Using AIPS++ for ATCA data

Paul Jones, ATNF

Note: based on an article from the February 2003 ATNF Newsletter.

Introduction

Many ATNF users will have heard of the AIPS++ Project to develop a new flexible package for radio astronomy data reduction and analysis. See the local version of the web pages at
http://www.atnf.csiro.au/computing/software/aips++/release/docs/aips++.html
(I will contract this web address to form .../docs/aips++.html for other references to these web pages).

Why should you want to use AIPS++ ? There are several reasons
1) it has capabilities that are not in other packages,
2) you can script new functions by combining lower level tools in glish,
3) it is likely to be the package for the new generation of radio telescopes, such as the GBT, LOFAR and ALMA, and
4) it is designed to have at least the functionality of AIPS and Miriad.
See the mission statement at .../docs/project/mission.html .

The AIPS++ Project has been going for a long time, but AIPS++ is not yet, in practice, delivering the full functionality, for example, of Miriad, for routine end-to-end processing of ATCA data. This is a bit of a "chicken and the egg" what-comes-first problem, as it is only when real data are used that bugs are found and users suggest what enhancements they need. The recent AIPS++ Technical Review suggests steps to fix this, so that AIPS++ will be able to handle common ATCA use modes end-to-end. Thus it is appropriate for new users of the ATCA to consider using AIPS++ in the future.

The current status for ATCA data, from tests that I have done is this: I have (see below) successfully taken ATCA data from the RPFITS data file, into AIPS++, displayed, flagged and calibrated the uv-data, made an image and deconvolved the image (Figure 3). I have done this for continuum data, and line data making a data cube, but the cube has channel or frequency planes (Figure 4), rather than velocity, as I have not yet been able to do the rest frequency/LSR correction.

It is not necessary, however, to do all steps of data reduction and analysis in AIPS++, so that even if there are still gaps, there are a lot very useful tools which ATNF users should find useful. You could, for example, flag and calibrate the ATCA data in Miriad, then transfer as a FITS file to AIPS++ for imaging and analysis. The "viewer" tool for display and analysis of images/data cubes is a useful stand-alone application.

There is a lot of documentation of AIPS++ on the Web, mirrored to many different sites of the AIPS++ (ex-)Consortium, the local version being
.../docs/user/documentation.html.
I don't want to summarise the information here, but rather to give a quick overview and a few hints, to help ATNF users to try AIPS++ particularly if they have a specific application for which it would be useful.

General AIPS++

AIPS++ consists of a range of "tools" with "functions" to operate on the data. The most basic level functions are written in C++, but this is transparent to the user, as they are combined together using the glish programming language into the higher level functions and tools. The tools are bundled together into "modules" and "packages" (for example the "imager" tool in the "imager" module in the "synthesis" package).

AIPS++ can be run from the Command Line Interface (CLI) as a series of glish commands, or for common data processing using Graphical User Interfaces (GUIs). I have mostly used the GUIs to do standard processing step-by-step, but for more efficient batch processing the CLI enables a script of glish commands to be run. The tools need to be loaded, and if you are going the GUI way, this can be done from the "tool manager" GUI.

The document "Getting Started with AIPS++" (.../docs/user/gettingstarted.html) is a good reference, and a place to start. However, I would jump quickly to the section "A Practical Demonstration: Making an Image from a visibility dataset" to run the test data. This runs through loading and plotting a uv dataset, making an image, cleaning it and displaying it, much as I describe below. I find actually using the tools on data a much better way to learn, than reading the manual.

The main AIPS++ data format is called a "measurement set" (MS) with a main data table and subsidiary tables in subdirectories (somewhat like Miriad). Most of the tools in AIPS++ are quite general, and not telescope dependent. The ATNF specific tools are in the package "atnf", and includes the module "multibeam" (for the Parkes Multibeam data) and "atcafiller". ATNF data from the Compact Array, Parkes, Mopra and the LBA VLBI correlator are written in RPFITS format, a local variant of FITS, see http://www.atnf.csiro.au/computing/software/rpfits.html .
Therefore, the ATCA rpfits data needs to converted to the AIPS++ MS, with the tool "atcafiller".

