# Developments in the AST library

David Berry

University of Central Lancashire, Preston, UK

## New to AST? Start here...

- AST is a library of functions that implement an object oriented model for describing physical coordinate systems, and the transformations that exist between them.
- It provides a comprehensive range of facilities for attaching world coordinate systems (WCS) to astronomical data, for retrieving and manipulating that information and for generating graphical output such as coordinate grids based upon it.
- It can read and write WCS information in several different forms, including FITS-WCS headers.
- It is written in pure ANSI C but also has interfaces for FORTRAN, Perl and Java (via JNI).
- It has built-in intelligence for identifying flavours of celestial, spectral, time and other coordinate systems (including compound systems that combine axes of different types) and determining how to transform between them. This allows general purpose code to be written that makes no assumptions about the nature of the coordinate systems.
- It includes a flexible and versatile "tool-kit" for creating and modifying collections of coordinate frames interconnected by arbitrarily complex transformations.
- It includes easy-to-use graphical facilities that allow the production of annotated 2D or 3D grids. Graphics are draw via a simple "driver" module which AST calls to draw lines, strings, markers, etc. AST includes drivers for PGPLOT; drivers for other graphics systems (e.g. Tcl/Tk, Java/Swing, etc.) can easily be (and have been) written.
- It is actively supported and developed by the Joint Astronomy Centre, Hawaii.
- It forms the basis of the coordinate handling facilities in the Starlink software collection, including GAIA, SPLAT, KAPPA, *etc*. It is also used in other non-Starlink software such as DS9 and XIMAGE.
- More information at: **http://www.starlink.ac.uk/ast**

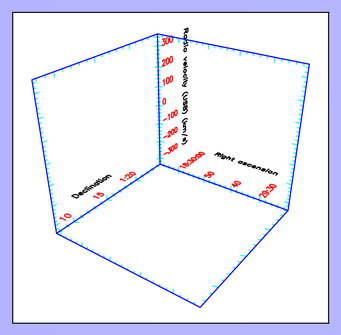



Figure 1: A set of annotated 3D axes drawn using the *Plot3D* class and the AST PGPLOT driver.

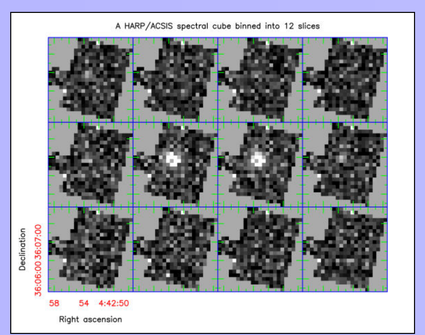



Figure 2: A single 2D image that uses a *SwitchMap* to associate different pixel->WCS *Mappings* with twelve different regions in the image.

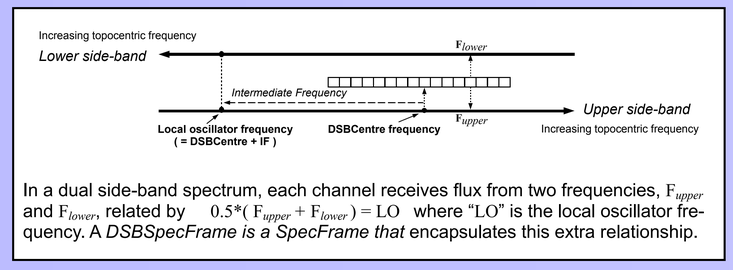
## New Features in AST:

AST has been in use within the Starlink software collection since 1997. It was last described at ADASS XIII (Strasbourg 2003). The most important of the new features added since then are:

- **Support for new coordinate systems:** In AST, the *Frame* class is used to describe a generic N-dimensional coordinate system. This base class encapsulates properties that are common by many coordinate systems, such as the observers geocentric location, the time of observation, and the units used on each axis. Sub-classes of *Frame* add more specialised coordinate system properties. The *SkyFrame* and *SpecFrame* classes describe 2D celestial longitude/latitude systems and 1D spectral systems, and have been available for some time. New classes have recently been added that model the following coordinate systems:

