

AIPS++ Consortium User Specifications

AIPS++ Consortium Development Group

1 Introduction

AIPS++ is an acronym for the Astronomical Information Processing System that is being designed and implemented by a consortium of seven radio astronomy institutions: the Australia Telescope National Facility (ATNF), the Herzberg Institute for Astrophysics (HIA) through the Dominion Radio Astrophysical Observatory (DRAO), the National Radio Astronomy Observatory (NRAO), the Netherlands Foundation for Research in Astronomy (NFRA), the Nuffield Radio Astronomy Laboratory (NRAL), the Tata Institute of Fundamental Research (TIFR) through the National Centre for Radio Astrophysics (NCRA) with GMRT headquarters at Pune, and the Berkeley-Illinois-Maryland Array (BIMA). AIPS++ is intended to replace the AIPS (Astronomical Image Processing System) with a more modern, more extensive, and more extensible software system.

This document is mainly based upon the User Specification documents prepared by each member of the consortium, with some use of other written contributions to the User Specifications Memo series. “Distillation documents”, written by the consortium members participating in the initial six months design phase in Charlottesville, have been extensively used in the preparation of this document.

2 General Characteristics of AIPS++

2.1 Guiding Principles

These specifications describe the capabilities needed in AIPS++ by astronomers who use telescopes operated by members of the consortium. We attempt to avoid expressing opinions on how such capabilities should be implemented. However, because AIPS++ should be optimized for the astronomer user, we do specify some aspects of the user interface that we consider essential.

AIPS++ must anticipate a wide range of experience within its user community. Both the user interface and the off-line documentation must address the disparate needs of novice (or occasional) users and of experienced users who may be analyzing technically demanding observations. To match the needs of users with a wide range of experience, a hierarchy of interfaces and documentation will be essential. Users will also need a hierarchy of programmability. At the lowest level of experience, this should allow them to connect major (and sometimes repetitive) steps in data processing conveniently. At the highest level, an efficient interface is needed to encourage development of new, experimental, algorithms and processing techniques.

The following principles are important in the design and implementation of AIPS++:

- **Accountability** – Data should have associated telescope performance (“monitor”) and processing histories so their origins and evolution can be easily reviewed and understood by astronomers. Unnecessary structures should not be imposed on data, and data should be accessible in both “raw” and any modified forms. However, it should be possible to have very flexible selection of data sub-sets.

- Astronomical terminology and concepts – Names and labels in the data processing system should use the common language of astronomy and mathematics.
- Programmability – It should be easy for users to prepare data processing “scripts” for repetitive or multi-stage processing, and to augment the system easily with new operations and algorithms.
- Easy Customization – The user should be able to flexibly select data processing packages to be used, the style of user interface, and environmental parameters such as directory names and output devices.
- Hiding complexity – Where possible the complexity of algorithms or multi-step processing should be hidden from the novice user.
- Confidence in results – Data processing diagnostics and the capability to un-do and re-do steps in the processing are essential for telescope data so the user can understand and have confidence in the data at any processing stage.
- Range of processing styles – the same software should be usable for both post-observing processing and remote or local on-line data analysis
- Future types of processing – the design should allow for future growth in network computing with use of remote displays, remote batch processing, and parallel processing of different machines or in machines with parallel processing architecture.

2.2 Scientific Goals

AIPS has been an acronym for “Astronomical Image Processing System”; however, its capabilities, and users’ requirements, have evolved far beyond image plane processing. AIPS++ should now be a general tool for turning telescope data, and model calculations, into scientific results. In some cases, e.g., graphics and tables, the results should be in publishable form. Most n-dimensional images are produced only as an intermediate step between raw data and useful results, however some constitute final scientific results and require reproduction in publishable form. A similar range of purposes has evolved for single dish data in systems such as UNIPOPS. The concept for AIPS++ should be that of an Astronomical Information Processing System.

In specifying a new software system, it is useful to consider what aspects of astronomical data processing have remained stable over the last 15 or so years. The most stable parts of array and single-dish processing systems are the *fundamental* descriptions of telescope data. For the major arrays, the basic description has accumulated more attributes (e.g. spectral channels, IFs) but it is still fundamentally a visibility data set – samples of a spatial coherence function in some convenient spatial or temporal order. Similarly, a final image is still an array of calibrated pixel intensities in a known coordinate system, polarization, and observing frequency. The “end results” are scientifically meaningful quantities extracted from one or more such images, and published visual representations of these images. Key, and probably stable, basic ingredients of a user specification are therefore the types of *data* to be handled (e.g. visibility data, single dish spectra, images, image cubes).

Basic operations on parts of data sets, such as Fourier transforms, least squares fitting algorithms, plotting, display, mathematical, and other standard functions are also relatively stable. We will call these basic operations *tools*. The second ingredient needed by users is an itemized tool kit of basic operations from which more complex astronomical applications can be assembled.

In contrast, the algorithms used to calibrate, construct and interpret data sets and images evolve as the astronomical community acquires experience and sophistication in data and image analysis techniques. The algorithms are the least stable elements of present software. They continually evolve or are replaced (either as explicit programs or as informal procedures that may involve astronomer interaction). The algorithms are embodied in *tasks* which can be implemented either as specific programs in a language such as C++ or as scripts in a higher level language. Many of the tasks that are now part of the lexicon of astronomical image processing will be embodied in AIPS++ at an early stage. The tools in the kit provided by AIPS++ must, however, be easily usable by astronomers to carry out new tasks whose nature and scope may evolve rapidly with time.

In these terms, the core of AIPS++ must provide a generic toolbox operating on specific data types. Given the finite resources available, the limitations of AIPS++ should be more in the diversity of data that can be handled rather than in what can be done to these data.

2.3 General Attributes of AIPS++

AIPS++ should have good command line interface with “full” programming capability. This should be at the level to eliminate, for most astronomers, the need to write FORTRAN or C++ programs. We view the issue of who will be able to develop applications programs as one of the most important issues for the future. “Full programming” capabilities with the AIPS++ “command language” is very important; however, the use of C++ and FORTRAN “template” programs that can be run “with” AIPS++ is also important. In addition, the current plan to have many astronomers doing C++/OOP programming for AIPS++ will require special attention to astronomer-oriented documentation, programming guides, and possibly things like programming “summer schools”. Assuming that everyone can learn what they need from industry-wide material aimed at professional programmers is unwise, and is likely to limit the AIPS++ pool of developers to too small a group with too little astronomical experience.

