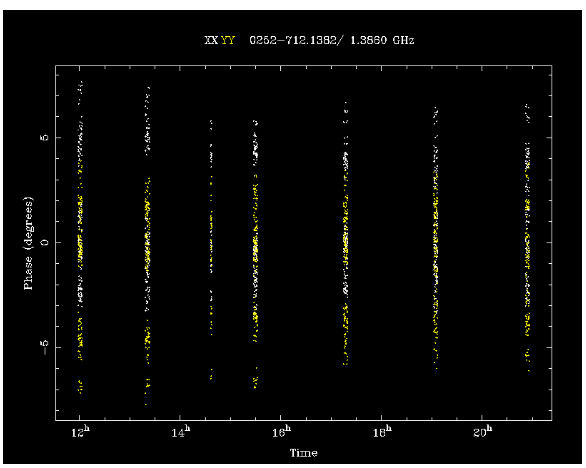
Time vs. phase using **uvplt** for plotting and inspecting the data, see Figure 4.2a; real vs. imaginary using **uvplt**. `Stokes=i` will be use to inspect intensity, see Figure 4.2b.

uvplt%

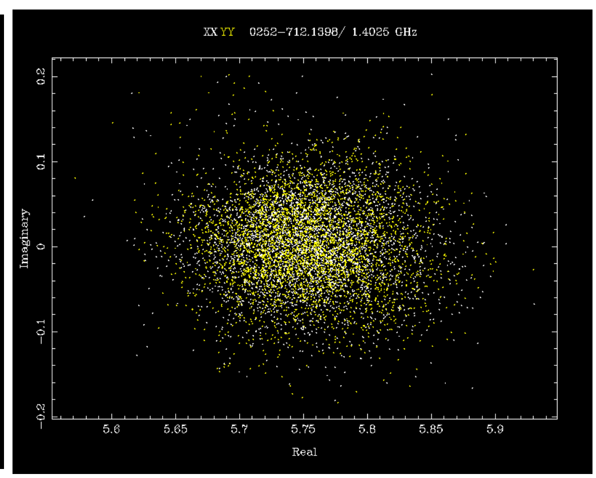
```
vis = j2027-51.1393/
line =
select =
stokes = i
axis = time,phase
xrange =
yrange =
average =
hann =
inc =
options = nobase
subtitle =
device = /xs
nxy = 1,1
size =
log =
comment =
```

uvspec%

```
vis = 1933-587.1393/
select =
line =
stokes = xx,yy
interval =
hann =
offset =
options =
axis = real,imaginary
yrange =
device = /xs
nxy = 1,1
log =
```



(a)



(b)

Figure 4.2: a) Time vs. Phase. Phases are centered around 0 degrees b) Real vs. Imaginary; points are in circular shape, real axis is centered on our flux value.

Not calibrated & issue with calibrations

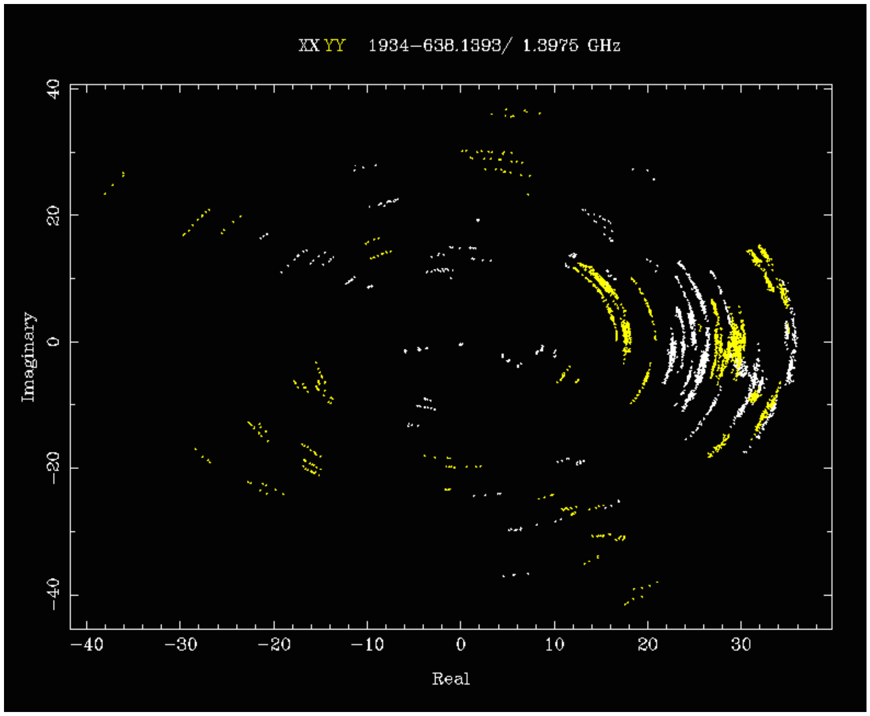


Figure 4.3: Real vs. Imaginary when the data are not calibrated.

Error recognition

Examples are taken from the talk by Emil Lenc, CSIRO Radio School, Narrabri, 2017.