Compact Array data

The document "Getting Results with AIPS++, Volume 3 - Telescope Specific Processing, ATCA Reduction" (by Mark Wieringa, .../docs/gettingresults/grvol3/node6.html) is good as a summary of the steps to reduce your ATCA data, with more detail than given here. However, you may also need to refer to the document "Getting Results with AIPS++, Volume 2 - Generic Processing" (.../docs/gettingresults/grvol2/grvol2.html) particularly "Chapter 1 - Synthesis Calibration", and "Chapter 2 - Imaging, Deconvolution and Self-Calibration" for more details.

I have used test ATCA data here from project C1080, which is 6 cm continuum and formaldehyde line absorption of G291.3-0.7 (NGC 3576, RCW 57A). This used the EW367 array, so the 6 km baselines were not used for imaging leaving only 5 antennas. Also, there were instrumental problems on antenna 5 (one polarisation was bad on one IF so CACAL was not applied on this antenna/IF at the start, giving delay errors and a phase gradient across the band, plus a phase step on all baselines when the problem was fixed, and CACAL applied). This was not the best data to use, but gave a good check on the flagging and calibration tools.

The main tools used are "atcafiller", "msplot", "calibrater", "imager" and "image/view". The data analysis steps are quite similar to AIPS or Miriad, so I give the Miriad equivalent tasks in brackets, for those who have also done data reduction in Miriad.

• The RPFITS data are loaded into AIPS++ and converted to a measurement set with "atcafiller" (like Miriad atlod).

• The uv-data are plotted and examined with "msplot". This is a very flexible tool to plot uv-coverage, or any two parameters X-Y eg. amplitude/phase vs time/frequency etc (like uvplt and uvspec). I used this tool to flag the data (edit = T, when starting) by highlighting outlying bad data on the screen (like tvflag). One slight annoyance to me was that the data could be displayed as an average over channels, but could not be flagged in this mode. An alternative way to flag data is "flagger" by specifying the time, baseline etc (like uvflag). Also AIPS++ has automated editing methods in the "autoflag" and "flagger" tools.

• The calibration of the uv-data is done with the tool "calibrater". This uses the formalism of Jones matrices (no relation) with calibration tables for B (bandpass), G (gain) and D (polarisation leakage) determined from the calibrator data and applied to the other sources. This "calibrater" tool is equivalent to the Miriad tasks mfcal, gpcal, gpplt, gpboot and gpcopy, but applied with a sequence of functions within the tool. The measurement set (MS) within AIPS++ has multiple columns in the MAIN MS table for DATA, CORRECTED_DATA and MODEL_DATA, so application of the calibration tables modifies the CORRECTED_DATA column. The MODEL_DATA for the calibrators, at this calibration step, is a point source, but when the MS is used to make deconvolved images (CLEAN, MEM etc), the image model is transformed to visibilities and put into this column. This can be used in "calibrater" for self-calibration of the source in conjunction with "imager".

• The tool "imager" is used to make the images from the MS visibility data. The dirty image and beam (point spread function, psf) can be made with the function "makeimage" (like Miriad invert). The image could be then deconvolved with the tool "deconvolver", but it is easier to use "imager" functions, which integrate the inversion and deconvolution into one step, such as the function "clean" (like Miriad invert, clean and restore). Parameters for the data selection and image are set with

