Radio Interferometry packages and formats

Anita Richards

UK ALMA Regional Centre

JBCA, University of Manchester

really???

MIRIAD

AIPS

GILDAS SOFTWARE

NEWSTAR

GIPSY

DIFMAP

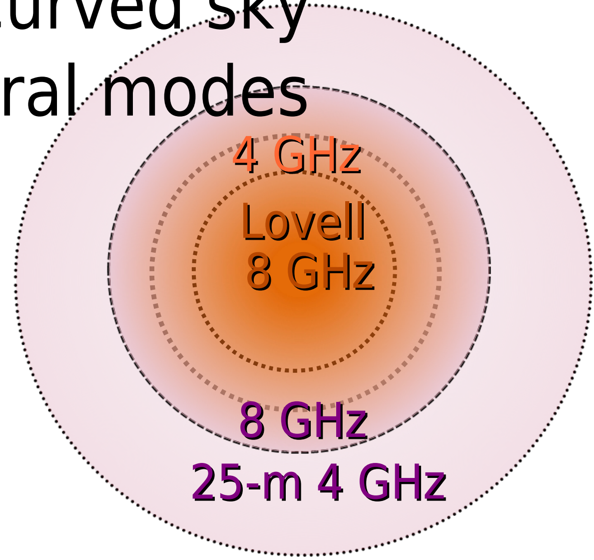
AIPS++

Astronomical Information Processing System

CASA

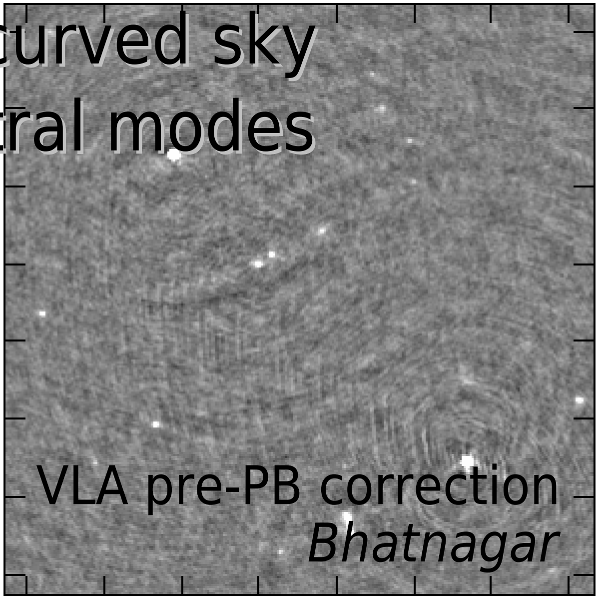
New-generation array demands

- Dishes and dipole arrays; multiple feeds
  - LOFAR Superterp to Global mm VLBI to SKA precursors
- Calibration
  - Instrumental & atmospheric measurements
  - Astrophysical, band-to-band, bandwidth switching
    - Fringe-fitting for first derivatives of phase (delay, rate)
  - Wide fields – direction-dependent calibration
- Wide-field/wide-band imaging
  - Mixed antenna diameters, mosaicing, curved sky
  - Multi-frequency synthesis, mixed spectral modes
  - Narrow channels, short integrations
- Huge raw data volumes
  - Pipelines and parallelisation
  - Automate flagging where possible



New-generation array demands

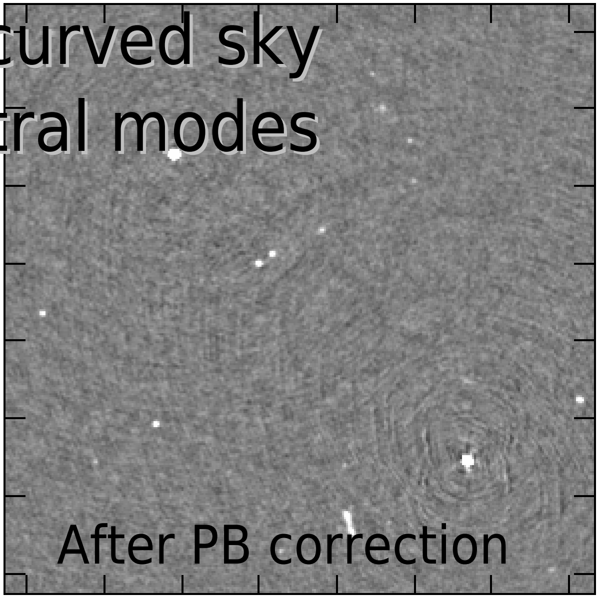
- Dishes and dipole arrays; multiple feeds
  - LOFAR Superterp to Global mm VLBI to SKA precursors
- Calibration
  - Instrumental & atmospheric measurements
  - Astrophysical, band-to-band, bandwidth switching
    - Fringe-fitting for first derivatives of phase (delay, rate)
  - Wide fields – direction-dependent calibration
- Wide-field/wide-band imaging
  - Mixed antenna diameters, mosaicing, curved sky
  - Multi-frequency synthesis, mixed spectral modes
  - Narrow channels, short integrations
- Huge raw data volumes
  - Pipelines and parallelisation
  - Automate flagging where possible



