4 CALIBRATING INTERFEROMETER DATA

This chapter focuses on ways to do the initial calibration of interferometric fringe-visibility data in *AIPS*. The sections which follow concentrate primarily on continuum calibration for connected-element interferometers, especially the VLA. However, the information in these sections is useful to spectral-line, solar, and VLBI observers as well. For additional advice on spectral-line calibration, see § 4.7; for advice on calibrating observations of the Sun, see § 4.8; and for the gory details of VLBI, read Chapter 9. After the initial calibration has been completed, data for sources with good signal-to-noise are often taken through a number of cycles of imaging with self-calibration. See § 5.4 for information on these later stages of the reduction process. For accurate calibration, you must have accurate *a priori* positions and structural information for all your calibration sources and accurate flux densities for at least one of them. It is best if the calibration sources are unresolved "point" sources, but it is not required.

For the basic calibrations, visibility ("*uv*") data are kept in "multi-source data sets," each of which contains, in time order, visibility data for one or more "unknown" sources and one or more calibration sources. Associated with these data are "extension" files containing tables describing these data. When VLA archive data are first read into *AIPS* a number of basic tables are created and filled with information describing the data set. These are

1. `AN` (antennas) for sub-array geometric data, date, frequency, polarization information, *etc.*,
2. `FQ` (frequency) for frequency offsets of the different IFs (IF pairs in VLA nomenclature),
3. `NX` (index) to assist rapid access to the data,
4. `SU` (source) for source specific information such as name, position, velocity, and
5. `TY` (temperature) for measured system temperatures.

A null `CL` table is also created at this time. VLBI, and especially VLBA, data sets will end up with even more table files. Calibration and editing tasks then create, as needed, other tables including

6. `BL` (baseline) for baseline-, or correlator-, dependent corrections,
7. `BP` (bandpass) for bandpass calibration,
8. `CL` (calibration) for calibration and model information,
9. `FG` (flag) for flagging (editing) information, and
10. `SN` (solution) for gain solutions from the calibration routines.

All of these tables can be written to, and read back from, FITS files along with the visibility data. These, and any other, *AIPS* tables can be manipulated and examined using the general tasks `PRTAB`, `TACOP`, `TABED`, `TAMRG`, `TASRT` and `TAFLG`.

The visibility data within the multi-source data set are not normally altered by the calibration tasks. Instead, these tasks manipulate the tabular information to describe the calibration corrections to be applied to the data and any flagging (deletion) of the data.

The *AIPS* programs discussed in this chapter are part of a package that has been developed to calibrate interferometer data from a wide range of connected-element and VLB arrays, especially the VLA and VLBA. These programs therefore support many functions (and inputs) that are not required when calibrating normal VLA data. The examples given below show only the essential parameters for the operation being described, but, to get the results described, it is essential that you check *all* the input parameters before running any task. Remember that *AIPS* adverbs are global and will be "remembered" as you proceed. A list of calibration-related symbols is given in § 13.6, but a possibly more up-to-date list can be obtained by typing ABOUT CALIBRAT in your *AIPS* session. More general information on calibration can be routed to your printer by typing DOCRT FALSE ; EXPLAIN CALIBRAT C_R, while deeper information on a specific task is obtained with EXPLAIN *taskname* C_R.

When you are satisfied with the calibration and editing (or are simply exhausted), the task SPLIT is used to apply the calibration and editing tables and to write *uv* files, each containing the data for only one source. These "single-source" *uv* files are used by imaging and deconvolution tasks that work with only one source at a time. Many of the tasks described in this chapter will also work on single-source files. For VLA calibration, there are several useful procedures described in this chapter and contained in the RUN file called VLAPROCS. Each of these procedures has an associated HELP file and inputs. Before any of these procedures can be used, this RUN file must be invoked with:

> RUN VLAPROCS C_R to compile the procedures.

4.1 Copying data into *AIPS* multi-source disk files

There are several ways to write VLA data to *AIPS* multi-source *uv* data sets on disk. They include:

1. For VLA observations on or after January 1, 1988, use FILLM to read the VLA archive tape (or a copy thereof) directly.
2. For VLA observations before January 1, 1988, use FILLM on a translation of the original archive tape. All VLA archive data are being copied to Exabyte tapes, and are being translated to the modern format as needed. Contact the VLA data analysts (phone 505–835–7359, e-mail analysts@nrao.edu) to obtain a translated copy of any old observing files.
3. For an *AIPS* multi-source data set written to a FITS tape during an earlier *AIPS* session, use UVLOD or FITLD to read the tape.
4. For single-source data sets that are already on disk and are very similar in structure, use UV2MS on one of them to create a multi-source data set, and then on each of the others to append them to that multi-source data set. Each of the input data sets should have the same number of polarizations, IFs, spectral channels, and "random parameters." UV2MS also makes no corrections for differences in observed source positions or frequencies. After all are appended, use UVSRT to put the data in time-baseline order and INDXR to make an index and initial (null) calibration file.
5. For single-source data sets that are already on disk and are not sufficiently similar in structure for the method above, use MULTI on each single-source file to convert to multi-source format. Then use DBCON to concatenate the individual multi-source files into one big multi-source file. Finally use UVSRT, if needed, to put the data in time-baseline order and INDXR to make an index and initial (null) calibration file.

Data from other telescopes can be read into *AIPS* only if they are written in *AIPS*-like FITS files already or if you have a special format-translation program for that telescope. The VLBA correlator produces a format which is translated by the standard *AIPS* task FITLD; see § 4.1.2. Translation tasks for the Westerbork Synthesis Telescope (WSLOD) and the Australia Telescope (ATLOD) are available from the Dutch and Australians, respectively, but are not distributed by the NRAO with the normal *AIPS* system.

4.1.1 Reading from a VLA archive tape using FILLM

To load a *uv* data file to disk from a VLA archive tape, you must (hardware) mount the tape on a tape drive and then (software) mount the tape inside the AIPS program. See § 3.9 for a discussion of this process. It is strongly recommended that you begin by obtaining an index of the contents of your data tape. Reference dates, time ranges, file numbers, frequencies observed, and the like are reported in the index and are needed to guide the actual loading of the data. To print an index of the archive tape, use task PRTTP:

> TASK 'PRTTP' ; INP CR	to review the inputs needed.
> NFILES 0 CR	to start at the beginning of tape.
> PRTLEV 0 CR	to give complete summaries; only PRTLEV = -3 actually affects the output (adversely).
> DOCRT FALSE CR	to send output to the line printer.
> GO CR	to index the tape.

Typical inputs to FILLM might be:

> TASK 'FILLM' ; INP CR	to review the inputs needed.
> OUTNA ' ' CR	to take the default output file name.
> OUTDI 3 CR	to write the data to disk 3 (one with enough space).
> DOUVCOMP TRUE CR	to write visibilities in compressed format to save disk space.
> DOCONCAT TRUE CR	to concatenate files if this is second tape.
> DOALL TRUE CR	to include data from all frequency bands, source qualifiers, and numbers of spectral channels, writing as many output data sets as needed.
> VLAOBS 'AC238' CR	to select only data from observing program AC238. The default is to load data from *all* programs.
> NFILES 4 CR	to skip the first 4 files on the archive tape.
> CPARM 30,0 CR	to average the data for for 30 seconds; default is no averaging. Data weights will depend on T_{ant} as they should.
> CPARM(6) 1 CR	to select VLA sub-array 1.
> CPARM(7) 2000 CR	to have observations within 2 MHz be regarded as being at the same frequency.
> CPARM(8) 2 CR	to use a 2-minute interval for the CL table; default is 5 min.
> CPARM(9) 0.5 CR	to use a 30-second interval for the TY table; default is the input data interval.
> DPARM 0 CR	to have no selection by specific frequency.
> REFDATE '*yyyymmdd*' CR	to specify the year, month, and day of the reference date. This should be the first date in the data set (or earlier). All times in AIPS will be measured with respect to that date and must be positive. The default is the first date included by the data selection adverbs, which may not be the desired one. Note that REFDATE is only a reference point; it does not affect which data are loaded from the tape.
> TIMERANG *db* , *hb* , *mb* , *sb* , *de*, *he* , *me* , *se* CR	to specify the beginning day, hour, minute, and second and ending day, hour, minute, and second (wrt REFDATE) of the data to be included. The default is to include all times.
> INP CR	to review the inputs.
> GO CR	to run the program when you're satisfied with inputs.