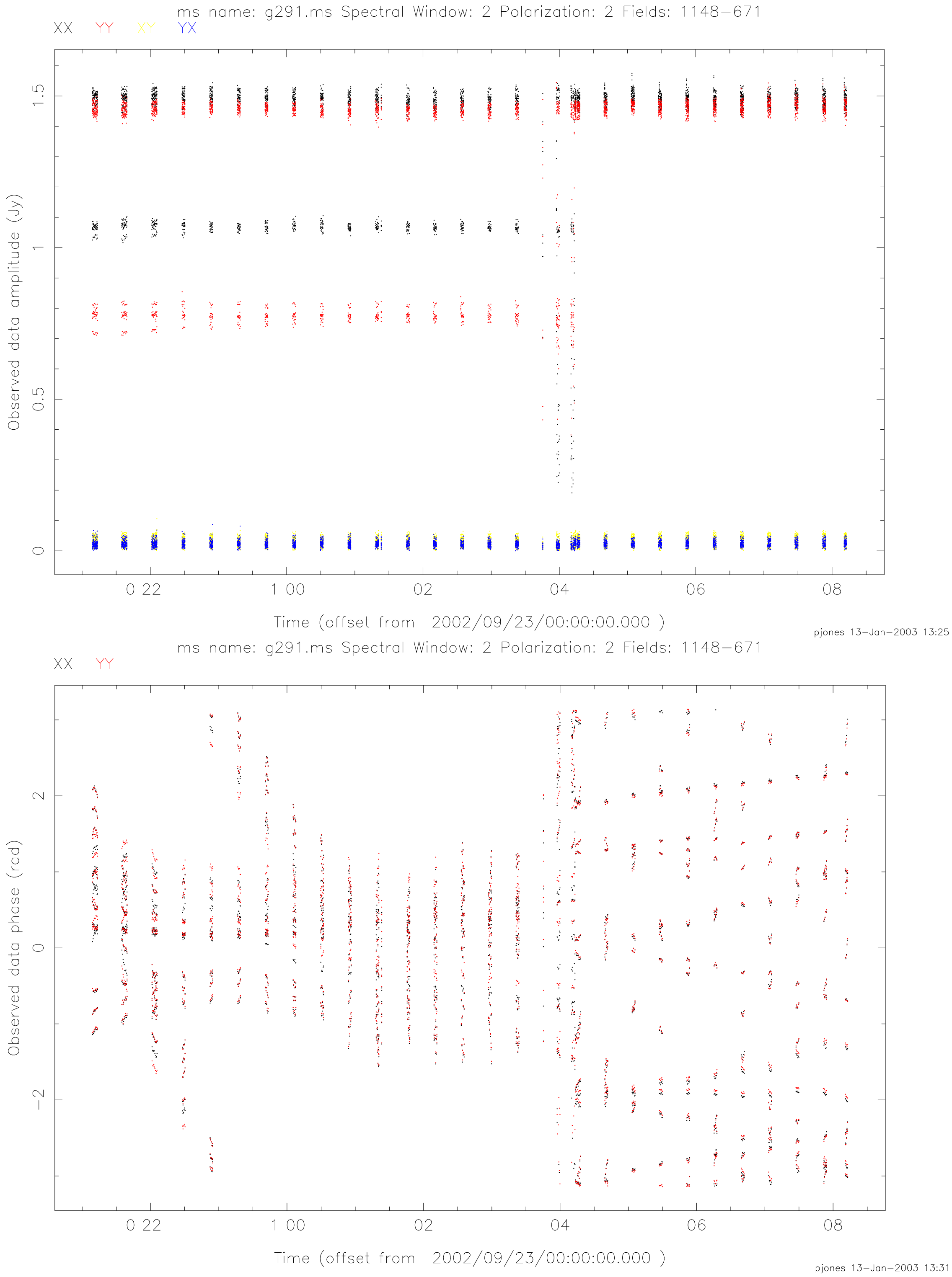
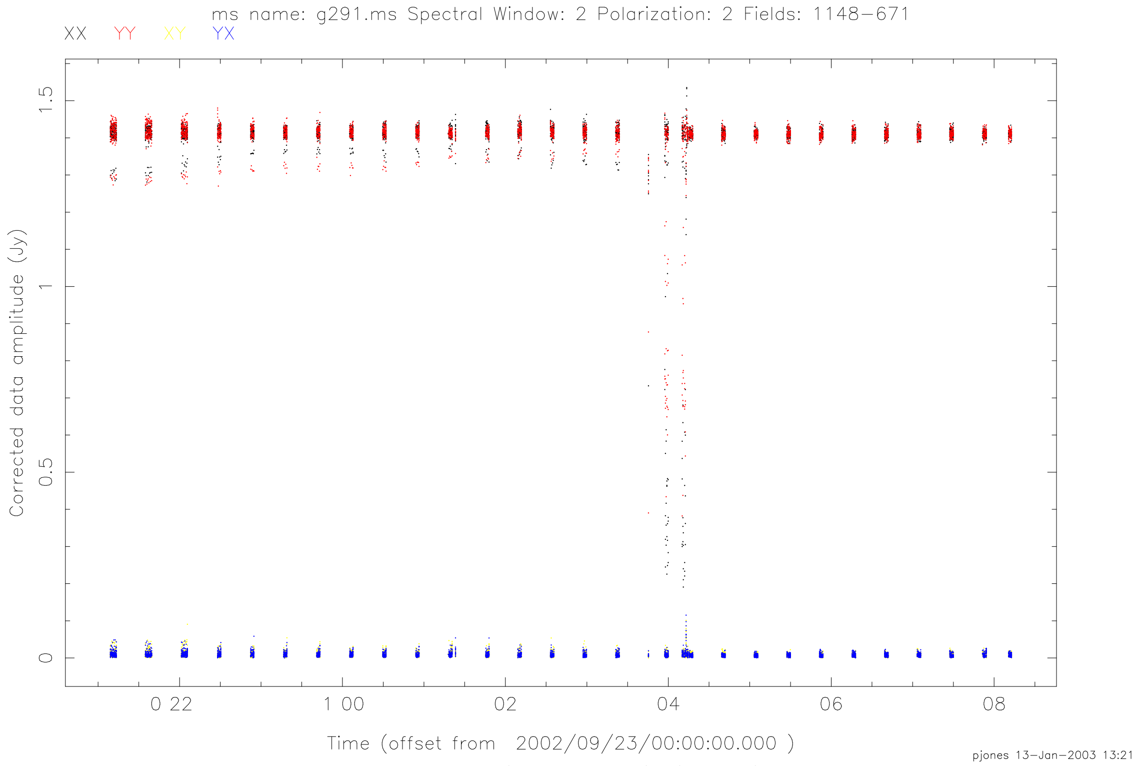


