AIPSLETTER

Volume XIII, Number 1: July 15, 1993

A newsletter for users of the NRAO
*A*stronomical *I*mage *P*rocessing *S*ystem

Written by a cast of *AIPS*

Edited by
Eric W. Greisen
National Radio Astronomy Observatory
520 Edgemont Road, Charlottesville, VA 22903–2475
804–296–0211, x209; TWX 910–997–0174; Fax 804–296–0278
aipsmail@nrao.edu
6654::aipsmail or ...!uunet!nrao.edu!aipsmail

Personnel Changes

Geoff Croes retired from the Observatory effective July 31, 1993. Geoff came to the NRAO in December 1990 to direct the Observatory's computing efforts. Among other things, he guided the Observatory through a major hardware acquisition and began development of the replacement for the *AIPS* software, AIPS++. He approached these with vision and determination. Despite this new direction, he took care to see that "Classic *AIPS*" was well maintained. Geoff and his wife Anneke will return to their home in Okanagan Falls, near Penticton, British Columbia, Canada. There he will spread his time between his newly purchased 486 PC, his piano, bridge with friends, and perhaps an occasional look on Internet at what's going on in AIPS++. We all wish Geoff and Anneke the best.

Geoff's replacement as Assistant Director for Computing Systems will be Richard Simon who will join the NRAO by the end of August. Richard comes to us from the Naval Research Lab where he is Head of the Radio/IR/Optical Sensors Branch of the Remote Sensing Division. In addition to the knowledge of computing and management experience he brings to this position, he has research interests in high resolution astronomy, both optical interferometry and VLBI.

Glen Langston moved from the *AIPS* group to the Green Bank Orbiting VLBI Project. He is developing real-time software for antenna control, developing OVLBI interfaces, and planning a low resolution 8.5/15 GHz all-sky survey with the OVLBI antenna. His absence is keenly felt.

New members of the AIPS++ team are Terri Bottomly who has joined NRAO in Socorro and Jim Horstkotte who has transferred from the VLBA Correlator project in Charlottesville.

15JUL93 Available

The `15JUL93` version of *AIPS* is now available. It contains significant improvements in VLBI data handling and calibration, in imaging with a variety of instrumental corrections, and in support for additional operating systems including Hewlett Packard's HP-UX and Sun's Solaris 2.1. It is difficult to summarize the changes since, over the nine-month development period of this release, the changes have been numerous and multi-faceted. A number of the new tasks contain corrections for instrumental effects which have not been corrected previously, but which are of comparable magnitude to the sky curvature and other much-discussed effects. An *AIPS* Memo or other publication on these corrections is planned. An experimental package of data interface routines based on the concepts of Object-Oriented methodology has matured and come into widespread use in this release.

Supported Systems

The level of support in $\mathcal{AIPS}$ for different computers and operating systems varies widely. In general, those systems which we have in-house will be supported best, followed by those to which we have access via guest accounts and internet. Systems for which contributed code has been submitted, but for which no verification has been possible, cannot be supported at an adequate level, unfortunately.

The following table attempts to summarize $\mathcal{AIPS}$ support for a variety of more common systems. "OS" stands for Operating System; "WS" for workstation.

Computer	OS version	Level of support
Sun Sparc	SunOS 4.1.x	Full: most common WS in-house at NRAO
	Solaris 2.1,2.2	Most recent port, seems to be stable
Sun-3	SunOS 4.1.x	Moderate: none left in NRAO $\mathcal{AIPS}$ group
IBM RS/6000	AIX 3.1, 3.2	Full: NRAO's workhorses
IBM 3090	AIX ?	Poor: one port done, failed `DDT` tests
Convex	ConvexOS 8,9	Reduced: NRAO phasing out Convexes
HP 9000/7xx	HP-UX 8, 9	Good: port done via internet at JPL
DEC-Mips	Ultrix 4.0,4.3	Moderate: Mips f77 okay, Dec f77 uncertain
DEC-Alpha	OSF/1	3^{rd}-party: seems to port easily
SGI	SG Irix ?	Poor: some contributed code (out of date)
Cray	UniCos (unix)	Moderate: many 3^{rd}-party bug fixes received
Alliant	?	Poor: very old port, status unknown
Masscomp	?	Poor: very old port, status unknown
VAX	BSD 4.2 unix	Poor: old port but may still work
VAX	VMS	Moderate: older port but may still work

We currently have a DECStation 3100 workstation in-house given to us by DEC for the purpose of software testing. We have received a long-term loan from DEC of an Alpha workstation running OSF/1.

The port to Hewlett Packard was made possible by the assistance of a number of people at JPL. We would especially like to thank Taoling Xie who started the process, Peter Wannier who arranged for NRAO guest accounts, Phil Withington who provided assistance as manager of the HP cluster we used, and Tom Kuiper and Bill Langer who allowed us to use their machines and much of their disk space.

Tape End-of-Data Changed

The $\mathcal{AIPS}$ task `FITTP` has always written 2 end-of-file (EOF) marks at the end of data. While this was fine for the 9-track tapes, it has recently come to our attention that it produces inconsistent behavior on DATs and Exabytes. The remainder of this note applies only to DATS and Exabytes; the handling of 1/2-inch reel tapes has not changed.

This behavior was fixed in the `15JUL93` version of $\mathcal{AIPS}$ on March 30 1993. All tapes written with `FITTP` have the correct ("single") EOF marks at the end of data. `FITTP` with `DOEOT = 1` will work as usual for these tapes. The behavior of the corrected `FITTP` with "older" (pre-3/30/93 or pre-`15JUL93`) tapes is as follows: if you use the `FITTP` to append additional files onto an older tape, *do not* use `DOEOT TRUE` (or the verb `AVEOT` or the task `AVTP`). It will position the tape after the old double EOF and write the data, but `PRTTP` will not be able to jump over the double EOF to find it. If you do this anyway, it is possible to position the tape beyond the double EOF (with verb `AVFILE`) so that `IMLOD`, `UVLOD`, `FITLD`, and `TPHEAD` can find your data. `PRTTP` (and things like `AVEOT`), however, will stop at the first double EOF.