Be careful when choosing the averaging time with CPARM(1). If you have a large data set, setting this time too *low* will make an unnecessarily large output file; this may waste disk space and slow the execution of subsequent programs. Setting it too *high* can, however, (1) smear bad data into good, limiting the ability to recognize and precisely remove bad data, (2) smear features of the image that are far from the phase center, and (3) limit the dynamic range that can be obtained using self-calibration. If you need a different (usually shorter) averaging time for the calibrator sources than for your program sources, use CPARM(10) to specify the averaging time for calibrators. See Lectures 12 and 13 in *Synthesis Imaging in Radio Astronomy*[1] for general guidance about the choice of averaging time given the size of the required field of view and the observing bandwidth.

Some words of warning about the use of NFILES are appropriate here. VLA archive tapes used to contain 3 tapes files for every actual data file. Most archive tapes today do not have these ANSI standard-label files, but still tend to begin with a header (non-data) tape file. If you set the NFILES adverb carefully based on the index printed by PRTTP, you should have no problem with these "excess" tape files.

FILLM is designed to read all your data from tape in one pass. All data meeting the selection criteria will be read from the input tape and filled into a *uv* multi-source file. Three selection criteria are always active: (a) TIMERANG, if non-zero, will restrict processing of data to the specified range of times (with respect to REFDATE); (b) VLAOBS, will restrict processing to the specified VLA observing code (which will be set to the code in the first valid data record if you do not specify it); and (c) CPARM(6) will restrict processing to the specified VLA sub-array (which, if you do not specify it, will be set to 1 if VLAOBS is not specified or to the first sub-array belonging to VLAOBS). All other selection criteria may be overridden by setting DOALL to TRUE.

Where possible, FILLM will try to place all data in one file. However, in many cases this is not possible. For instance so-called "channel 0" data from a spectral-line observation will be placed in a separate file from its associated line data. Similarly, scans which have differing numbers of frequency channels will also be placed into separate files. Another case is observations made in mode LP, *i.e.*, one IF-pair is set to L band, the other to P band. In this case the two bands will be split into separate files. Yet another case arises when there are observations of different bandwidths. All of this should be relatively transparent to the user.

If your data are on multiple tapes, you can write them all into the same data file by specifying DOCONCAT on the second (and subsequent) runs of FILLM.

A significant reduction in the disk space used may be achieved using the compressed format invoked by adverb DOUVCOMP; this factor is 1.89 for 2-IF continuum data and approaches 3.0 for line data. Most, but not quite all, tasks can process compressed *uv* data. The task UVCMP allows you to change the formats of *uv* data sets between *compressed* and *uncompressed*, if required to use one of these aberrant tasks.

FILLM and many *AIPS* tasks are able to handle multiple, logically different, frequencies within a multi-source data set. FILLM does this by assigning an FQ number to each observation and associating a line of information about that frequency in the FQ file associated with the data set. Users should note that this concept can become quite complicated and that not all tasks can handle it in full generality. In fact, most tasks can only process one FQ number at a time. Polarization calibration works only on one FQ at a time since the antenna file format allows for only one set of instrumental polarization parameters. Therefore, it is *strongly* advised that you fill continuum experiments which involve multiple frequencies into separate data sets. FILLM will separate bands automatically, but you will have to force any remaining separation. To do this, (a) use the QUAL adverb in FILLM, assuming that you have used separate qualifiers in OBSERVE for each frequency pair; (b) use the DPARM adverb array in FILLM to specify the desired frequencies precisely; or (c) use the UVCOP task to separate a multiple FQ data set into its constituent parts. Note that the first two options require multiple executions of FILLM, while the third option requires more disk space.

Spectral-line users and continuum observers using different frequencies in the same band should be aware of

[1] *Synthesis Imaging in Radio Astronomy*, Astronomical Society of the Pacific Conference Series, Volume 6 "A Collection of Lectures from the Third NRAO Synthesis Imaging Summer School" eds. R. A. Perley, F. R. Schwab and A. H. Bridle (1989)

the FQ entry tolerance. Each frequency in a *uv* file will be assigned an FQ number as it is read from tape by FILLM. For spectral-line users, the observing frequency will normally change as a function of time due to Doppler tracking of the Earth's rotation, or switching between sources or between spectral lines; in general, this will cause different scans to have different FQ numbers. FILLM assigns an FQ number to a scan based on the FQ tolerance adverb CPARM(7) which defines the maximum change of frequency allowed before a new FQ number is allocated. The default (CPARM(7) = 0) assigns the same FQ number to all data in spectral-line data sets. If CPARM(7) is positive, a scan will be assigned to an existing FQ number if

$$||\nu_{current} - \nu_{firstFQ}|| < \mathtt{CPARM(7)}$$

where $\nu_{firstFQ}$ is the frequency of the first sample to which the particular FQ number was assigned. If no match is found, then a new FQ number is created and assigned and another line added to the FQ table file. Alternatively, if CPARM(7) is negative then the FQ tolerance is assumed to be half of the maximum frequency difference caused by observing in directions 180 degrees apart (*i.e.*, $\Delta\nu = 10^{-4} \times \nu$).

An example: if an observer observes the 1612, 1665 and 1667 MHz OH masers in VY CMa and NML Cygnus, then presumably he would like his data to have 3 FQ numbers, one associated with each OH transition. However, running FILLM with CPARM(7) set to -1 would produce 6 FQ numbers because the frequency difference between the masers in VY CMa and NML Cygnus is greater than the calculated tolerance of 160 kHz. Therefore, in order to ensure that only 3 FQ numbers are assigned, he should set CPARM(7) to 1000 kHz. The default setting of CPARM(7) = 0 would result in all data having the same FQ number, which is clearly undesirable.

For most continuum experiments the FQ number will be constant throughout the database. Normally any change in frequency should be given a new FQ number. To achieve this, FILLM treats CPARM(7) differently for continuum. If CPARM(7) ≤ 0.0, then FILLM assumes a value of 100 kHz. A positive value of CPARM(7) is treated as a tolerance in kHz as in the spectral line case.

Note: *If your uv database contains several frequency identifiers, you should go through the calibration steps for each* FQ *code separately.*

FILLM is prepared to try to read past up to 50 parity or other tape errors. Do not be alarmed by a few warning messages, especially at the end of tape on old 9-tarck, half-inch tapes. These are relatively normal and will cause not harm.

If FILLM is executing correctly, your message terminal will report the number of your observing program, the VLA archive tape format revision number, and then the names of the sources as they are found on the tape. Once FILLM has completed, you can find the database on disk using:

> INDI 0 ; UCAT C_R

This should produce a listing such as:

```
Catalog on disk  3
Cat Usid Mapname      Class  Seq  Pt     Last access      Stat
  1  103 25/11/88    .X BAND.   1 UV 05-FEB-1994 12:34:16
```

You might then examine the header information for the disk data set by:

> INDI 3 ; GETN 1 ; IMH C_R

This should produce a listing like:

```
Image=MULTI     (UV)         Filename=25/11/88    .X BAND.   1
Telescope=VLA                Receiver=VLA
Observer=AC238               User #=  103
Observ. date=25-NOV-1988     Map date=05-FEB-1994
# visibilities    191317     Sort order  TB
Rand axes: UU-L-SIN  VV-L-SIN  WW-L-SIN  BASELINE  TIME1
```

```
          SOURCE  FREQSEL WEIGHT  SCALE
 ----------------------------------------------------------------
 Type    Pixels   Coord value  at Pixel    Coord incr   Rotat
 COMPLEX      1    1.0000000E+00    1.00 1.0000000E+00    0.00
 STOKES       4   -1.0000000E+00    1.00-1.0000000E+00    0.00
 IF           2    1.0000000E+00    1.00 1.0000000E+00    0.00
 FREQ         1    8.4110000E+09    1.00 1.2500000E+07    0.00
 RA           1     00 00 00.000    1.00      3600.000    0.00
 DEC          1     00 00 00.000    1.00      3600.000    0.00
 ----------------------------------------------------------------
 Maximum version number of extension files of type HI is   1
 Maximum version number of extension files of type AN is   1
 Maximum version number of extension files of type NX is   1
 Maximum version number of extension files of type SU is   1
 Maximum version number of extension files of type FQ is   1
 Maximum version number of extension files of type CL is   1
 Keyword = 'CORRMODE'  value = '        '
 Keyword = 'VLAIFS  '  value = 'ABCD    '
```

