6 Displaying Your Data

This chapter is concerned with the ways in which you may display your data. There are a number of tasks for generating "plot files" which contain graphics commands for the making of various displays of your *uv* and image data. All of these now offer a "preview" option to draw the plot directly on the $\mathcal{AIPS}$ TV, rather than putting the commands into a file. Once the files are created, a variety of tasks may be used to translate them into displays on various devices, such as the $\mathcal{AIPS}$ TV and graphics windows and PostScript printers. There are also verbs to display and manipulate your images on the $\mathcal{AIPS}$ TV and a single task `TVCPS` to capture that display, if desired, into a PostScript file for printing, recording on film, or even including in your scientific papers.

Several indices of the $\mathcal{AIPS}$ software are relevant to this discussion. To generate current lists of $\mathcal{AIPS}$ functions on your workstation window (or terminal) use ABOUT HARDCOPY C_R, ABOUT INTERACT C_R, ABOUT PLOT C_R, ABOUT TV-APPL C_R, and ABOUT TV C_R. Recent versions of these indices are reproduced in Chapter 13 of this $\mathcal{C}ook\mathcal{B}ook$.

6.1 Getting data into your $\mathcal{AIPS}$ catalog

By the time you reach this chapter, most of your data will probably already be loaded into your $\mathcal{AIPS}$ catalog either by reading an external tape or disk or by being generated by some $\mathcal{AIPS}$ task. Visibility data which are not presently on disk may be read by the $\mathcal{AIPS}$ tasks `FILLM`, `UVLOD` and `FITLD`; see § 4.1 and § 5.1 for details. Images that are generated by other imaging systems, (*e.g.*, images from non-NRAO radio telescopes or non-radio images) can be transported to $\mathcal{AIPS}$ by writing them out of the other imaging system on tape or disk in the standard FITS format. The tasks `IMLOD` and `FITLD` can then be used to read them into $\mathcal{AIPS}$. These tasks are also used to read images saved with `FITTP` and `FITAB` from previous $\mathcal{AIPS}$ sessions.

6.1.1 `IMLOD` and `FITLD` from FITS-disk

`FITLD` and `IMLOD` can read FITS-format images from external disk files or magnetic tape into your $\mathcal{AIPS}$ catalog. The disk-file option is indicated by setting the adverb `DATAIN` to a non-blank value. Disk image files must be read in only one at a time per execution of `IMLOD`, but `FITLD` can read more than one FITS-disk file if the file names are identical except for sequential post-pended numbers beginning with 1. `DATAIN` is a string of up to 48 characters that must completely specify the disk, directory, and name of the input disk file to your computer's operating system. See § 3.10.3 for a discussion of FITS-disk files.

One "feature" of $\mathcal{AIPS}$ complicates this otherwise straightforward disk analog of FITS tape reading. `AIPS` translates all of your alphabetic inputs to upper case (this was demanded by users who otherwise became confused between upper and lower cases).[1] So if your computer distinguishes upper and lower cases for disk, directory, or file names, you probably should do two things to prepare for this before running $\mathcal{AIPS}$. First, restrict your external disk file names to upper-case characters and numbers. Second, set an upper-case "environment variable" or "logical" to point to the disk and directory where your FITS-disk images are stored before you run `AIPS`. You may need help from your System Manager when doing this for the first time. A common strategy on UNIX machines is to create an upper-case logical name after logging in but before starting up `AIPS`:

% setenv MYLOGICAL *myarea* C_R if using C-shell, or

[1] If you omit the close quote on the character string, it is not converted to upper case, allowing you to circumvent this $\mathcal{AIPS}$ limitation. The string without a close quote must, of course, be the last thing on the line.

$ export MYLOGICAL=*myarea* C_R   if using korn, bourne, or bash shells,

where MYLOGICAL is an all-upper-case string of your choice and *myarea* is the full path name of the disk directory that contains your FITS-disk data. *AIPS* usually provides a public disk area known as FITS which you may use.

Then, once inside AIPS, tell FITLD or IMLOD:

> DATAIN 'MYLOGICAL:IMAGE.DAT' C_R	to read in the FITS-disk file *myarea*:IMAGE.DAT.

To check that the file name is correct, type:

> TPHEAD C_R

Your terminal will then list information about the image header of the disk file.

> OUTDI *n* C_R	to specify writing the image to your *AIPS* catalog on disk *n*.
> OUTNAME '*your-chosen-name*' C_R	to specify the output disk file name in *AIPS*; the default is the image name on tape if FITTP was used to write the image to tape.

The string *your-chosen-name* can be any (≤ 12-character) title that you want to use as the image name within *AIPS* and should be specified for images from other image-processing software systems. FITLD also allows you to specify the 6-character image "class" parameter. Use OUTCLASS '*abcdef*' C_R, if you wish to change the class from that in your input file as the image is read or if the image comes from a "foreign" system.

> OUTS -1 C_R	to keep the sequence number the same as that in the file; the default is the highest unique number for images with this name and class in your current *AIPS* catalog.
> NCOUNT *m* C_R	in FITLD only, to load *m* images consecutively starting with the file specified by DATAIN with a 1 post pended. The other files must be the same name but with sequential numbers post pended. If you use this option, do *not* specify the OUTNAME unless you want the same name for all the new images in your catalog.
> GO FITLD C_R	or IMLOD, to run the task.

If OUTNAME is left unspecified, it defaults to the "name" of the image read from the FITS header — either the name previously used in earlier image processing or the source name. If OUTCLASS is unspecified, it defaults to the Class previously used in earlier image processing or to a compound name (*e.g.*, IMAP, IBEM, QMAP, ICLN) which attempts to describe the image. These defaults are frequently good ones when you are loading multiple consecutive images with NCOUNT > 0. You may of course change the *AIPS* image and class names later by using RENAME (see § 3.3.3 of this *CookBook*).

IMLOD and FITLD can also read images from magnetic tape; see the instructions in § 3.9. Use the verbs MOUNT to mount the tape, AVFILE to advance the tape to the desired file, and TPHEAD to check that the tape is correctly positioned. Set NFILES=0, NCOUNT to the number of consecutive tape files you wish to load, and the say GO to either FITLD or IMLOD.

6.1.2 Image coordinates

Images of the celestial sphere must be rendered with some sort of coordinate "projection." *AIPS* pioneered the handling of such coordinates, supporting true projective coordinates called SIN (orthographic), TAN (gnomic), ARC (zenithal equidistant), and STG (stereographic) as well as a special version of orthographic suited for East-West interferometers (NCP). *AIPS* also supports modern[2] interpretations of GLS (SFL Sanson-Flamsteed), MER (Mercator), AIT (Hammer-Aitoff), CAR (Plate carrée). MOL (Molweide), and PAR

[2] Calabretta, M. R. and Greisen, E. W. 2002, "Representations of celestial coordinates in FITS," *Astr. & Ap.*, 395, 1077.

(parabolic) "projections." These are normally used to represent large sections of the celestial sphere. Users should be warned that reference pixel values other than zero lead to *oblique* projections.

Fits to the coordinates on optical images often lead to a modest amount of skew in the image. This is represented in the `PC`*i_j* or `CD`*i_j* FITS header cards. If these cards contain any significant skew, `IMLOD` and `FITLD` will tell you about it. In such cases, run `DSKEW` on the images after they have been loaded into *AIPS*. This is the only task that understands coordinate skew and it will re-grid the images for use by all other *AIPS* tasks.

6.2 Printer displays of your data

The most old fashioned way to look at your data — and the most exact — is simply to print it out and read the numbers. *AIPS* provides a variety of tasks and verbs to print visibility data, image data, tabular data, and miscellaneous other information. All of these tasks and verbs allow you to specify where the printed output goes using two adverbs, `DOCRT` and `OUTPRINT`. If `DOCRT` ≤ 0 and `OUTPRINT` is blank, then the output is placed in a temporary file and queued to the printer you selected when starting `AIPS`; see § 2.2.3. If printing directly to the printer is requested, most tasks and verbs count the number of lines of print and request permission to proceed if the count is large. (Type PRINTER *n* C_R to change the line printer selection to that numbered *n*; type PRINTER 999 CR to see the devices available to you and their assigned numbers.) If `DOCRT` ≤ 0, a non-blank `OUTPRINT` specifies a text file into which the output is to be written; see § 3.10.1. The current output is appended to the file if it already exists. Thus, you can combine a number of printed outputs for later editing and/or printing. When `DOCRT = -1`, the output print file will contain full paging commands and headers. To suppress some of this, use `DOCRT = -2` or to suppress almost all of it, use `DOCRT = -3`. This last is especially helpful when writing programs to read the text file. If `DOCRT` > 0, the output is directed to your workstation window or terminal. All printer verbs and tasks are able to respond to both the width and height of your workstation window. Set `DOCRT = 1` to use the current width; set `DOCRT` $= n \geq 72$, to use *n* as the width of the display window. Since most print tasks display more information on wider windows, we recommend widening your window to 132 characters and specifying `DOCRT = 1`. The print routines will pause whenever the screen is full and offer you the choice of continuing or quitting. Thus, you can start what might be a very long print job, find out what you wanted to know after a few screens full, and quit without using up any trees.

6.2.1 Printing your visibility data

Before beginning calibration, it is a very good idea to make a -summary list of the contents of your data set. `LISTR` with `OPTYPE = 'SCAN'` will list the contents of each scan in the data set. `LISUN` produces the same output with the addition of the angle between each scan and the Sun. It also honors `TIMERANG` and `SOURCES` unlike `LISTR` in its `SCAN` mode. `DTSUM` also produces a listing summarizing the data set in either a condensed or full form.

The most basic display of your visibility data is provided by `PRTUV` which lists selected correlators in the order they occur in the data set:

> TASK 'PRTUV' ; INP CR	to review the inputs.
> INDI *n* ; GETN *ctn* CR	to select the disk and data set to print.
> CHANNEL *c* ; BIF 1 CR	to print starting with channel *c* from IF 1.
> BPRINT *m*; XINC *i* CR	to print every $i^{\rm th}$ visibility starting with the $m^{\rm th}$ visibility in the data set.
> DOCRT 1 ; GO CR	to run the task with display on the terminal.

When you have seen enough, enter q C_R or Q C_R at the page-full prompt.

You may limit the sources, range of projected baselines, and times displayed and may select only one baseline or one antenna. `PRTUV` does not apply calibration or flagging tables. To get a similar display with all "standard" calibration, flagging, and data selection (optionally) applied, use the task `UVPRT`. `LISTR` also uses all of the calibration options to list the data in simple lists or in a display showing all the baselines at each time in a matrix form. `SHOUV` also lists calibrated visibility data with options to average all channels in each IF and to display closure rather than observed phases. `ANBPL` converts baseline-based amplitudes, phases, or weights into antenna-based values and prints and/or plots them. The display of antenna-based weights before and after amplitude self-calibration is a particularly useful tool for spotting calibration/instrumental problems.

There are a number of tasks used to diagnose possible problems in your data and to print information about them. `UVFND` examines a data set for excess fluxes, excess apparent V-polarization, or simply any data with a specified fringe spacing and position angle or a specified range in u and v. As it does this, it also checks for bad antenna numbers, bad times, and (optionally) bad data weights. `CORER` examines a data set for excessive mean values and rms in each correlator (after applying calibration and flagging and then subtracting a point source at the origin). `RFI` examines the rms fluctuations in the real and imaginary visibilities of each correlator looking for (and reporting) periods of apparent RF interference. `UVDIF` directly compares two data sets reporting any excess differences. It is useful for determining whether your latest operations (flagging, self-cal) have made a significant (or any) difference.

6.2.2 Printing your image data

The most basic display of an image is a print out of the numbers it contains. Such a display is provided by `PRTIM`:

> TASK 'PRTIM' ; INP C_R	to review the inputs.
> INDI *n* ; GETN *ctn* C_R	to select the disk and image to print.
> NDIG 3 C_R	to use 3 digits, printing numbers between -99 and 999 with appropriate power of 10 scaling.
> FACTOR 10 C_R	to raise the default scaling by a factor of 10, overflowing regions of high values to see low valued regions better.
> BLC 0 ; TRC 0 C_R	to see the whole image.
> XINC 2 ; YINC 2 C_R	to see every other column and every other row.
> DOCRT FALSE ; GO C_R	to print the image on the selected printer.

Other imaging tasks which can use the printer are `BLSUM` and `ISPEC`, which compute and print spectra by summing over regions of each plane in a data cube (see § 8.6), and `IMFIT`, `JMFIT`, and `SAD`, which fit one or more Gaussians to an image (see § 7.5). `IMTXT` writes an ASCII-formatted file containing an image.

6.2.3 Printing your table data

If you have any doubts about the contents of tables in AIPS, it is best to resolve them by looking at the contents of the tables involved. `PRTAB` is a very general task which will print the contents of any AIPS table file. For example, to print flag table version 1:

> TASK 'PRTAB' ; INP C_R	to review the inputs.
> INDI *n* ; GETN *ctn* C_R	to select the disk and catalog entry to print.
> INEXT 'FG' ; INVERS 1 C_R	to select flag table version 1.

> BPRINT 0 ; EPRINT 0 ; XINC 1 C_R	to print everything.
> DOHMS TRUE C_R	to print times in sexagesimal notation.
> DOCRT 1 ; GO	to print the flag table on the terminal.

When you have seen enough, enter q C_R or Q C_R at the page-full prompt. For a table with a significant number of columns, PRTAB shows all rows for the first columns and then loops for the next set of columns. To see all columns for some rows, set a low EPRINT value or be very patient. Enter a list of column numbers in BOX to see only some of the columns. NCOUNT, BDROP and EDROP control which values are displayed in those columns having more than 1 value per row. Adverb RPARM lets you limit the display to rows having specific column values within specified ranges, while NDIG controls the format and accuracy used to display floating-point columns.

Some of the tables have specialized printing programs. These include PRTAN for antenna tables, PRTCC for Clean component tables, SLPRT for slice files, and LISTR with OPTYPE = 'GAIN' for calibration, solution, and system temperature tables. The verb EXTLIST will list information about various extension files, particularly slice and plot files (see below), which may be printed with PRTMSG. OFMLIST is a verb to print the contents of an *AIPS* TV color table. Finally, task TBDIF will compare columns of two tables and print information about their differences. PRTSY and SYPRT offer differeing views od the contents of EVLA SysPower tables.

6.2.4 Printing miscellaneous information

There is a variety of miscellaneous information which may also be sent to the printer in the same way. Verb PRTMSG prints selected contents of the *AIPS* message file; see § 3.2. Verb PRTHI prints selected lines from a history file; see § 3.4. Pseudo-verb ABOUT prints lists of *AIPS* symbols by category while pseudo-verbs HELP and EXPLAIN print information about a selected symbol; see § 3.8. Task PRTTP prints the contents of magnetic tape volumes and pseudo-tape disk files; see § 5.1.1 and § 3.9.4. Task PRTAC, which may also be run in a stand-alone mode, prints information selected from the *AIPS* accounting file.

Task TXPL will attempt to represent an *AIPS* plot file (see below) on the printer. This will not work well for complicated plots, but, for simple plots, it may be the only way someone running over a slow telephone line can see his/her data in plot form.

6.3 Plotting your data

The basic concept in *AIPS*' plotting is to use some task to create and write a device-independent plot file as a PL extension file to a cataloged image or visibility data set and then to use some device-dependent task to interpret that file for the desired output device. Plot files are not overwritten by subsequent plot tasks. Instead they make new plot files with higher "version" numbers. The device-dependent tasks include TVPL (*AIPS* TV devices including XAS), TKPL (Tektronix graphics devices including *AIPS*' TEKSRV server), TXPL (line printers), and LWPLA (PostScript printer/plotters). Tasks called PRTPL, QMSPL, and CANPL support antique Versatec, QMS, and Canon printer/plotters. To plot on a PostScript printer/plotter:

> TASK 'LWPLA' ; INP C_R	to review the inputs.
> INDI *n* ; GETN *ctn* C_R	to select the disk and catalog entry to print.
> PLVER *m* ; INVERS 0 C_R	to plot the m^{th} plot file only.
> OUTFILE ' ' ; GO C_R	to do the plot immediately.

LWPLA offers the option to save the file for later plotting or inclusion as encapsulated PostScript in other documents. It also has options to control scaling, output paper size, width and darkness of lines, and transformation of grey-scale intensities. It can write more than one plot file at a time and can append new plots to existing output files. Multi-plot files are not "encapsulated" but may be printed and viewed with tools such as gv or ghostview. Note that PostScript files are text files and *AIPS* writes particularly simple PostScript so that it can be modified by the users. See HELP POSTSCRIPT for suggestions including information on deleting and adding labels and arrows and on converting the PostScript to other formats like jpg without loss of resolution.