Documentation for AIPS++ should be available both on-line and in hard copy. This should have multiple levels ranging from simple “help” to extensive information, and dealing with both specific applications and individual parameters.. Consistency between hard copy and on-line documentation is imperative. Multi-window environments, as mentioned above, should allow context-sensitive information to be displayed by “clicking” on appropriate items. While the implementation aspects of a UNIX “man” page might be useful, the displayed information should be easily understandable to user-astronomers.

Multiple levels of user interface would be desirable to allow for both novice users and experienced expert. User selection of the style of interaction and the range of “packages” to be used should be possible. Choice of the user interface should have no effect on the code used in processing.

Styles of user interface are difficult to decide upon, and are very dependent upon user experience and preference. The discussion in Wood (1991) is an example of a useful approach to the user interface that goes into details we have not discussed here. We recommend planning a number of available styles, and extensive user testing of each of them during early phases of AIPS++ development, as opposed to deciding upon one approach and precluding all others. The idea that the user interface is just another applications task, that can take many forms, is probably very important in planning for the future with a wide range of user needs and expertise.

A combination of the inclusion of single dish data reduction as part of the domain of AIPS++, and the increased use of “nearly real-time” data processing and remote observing

for both single dish observing and arrays, makes the use of AIPS++ as an integral part of the observing process very important. This should not change or add to the processing and display needs of AIPS++, but rather adds to the richness of the tools that can be used to support the users' involvement in the observing process. In addition the post-processing tools need by instrumental staff to maintain their instruments have great commonalty with the things a knowledgeable astronomer would like to see and do during the observing process. The observer would like:

- capability to see instrumental status data both at the telescope and on remote display devices connected to the telescope by networks;
- automatic first order data editing and calibration where possible;
- as much immediate data processing and display as feasible in real-time;
- to be able to make changes in the observing program during observing “runs”; and
- to be able record data processed in “nearly real-time” on transportable media for further processing

In addition to the use of normal AIPS++ processing tasks, this list of needs makes it necessary for preparation and changing of observing programs to be immediately possible. Indeed, the preparation of observing programs may become one of the extended tasks of AIPS++ for some instruments.

The simulation of data produced by real instruments, based upon assumed models of sources, is an additional capability that is essential for AIPS++. This should be viewed as a necessary part of the testing of AIPS++ applications software (both for de-bugging and evaluating efficiency of processing), and as a tool for the astronomer that provides both more realistic preparation for observing and the necessary tools to compare models and data in AIPS++.

3 Data

3.1 The General Nature of the Data for AIPS++

The data will come principally from radio telescopes although AIPS++ must allow import of images and data from other wavelengths. The primary data types that are needed to support the AT, BIMA, EVN, GMRT, MERLIN, VLA, VLBA, WSRT, the future mmA, and the various instrument packages on the JCMT, GBT, the 12m and the 43m are as follows:

1. Telescope status information
2. Total power and phased array data sequences reflecting switched or time series observations
3. Spectra
4. Images
 - (a) Planar images at radio, optical, X-ray, etc. wavelengths
 - (b) Spectral cubes - images in multi-spectral regions
 - (c) Time cubes - time-ordered images of variable sources

5. Coherence function (visibility) data from correlation arrays
 - (a) arrays with real-time delay and phase variation correction
 - (b) tape recording arrays where the correlator output is coherence function data for a range of time lags (or transformed frequencies)
6. Calibration tables
7. Data editing information
8. Computed models
9. Processing histories

Some of these data categories are naturally associated with each other; it is also important to be able to group some together when appropriate, e.g. in mosaicing observations. Some of these data types are either super-sets or sub-sets of others; it is important to be able to compose super-sets out of sub-sets and to decompose super-sets into their sub-sets.

It is important that the astronomer have access at all stages of data processing to the conditions under which an observation was made, and to what has been done to it in the data processing. The ability to wipe the slate clean has proven its utility over and over again in many data processing systems. Hence the database should carry both telescope-provided status information and a processing history in formats that make it easy to “start over” if processing goes awry. This supplementary information begins with data structures with telescope information as a function of time, position, or other data identifiers such as telescope name, latitude, etc. It continues with data processing history sufficient to understand, un-do, and re-do that processing.

Another view of the data relates to different uses and time scales of use. These uses lead to three major categories of software: on-line data analysis; system support software for staff operating and diagnosing the operation of the instrument; and observers analysis software. For single telescopes the observer has often done a major fraction of data analysis at the telescope as part of observing process. Recent hardware and networking developments have made such on-line data analysis feasible even for high data rate instruments like the large arrays and single dishes with fast sampling spectral processors. Most telescopes have, or soon will have, full remote and local analysis capability. In all these system data should be accessible to the user as soon as practicable in nearly real-time. For all these reasons data analysis software in the AIPS++ system should provide for the needs of the above mentioned three categories of software.

Different single dishes and arrays have different approaches to data handling in which similar words mean different things. These “cultural” difference in language must be directly addressed and data descriptions and terminologies that are consistent at all telescope and development sites must be created and maintained.

3.2 User-Oriented Data Organization

From the point of view of the user the highest level identification of the problem is what we will call a “project”. Projects are aimed at obtaining answers to scientific questions. Answers to these scientific questions frequently involve obtaining data from a variety of telescopes. Some projects require radio data from both single dish and array observations from the same or different instruments, each serving a different “purpose”. Observations for each instrument are

organized into observing “runs” with sequences of “scans” with identical instrumental and observing parameters. Each scan contains “sub-scans” with data elements in the form of spectra, time instances of coherence function data or spectra, etc., that are associated with instances of time. Astronomers need to deal with this hierarchy of data: project, purpose, instrument, observing run, scans, and sub-scans. It would be very helpful if the astronomer could be aided in dealing with things according to this hierarchy. Data that are viewed as simple sequences of data from stand-alone telescopes leave the astronomer to impose a mental image of project/instrument/purposes and then runs/scans on the simple data elements. The future mmA will be a case where the same instrument will generate both single dish and coherence function data sets. This makes it a prime example where the same instrument will serve diverse instrumental purposes for a wide variety of “projects”.