This header identifies the file as a multi-source file (`Image=MULTI`) with 191317 floating-point visibilities in time-baseline (`TB`) order. There are two entries on the `IF` axis. These correspond to the VLA's "AC" and "BD" IF-pairs respectively. The description of the frequency (`FREQ`) axis shows that the first IF ("AC") is at 8411 MHz and has 12.5 MHz bandwidth. The parameters of the second IF-pair ("BD") are determined from the data in the `FQ` table file and cannot be read directly from this header; these values are shown in the `'SCAN'` listing from `LISTR`. The header shown above indicates that the data are in compressed format since the number of pixels on the `COMPLEX` axis is 1 and the `WEIGHT` and `SCALE` random parameters are present. Uncompressed data does not use these random parameters and has 3 pixels on the `COMPLEX` axis.

The term "IF" can be confusing. At the VLA, IFs "A" and "C" correspond to right-hand and left-hand circularly polarized (RHC and LHC) signals, respectively, and are normally for the same frequency in an observing band. Such pairs, if at the same frequency, are considered to be one "IF" in AIPS. An observation which was made in spectral line mode "`2AC`" is considered at the VLA to have two "IFs" whereas within AIPS this would be filled as one "IF" with two polarizations if they were both observed with the same frequency, the same number of channels, and the same channel separation. If these conditions do not hold, then they are filled into separate *uv* files, each with a single IF and a single polarization. The term "sub-array" is also confusing. At the VLA — and in task `FILLM` — sub-array means the subset of the 27 antennas actually used to observe your sources. (The VLA allows up to 5 simultaneous sub-arrays in this sense.) In the rest of AIPS, sub-array refers to sets of antennas used together at the same time. If observations from separate times (*e.g.*, separate array configurations) are concatenated into the same file, then AIPS will regard the separate sets of antennas as different "sub-arrays" whether or not the same physical antennas occur within more than one of these sub-arrays.

If your experiment contains data from several bands `FILLM` will place the data from each band in separate data sets. Also, if you observed with several sets of frequencies or bandwidths in a given observing run these will be assigned different `FQ` numbers by `FILLM`. You can determine which frequencies correspond to which `FQ` numbers from the `'SCAN'` listing provided by `LISTR`. Line data are divided into the "channel 0" (central 3/4 of the of the observing band averaged) and the spectra. Data observed in the "LP" mode (or any other two-band mode) will be broken into separate data sets, one for each band.

As a practical note, setting the start and stop times of the data for your experiment with `TIMERANG` will cause `FILLM` to read all valid data up to the stop time you have specified and then exit normally. This way, it will not read to the end of the VLA archive file. The default action of `FILLM` (to read to the end of the tape if `TIMERANG = 0`) can be particularly annoying if your data are at the very start of a large archive data file.

When the data are successfully loaded to disk, type DISMOUNT CR to dismount your input tape.

4.1.2 Reading from a FITS tape with FITLD

FITLD is used to read FITS-format tapes into *AIPS*. It recognizes images, single- and multi-source *uv* data sets, and the special FITS *uv*-data tables produced by the VLBA correlator. In particular, VLA data sets that have been read into *AIPS* previously with FILLM and then saved to tape (or pseudo-tape disk) files with FITTP can be recovered for further processing with task FITLD. (The older task UVLOD will also work with *uv* data sets in FITS format, but it cannot handle image or VLBA-format files.)

A multi-source data file with all of its tables can be read from a FITS tape by:

> TASK 'FITLD' ; INP CR	to review the inputs needed.
> INTAPE *n* CR	to specify the tape drive for input from tape.
> INFILE '*filename*' CR	if the input is from a FITS disk file (see § 3.10.3).
> DOUVCOMP TRUE CR	to write visibilities in compressed format.
> OUTNA ' ' CR	take default (previous *AIPS*) name.
> OUTCL ' ' CR	take default (previous *AIPS*) class.
> OUTSEQ 0 CR	take default (previous *AIPS*) sequence #.
> OUTDI 3 CR	to write the data to disk 3 (one with enough space).
> INP CR	to review the inputs (several apply only to VLBA format files).
> GO CR	to run the program when you're satisfied with inputs.

FITLD is the equivalent of FILLM, but for output from the VLBA, rather than the VLA, correlator. The data-selection adverbs SOURCES, QUAL, CALCODE, and TIMERANG and the table-control adverbs CLINT and FQTOL are used, for VLBA-format data only, in FITLD in ways similar to the data-selection and control adverbs of FILLM. See Chapter 9 for more specific information.

4.2 Record keeping and data management

4.2.1 Calibrating data with multiple FQ entries

In general an observing run with the VLA, especially a spectral-line run, will result in a *uv* data file containing multiple FQ entries. In early versions of the *AIPS* software, the different FQ entries would automatically have been placed in different physical files. Now, FILLM allows you to place all of them in the same file. This may be convenient, but it has a number of costs. If a file contains multiple, independent frequencies, then it occupies more disk space and costs time in every program to skip the currently unwanted data (either a small cost when the index file is used or a rather larger cost when the file must be read sequentially). Since multiple frequencies are still not handled correctly in all programs (*i.e.*, polarization calibration) and since it is not possible to calibrate all of the different FQ data in one pass, you might consider separating the multiple frequencies into separate files (as described in § 4.1.1). In either case, you must calibrate each frequency with a separate pass of the scheme outlined below. There are three adverbs to enable you to differentiate between the different FQ entries: FREQID enables the user to specify the FQ number directly (with -1 or 0 meaning to take the first found); SELFREQ and SELBAND enable the user to specify the observing frequency and bandwidth to be calibrated (the tasks then determine to which FQ number these adverbs correspond). If SELFREQ and SELBAND are specified they override the value of FREQID.

There are certain bookkeeping tasks that must be performed between calibrating each FQ set. First, you must ensure that you have reset the fluxes of your secondary calibrators by running SETJY with OPTYPE =

'REJY' — if not, this will cause the amplitudes of your data to be incorrect. Second, it is wise to remove the SN tables associated with any previous calibration using the verb EXTDEST. Although this is not strictly necessary, it will simplify your bookkeeping.

A practical note: it is often useful to have used different qualifiers for different frequencies. This gives you another "handle" on the data. Unfortunately, not all programs use the QUAL, or even the CALCODE, adverb.

4.2.2 Recommended record keeping

It is useful to print a summary of the time stamps and source names of the scans in your data set. This reminds you of the structure of your observing program when you decide on interpolation and editing strategies, and may help to clarify relationships between later, more detailed listings of parts of the data set. It is also useful to have a printed scan summary and a map of the antenna layout if you need to return to processing the data months or years later. Finally, it is also making sure that all *AIPS* input parameters have their null (default) values before invoking the parts of the calibration package, such as CALIB, that have many inputs. The null settings of most parameters are arranged to be sensible ones so that basic VLA calibration can be done with a minimum of specific inputs; but some inputs may lose their default values if you interleave other *AIPS* tasks with the calibration pattern recommended below. Therefore, you should *always* review the input parameters with INP *taskname* C_R before running task *taskname*.

We suggest that you begin a calibration session with the following inputs:

> RESTORE 0 C_R	to set all inputs to null (default) values.
> TASK 'LISTR' ; INP C_R	to review the inputs needed.
> INDI *n*; GETN *m* C_R	to select the data set, $n = 3$ and $m = 1$ in FILLM example above.
> TPUT CALIB C_R	to store null values for later use with CALIB.
> OPTYP 'SCAN' C_R	to select scan summary listing.
> INP C_R	to review the inputs for LISTR.
> GO C_R	to run the program when the inputs are set correctly.