LWPLA and all plot tasks offer some "coloring" options. These are illustrated in the color pages at the end of this chapter. The grey-scale plotting tasks, including GREYS, PCNTR, and KNTR, can now enhance the grey-scales with a transfer function and then pseudo-color them with a color table. See § 6.4.3 for a short discussion of "output-function memory" tables which may be read into the above tasks or to LWPLA using the adverb OFMFILE. Lines plotted on top of grey scales (*e.g.*, contours, polarization vectors, stars) may be "dark" when the grey scale intensity is high. LWPLA may be instructed to plot these as bright if adverb DODARK is false. All plot programs can draw lines of different types in both bright and dark forms. In LWPLA, if DOCOLOR is true, the array adverb PLCOLORS(i, j) controls the red, green, and blue colors ($i = 1, 2, 3$, resp.) of line types $j = 1$ – 10. The normal meanings of these types are:

1. Bright labeling, tick marks, surrounding lines
2. Bright lines, usually contours or model curves
3. Bright lines, usually polarization vectors
4. Bright lines, usually symbols such as stars, visibility samples
5. Dark labeling text inside plot area
6. Dark lines, usually contours
7. Dark lines, usually polarization vectors
8. Dark lines, usually symbols such as stars
9. Bright labeling outside the main plot area, e.g. titles, tick values and types, documentation
10. Background for the full plot

Some plot tasks have the ability to control the colors of their line drawing independent of these line types. These colors may be controlled only when making the plot file with the particular task. PCNTR and KNTR have acquired the ability to color each contour level under control of the adverb RGBLEVS. Using system RUN file SETRGBL, procedures CIRCLEVS, RAINLEVS, FLAMLEVS, and STEPLEVS are available to help you set the values of RGBLEVS. Examples are shown on the color pages at the end of this chapter. With LWPLA and these options one may prepare extremely effective displays — or hopelessly bad ones — for use in talks and, since the prices have become reasonable, even in journals. Note that most journals want color images in CMYK (cyan-magenta-yellow-black) rather than RGB; use DPARM(9) = 1 in LWPLA to get PostScript files with this color convention. Note that these two color representations usually require different "gamma" corrections; the adverb RGBGAMMA allows this control in LWPLA.

All *AIPS* plot tasks now offer a "preview" option. If you set DOTV = TRUE when running any plot task, then the plot appears immediately on the *AIPS* TV display and no plot file is generated. This option allows you to make sure that the parameters of the plot are reasonable and lets you avoid making files and wasting paper for quick-look plots. Additional options allow you to control which graphics channel is used for the line drawing (GRCHAN) and to select pixel scaling of the plot at your specified location on the TV screen (TVCORN). These two options allow you to view more than one plot at a time on the TV, usually for purposes of comparison. Each graphics channel on the TV has a different color and a complementary color is used when two or more channels are on at the same point. This allows for a fairly detailed and effective comparison of plots, all of which may be captured with task TVCPS (see below). Be aware that most tasks

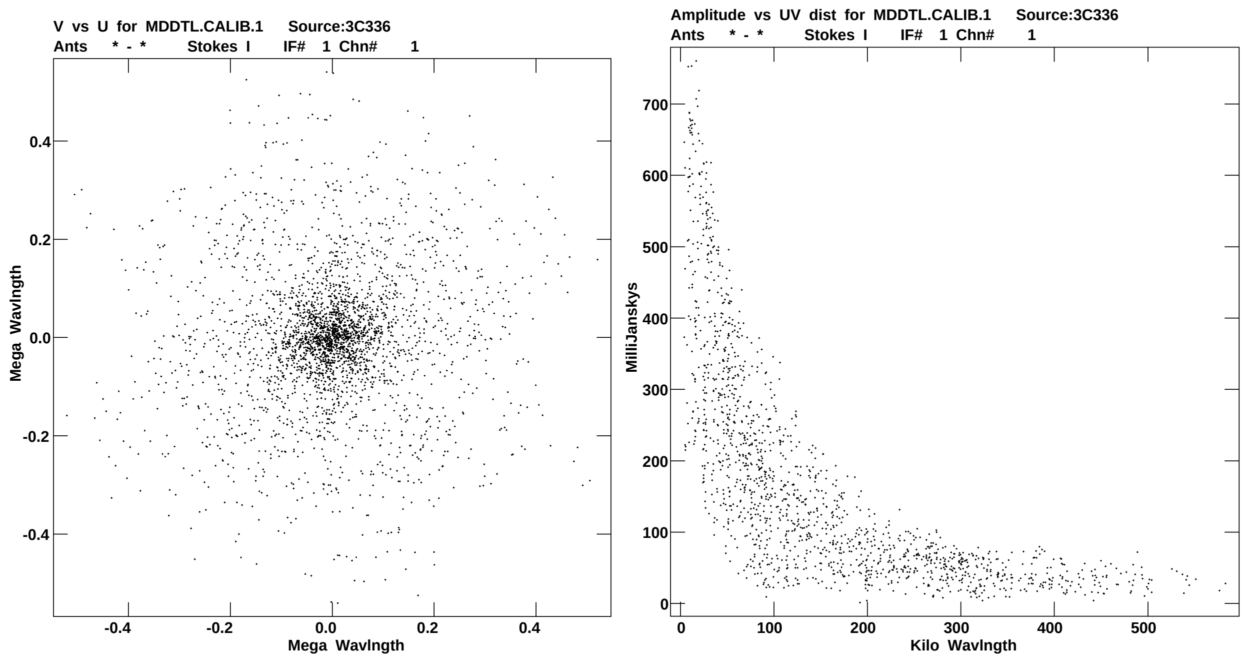


Figure 6.1: `UVPLT` displays of where the data were observed in the *uv* plane (left) and of the visibility amplitudes as a function of baseline length (right). The data on 3C336 were provided by Alan Bridle from observations made with the VLA on 6 December 1987.

now interpret `GRCHAN = 0` as an instruction to use graphics channels 1 through 4 for line types 1 through 4 and graphics channel 8 for dark vectors. The comparison function is only achieved by specifying `GRCHAN`; see HELP GRCHAN for more information. Tasks that produce multiple plot files pause for 30 seconds at the end of each plot when `DOTV = TRUE`. This allows you to stop the task (TV button `D`), hurry it along (TV buttons `B` or `C`), or make it pause indefinitely (TV button `A`) until another TV button is pressed.

You can review the parameters of the plot files associated with a given image or visibility data set by typing:

> INDI *n* ; GETN *ctn* C_R	to select the disk and catalog entry to print.
> INEXT 'PL' ; EXTLIST C_R	to list summaries of the plot file contents.
> PLVER *m* ; PLGET C_R	to recover all adverb values used when making the specified plot file.

Plot files (and other "extension files") are automatically deleted when an image is deleted by `ZAP`. However, large plot files should be deleted as soon as they are no longer needed:

> INP EXTDEST C_R	to review the inputs required.
> INEXT 'PL' ; INVERS *m* C_R	to set the type to `PL` (plot) and the version number to be deleted to *m*. $m = -1$ means all and $m = 0$ means the most recent (highest numbered).
> EXTDEST C_R	to do the deletion.
> INVERS 0 C_R	to reset the version number to its default — usually advisable.

Plot files are not amenable to the FITS format and so are not written by `FITTP` and `FITAB`. They may be copied from one catalog entry to another with `TACOP`.

6.3.1 Plotting your visibility data

The most basic plot program for visibility data is called UVPLT. It allows you to select the *x* and *y* axes of the plot from real, imaginary, amplitude, log of amplitude, phase and weight of the visibility, time, hour angle, elevation, azimuth, and parallactic angle, and projected baseline length, position angle, *u*, *v*, *w*, frequency, spectral channel number, and baseline number. It offers all of the usual calibration and data selection options and it plots the selected points individually and/or in a controlled number of bins along the *x* axis. It can display multiple IFs and spectral channels at once, including averaging groups of spectral channels. It can also time average the data (SOLINT minutes) prior to plotting. For example, to plot calibrator phases as a function of time for all baselines to one antenna:

> TASK 'UVPLT' ; INP C_R	to review the inputs.
> INDI *n* ; GETN *ctn* C_R	to select the disk and catalog entry of the data set.
> BPARM = 11 , 2 C_R	to plot time in hours on the *x* axis and phase in degrees on the *y* axis.
> SOURCES ' ' ; CALCODE '*' C_R	to select all calibrator sources.
> XINC 4 C_R	to plot only every fourth selected sample.
> ANTENNA 2,0 ; BASELINE 0 C_R	to do all baselines with antenna 2.
> DOTV = -1 ; GO C_R	to make a plot file of these data.

After UVPLT is running, or better, after it has finished:

> PLVER 0 ; GO LWPLA C_R	to plot the latest version on a PostScript printer/plotter.

There are several other tasks to plot your visibility data. VPLOT plots all of the parameters offered by UVPLT, but one baseline at a time with multiple baselines per page (*i.e.*, per plot file) and multiple pages per execution. CLPLT is a similar task but restricted to plotting closure phases around baselines involving 3 antennas as a function of time. CAPLT is a similar task but restricted to plotting closure amplitudes around groups of 4 baselines as a function of time. All three of these can also plot a source model (based on Clean components) as well as the observations. In 31DEC22, task CLOSE can plot the rms of closure phase or amplitude as a function of spectral channel. ANBPL can create plots of amplitudes, phases, or weights converted to antenna-based quantities; this function is helpful in diagnosing problems with your data. For spectral-line users, POSSM plots visibility spectra averaged over selected baselines and time intervals. It can also plot various spectral tables (bandpass and others), as well as the Fourier transform of visibility spectra (the auto- and cross-correlation functions). POSSM can also plot other spectral tables (BD, PD, CP, PC, and PP). For VLBI users (primarily), FRPLT plots visibilities versus time or, more importantly, the fringe rate spectrum. To examine the statistical distribution of your data, try UVHGM which plots histograms showing the number of samples or weights versus a wide variety of parameters. Optionally, it will fit a Gaussian to the histogram and plot the result. The task UVIMG will grid visibility data into an image on a variety of axes, while the task UVHIM will make an image of a two-dimensional histogram of your *uv* data. All image display functions below may then be used to view the results. For example, if the axes of the histogram image are the real and imaginary parts of your data, then the image will demonstrate the amplitude and phase stability present (or missing) in your data.

Two tasks, new in 31DEC21, compute and then plot statistical information from calibrated autocorrelation data. VBRFI computes the statistics as a function of frequency and antenna and then prints them to a text file and, optionally, plots them. PLRFI can plot one or more of the text files written by VBRFI. As the names suggest, these are intended to describe the RFI environment of the antennas.

A number of visibility plotting and editing tasks offer the DO3COLOR option to separate multiple IFs, spectral channels, sources, times, and the like when plotted in the same plot area. Task PLOTC will make plots showing the colors used by such tasks with labels to assist in identifying which color goes with which data parameter. These plots may be saved as plot files and shown directly on the TV just as with all other plot tasks.

Three tasks allow you to examine selected portions of a *uv* data set statistically. UVRMS averages the selected data in SOLINT time intervals and then can plot the average values as a function of time and as a histogram. A variety of statistical parameters are computed and printed. SPRMS determines and plots the mean and standard deviation of the selected data as a function of spectral channel. Note that both tasks will combine data from multiple sources, baselines, etc. if so instructed. RIRMS computes statistics over the included visibility data and then prints a matrix over baselines of the real and imaginary parts separately. Histogram plots and plots over time may be generated.

Two plot programs actually convert visibility data to the image plane for plotting. Observers of point objects which might vary with time either intrinsically or by scintillation (*e.g.*, stars, masers) might wish to try DFTPL, which plots the Fourier transform of the data shifted to a selected position as a function of time. Control over Stokes including polarization amplitude and popsition angle are available in 31DEC24. Tasks which save an image for later plotting include DFTIM which makes an image of such data as a function of frequency and time and ACIMG which makes a time and frequency image of autocorrelation data by antenna. VLBI spectral-line observers may need to use FRMAP, which performs imaging via fringe-rate inversion and plots the loci of possible source positions.

6.3.2 Plotting your image data

Image data may be drawn in a variety of ways including contours, grey or color levels, row tracing, and statistical. All plots are drawn with labeled tick marks although these may be suppressed with the LTYPE parameter. This LTYPE parameter is used to control the type of axis labels, the degree of extra labeling, the presence of the line giving the plot time and version, and also how metric scaling is done on the axis units. Read HELP LTYPE for details. For plots having significantly non-linear coordinate axes, *e.g.*, wide-field images, it is sometimes useful to draw a full, non-linear coordinate grid rather than just short lines at the edges of the plot. Tasks like CNTR and even the verb TVLABEL offer this option; enter DOCIRC TRUE C_R.

Plot symbols (*e.g.*, plus signs) may be drawn on the plots produced by CNTR, PCNTR, GREYS, KNTR and several of the other tasks mentioned below. In these tasks, the parameters which controls the plotting are STFACTOR, a scale factor for the symbols, and STVERS, a choice of the desired stars extension file. When using this option, there must be a table of "star" positions associated with the image being plotted. To create one, enter EXPLAIN STARS C_R to learn the format of the input data file and the parameters for the task. See also Appendix Z or your local equivalent for instructions on editing text files. A star file may also be created by MF2ST from a model fit file produced by task SAD (see § 7.5.3 and § 10.4.4). Tasks IMFIT, JMFIT, and SAD may now also produce "star" extension files directly.

Example outputs of the following three tasks are given in Figure 6.2.

6.3.2.1 Contour and grey-scale plots

The most basic contour drawing task is CNTR. In addition to the usual image selection parameters, you may specify:

> TASK 'CNTR' ; INP C_R	to tell you what you may specify.
> BLC 250 , 230 C_R	to set the bottom left corner of plot at 250, 230 (in pixels with 1,1 at extreme bottom left of the image).
> TRC 300, 330 C_R	to set the top right corner of plot at 300, 330.
> CLEV 0 ; PLEV 1 C_R	to get contour levels at 1% of the peak image value.
> PLEV 0 ; CLEV .003 C_R	to get contour levels at 3 mJy.

> LEVS -1, 1, 2, 4, 6 C_R	to get actual contours at -1, 1, 2, 4, and 6 times the basic level set by PLEV or CLEV. LEVS need not be integers, but very fine subdivisions cannot be represented accurately on the plot.

N.B., if you request more than one negative level with the LEVS input, you *must* use commas between the negative levels. Otherwise the minus sign(s) will be treated as subtraction symbols and the desired levels will be combined into a single negative level by the AIPS language processor. BLC and TRC can be initialized conveniently from the TV display using the cursor with the TVWIN instruction (see § 6.4.4). Then check:

> INP C_R	to review what you have specified.
> GO C_R	to run the task when you're satisfied with the inputs.

This generates a plot file as an extension to your image file, with the parameters you have just specified. Watch the *AIPS* monitor (which, on some systems, is your terminal) to see the progress of this task. If the "number of records used" in the plot file is over 200, the contour plot will be messy (unless the field is also large). In this case, check that you have not inadvertently set PLEV or CLEV, for example, to unrealistically low values. Printing a large, messy plot file on the printer can take a considerable length of time and will inconvenience other users. Consider plotting directly on the TV first (DOTV = TRUE) to check on your selection of contours.

PCNTR plots polarization vectors on top of contours and/or grey-scales. You may make a polarized-intensity image and a polarization position-angle image from the Q and U images (see § 7.1.2) or use the Q and U images themselves. Then:

> TASK 'PCNTR' ; INP C_R	to review the input parameters.
> INDI *n1* ; GETN *ctn1* C_R	where *n1* and *ctn1* select the disk and catalog numbers of the image to be contoured.
> IN2DI *n2* ; GET2N *ctn2* C_R	where *n2* and *ctn2* select the Q or polarized intensity image.
> IN3DI *n3* ; GET3N *ctn3* C_R	where *n3* and *ctn3* select the U or position-angle image.
> IN4DI *n4* ; GET4N *ctn4* C_R	where *n4* and *ctn4* select the grey-scale image.
> PCUT *nn* C_R	to blank out vectors less than *nn* in the units of polarized intensity.
> ICUT *mm* C_R	to blank out vectors at pixels where the total intensity (image 1) is less than *mm* in the units of image 1.
> DOFRACT 1 C_R	to convert the polarization lines to fractional polarization, dividing by the I image, before plotting.
> FACTOR *xx* C_R	to set the length of a vector of 1 (in units of total or fractional polarization) to *xx* cell widths.
> DOCONT 1 ; DOVECT 1 ; DOGREY 4 C_R	to request vectors plus contours of image 1 and grey scale of image 4.
> PIXRAN T_{min}, T_{max} ; FUNCTYP ' ' C_R	to scale linearly the grey-scale values from T_{min} to T_{max}.
> OFMFILE 'RAINBOW' C_R	to use the standard rainbow-colored OFM table to pseudo-color the grey scales.
> CBPLOT 1 C_R	to plot the Clean beam in the lower left corner. See HELP CBPLOT for numerous options.
> PVPLOT 4 C_R	to plot a polarization vector of stated length in Jy/beam in a box in the upper left corner.
> INP C_R	to review your inputs and remind you of others. Most are similar to those in CNTR and sensibly defaulted.
> GO C_R	to generate the plot file, which can then be routed to output devices via TKPL, TVPL, LWPLA etc.

