AT 39.3/022

also AT 46.3/025
Draft May 19, 1992

AIPS++ Consortium User Specifications
ATNF Priorities

ATNF Staff and Users

Australia Telescope National Facility
CSIRO, PO Box 76,
Epping, NSW 2121
Australia

1 Introduction

This specification is designed to state the requirements for the AIPS++ software system specific to ATNF users and to indicate the relative priorities of these requirements. This document follows 'AIPS++ User Specifications' (AT/46.3/021) and is intended to supplement 'AIPS++ Consortium User Specifications' (draft dated March 6 1992), together with the requirements and priorities from other observatories. General priorities (§2) are placed in two blocks: high priority and lower priority. Specific priorities (§3) are listed with the lower priority items bracketed by [ ].

2 General Priorities

ATNF users require for AIPS++ applications which facilitate the quick, efficient and friendly processing of data, the displaying of that data in a manner that facilitates interpretation and the creation of publication-quality plots and images. The abilities and experience of the Australian user community (in radio astronomy) ranges of from novices to experts, with the novices in the majority at the present time.
Once the structure of the AIPS++ environment has been created, applications that allow for the handling of the following types of data should take priority:

- Compact Array interferometry (1–115 GHz frequency range, 30–6000-m baseline separations)
- Long Baseline Interferometry (1-25 GHz frequency range, > 100 km baselines)

See the ATNF AIPS++ Technical Specifications document (AT 46.3/023) for technical details.
The ability to handle the following types of data should also exist from the outside:

- Space-based VLBI
- Single-dish spectral-line
- Single-dish continuum
- Single-dish pulsar

However, creation of applications by the AIPS++ consortium for these four data types should take second priority until 'core' applications exist for Compact Array and Long Baseline Array interferometry. We imply no ranking in scientific merit in the above. These priorities arises from the fact that much single-dish analysis is already being performed in a satisfactory manner.

3 Specific Priorities

1. We require the ability to read in, to edit, to calibrate, to average, to filter and to image multi-frequency, multi-IF, multi-field data in a simple, quick and efficient manner and to have the ability to easily incorporate new and possibly complex algorithms. We require a full set of tools for the deconvolution and the subsequent analysis of $n-$dimensional images. Ease of usage is the highest priority for the novice.
2. AIPS++ should at least span the present useful functionality of AIPS for interferometer data.
3. We require AIPS++ to replace AIPS totally by 1994. In case of timescales slipping, we would prefer to see a basic AIPS++ with limited capability (i.e. mainstream AIPS++ functionality) released within that time, rather than waiting longer for all functionality to be implemented.
4. We require the ability to access files in real-time. There is no reason not to commence the inspection and even editing and calibration of data whilst observations are in progress. This is a feature of many single-dish data formats and assists greatly with the speed of data reduction and the accurate assessment of data quality in real time.
5. We require the ability to form widefield 2-dimensional images from a series of scans or pointing centers not necessarily on a regular grid.
6. We require the ability to handle multi-frequency, full polarisation, binned pulsar data.
7. We require the ability to accurately apply doppler corrections to AT data in software. These corrections should at least allow the choice of optical or radio conventions and LSR or heliocentric rest-frames (for the optical convention, a regridding of data in velocity is required).
8. AIPS++ can be restricted to run on Posix machines and in a X-Windows environment. However, to ease possible future migration to other operating systems and windowed environments, operating system and window environment dependencies must be kept to a minimum, and confined to a well-defined set of modules/classes (much like the current AIPS Z and Y routines).
9. A failing of many current packages is load-time overheads for reading application executables or images from disk. This can make wall-clock time vastly exceed CPU time when doing relatively simple, but repeated, data manipulation. This may be remedied in a number of ways, and we leave it to the implementers to solve this problem (e.g. with an image stack and an application stack).
10. Interfaces should draw a distinction between required and optional parameters. In all cases, we prefer that such 'adverbs' not be global across applications, unless specifically requested (e.g. turning on calibration for all applications). There should be consistency between adverb names across applications, and it should be possible to transfer (e.g. cut-and-paste) adverbs between applications.
11. [For the benefit of remote users accessing AIPS++ remotely over slow terminal lines, it would be useful to provide a user interface running under DOS-Windows and/or Mac-OS that generates the appropriate line commands.]
12. [It should be recognised that within the lifetime of AIPS++, input to the package may be by voice or pen-pad, and hooks, where known, should be provided as appropriate.]

13. All documentation, including tutorial and cookbook material, should be available on-line. Hypertext may well be an effective way of handling all documentation.
14. Multiple image windows should be allowed, and the distinction between line graphics and image windows should be removed. Multiple overlaying of images within windows should be allowed, using different look-up tables if the hardware allows.
15. It must be possible to present high quality diagrams, with labelling etc. This might best be achieved by providing easy transfer to other packages such as Publisher, Harvard Graphics, or MacDraft.
16. One particular area where AIPS is weak is in the overlaying of images from other observatories. For example, digitised data from all-sky Schmidt surveys is now readily available in the form of positions, identifications, isophotal shapes and magnitudes. AIPScan only plot uniformly sized crosses in overlay planes. Handling of overlay planes themselves needs improving over the current implementation on Sun workstations.
17. Research is currently being done at the AT and elsewhere on using advanced visualisation engines to provide rapid interactive visualisation to data cubes. Such cubes are composed of voxels distinguished not only by colour and brightness but also by opacity, and ray tracing then gives efficient visual cues to the user. These cues are enhanced by techniques such as interactively manipulating the cube, and displaying it stereoscopically. While such facilities are at present available only on specialised hardware, the increasing power of workstations means that such facilities should be accessible to all users within a few years, and AIPSmust be capable of supporting such concepts.
18. It should be possible to store closure phase explicitly, both for VLBI and optical aperture synthesis.
19. It should be possible to store complex images explicitly, rather than as 'REAL' and 'IMAG' images as in present AIPS.
20. Applications which generate listings of data should consider the needs of users who want to export their data (as ASCII tables) to spreadsheet and database applications. For example, headings and comments in generated lists should be prefixed by a # character, to allow these to be easily stripped out, or treated as comments. Tasks should be provided explicitly to allow transfer of data and tables to common packages such as Excel, 1-2-3, and dBase.
21. [Consideration should be given to the implementation of 'real-time' mapping which will be possible, even for large data sets, with future generations of computers. Ideally, a user would like to see u-v data displayed in one window, and the resultant image in another window, with a FFT-MAP-CLEAN-SELFCAL pipeline running continuously between them. Thus the user can edit his/her data in one window and see the resultant image a few seconds later in the other. Alternatively, a calibration icon may be double-clicked, a parameter tweaked, and the effect seen in real time.]
22. Data must be uncorruptable by errors! When error handling is implemented, it should give a true indication of the stage at which the error occurred.