Note that the RESTORE 0 sets the adverbs to select all sources and all times and to send printed output to the printer rather than the terminal. It is also very useful to have a printed summary of your antenna locations, especially a list of which ones you actually ended up using. To do this, enter

> NCOUNT 0 C_R	to do all antennas
> INVERS *n* C_R	to do sub-array *n*
> GO PRTAN C_R	to print the list and a map of antenna locations.

In looking over the output from LISTR, you may notice that some of the sources you wish to use as calibrators have a blank "Calcode". To mark them as calibrators, use:

> TASK 'SETJY' ; INP C_R	to select the task and review its inputs.
> SOURCES '*sor1*' , '*sor2*' , '*sor3*' , . . . C_R	to select the unmarked calibrator sources.
> OPTYPE 'RESE' C_R	to reset fluxes and velocities.
> CALCODE 'C' C_R	to mark the sources as "C" calibrators.
> GO C_R	to run the task.

This operation will let you select the calibrators by their Calcodes rather than having to spell out their names over and over again. You may wish to consider separate calibrator codes for primary and secondary gain calibrators to make them easier to separate. Beginning with the 15JUL94 release, you may reset a calibrator code to blank by specifying CALCODE = '----'.

4.3 Beginning the calibration

After loading the data to disk, it has been traditional to begin with a substantial session of data checking and editing. With data from the VLA, this is always time consuming and often not necessary. Nonetheless, it is probably a good idea to check for two specific kinds of problems before beginning the actual calibration. These are corrupted data in the first record of most scans and totally dead antennas. Many other problems in the data are quickly and easily diagnosed by carefully inspecting the solution tables produced from the calibrators on un-edited data. Missing antennas and erratic amplitudes due to sampling problems and RF interference can be spotted from the `SN` tables and the closure-error messages produced by `CALIB`. If you *can't* spot errors from these, you may not need to edit the calibrator data. If the `SN` tables have well-behaved phases for most antennas and rapidly rotating phases for one or two, then you may need to apply baseline corrections rather than editing. See § 4.4.4 for details of how to make antenna-position corrections.

The next section tells how to detect simple problems in the data and eliminate them to reduce the warnings from the calibration tasks. The following sections tell you how to enter fluxes for the primary calibrator sources and do a preliminary calibration for all calibrators. In so doing, you should generate one or more solution (`SN`) tables containing the complex gains at the times of the calibration observations. These tables may be examined for problems with the observations. If you find problems, then you need to edit the data or apply baseline corrections and should consult § 4.4. If you do not find problems, you may proceed directly to § 4.5. (Of course, you may decide to edit the data from your program sources at a later stage of the data reduction and have to return to § 4.4 then.)

4.3.1 Initial editing

The warning messages from the calibrations described in the next sections may be reduced by flagging those antennas which were not actually working, but which were not flagged by the on-line system. Another problem that has plagued the VLA (and other interferometers) persistently is that the first record in scans can be corrupted; usually its amplitudes are lower than they should be. These data can be flagged using `TVFLG` or `UVFLG`, but this can be time consuming. The task `EDITA` described in § 4.4.2 is now likely to be the best initial (and perhaps only) editing tool which you need. For a more traditional approach, we recommend that you do the following before beginning your regular data editing. Use the task `LISTR` on your terminal (to save time and paper) to see if you have the problem:

> TASK 'LISTR' C_R	to set the data listing task
> INDI *n* ; GETN *m* C_R	to select the data set, $n = 3$ and $m = 1$ above.
> OPTYPE 'LIST' C_R	to select column listing format
> ANTEN *a1* , 0 C_R	to select one reliable antenna to display.
> BASEL 0 C_R	to select all baselines to this antenna.
> SOURCES ' ' ; CALCODE '*' C_R	to select all calibrator sources only.
> TIMER 0 C_R	to select all times.
> STOKES 'RR' C_R	to examine only one Stokes at a time.
> BIF 1 ; EIF BIF C_R	to specify the "AC" IFs only; it is quicker to look at only 1 IF at a time although more than one can be listed in sequence.
> FREQID 1 C_R	to select `FQ` number 1 (note that `FQ` numbers must also be done separately).
> DOCRT 132 C_R	to see full width display on the terminal. Use your window manager to stretch the window to $\geq$ 132 characters width.
> DOCALIB -1 C_R	to turn off calibration.
> DPARM 0 C_R	to select amplitudes with no averaging.
> INP C_R	to re-check *all* the inputs parameters.

> GO C_R	to start the task.

The task will prompt you for a C_R after each "page full" of output. When you have seen enough, enter **Q**. This display will let you determine whether the start-of-scan problem infects your data and, if so, how badly. If it is rare, forget it for now and use manual flagging methods later if needed. If it is widespread, use the *AIPS* task **QUACK**:

> TASK 'QUACK' C_R	
> SOURCES ' ' C_R	to select all sources.
> TIMER 0 C_R	to select all times.
> ANTENNAS 0 C_R	to select all antennas.
> FLAGVER 1 C_R	to insert flagging information in FG table 1.
> OPCODE 'BEG' C_R	flag first **APARM(2)** min of each scan.
> REASON 'BAD START OF SCAN' C_R	reason for the flagging.
> APARM 0 , 1/6 , 0 C_R	flag first 10 seconds of each scan.
> GO C_R	

The display generated above will also allow you to determine quickly which antennas are absent, which antennas are present but dead, and, with more careful examination, which antennas are flaky and may need special consideration. "Dead" antennas are visible in this display as columns with small numbers — columns that differ by factors of two or so from the others are generally fine. To be thorough, it is probably best to check the other IF:

> BIF 2 ; EIF 2 C_R	to specify the "BD" IFs.
> GO C_R	to run the program again.

as well as **STOKES = 'LL'**.

To remove the dead antennas, run **UVFLG**. For example, if antennas 6, 9, and 22 were bad for the full run in both IFs and Stokes, they could be deleted with

> TASK 'UVFLG' ; INP C_R	to select the editor and check its inputs.
> TIMER 0 C_R	to select all times.
> BIF 1 ; EIF 2 C_R	to specify the "AC" and "BD" IFs.
> BCHAN 0 ; ECHAN 0 C_R	to flag all channels.
> FREQID 1 C_R	to flag only the present **FQ** number.
> ANTEN 6 , 9, 22 C_R	to select the antennas.
> BASEL 0 C_R	to select all baselines to these antennas.
> STOKES ' ' C_R	to select all Stokes.
> REASON = 'ZOMBIE ANTENNA' C_R	to set a reason.
> FLAGVER 1 C_R	to select the first (only) flag table.
> INP C_R	be careful with the inputs here!
> GO C_R	to run the task when ready.

4.3.2 Primary flux density calibrators

The flux densities of 3C286 (1328+307) and 3C48 (0134+329) on the scale of Baars *et al.* (Astr. & Ap., **61**, 99 (1977)) are given in the 1990 VLA Calibrator Manual as:

3C286:

$$\log S = 1.480 + 0.292 \log \nu - 0.124 (\log \nu)^2$$

3C48:

$$\log S = 2.345 + 0.071 \log \nu - 0.138 (\log \nu)^2$$

where S = flux density in Jy and ν is Frequency in MHz. These values are, at a few selected frequencies:

Frequency (MHz)	S 3C286 (Jy)	S 3C48 (Jy)
1465	14.51	15.37
1680	13.55	13.76
4885	7.41	5.36
8415	5.20	3.15
14765	3.48	1.75
15035	3.44	1.71
22485	2.53	1.09

Careful measurements made with the D array of the VLA have shown that the Baars *et al.* (1977) coefficients are in error slightly, based on the assumption that the Baars' expression for 3C295 is correct; see the 1990 VLA Calibrator Manual. Revised values of the coefficients have been derived by Rick Perley. Task `SETJY` has these formulae built into it, giving you the option (OPTYPE 'CALC') of letting it calculate the fluxes for primary calibrator sources 3C295, 3C48, 3C286, 3C147, 3C138, and 1934-638. The default setting of APARM(2) = 0) will calculate the flux densities of 3C48, 3C147, and 3C286 according to the new (Perley) coefficients, while APARM(2) = 1 will calculate the flux densities using the original Baars *et al.* coefficients. `SETJY` will recognize both the 3C and IAU designations (B1950 and J2000) for these sources. You may insert your own favorite values for these sources instead (OPTYPE = ' ') and you will have to insert values for any other gain calibrators you intend to use.