VLA pre-PB correction
Bhatnagar

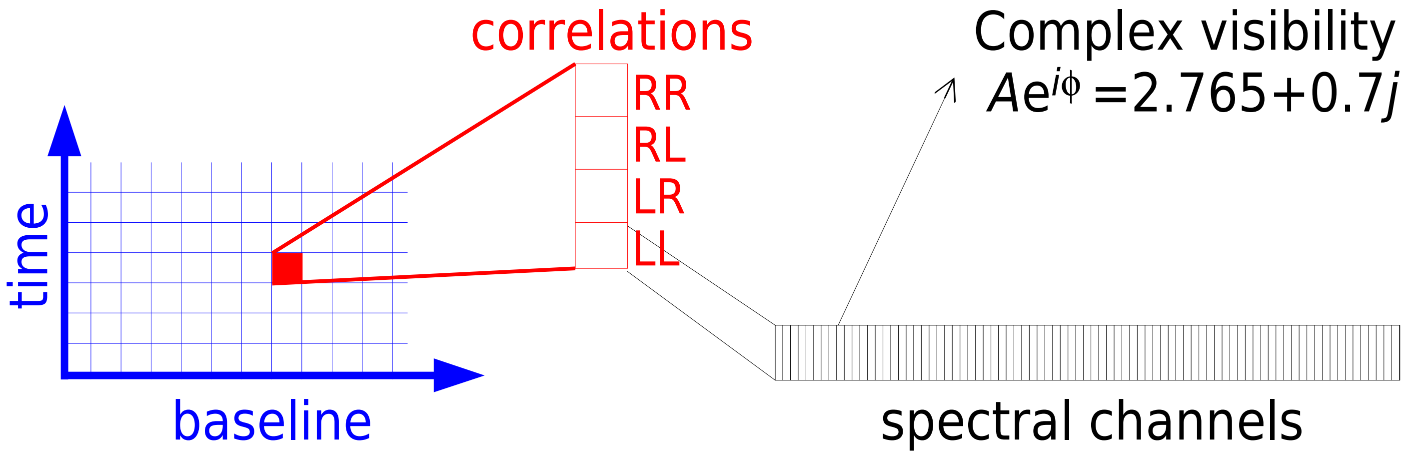
New-generation array demands

- Dishes and dipole arrays; multiple feeds
  - LOFAR Superterp to Global mm VLBI to SKA precursors
- Calibration
  - Instrumental & atmospheric measurements
  - Astrophysical, band-to-band, bandwidth switching
    - Fringe-fitting for first derivatives of phase (delay, rate)
  - Wide fields – direction-dependent calibration
- Wide-field/wide-band imaging
  - Mixed antenna diameters, mosaicing, curved sky
  - Multi-frequency synthesis, mixed spectral modes
  - Narrow channels, short integrations
- Huge raw data volumes
  - Pipelines and parallelisation
  - Automate flagging where possible



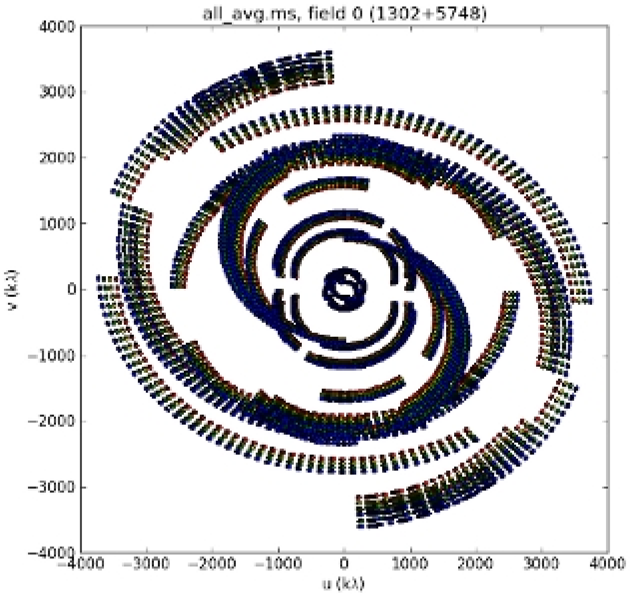
What's in interferometry data?