Unless images 2 and 3 are of Q and U polarization, the lengths of the vectors are controlled by image 2 while the directions of them are controlled by image 3. Clearly this program can also be used for other combinations of images, so long as one of them represents an angle. PCNTR and following tasks have the option to draw an image of the Clean beam under control of adverb CBPLOT. Polarization vectors may be plotted with the color representing the angle. The value of POL3COL, if greater than zero, is that angle represented in pure red from 0 to 180 degrees. A color "spray" is plotted to calibrate the eye. To clarify the length scale of polarization vectors, a vector may be drawn in a box in an image corner under control of PVPLOT with the length of that vector in Jy/beam shown in a text line below the plot. The ability to plot multiple spectral planes in colored contour or polarization vectors was also added. The adverb CON3COL controls this function. The ability to select the colors of each contour level with RGBLEVS has been added. These color functions are displayed at the end of this chapter in the color pages.

GREYS creates a plot file of the grey-scale intensities in the first input image plane and, optionally, a contour representation of a second input image plane. Like the other grey-scale plotting tasks, GREYS can interpret a true-color (RGB) image cube in its "true" colors. Unlike the others, it can construct the true-color image from 3 separate image planes. A sample set of inputs could be:

> TASK 'GREYS' ; INP C_R	to review the inputs.
> DOCOLOR 1 CR	to specify that a "true-color" image is to be plotted.
> INDISK *n1* ; GETN *ctn1* CR	to select the red image.
> IN3DISK *n3* ; GET3N *ctn3* CR	to select the green image.
> IN4DISK *n4* ; GET4N *ctn4* CR	to select the blue image.
> PIXRAN T_{minr}, T_{maxr} ; FUNCTYP 'SQ' CR	to scale by a square-root function red values from T_{minr} to T_{maxr}.
> APARM $T_{ming}, T_{maxg}, T_{minb}, T_{maxb}$ CR	to scale green and blue values similarly over ranges T_{ming} to T_{maxg} and T_{minb} to T_{maxb}, respectively.
> BLC 250 , 250 , 3 CR	to select the lower left corner and the plane in the first image.
> TRC 320 , 310 , 12 CR	to select the upper right corner in the first image and, with TRC(3), the plane in the second image.
> DOCONT TRUE CR	to specify that contours are to be drawn.
> IN2D *n2* ; GET2N *ctn2* CR	to select the contour image.
> PLEV 0 ; CLEV 0.005 CR	to select 5 mJy/beam contour increments.
> LEVS -3 , -1 , 1 3 10 30 100 CR	to plot contours at -15, -5, 5, 15, 50, 150, and 500 mJy/beam.
> DOWEDGE 2 CR	to plot a 3-color step-wedge along the right-hand edge; 1 for along the top and 0 for no wedge.
> DOPROFIL 1 CR	to plot a "profile" of the average of each column of the image.
> DOTV FALSE ; GO CR	to make the plot file.

In 31DEC25, GREYS has the added option of plotting a "profile" at the top averaging all values in each column of the image and/or at the right averaging all values in each row of the image. This is illustrated in Figure **??**. New task PRPLT will plot up to 3 profiles generated by averaging an image cube along 2 axes as a function of the third axis.

When GREYS has finished, run LWPLA to view the plot file. Note that LWPLA has a variety of options which control the plotting and scaling of the grey-scale images without having to rerun GREYS. In this example case, you should remember to set FUNCTYPE = ' ' and DPARM = 0 (or at least the first 4 values to 0) in LWPLA to avoid additional scaling. You may wish to color the labeling, contours, and background with DOCOLOR=1 and PLCOLORS with LWPLA. The procedure TVCOLORS will set PLCOLORS to match the TV graphics-plane colors, DEFCOLOR will set PLCOLORS to the standard TV colors even in the absence of a TV display and OKCOLORS will set colors which work well with a white background, using less printer ink. See examples on the color pages at the end of this chapter.

There are two other contour drawing tasks which offer additional options. KNTR is able to draw multiple contour, polarization, and/or grey-scale images in a single plot file, primarily to show multiple planes of a spectral-line cube; see § 8.5.4. KNTR uses a different, and probably superior, method of drawing the contour

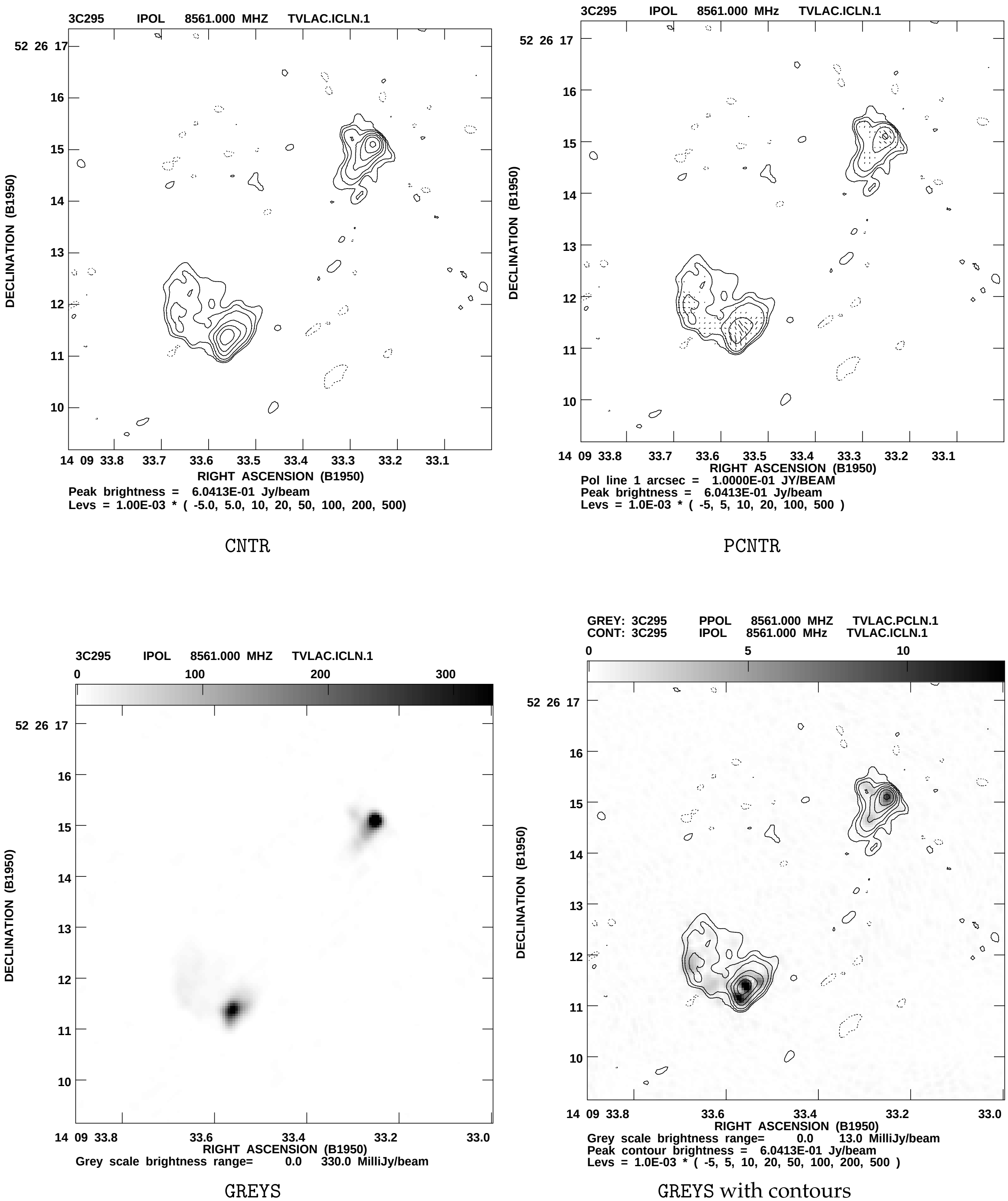


Figure 6.2: Contour, polarization, and grey-scale plots of an image

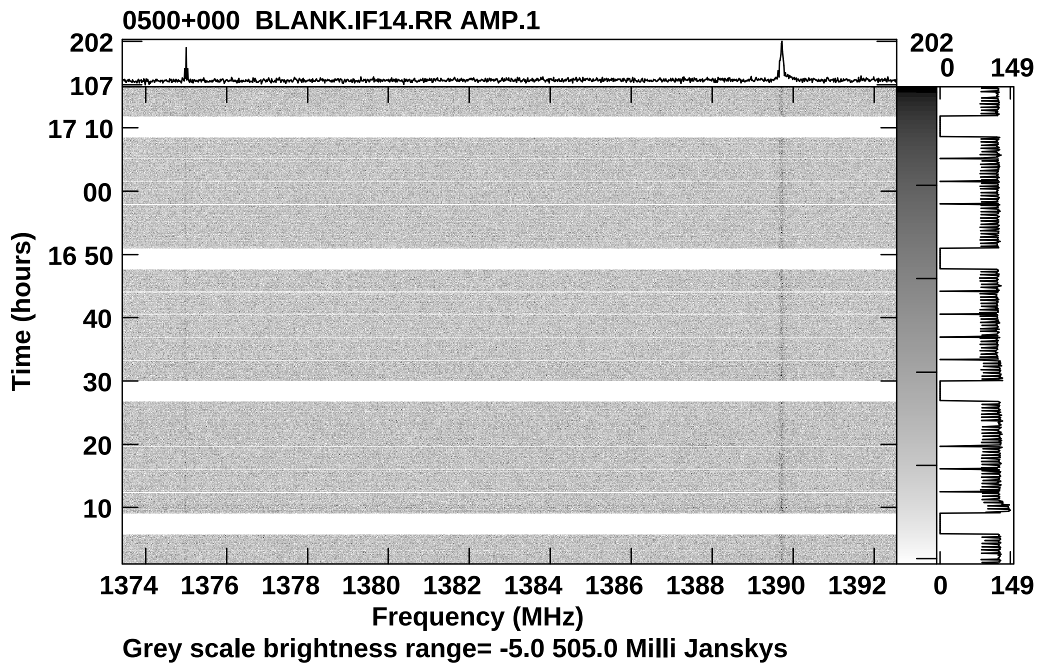


Figure 6.3: GREYS plot of the output of DFTIM with profiles at top and right-hand side (DOPROFIL = 3).

lines. It also can use color to represent different spectral channels and/or polarization angles or simply different colors for the different contour levels. CCNTR is virtually identical to CNTR except that it can draw extra symbols on the plot representing the locations and intensities of source model components found in CC (Clean component) or MF (model fit Gaussians from SAD, see § 7.5.3 and § 10.4.4).

6.3.2.2 Row tracing plots

There are a number of tasks which plot rows directly. Two of these are for use with single image planes while others are more intended for use with, *e.g.*, spectral-line data cubes transposed into velocity-ra-dec order. Of the former, PLROW is the simpler. It makes a plot file of all selected rows in an image plane. Each row is plotted as a slice offset a bit from the previous row. Low intensities which are "obscured" by foreground (*i.e.*, lower row number) bright features are blanked to keep the plot readable. Example inputs would be:

> TASK 'PLROW' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to select the image on disk *n* catalog slot *ctn*.
> BLC 100 ; TRC 300 C_R	to select the sub-image from (100, 100) to (300, 300).
> YINC 3 C_R	to plot only every 3rd row.
> PIXRANGE -0.001 0.050 C_R	to clip intensities outside the range -1 to 50 mJy.
> OFFSET 0.002 C_R	to set the intensity scaling such that 2 mJy separates rows of equal intensity.
> INP C_R	to check the inputs.
> GO C_R	to run PLROW.
> GO LWPLA C_R	to display the plot file on the laser printer after PLROW has finished.

The plot files produced by PLROW are a simple, special case of those produced by PROFL. This task makes a

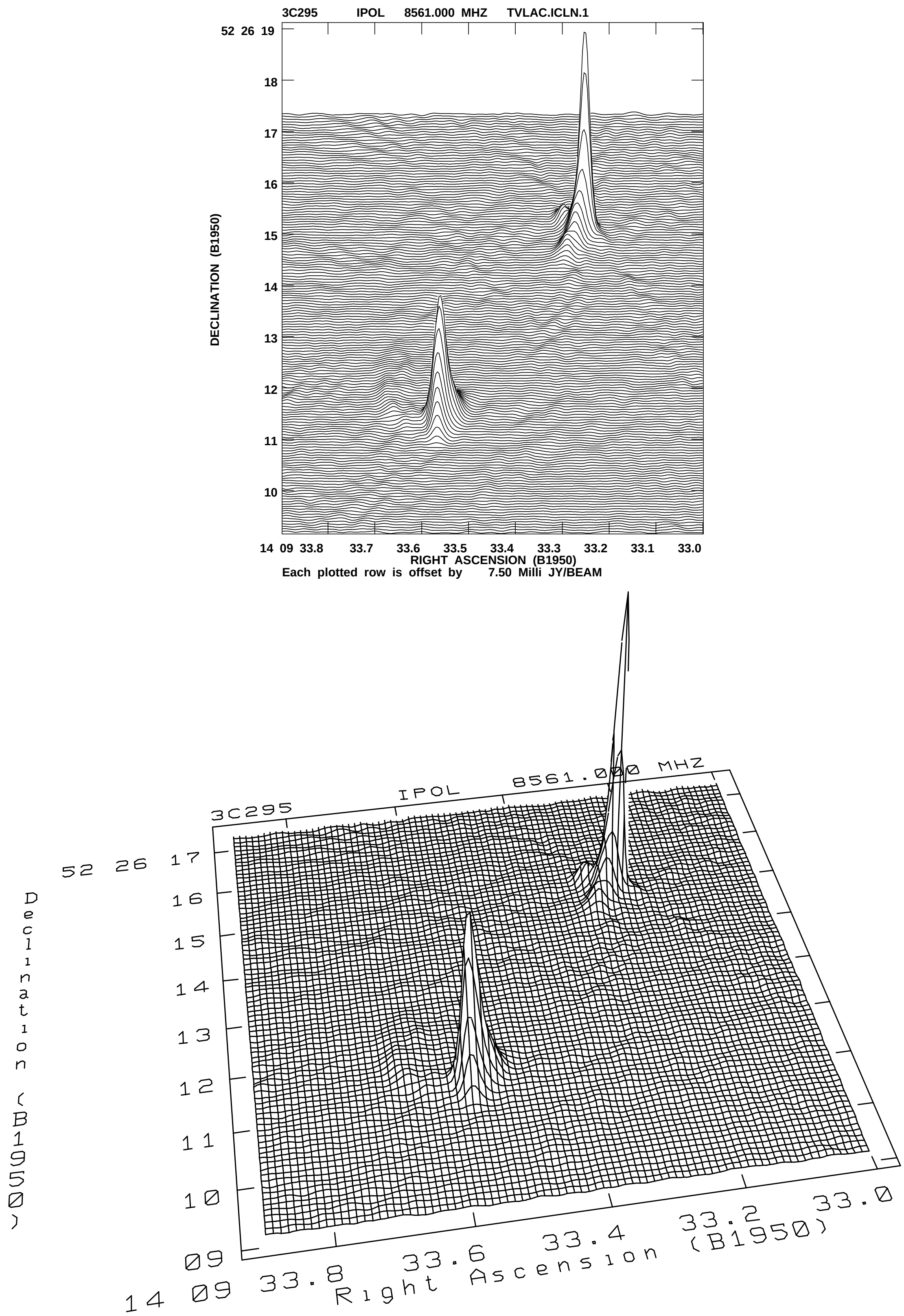
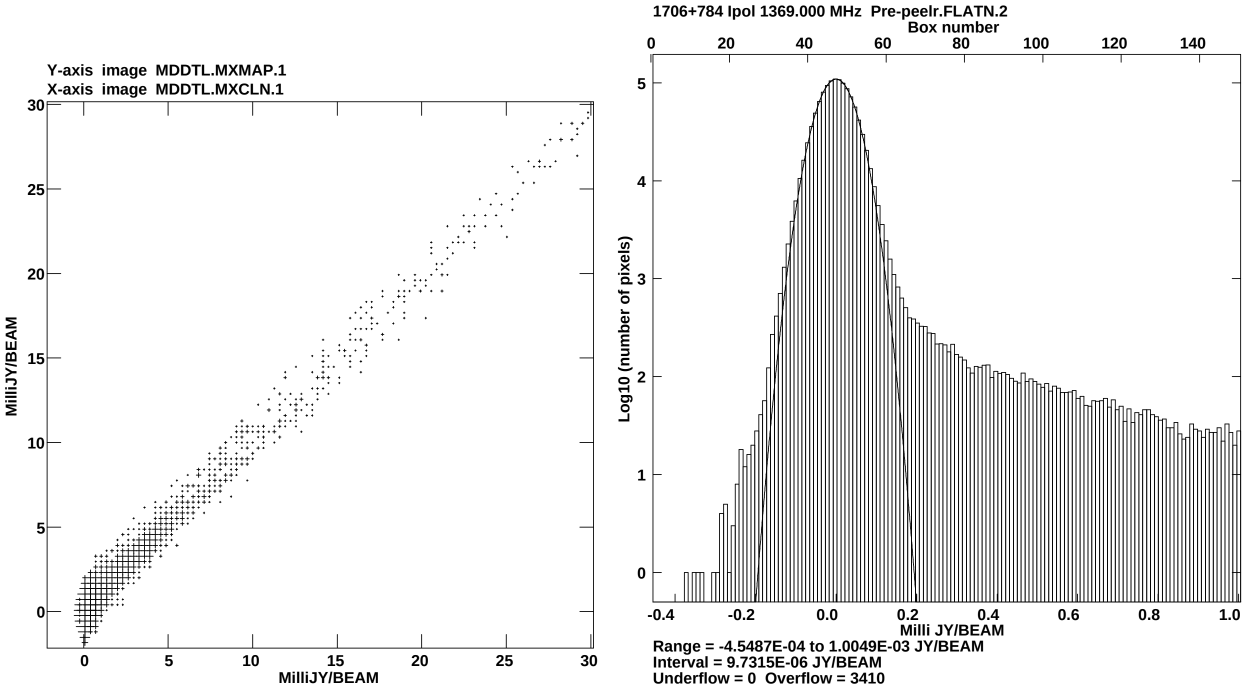


Figure 6.4: Row plots of an image with (bottom, PROFL) and without (top, PLROW) perspective



IMVIM (binned) IMEAN

Figure 6.5: Plots of statistical parameters of an image.