Unfortunately, since both 3C48 and 3C286 are resolved by the VLA in most configurations and at most frequencies, they cannot be used directly to determine the amplitude calibration of the antennas without a detailed model of the source structure. Sources which are small enough to be substantially unresolved by the VLA have variable flux densities which must be determined in each observing session. A common method used to determine the flux densities of the secondary calibrators from the primary calibrator(s) is to compare the amplitudes of the gain solutions from the procedure described below.

Use `SETJY` to enter/calculate the flux density of each primary flux density calibrator. The ultimate reference for the VLA is 3C295, but 3C286 (1328+307), which is slightly resolved in most configurations at most frequencies, is the most useful primary calibrator. If you follow past practice at the VLA, you may have to restrict the *uv* range over which you compute antenna gain solutions for 3C286, and may therefore insert a "phony" flux density appropriate only for that *uv* range at this point. `CALIB` also has an option that will allow you to make use of Clean component models for primary flux density calibrators, in which case this step is redundant. An example of the inputs would be:

> TASK 'SETJY' ; INP C_R	
> SOURCES '3C286' , ' ' C_R	if you used 3C286 as the source name.
> ZEROSP 7.41 , 0 C_R	I flux 7.41 Jy, Q, U, V fluxes 0.
> BIF 1 ; EIF 1 C_R	selects "AC" IF.
> INP C_R	to review inputs.
> OPTYPE ' '	use values given in `ZEROSP`.
> GO C_R	when inputs okay.
> BIF 2 ; EIF 2 C_R	selects "BD" IF.
> ZEROSP 7.46, 0 C_R	I flux 7.46 Jy at the 2^{nd} IF, Q, U, V fluxes 0.
> GO C_R	

Note that, although `SOURCES` can accept a source list, `ZEROSP` has room for only one set of I, Q, U, V flux densities. To set the flux densities for several different sources or IFs, you must therefore rerun `SETJY` for each source and each IF, changing the `SOURCES`, `BIF`, `EIF`, and `ZEROSP` inputs each time.

If you wish, you can let **SETJY** calculate the fluxes, in which case it is able to do both IFs together.

> TASK 'SETJY' ; INP C_R	
> SOURCES '3C286' , ' ' C_R	if you used 3C286 as the source name.
> BIF 1 ; EIF 2 C_R	will calculate for both "AC" and "BD" IFs.
> OPTYPE 'CALC' C_R	perform the calculation.
> APARM(2) = 0 C_R	to use the VLA "1990" coefficients.
> INP C_R	to review inputs.
> GO C_R	when inputs okay.

4.3.3 First pass of the gain calibration

You are now ready to begin the actual calibration process. The first step is to determine a set of antenna gain solutions for both the primary and secondary flux density calibrator(s) within any *uv* limits that apply. This will be your first contact with **CALIB**, the central task in the *AIPS* calibration program. Most of the complexity of **CALIB** can be hidden using the procedure **VLACALIB**. Before attempting to invoke this procedure, you must first

> RUN VLAPROCS C_R	to compile the procedures.

Both 3C286 and 3C48 are resolved by the VLA in some configurations and frequencies. Point models for these sources are therefore only accurate over a limited range of baseline length. The range of baseline length used can be controlled by the adverb **UVRANGE**. If there are too few baselines to a given antenna, accurate solutions may not be possible; therefore, it is frequently necessary to limit the antennas used to the inner antennas on each arm. (The antenna pad numbers which include the order number from the array center on each arm can be determined by running **PRTAN**; see § 4.2.2.) **CALIB** may fail to produce any valid solutions if antennas with no data are included in the solution. The VLA Calibrator Manual suggests the following sets of **UVRANGE** (in kiloλ) and inner number of antennas.

```
3C48, 3C147, 3C138:
Band     UVRANGE    Array   No. ant. per arm   Notes
----   -----------  -----   ----------------  -----------------
90cm      0- 40      All          All

20cm      0- 40       A            7
            "       B,C,D         All

6cm       0- 40       A            3
            "       B,C,D         All

3.6cm     0- 40       A            2
            "         B            6
            "        C,D          All

2cm       0- 40       A            1           Not recommended
            "         B            4
            "        C,D          All

1.3cm     0- 40       A            1           Not recommended
            "         B            3
            "        C,D          All
```

3C286:

Band	UVRANGE	Array	No. ant. per arm	Notes
90cm	0- 18	A	7	
	"	B,C,D	All	
20cm	0- 18	A	4	
	"	B,C,D	All	
	90-180	A	All	Reduce flux 6%
6cm	0- 25	A	1	Not recommended
	"	B	4	
	"	C,D	All	
	150-300	A	All	Reduce flux 2%
3.6cm	50-300	A	3	Reduce flux 1%
	"	B	7	Reduce flux 1%
	"	C	All	Reduce flux 1%
	0- 15	D	All	
2cm	0-150	A	3	
	"	B,C,D	All	
1.3cm	0-185	A	2	
	"	B	7	
	"	C,D	All	

The values of `UVRANGE` for each secondary calibrator may be determined from the VLA Calibrator manual or by using `UVPLT` to plot the amplitudes as a function of baseline length. Since the latter works correctly only after a complete calibration has been done, it is often reasonable to use the 3C286/3C48 restrictions for all calibrator sources (at this stage). If your secondary calibrators are point sources over most baselines, then it may save you time to do the full calibration now. Not only will it save you, possibly, from re-running `CALIB` at a later time with a wider `UVRANGE`, but it will provide information on the data quality from the longer baselines.

Once you have read in procedure `VLACALIB`, you may invoke `CALIB` as follows:

> INDI *n* ; GETN *m* CR	to select the data set, $n = 3$ and $m = 1$ above.
> CALSOUR = '3C286' , ' ' CR	to name primary flux calibrator(s) individually, or
> CALCODE = '*' CR	to select any non-blank calibrator code.
> UVRANGE *uvmin uvmax* CR	*uv* limits, if any, in kiloλ.
> ANTENNAS *list of antennas* CR	antennas to use for the solutions, see discussion above.
> REFANT *n* CR	reference antenna number — use a reliable antenna located near the center of the array.
> MINAMPER 10 CR	display warning if baseline disagrees in amplitude by more than 10% from the model.
> MINPHSER 10 CR	display warning if baseline disagrees by more than 10 deg of phase from the model.
> FREQID 1 CR	use `FQ` number 1.
> INP VLACALIB CR	to review inputs.
> VLACALIB CR	to make the solution and print results.

This procedure will first run `CALIB`, then print any messages from `CALIB` about closure errors on the line printer, and finally run LISTR to print the amplitudes and phases of the derived solutions. Plots of these values may be obtained using task SNPLT.

If the secondary calibrators require different values of UVRANGE, then CALIB must be run several times. Attached to your input data set is a solution SN table. Each run of CALIB writes in this table (if SNVER = 1, for the times of the included calibration scans, the solutions for both the "AC" and "BD" IFs using the flux densities you set for your calibrators with SETJY or GETJY. (CALIB assumes a flux density of 1 Jy if no flux density is given in the SU table.) If a solution fails, however, the whole SN table can be compromised, forcing you to start over. It is possible to write multiple SN tables with SNVER = 0. Later programs such as GETJY and CLCAL will merge all SN tables which they find (if told to do so). Tables with failed solutions must be deleted.