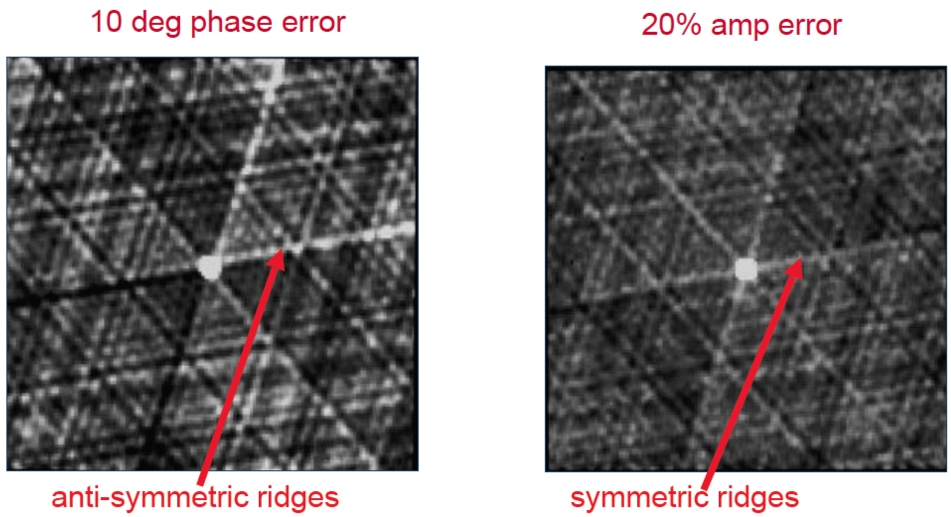
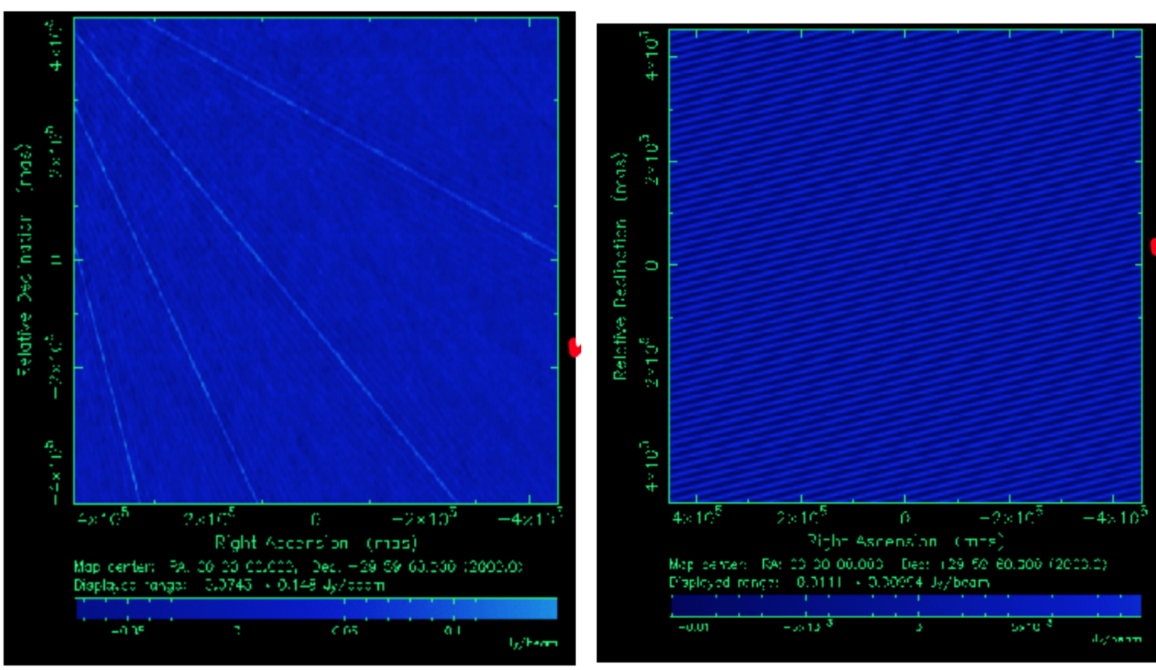


Figure 4.4: Amplitude and Phase errors



(a) (b)

Figure 4.5: a) There is a source outside image b) RFI

Chapter 5

Imaging

First we need to remove the continuum from our data, since we are interested in line emission. This can be done with the task `uvlin`; tasks uses channels with no line emission to model the continuum and removes it. Input is our source, `order = 1` is usually good (except in case there is bright continuum source or the Sun is close by, or some mysterious interference is appearing, then we will see whether to use higher order, or proceed with the `option=sun,twofit`).

uvlin%

```
vis = j2027-51.1393/
select =
line =
chans = 200,6000,13000,16000
out = j2027-51.uvlin
order = 1
offset =
mode =
options =
```

`Chans = 200,6000,13000,16000` – this will highly depend on your data (here is useful to remind about information received with the task `uvlist`). We want to select the channel ranges that contain only continuum (line free), while skipping edges of our data.

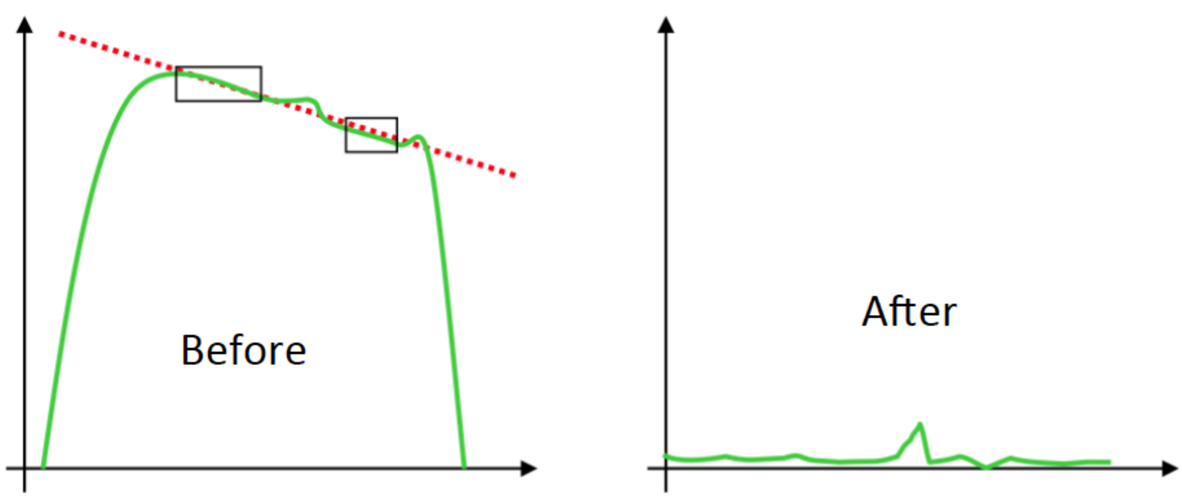


Figure 5.1: Schematics of the continuum subtraction. Image taken from Katherine Johnston talk 'Planning and reducing spectral line projects', ERIS.