- Straight from each dish: observatory formats
  - Standardised for VLBI
- Correlated data
  - Time series of complex visibilities



- Off-line storage and processing
  - Associate metadata firmly with data
  - Data should be easy to find and transport

Interferometry data



- Observation axes
  - Time, baseline, pointing direction*, frequency
    - ⇒ *uvw* coordinates
      - *(strictly, phase centre)
- Measured quantity
  - Complex visibility: Amplitude, phase, (weight)
    - per baseline, time interval, spec channel, polarization
- Metadata including
  - Telescope properties
    - Antenna positions, diameters, receivers, subarrays ....
  - Observational info
    - Fields, intents...
    - Weather, Tsys, flagging information ...

Science Data Model (SDM) format

```
uid___A002_X10e085_X148
|-- ASDM.xml
|-- ASDMBinary
|    |-- uid___A002_X10e085_X149
|    |-- uid___A002_X10e085_X14a
|    |-- uid___A002_X10e085_X14b
.
.
|    |-- uid___A002_X10e085_X1e9
|    `-- uid___A002_X10e085_X1ea
|-- AlmaRadiometer.xml
|-- Antenna.xml
|-- CalAmpli.xml
|-- CalBandpass.xml
|-- CalData.xml
|-- CalDevice.xml
|-- CalPhase.xml
|-- CalReduction.xml
|-- CalWVR.xml
|-- ConfigDescription.x
|-- CorrelatorMode.xml
|-- DataDescription.xml
|-- ExecBlock.xml
|-- Feed.xml
|-- Field.xml
|-- Focus.xml
|-- FocusModel.xml
|-- Main.xml
|-- Pointing.bin
|-- Pointing.xml
|-- PointingModel.x
|-- Polarization.xm
|-- Processor.xml
|-- Receiver.xml
|-- SBSummary.xml
|-- Scan.xml
|-- Source.xml
|-- SpectralWindow.xml
|-- State.xml
|-- Station.xml
|-- Subscan.xml
|-- SwitchCycle.xml
`-- Weather.xml

1 directory, 196 files
```

- Native format for ALMA, Jansky VLA etc.
  - Compact, static binary data
  - Accessible xml metadata
    - Human readable (ish)

Data formats

- Correlator output usually archived as SDM or FITS
- SDM compact, easy to transport & access metadata
  - Slow to access binary data – not suitable for analysis
- Convert to FITS or Measurement Set
  - FITS-IDI more usual for storage; UVFITS for processing
    - Venerable, standards diverge (http://fits.gsfc.nasa.gov/)
      - Unstructured stream

```
3C286CPREPOL.SPLAT
SIMPLE  =                    T /
BITPIX  =                  -32 /
NAXIS   =                    7 /
NAXIS1  =                    0 /No standard image just group
NAXIS2  =                    3 /
NAXIS3  =                    4 /
NAXIS4  =                  128 /
NAXIS5  =                    4 /
NAXIS6  =                    1 /
NAXIS7  =                    1 /
EXTEND  =                    T /This is the antenna file
OBJECT  = 'MULTI   '           /Source name
TELESCOP= 'e-MERLIN'           /
INSTRUME= 'VLBA    '           /
OBSERVER= '3C286_MO'           /
DATE-OBS= '2014-01-12'         /Obs start date YYYY-MM-DD
BSCALE  =    1.00000000000E+00 /REAL = TAPE * BSCALE + BZERO
BZERO   =    0.00000000000E+00 /
BUNIT   = 'UNCALIB '           /Units of flux
EQUINOX =      2.000000000E+03 /Epoch of RA DEC
VELREF  =                    3 />256 RADIO, 1 LSR 2 HEL 3 OBS
OBSRA   =    2.02784500000E+02 /Antenna pointing RA
OBSDEC  =    3.05091600000E+01 /Antenna pointing DEC
```

```
CTYPE2  = 'COMPLEX '           /1=real,2=imag,3=weight
CRVAL2  =    1.00000000000E+00 /
CDELT2  =     1.000000000E+00 /
CRPIX2  =     1.000000000E+00 /
CTYPE3  = 'STOKES  '           /-1=RR, -2=LL, -3=RL, -4=LR
CTYPE4  = 'FREQ    '           /Frequency in Hz.
CTYPE5  = 'IF      '           /Freq. group no. in CH table
CTYPE6  = 'RA      '           /Right Ascension in deg.
CTYPE7  = 'DEC     '           /Declination in deg.
PCOUNT  =                    9 /
PTYPE1  = 'UU---SIN'
PSCAL1  =    1.90646864812E-10 /
PTYPE2  = 'VV---SIN'
PSCAL2  =    1.90646864812E-10 /
PTYPE3  = 'WW---SIN'
PSCAL3  =    1.9064686
PTYPE4  = 'DATE    '
PZERO4  =    2.4566695
PTYPE6  = 'BASELINE'
PTYPE7  = 'SOURCE  '
PTYPE8  = 'INTTIM  '
PTYPE9  = 'CORR-ID '
```