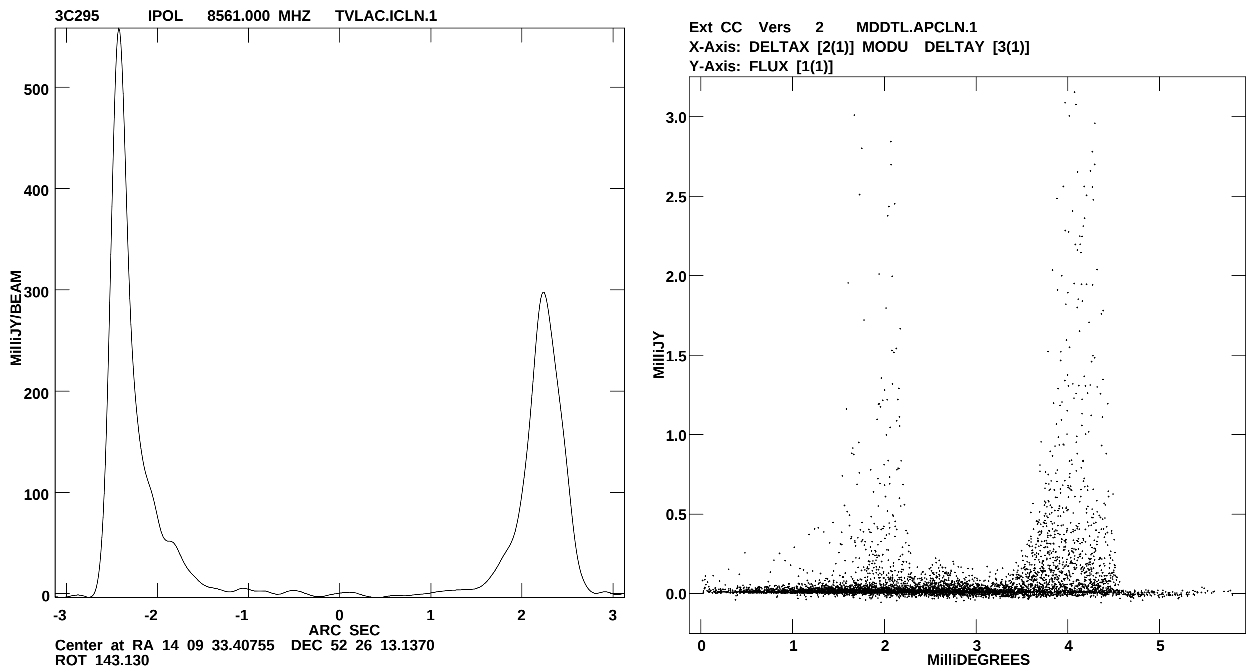
plot file of a "wire-mesh" representation of an image plane complete with user-controlled viewing angles and correct perspective. Enter EXPLAIN PROFL C_R for a full description. Both of these tasks are especially useful where the signal-to-noise ratio is high and examples of them are given in Figure 6.3.2.2.

In Chapter 8 we discuss the computation and use of "slices," one-dimensional profiles interpolated along any line in an image plane. Once a slice has been computed, it may be plotted by SL2PL on the TV or into a device-independent plot file.

Three other row-plotting tasks, PLCUB, ISPEC, and BLSUM, are designed primarily for spectral-line and other data "cubes" (see § 8.5.4 and § 8.6). PLCUB makes one or more plot files showing the intensities in each selected row. The row subplots are positioned in a matrix in the coordinates of the 2nd and 3rd axes of the cube. ISPEC averages circular or rectangular areas in each plane of a cube and plots the resulting spectrum. It can also save the output in a SLice file. RSPEC does the same except that it plots the robust or median absolute deviation ("MAD") rms in each plane rather than the data. RSPEC has options to write out a signal-to-noise image and/or a text file of channel weights as well. BLSUM can plot spectra from irregular regions selected on the TV with a "blotch" algorithm.

6.3.2.3 Miscellaneous image plots

IMVIM allows a variety of image comparisons by plotting the pixel values of one image against the pixel values of another image. The special options include binning the values (and plotting symbols proportional to the number of samples in a bin) and shifting one of the images in x and/or y with respect to the other. The former reduces large scatter diagrams to more manageable sets of numbers while the latter allows cross-correlation functions to be developed.



SL2PL TAPLT

Figure 6.6: Slice and table plots.

IMEAN prints the mean, rms, and extrema inside or OUTSIDE a user-specified window in an image. It also prints the intensity at, and rms of, the noise peak in the histogram and returns these values as adverbs to AIPS. It also returns an array in which TRIANGLE(i) is the brightness level above which there are i per cent of the image pixels. This is often useful in setting PIXRANGE for optical images. Optionally, IMEAN plots histograms of image intensities over the window using a user-specified number of summing cells over a user-specified range of intensities. Optionally, it also plots the fit to the noise peak on the user-specified histogram. An example of this is also shown in Figure 6.5.

6.3.3 Plotting your table data

TAPLT is a very general task to plot information from AIPS table extension files. It can plot a histogram of a function of the values in one or two columns of the table and it can plot a function of one or two columns against another function of another one or two columns. The latter can be summed or averaged in bins or have every point plotted. At first blush, the inputs seem rather complicated, but the results may well justify some effort to understand them. For example, to plot Clean component fluxes as a function of radius from the image center:

> TASK 'TAPLT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to select the image on disk *n* catalog slot *ctn*.
> INEXT 'CC' ; XINC 1 C_R	to select every row in the Clean components file.
> BCOUNT 1 ; ECOUNT 0 C_R	to select all rows in the table.
> APARM = 2 , 1, 3 , 1 , 16 , 1 , 1 C_R	to plot the modulus of columns 2 and 3 on the x axis (*i.e.*, $\sqrt{\Delta_x^2 + \Delta_y^2}$) and column 1 on the y axis (*i.e.*, component flux).
> BPARM = 0 ; CPARM = 0 C_R	to use self-scaling of the plot and no scaling of the column values.

> DOTV TRUE ; GO to plot the fluxes on the TV screen.

You may need to set the scaling with BPARM after seeing the preview plot in order, for example, to bring out the details in the low-level components.

There are a few tasks intended to make plotting specific kinds of tables rather easier. SNPLT plots calibration, solution, system temperature and EVLA SysPower tables for selected antennas as a function of time, antenna elevation, hour angle, azimuth, or sidereal time. It will make several plots per page (one antenna per plot) and multiple pages if needed, or you can plot the most discrepant value over all antennas in a single plot. In one execution, polarizations and IFs may be plotted on separate plots or together on the same plots, optionally separated by color. SNPLT also has the option to separate sources by color instead. You can also plot multiple parameter types in the same execution. SNBLP plots calibration solutions much like SNPLT, but on a baseline basis, to investigate exactly how the solutions affect the visibility data. SNIFS is similar to SNPLT but is intended to compare solutions at different IFs, plotting the IF on the x axis with a variety of binning options.

POSSM will plot bandpass tables when APARM(8) is set to 2. Values 3 – 10 plot bandpass-like tables from BLCHN (BD), PCAL (PD antenna polarization and CP source polarization), VLB pulse-cal (PC), and RLDIF (PP phase difference). POSSM can also do multiple plots per page with all the usual data selection adverbs. BPLOT also plots BP and the other similar tables, with multiple times for one antenna or multiple antennae for one time on each plot. Combined with difference and coloring options, this allows close comparison of bandpasses as functions of time or antenna. BLPLT makes a similar plot of the BL table written by BLCAL. BPEPL overplots multiple times within a BP table using color to separate times. It is similar to BPEDT without the interactivity. In 31DEC23, BPPLT plots selected portions of bandpass tables as a function of time in a manner similar to SNPLT. It is particularly useful when BPASS is used instead of CALIB to provide the gain calibration. WETHR plots the data in a WX weather table including parameters computed from those data such as relative humidity. FGPLT plots the times of selected flags from a flag table. PRTAN can plot antenna locations. Task TEPLT may be used to plot the ionospheric Faraday rotation and other parameters used by TECOR and written in a TE table. New tasks ALBUS and RIFRM also write the TE table filling in some of the columns.

6.3.4 Plotting miscellaneous information

There are a few other tasks which create plot files, but which do not fit into the categories above. The most general of these is PLOTR which can plot up to 30 sets of *(x,y)* points input from a text file with coloring options. In 31DEC25. PLOTR can also read and plot a column of error bars if desired. Task PCHIS also reads a text file to produce a histogram plot of the values in any selected column of the file. CONPL is a task which plots AIPS convolving functions (used by various *uv*-data gridding tasks such as IMAGR) and their Fourier transform, expected signal-to-noise ratio, or convolution with a user-specified Gaussian. IRING integrates an image in concentric annuli about the user-specified object center with specified major axis position angle and inclination. The results may be placed in a plot file for later display. GAL calculates the orientation and rotation curve parameters of a galaxy from an image of the predominant velocities. The observed rotation curve is plotted together with the fitted model curve. LOCIT fits antenna location corrections to SN tables; the residual phases may be plotted as a function of sample number.

There are several tasks which create RGB cubes for later display by tasks such as GREYS and KNTR. These include RGBMP which does a weighted sum of the planes of a data cube, HUINT which uses two images as hue and intensity to construct an RGB cube, and TVHUI which interactively uses three images as intensity, hue, and saturation to construct an RGB cube by a different algorithm. Task SCLIM will scale and clip image planes to be used as inputs to LAYER, which produces an RGB cube from the colored sum of up to 10 input image planes using a complicated and general algorithm.

6.4 Interactive TV displays of your data

The AIPS TV display allows you to look at your image data in detail and to set parameters by pointing at interesting features visible on the screen. Although AIPS will run on a variety of hardware display devices (*e.g.*, I^2S Models 70, 75 and IVAS and DeAnza), it is now used almost exclusively with the X-Windows TV-simulation program called `XAS`. See § 2.3.2 for information on using this basic AIPS tool. If the `XAS` window has closed for any reason, you may restart it without exiting `AIPS` with the verb `STARTTV` or `TVRESTOR`. Information on TV verbs and tasks may be found by entering ABOUT TV C_R, ABOUT INTERACT C_R, and ABOUT TV-APPL C_R, or by consulting the corresponding sections of Chapter 13 of this *CookBook*.

6.4.1 Loading an image to the TV

The simplest way to load an image from your catalog to the TV and then to manipulate the display is with the procedure called `TVALL`:

> INP TVALL C_R	to review the input parameters.
> INDI *n* ; GETN *ctn* C_R	to select the disk and image name parameters from the catalog.

Use one of the following commands to specify the initial transfer function that converts your image file intensities to display pixel intensities:

> FUNC 'LN' C_R	linear—this is the default.
> FUNC 'LG' C_R	logarithmic.
> FUNC 'L2' C_R	"extra" logarithmic.
> FUNC 'SQ' C_R	square-root, often a good compromise
> FUNC 'NE' C_R	negative linear.
> FUNC 'NG' C_R	inverse logarithmic.
> FUNC 'N2' C_R	inverse "extra" logarithmic.
> FUNC 'NQ' C_R	negative square root.
> PIXRA *x1*, *x2* C_R	to load only image intensities *x1* to *x2* in the units of the image. The default is to load the full range of intensities in the image.

(The slopes and intercepts of the display transfer functions can be modified later, but the above options let you choose initially between linear, square-root, logarithmic, and negative displays and restrict the range of intensities that is loaded). Then:

> TVALL C_R	to load the selected image.

The image should appear on the TV screen in black-and-white. If you see a new image but it is not what you expected, hit button `D` to end `TVALL` and then review your inputs with:

> INP TVALL C_R

If no image appears, make sure that `AIPS` started up with your workstation assigned as the display; see § 2.2.3 for information on starting up `AIPS` and assigning the TV display.

After your image has been displayed on the TV by `TVALL`, your trackball with its buttons (which should be labeled `A`, `B`, `C`, and `D`) — or your workstation mouse with keyboard buttons `A`, `B`, `C`, and `D`) — can be used to modify the display transfer functions, coloring and zoom. Pressing button `A` alone enables black-and-white and color-contour coding of the image intensities, successively. Adjust the cursor position on the TV (using the trackball or mouse) to vary the slope and intercept of the display transfer function. `TVALL` will superpose a calibrated horizontal wedge on the image. This should help you to choose the optimum cursor setting for the display. Black-and-white displays are generally much more suitable than color for high-dynamic-range images, while color contouring may be used to accentuate interesting features. (Note also that a much wider range of image-coloring options is available outside `TVALL` by invoking `TVPSEUDO`, `TVPHLAME`, and `TVHELIX`.)

Pressing buttons B and C adjusts the zoom of the display: B to increase the magnification and C to decrease it. When these buttons are enabled, the cursor controls the position of the center of the zoomed field of view. Magnification factors of 1 through 16 are available on most workstations. Note that your terminal issues instructions when buttons are pressed, but that it is in the death-like grip of TVALL otherwise until you press button D to exit from it.

The size of the image which can be displayed depends on the size (in pixels) of your workstation screen or TV display and some other parameters which can vary from site to site. A typical Sun workstation can display up to 1024 by 900 pixels. Then, if the image is larger than about 800 pixels or so in the *y*-direction, portions of the labeling of the wedge (the units) will be omitted or superposed on top of the wedge (the tick numeric values). A useful technique for displaying large images is to load only alternate pixels. The command:

> TXINC 2 ; TYINC 2 C_R	to load every other x and y pixel.

before TVALL would do this. Also use:

> TBLC = *n1*, *n2*, *n3*, . . . C_R	bottom left pixel to load.
> TTRC = *m1*, *m2*, *n3*, . . . C_R	top right pixel to load.

to limit the displayed field. A small image may be interpolated to fill the TV screen by setting TXINC = -1 ; TYINC = -1 C_R. Recent versions of XAS allow the verb TVROAM to function again. This verb loads adjacent portions of an image to as many as 16 memories and then "roams" with a split screen to allow you to view any contiguous portion of the loaded image. REROAM resumes roaming an already loaded roam-mode image, perhaps after adjustment of the transfer function of all channels with TVTRANSF and TVCHAN = 0. The verb ROAMOFF allows you to convert the final split-screen image into a normal, more useful single-plane image.

TVALL is a procedure that insures that the desired graphics and TV channels are on and cleared and the others off, then loads the image with verb TVLOD, loads a wedge with TVWEDGE, and labels the wedge with TVWLABEL. You do not have to use TVALL, which can be rather slow, simply to load a new image on the TV. Use TVLOD instead. TVWEDGE has a variety of options concerning the width of the wedge and its position and direction; type HELP TVWED C_R for details.

The task TVHLD was completely revised to load images to the TV with histogram equalization. The task has interactive controls over the intensity range and method of computing the histogram and can write out a histogram-equalized image in arbitrary, non-physical units for display by the various plotting tasks.

6.4.2 Manipulating the TV display

There are a number of verbs which allow you to manipulate the display, including:

> STARTTV C_R	to start the TV, message, and Tektronix servers when one or more of them have exited.
> TVINIT C_R	to initialize the entire TV and TV image catalog. The image catalog is now kept in XAS and so is always current.
> TVON n C_R	to turn on TV grey channel(s) with n being a bit pattern of the desired channels: $2^{(i-1)}$. Thus TVON 4 turns on channel 3. You may have to turn off other channels to see the desired channel, since the sum of two images may be rather odd.
> TVOFF m C_R	to turn off channel m, where m is a bit pattern as with TVON.
> GRON n C_R	to turn on one or more of the 8 graphics channels, where n is a bit pattern.
> GROFF m C_R	to turn off one or more graphics channels.
> TVCHAN n ; TVCLEAR C_R	to zero a TV channel.