We can now finally make our so-called 'dirty' map and beam images from the visibilities, with task `invert`. In case your source was observed over a few days or with multiple baselines then you need to flag and calibrate each dataset separately (by dates) so you will have multiple `uvlin` files, you should put **All uvlin files** into `invert` task: **in = one.uvlin, two.uvlin, five.uvlin**.

invert%

```
vis = j2027-51.uvlin/
map = j2027-51.imap
beam = j2027-51.ibeam
imsize = 256
cell = 20
offset =
fwhm =
sup =
robust = 0.5
line = felocity,160,5100,10,10
ref =
select = -ant(6)
stokes = i
options = double
mode =
slop =
```

Here we will test few setups. `cell` it's number in `arcsec` that will depend on the synthesized beam, with few iteration we will get to a number (depending on the baseline). `line` these numbers will depend on our source, first number determines how large will be our cube, second number determines start velocity, third and forth determines channel and velocity width. `-ant(6)` we will always setup, since we are not using 6th antenna in our analysis.

Different robust parameters

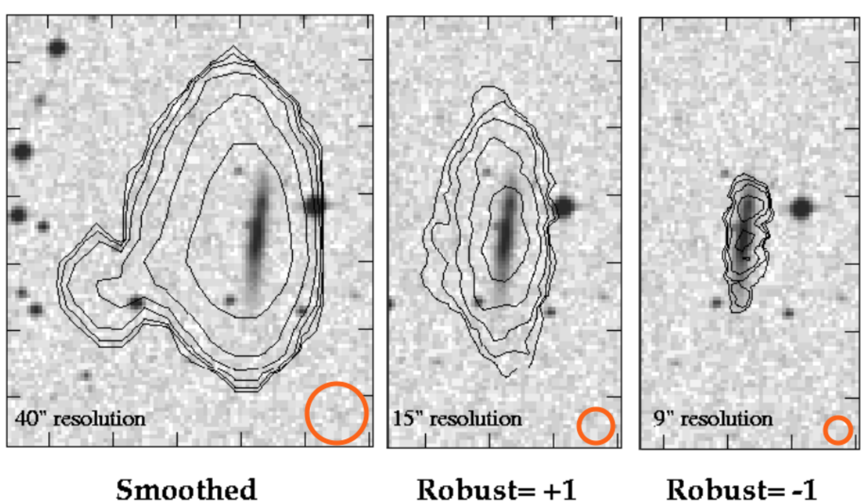


Figure 5.2: Different robust (weighting parameter) values gives a trade off between resolution and sensitivity. Image taken from: John Hibbard talk at Ninth Synthesis Imaging Summer School

The CLEAN algorithm is used to separate the components of the image that are due to the antenna response from the actual emission from the sky.

clean%

```
map = j2027-51.imap/
beam = j2027-51.ibeam/
model =
out = j2027-51.icmp
gain =
options = double
cutoff = 0.04
niters = 4000
region =
phat =
minpatch =
speed =
mode =
clip =
```

Task `clean` performs deconvolution in order to remove sidelobes of a 'dirty' beam by trying to fill out regions which not sampled in the $u--v$ plane. We can set up `cutoff` limit to 3-4-5σ above the theoretical RMS obtained from the `invert` task.

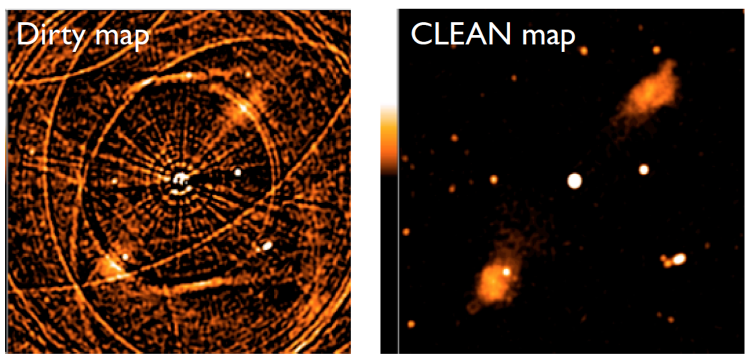


Figure 5.3: Image taken from Prof. Mike Garrett: Basics of radio interferometry & Aperture Synthesis

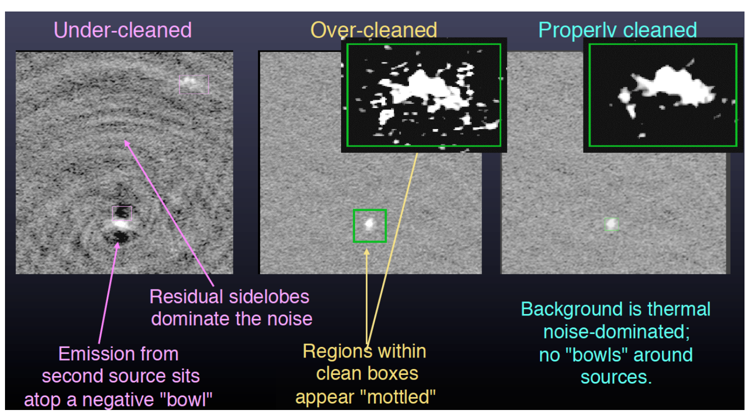


Figure 5.4: Example of a cleaning outcomes. Image taken from Greg Taylor talk: 'Error Recognition and Image Analysis', Fourteenth Synthesis Imaging Workshop.

restor%

```
model = j2027-51.icmp/
beam = j2027-51.ibeam/
map = j2027-51.imap/
mode =
fwhm =
pa =
options =
out = j2027-51.2icln
```

We can finally generate `'clean'` map. Task `restor` generates a 'CLEAN' map, calculating residuals and convolving a model by a Gaussian beam.