The LISTR outputs provided by VLACALIB should be examined carefully to check on the calibration; amplitudes should be consistent (both among antennas and among time stamps) and phases should vary smoothly. If you decide that the solutions are not acceptable (*e.g.*, there are no valid solutions) *and* you are creating a new SN table on each run of CALIB, then delete that SN table using EXTDEST before proceeding. The later stages of processing assume that all extant SN tables are valid. Note that re-running CALIB on the same SN table simply over-writes the old solutions with new ones. CALIB gives messages which indicate the number of valid and invalid solutions which should help you evaluate the results. If VLACALIB is run using the values of MINAMPER and MINPHSER shown above, it will print a list of baselines and times which show substantial "closure" errors. (If you use CALIB directly rather than VLACALIB, you may use these adverbs plus CPARM(2-4) to get additional reports and statistics on closure errors.) It is important to remember that normal thermal noise and, at longer wavelengths, background confusion cause closure errors too. Thus, some closure error on weaker calibrators is to be expected and may be ignored. Interpreting closure errors is a real art, but a couple of gereralizations are possible. If the same closure error shows up in both polarizations and both IFs, then you have probably got a resolved object. If one antenna dominates the closure list, especially if it is at only one IF and/or one polarization, then you have got a bad antenna. If the errors are uniformly small, distributed amongst all antennas, and not correlated between IFs or polarizations, then you have simply noise and/or background confusion. In tis case, do *not* edit the data — the randomness of the "errors" nearly always averages out nicely and the solution is jsut fine. Large or systematic errors indicate either that the calibrator source is resolved or that there are problems with the data requiring editing. If a calibrator is being resolved, delete the bad SN table and re-run VLACALIB with an appropriate UVRANGE.

4.4 Assessing the data quality and initial editing

At each stage in the data calibration process, it is a good idea to take a look at the data to determine their quality and then to "flag" (edit, delete) those that are suspect or clearly bad. Having begun the actual calibration, it is important to get an impression of the overall quality of the data and to edit out any obviously corrupted data, (*e.g.*, bad integrations that were not detected and expunged by the on-line monitoring system, high amplitudes due to interference, unstable amplitudes due to undetected equipment problems, *etc*). During the initial calibration, you need to do this only on the observations of calibration sources. However, at a later stage, you may also need to apply techniques similar to those described below to your program sources. If you do edit any calibration data at this point, you must re-run CALIB following the instructions given above for the affected sources.

The philosophy of editing and the choice of methods are matters of personal taste and the advice given below should, therefore, be taken with a few grains of salt. When interferometers consisted of only a couple of movable antennas, there was very little data and it was sparsely sampled. At that time, careful editing to delete all suspect samples, but to preserve all samples which can be calibrated, was probably justified. But modern instruments produce a flood of data, with the substantial redundancy that allows for self-calibration on strong sources. Devoting the same care today to editing is therefore very expensive in your time, while the loss of data needlessly flagged is rarely significant. A couple of guidelines you might consider are:

- Don't flag on the basis of phase. At least with the VLA, most phase fluctuations are due to the atmosphere rather than the instrument. Calibration can deal with these up to a point, and self-calibration (if you have enough signal) can refine the phases to levels that you would never reach by flagging. The exceptions are (1) IF phase jumps which still happen on rare occasions, and (2) RF interference which sometimes is seen as an excursion in phase rather than amplitude.

- Don't flag on minor amplitude errors, especially if they are not common. Except for very high dynamic range imaging, these will not be a problem, and in those cases, self-calibration always repairs or sufficiently represses the problem.

- Don't flag if `CALIB` reports few closure errors and the `SN` tables viewed with `EDITA`, `SNPLT`, and `LISTR` and the calibrator data viewed with the matrix format of `LISTR` show only a few problems.

There are two general methods of editing in *AIPS*. The "old-fashioned" route uses `LISTR` to print listings of the data on the printer or the user's terminal. The user scans these listings with his eyes and, upon finding a bad point, enters a specific flag command for the data set using `UVFLG`. While this may sound clumsy, it is in fact quite simple and by far the faster method when there are only a few problems. In a highly corrupted data set, it can use a lot of paper and may force you to run `LISTR` multiple times to pin down the exact problems. The "modern" route uses interactive ("TV"-based) tasks to display the data in a variety of ways and to allow you to delete sections of bad data simply by pointing at them with the TV cursor. These tasks are `TVFLG` (§ 4.4.3) for all baselines and times (but only one IF, one Stokes, and one spectral channel) at a time, `SPFLG` (§ 10.2.2) for all spectral channels and times (but only one baseline, one IF, and one Stokes) at a time, `EDITA` (§ 4.4.2) for editing based on `TY` (T_{ant}), `SN` or `CL` table values, and `EDITR` (§ 5.5.2) for all times (but only a single antenna (1–11 baselines), channel, IF, and Stokes) at a time. `TVFLG` is the one used for continuum and channel-0 data from the VLA, while `SPFLG` is only used to check for channel-dependent interference. `SPFLG` is useful for spectral-line editing in smaller arrays, such as the Australia Telescope. (The redundancy in the spectral domain on calibrator sources helps the eyes to locate bad data.) `EDITR` is more useful for small arrays such as those common in VLBI experiments. `EDITA` has been found to be remarkably effective using VLA system temperature tables. All four tasks have the advantage of being very specific in displaying the bad data. Multiple executions should not be required. However, they require you to look at each IF, Stokes, channel (or baseline) separately (unless you make certain broad assumptions) and they require you to develop special skills since they offer so many options and operations with the TV cursor (mouse these days). A couple of general statements can be made

- For highly corrupted data (say with considerable RF interference, significant cross-talk between antennas, or erratic antennas) `TVFLG` is definitely preferred. It gives an overall view of the data which is far superior to that given by `LISTR`. RFI and similar problems are more troublesome at lower frequencies, so `TVFLG` is probably preferred for L, P, and "4" bands.

- Most VLA data at higher frequencies are of good quality and the flexibility and power of `TVFLG` are not needed. In such cases, `LISTR` with `OPCODE = 'MATX'` can efficiently find scans with erroneous points.

- The displays given by `TVFLG` and, to a lesser extent, `LISTR` in its `MATX` mode are less useful when there are only a few baselines. Thus, for arrays smaller than the VLA, users may wish to use `SPFLG` on spectral-line data sets and `EDITR` on continuum data sets.

- A reasonable strategy to use is to run `LISTR` first. If there are only a few questionable points, use `LISTR` and `UVFLG`, otherwise switch to an interactive task, such as `EDITA` to begin with followed by `TVFLG`.

4.4.1 Editing with LISTR and UVFLG

Data may be flagged using task **UVFLG** based on listings from **LISTR**. To print out the scalar-averaged raw amplitude data for the calibrators, and their *rms* values, once per scan in a matrix format, the following inputs are suggested:

> TASK 'LISTR' ; INP CR	to review the inputs needed.
> INDI *n*; GETN *m* CR	to select the data set, $n = 3$ and $m = 1$ above.
> SOURCES ' ' ; CALCODE '*' CR	to select calibrators.
> TIMER 0 CR	to select all times.
> ANTENNAS 0 CR	to list data for all antennas.
> OPTYPE 'MATX' CR	to select matrix listing format.
> DOCRT FALSE CR	to route the output to printer, not terminal.
> DPARM 3 , 1 , 0 CR	amplitude and *rms*, scalar scan averaging.
> BIF 1 CR	to specify the "AC" IFs (note that IFs must be listed separately).
> FQID 1 CR	to select **FQ** number 1 (note that **FQ** numbers must also be done separately).
> INP CR	to review the inputs.
> GO CR	to run the program when inputs set correctly.
> BIF 2 CR	to specify the "BD" IFs.
> WAIT ; GO CR	to wait for the previous execution to finish and then run the program again.

For unresolved calibrators, the VLA on-line gain settings normally produce roughly the same values in all rows and columns within each matrix. At L, C, X, and U bands, these values should be approximately 0.1 of the expected source flux densities. At P band, the factor is about 0.01. The factors for other bands are unspecified. Any rows or columns with consistently high or low values in either the amplitude or the *rms* matrices should be noted, as they probably indicate flaky antennas. In particular, you should look for

- In the amp-scalar averages, look for *dead* antennas, which are easily visible as rows or columns with small numbers. Rows or columns that differ by factors of two or so from the others are generally fine. Such deviations mean only that the on-line gains were not set entirely correctly.
- In the *rms* listings, look for discrepant high values. Almost all problems are antenna based and will be seen as a row or column. Factors of 2 too high are normally okay, while factors of 5 high are almost certainly indicative of serious trouble.