Rather, to append to an "old" tape with the new `FITTP`, position the tape using `AVFILE` (specifying the number of files to skip). You can then test that it is actually at the end of the data (in the middle of the old double-EOF mark) by running `TPHEAD`; it will give the message:

```
TPHEAD: END-OF-FILE = END-OF-INFORMATION?
```

This indicates that the tape is indeed at the right place, and you can now use `FITTP` with `DOEOT = -1`. From this point on, `FITTP` with `DOEOT = 1` and all other tape tasks using this tape will work as expected (using

`15JUL93` and future versions). If you use `AVFILE` to advance to the end of the tape and have not advanced over enough files, `TPHEAD` will give you header information. Try advancing some more files using `AVFILE` and try `TPHEAD` again. When you're at the right place, you'll get the `END OF INFORMATION?` message as above.

If you advance *too far* using `AVFILE` and advance past the end of your data, you will get the message:

`TPHEAD: END OF INFORMATION`

(note the lack of a question mark and the `END-OF-FILE` part of the message). At this point, use `AVFILE` with `NFILES = -1`; then run `TPHEAD` and you will get one of the above two `TPHEAD` responses. Follow the above description to append to your tape.

VLBI Summer School

NRAO held a summer school June 23–30 on Very Long Baseline Interferometry and the VLBA. The school was held on the campus of the New Mexico Institute of Mining and Technology in Socorro, New Mexico.

The NRAO has established a tradition of running summer schools every three years or so in order to educate graduate students and astronomical researchers in the techniques of radio interferometry. These schools have concentrated on techniques of relevance to the VLA; however, with the impending opening of the VLBA, it was decided that this year's summer school should concentrate on VLBI techniques. The school was attended by approximately 100 students and the lectures were given by NRAO staff and recognized VLBI experts from other institutions. The proceedings will be published later this year.

One highlight of relevance to the $\mathcal{AIPS}$ project is that, during the weekend, demonstrations were run in which a MkIII VLBI data set was calibrated and reduced using the current VLBI software within $\mathcal{AIPS}$. That these demonstrations ran smoothly is a testament to the new maturity of the VLBI calibration software within the system.

Improvements for Users in 15JUL93

VLBI data formatting and fringe fitting

The task `FITLD` is able to read standard FITS images and *uv* files, making `IMLOD` and `UVLOD` somewhat obsolete. However, its main reason for existence is to read *uv* data from FITS tables produced by the VLBA along with other table files containing source, calibration, astrometric, etc. information to accompany them. Work toward this goal has started in the `15JUL93` version, although more remains to be done as the VLBA correlator advances. VLBA users should check the anonymous ftp patch area for the latest developments (see the "Patch Distribution" article below). The task to translate Haystack MKIII VLBI format "A" tapes into $\mathcal{AIPS}$ (`MK3IN`) had bugs fixed in the differential precession correction to u and v, in tape file skipping, and in file concatenation. It was given the `SOURCES` adverb to allow source selection and more CL table entries per scan to allow for space VLBI. The new task `CL2HF` was written to convert $\mathcal{AIPS}$ calibration tables to Haystack FRNGE tables. The intention is to transfer $\mathcal{AIPS}$ fringe-fitting results to the Goddard CALC/SOLVE astrometry and geodesy package. The output of `CL2HF` is converted to the format required by SOLVE using the new task `HF2SV`.

A baseline-oriented approach to fringe fitting of VLBI data appears in the `15JUL93` release. New task `BLING` determines residual group delay and phase rate for each baseline in an array. `BLAPP` is a new task which takes these results and solves for delay and rate on a per-telescope basis. `BLING` does not require a source model, can use cross polarization data, is flexible in setting search windows, and can handle longer scans for a given number of channels and IFs. However, it requires higher signal-to-noise ratio data and is slower than `FRING` (which was improved by giving it a much larger search window). Another new task `MBDLY` fits multi-band delays from the IF phases in an SN table (created by `FRING`) and writes a new SN table.

A new procedure called `VLBA` was written to aid in the examination of data from the VLBA Correlator. This procedure will read the VLBA FITS file, generate a CL table, plot the total power data, fringe-fit, plot the delay/rate solutions, apply the delay/rate solutions and plot the calibrated visibility amplitudes and phases.

Real time filling of VLA data

Normally, VLA data is brought into the $\mathcal{AIPS}$ system through the task FILLM. It reads a tape containing data from the VLA on-line system and converts the data into the $\mathcal{AIPS}$ multi-source *uv* format. The workstation **miranda** at the VLA Site provides a second path which bypasses the magnetic tape stage. The data go straight from the on-line ModComp computer into $\mathcal{AIPS}$ using a special setup which makes the ModComp data stream appear to be another tape device.

FILLM normally creates as few files as possible; if new data are encountered which "fit"' an existing multi-source file which was created during that same run of FILLM, these data are be appended to that multi-source file. This is fine for off-line FILLM from tape, but, in the on-line case, users typically want prompt access to the newly filled data. If a file is in use (for calibration, *et al.*), FILLM cannot append to that file any more, and will fail. For this reason, on-line FILLM used to create a new file for every new scan. This ensured the accessibility of each scan as soon as it was finished, but left the user to cope with many small *uv* files.

The 15JUL93 version of FILLM includes the $\mathcal{AIPS}$ TELL mechanism to allow interaction with the executing task. On-line FILLM now starts as the off-line FILLM: appending when possible and creating as few files as possible. Using TELL, the user can "tell" FILLM to create a new *uv* file at the beginning of the next scan, whether or not this scan fits into a previous file. FILLM will then add to this next file, *never* to the first file, which therefore becomes completely available for further $\mathcal{AIPS}$ processing. Once the user wants to process the second file, another TELL command will close the second file, and create a third file. There are also other kinds of TELL commands to allow different interactions with FILLM. A more elaborate description of interactive FILLM commands is available at the AOC.

A consequence of this FILLM upgrade is that the interactivity is not restricted to running FILLM on-line at the VLA site. The same TELL commands are also effective when running FILLM off-line at your workstation; SHOW FILLM will show you the TELL parameters. Generally, this new flexibility is not very useful off-line.

Recently, the fast T1 link between the VLA site and the AOC in Socorro has made it possible to fill VLA data in real time at the AOC. First tests have been successful, but, pending policy decisions, this possibility is not yet open for general observers.