This will create data cube which we will analyze (see Figure 5.5). Going through channels, where each 'channel map' represents the spatial distribution of line flux at each successive velocity. Basically, we can watch Doppler effect in action.

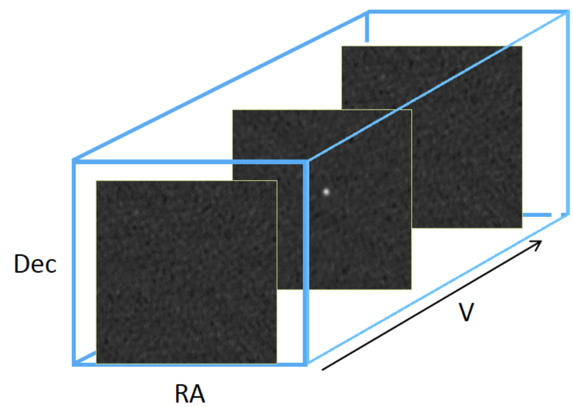


Figure 5.5: How can data cube look like; 2 position axis (RA: Right Ascension; Dec: Declination) and velocity as third dimension (could be frequency).

You can see your cube in `KVIS`, see Section 6.

Chapter 6

Analysis

Get a simple statistics of the data cube; (in `kvis` it is also possible doing the same by hovering mouse over some part of the cube and pressing `s`)

imstat%

```
in = j2027-51.icln/
region =
axes =
plot =
cutoff =
device = /xs
options =
beam =
log =
```

We want to extract from the cube moment maps (see Figure 6.3): Moment 0 map represents the integrated flux over a certain region in our cube; Moment 1 map represents velocity map of our source (we can see rotation); Moment 2 map represents velocity dispersion of our source.

moment%

```
in = j2027-51.icln/
region = kms,image(5520,6140)
out = j2027-51.2imom
mom =
axis = 3
clip = 0.008
span =
rngmsk =
pkmask =
```

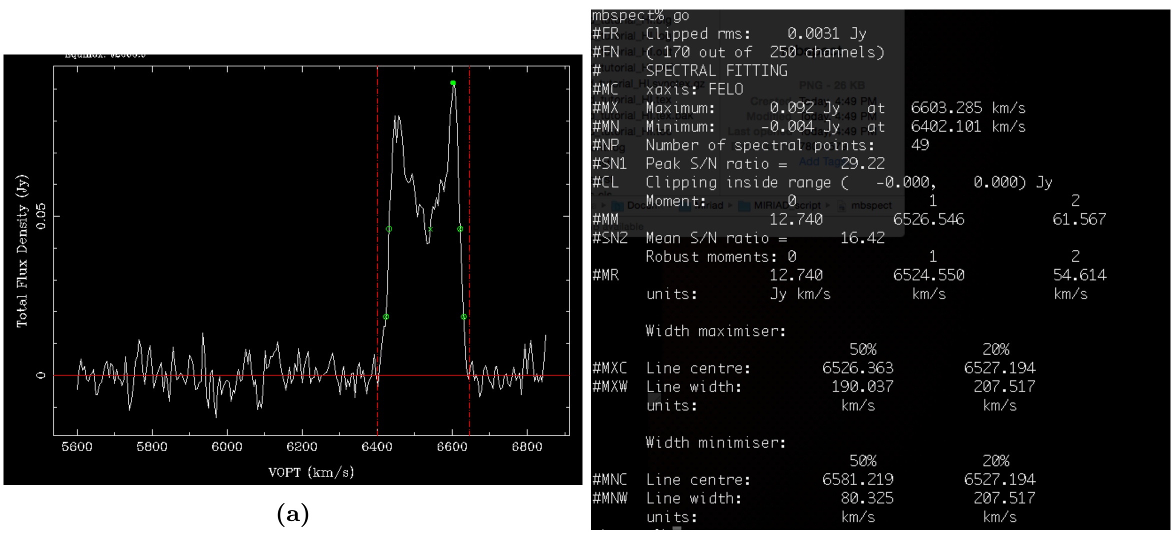
`clip` parameter will depend on RMS, mostly set up at 3×RMS. `region` selection of the region inside the cube where we see emission from the source, in brackets is given velocity range where it appears - you can determine value using `kvis`.

We can see spectrum of our source; HI emission line in this case using `mbspect`

mbspect%

```
in = j2027-51.icln/
out =
coord = 20:28:08.379,-51:28:31.18
width = 11
xaxis = VELO
yaxis = sum
xrange =
yrange =
hann =
order = 2
options = measure
clip =
mask = 5780,6270
profile = 5780,6270
device = /xs
csize =
lines =
colors =
log =
comment =
```

Here we have to input several informations: `coord` input galaxy coordinates - if we know our source. You can check **NED** for that. `width` will depend on the size of our source - we can see number of pixels viewing `moment 0` map in `kvis`. `mask, profile` will set the limit of the velocities where we see our emission line; `options = measure` will perform fitting and give line properties. See Figure 6.1 for the `mbspect` output.



(a)

(b)

Figure 6.1: a) HI emission line. You can see corrected baseline (horizontal line on 0), vertical lines mark part where we see HI emission line, green circles mark position where certain parameters are being measured; b) Properties of the HI emission line: values of the integrated flux (Moment 0); Systemic velocity (Moment 1) and velocity dispersion (Moment 3).