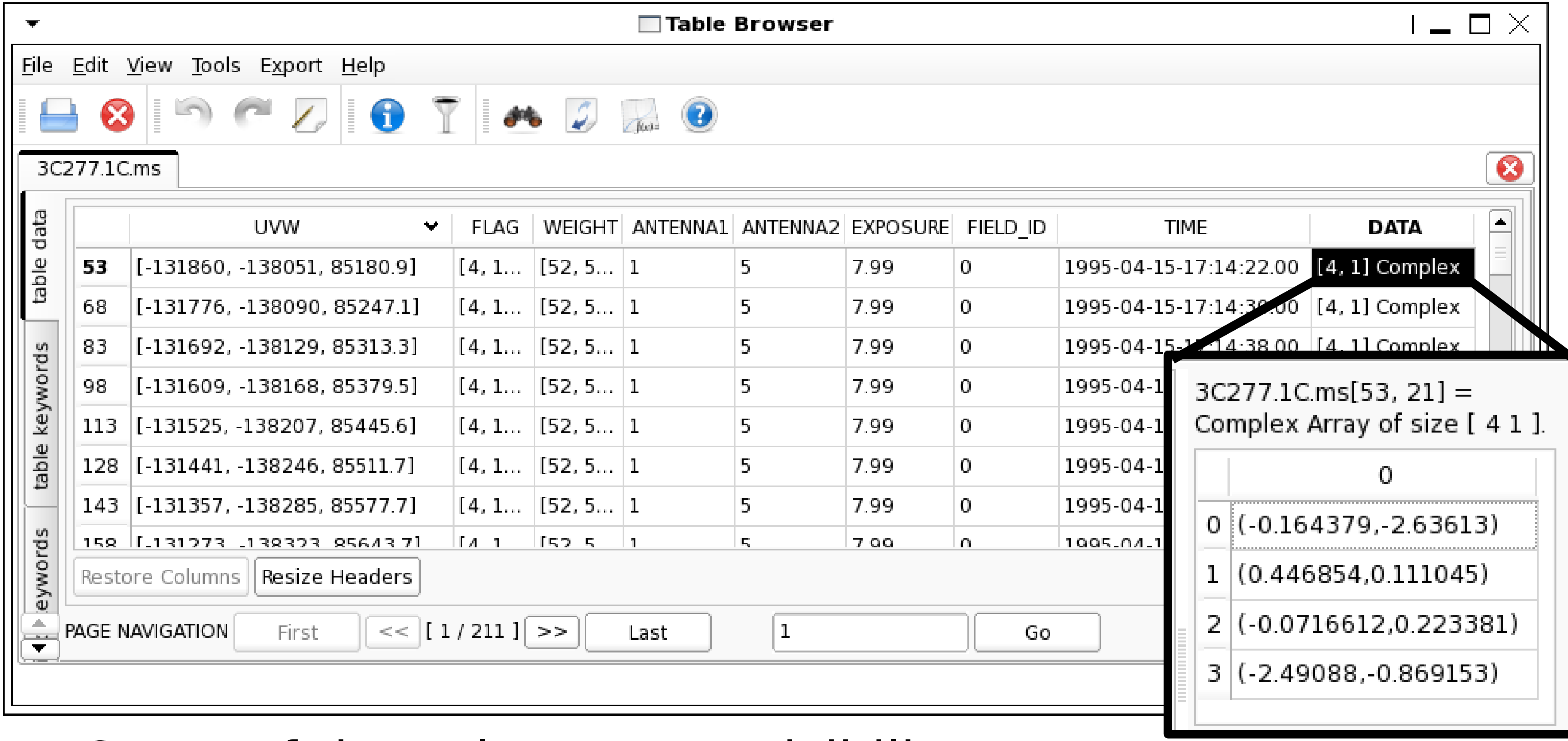
FITS The Astronomical Image and Table Format
Flexible Image Transport System