UV data manipulation and display

There are a number of new tasks to manipulate *uv* data. UVADC Fourier transforms a Clean point model, corrects for geometric distortions, and adds the result to a *uv* data set. This task is designed for use to correct the effects of minor geometric distortions due to curvature of the sky and, optionally, (1) bandwidth smearing, (2) errors in the nominal bandwidth, and (3) errors in the nominal center frequency. It is mostly of use for snapshots with the VLA. The new task PHASE estimates the coherence time on each baseline in a *uv* data set. Optionally, deviant data may be flagged and an output data set written with each baseline averaged based on its coherence time. The new task UBAVG averages a *uv* data set with the averaging time set for each baseline by the desired image field of view.

The new task SPECR may be used to increase or decrease the number of spectral channels in a *uv* data set. It does this by Fourier transforming the input data, truncating or padding the correlation function, and Fourier transforming back to the spectral domain. The new task UV2MS appends a single-source *uv* data set to a multi-source data set. This avoids the use of MULTI followed by DBCON, although INDXR and UVSRT are still required. The new task MULIF adds an IF axis (or changes its dimensionality) to a *uv* data set. This may be necessary for certain tasks which expect an IF axis or to use DBCON on data sets that have different IF dimensions. Older task UVLIN which fits a continuum to a spectral-line *uv* data set was cleaned up and corrected by Juan Uson, while task AVSPC was given the option to boxcar-average n spectral channels down to m channels.

In the *uv* display area, new task DTSUM provides the user with a matrix listing of the number of visibilities per baseline per scan. It also provides some scan information and lists the names of the antennas. The new task CLPLT replaces VBPLT for the purpose of plotting closure phases and model. VBPLT was also changed to average over time, channels, and/or IFs before plotting, and to offer the option of plotting the RMS and various random parameters instead of the data. POSSM was changed to allow multiple plots per page, to use the normal BASELINE adverb, to average over scans, and to plot cross- and auto-correlation functions of the data. POSSM also had numerous minor corrections and improvements, chief of which was to plot the amplitude rather than the real part of the bandpass function when so instructed. User control over the size of plotted points was added to UVPLT. User control over the scaling of the display and an option to display error bars rather than weights were added to PRTUV.

UV data calibration and editing

The new task FRCAL uses a polarized model of the source along with the observations to estimate and remove the effects of ionospheric Faraday rotation. As the model of the polarized emission will be corrupted by the Faraday rotation, multiple iterations of imaging followed by Faraday self-calibration may be needed. In addition, the polarization calibration may be adversely affected by variable Faraday rotation and the instrumental polarization (task PCAL) may need to be included in the Faraday rotation self-calibration loop for the polarization calibrator.

Tasks CALIB, FRING, and UVSUB were made very much faster by allowing gridded model computation methods to be applied to data in time-baseline order. This exposed some bugs related to images that were both shifted and rotated and also forced the addition of adverbs CMETHOD and CMODEL to CALIB. A channel-selection bug was also fixed in CALIB. A number of routines for doing polarization-coherent re-referencing of phases, delays, and rates to a common reference antenna in an SN table were added for use in some of the new tasks. The division and subtraction of a model image from compressed data was made functional. The Baars *et al.* coefficients for 3C295, a fundamental VLA calibrator, were added to SETJY.

The data editing task IBLED was changed to allow smaller frames to expand to fill the main display area. Such frames arise with small data sets or may be forced by the user setting a small frame with the interactive SELECT FRAME function. IBLED can also handle larger total amounts of data now. TVFLG, SPFLG and IBLED had a number of minor fixes made to allow no data flagging on input, to keep the all-channel and all-IF flags true when there is only one channel or IF in the input, and to prevent critical parameters from being uninitialized when the first display has no valid data.

Imaging

In this release there are two new and significant tasks for making images. SCMAP will do multiple iterations of self-calibration and imaging, optionally starting with a one-component model. Each iteration consists of (1) a visibility-based imaging and Clean deconvolution (MX-like), (2) a determination of the appropriate number of model components and the self-calibration *uv* range, and (3) a self-calibration of the visibility data based on the new model. A final imaging/deconvolution is then done on the final *uv* data. The products of this task are a dirty beam, a Clean image, and a calibrated set of *uv* data. This task is similar to Jodrell Bank's difference-mapping process, but without the interactive display options (so far). WFCLN also does a visibility-based Clean similar to MX, but it offers a variety of corrections related to imaging wide fields and/or multi-frequency or wide-band observations. These are (1) relative frequency-dependent primary beam corrections in the Clean subtractions from the visibility data, (2) a simple correction for source spectral index, (3) a correction for errors in assumed center frequency, (4) the use of a double size beam if possible for Cleans, (5) labeling images with average zenith and parallactic angles so that snapshot VLA images can be corrected with OHGEO for misaligned coplanar array geometric distortion, and (6) supporting 3-D DFT imaging to avoid non-coplanar array image distortion.

It has been found that Clean always underestimates the flux of a source due to "leakage" of some of the flux into components placed on peaks in the sidelobes and/or noise. To reduce this effect, the new task CCSEL will filter out weak and isolated components from the Clean components file. The FLUX adverb for MX was given additional meanings to provide more ways to determine when to stop the Cleaning. The point at which modeling switches from DFT to gridded subtraction was switched to a correct formula for Suns, which should speed up Cleaning and other model subtractions. A bug in the recording of shifts in headers of rotated images by UVMAP and MX was corrected. This error could lead to errors in the gridded subtraction of models based on these images. Another bug in MX was found that could lead (fairly rarely) to a weak sine wave in the output image. The "Prussian hat" option in APCLN was made functional and the task RSTOR was corrected to stop the addition of an uninitialized disk file to the model components.

Image handling and display

The new task OHGEO does an interpolation of one image to the geometry defined by another (like HGEOM). It offers additional options to (1) correct for 3-D distortions caused by a misaligned, but coplanar, array (as in VLA snapshots) and (2) a radial scaling of the image to correct for the interaction of a finite bandpass with the antenna primary beam size. The default positioning of the input image in the output image by HGEOM was changed for this release. The new task PASTE inserts or adds a sub-image into an image. Blanked pixels in the input do not alter the corresponding pixels in the output and image coordinates are not checked (user beware!). The new task BLWUP is used to create a larger image through pixel replication. Task SUBIM was given new options to control what is written to the output (pixel, average, maximum, minimum) when only every XINC by YINC pixel is selected. The new task IMTXT writes an $\mathcal{AIPS}$ image to a text file in the same format as used by the $\mathcal{AIPS}$ task FETCH.