Figure 1: The uncalibrated amplitude and phase for calibrator 1148-671 shown with tool "msplot".



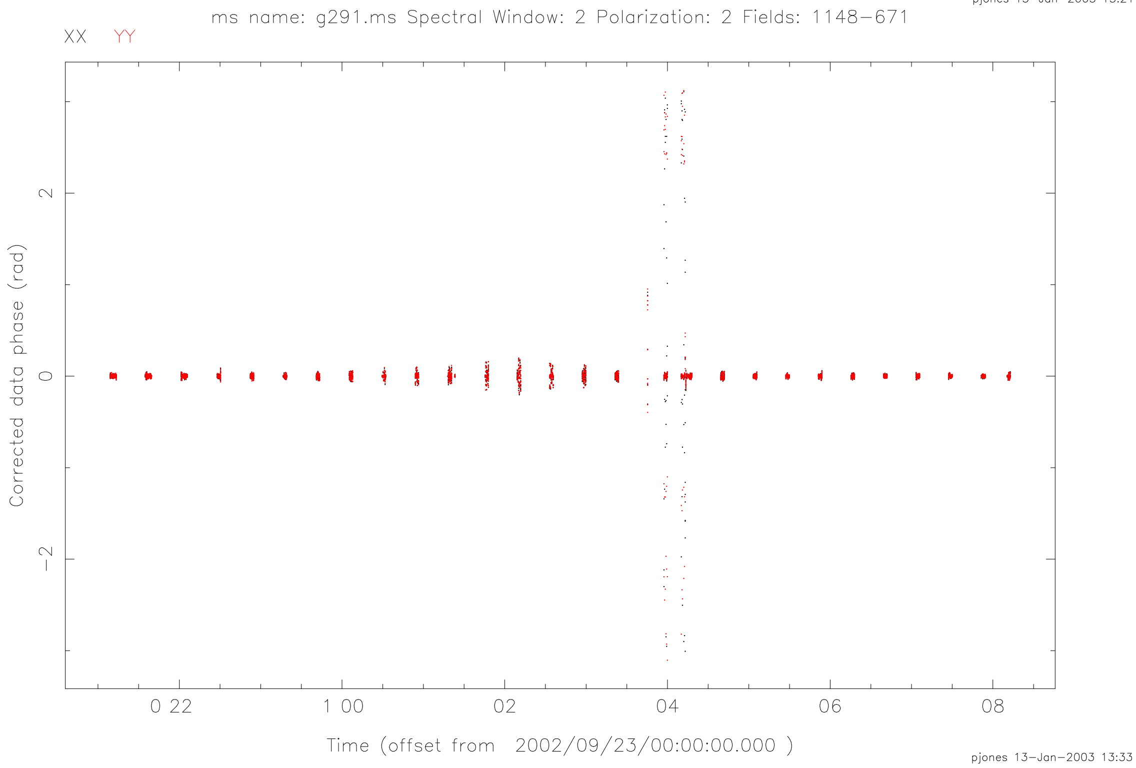


Figure 2: The calibrated amplitude and phase for calibrator 1148-671 showing the flagging and calibration.