Measurement Set visibility data

- Directory of Tables
  - MAIN table
    - Binary visibilities
  - Observational properties
  - Metadata
- Similar format for images
- Easy to access
  - Human or software
- http://casa.nrao.edu/Memos/229.html

```
> tree jupiterallcal.split.ms
jupiterallcal.split.ms
|-- ANTENNA
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- DATA_DESCRIPTIO
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- FEED
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   `-- table.lock
|-- FIELD
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   `-- table.lock
|-- FLAG_CMD
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- HISTORY
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- OBSERVATION
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- POINTING
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.f1
|   |-- table.info
|   `-- table.lock
|-- POLARIZATION
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   `-- table.lock
|-- PROCESSOR
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- SOURCE
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   `-- table.lock
|-- SPECTRAL_WINDOW
|   |-- table.dat
|   |-- table.f0
|   |-- table.f0i
|   |-- table.info
|   `-- table.lock
|-- STATE
|   |-- table.dat
|   |-- table.f0
|   |-- table.info
|   `-- table.lock
|-- table.dat
|-- table.f0
|-- table.f1
|-- table.f2
|-- table.f2_TSM1
|-- table.f3
|-- table.f3_TSM1
|-- table.f4
|-- table.f5
|-- table.f6
|-- table.f6_TSM0
|-- table.f7
|-- table.f7_TSM1
|-- table.f8
|-- table.f8_TSM1
|-- table.info
`-- table.lock
```

Measurement Set MAIN table



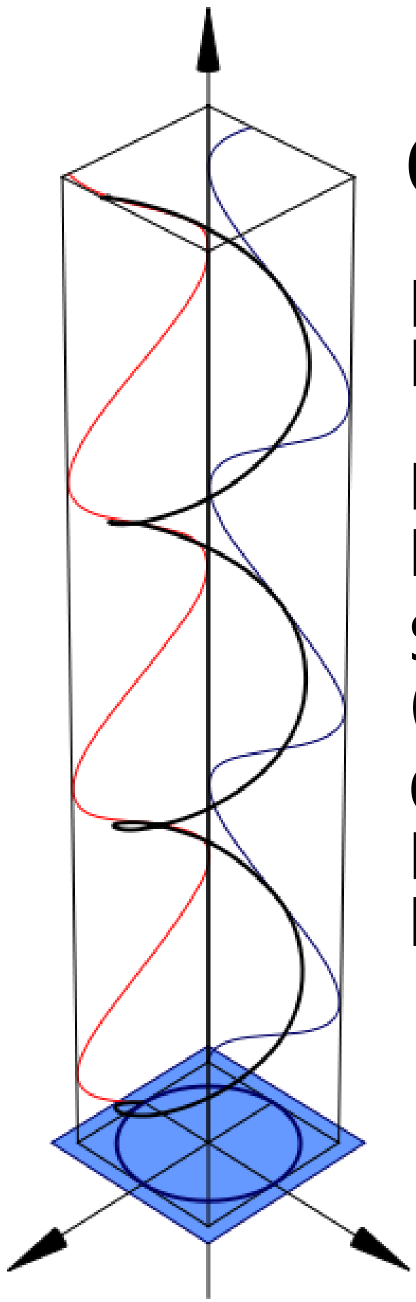
- Some of the columns per visibility
  - **Data**: Complex value for each of 4 correlations (LL RR LR RL) per spectral channel
    - Inspect in CASA `browsetable` or write to file

Visibility data: Measurement Set format

MAIN	Model, e.g.:	Corrected data	Flags
DATA *Original visibilities*	*FT of image made from MS* *FT of supplied model image* *FT of point flux density*	*Copy of visibilities with calibration tables applied* (Used in imaging not calibration)	(Edits are stored here first; backup tables can be made and used to modify)

- Unix-like directory structure with binary data and ascii metadata files arranged in subdirectories
- Additional tables in MS and free-standing:
  - *Admin*: Antenna, Source etc.
  - *Processing*: calibration, flags, etc.