The display of "stars" on various $\mathcal{AIPS}$ plots was enhanced during this cycle. In particular, the ability to translate coordinates from one epoch/type to another was added to the star-table creation task STARS. The adverb STFACTOR was extended in all appropriate plot tasks to control the labeling of the stars.

TV servers and programs

The changes made to XAS to speed its performance (shared memory, delayed screen updates, larger buffers, etc.) were detailed in $\mathcal{AIPS}$ Memo 81 (see the "Latest $\mathcal{AIPS}$ Memos" article). The frame rates reported in the Memo for the Sun workstation were measured using an X-Windows server provided by Sun called xnews. Using instead the version of the MIT X11 Release 5 server designed for Suns seems to make quite a difference in performance as the table below reveals. The results aren't totally one sided. The xnews server is faster at nearly full screen operations with shared memory, but slower at all other operations tested. The Sun numbers were re-measured as this AIPSLETTER was written and all were found to be improved by 5–15% over those measured only a few months ago. We do not know what we did right. (The computer called primate is a Sun SPARCstation IPX running Sun OS 4.1.2 while rhesus is an IBM RS6000/560 running AIX 3.2.)

TV display maximum frames / second (larger numbers are better)

Function	Size		xnews	X11R5	IBM X11R4	xnews	X11R5	IBM X11R4
Computer			primate	primate	rhesus	primate	primate	rhesus
Shared memory?			No	No	No	Yes	Yes	Yes
TVblink	518	518	4.8	5.5	10.1	14.2	18.7	18.8
TVblink	1142	800	1.5	1.8	2.5	9.8	8.4	6.3
TVmovie no zoom	258	198	21.0	24.8	34.0	49.3	80.0	52.0
TVmovie 2x zoom	570	396	3.5	4.7	5.1	7.8	9.7	7.4
TVmovie 3x zoom	855	594	1.8	2.3	2.7	4.3	5.5	4.3
TVmovie 4x zoom	1140	792	1.1	1.4	1.6	3.7	3.6	2.8

XAS was improved for the 15JUL93 release in a variety of mostly technical ways. Users may now request a higher maximum grey level ($\leq$ 237 typically) using their .Xdefaults file, although this will usually require XAS to use its own colormap. The most important correction was to add code to release any shared memory segments whenever XAS terminates. Previously, many of the ways in which XAS was killed left some of the shared memory allocated, which, in time, would cause the host computer's resources to be exhausted. We also added code in the initialization of XAS to create color maps when the default visual is not being used and to try for a GreyScale visual if a PseudoColor visual is not available on the particular display. This rather technical fix allows XAS to run on non-color and other atypical displays. The use of X11 Release 4 features was debugged and the error reporting facility was corrected.

A new task called TVCPS was written to capture whatever is displayed on the $\mathcal{AIPS}$ TV and write it into a color PostScript file. This file may be displayed immediately on an appropriate device (printer, film recorder) or saved in the user's area for later printing or inclusion in other documents. TVCPS allows the user to control the gamma correction for each color on the output device. Tasks TVRGB and TVHUI which handle 3-color imagery on any $\mathcal{AIPS}$ TV were corrected further in this release. In particular, an option to do full color optimization was added to TVHUI. A new verb called TVPHLAME was added to do a flame-like (dark red through yellow to white) pseudo-coloring on $\mathcal{AIPS}$ TVs. It has a variety of interactive intensity and coloring options.

Message server

The 15JUL93 version of $\mathcal{AIPS}$ offers a message server to go with the TV, graphics, and tape servers. The message server is a program called MSGSRV which is started automatically when the AIPS program is started (unless one is already running on your workstation) much like XAS. The server then opens a window on your workstation using X Windows and waits to receive messages from $\mathcal{AIPS}$ tasks. When it gets a message, it prepends the name of the computer which sent it and displays the result in its window. You can control what sort of window is used with an environmental variable called AIPS_MSG_EMULATOR. The default value is xterm, but other values such as aixterm, cmdtool, dxterm, hpterm, and shelltool can be used on appropriate systems. The option may be turned off by setting this variable to none. Users can also change how the message server behaves by setting variables in their home .Xdefaults file. Window colors, placement, names, scroll bars, memory sizes, and the like can be controlled for most window types. See the help file called MSGSERVER for details.

Except for the location of the messages, the message server is virtually transparent to users and programmers alike. Messages appear as fast through the server as they do on the user's main window. If the server should fail, either temporarily or completely, the messages switch automatically from the server to the main window with no more than a couple of error messages and no loss of speed. The message communication is done by a very low level Z routine, so programmers simply issue calls to MSGWRT as usual and even MSGWRT was not changed. If you tire of the message server, you may turn it off at any time by typing Control-C in the message window, or by using any of the window deletion or kill mechanisms of your workstation.

Miscellaneous changes

Other corrections and improvements made to $\mathcal{AIPS}$ for the 15JUL93 release include:

HINOTE **New** verb to append a line or a whole text file to the end of a history file.

MOUNT Tape mounts now attempt to report the type, density and location of the tape device which is mounted. This often requires that the tape be mounted physically before it is mounted in software. Some systems cannot recognize all tape types, so some of the mount messages are a bit uncertain. Tape density is selected for Exabytes ("22500" for 5 Gbyte) and 1/2-inch tapes during MOUNT.

AVEOT Advances to end-of-data on magnetic tape now attempt to report the absolute positioning achieved (if possible) or at least the number of files skipped. This required a change to TPMON which makes it incompatible with previous releases.

TAPQL **New** task to export $\mathcal{AIPS}$ tables to the Postgres relational database management system.

LWPLA The option to plot vectors with shades of grey rather than black was added.

MSGKILL The option to omit only certain messages was added.

POPS Numerous tests on stack overflow and underflow were added or corrected. Underflows due to user typos could undermine POPS language execution.

printing Print routines were altered to avoid blank pages at the beginning of print jobs.

plotting UVPLT and several other tasks were corrected to avoid unnecessary duplication of plotted points.

waiting In acknowledgement of the improvements in our computers, the wait cycles for task initiation and WAITTASK were substantially shortened .