> GRCHAN *m* ; GRCLEAR C_R	to zero a graphics channel.
> TVZOOM C_R	to set the zoom magnification and center interactively, follow instructions on the screen.
> OFFZOOM C_R	to reset the zoom and zoom center to null.
> GREAD C_R	to read the current color of a specified graphics overlay channel into `RGBCOLOR`.
> GWRITE C_R	to change the color of a graphics overlay channel to that specified by `RGBCOLOR`. This may be done for aesthetic reasons or because the default colors may not show up well when captured by `TVCPS` and printed on a color printer.
> TVPOS C_R	to read the TV cursor position, returning adverbs for use in procedures or other verbs.
> IM2TV C_R	to convert an image pixel in `PIXXY` to the corresponding TV pixel.
> TVDATA C_R	to display lots of information about your current TV display.
> TVFACTS C_R	to display lots of information about your current TV display.

6.4.3 Intensity and color transfer functions

The *AIPS* model of a TV postulates two intensity transfer functions, called the LUT and the OFM, which are basically multiplicative. In most circumstances, the LUT is used for black-and-white enhancements and the OFM for coloring, but both can be used for either. To manipulate the LUT interactively, while leaving the pseudo-coloring alone, use the `TVTRANSF` verb. The cursor position controls the slope and intercept of the transfer function, buttons `A` and `B` switch a plot of the transfer function on and off, button `C` switches the sign of the slope, and button `D` (as always) exits. In `31DEC21`, verb `TVBANDW` implements 3 types of black-and-white modifications of the LUT: a series of step wedges, sine waves, or stairs in intensity. To turn the LUT back to normal, enter `OFFTRAN`.

A rich zoo of color coding is available with `TVPSEUDO`, which alters the OFM while leaving the LUT alone. Repeated hits on button `A` select a variety of color triangles, button `B` selects a circle on hue, and repeated hits on button `C` select a variety of color contours. First-time users should experiment with the *AIPS* coloring options until they develop an intuitive feel for the effects of cursor settings on the image appearance. The wedge displayed by `TVALL` adjusts to the alternative colorations selected with `TVPSEUDO`, and it is helpful to watch changes in both the wedge and the image. A flame-like coloring is available with `TVPHLAME`, or variations on the scheme with repeated hits on buttons `A` or `B`. `TVHELIX` does a helix in color space attempting to make a monotonic increase in perceived intensity. The buttons change direction, number of rotations, and saturation. For images that have significant negative and positive values, `TV2COLOR` uses two color wedges increasing away from the user-adjusted central value. In these verbs the cursor position controls aspects of the coloring such as enhancements, richness, or cycles of hue. To turn off pseudo-coloring, enter `OFFPSEUD`.

A set of less well-known verbs is available to allow you to create, manipulate, and save desirable versions of the OFM table. `OFMSAVE` allows you to save a named OFM, `OFMDIR` lists all saved OFMs belonging to you or generally available from the *AIPS* distribution, `OFMGET` loads a named OFM, `OFMZAP` deletes a named OFM, and `OFMLIST` prints the current OFM. `OFMCONT` is an elaborate interactive verb which allows you to set the hue, intensity, and saturation of the OFM divided up into a number of color contours. Each of these contours can be a constant level or a step wedge. `OFMADJUS` is another elaborate interactive verb to alter pieces of the OFM, while `OFMTWEAK` is a simpler verb to stretch the OFM.

The OFM received particular attention in `31DEC25`. A significant number of OFM tables described on the Internet were adapted and added to the *AIPS* distribution. A new pseudo-coloring verb `OFMCOLOR` was

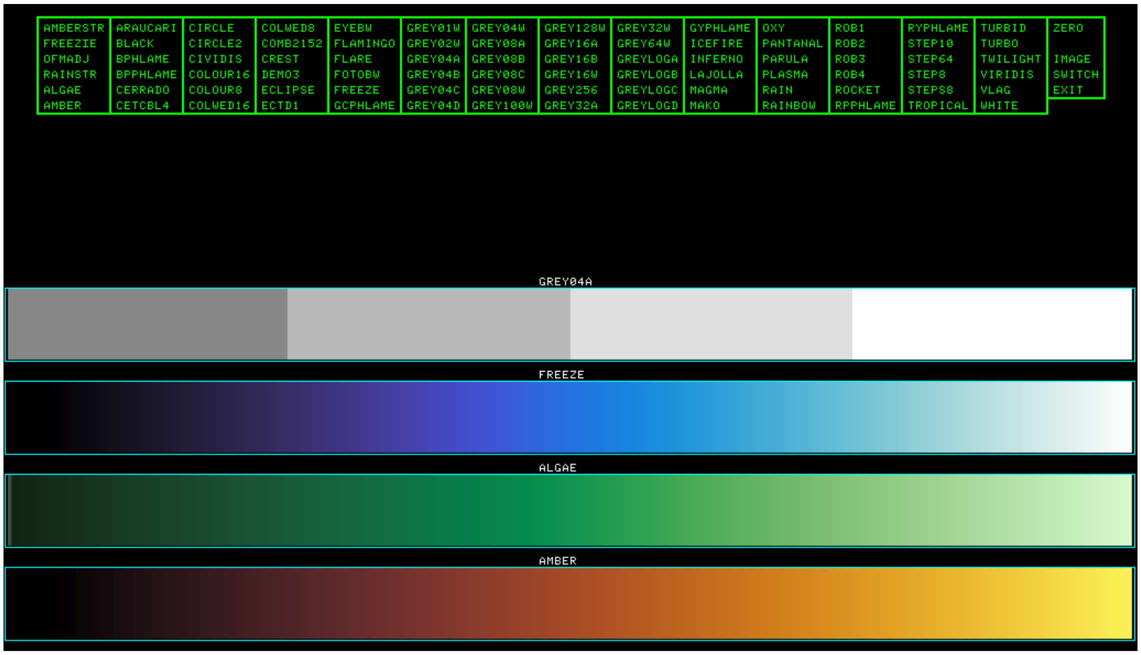


Figure 6.7: Example step wedge display from OFMPL

added to AIPS and to the menus of multiple image-display tasks such as TVSPC and XGAUS. This function begins by loading one of the new OFMs. You may then select different OFMs by hitting buttons A, B, or C. Each button selects from a list of 10 of the new OFMs. Having selected an OFM, you may then shift the center of the function using the horizontal position of the cursor. Another new verb OFMSTRCH does this stetching of the current OFM outside of OFMCOLOR. To view the available OFMs distributed with AIPS plus any that you may have developed, the new task OFMPL was developed. It has three displays: up to 4 step wedges, a plot of the red, green, and blue parts of the OFM, and an optional image colored by the OFM. All three of these show a menu at the top of the TV listing all available OFMs plus items to select the desired display. See an example in Figure 6.4.3.

6.4.4 Setting parameters with the TV

One reason to load the image to the TV is to set adverbs for use by other verbs and tasks. Verbs which use the TV cursor to set adverbs include:

> TVNAME C_R	to set INDISK, INNAME, etc. to the name parameters of the image currently visible. If there is an ambiguity, you will be asked to move the cursor to the desired image and press a button.
> MFITSET C_R	allows you to set the adverbs for IMFIT and JMFIT, namely the image name parameters, window, and initial guesses for the Gaussians from the image on the TV.
> SETXWIN (*dx*,*dy*) C_R	reads pixel coordinate of the center of a *dx*-pixel by *dy*-pixel window and sets the adverbs BLC and TRC.

> TVWIN C_R — reads pixel coordinates from the next two cursor positions at which a trackball button is depressed. The TV graphics shows the current shape and position of the window. Button A allows you to switch to (re)setting the other corner while the other buttons exit after both corners have been set. TVWIN uses the pixel coordinates to set up the bottom left (BLC) and top right (TRC) corners of an image subsection, *e.g.*, for input to the contouring programs CNTR and PCNTR, to the mean/rms calculator IMEAN, and to many other tasks.

> SETSLICE C_R — works like TVWIN above to set BLC and TRC. Instead of a rectangle however, the display shows a diagonal line which is useful for setting the ends of slices.

> TVBOX C_R — is similar to TVWIN above except that it is used to set up pixel coordinates to define rectangular *or circular* Cleaning areas for the *AIPS* Clean tasks. The adverbs NBOXES and CLBOX are set. The circular option appeared in the 15JUL95 release and is not supported by APCLN.

> REBOX C_R — allows revision using the TV of the Cleaning areas set previously with TVBOX. Revises NBOXES too.

> DELBOX C_R — allows deletion using the TV of the Cleaning areas set previously with TVBOX. Revises NBOXES too.

> FILEBOX C_R — is REBOX for boxes in the text file used with IMAGR; multiple fields and many more boxes are allowed.

> DFILEBOX C_R — is DELBOX for boxes in the text file used with IMAGR; multiple fields and many more boxes are allowed.

Task FILIT performs FILEBOX and much more on a set of images, usually the multiple facets of a wide-field imaging. It offers a wide range of image display options including handling images larger than the TV, allows editing of the box information, creates boxes via the auto-boxing algorithm, and checks the multiple facets for overlapping boxes. This new task may be the quickest and easiest way to review a set of facet images.

6.4.5 Reading image values from the TV

There are several facilities for reading out intensity and position information from displayed images using the TV cursor:

> IMPOS C_R — displays the two coordinate values (*e.g.*, RA and Dec) from the cursor position when any button is depressed. Adverbs TVBUT and COORDINA are returned.

> IMXY ; IMVAL C_R — displays the image intensity and the two coordinate values (*e.g.*, RA and Dec) from the cursor position when any button is depressed. Adverbs PIXXY, TVBUT, PIXVAL, and COORDINA are set.

> CURVALUE C_R — continuously displays (in the upper-left corner of the TV) the pixel coordinates and the image intensity in user-recognizable units at the position selected by the TV cursor.

> TVFLUX C_R — displays image intensities and coordinates whenever a TV button is pressed, looping until button D is pressed. Adverbs for the first image name are set as well as PIXXY, TVBUT, PIXVAL, and COORDINA for the last pixel selected.

> TVDIST CR	displays the angular length and position angle of the spherical vector between two pixels in one or two images shown on the TV. Name adverbs for input files 1 and 2 are set as well as adverbs `PIXXY`, `PIX2XY` and `DIST`.
> TVMAXFIT CR	whenever a TV button is pressed, fits a quadratic function to the image to find the position and strength of an extremum, looping until button D is pressed. Adverbs for the first image name are set as well as `PIXXY`, `TVBUT`, `PIXVAL`, and `COORDINA` for the last object selected.
> TVSTAT CR	determines the mean, rms, extrema and integrated intensity (if appropriate) in user-defined "blotch" regions within the image currently displayed on the TV. The regions are irregular polygons selected with the TV cursor. Type EXPLAIN TVSTAT CR for details. Adverbs `PIXAVG`, `PIXSTD`, `PIXVAL`, `PIXXY`, `PIX2VAL`, and `PIX2XY` are set.
> COPIXEL CR	to convert between pixel and astronomical coordinates for an image; allows determination that a coordinate is not inside an image.

6.4.6 Labeling images on the TV

There are a number of facilities for labeling images on the TV including:

> CHARMULT /CR	to scale the character size used by the TV by 1 through 5 times the standard size which is 7 by 9 pixels. Many menu-driven interactive tasks include this option when run on larger TV displays.
> TVLABEL CR	to draw standard axis labels around the visible image. You may control the type of labeling and whether coordinates are shown as a grid or short tick marks. A Clean beam may be plotted. If more than one image is visible, you will be asked to indicate which one you want with the cursor and any button.
> TVWLABEL CR	to draw axis labels around the visible intensity wedge.
> TVANOT CR	to draw a text string into a grey-scale channel or a graphics plane at a location specified via an adverb or via the TV cursor.
> TVLINE CR	to draw a straight line into a grey-scale channel or a graphics plane at locations specified in part or in whole via adverbs or via the TV cursor.
> TVILINE CR	to draw a straight line into a grey-scale channel or a graphics plane between two image pixel coordinates.
> COSTAR CR	to plot a "star" positions at a user-specified coordinate on the TV image.
> TVSTAR CR	to plot "star" positions from an `ST` file on top of the visible image; see § 6.3.2.

6.4.7 Comparing images on the TV

It is often useful to compare two images, *e.g.*, to decide whether one contains artifacts that are not present in another at the same frequency, or to look for frequency-dependent features at constant resolution. AIPS provides several tools for such image comparisons.

The first tool is a capability for loading multiple images to the same plane (or channel) of the TV device. The parameter `TVCORN` specifies where the bottom left corner of the image or image subsection will be positioned in the TV frame by `TVLOD` or `TVALL`. If `TVCORN` is left at zero, `TVLOD` and `TVALL` adjust it to center the displayed image. You may however use `TVCORN` to control loading successive images to different regions of the display with successive executions of `TVLOD`. For example, the following commands would load two 512 by 512 pixel images from slots 1 and 2 on disk 1 *side-by-side* on channel 1 of a 1024 by 900 TV display:

> INDI 1; GETN 1 C_R	to select the input disk and the first image.
> TVCH 1 C_R	to select TV channel 1 for the loading.
> TVCORN 1 193 ; TVLOD C_R	to load the first image.
> GETN 2 C_R	to select the second image.
> TVCORN 513 193; TVLOD C_R	to load the second beside the first.

You could then adjust the color coding, transfer function, etc. for both images simultaneously with `TVFIDDLE` or `TVTRAN`. You may load as many as 256 images to a single TV plane with this technique, which is therefore a powerful method for making "montages." The number of simultaneous images is limited mostly by your image sizes, the need to avoid overlaps (which are allowed if you want) — and your ability to do the arithmetic for appropriate `TVCORN` settings! You are also limited by the need for all the images in one plane to share that plane's transfer function. Judicious use of the `PIXRANGE` and `FUNC` inputs to `TVLOD` permits making useful montages of disparate images, however.

A second tool is the classic "blink" technique from optical astronomy. `TVBLINK` allows you to load images to two different planes of the TV memory and then to alternate the display rapidly between the two. The two images described above could be "blinked" against each other by the following command sequence:

> INDI 1; GETN 1 C_R	to select the input disk and first image.
> TVINIT; TVCORN 0	to clear the TV and restore the default positioning.
> TVCH 1; TVLOD C_R	to load the first image on plane 1.
> GETN 2 C_R	to select the second image.
> TVCH 2 ; TVLOD C_R	to load the second image on plane 2.
> TV2CH 1 ; TVBLINK C_R	to blink planes 1 and 2.

The rate and duty cycle of the blinking, and the transfer functions applied to the planes, are controlled interactively with the TV cursor. Instructions for these operations appear on your terminal while `TVBLINK` is running.

The task `PLAYR` provides a menu-driven method to enhance and blink two images and to develop and save TV color tables ("OFMs").

For data cubes (*e.g.*, frequency or time sequences of images), the verbs `TVMOVIE` and `TVCUBE` combine the two previous techniques. These are described in more detail in § 8.5.4. Both verbs load one or more image planes with as many planes from the cube as possible (and as requested). Then they display each frame in sequence with interactive controls over the frame rate in movie mode, the chosen frame in single-frame mode, and the brightness, contrast, and color of the displayed images. `TVMOVIE` makes a somewhat more efficient movie sequence, but `TVCUBE` makes a better montage by using a more normal arrangement of the image planes. The `DOALL` adverb allows these verbs to loop over more than just the third axis if the "cube" has more than three multi-pixel dimensions.

Certain real TV displays used to provide powerful tools to compare images using color as well as intensity to represent real information. `XAS` implements some of these tools on workstations capable of displaying a virtually unlimited number of colors. Perhaps the most powerful of these is the "hue-intensity" display in which one image controls the intensity of the displayed image and the other controls the hue. A number of uses for this algorithm are obvious, including spectral-line moment images (intensity from total line integral and velocity setting color), polarization images (intensity from the total or polarized intensity image and polarization angle setting color in a circular scheme), and depolarization observations (color set by a two-frequency depolarization image). The best implementation of this algorithm has been in the interactive verb TVHUEINT:

> INDI *d1*; GETN *ctn1* C_R	to select the intensity image.
> TVINIT; TVCORN 0	to clear the TV and restore the default positioning.
> TVCH 1; TVLOD C_R	to load the first image on plane 1.
> INDI *d2*; GETN *ctn2* C_R	to select the hue image.
> TVCH 2 ; TVLOD C_R	to load the second image on plane 2.
> TVCH 1 ; TV2CH 2 ; TVHUEINT C_R	to display a full-color view where the intensity is controlled by image 1 and the hue by image 2 and to interactively adjust that display.

Instructions for altering transfer functions and reversing the roles of the two images appear on your terminal while TVHUEINT is running. You may create a labeled two-direction step wedge to accompany the hue-intensity display with the verb HUEWEDGE.