In this document we list preparation for observing as an AIPS++ task. This is partly because simulation, using AIPS++ processing tools, can be very useful in understanding an observing program during the planning and preparation process. In addition, it is at this stage that the user imposes the logic of project/instrument/purposes/runs/scans on the observing process, and this logic must be remembered and used as part of the data reduction and processing. If tools were available in AIPS++ to aid the user in passing on and using this logic all the way through data processing, it would be very helpful. It would be analogous to having and updating the map of a maze that can be used while passing through the maze. Data processing is very much like a maze to be negotiated for most astronomers, and assistance in dealing with the higher level purposes of data would be very useful.

The above can be describe more technically by saying that data sets should have a hierarchy of descriptor (or “header”) items, with descriptor items being identified by context information (such as name, position, etc., for images). These data descriptors should allow specification ranging very large, merged data sets to basic elements like pixels or u-v data points. It should be possible to eliminate redundancy by describing information on a sufficiently high level while allowing exceptions by overriding this information at a lower level; that is, mixtures of positive and negative data/information specifications.

4 Specific Requirements

This section contains long lists of brief description of the elements of the user requirements from all the consortium specifications. It is based on a merger of the distillation of specifications by AIPS++ working groups in the areas of “User Interfaces”, “UV Data System and Processing Requirements”, and “Image Handling” – and the original material from individual consortium user requirements.

4.1 User Interfaces

4.1.1 General

- The user should be able to choose one of a variety of user interface styles. The most important are:
 - an AIPS++ command line interface (CLI) which is a programmable interpreter;
 - a basic graphical user interface (GUI) for X-windows workstations.
 - A FORTRAN program interface to AIPS++ tasks at the level of the host operating system (used by FORTRAN programs who prefer this method of adding software

capabilities);

- Useful, but lower priority interfaces, are:
 - Execution of AIPS++ tasks at the level of the host operating system (used mainly by programmers and expert astronomers)
 - A data flow graphical processing environment
 - Additional interpreter or GUI interfaces
- All interfaces should use the same language for “commands” and parameters, and where possible have the same look-and-feel
- Users should be able to execute operating system command sequences from any interface
- Application “task” parameters should have the following properties:
 - All parameters should have assigned defaults, possibly dependent on context
 - Selected parameters should be user-assigned according to the mode of user interface
 - Parameters should NOT be global across applications, unless specifically requested or used to specify processing environment and scope
 - Names must be consistent across applications, using astronomical terminology where possible
 - Parameters should be able to be passed between applications when applications share the same parameters
 - Checking of parameters by applications before execution starts should be done whenever possible (the user will be warned about inconsistent, unusual, or dangerous combinations, and, for the latter, may refuse executions with re-entry dependent on the type of user interface)
 - It should be possible to save, edit, and restore, parameters sets for applications
 - For some applications it would be useful if a task in execution could have some parameters changed by user request
- A processing history or log should be maintained for documentation and re-execution after editing
- Users should be able to choose any editor supported by the operating system for editing ASCII data, parameter lists, processing histories, etc.
- A variety of help levels, preferably context sensitive, appropriate to novices, experts, etc.
- There must be a batch capability for all user interfaces with capabilities to monitor, interrupt, and modify batch operations
- There must be error handling at all levels in as user-understandable form as feasible
- It would be useful to have multi-tasking operation for CLI, batch, and GUI interfaces
- It would be very desirable if the system could warn of high resource usage (CPU, disk, tape) before applications are executed (advise and request confirmations)
- Planned developments in the use of parallel processing make it very important that one be able to develop and optimize algorithms based on parallel architectures, particular for large mosaicing and spectral line applications

4.1.2 AIPS++ Command Line Interface

- Must be usable from ASCII terminal and remote network nodes; this particularly includes PC's or terminals with alternation between ASCII modes and Tektronix emulation modes
- Must be programmable in the sense of supporting variable assignment, conditional statements, control loops, string manipulation, functions, and procedures
- Should have capabilities to build, read in, and write out "procedures"
- Should have a beginner mode, with prompts in plain English, and advanced mode
- Must have a "batch" facility for execution of series of CLI commands
- Should have command-line recall and editing facilities
- Should allow uses to define their own commands and procedural sequences of command lines
- Must have access and control of AIPS++ objects and data structures
- Should have user selectable input and output data streams, and display devices
- Must support normal arithmetic operations for intrinsic, user-defined, and image/spectral data types
- As much as possible, applications written for the AIPS++ interpreter should look the same (to users) as those written in the compiled language
- A very useful feature would be "un-do" operations wherever this is feasible

4.1.3 Graphical User Interface

The basic graphical interface should be the primary user interface for users in 1994. It should be the most attractive one for most AIPS++ users, and maybe even for experts. It should be a window-oriented graphical interface with pull-down menus (for application selection and parameter specification), multiple windows, and pop-up menus for context sensitive help. Menus for application selection and parameter specification should have pop-up sub-menus with options/parameters depending on menu context

It would be desirable if there were an advance GUI with visual programming of applications. Icons/glyphs for individual "tool" components and connecting lines for passing of data. Sequence of graphical task and data flow connections should be capable of being saved, edited, and retrieved.

4.1.4 User Documentation

Documentation for AIPS++ must be a planned part of the AIPS++ development. It should be

- Uniform in style, which may require a central documentation editor or a single, capable technical writer;
- Prepared by, or in real consultation with, experts on the material;

- Use astronomical and mathematical terminology wherever possible;
- Not be the same as programming documentation except for the case of AIPS++ programming guides or cook-books aimed at astronomers;
- Be completely available on line, based on the same textual material used in printed documentation;
- Be used in the connection with “help” with context sensitivity and capability to search on keywords or names (in or out of context);
- Have complete documentation, with references, of algorithms used and their effects on data
- Have several levels including
 - User cookbooks;
 - Application descriptions;
 - On-line help.