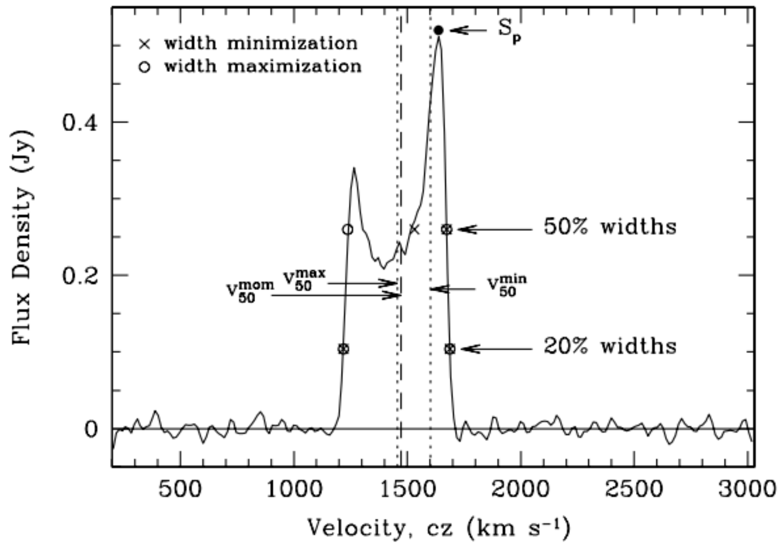


Figure 6.2: HI emission line with marked parameters, Figure is taken from Meyer et al. 2004.

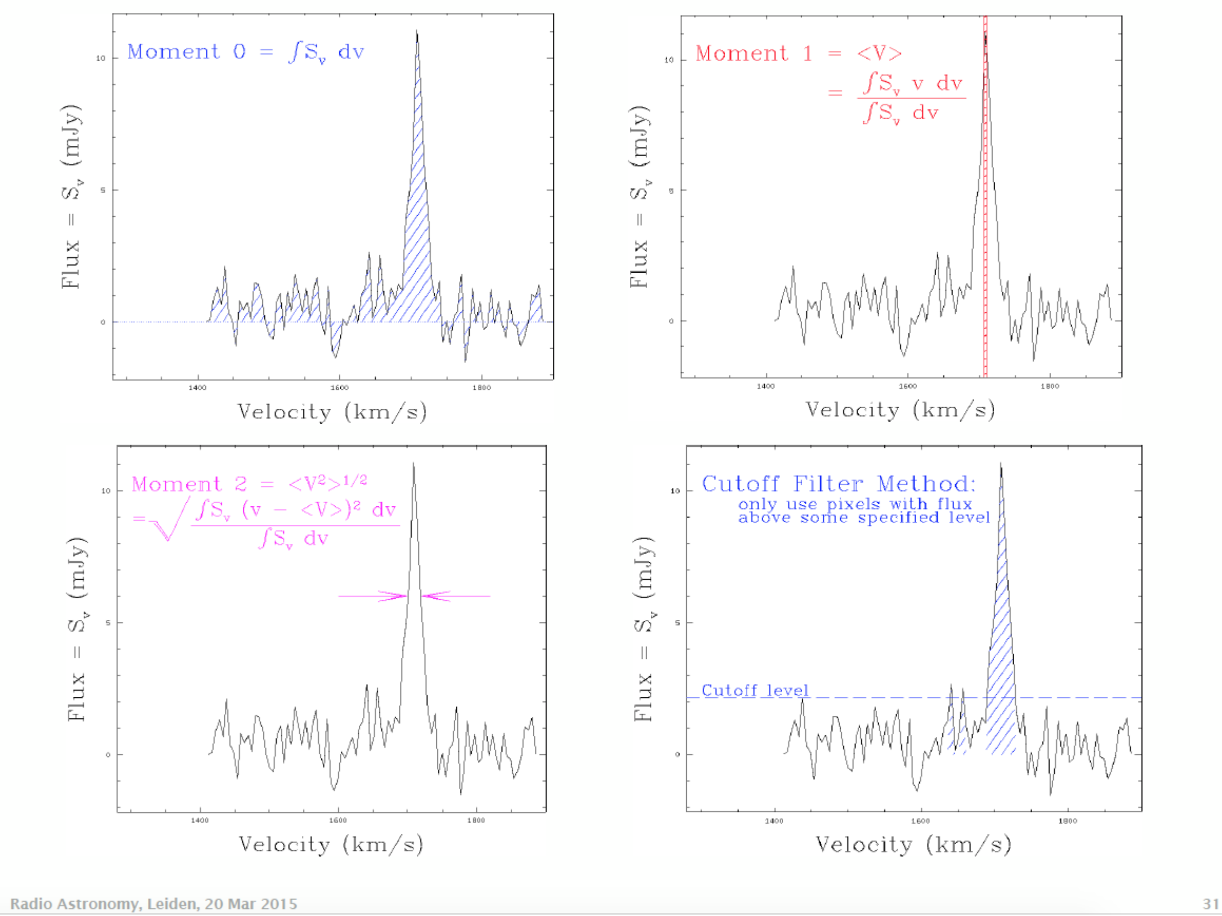


Figure 6.3: Illustration of moment maps. Slide taken from Huib Jan van Langevelde talk ‘Spectroscopy’.

Data analysis in KVIS

We loaded **kvis** with command: `module load karma`; while inside `miriad`, we can open `kvis` by typing: `kvis&` – this will allow us to continue using miriad terminal.