Until the 31DEC14 release, the only way to capture this image was with TVCPS, which is fundamentally limited by the size and resolution of the TV display. Control over labeling, character font, number of pixels, and the like is severely limited. To overcome these limitations, a new task called HUINT was written. It performs the functions of TVHUEINT with the option of saving the resulting image and a two-dimensional step wedge as 𝒜ℐ𝒫𝒮 cataloged images. The task enters an interactive mode in which the user selects options from a menu shown on a graphics overlay plane. Options are to enhance each of the images individually, to select linear, square-root, or one of 2 logarithmic transfer functions for intensity or hue, and to exit with or without final updates and an output file. The output image(s) can then be made into plot files with standard plot tasks like GREYS and KNTR and then converted into PostScript by LWPLA. HELP POSTSCRIPT contains some guidance on how to combine PostScript renderings of the color image and the color step wedge into a single file. Note that the images saved by HUINT are not in any recognizable physical units. Also note that KNTR should be run with DOCOLOR=1; DOWEDGE=0 FUNCTYPE=' ' and LWPLA should be run with RGBGAMMA=1 and no extra enhancements.

The two functions above require a display that can do an almost unlimited number of colors. Unfortunately, some older workstations can display only 256 simultaneous colors (or even fewer). XAS supports both kinds of workstation. For these limited workstations, there are two tasks, also discussed in § 8.6, which attempt to recover much of this capability by trying to optimize color assignments over the limited range available. The first of these, TVHUI produces a composite display in which the intensity is set by one image, the hue is derived from another image, and the saturation is optionally derived from a third image. An interactive menu allows you to enhance each of the images individually, to select linear, square-root, or logarithmic transfer functions for the intensity and hue images, to select the sub-image used during interactive enhancements, to repaint the full image, and to exit with or without writing out the final three-color image. The task has the option of writing an image of a hue-intensity step-wedge as well. A number of uses for the saturation portion of this task are obvious, including spectral-line moment images (line width setting saturation) and polarization images (polarization intensity setting saturation). TVHUI is also useful on full-color displays, but for such displays, you may do much of the work of this task in HUINT. TVHUI attempts to write out the full range of input intensities scaled by the selected transfer functions. Thus, its output images are different from those of HUINT and may, or may not, be preferable in particular cases.

There is also an 𝒜ℐ𝒫𝒮 task called RGBMP which writes three-color cubes using weighted sums over a data cube. Three-color cubes also arise when digitizing color photographs of real scenes. The second task, TVRGB, can be used to display these three-color cubes or to generate a three-color display from any three 𝒜ℐ𝒫𝒮 image planes. Common examples of the latter are the superposition of radio continuum and/or line data on optical or X-ray images, and color-coding of effective temperatures or spectral indices from 3-channel continuum data. TVRGB can also be used to color-code different types of depolarization effects from multi-frequency polarimetry. Like TVHUI, TVRGB offers a simple menu to enhance each of the images individually or all together, to select the window specifying the sub-image which is used during interactive enhancements, to repaint the full image on the TV, and to exit. TVRGB does not write an output image per se, but it can be instructed to write out a full 24-bit color PostScript plot file to be sent to a color printer. Its display (or any other TV display including that of TVHUI) can be captured and sent to a color printer; see § 6.4.10 below.

On full-color workstations, three-color images may be displayed by loading each color plane to a separate TV memory. Then each memory is turned on in the desired color only using TVON with the usually ignored COLORS adverb. If the red image is in TV channel 1, the green in 2 and the blue in 3, the verb TV3COLOR is a short-cut for all the parameter setting.

> FOR TVCH=1:3; TBLC(3)=TVCH; TVLOD; END; TV3COLOR C_R

In 31DEC25, the procedure TV3LOD was added to make this type of display even more convenient.

6.4.8 Slice files and the TV display

In Chapter 8 we discuss the computation and use of "slices," one-dimensional profiles interpolated along any line in an image plane. Once a slice has been computed, it may be plotted by TVSLICE on the TV display in your choice of graphics channel. A second slice may be plotted on top of the first with TVASLICE. The TV graphics display is used to prepare initial guesses for SLFIT, which fits Gaussians to slices. The verbs involved are:

> NGAUS *n* ; TVSET C_R	to set the number of Gaussians to be fitted to *n* and then to prepare an initial guess at the parameters by pointing at the peaks and half width points on a graphics plot of the slice.
> TV1SET *j* C_R	to revise the initial guess for the j^{th} Gaussian.
> TVGUESS C_R	to plot the initial guess of the model on the TV graphics, erasing any previous plot.
> TVAGUESS C_R	to add a plot of the initial guess of the model to the current slice plot on the TV graphics.
> TVMODEL C_R	to plot the fit model on the graphics device, erasing any previous plot.
> TVAMODEL C_R	to add a plot of the fit model to the current slice plot on the TV graphics.
> TVRESID C_R	to plot the data minus the fit model on the TV graphics, erasing any previous plot.
> TVARESID C_R	to add a plot of the residuals (data minus model) to the current slice plot on the TV graphics.
> TVCOMPS C_R	to plot the individual components including baseline of the fit model on the TV graphics, erasing any previous plot.
> TVACOMPS C_R	to add a plot of the individual components including baseline of the fit model to the current slice plot on the TV graphics.

The units for slice model parameters are those of the plot, so it is convenient to set them with these verbs. These same operations may also be done on the Tek graphics device (§ 6.5.2), but modern X-Windows emulations of such devices seem to have problems with cursor reading. The TV verbs allow multiple colors for plot comparisons using different GRCHANs.

6.4.9 Other functions using the TV

There are a number of tasks which use the TV to give the user real interactive input to the operation based on the images displayed by the task on the TV. These include BLANK (§ 8.6) to blank out non-signal portions of an image, BLSUM (§ 8.6) to sum images over irregular blotch regions plotting or printing out summed spectra (and saving SLice files), TVFLG (§ O.1.6) to edit visibility data based on grey-scale displays of some function of the visibility with baseline on the x axis and time on the y axis, SPFLG (§ 8.1, § 10.2.2) to edit visibility data based on grey-scale displays of some function of the visibility with spectral channel for all IFs on the x axis and time on the y axis, EDITR (§ 5.5.2) to edit visibility data based on plots of visibility versus time, 1–11 baselines at a time, EDITA (§ 4.3.11) to edit visibility data based on plots of system temperature (TY or SY tables) or antenna gains (SN or CL tables), WIPER to edit visibility data using UVPLT-like displays, and SNEDT to edit TY, SY, SN, and CL tables themselves. TV-based tasks have appeared to edit visibility data displayed on a $u-v$ grid (UFLAG), to edit visibility or bandpass data based on graphical displays of bandpass solutions (BPEDT), and two tasks to edit pulse-cal tables (PCFLG, PCEDT).

The imaging tasks, IMAGR, SCIMG, and SCMAP display the results of the computation at its current stage on the TV and provide a menu of interactive options to the user. The menu includes the usual display enhancements, the ability to choose among the images being computed, the ability to set Clean windows in those images, and the ability to end the computation at its current stage. The computation will resume when instructed via the menu or after a period of inactivity. A number of older iterative tasks use the TV to display the results of the computation so far and then prompt the user to hit button D within some number of seconds to stop the computation. Tasks that do this include APCLN, SDCLN, VTESS, and several less significant tasks. UVMAP uses the TV simply to draw a picture indicating which cells are sampled in the *uv* plane. All of these tasks are described in Chapter 5.

The multi-source fitting task SAD was provided with a companion task TVSAD which allows interactive setting of the initial guess parameters for each island being fit. It allows re-trying the fits and adjusting the island boundaries, and can be run non-interactively until the fit finds a problem, after which the interactive mode resumes. AIPS Memo 119[3] describes the operation of this task in detail.

There is a set of related tasks for analysis of data cubes transposed so that the first axis is the one on which baselines or Gaussians are to be fit. In the case of a spectral-line cube, the image would be transposed so that velocity is the first axis. TVSPC is an interactive task which displays spectra from one or two transposed cubes at pixels chosen on the TV from some image of the overall field. A third cube may also be displayed with the spectral plane chosen by the cursor on the displayed spectra. See the associated AIPS Memo[4]. There is now an additional task TVIEW which does an interactive display of an image cube in all three tranpostions; it also has an AIPS Memo[5]. One can also use XPLOT (with DOTV 1) first to get an idea of what the profiles really look like. It uses a flux cutoff to determine which profiles to display and prompts you for permission to continue after each plot. XBASL is used to remove $n^{\rm th}$-order polynomial baselines from each spectrum. It has a batch mode of operation and an interactive mode which uses the graphics display to plot each spectrum and to accept guidance on which channels to use in determining the baselines.

XGAUS fits up to 8 Gaussians plus a linear baseline to each spectral profile. It builds a table of solutions at each pixel and may be re-started until you are satisfied, after which images of the fit Gaussian parameters and their uncertainties may be written to disk. In its interactive mode, it plots each selected spectrum on the TV graphics planes and accepts guidance on the initial guesses for the Gaussians. After all pixels have been done, it displays images of the fit parameters and offers a variety of options for editing and correcting the results. Tasks similar to XGAUS have been written to fit polarization spectra (RMFIT) and to fit for Zeeman splitting (ZEMAN) of the line viewed simply or as a set of Gaussians found by XGAUS. Plot tasks XG2PL and

[3]Greisen, E. W. 2014, "TVSAD: interactive search and destroy" AIPS Memo 119, https://www.aips.nrao.edu/aipsdoc.html
[4]Greisen, E. W. 2025, "Exploring Image Cubes in AIPS," AIPS Memo 120 revised, https://www.aips.nrao.edu/aipsdoc.html
[5]Greisen, E. W. 2025, "Further Exploration of Image Cubes in AIPS," AIPS Memo 124 revised, https://www.aips.nrao.edu/aipsdoc.html

RM2PL have been written to display these spectral fits in standard AIPS plot files. See AIPS Memo 118.[6] Corresponding tasks AGAUS and ZAMAN are available to perform similar analysis of absorption-line image cubes. See AIPS Memo 122.[7] In 31DEC24, a similar task called TVSPX has appeared to assist in fitting spectral indexes to image cubes. In 31DEC25, another similar task LISPX fits spectral indices to spectra in text files. See AIPS Memo 126.[8]

6.4.10 Capturing the TV

Having done all the work to prepare the absolutely perfect display on your TV screen, it would be a good idea to capture it before someone, such as the local power company, does a TVINIT. See § Z.1.2.2 for a discussion of Unix tools to do this. We recommend, however, TVCPS to capture the image on your TV, including graphics overlay channels if they are on, and to write the result to an encapsulated PostScript file. This file can be printed immediately on a black-and-white or color printer or on any other device which understands PostScript. It can also be saved for later printing or inclusion in other documents. TVCPS was used to make the picture on the title page of this CookBook, the picture of TVFLG's display in Chapter 4, and the picture of a right ascension - velocity - declination data cube in Chapter 9. TVCPS bases its picture on the current size of your TV display. If you are using a workstation with XAS, be sure to adjust the size of the display window to encompass all of your image plus a modest border. If you leave a large border, you will get a large border in your output. TVCPS understands both pseudo- and full-color XAS displays. Your TVCPS session could look like:

> TASK 'TVCPS' ; INP C_R	to review the inputs.
> OUTFILE 'MYAREA:TV.PIX C_R	to save the output in a file called TV.PIX in an area defined by the logical MYAREA; see § 3.10.1.
> OPCODE 'COL' C_R	to make a color picture.
> APARM 8.5 , 11 C_R	to set the output device size to 8.5 by 11 inches, appropriate to standard quarto paper.
> GO C_R	to run the task when the inputs are set.

TVCPS has an option to add a character string below the image with adverb REASON. If your image is too large to fit on the TV, you can instruct TVCPS to read the image from disk with DOTV = -2 C_R, using adverbs TBLC, TTRC, TXINC, and TYINC. When doing this, you should turn off the graphics display (GROFF 32767 C_R) since it is not aligned properly. This option works when multiple images are visible at the same time, but only when they are in separate TV planes as fully overlapped 3-color images. Some color printers and recorders have rather different transfer characteristics than the workstation screen. TVCPS offers the option to remove the "gamma correction" used for your workstation and to apply a different one appropriate to your color recorder. TVCPS now offers the option to represent colors using the CMYK (cyan-magenta-yellow-black) used in printing rather than the familiar RGB colors system. CMYK displays often require different gamma corrections from those used for RGB. To see the effects of the gamma correction, try the verb GAMMASET in AIPS. You may need to use GWRITE to select better colors for the graphics overlay planes as well. The default black background may be set to and color with adverb RGBCOLOR.

§ Z.1.2.2 contains a discussion of Unix tools to capture the TV screen which work even when the AIPS display task has control of the display. The import program is perhaps the more useful of these.

[6] Greisen, E. W. 2025, "Modeling Spectral Cubes in AIPS," AIPS Memo 118 revised, https://www.aips.nrao.edu/aipsdoc.html
[7] Greisen, E. W. 2025, "Modeling Absorption-line Cubes in AIPS," AIPS Memo 122 revised, https://www.aips.nrao.edu/aipsdoc.html
[8] Greisen, E. W. 2025, "Spectral index fitting in AIPS," AIPS Memo 126, https://www.aips.nrao.edu/aipsdoc.html

6.5 Graphics displays of your data

In the dim dark past, Tektronix invented some nice graphical display devices and an inconvenient but functional way to talk to them. This communication language became so embedded in software that workstation vendors now provide X-Windows windows that understand it. AIPS also arose from this dim dark past and once upon a time talked to those lovely green screens. To retain the graphics capability, we now provide a TEKSRV server which will provide a Tektronix-like graphics screen on which certain AIPS tasks and verbs plot. When AIPS starts up on workstations, it brings up a window called TEKSRV. Leave this window in its iconic state; it is only a marker for the presence of the server. The first time you write to TEKSRV, it will create and open a window called TEKSRV (Tek) in which the plot is done. You can re-size this window within some limits and the plot will automatically re-size itself. When the workstation cursor is in the Tek window, it changes to a diagonal arrow pointer. When an AIPS task or verb tries to read from the Tek window, this pointer becomes a plus sign. You should position the pointer to the desired location *without* touching the keyboard or the mouse buttons. When the pointer is exactly where you want it, press any mouse button or any key (*except* RETURN) to return the pointer position to the program. Note that the functions using the graphics display are not quite as friendly as those on the TV. This is due to the inability to erase a piece of a plot without erasing all of it. You can erase the full screen with TKERASE, which will keep the window from redrawing a big plot on every expose event. In recent years, we have found unfortunately that cursor reading from Tektronix emulation screens can be unreliable. Therefore, every TK function described below also has a TV version, described in a previous subsection. The TV has many other advantages over the Tektronix emulator and now is nearly as fast. The verb STARTTV will restart this server if it has, for some reason, exited.

6.5.1 Plotting data and setting values with the graphics display

TKPL interprets AIPS plot files to the graphics window or device. The graphics screen can be used to read back data values and set adverbs, much like the TV:

> TKXY C_R	to read the graphics cursor position, setting adverb PIXXY; requires a contour or comparable image to be shown on the screen.
> TKXY ; IMVAL C_R	to read the graphics cursor position and return the image value and coordinates of the selected position.
> TKPOS C_R	to read the graphics cursor position and return the image coordinates of the selected position.
> TKWIN C_R	to read the graphics cursor position twice, first setting BLC and then setting TRC.
> TKBOX(*i*) C_R	to read the graphics cursor position twice, first setting the lower left and then the upper right of Clean box *i*.
> TKNBOXS(*n*) C_R	to set NBOXES to *n* and then set all *n* Clean boxes using the graphics cursor.

All of these verbs require a graphics display of a plot file produced by CNTR, PCNTR, GREYS, or SL2PL. The window procedures don't make much sense with a slice plot, but they will work.

6.5.2 Slice files and the graphics display

In Chapter 8 we discuss the computation and use of "slices," one-dimensional profiles interpolated along any line in an image plane. Once a slice has been computed, it may be plotted by `TKSLICE` on the graphics display. A second slice may be plotted on top of the first with `TKASLICE`. The graphics display is used to prepare initial guesses for `SLFIT`, which fits Gaussians to slices. The verbs involved are:

> TKVAL C_R	to return the flux level pointed at by the graphics cursor.
> NGAUS *n* ; TKSET C_R	to set the number of Gaussians to be fitted to *n* and then to prepare an initial guess at the parameters by pointing at the peaks and half width points on a graphics plot of the slice.
> TK1SET *j* C_R	to revise the initial guess for the $j^{\rm th}$ Gaussian.
> TKGUESS C_R	to plot the initial guess of the model on the graphics device, erasing any previous plot.
> TKAGUESS C_R	to add a plot of the initial guess of the model to the current slice plot on the graphics device.
> TKMODEL C_R	to plot the fit model on the graphics device, erasing any previous plot.
> TKAMODEL C_R	to add a plot of the fit model to the current slice plot on the graphics device.
> TKRESID C_R	to plot the data minus the fit model on the graphics device, erasing any previous plot.
> TKARESID C_R	to add a plot of the residuals (data minus model) to the current slice plot on the graphics device.