Polarization jargon



CIRCULAR

Left-hand
LHC, L, LL

Right-hand
RHC, R, RR

Stokes V =
(RR-LL)/2

Cross hands
LR RL make
linear

LINEAR

Stokes Q = (RL + LR)/2

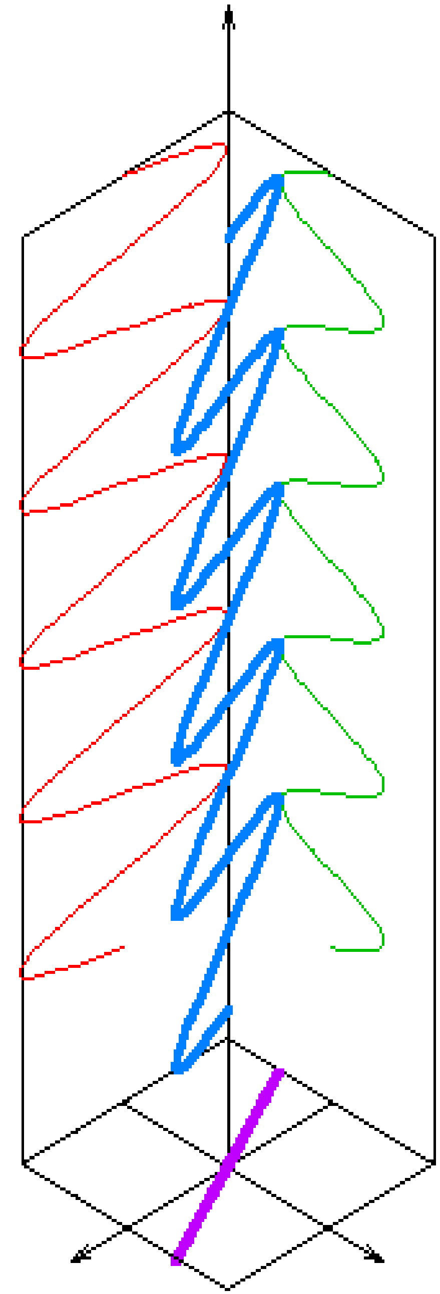
Stokes U = (RL - LR)/2i

Polarized intensity
$P = \sqrt{(Q^2 + U^2 + V^2)}$

Polarization
angle χ = ½ atan2(U/Q)

Linear feeds X,XX, Y,YY

Cross hands XY YX
make circular



Stokes I = (LL + RR)/2 = (XX + YY)/2

beware, some packages' definitions differ in sign or by a factor of 2

Diagrams thanks to Wikipaedia

FITS axes labels

- Axes contain one+ pixels
- Quantization of physical variable e.g.
  - Position in RA
  - Frequency
  - Label
    - Types of polarization ⇒
      - I (one 'pixel')
      - IQUV (four 'pixels')

CASA

- Polarizations also termed correlations

Polarization type	Label	FITS code
Total	I	1
Linear	Q	2
Linear	U	3
Circular	V	4
Circular	RR	-1
Circular	LL	-2
Linear	RL	-3
Linear	LR	-4
Linear	XX	-5
Linear	YY	-6
Circular	XY	-7
Circular	YX	-8
Undef	UNDEF	---
Linear	POLI	5
Linear	POLA	6

Data Reduction Package requirements

- Import data and metadata
- Derive and apply calibration
  - Tables provided with data, may need converting
    - e.g. Tsys
  - User-supplied parameters
    - e.g. antenna position corrections
  - Calibration derived from observed sources
    - Bandpass, phase-reference, self-calibration...
- Display and report data properties
  - Flagging
- Fourier transform and produce images and spectra
  - Clean algorithm, maybe Maximum Entropy
  - Mosaicing, weighting/tapering ...
- Image analysis

Data Reduction Package requirements

Importing, instrumental calibration

Observatory-specific, probably needs particular package

Astrophysical calibration and imaging

Needs suitable interferometry package

Image analysis

Can export as FITS and use many types of software

Other considerations

What do your collaborators use?

Speed, ease of installation/maintenance/**scripting**

CASA developed for ALMA and (J)VLA

- `aips++` development in `c++` started in ~1994
  - Intended to be to maintain/develop/parallelise
- User-friendly python wrapper since 2007
  - *C*ommon *A*stronomy *S*oftware *A*pplication
    - Development by NRAO, ESO etc.
  - 'Task' interface or scripting
    - Underlying `aips++` toolkit available
      - Basis of LOFAR etc. software
- Measurement Set data format
  - Imports SDM, FITS, old VLA, GMRT and Miriad format
  - Exports SDM, FITS (images or UVFITS)
    - Apply all flags and calibration before inp/exporting
  - Use `tar` on MS directories for transport

CASA

CASA capabilities

- CASA supports all usual off-line data reduction for ALMA, VLA
  - Calibration, imaging, mosaicing, related single-dish
  - Handles mixed bandwidths (line+continuum)
  - Linear or circular polarization
  - Support for (sub-)mm line identification
    - Viewer good for extracting spectra from cubes
  - Limited image analysis and publishable plotting
- Also used for many other interferometers
  - LOFAR, GMRT, e-MERLIN, ATCA, SMA, VLBI
  - May have to start in 'native' package
    - Support for full fringe-fitting under development
    - Accurate primary beam corrections need extension of capabilities to use observatory-supplied beam patterns