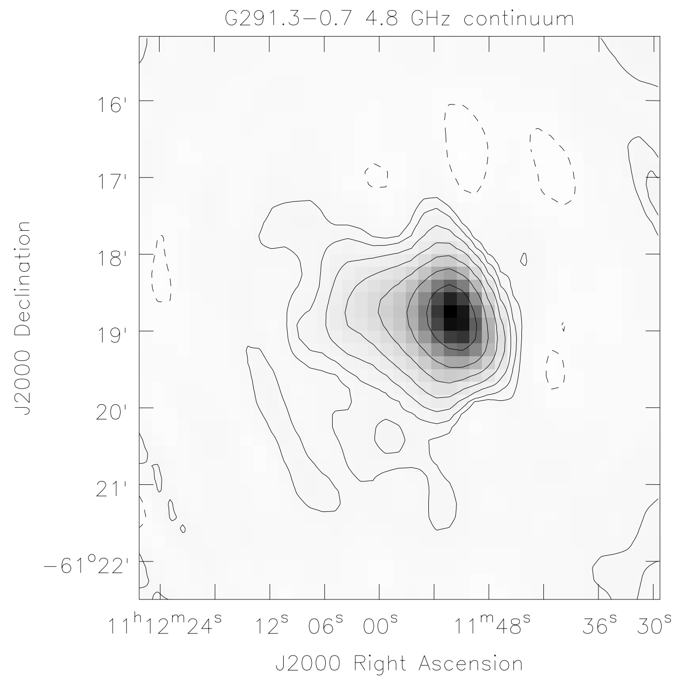


Figure 3: The ATCA 6 cm continuum I image of G291.3-0.7 produced with AIPS++ end-to-end. The contours are -0.25, 0.25, 0.5, 1, 2, 4, 8, 16 Jy/beam.

functions “setdata” and “setimage”. There are a lot more functions that can be used to set weighting, tapering, clean regions etc.

• The function “view” in the tool “image” can be used to display the images. This comes up with a default grey-scale image in a GUI, which can be zoomed and the lookup table fiddled, and an adjust window pulled up to change the colour table etc. I like to display contours on the grey-scale (figure 3), so I added a contour overlay of the same data, with parameters specified in a second adjust window. This is all standard functionality (like kview or Miriad cgdisp) but quite nice, when you get used to it. You can also get these functions with “viewer” from the unix/linux command prompt, rather than going through AIPS++ “image” tool.

Comments

So how good are the final images ? For the C1080 data I used, the continuum I Stokes image (33 channels over 128 MHz) is dynamic range limited, as shown in figure 3 with peak 28 Jy, and lowest contours +/- 0.25 mJy or around 1 This was without cleaning boxes specified, and probably could be improved. The Stokes Q, U and V images do