4.2 Data Handling

4.2.1 General Considerations

As discussed earlier, a large fraction of the data processing in AIPS++ can be described as “data handling”. These data can be images, single dish data sets, coherence function data sets, telescope performance data, model data, or any data set that can be imported into the system.

4.2.2 Data Import and Export

- Data from consortium instruments are to be imported into AIPS++ with software tailored for each telescope, producing data file(s) with data that can be read, modified, and written using AIPS++ data I/O routines.
- AIPS++ data files must be written to, and read from, tape and other transportable media using AIPS++ data base I/O software.
- FITS and UVFITS data I/O to both disk and transportable media must be supported.
- Data import and export for general cases should be supported in the form of reading and writing tabular formats in ASCII or binary form. In the case of ASCII tables, support of “columns” in both numerical and “string” form should be supported.
- Transfer of data over networks should be supported.
- Data files in AIPS++ should be identifiable, and accessible, using host operating system or user-written software.
- Data book-keeping software, both inside and outside AIPS++, should be available to help users organize and otherwise deal with their data.
- Searching, summarizing, copying, and concatenating contents of AIPS++, FITS, and UVFITS data files, with selection based upon data parameters, must be possible for both disk files and transportable media.

- Support of large data sets on multiple “tapes” must be supported for AIPS++, FITS, and UVFITS data files.
- The data system file format(s) should be accessible from all supported machines without conversion
- Distinctions such as “multi-source” and “single-source” data sets should be avoided
- Applications should function on n-dimensional data in arbitrary sort order with it being possible to hide sorting from the user
- It should be possible to extract data subsets from a data set, manipulate it with the AIPS++ (IDL-like) command language, treat these data like any other data for computation and display, and optionally put it back in the original data set
- Flexible transport of data sets or subsets to and from AIPS++ and other major data reduction packages should be supported
- Data coordinate handling must very general
 - Support for non-regular increments in coordinate axes
 - Flexible and reversible coordinate transformations must be supported
 - Coordinate and ephemeris information at the level needed for astrometric and near field imaging should be supported
- Support for errors in data must be fundamental to the data system, providing a basis for support of data error handling in a variety of applications
 - Error images associated with astronomical images
 - Easy generation of error models
 - Error propagation through a series of data processing steps
 - Properly formatted errors when data are extracted for tabulation
- The user should have access to both data values and all the associated (‘header’) information that govern the interpretation of the data
- Processing histories should be maintained for all data sets, with easy review and re-use by the user
- It should be possible to import general astronomical data (e.g. catalogues) into AIPS++
- For applications where the built-in tasks and command language features are insufficient, there needs to be a program interface (outside AIPS++) to allow the casual programmer reasonable access to the data; some flexibility and efficiency can be sacrificed in making this interface comparatively simple; FORTRAN programmers should be supported

4.3 Nature of Instrumental Data

4.3.1 General

- All data from single dishes or interferometers should be assumed to involve full measurement of the electromagnetic field involving all four Stokes parameters and the equivalent complex polarization representations, with support for, and transformation between, both forms
- Multiple frequency bands may be simultaneously observed (e.g., for observing multiple lines simultaneously or multi-frequency synthesis), with variable numbers of channels in each band
- Frequency axes may be non-linear (e.g. as produced by acousto-optical spectrometers) and time variable
- Polarization measurements may be time switched if all polarization measurements are not obtained simultaneously
- Data combinations for different observations may have different numbers of spectral channels and channel widths which may need to be accommodated within single data sets
- As much as possible the data handling system should support generalized imaging handling rather than having one system for images and another for all other data, allowing scope for “vector images”, complex images, images with associated errors, double precision images, etc.
- The user should be able to use the host operating systems capabilities and utilities to manage data sets, using normal file names and directory hierarchies
- Capability to transform tabular data file formats to AIPS++ instrumental data formats will allow general importation of non-standard types of instrumental data
- Instrumental performance and meteorological data need to be associated with instrumental data sets either directly or with associated data “tables”
- The data system must deal with telescope dependent instrumental data
- Data for focal plane arrays, or multi-beam feeds, with arbitrary geometry (e.g., field rotation during the observations) characteristics must be supported
- Mosaicing observations may have many (~ 1000) pointing centers which must be supported for single dish data, interferometer data, imaging, image processing, and image display
- Must support rapid time switching of polarizations, frequencies, and pointing centers (i.e., they may change for every integration)
- Time-series data for total power measurements and visibilities must be supported (e.g. pulsar data, time variable sources)
- Error measures or estimates (e.g., weights) should be regarded as standard in an observation

- The data system must allow for simultaneous processing of “associated” data sets, such as different (e.g., by calibration, integration time, fringe fitting, etc.) versions of the “same” observation, so the best can be selected later
- Correlation data in the form of 16-bit integers or 32-bit floating point must be supported
- It is desirable that the data system be as extensible as possible, including new data types
 - Triple correlation data, including cases where visibilities have different frequencies
 - Optical interferometer data
- It is desirable to be able to import, search, and select data, including spectra, images, etc., from instrumental data catalogues and archives

4.3.2 Single Dish and Summed/Phased Array Data

- There must be support for the major types of single dish data in the system
 - 1-D spectra, both evenly and non-evenly spaced in frequency (e.g., taken with AOS spectrometers), where an associated 1-D array identifies spectral frequencies
 - 1-D sequence of total power (continuum) measurements, possibly unevenly spaced and associated with a 1-D sequence of pointing positions or time
 - 1-D sequences of data values taken at arbitrary positions, times, foci, etc. and used for tipping, continuum on-off, focusing, and pointing observations (the previous two items can be viewed as subsets of this more general organization scheme for data)
 - 2-D matrices of data values as a function of (x-position, y-position), (position, frequency), (frequency, time), (position, time), (time, pulsar phase), (pulsar phase, frequency), etc., where both axes may be non-linear or non-parameterizable
 - 3-D “cube” of data values as a function of (x-position, y-position, frequency), (x-position, y-position, radial velocity), (time, frequency, pulsar phase), (x-position, y-position, time)
- Total power auto-correlation data must be supported
- There should be support of bit-field data for pulsar observations
- There must be data handling for fast sampling spectrometers with from 128 to 32768 channels of data, producing very large data “cubes”, both for spectroscopy and observing where interference excision is important
- There must be support for data from fast sampling surveys (basket weaving, mosaic sampling)

4.3.3 Interferometer Data

Interferometer data should be regarded having potentially inhomogeneous antenna properties, but this should not preclude dealing with simplified cases where homogeneity can be assumed.