  1. *Time* - The *TimeFrame* class adds the following properties to those inherited from *Frame*:
     - *System*: Can be "Julian Date", "Modified Julian Date", Julian epoch", or "Besselian epoch".
     - *TimeScale*: Can be TAI, UTC, UT1, GMST, LAST, LMST, TT, TDB, TCG or TCG (whilst not all of these are true time scales, they all provide similar functionality in AST and so are grouped together).
     - *TimeOrigin*: A zero point. This allows a *TimeFrame* to describe offsets from a given moment in time rather than absolute time values.
  2. *Dual side-band spectra* - In a dual side-band spectrum, each channel receives flux from two different frequencies that are equally spaced on either side of a local oscillator frequency. The *DSBSpecFrame* class extends the existing *SpecFrame* class by adding the following properties that allow the two corresponding spectral positions to be associated with each other:
     - *SideBand:* Indicates whether the parent *SpecFrame* represents the upper or lower side-band.
     - *DSBCentre*: The central position of interest in the spectrum in either the upper or lower side-band (as indicated by the sign of the *IF* property*).*
     - *IF:* The intermediate frequency. This is the signed difference between the *DSBCentre* frequency and the local oscillator frequency.

  Axis values for a *DSBSpecFrame* can be in any of the spectral systems supported by the parent *SpecFrame* class (frequency, radio velocity, optical velocity, *etc*).




In a dual side-band spectrum, each channel receives flux from two frequencies, $F_{upper}$ and $F_{lower}$, related by $0.5*(F_{upper} + F_{lower}) = LO$ where "LO" is the local oscillator frequency. A *DSBSpecFrame is a SpecFrame that* encapsulates this extra relationship.

  3. *Observed quantity* - The *FluxFrame* class describes various systems used to measure the signal level in an observation. Currently, these are limited to flux or surface brightness per unit frequency or wavelength. There are plans to extend this to include magnitudes and antenna temperature. A *FluxFrame* can be associated with a *SpecFrame* that gives details of the frequency or wavelength system in use.

  All classes of Frame include methods that allow the transformation between two arbitrary Frames to be determined automatically. Thus AST encapsulates all the complexity of conversions between different time scales, spectral systems, celestial coordinate systems, *etc.*

- **Regions:** The abstract *Region* class extends the *Frame* class by associating a shape with a coordinate system. Concrete sub-classes of Region are provided that represent boxes, spheres, ellipses, polygons, isolated points, etc. The *CmpRegion* class is a *Region* that combines two other *Regions* (of any type) using either a logical AND or logical OR operator. A *Prism* is a *Region* that extrudes another *Region* into one or more orthogonal dimensions. All *Regions* can be negated, mapped into other coordinate systems, tested for overlap with other *Regions*, etc.
- **3D graphics:** The new *Plot3D* class draws annotated axes for a 3D coordinate system. It follows the practices of the existing Plot class (used for drawing 2D annotated axes in GAIA, SPLAT, DS9, *etc*.) in that it calls an external driver module to draw 3D lines, text strings and markers. AST comes with a driver that uses the popular PGPLOT graphics package (this driver includes code for projecting from 3D into 2D). Writing drivers for other graphics packages is a straight-forward task. Fig 1. Shows an example of a set of axes drawn using Plot3D and the AST PGPLOT driver. See also Peter Draper's poster, P1.067, "*GAIA 3D: Volume visualisation of data-cubes*".
- **New Mappings:** The following concrete sub-classes of the abstract *Mapping class* have been added:

  1. The *SelectorMap* classifies positions according to their location within some coordinate system. A *SelectorMap* encapsulates a list of *Regions,* all of which must refer to the same coordinate system. When used to transform a position within this coordinate system, the *SelectorMap* identifies which *Region* (if any) contains the position, and returns the index of that *Region* within the whole list of encapsulated *Regions*.
  2. The *SwitchMap* provides a means of switching between alternate Mappings. A SwitchMap encapsulates several other *Mappings*, one of which is used to transform input positions. The *Mapping* to use is selected by supplying its index as an extra input to the *SwitchMap*.

  A combination of a *SelectorMap* and a *SwitchMap* allows (for instance) an image to be divided into regions and a different *Mapping* to be asociated with each region. Fig 2. shows such an image in which each rectangular area represents a different slice out of a spectral cube, and has its own pixel->WCS *Mapping*.

- **Other changes**: Full details are in the AST documentation:
  - Support for HEALPix projections.
  - Support for (azimuth,elevation) systems in *SkyFrame*.
  - AST now has no external dependences.
  - AST objects can now be stored externally as XML .
  - New array re-binning functions that conserve flux.
  - A compound *Frame* (*CmpFrame)* can now be searched for specified sub-Frames using the astFindFrame method.

Dr David Berry
d.berry@jach.hawaii.edu