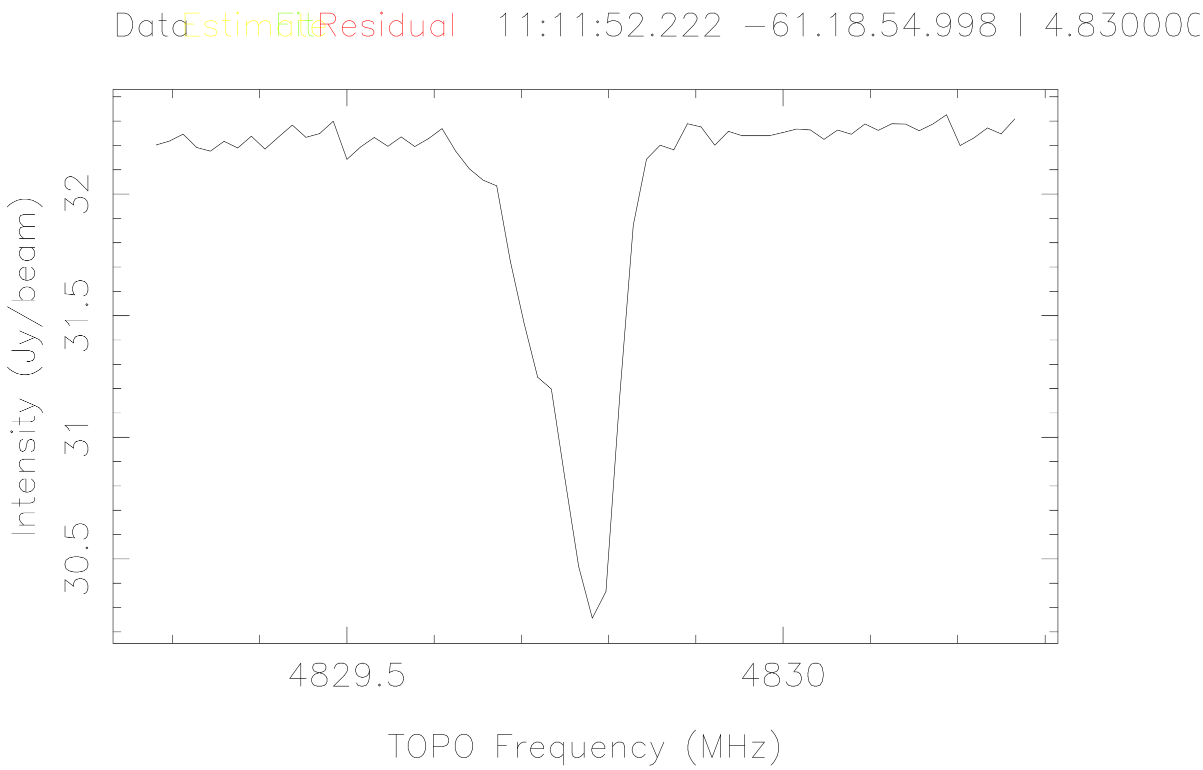


Figure 4: The ATCA spectrum of G291.3-0.7 showing formaldehyde absorption, from the AIPS++ data cube.

not show any artifacts above the noise, which is a good sign that the calibration has worked. However, the same data reduced through Miriad gives an I image with better dynamic range, so it may be that I did not do as well in flagging and calibration in AIPS++ as I did in Miriad. Recall that there were problems with this C1080 data. I also made a data cube with the other IF (513 channels over 8 MHz), selecting the central 64 channels including the formaldehyde absorption. The image planes, which can be displayed as a movie in "view", look very similar to figure 3, confirming that the limit was dynamic range not sensitivity. The spectrum at the pixel near the continuum peak is shown in figure 4, from the "profilefitter" tool. This line image is not continuum subtracted, and this could be done in the image plane, but it would be better to do the continuum subtraction in the visibilities (like Miriad uvlin), which will be in an upcoming AIPS++ tool. The cube frequency axis was just the observed channels, but I have not yet been able to get the ATCA uv-data into the correct velocity, using the rest frequency and LSR correction. The ATCA frequencies are as observed, not doppler corrected for the motion of the Earth and Sun, so the corrections need the right header information of observatory position, time, velocity reference systems etc.

VLBI - LBA

VLBI synthesis imaging should also be able to be done in AIPS++ eventually, but not yet. The Marsfield LBA correlator produces output in RPFITS format, so I did a test of loading LBA data into aips++ with the "atcafiller" tool. This worked quite well (somewhat surprisingly as "atcafiller" was not really designed for this), and the "measurement set" could be plotted and edited with "msplot". However, the VLBI-specific tools for calibration are not yet ready, so a full end-to-end reduction to final images for ATNF LBA data can not yet be made.

See the AIPS++ v2.0 Development Plan and notes of VLBI Processing requirements on the NRAO AIPS++ Users Group web site
(http://www.aoc.nrao.edu/ smyers/aips++/) for the plans for VLBI in AIPS++.

Conclusion

ATNF users may find AIPS++ useful for specific data applications that are not covered, or are not as easy, in other packages. At this stage, it appears that continuum ATCA data can be processed end-to-end in AIPS++, but there may be some bugs or missing functions to do this for spectral line velocity data. Also, it is fair to say, AIPS++ still has bugs that may make it a bit frustrating to use, so we are not (yet) recommending AIPS++ to all users.