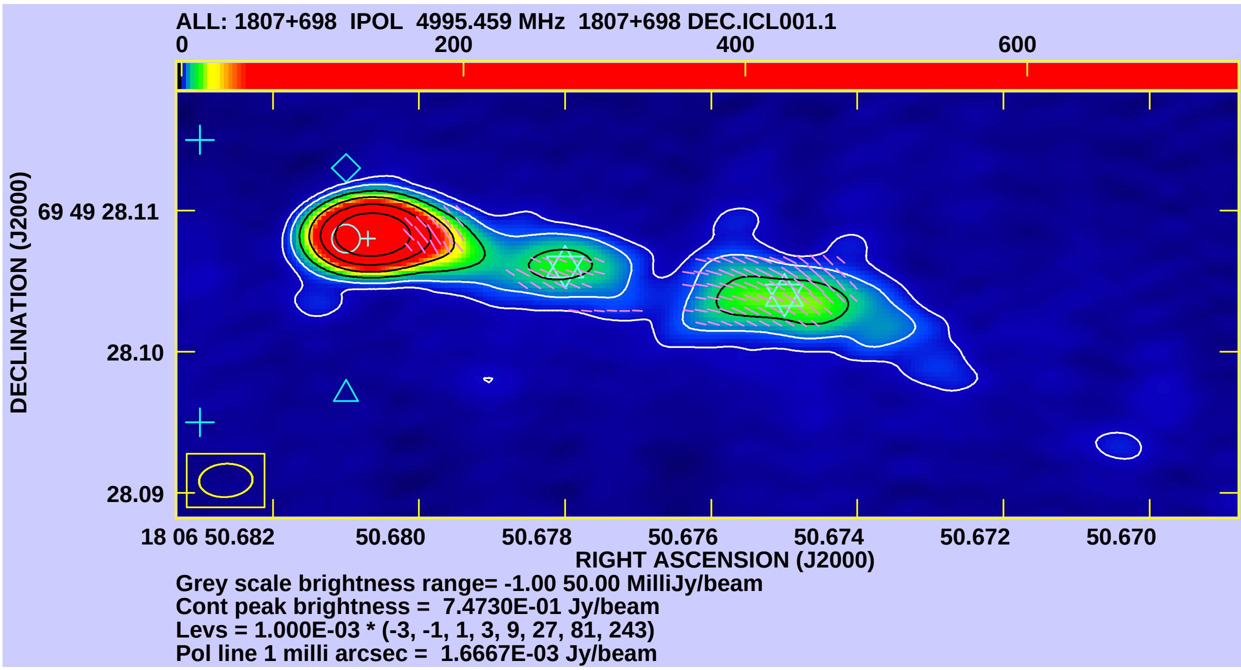
The units for the slice model parameters are fairly problematic, so we recommend using these graphical input and output functions. At least, they all have the same strange ideas. See § 6.4.8 for the verbs that allow this same processing using the TV display.

6.5.3 Data analysis with the graphics display

Tasks `XPLOT` and `XBASL` have modes which use the graphics display. They also can use the TV display which is now as fast as the graphics and more reliable.

6.6 Examples of color plotting

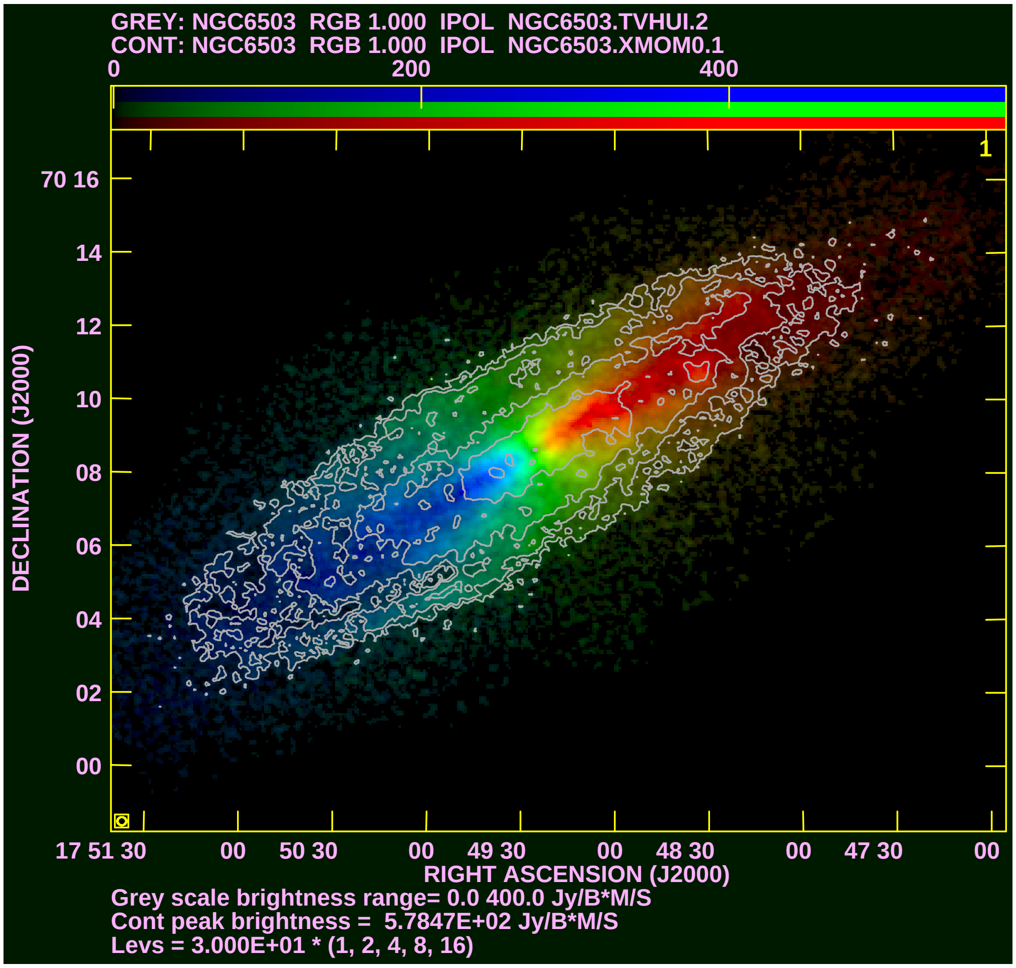
The following six pages are in color and should be printed on a color PostScript printer if possible. They will also print on a black and white printer but will then not show what they are supposed to illustrate.




```
KNTR:  Task to generate a plot file for a contour and grey plot
DOCONT        1                  > 0 => do contours
                                 (1 or 2 => which name)
DOGREY        1                  > 0 => do grey scale
                                 (1 pr 2 => which name)
DOVECT        1                  > => do polarization vectors
                                 (1 or 2 => which is IPOL)
                                 Contour or grey or IPOL
INNAME     '1807+698 DEC'        First image name (cube?)
INCLASS    'ICL001'              First image class
INSEQ         1                  First image seq. #
INDISK        3                  First image disk drive #
                                 Polarization intensity image:
IN3NAME    '1807+698 DEC'          (name) blank => INNAME
IN3CLASS   'QCL001'                (class) blank => 'PPOL'
IN3SEQ        1                    (seq. #) 0 => high
IN3DISK       3                    Disk drive #,   0 => any
                                 Polarization angle image:
IN4NAME    '1807+698 DEC'          (name) blank => INNAME
IN4CLASS   'UCL001'                (class) blank => 'PANG'
IN4SEQ        1                    (seq. #) 0 => high
IN4DISK       3                    Disk drive #,   0 => any
PIXRANGE   -1.00E-03      0.05   Min,Max of image intensity
FUNCTYPE   'SQ'                  Image intensity transfer func
OFMFILE    '                                                 '
LTYPE         3                  Type of labeling: 3 standard
DOALIGN       1                  > 0 => images must line up
CLEV        0.001                Absolute value for levs
LEVS         -3          -1      Contour levels (up to 30).
              1           3        9          27
             81         243      *rest 0
FACTOR     1000                  Mult. factor for Pol vector
XINC          3                  X-inc. of Pol vectors. 0=>1
YINC          3                  Y-inc. of Pol vectors. 0=>1
PCUT        0.001                Pol. vector cutoff. P units.
ICUT        0.001                Int. vector cutoff. I units.
DOWEDGE       3                  = 3 => put on top using full
                                        range of image values
STFACTOR      1                  Scale star sizes: 0 => none.
CBPLOT        1                  Position for beam plot:&
                                   1: lower left (default)
DARKLINE      0.33               Switch to dark lines when
                                 grey-scale > DARKLINE 0-1

LWPLA:   Sends plot file(s) to a PostScript printer or file
INNAME     '1807+698 DEC'        Image name (name)
INCLASS    'ICL001'              Image name (class)
INSEQ         1                  Image name (seq. #)
INDISK        3                  Disk drive #
PLVER         2                  Version # of PL file. 0=>last
FUNCTYPE   '  '                  'NE','LG','NG', 'SQ', 'NQ'
                                 else linear
DPARM      *all 0                (1,2) Clip recorded grays
DODARK        1                  Paint dark vectors as "dark"
OFMFILE    'RAINBOW'             Color grey scales....
DOCOLOR       1                  Use PLCOLORS ?
PLCOLORS      1          1       Line, character, background
              0          1       colors - see HELP.
              1          1          1          0.5
              1          0          1          1
              0          0          0          0
              0          0          0          0
              0          0          0          0
              0          0          0          0.8
              0.8        1
```

Figure 6.8: `KNTR` does polarization lines, contours, and grey-scale. Then `LWPLA` converts the grey-scale to pseudo-color and colors the lines making dark contours dark but dark polarization lines and stars bright. Data courtesy of Greg Taylor.

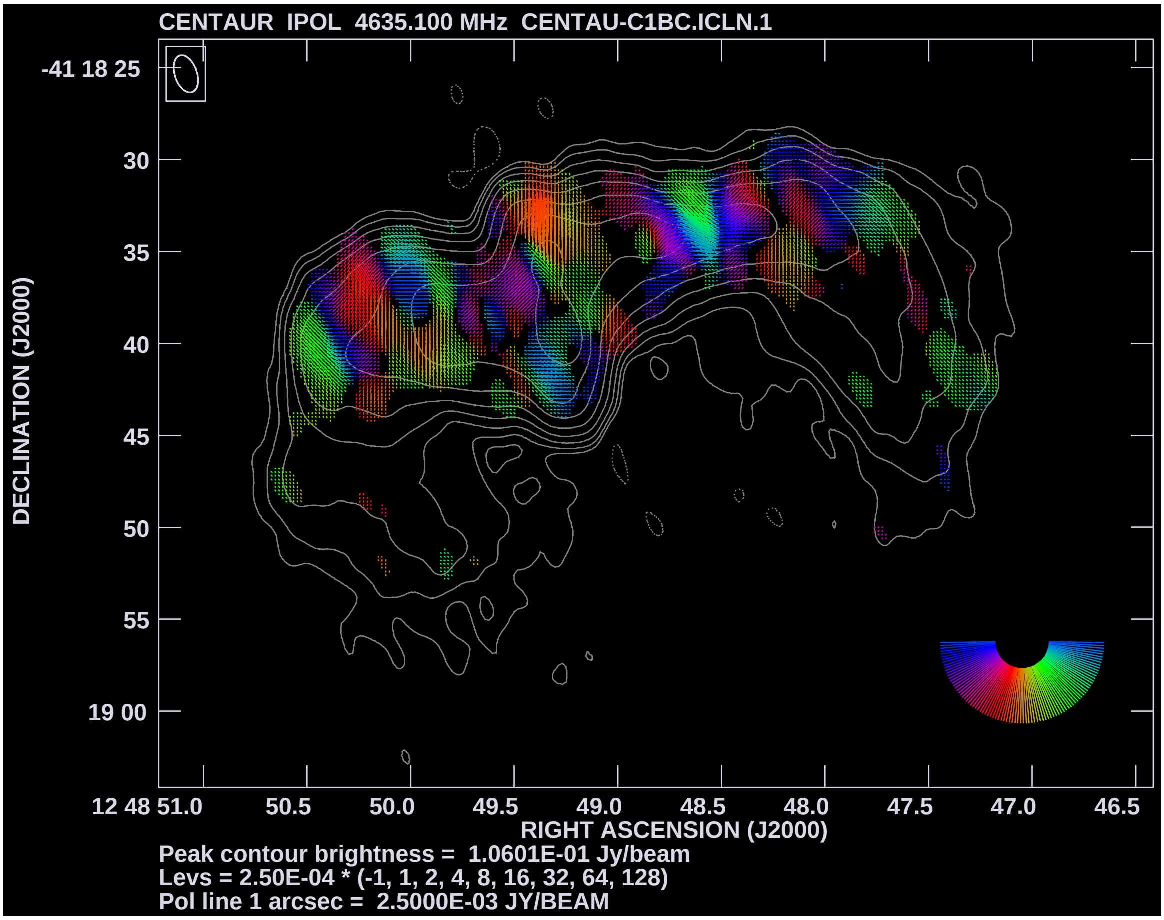



```
KNTR:  Task to generate a plot file for a contour & grey plot
DOCONT        2                    > 0 => do contours
                                   (1 or 2 => which name)
DOGREY        1                    > 0 => do grey scale
                                   (1 pr 2 => which name)
INNAME      'NGC6503     '         First image name (cube?)
INCLASS     'TVHUI '               First image class
INSEQ         2                    First image seq. #
INDISK        3                    First image disk drive #
                                   Contour or grey or IPOL
IN2NAME     'NGC6503     '         Second image name
IN2CLASS    'XMOMO '               Second image class
IN2SEQ        1                    Second image seq. #
IN2DISK       3                    Second disk drive #
PIXRANGE      0         400        Min,Max of image intensity
FUNCTYPE    'SQ'                   Image intensity transfer func
                                     'SQ' Square root
DOCOLOR       1                    Do RGB images as 3-color?
LTYPE        -3                    Type of labeling: 3 standard
                                   <0 -> no date/time
DOALIGN       1                    > 0 => images must line up
                                      (see HELP DOALIGN)
DOBLANK      -1                    Draw boundary between blanked
                                   areas and good areas?
DOWEDGE       3                    = 3 => put on top using full
                                         range of image values
```

```
CBPLOT        1                    Position for beam plot:
                                     1: lower left (default)
DODARK        1                    Plot dark vectors as black?
DARKLINE      0.33                 Switch to dark lines when
                                   grey-scale > DARKLINE 0-1

LWPLA:   Sends plot file(s) to a PostScript printer or file
INNAME      'NGC6503     '         Image name (name)
INCLASS     'TVHUI '               Image name (class)
INSEQ         2                    Image name (seq. #)
INDISK        3                    Disk drive #
FUNCTYPE    '  '                   'NE','LG','NG', 'SQ', 'NQ'
                                   else linear
DPARM       *all 0                 (1,2) Clip recorded grays
DODARK        1                    Paint dark vectors as "dark"
OFMFILE     *all ' '               Color grey scales....
DOCOLOR       1                    Use PLCOLORS ?
PLCOLORS      1         1          Line, character, background
              0         0.67       colors - see HELP.
              0.67      0.67          0          0
              0         0             0          0
              0         0             0          0.67
              0.67      0.67          0          0
              0         0             0          0
              1         0.7           1          0
              0.1       0
```

Figure 6.9: KNTR interprets the output of TVHUI as a three-color RGB image and overlays moment-0 contours. LWPLA adds coloring to the lines, using a less than pure white for both bright and dark contours so that they are not so dominant. Data courtesy of Eric Greisen, Kristine Spekkens, and Gustaaf van Moorsel.