- Antenna size, system temperatures, and frequency band-passes may differ widely
- The input data and calibration procedures may vary from antenna to antenna
- Integration time may vary from antenna pair to antenna pair

4.3.4 VLBI Data

- Visibility data handling must support for many correlator formats, including MkII, S-2, K-4, MkIII, and VLBA modes
- The data system should support the merging of correlation data and associated calibration data from different correlators and allow the user to deal with duplicate correlations
- VLB antennas in space will require support for orbital position dependence including acceleration terms

4.3.5 Mosaicing Data

- Both single dish and aperture synthesis data may need to be merged in the mosaic imaging process
- Self-calibration with cross-referencing to data for overlapping areas should be supported
- Data handling for multiple pointing centers, including effects of beam shapes and pointing errors for each center, should be handled in a convenient manner

4.4 Data Correction and Calibration

4.4.1 General

- Data should be selectable in terms of identification with a particular type of calibration observation
- Both standard and user-defined models of data behavior should be usable in determining calibration information from data sets
- Instrumental behavior that affects calibration should be integrable in the calibration process through a mixture of parameterized functions and models in tabular form
- Data correction based upon standard and user-defined functions, with user supplied parameters, should be possible
- Calibration and correction of data should be reversible, with the capability to BOTH store calibration/correction information and apply it “on-the-fly” during processing, and apply this calibration/correction information “once and for all”, creating new, calibrated data sets
- Calibration should be made as generic as possible, with telescope-specific methods kept to a minimum
- Calibration/correction of data should be possible from derived tables of instrumental parameters (e.g., system temperature vs. time, gain vs. elevations), with derivation of such tables from calibration observations
- The calibration process should include flexible averaging of calibration data and application with interpolations or weighted averaging, all under control of the user
- Cross-calibration from different instruments should be possible (e.g. flux scale, pointing) particular when data from different arrays are to be combined

- Model fitting should be possible in both the image and u-v planes, and it should be possible to use the resultant models for further calibration and self-calibration
- There must be simulation programs for single dish, interferometer, and mosaicing data bases for both planning and comparison of data with models - with optional error generation for thermal noise, pointing errors, primary beam errors, atmosphere, antennas surface errors, beam-switching for total power, etc.

4.4.2 Single Dish and Summed/Phased Array Data

- Flexible spectral fitting for components and baselines; interpolation/blanking of bad channels
- De-dispersing of spectral, long time series data for pulsars with analysis and fitting in the intensity-frequency-time domain
- Telescope pointing and beam pattern determination and correction
- Analysis of telescope performance data: pointing, telescope-tipping, focusing, and holographic data
- Deconvolution of ‘channel’ shapes and ‘frequency-switched’ data
- Analysis of telescope instrumental data in “nearly” real time
- Phased display of selected time sequence data
- Special intensity and polarization calibration of phased-array data

4.4.3 Interferometer Data

- Antenna-based determination of calibration and self-calibration functions should be the primary form of calibration determination wherever possible
- There must be capability to make phase and/or amplitude corrections of data based upon difference between the data and modeled data sets, where the latter are usually derived from imaging of the same or highly related data
- Redundancy in data (possibly including crossing points) should be used whenever possible as an additional constraint on calibration and self-calibration
- Determination of, and application of corrections for, closure errors should be possible with flexible averaging of input closure information
- Fringe fitting for a range of spectral channels and fringe rates (normally only for VLBI data) should be possible by baseline, as well as globally by antenna
- Spectra calculation from complex summing to visibilities in each spectral channel for user-specified positions in the field of view
- Interferometric pointing, baseline, and beam pattern fitting and related analysis
- Application and de-application of astrometric/geodetic correction factors with complete and reversible histories

- Calibration of data for effects of the ionosphere, utilizing data at multiple frequencies and/or external data on variations of electron content
- Calibration for non-isoplanicity using special extensions of self-calibration

4.4.4 Mosaic Total Power and Interferometer Data

- Calibration of mosaic data bases with necessary cross-referencing of multi-pointing information; this makes it necessary for individual data points to be associated with pointing center information, which is a special problem for interferometric data
- Calibration parameterization must include specification of pointing centers to be used, because of the need for “all-pointings-at-once” operations, in addition, it must be possible to specify a subset of all pointing for a given operation
- Determination and correction for pointing errors, and errors in beam shape, using mosaic self-calibration techniques, will be important

4.4.5 Additional Considerations for VLBI data

- Special limitations on calibration for VLBI data
 - Because of independent atmospheres, clocks, and LO’s at each antenna, there are uncertainties in phase referencing for fringes, and variations in coherence time
 - Because almost all geometrical effects scale with baseline length, one needs the most accurate geometry for earth- and space-based interferometry
 - The correlated visibility functions are essentially phaseless, so self-calibration of phase is the only possibility
 - Antenna characteristics can be very different across the array, hence one must be careful about antenna-based simplifying assumptions
- For amplitude calibration one must use the T_{sys} and $^{\circ}\text{K}/\text{Jy}$ determined for each antenna, and determination of these quantities are an essential part of the calibration process
- For spectral line sources one can do amplitude calibration with auto-correlation spectra plus calibration at one antenna
- Accurate Doppler correction for each spectral channel is essential
- Fringe-fitting with sets of data with as many baselines as possible is essential, with the limitations that sources must be detected in a few minutes and only bright sources can be observed unless one does phase-referencing
- For polarization calibration, all calibration sources are resolved and the polarized intensity distribution may not be like the total intensity distribution, therefore one must iteratively determine both source polarization structure and instrumental polarization
- Polarization calibration must use an ellipticity-orientation model for feed polarization, which is non-linear and computationally “expensive”, because one must calibrate mixed linear and circular polarization characteristics (requires very careful amplitude and phase calibration)