Velocity field analysis task GAL

The galaxy velocity field analysis task GAL was put into *AIPS* in 1984, and has remained there essentially unchanged. In the meantime the task underwent steady further development at a remote *AIPS* site. The current version has been included in the 15JUL93 version of *AIPS*.

Given a velocity field, GAL will solve for parameters including the position of the kinematical center, the orbital position angle and inclination, and the systemic velocity. GAL can solve for the whole velocity field in one iteration, in which case the user must supply a functional form for the rotation curve. This specified rotation curve is controlled by one to three parameters, which are also fitted. Alternatively, GAL can work on narrow rings in a galaxy. In this case, the constant rotational velocity within the ring is fitted, and no functional form for the rotation curve need be specified. Typically, one runs GAL on the whole field first, deriving global values for the various parameters. GAL is then run on the individual rings, using the previously found values as initial estimates. The user may prefer to keep some of the parameters fixed at their global values, if it is judged that these parameters will not vary with radius. Examples are the central position and the systemic velocity. Parameters which may be expected to vary with radius are position angle, inclination, and, of course, rotational velocity. Optionally, GAL will create a residual velocity field. For more information, see the full GAL explain file.

The new GAL has various important improvements over the old version:

1. The user may specify a "weighting" image using IN2NAME. This is an image which has to coincide completely with the input velocity field. Each pixel in this image contains the weight to be applied to the equivalent pixel in the velocity field. Typical candidates for this weighting image are the total HI field, the square of the HI field, the inverse of the second moment map, the skewness of the profile, or a cleverly chosen combination of all of these.
2. GAL now handles larger velocity fields up to 512 by 512 pixels. Users with apparently larger problems should consider using SUBIM with XINC and YINC, since many velocity fields are heavily oversampled angularly. If really needed, the code can be adapted fairly easily to allow even larger velocity fields.
3. GAL now also creates a residual velocity field based on an input file. In this case, GAL does not do any fitting, but calculates the residual field straight from the input file which specifies rings, and the value of the various parameters in these rings.
4. The output plot can be routed directly to the TV using DOTV.

Improvements Primarily for Programmers in 15JUL93

Simplified *AIPS* programmer interface

AIPS contains an experimental package of data interface routines based on the concepts of Object-Oriented methodology. This package provides greatly simplified Fortran access to both the contents of data structures (images, tables and uv data sets) and operations on entire data structures (*e.g.*, image arithmetic, uv data self calibration). Access to components of data structures (*e.g.*, image pixels) is provided without requiring knowledge of AIPS I/O or catalog structures. This package has advanced to the stage where it can be used to develop custom processing tasks as well as new techniques. Performance of these routines is generally comparable to the equivalent in standard *AIPS*.

The 15JUL93 release features many new operations on data structures (especially imaging, Cleaning and self calibration) as well as significantly fewer bugs in the more primitive data interfaces. The growing use of this package in *AIPS* in emphasized by the number of new tasks in this release using it. They are BLAPP, BLING, CCSEL, CL2HF, DTSUM, FRCAL, HF2SV, IMTXT, MBDLY, MULIF, OHGEO, SCMAP, SPECR, PASTE, PHASE, TAPQL, UBAVG, UV2MS, and WFCLN. Tasks which appeared first in the 15OCT92 release which use this package are CCGAU, CXCLN, POLSN, TBSUB, and TBTSK.

The following lists describe the principal data structure operations. More details are available in the document "Object-Oriented Programming in AIPS Fortran" available in an up-to-date form by anonymous ftp. See the article below on "Latest *AIPS* Memos" for information on retrieving a copy.

Image operations

- Image arithmetic ($+ - * /$)
- FFTs of images
- Convolutions
- Deconvolution (both APCLN- and MX-type Cleans including round Clean windows)
- Interpolation, including geometric conversions/corrections
- Complex images (allows for *AIPS* "real" only images)

Table operations

- Sort
- Merge contents of a table

UV data operations

- Application of calibration and editing on read
- Time averaging
- Imaging (DFT or FFT with optional uniform weighting)
- Self Calibration
- Sorting
- Arithmetic ($+-/$) with Fourier transform of an image model; this can include the effects of geometric distortion and/or the frequency dependent primary beam shape.

Miscellaneous changes for programmers

Several rather technical but useful changes were made to this release which are primarily of interest to programmers. The most important was a rewrite of the *AIPS* Fortran preprocessor to allow INCLUDE files, including LOCAL INCLUDEs, to use the INCLUDE statement. We allow nesting only to three levels, but this should be more than sufficient. The main use of this will be to have INCLUDE files incorporate any needed parameter files and to have LOCAL INCLUDE sections incorporate any standard files that are to appear everywhere as well. The preprocessor, PP, insures that a file is not included twice within the same Fortran section. A parameter INCLUDE file called ZPBUFSZ was created to define "small" and "large" buffer sizes for *uv* data. System-dependent versions of this file should be created wherever systems have less or more memory in their computers than we assume is normal. This parameter file was put in most *uv* programs, substantially raising the buffer sizes in many cases and causing the emergence of a few bugs due to equivalences (MX) and assumptions about buffers being powers of 2 (subroutine MINSK used in two-dimensional FFTs). The parameter INCLUDE file PUVD was given a new parameter MAXCIF to define the maximum product of the number of channels times the number of IFs. This is to avoid the use of the product of the parameters MAXCHA and MAXIF which is much larger than any current instrument would require. Some programs have been corrected to use the new parameter, but others, especially those dealing with bandpass calibration, remain to be done. The *uv* data format was changed to add the integration time as a random parameter. This is transparent to all tasks except those that need the parameter and any that make stupid assumptions about the order and number of random parameters. Compression of *uv* files is now done with a standard UCMPRS subroutine rather than the messy code copied so many places from FUDGE.