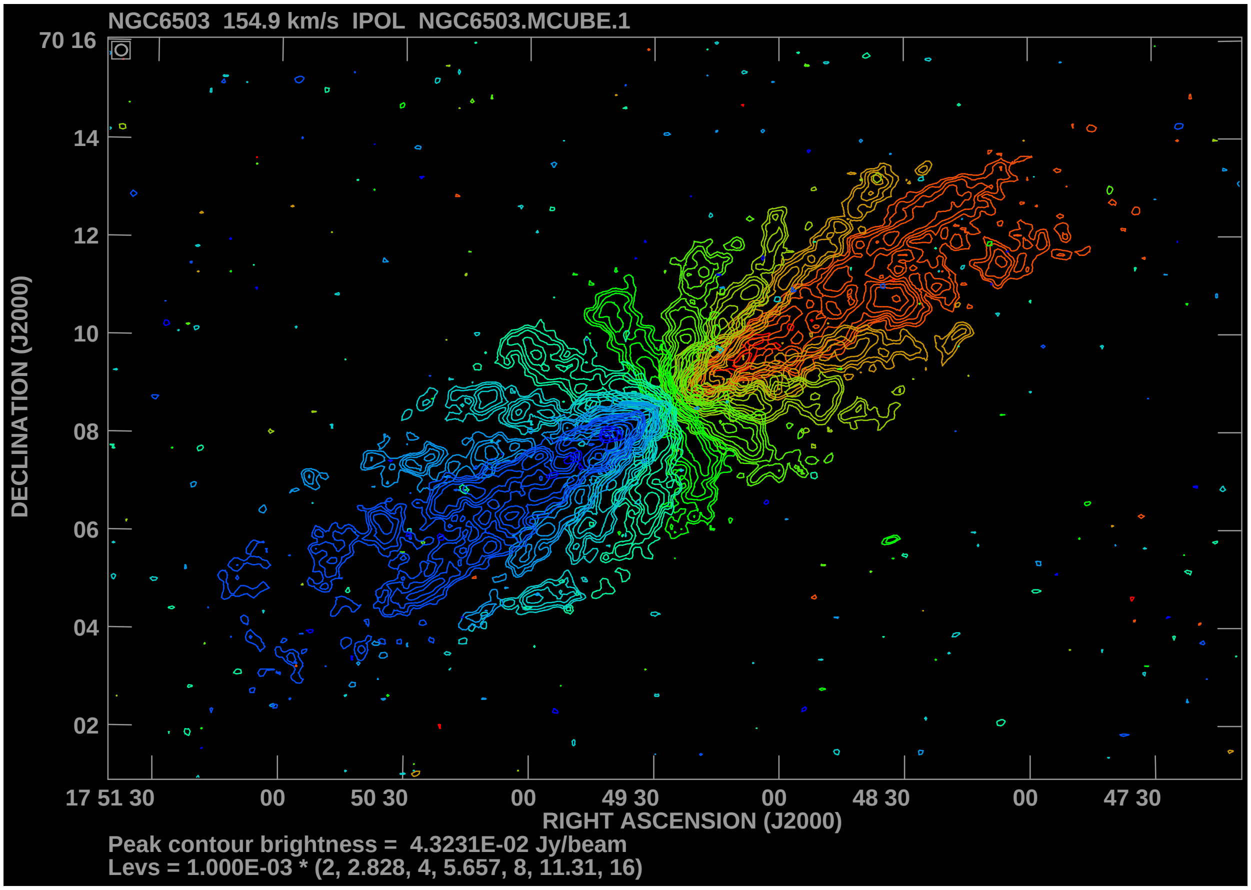



```
PCNTR:  Task to generate plot file for contour plus pol. vectors
DOCONT        1                     Draw contours? > 0 => yes
DOVECT        1                     Draw pol. vectors? > 0 => yes
DOGREY       -1                     Draw grey-scale image?
                                    Total intensity image:
INNAME     'CENTAU-C1BC '             Image name (name)
INCLASS    'ICLN  '                   Image name (class)
INSEQ         1                       Image name (seq. #)
INDISK        1                       Disk unit #
                                    Polarization intensity image:
IN2NAME    'CENTAU-C1BC '             (name) blank => INNAME
IN2CLASS   'QCLN  '                   (class) blank => 'PPOL'
IN2SEQ        1                       (seq. #) 0 => high
IN2DISK       0                       Disk drive #, 0 => any
                                    Polarization angle image:
IN3NAME    'CENTAU-C1BC '             (name) blank => INNAME
IN3CLASS   'UCLN  '                   (class) blank => 'PANG'
IN3SEQ        1                       (seq. #) 0 => high
IN3DISK       0                       Disk drive #,  0 => any
LTYPE        -3                     Type of labeling:
                                    <0 -> no date/time
CLEV       2.500E-04                Absolute value for levs
LEVS         -1           1         Contour levels (up to 30).
              2           4           8          16
             32          64          128         *rest 0
FACTOR     1000                     Mult. factor for Pol vector
                                       (see HELP)
```

```
ROTATE        0                     Angle to rotate Pol vector
                                      (in degrees)
XINC          1                     X-inc. of Pol vectors. 0=>1
YINC          1                     Y-inc. of Pol vectors. 0=>1
PCUT       1.250E-04                Pol. vector cutoff. P units.
ICUT       2.500E-04                Int. vector cutoff. I units.
POL3COL     160                     Color polarization vectors
                                    value in degrees = red
CBPLOT        4                     Position for beam plot:
                                      4: upper left
```

```
LWPLA:   Sends plot file(s) to a PostScript printer or file
RGBGAMMA      1           1         Gamma correction to apply
              2.5
DPARM      *all 0                   (1,2) Clip recorded grays
                                       before FUNCTYPE (0 to 1)
DOCOLOR       1                     Use PLCOLORS ?
PLCOLORS      0.85        0.85      Line, character, background
              0.9         0.5       colors - see HELP.
              0.5         0.5         0           0
              0           0           0           0
              0           0           0           0
              0           0           0           0
              0           0           0           0
              0.85        0.85        0.9      *rest 0
```

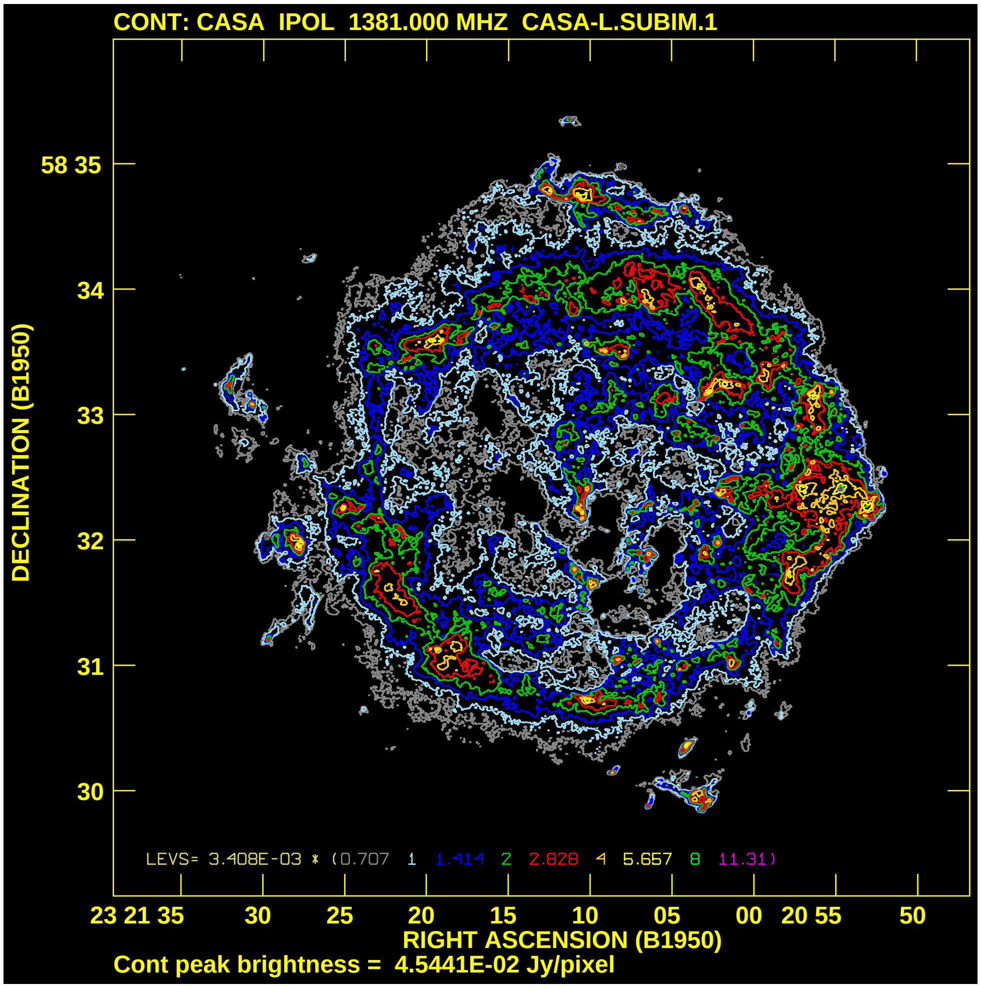
Figure 6.10: PCNTR plots contours and polarization vectors of Centaurus A. Color is used to show the complex changes in polarization position angle since the angles of short lines cannot be seen accurately. Data courtesy of Greg Taylor. For a discussion of this amazing pattern see Taylor, G.B., Fabian, A.C., & Allen, S.W. 2002, MNRAS, 334, 769, astro-ph/0109337 "Magnetic Fields in the Centaurus Cluster."




```
PCNTR:  Task to generate plot file for contour plus pol. vectors
DOCONT        1                        Draw contours? > 0 => yes
DOVECT       -1                        Draw pol. vectors? > 0 => yes
BLC          51         167            Bottom left corner of images
             39        *rest 0
TRC         397         393            Top right corner of images
             89        *rest 0
CON3COL       5                        > 0 => overplot contours in
                                       color of multiple planes
                                       ZINC is CON3COL.
CLEV       0.001                       Absolute value for levs
LEVS          2           2.8284       Contour levels (up to 30).
              4           5.6569          8           11.3137
             16        *rest 0
CBPLOT        4                        Position for beam plot:
                                         4: upper left
```

```
LWPLA:    Sends plot file(s) to a PostScript printer or file
LPEN          3                        Pen width (dots).
RGBGAMMA      1           1            Gamma correction to apply
              2.5
DPARM      *all 0                      (1,2) Clip recorded grays
DOCOLOR       1                        Use PLCOLORS ?
PLCOLORS      0.6         0.6          Line, character, background
              0.6         0.06275      colors - see HELP.
              1           0               1           0.6706
              1           0               1           1
              0           0               0           0
              0           0               0           0
              0           0               0           0
              0.6         0.6             0.6       *rest 0
```

Figure 6.11: PCNTR plots contours every fifth plane from a data cube using colors related to the velocity. LWPLA adds coloring to the labeling and background and applies a gamma correction to blue. Data courtesy of Eric Greisen, Kristine Spekkens, and Gustaaf van Moorsel.

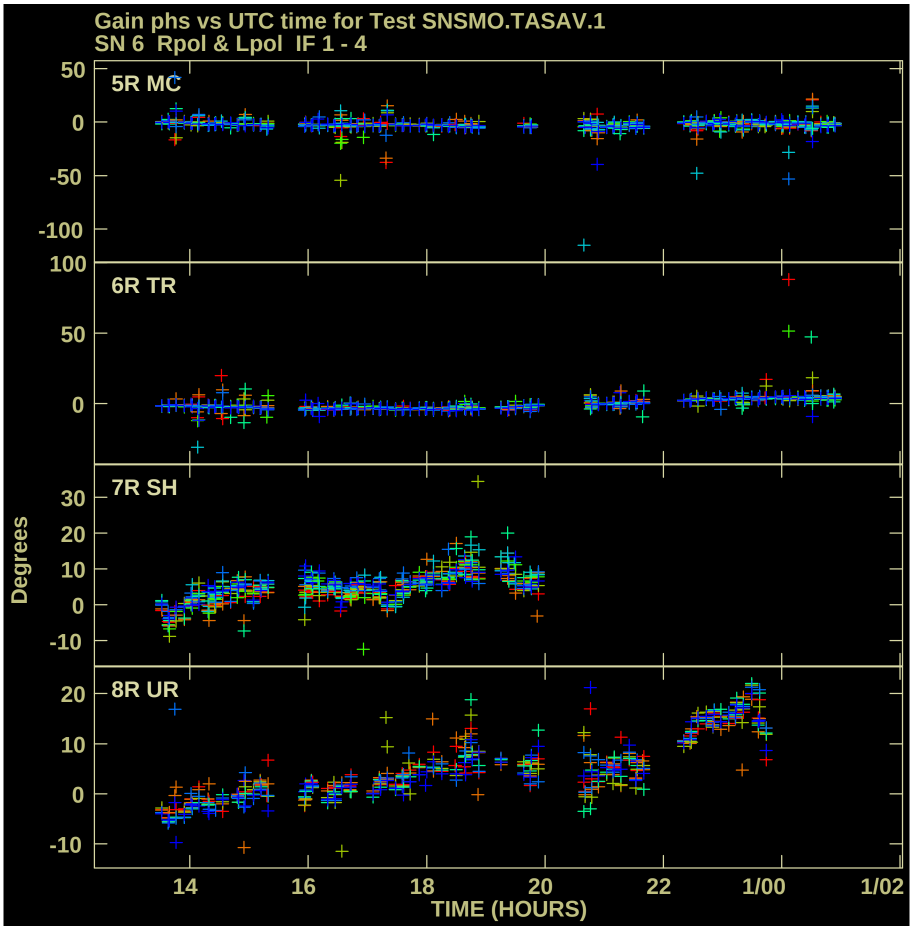



```
KNTR:  Task to generate a plot file for a contour & grey plot
DOCONT        1                    > 0 => do contours
                                   (1 or 2 => which name)
DOGREY       -1                    > 0 => do grey scale
                                   (1 pr 2 => which name)
DOVECT       -1                    > => do polarization vectors
                                   (1 or 2 => which is IPOL)
LTYPE        -3                    Type of labeling: 1 border,
                                   2 no ticks, 3 standard, 4 rel
                                   to center, 5 rel to subim cen
                                   6 pixels, 7-10 as 3-6 with
                                   only tick labels
                                   <0 -> no date/time
                                   special values for RGBLEVS
CLEV        0.003408               Absolute value for levs
                                      (used only if PLEV = 0).
LEVS          0.7071      1        Contour levels (up to 30).
              1.4142      2           2.8284      4
              5.6569      8          11.3137    *rest 0
CON3COL       0                    Color the contours by plane
```

```
RGBLEVS       0.2562      0.2562    Color each value of LEVS
              0.2562      0.3511      0.7297      0.9035
              0           0           1           0
              0.6205      0           1           0
              0           1           0.6205      0
              1           1           0           0
              0.8503      0           0.6594      0
              0.6594    *rest 0

LWPLA:  Sends plot file(s) to a PostScript printer or file
LPEN          3                     Pen width (dots).
RGBGAMMA      2.2         2.2       Gamma correction to apply
              2.2
OFMFILE     *all ' '                Color grey scales....
DOCOLOR       1                     Use PLCOLORS ?
PLCOLORS      1           1         Line, character, background
              0           0.06275   colors - see HELP.
              1           0           1           0.6706
              1           0           1           1
              0           0           0           0
              0           0           0           0
              0           0           0           0
              1           1         *rest 0
```

Figure 6.12: KNTR plots contours of Cassiopeia A with each contour level separately colored under control of adverb RGBLEVS. The values of RGBLEVS were set by a procedure call STEPLEVS(10) made available by RUN SETRGBL. Image is from the *Images from the Radio Universe* CD, 1992, NRAO with the particular image from Anderson M., Rudnick, L., Leppik, P, Perley, R. & Braun, R. 1991, ApJ, 373, 146.




```
SNPLT:  Task to plot selected contents of SN, TY, PC or CL file.
INEXT      'SN'                    Input 'SN','TY','PC', 'CL'
INVERS       6                     Input table file version no.
STOKES     '    '                  Stokes type to plot: R, L,
                                      RR, LL, RRLL, DIFF, RATO
BIF          0                     First IF to plot, 0=>1.
EIF          0                     Last IF to plot 0 -> highest
ANTENNAS     5          6          Antennas to plot 0=>all
             7          8          *rest 0
NPLOTS       4                     Number of plots per page
XINC         1                     Plot every XINC'th point
OPTYPE     'PHAS'                  Data to be plotted:
                                   'PHAS','AMP ','DELA','MDEL',
                                   'RATE','TSYS','TANT','ATM ',
                                   'GEO ','DOPL','SNR ','SUM '
                                   'CCAL', 'DDLY', ' '=phas.
OPCODE     'ALSI'                  Type of plot:
                                   'IFDF' => diff BIF and EIF
                                   'IFRA' => ratio BIF and EIF
                                   'ALIF' => combine all IFs
                                   'ALST' => combine all Stokes
                                   'ALSI' => all IFs & Stokes
DO3COL       1                     > 0 use 3-color symbols for
                                   ALIF, ALST, ALSI OPCODEs and
                                   SUM OPTYPE.
XAXIS        0                     Variable data is to be
                                   plotted against, 0=>time.
```

```
FACTOR       1.5                   Scale plot symbols by FACTOR
                                   0 -> 1
LTYPE       -3                     Type of labeling: 1 border,
                                   2 no ticks, 3 - 6 standard,
                                   7 - 10 only tick labels
                                   <0 -> no date/time

LWPLA:   Sends plot file(s) to a PostScript printer or file
PLVER        0                     Version # of PL file. 0=>last
INVERS       0                     PL file version #, upper
                                   limit if > PLVER
ASPMM        0                     Arc sec. per mm. 0=self scale
LPEN         3                     Pen width (dots).
RGBGAMMA     1          1          Gamma correction to apply
             2.5
DPARM      *all 0                  (1,2) Clip recorded grays
                                       before FUNCTYPE (0 to 1)
DOCOLOR      1                     Use PLCOLORS ?
PLCOLORS     0.85       0.85       Line, character, background
             0.65       0.6        colors - see HELP.
             0.6        0.6           0          0
             0          0             0          0
             0.75       0.75          0.5        0
             0          0             0          0
             0          0             0          0
             0.75       0.75          0.5      *rest 0
```

Figure 6.13: SNPLT plots phases for four antennas with color indicating polarization and IF channel. Stokes 1, IF BIF is pure red changing through yellow, green, and cyan to Stokes 2, IF EIF as pure blue. When all symbols lie on top of each other, the last one (pure blue) will dominate.