Libraries use Measurement Equation

$$\underline{V}_{ij} = \mathbf{M}_{ij}\mathbf{B}_{ij}\mathbf{G}_{ij}\mathbf{D}_{ij}\int \mathbf{E}_{ij}\mathbf{P}_{ij}\mathbf{T}_{ij}\mathbf{F}_{ij} S \underline{I}_{\nu}(l,m) e^{-i2\pi(u_{ij}l+v_{ij}m)} dl\,dm + \underline{A}_{ij}$$

Vectors

Starting point

$\underline{V}$ isibility = *f(u,v)*

Goal

$\underline{I}$ mage

$\underline{A}$dditive baseline error

Scalars

Methods

S (mapping $\underline{I}$ to observer polarization)

l,m image plane coords

u,v Fourier plane coords

i,j telescope pair

Jones Matrices

Hazards

Multiplicative baseline error

Bandpass response

Generalised electronic gain

Dterm (pol. leakage)

E (antenna voltage pattern)

Parallactic angle

Tropospheric effects

Faraday rotation

Using the Measurement Equation

- *Hamaker, Bregman & Sault 1996*
  - Decompose into relevant calibration components e.g.
- $\underline{V}_{ij}^{obs} = \mathbf{B}_{ij}\mathbf{G}_{ij}\mathbf{D}_{ij}\mathbf{P}_{ij}\mathbf{T}_{ij}\mathbf{F}_{ij}\underline{V}_{ij}^{ideal}$
  - Chose one (or a few) at a time
    - Usually solve fastest-varying first
      - (so averaging over slower-varying)
  - Compare data with model or idealisation
    - Linearise and solve by χ^2 (or other) minimization
  - (AIPS etc. use similar algorithms for gain calibration)
- Visibility data are stored in Measurement Sets
  - Accessible directories of tables

Starting CASA

- See web links for downloads (or http://casa.nrao.edu)
  - Don't forget the Cookbook!
- Start like `<path>/casa-release-4.4.0-el15/casapy`
  - You can set up an alias or whatever is convenient
    - *Don't reduce data inside the CASA installation!*
  - This starts the iPython environment
    - Interactive input to tasks in the xterm
    - Logger (see toolbar for display, export options)
  - Access to shell
    - Direct simple commands e.g. `ls`
    - Prefix any shell command with ! e.g. `!more *py`
- Python
  - Case sensitive
  - Zero indexed (e.g. 27 antennas numbered 0~26)
    - **Run any scripts or functions you want**

CASA

Using CASA

- `default gaincal` resets default values
- `help('gaincal')` for more details
- Simplest input to tasks is `parameter=value`
  - In this mode, variables are global
    - `solint='1min'` will appear in all tasks until reset
  - Check/view parameters with `inp gaincal`
  - Run task by typing `gaincal`
    - `saveinputs(gaincal,'gctry1.save')` saves
    - `execfile('gctry1')` restores
      - `gctry1.save` is text file, view using `!more gctry1`
- For scripting, use:
  - `gaincal(vis='super.ms', field='supernova', caltable='super.p1', solint='1min')`
    - Now variables are always default unless set

Astronomical Image Processing System

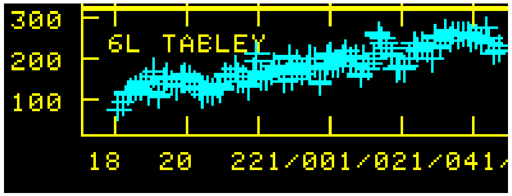
- Originated by NRAO for VLA in 1978
  - Fortran, C
  - Limited built-in scripting/math operations
  - Historically most widely used package for cm-wave
    - VLA, MERLIN, VLBI ... many more interferometers
    - Some support for single dish and any FITS images
  - Very wide functionality from calibration to analysis
- Specialised VLBI calibration and elderly formats
- Many sophisticated image analysis tasks
  - Lovely postscript plots for publication
- Python wrapper (Parseltongue) for easier scripting

AIPS jargon

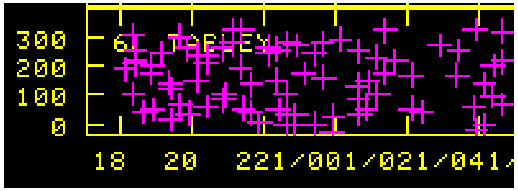
- Major operations are performed using Tasks
  - FITLD loads data, CALIB performs calibration etc.
- Input parameters to Tasks are set by Verbs
  - >Task 'CALIB'; CALSOUR 'MKN273'; SOLINT 1
  - Words/names in 'inverted commas'; numbers bare
  - *Not* case sensitive, in general
  - Inside AIPS, 12-character limit on file/source names
- To set all defaults: >RESTORE 0
  - Beware: will give values typical for VLA data
    - You will have to set parameters suitable for your data
- To exit and kill all AIPS windows: >KLEENEX