The handling of library files for *AIPS* subroutines has been extended. It is now possible on Suns to have two parallel libraries, one compiled with full optimization and one with debug and little or no optimization. The debug libraries are needed only if there is significant code development planned, such as in Charlottesville and Socorro. System managers may choose this option during installation procedure INSTEP1. At this time they may also choose to have the libraries be "shared" on Suns and HPs. This makes the executable files for tasks much smaller and allows a subroutine to be replaced without relinking all the tasks which use it. However, tasks which used shared libraries include *all* of those libraries and hence require large amounts of swap space on disk for routines that they do not use. This is one of the reasons to keep local buffers in subroutines small. Some tasks are reputed to work only when linked with the standard, un-shared libraries; these are handled automatically using a list of tasks in `NOSHARE.LIS`.

Ports to new operating systems

With the `15JUL93` version of *AIPS* we have ported the package to two additional flavors of UNIX — HP-UX from Hewlett Packard and Solaris 2.*x* from Sun. The port to Solaris was reputed to be difficult; Sun provides CDs, porting tools, books, and so on to guide the people charged with porting code from Berkeley-based Sun OS 4 to Bell System V based Solaris. However, in the case of *AIPS* the port was not as difficult as might have been expected. This is to the credit of Kerry Hilldrup and Pat Murphy who kept our Z routines fairly simple and general and to Chris Flatters who began our migration to Posix-standard code some time ago.

The main requirements of a port to Solaris is a conversion of all C routines to ANSI-standard C, the elimination of all purely Berkeley system functions, and the conversion so far as is possible to functions and options which are specified in Posix. Thus, the primary reference book used was the ISO/IEC 9945-1 **Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language]**. The second most valuable reference was **Advanced Programming in the UNIX Environment** by W. Richard Stevens. The Sun porting guides were, of course, also useful.

The conversion to ANSI-standard C was, in general, fairly simple. The main requirement is to provide function prototypes for all functions called by an *AIPS* C routine and to provide return types of some sort (including `void`). This means including a number of system files (especially `unistd.h`) and declaring all *AIPS* routines called by C by including our first *AIPS* C include file called `$INC/AIPSPROTO.H`. A number of system include files have new names in Bell System V, but almost all of these name changes have been anticipated so that Bell systems generally provide the files under the new names as well. An example of this was the change from `strings.h` to `string.h`. Unfortunately, not all compilers understand ANSI constructs, so we must provide both new and old forms separated by a test on the symbol `__STDC__`. This symbol is positive for truly compliant compilers, 0 for compilers that accept old and most (but not all!) new forms, and undefined for old compilers. As an example of the new form used in *AIPS*, consider

```
#define Z_mi2__
#include <unistd.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <AIPSPROTO.H>

#if __STDC__
   void zmi2_(char *oper, int *fcb, char *buff, int *nblock,
      int *nbytes, int *ierr)
#else
   zmi2_(oper, fcb, buff, nblock, nbytes, ierr)
   char oper[], buff[];
   int *fcb, *nblock, *nbytes, *ierr;
#endif
```

The first line defines a local symbol which also appears in `AIPSPROTO.H` to prevent us from defining `zmi2_` twice. Then we include four of the most common system includes to define the functions and error code symbol used by `zmi2_`. The *AIPS* include is provided to allow `zmi2_` to call a Fortran routine `ZMSGWR` which provides access for C routines to `MSGWRT`. Finally, the function and its arguments are defined in the ANSI manner if `__STDC__` is positive and in the old way if it is not. This is somewhat pedantic since ANSI does allow the old form (at the moment) in either case, but it is useful to emphasize the different forms.

Our most Berkeley-based code is in the area of interprocess communication via sockets. It turns out that all

UNIX systems support those functions. As a result, we moved those functions from `$APLBERK` to `$APLUNIX` and did not rewrite them during the port. We might get slightly more efficient interprocess communication with streams, so it may be good to experiment with them in future, although streams may not be fully supported in Berkeley-based systems. A few Berkeley functions simply required renaming (*e.g.*, `index` to `strchr`, `bcopy` to `memcpy`, `memcpy` to `memset`). Others were omitted or heavily changed in Posix. These included a new way to access signal handling, a need to write our own password routine, a need to extend our own "create logical" routine, and our own "create a temporary file name" routine. Stevens' book was very helpful with these. In a number of cases, the old code simply used standard values where Posix defines symbolic values which we now use for things like file permissions. The minimum-match function for files in a directory was completely rewritten to use standard functions such as `readdir` and to do our own handling of the string matching. This allowed us to delete several very divergent versions of `ZTXMA2.C` which used very vendor-specific functions. In fact, quite a number of vendor-specific Z routines were deleted when the new general routines were found to be sufficient.

Some of the code is explicitly not Posix compliant. Besides the sockets, there are timers and magnetic tapes. The timers are mostly used for delaying a task, but the Posix function `sleep` takes only integer seconds which is useless for `TVBLINK` and other quick operations. Instead, we use the non-standard `setitimer` to signal the task repetitively in intervals as small as microseconds until it manages to wake up from `sleep`. Everything about magnetic tapes is non-standard. Sun appears to have no intention to develop a more standard interface within Solaris, at least for the next year or longer. Experimentation with tapes made it clear that we could use most of our old code with minor modifications for Exabytes (and presumably 1/2-inch tapes which we cannot test). However, Solaris does not support backward-space-file over a file mark for DATs. It will back up to just after a file mark, but not to just before the file mark. This causes serious modifications to `ZTAP2` about which there remains some uncertainty. The man pages claim that one can go back an extra file and then advance a large number of records to get the positioning one wants. This is too slow to be practical, but we cannot use it because it was found not to work. Solaris also stops before an end-of-file when it reads it, while other systems leave the tape positioned after the EOF. This is corrected with an advance file in `ZTPWA2`, but introduces additional uneasiness.

The port to Solaris led to numerous changes to our shell scripts. Most of these changes, however, were debugging the compilation scripts that allow us to keep both an optimized and a debug version of things. The two shell commands that had to be changed were `hostname`, which is replaced by `uname -n` (which has a bug under Sun OS limiting node names to $<$ 10 characters), and the options to the `ps` command which have to be different on Berkeley and Bell systems.

Patch Distribution

Since $\mathcal{AIPS}$ is now released only semi-annually (or even less frequently), we have developed a method of distributing important bug fixes and improvements via *anonymous* `ftp` on the NRAO Cpu `baboon` (192.33.115.103). Documentation about patches to a release is placed in the anonymous-ftp area `pub/aips/`*release-name* and the code is placed in suitable subdirectories below this. Reports of significant bugs in `15OCT92` $\mathcal{AIPS}$ have been relatively few; however, the documentation file `pub/aips/15OCT92/README.15OCT92` mentions the following items:

FILLM When appending to existing files, `FILLM` failed to update a pointer to the sky frequency. If the previous observing mode had a different correlator mode, a completely wrong sky frequency was used, resulting in crazy u ,v, and w's.