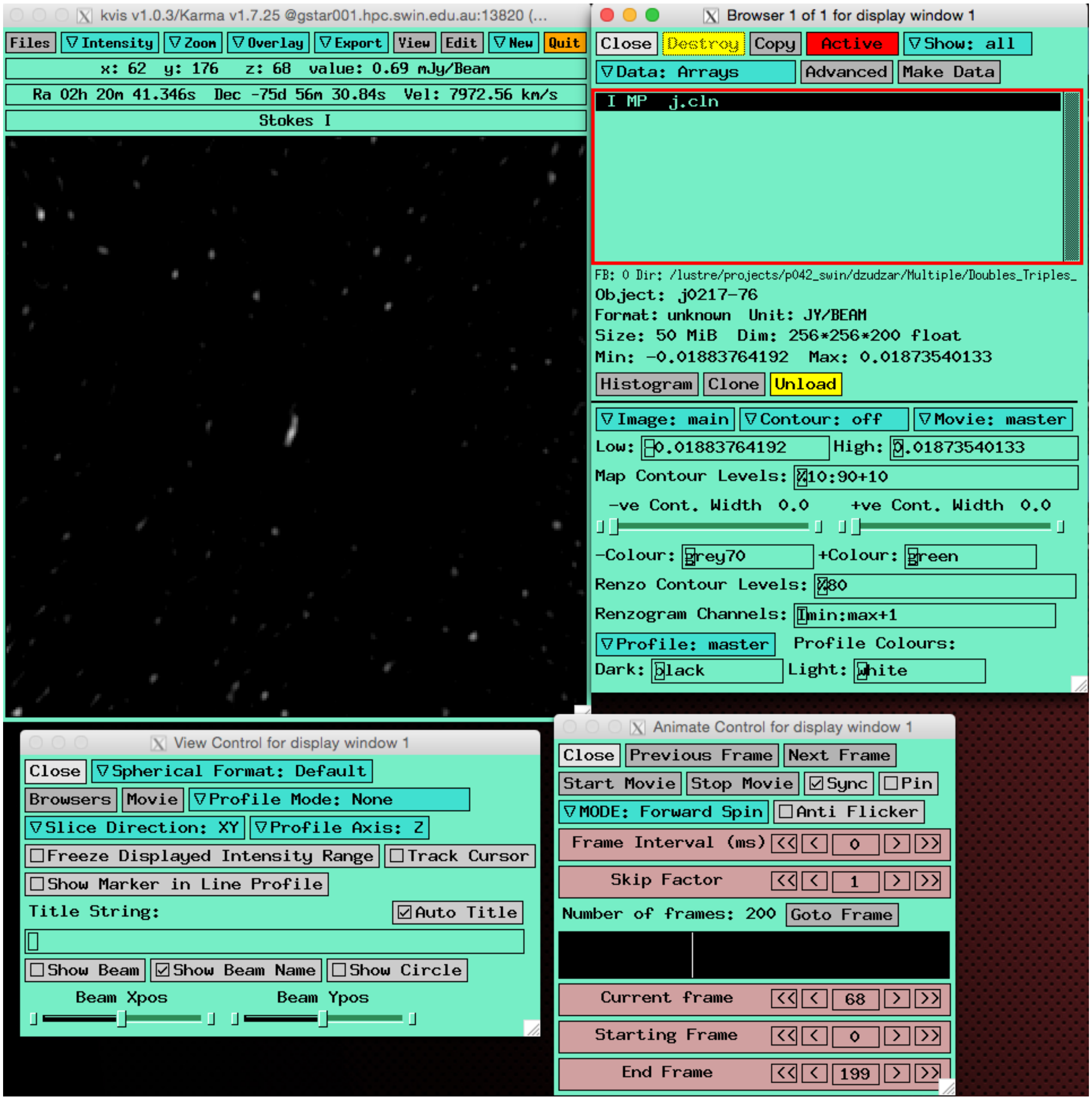


Figure 6.4: `KVIS window`. `File` – open data. `view` – movie: `Start movie` to 'play' datacube. There are plenty of options for inspection the datacube. You can open multiple files and you can see list of them, by clicking on specific file and selecting `image: main` image will show up.

Overplotting images

You can query for example https://skyview.gsfc.nasa.gov/current/cgi/query.pl to find various images, download them as a `fits` file. Open image in `kvis`; set up this image as `main` and set up your e.g. Moment 0 map as `alt` placing `contour map`.