Keep sight of the physics

- Brain gets filled with package jargon
  - `task 'CALIB'; calsour 'phaseref'; solint 0.5; docal 100; aparm(7) 3; gainuse 5; solmo 'p'`
- Rember this means
  - Take the visibility data for the phase ref and apply existing calibration table 5; minimum snr 3
  - If no other model is given, a point source at the field centre will be used
  - Compare the data with the model phase and calculate the corrections needed

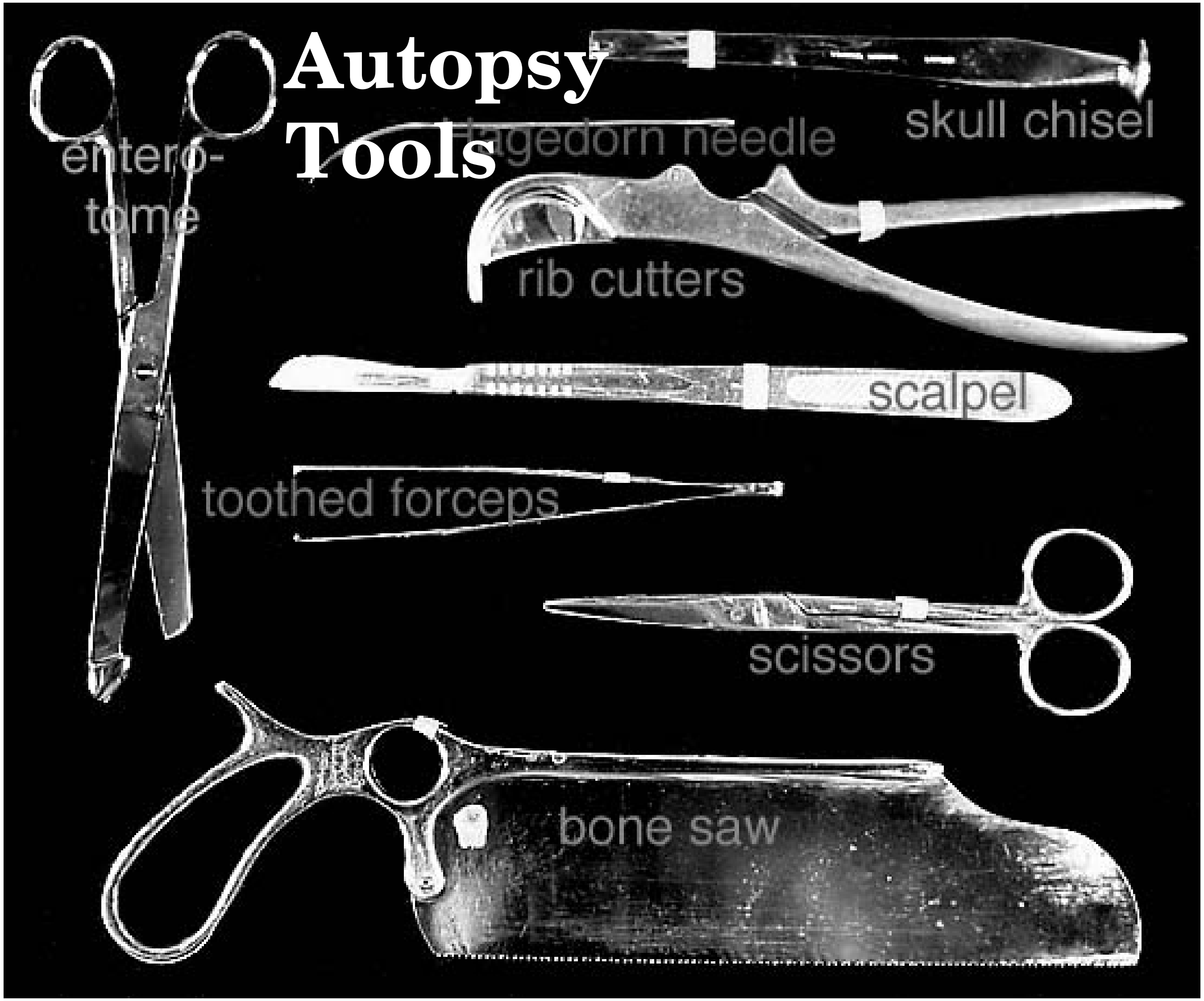


- That way you will know to expect
  - and what to check if you get
    - If it looks rubbish, it probably is!



Keep a full processing history

- Use scripts, or
- Note parameter values
  - Examples for further processing
  - Troubleshooting postmortem



An experienced radio astronomer