- Calibration and self-calibration on the same, with both depending on deconvolved source models, and this process can take tens to more than a hundred iterations
- Calibration has baseline-dependent factors because of mismatched frequency passbands in non-identical telescopes
- Full phase calibration is an iterative process involving limits set by: astrometry, geodesy, and weak source imaging/detection, therefore one needs:
 - very accurate geometric models, typically to at least 1/10 of a wavelength accuracy
 - knowledge of location of the Earth’s pole and UT1, both of which are generally known only after astrometric/geodetic analysis
 - values of ionospheric delay as determined from measurements at simultaneous frequencies, or external measurements of ionospheric electron content
 - measurement of properties of troposphere dry terms (from surface meteorological measurements) and wet terms (Kalman filtering, GPS multi-frequency satellite measurements, WVR)
 - instrumental delays as determined from phase calibration signals
 - knowledge of non-rigidity of the earth due to earth tides and atmospheric loading
- Very accurate coordinate systems are required; geodesy uses a system based on solar system barycenter
- Full history of telescope behavior/environment and assumed correlator model must be part of data subjected to global fringe-fitting

4.4.6 Data Editing

- Data display and editing should be seen as generic tools applicable to single dish, interferometer, and other forms of data
- Data visualization for evaluation and editing purposes should be seen as an integral, or closely coupled, aspect of the data system
- It should be possible to do interactive editing based upon display, with “zoom” or magnification, and menu selection of editing options
- Various “viewing strategies” should be available
 - For interferometer data, baseline by baseline display (with magnification of local areas) and interactive editing (including multiple, simultaneous baselines) using both Intensity-time-baseline displays and Intensity displays in u-v plane
 - Displays of spectra and spectral cubes aggregated in various ways (spectra vs. time, averaging in time, averaging of channels)
 - Selection of data by specifying windows in space and/or time
 - Selection of arbitrary cuts through data (e.g. circular, radial, or a user-defined locus) through selected data coordinates
 - Display of expanded data aggregates (e.g., pointing and clicking on an average multi-channel region of data to show the component spectrum)

- Comparison displays of generic model data (from fitted components) with observed and/or processed data, including display of data with model subtracted or divided
- Data editing should be reversible, with the capability to store, apply, and un-do editing information
- Data editing should be possible on the basis of monitor/observing log data
- Editing should be possible from “consistency check” information, particularly where there is redundancy or (for interferometric data) where there are crossing points in the u-v plane
- It is desirable to have parameter-driven, automated flagging for large data sets
- Editing must be possible based upon difference between data and models generated during self-calibration
- Data editing based upon recognition of interference patterns in intensity-time-frequency data is very important, particularly for low frequency observations

4.5 Imaging and Image Processing

In this section we consider the formation of images from edited, calibrated data. While this is mainly image computation and deconvolution, it must be remembered, that for the user, imaging and image deconvolution is an integral part of the process of data inspection/editing, calibration, imaging, self-calibration, data/image display, spectrum/time/image analysis, and production of hard copy for publication purpose. This process must be well integrated for the convenience of the user. It should be possible to easily “mix-and-match” self-calibration, data transformation, and de-convolution “tools”, for example, using CLEAN to deconvolve in the early stages, and maximum entropy later on when CLEAN begins to be less useful. This is related to the need to make self-calibration use a generic model, which could be a table of CLEAN-components, a table of Gaussian components, or an image.

4.5.1 Image and Spectral Image Formation

- Image construction from calibrated total power data (beam-switched, multi-beam, focal plane array) data sequences from single antennas and phased arrays, with and without spectrometers, is required
 - Spectral line cube formation
- Image construction using u-v data sets must be possible with a range of capabilities
 - Computation of “dirty” images and point spread functions by 2-D FFT of selected, sorted, and gridded data with user control of data selection, gridding algorithm and its parameters, and image parameters (image size, cell sizes, polarization)
 - Flexible computation of data cubes where the third axis is frequency/velocity or time
 - Simultaneous, multiple field imaging with un-gridded data subtraction using MX-like algorithms
 - Direct Fourier transform imaging of arbitrary (and usually small) size fields
 - Imaging after subtraction for sources

- Imaging of spectral line data sets with continuum subtraction based upon continuum data, or continuum models
 - Estimation and input of zero-spacing flux density and appropriate weighting
- Mosaic image construction using mixture of u-v data sets and single dish data for multiple antenna pointing centers
 - Linear combination of pre-deconvolved images, weighting determined by primary beam
 - Linear mosaic algorithm with linear deconvolution (MOSLIN in SDE)
 - Non-linear (MEM-based) mosaic algorithm (VTESS, UTESS in AIPS, MOSAIC in SDE)
 - Cross-calibration (enforced consistency) between data taken with different instruments (flux scale, pointing)
 - Pointing self-calibration to determine corrections for both single dish and visibility data
 - 3-D mosaicing allowing for sky curvature (non coplanar baselines)
 - Self-calibration and editing of all pointings in one processing step
 - Capability to determine the primary beam(s) from a mosaic image and its related data sets
 - Ability to deal with any primary beams in different forms (analytic 1- and 2-D, tabular), including user modification of primary beam models
- Imaging using multiple-frequency data sets and a user-defined model for spectral combination “rules” must be possible
- Imaging computation should generally take multiple data sets where this makes sense
- Imaging data selection should flexibly allow use of data sub-sets, with data selection based upon time, antenna, frequency, and ranges of other data (including monitor data)
- 3-D imaging of data affection by sky curvature (wide-field problem) is essential
- Imaging wide fields large than the isoplanatic region is essential
- Near field imaging of nearby objects like comets and asteroids must be possible
- Special VLB imaging requirements:
 - Need more accurate handling of precession, nutation, and aberration in the u,v w used for imaging
 - Larger gaps in u-v plan data produces “dirtier” beams and greater need for image modeling/deconvolution
 - Fields of view not radially smeared due to finite bandwidths are relatively small, so one needs “fringe-rate” imaging, and multi-pointing processing for widely spaced sources in the field
 - Near field problems for solar system objects are more important because of the larger baselines
 - Need to do Lorentz transformation to inertial reference frame because of the relativistic distortion due to the Earth’s orbital motion

4.5.2 Image Transformations

In this section we list some of the image-specific transformations of data that are very general operations. Many image transformations are basically transformations of images as arrays of numbers, so we include these operations in the upcoming sections on data “structure” transformation.