The next step is to locate the bad data more precisely. Suppose that you have found a bad row for antenna 3 in right circular polarization in IF 2 between times ($d1$, $h1$, $m1$, $s1$) and ($d2$, $h2$, $m2$, $s2$). You might then rerun **LISTR** with the following new inputs:

> SOURCES ' ' CR	to select all sources.
> TIMER *d1 h1 m1 s1 d2 h2 m2 s2* CR	to select by time range.
> ANTENNAS 1 , 2 , 3 CR	to list data for antenna 3 with two "control" antennas.
> BASEL 1 , 2 , 3 CR	to list all baselines with these three antennas.
> OPTYPE 'LIST' CR	to select column listing format.
> DOCRT 132 CR	to route the output to terminal.
> DPARM = 0 CR	amplitude only, no averaging.
> STOKES 'RR' CR	to select right circular.
> BIF 2 CR	to specify the "BD" IFs.

> FLAGVER 1 C_R	to choose flag table 1.
> GO C_R	to run the program.

This produces a column listing on your terminal of the amplitude for baselines 1–2, 1–3 and 2–3 at every time stamp between the specified start and stop times. The '1–2" column provides a control for comparison with the two columns containing the suspicious antenna.

Note that "amp-scalar" averaging ignores phase entirely and is therefore not useful on weak sources, nor can it find jumps or other problems with the phases. To examine the data in a phase-sensitive way, repeat the above process, but set DPARM(2) = 0 rather than 1. Bad phases will show up as reduced amplitudes and increased *rms*'s.

Once bad data have been identified, they can be expunged using UVFLG. For example, if antenna 3 RR was bad for the full interval shown above, it could be deleted with

> TASK 'UVFLG' ; INP C_R	to select the editor and check its inputs.
> TIMER *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to select by time range.
> BIF 2 ; EIF = BIF C_R	to specify the "BD" IFs.
> BCHAN 0 ; ECHAN 0 C_R	to flag all channels.
> FREQID 1 C_R	to flag only the present FQ number.
> ANTEN 3 , 0 C_R	to select antenna 3.
> BASEL 0 C_R	to select all baselines to antenna 3.
> STOKES 'RR' C_R	to select only the RR Stokes (LL was found to be okay in this example).
> REASON = 'BAD RMS WHOLE SCAN' C_R	to set a reason.
> FLAGVER 1 C_R	to select the first (only) flag table.
> INP C_R	be careful with the inputs here!
> GO C_R	to run the task when ready.

Continue the process until you have looked at all parts of the data set that seemed anomalous in the first matrix listing, then rerun that listing to be sure that the flagging has cleaned up the data set sufficiently. If there are lots of bad data, you may find that you have missed a few on the first pass. If you change your mind about a flagging entry, you can use UVFLG with OPCODE = 'UFLG' to remove entries from the flag table. (Note that, if you use different REASONs for your different flag entries, then you can also undo all flags with a given REASON using OPCODE = 'REAS' in UVFLG.) If the table becomes hopelessly messed up, use EXTDEST to delete the flag table and start over or use a higher numbered flag table. The contents of the flag table may be examined at any time with the general task PRTAB and entries in it may also be removed with TABED and/or TAFLG.

4.4.2 Editing with EDITA

The task EDITA uses the graphics planes on the AIPS TV display to plot data from tables and to offer options for editing (deleting, flagging) the associated *uv* data. At this time, only the TY (system temperature), SN (solution), and CL (calibration) tables may be used. We recommend using EDITA with the TY tables to do the initial editing of VLA data sets, probably before running the programs described in § 4.3. For accuracy in evaluating and flagging your data, it is a good idea to have the TY table filled with the same interval as the data themselves; see § 4.1.1. Try:

> TASK 'EDITA ; INP C_R	to review the inputs needed.
> INDI *n* ; GETN *m* C_R	to select the data set, $n = 3$ and $m = 1$ above.
> INEXT 'TY' C_R	to use the system temperature table.
> INVERS 0 C_R	to use the highest numbered table, usually 1.

> TIMER 0 C_R	to select all times.
> FREQID 3 C_R	Select FQ entry 3.
> BIF 1 ; EIF 2 C_R	to specify both VLA IFs; you can then toggle between the two interactively.
> ANTENNAS 0 C_R	to display data for all antennas.
> FLAGVER 1 C_R	to use flag (FG) table 1.
> SOLINT 1/60 C_R	to declare that samples within 1 second of each other are regarded as occurring at the same time.
> DOHIST FALSE C_R	to omit recording the flagging in the history file.
> DOTWO TRUE C_R	to view a 2nd observable for comparison
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

If you make multiple runs of EDITA, it is important to make sure that the flagging table entries are all in the same version of the FG table. To ensure this you should set FLAGVER to 1 and keep it that way for all runs of EDITA. A sample display from EDITA is shown on the next page.

The following discussion assumes that you have read § 2.3.2 and are familiar with using the *AIPS* TV display. An item in a menu such as that shown in the figure is selected by moving the TV cursor to the item (holding down or pressing the left mouse button). At this point, the menu item will change color. To obtain information about the item, press *AIPS* TV "button D" (usually the D key and also the F6 key on your keyboard). To tell the program to execute the menu item, press any of *AIPS* TV buttons A, B, or C. Status lines around the display indicate what is plotted and which data will be flagged by the next flagging command. In the figure below, only the displayed antenna (1), IF (1), polarization (2), and time range will be flagged. You must display at least a few lines of the message window and your main AIPS window since the former will be used for instructions and reports and the latter will be needed for data entry (*e.g.*, antenna selection).

The first thing to do with EDITA is to look at all of the polarizations, IFs, and antennæ, in order to flag the obviously bad samples (if any). Use SWITCH POLARIZATION to switch between polarizations and ENTER IF to select the IF to edit. These options appear only if there is more than one polarization and/or more than one IF in the loaded data. Use ENTER ANTENNA to select the antenna to be flagged and ENTER OTHER ANT to select secondary antennæ to be displayed around the editing area. If the secondary antennæ have no obvious problems, then they do not have to be selected for editing. EDITA will plot all of the times in the available area, potentially making a very crowded display. You may select interactively a smaller time range or "frame" in order to see the samples more clearly. It is necessary to select each frame in order to edit the data in that frame so it helps to make the TV screen as big as possible with the F2 button or your window manager. Note that the vertical scales used by EDITA are linear, but that the horizontal scale is irregular and potentially discontinuous. Integer hours are indicated by tick marks and the time range of the frame is indicated. Use FLAG TIME or FLAG TIME RANGE to delete data following instructions which will appear on the message window. While you are editing, the source name, sample time and sample value currently selected will be displayed in the upper left corner of the TV screen. This information can also be used to determine if QUACK is needed. In the case of the data displayed in the figure, it was found that the source name changed to the source of the new scan *before* the T_{sys} reached the value appropriate to the source. This is seen in the scan with higher T_{sys} on the good antennæ, while antenna 1 seems to be dubious over the last half of the observation. Therefore, besides heavy editing of antenna 1 polarization 2, QUACK was needed with these data (which is almost always the case at least for spectral-line VLA observations).

Having flagged all obviously bad points, select SWITCH ALL IF, SWITCH ALL TIME, SWITCH ALL ANT, and SWITCH ALL POL so that the next flag command(s) apply to all of the data. Set the SCAN LENGTH long enough to include the shorter of the full scan and about 12 samples. Then display the difference between the current sample and the running mean by selecting SHOW TSYS - <T>. Use FLAG ABOVE and FLAG BELOW to flag all samples more than a few sigma away from the local mean. Finally, apply your flagging to your *uv* data set by selecting `EXIT`.