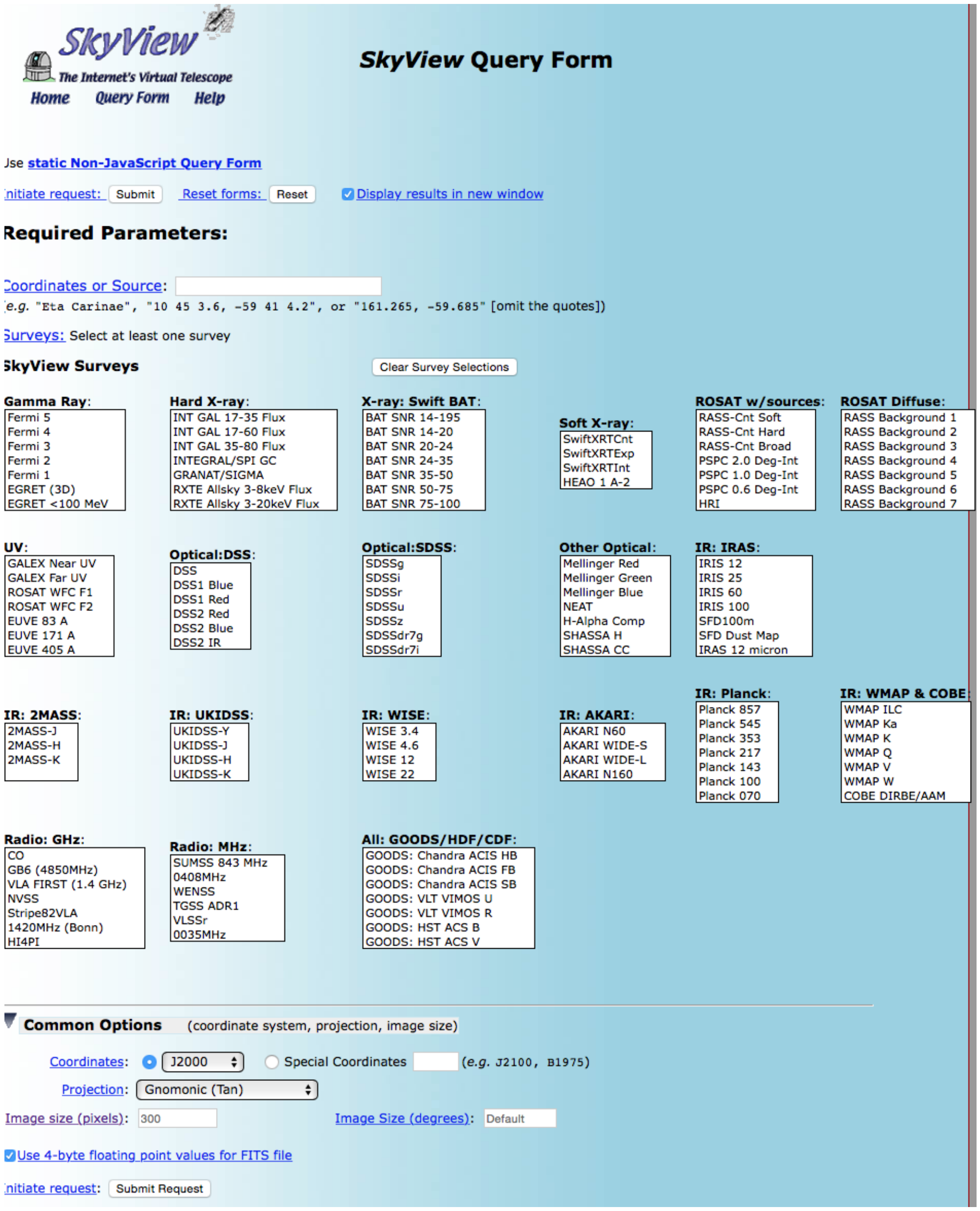


Figure 6.5: SkyView: import coordinates of your source (or name); select with mouse click what type images you want to see (holding `ctrl` button will allow multiple selection); Enter image pixel size (e.g. 1000) and image degree size (e.g. 0.5) and press `Submit Request`

Some formulas

Moment 0:

$$M_0 = \int I(v)dv, \tag{6.1}$$

gives the integrated flux over the spectral line [Jy km s^{-1}].

Moment 1:

$$M_1 = \frac{\int vI(v)dv}{\int I(v)dv}, \tag{6.2}$$

gives the intensity-weighted velocity of the spectral line [km s^{-1}].

Moment 2:

$$M_2 = \sqrt{\frac{\int I(v)(v - M_1)^2 dv}{\int I(v)dv}}, \tag{6.3}$$

gives the velocity dispersion of the gas along the line of sight [km s^{-1}].

Mass was calculated for each galaxy:

$$M_{\rm HI}[{\rm M}_\odot] = 2.356 \times 10^5 \cdot D^2 F_{\rm HI} \tag{6.4}$$

where D is distance to the galaxy group in the units of [Mpc], $F_{\rm HI}$ is integrated flux density in units [Jy km s^{-1}] obtained from the task `MBSPECT`. Relation with redshift is not used in order to be compared with the HIPASS mass.

Other used formula for mass is:

$$M_{\rm HI}[{\rm M}_\odot] = 2.356 \times 10^5 \cdot D^2 F_{\rm HI}/(1+z) \tag{6.5}$$

where z is the redshift, for nearby galaxies: $z \approx v/c$; v – systemic velocity; c – the speed of light.

Assuming that the gas is optically thin, the integrated flux over the spectral line from the Moment 0 map is converted into the HI column density $N_{\rm HI}$ (with this you get HI contours) using:

$$N_{\rm HI} = 1.823 \times 10^{18} \frac{606}{\theta_{\rm min}\theta_{\rm maj}} M_0, \tag{6.6}$$

where $N_{\rm HI}$ is column density in units [cm^{-2}], flux density from the Moment 0 map and $\theta_{\rm min}\theta_{\rm maj}$ is the beam size [arcsec^2].

Radio vs optical definition of velocity

The so-called optical definition of velocity:

$$v_{\rm optical} = c\frac{\lambda - \lambda_0}{\lambda_0} = cz, \tag{6.7}$$

while the so-called radio definition is:

$$v_{\rm radio} = c\frac{\nu - \nu_0}{\nu_0}, \tag{6.8}$$

where ν_0 is the rest frequency of the spectral line, v is the line-of-sight velocity of the source, and c is the speed of light and z is the redshift of the source.

Export/Import data into FITS file

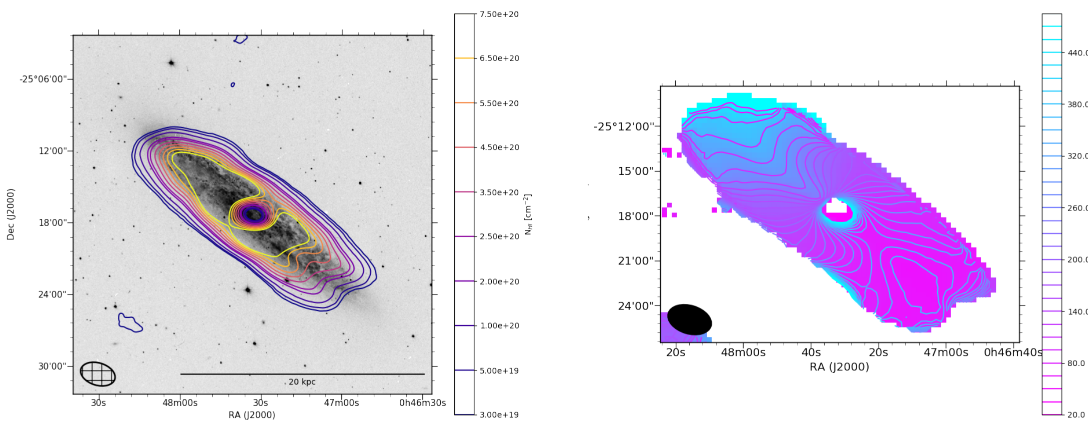
fits%

```
in = image.0mom/
op = xyout
out = image.0mom.fits
```