- Image de-convolution from dirty image and point-spread-function
 - Hoǧbom CLEAN
 - Clark-Hoǧbom CLEAN
 - Cotton-Schwab CLEAN
 - Smoothness-stabilized CLEANs
 - Maximum entropy
 - Maximum emptiness

4.5.3 Data “Structure” Transformations

Data “structures” are assumed to be 1-, 2-, 3- (or n-) dimensional aggregations of data values. Included are tabular data structures that are special two-dimensional arrays which may have different (numerical) data contents for each “column”. Most of the requirements in this section should apply to any type of data, and where the operations are meaningful only for certain data types this will be noted.

- Extraction and creation of new data structures
 - Selection of a lower-dimension structure from a higher-dimension data structure (i.e., a vector from a plane or cube, a plane from a cube, a cube from a 4-dimensional structure, or an n-dimensional sub-structure from an n-dimensional structure) based on reasonable extraction criteria must be supported.
 - Selection of sub-structures as in the previous item, but with user-selectable, arbitrary “rotation” angles and regular interpolation in each dimension (i.e., arbitrary lines through planes, rotated sub-planes from planes, arbitrary planes from cubes).
 - Creation of large n-dimensional structures from smaller n-dimensional structures (tessellation of planes, cubes)
 - Extraction of vectors perpendicular to a curvilinear, user-defined track in a plane
 - Extraction of a new structure based on interpolation with respect to different coordinate system
 - Extraction of a new structure with different spatial/velocity/etc. resolution, possibly based on a new coordinate system, using convolving, fitting, or de-convolving functions
 - Extraction of a new data structure based on SQL-like queries on data values and parameters
- Generalized data structure arithmetic
 - Mathematical operations and functions for numbers and n-dimensional (vectors, planes, cubes, ...) allowing creation of new data structures

- Averaging, summing, weighted summing of data structures
- General tensor arithmetic
 1. Unary and binary matrix operations
 2. Data structure creation with specification of indices and dimensions
 3. Concatenation
 4. Inner and outer vector products
 5. Matrix inversion
- Spread-sheet like processing with arrays and numbers
- Specialized Operations on Data Structures
 - n-dimensional cube rotation and transposition
 - forward and inverse Fourier transforms for real and complex arrays as appropriate
 - non-generic Fourier transforms
 - smoothing, convolving, filtering, and histogram equalization
 - max/min, sigma clipping, mean, median, mode, edge operations
 - differentiation (gradient, divergence, curl, Laplacian) operations for vectors
 - Interpolations through blanked (missing) areas
 - “Linear” and “non-linear” registration of images
 - Operations on arrays that apply/remove primary beam or gridding correction functions
 - De-convolution of channel shapes
 - De-convolution of frequency-switched spectral line data
 - Support for error handling in analysis tasks
 - Source subtraction for standard (Gaussian) and user-defined models
 - Filtering with standard (e.g. Sobel, unsharp mask) and user-definable filters
 - Source subtraction in both image and u-v domains
 - Correction of data for source motion (asteroids, comets)
 - Modification of planetary and solar data to remove effects of disk emission, motion, and rotation
- Statistical Analysis of Data Structures
 - Histogram displays of data in selected regions
 - Noise statistics of selected areas (mean, median, mode, rms, χ^2 , etc.)
 - Power spectrum analysis
 - Structure function analysis
 - Cross-correlation analysis
- Defining Regions within data structures
 - Interactive input of rectangular (box) and curvilinear (blotch) areas of interest based upon pixel or coordinate specification, or cursor “point and click”

- Capability of saving and restoring regions of interest in files
- Flexible identification of blanked-pixel (-voxel) regions
- Capability of blanking regions based on noise limitations
- Data Analysis Operations
 - Spectral line fitting for components (Gaussian, user-defined) and baselines (polynomials, sinusoids, user-defined)
 - Automatic finding (fitting) of sources in images, generating lists of source positions and intensities
 - Fitting and removing source components (Gaussians, parabolas, etc.)
 - Functional fitting in continuum cubes (rotation measure, spectral curvature, user-defined)
 - Fitting n-dimensional surfaces with linear and non-linear (χ^2 techniques)
 - Least squares fitting (with error analysis) to ordinary and orthogonal polynomials, for equally spaced and unequally spaced data
 - Spline fitting and interpolation
 - Zonal averaging in elliptical/spheroidal rings/shells

4.6 Image Analysis

Many of the major functions of image analysis have already been discussed under the general category of data structure transformations and analysis. This is an area of applications that is highly dependent upon astronomer specification of the needs for a particular problem. For this reason tools for this analysis, and programmability by the astronomer, are most important. A few cases that illustrate advanced problems are the following,

The extraction of information from data cubes is one of the most important, but computationally (and visually) difficult areas of image analysis. The visualization problem in general requires both special hardware and flexible analysis software. The relation of most data cubes to spectroscopy, and the importance of radiative transfer to spectroscopy, presents a basic need for the astronomer to analyze data in an environment where it is possible to compute and compare models derived from spectral radiative transfer. Since this cannot be viewed as the job of any instrumental support group, the versatile programming of computational tools is the most important thing for the astronomer who has reached this stage of “image analysis”.

In addition to spectroscopy, image analysis and comparison, for some problems in the future will require dynamical gas/fluid computations. For example, analyzing HI or molecular line images of galaxies as snapshots of “fluids” means the divergence, curl, and Laplacian of vector fields must be calculated to study continuity, vorticity, and viscosity.

Moving source problems, particularly the difficult case of solar imaging, requires special modeling or data corrections. Rotation and registration of imaging taken at different times and locations, and with different instruments, requires special treatments dependent upon the scientific problem at hand, which is usually in the solar system domain involving the Sun, planets, asteroids, comets, etc.