ZRLR64 For DECStations and DECSystems running ULTRIX only (actually for any computer using IEEE floating point format that has little endian byte order), bytes were not getting swapped. This can affect antenna tables and any other double precision output on FITS tape or disk file. In particular, tasks `FITTP`, `VLBIN`, and `TVMON` were affected.

XAS.SHR Changed it to make sure that it is using the default visual and, if it is not, to create and install its own color table before creating the window. Added code to use a GrayScale visual if a PseudoColor visual is not available.

Note that we do not revise the original `15OCT92` tape or `tar` files for these patches. No matter when you received your `15OCT92` tape, you must fetch and install these patches if you require them. See the `15APR92` AIPSLETTER for an example of how to fetch and apply a patch. The first two patches above were also listed for the `15APR92` release.

As bugs in `15JUL93` are found, the patches will be placed in the `ftp` area for `15JUL93`.

Latest *AIPS* Memos

Below is a list of the latest *AIPS* Memos, of which only Memo 83 is new in this AIPSLETTER.

MEMO	DATE	TITLE and AUTHOR
78	92/06/01	Object-Oriented Programming in AIPS Fortran
		W. D. Cotton, NRAO
79	92/06/09	Polarization Calibration of VLBI Data
		W. D. Cotton, NRAO
80	92/06/30	Remote Tapes in AIPS
		Eric W. Greisen, NRAO
81	92/08/26	Tape and TV Performance in AIPS
		Eric W. Greisen, NRAO
82	92/09/24	Replacing the Convexes — New Color Algorithms in AIPS
		Eric W. Greisen, NRAO
83	92/12/14	Dual Libraries and Binaries in AIPS
		Patrick P. Murphy, NRAO

Note that the version of *AIPS* Memo 83 now available has been modified slightly since it was distributed last December. In addition, a heavily revised edition of *AIPS* Memo 78 is available. It appears as file `AIPSOOF.TEX` and, in postscript form, as `AIPSOOF.PS`.

To order, use an *AIPS* order form or e-mail your request to aipsmail@nrao.edu. Memos can also be obtained via anonymous `ftp`. Some memos in the anonymous `ftp` area do not have computer readable figures. If a memo you have fetched lacks figures, write for a paper copy.

To use `ftp` to retrieve the memos:

1. `ftp baboon.cv.nrao.edu` or 192.33.115.103
2. Login under user name anonymous and use your e-mail address as a password.
3. `cd pub/aips/TEXT/PUBL`
4. Read `AAAREADME` for more information.
5. Read `AIPSMEMO.LIST` for a full list of *AIPS* Memos.

AIPS Memos from Number 69 through 83 are present in this area as well as a few of the earlier ones. All are available in paper form from Ernie Allen at the addresses in the masthead. Note that the anonymous ftp areas for memos, the `COOKBOOK`, and other text files have been changed to parallel the areas in the main *AIPS* directory tree.

AIPS Verification Package to be Changed

The set of $\mathcal{AIPS}$ procedures and data files known as `DDT` (for Dirty-Dozen Test) has become a standard for verifying that basic $\mathcal{AIPS}$ tasks continue to function correctly and for measuring the performance of computer systems. We run it, and its younger siblings `VLAC` and `VLAL`, routinely every few weeks in Charlottesville as a continuing quality check on the developing software. Over time, however, the tests have become a bit "shop-worn." The worst problem is that `DDT` depends heavily on `ASCAL`, a task we no longer recommend to users. Also, as the software has changed, the "right" output data have also changed slightly. The accumulation of these changes makes some of the curent output images only agree to 9 bits or so with the "master" images. Because of this, minor programming errors could now slip by undetected.

To correct matters, we intend to make two changes shortly. First, we will substitute `CALIB` for `ASCAL`. Had this gone smoothly, the new `DDT` would have been part of the `15JUL93` release. However, initial tests revealed that gridded model subtraction in `CALIB` and numerous other tasks did not work correctly, at least in the presence of both shifts and rotations. The bugs, some Paleozoic, have been found and corrected in `15JUL93`, but the resulting `DDT` was delayed too long to be a regular part of this release. Instead, we will announce it soon and place it in the anonymous ftp area. The second change will be to compute new master images. We will then start with high agreement between computed and master images and should be able to detect fairly subtle errors. The new master images will also be available by anonymous ftp, including `LARGE` if we can find enough disk space for it.

The `VLAC` and `VLAL` tests are in better shape, but we do expect to make improvements to the checking of tables and, perhaps, recompute the master files.

An important aspect of the `DDT` test is its ability to measure a computer system's ability to handle a real mix of operations. This is an important addition to the standard performance tests run on computers. They measure questions of tight vectorization, small loops, and the like, but fail to address the problem as a whole which is all the end user actually cares about. In one notorious case, a vendor claimed through the standard tests to have a substantially improved performance from a new (and expensive) cpu board. When the board was installed, however, `DDT` was able to confirm the users' impressions that the delivered performance had not changed. The current formula used for the "$\mathcal{AIPS}$ Mark" is

$$M_A = \frac{5000}{T - 0.6T_a}$$

where T is the total real time of the `LARGE DDT` test in seconds and T_a is the total real time of the `ASCAL` step in seconds. The performances reported in $\mathcal{AIPS}$ Memo No. 75 in these units were 2.08 (IBM RS/6000 550), 1.01 (Convex C1), 0.69 (Sun SparcStation 2), and 0.36 (Sun IPC). IBM reported measurements of 2.565 $\mathcal{AIPS}$ Marks for an RS/6000 560 and 2.122 for a model 550. Colin Lonsdale (Haystack) reports a value of 2.26 $\mathcal{AIPS}$ Marks for an HP 735 under conditions which suggest that this might be somewhat of an underestimate. Mauro Nanni (Bologna) reports 2.34 $\mathcal{AIPS}$ Marks for a DEC Alpha 3000/500 and 1.73 for a Convex C210. We will need to develop a new formula for M_A that reproduces these results with the revised `DDT`. We will announce the results in an $\mathcal{AIPS}$ Memo or other suitable forum.