`op = xyout` means that miriad file will be exported into specified .fits file. If you want to input .fits file into miriad: in = your input file, `op = xyin`, and named miriad file in out.

NGC253

Once you have your final data, you can export everything into `fits` file and then use ALL tools available to do AMAZING stuff, for example Figure 6.6 was made in Python (using astropy `http://www.astropy.org`; matplotlib `http://matplotlib.org/` and APLpy `https://aplpy.github.io/`)



(a) **(b)**

Figure 6.6: a) Example of the column density map overlaid on the optical DSS image; b) Example of the Moment 1 map.

Observations with different baseline configurations

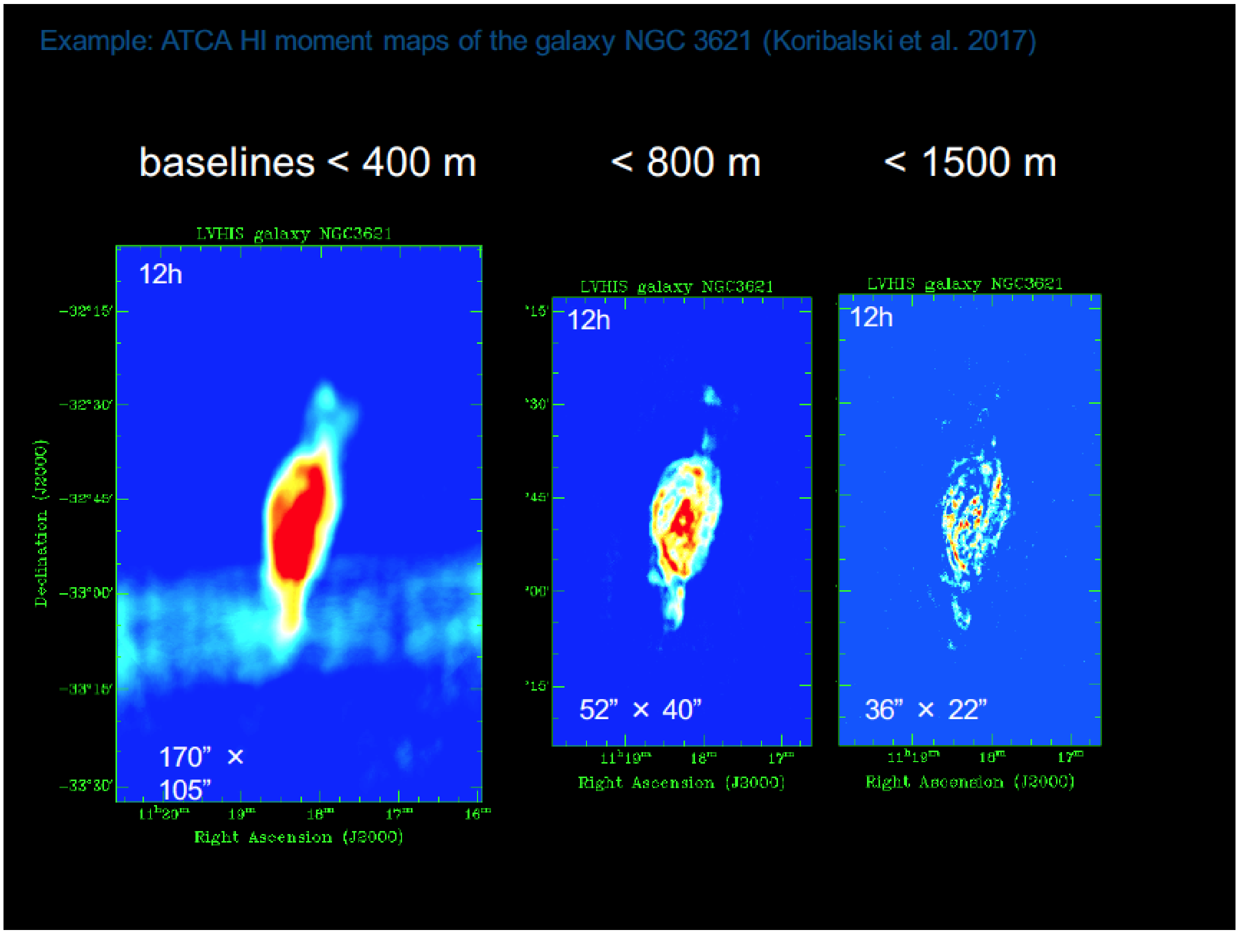


Figure 6.7: With different baselines we are sampling different scales of the observed source. Slide taken from Barbel Koribalski talk 'Overview of 3D Radio Techniques', CSIRO Radio School 2017.

References

Meyer M. J., et al., 2004, MNRAS, 350, 1195

Sault R. J., Teuben P. J., Wright M. C. H., 1995, in Shaw R. A., Payne H. E., Hayes J. J. E., eds, Astronomical Society of the Pacific Conference Series Vol. 77, Astronomical Data Analysis Software and Systems IV. p. 433 (arXiv:astro-ph/0612759)