4.7 Data Display and Recording

By data display we mean listings, plots, and “pictures” that are useful in examining data and results derived from data. By recording we mean hard copy of these data displays. The form of data display depends on the user interface. The form of data recording may depend upon printer and other hardware so output files should be as device-independent as possible, with separate production of device-dependent files.

4.7.1 General

- User selection of display and recording devices
- Flexible numerical data display (including output to files and printers) in the form of numerical tables
- Flexible plotting, and data specification for, one data variable vs. another, with optional error bars, with point type, line type, and color differentiation for multiple plots
- Contour plots of 2-D data arrays with optional number labeling, distinguishing negative contours and depressions, with color differentiation where possible
- Ruled surface plots of 2-D data arrays, with color display on that surface for another 2-D data array
- Rendered surface displays of a data cube from an n-dimensional array, with rotation, aspect, and external lighting control
- “Opacity” summation display for a displayed data cube
- Projection of user-selected image planes, or summed images, on the “sides” of an 3-D image “box”
- It should be possible to request diagnostic warnings if plot, contour, etc. are below designated noise levels
- Tiled displays of 1-D and contour plots
- Calibrated wedge displays for color and grey scale representations
- User-definable color palettes and transfer functions
- Useful “header” information should appear by default on plots, but user-defined annotations should be possible
- Flexible overlay capabilities for comparison of different types of data
- Capability of displaying tabular data in one or two windows with a corresponding X-Y plot in another, with interactive identification of points in plot with entries in table

4.7.2 Spectral Domain

- Plots of spectral profiles with user-defined superposition or tiling
- Sub-windows of spectra associated with images should be displayable, user-movable, plots with respect to an image using either superposition or with lines connect image position to spectral window display
- Contour, grey-scale, and color plots of axis of an n-dimensional plot as a function of any other (two) coordinates should be possible - subsets of these would be the common longitude-velocity plots for specific latitudes
- Flexible extraction of spectra or sequences of spectra from user-defined regions in a spectral cube
- Display of spectra and spectral cubes with and without model or continuum subtractions

4.7.3 Time Domain

- Image, ruled surface, etc. displays of variable data where one axis is time
- Period phased plots of data with user-defined binning

4.7.4 Images (Including Spectral)

- Image displays in windows with numerical and/or analog control of parameters, transfer functions, and color tables
- Cursor feedback facility of numerical information in displayed images
- Multiple image display windows (different displays of data for the same coordinate space) and overlaying of images in a given window
- Intensity-hue display and independent RGB image superposition or comparison (for appropriate hardware) of two or three images
- 4-D display of image information where intensity is a rendered surface and color on that surface is coded for a fourth parameter like rotation measure, polarization, spectral index, etc.
- User-controlled “blinking” of images
- Multi-panel displays of images related by frequency (velocity) time or other (third) dimensions
- Flexible “movie” displays of images as a function of frequency, time, etc., with interactive control of speed, zoom, and pixel display range - and optional averaging of “frames”
- Facility to return/display spectra and other data for cursor selected points (or regions) in a spectral line “cube”
- Polarization image displays with flexible display sensible combinations of intensity, polarized intensity, percentage polarization, position angle, etc.

- Image displays with histogram equalization
- Plotting of pixel values in one image vs. pixel values in another for user-defined regions
- Optional pixel histogram displays associated with images
- Superposition of multiple coordinate grids on images - pixel, equatorial, galactic, ecliptic, etc.
- “Smart” superposition of contours on image displays (contours adjust grey scale or color depending on background)
- Support of “all-sky” displays of data, wrap-around contouring
- Snapshot hard copy of both separate windows and multi-window screen displays
- Translation of image displays to input files for high quality plotting, grey scale, and color copy devices, preserving transfer functions and color palettes where appropriate
- Screen scratch pad capability added to images (and their coordinate overlays), including insertion of descriptive lines, curves, boxes, shaded areas, and text - with transfer of all to hard copy devices
- Capability to transform on-screen displays to device independent (or equivalent) files that can be used in manuscripts

5 Conclusions

The preparation of specifications for AIPS++ involves two major complications. The first is that the instruments, and the types of measurements for which they are designed, are diverse and complicated at levels of detail that are important for many applications. However, this can be handled by careful attention to detail and a judicious balance between generality and those details. In this document we have dealt mainly with generalities, so technical details needed in the implementation of software for specific instrumentation must be dealt with elsewhere.

The second, and most severe problem, is that while it may be (relatively) easy to specify what we need for the science of the past and present, the most important needs are for the scientific problems of instruments and astronomers in the future. Careful consideration of what this means for a software system leads to one general conclusion: availability of computational tools, user-programmability of these tools, and easy transferability of data between software systems are the most important capabilities that one can have for the future of any scientific software system.

6 References

- ATNF Staff, 1991, *ATNF AIPS++ User Specifications*, AIPS++ User Specifications Memo 106.
- BIMA 1991, *AIPS++ User Specifications: BIMA Version*, AIPS++ User Specifications Memo 108.
- Cornwell, T.J. 1990, *Report of the Software Advisory Group*, AIPS++ User Specifications Memo 102.

- DRAO 1991, *DRAO User Requirements - AIPS++*, AIPS++ User Specifications Memo 111.
- Foster, R., Haynes, M., Heyer, M., Jewell, P., Maddalena, R.J., Matthews, H.
 Reich, W., and Salter, C. 1991 *Requirements for Data Analysis Software for the Green Bank Telescope*, GBT Memo 72.
- GMRT group, 1992, *GMRT Requirements Documents*, AIPS++ User Specifications Memo 114.
- Hjellming, R.M. 1991, *Miscellaneous Suggestions for AIPS++*, AIPS++ User Specifications Memo 107.
- Liszt, H.S. 1992, *A Single-Dish Data Handling Environment for AIPS++*, AIPS++ User Specifications Memo 112.
- Noordam, J.E. 1991, *Dutch Requirements for AIPS++*, AIPS++ User Specifications Memo 112.
- Shone, D.L. 1992, *Jodrell Bank User Requirements for AIPS++*, AIPS++ User Specifications Memo 110.
- Wood, D.O.S. 1991, *The AIPS++ User Interface*, AIPS++ User Specifications Memo 104.