World Coordinates and Other FITS Documents

The subject of "world coordinates" was first addressed in $\mathcal{AIPS}$ with the publication of $\mathcal{AIPS}$ Memos Number 27 and 46. In these memos, the scheme by which $\mathcal{AIPS}$ would handle celestial and velocity coordinates was explicated. Other software systems have used the same scheme for, among other things, infrared and X-ray imagery. In 1988, a group of people representing a wide variety of areas in astronomy met in Charlottesville to discuss how coordinates really should be represented in FITS. That meeting concluded that, with some additions, we could use a scheme similar to that used by $\mathcal{AIPS}$ for celestial coordinates. We were unable to reach any agreement on velocity/frequency coordinates at that meeting. Bob Hanisch (STScI) and Don Wells (NRAO) wrote up some notes from that meeting, but nothing further was done until late in 1992. Then, at the ADASS '92 meeting in Boston, it was decided that a definitive paper on coordinates was now an urgent requirement. Accordingly, Eric Greisen (NRAO) and Mark Calabretta (ATNF) have written a paper entitled "Representations of Celestial Coordinates in FITS." This paper is a draft of a proposed

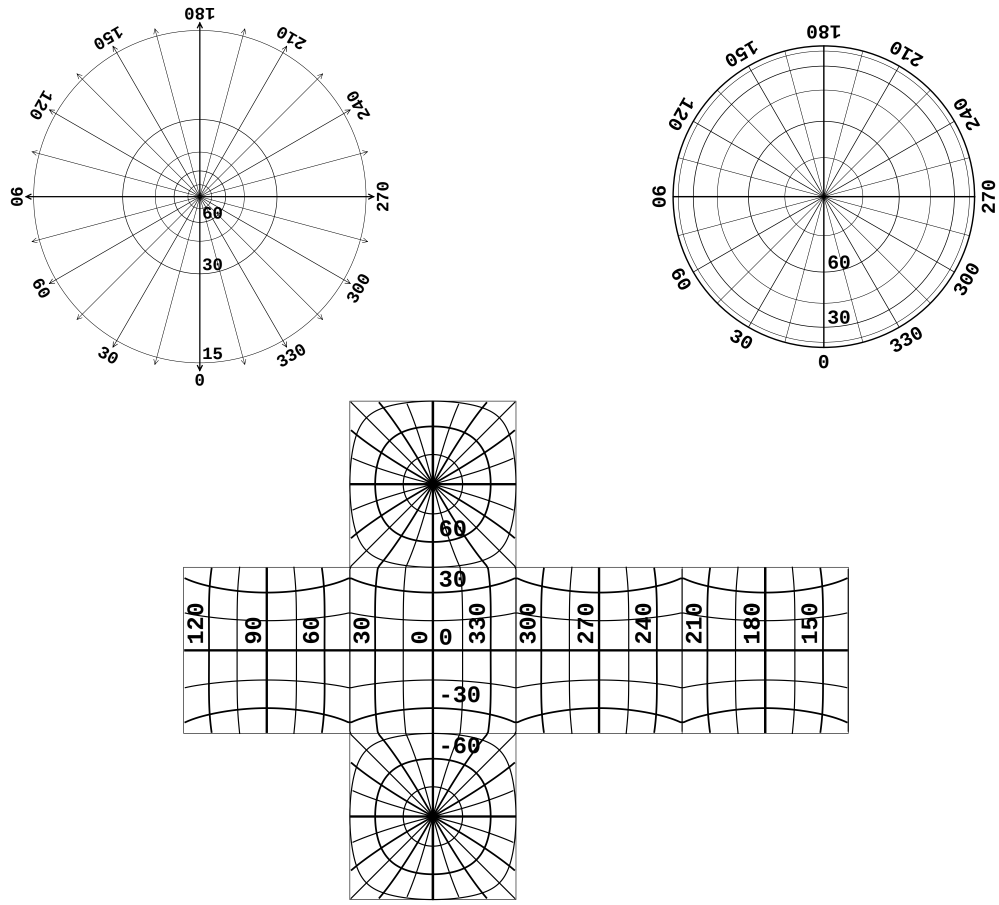

Figure 1: *top left:* `TAN` projection, diverges at $\theta = 0$; *top right:* `SIN` projection, North and South sides begin to overlap at $\theta = 0$; *bottom:* Cobe Quadrilateralized Spherical Cube projection, no limits

standard and we urge all interested $\mathcal{AIPS}$ users and other astronomers to consider the document closely since it will affect all software systems and all representations of coordinates in Astronomy.

This document may be obtained in postscript form by anonymous ftp from the server called `fits` (fits.cv.nrao.edu, 192.33.115.8). It is in directory area `fits/documents/wcs` and is called `wcs.ps.all.Z`. Note that the file is in UNIX compressed form which requires fetching in binary mode and uncompressing. This file is interesting in part because all of the figures are, in fact, postscript programs which compute and draw the geometries using the computers within your postscript printers. A couple of examples of figures from this paper are shown here. We have found that some "postscript" printers cannot manage some of the figures. Other versions of the draft document, called "most", "some", and "none", are also available in this ftp area. These contain fewer of the figures and may then be more printable if less complete. A paper copy may be obtained from Ernie Allen at any of the addresses in the masthead. This anonymous ftp area also contains modern postscript versions of the old $\mathcal{AIPS}$ Memos (`aips27.ps` and `aips46.ps`), the Hanisch and Wells notes (`wcs88.ps`), and a poster and talk on the new proposal given at the Berkeley AAS meeting in June (`Poster93.ps` and `Talk93.ps`).

The `fits` anonymous ftp server provides a wealth of information about FITS above and beyond the coordinates discussion. Anyone interested in how we exchange data would profit by wandering through the wide variety of things offered through this server. Since how we interchange data more or less controls how we structure data internally and, hence, how we think about astronomical problems, all astronomers ought to have at least some interest in this subject. We recommend starting by reading the `README` file in the ftp subdirectory `fits` and then following what interests you. The proposals on binary tables, image extensions, and blocking of FITS files which are now being officially considered and endorsed are found in area `fits/documents/proposals`.