At this point, return to § 4.3.1 to run `QUACK` followed by the first pass of the gain calibration. Then run `TVFLG` below with `DOCAL TRUE` so that the data will be displayed on the same flux scale for all baselines.

4.4.3 Editing with TVFLG

If your data are seriously corrupted, contain numerous baselines, and you like video games, `TVFLG` is the visibility editor of choice. The following discussion assumes that you have read § 2.3.2 and are familiar with using the *AIPS* TV display. The following inputs are suggested:

> TASK 'TVFLG' ; INP C_R	to review the inputs needed.
> INDI *n* ; GETN *m* C_R	to select the data set, $n = 3$ and $m = 1$ above.
> SOURCES ' ' C_R	to select all sources.
> TIMER 0 C_R	to select all times.
> STOKES 'RRLL' C_R	to select both right and left circular polarizations; you can then toggle between RR and LL interactively.
> FREQID 3 C_R	Select `FQ` entry 3.
> BIF 1 ; EIF 2 C_R	to specify both VLA IFs; you can then toggle between the two interactively.
> ANTENNAS 0 C_R	to display data for all antennas.
> BASELINE 0 C_R	to display data for all baselines.
> DOCALIB FALSE C_R	to avoid applying calibration to raw data.
> FLAGVER 1 C_R	to use flag (FG) table 1.
> DPARM = 0 C_R	to use default initial displays and normal baseline ordering.
> DPARM(6) = 30 C_R	to declare that the input data are 30-second averages, or to have the data averaged to 30 seconds.
> DOCAT 1 C_R	to save the master grid file.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

If you make multiple runs of `TVFLG`, it is important to make sure that the flagging table entries are all in the same version of the `FG` table. To ensure this you should set `FLAGVER` to 1 and keep it that way for all runs of `TVFLG`.

`TVFLG` begins by constructing a "master grid" file of all included data. This can be a long process if you include lots of data at once. It is probably better to use the channel selection, IF selection, source selection, and time range selection adverbs to build rather smaller master grid files and then to run `TVFLG` multiple times. It will work with all data included, allowing you to select interactively which data to edit at any one moment and allowing you to resume the editing as often as you like. But certain operations (such as undoing flags) have to read and process the entire grid, and will be slow if that grid is large. The master grid file is always cataloged (on `IN2DISK` with class `TVFLGR`), but is saved at the end of your session only if you set DOCAT = 1 (actually > 0) before starting the task. To resume `TVFLG` with a pre-existing master grid file, set the adverb `IN2SEQ` (and `IN2DISK`) to point at it. When resuming in this way, `TVFLG` ignores all of its data selection adverbs since they might result in a different master grid than the one it is going to use. If you wish to change any of the data selection parameters, *e.g.*, channels, IFs, sources, times, or time averaging, then you must use a new master grid.

Kept with the master grid file is a special file of `TVFLG` flagging commands. This file is updated as soon as you enter a new flagging command, making the master grid and your long editing time virtually proof from power failures and other abrupt program terminations. These flagging commands are not entered into your

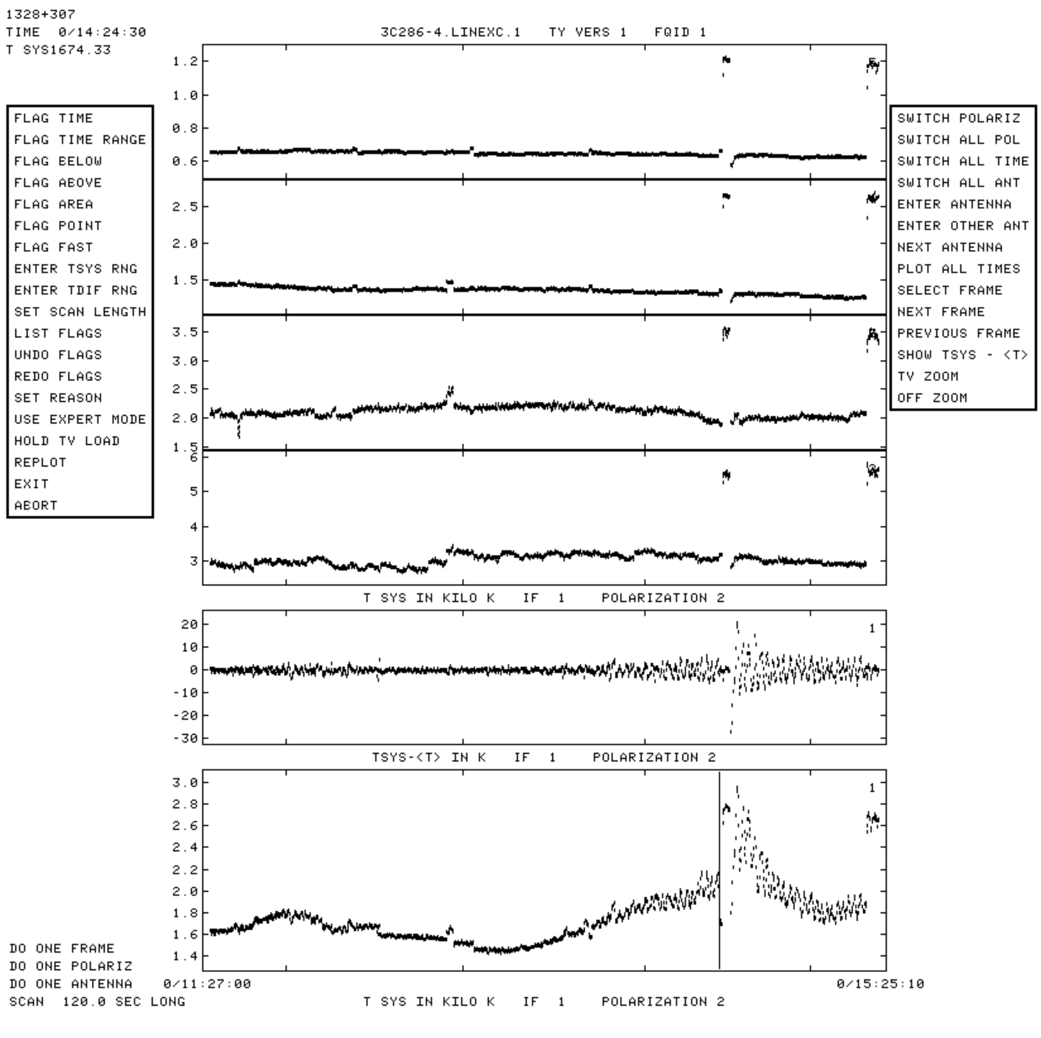

Figure 4.1: A display of a sample TV screen from `EDITA`, made using the *AIPS* task `TVCPS` to produce a negative black-and-white display. The `EDITA` menu (in the boxes), the status lines (at the bottom), the editing area (bottom) of a portion of the data from the selected antenna (1), the subsidiary plots of data from selected secondary antennæ (2, 3, 4, 5), the edit tool (bar or box), and the edit location values are displayed in different graphics planes which normally appear in different colors.

actual *uv* data set's flagging (FG) table until you exit from TVFLG and tell it to do so. During editing, TVFLG does not delete data from its master grid; it just marks the flagged data so that they will not be displayed. This allows you to undo editing as needed during your TVFLG session(s). When the flags are transferred to the main *uv* data set, however, the flagged data in the master grid are fully deleted since undoing the flags at that point has no further meaning. When you are done with a master grid file, be sure to delete it (with ZAP) since it is likely to occupy a significant amount of disk.

TVFLG keeps track of the source name associated with each row of data. When averaging to build the master grid and to build the displayed grids, TVFLG will not average data from different sources and will inform you that it has omitted data if it has had to do so for this reason. For multi-source files, the source name is displayed during the CURVALUE-like sections. However, the flagging table is prepared to flag *all* sources for the specified antennas, times, *etc.* or just the displayed source. If you are flagging two calibrator scans, you may wish to do all source in between as well. Use the SWITCH SOURCE FLAG interactive option to make your selection before you create flagging commands. Similarly, you will need to decide whether flagging commands that you are about to prepare apply only to the displayed channel and/or IF, or to all possible channels and/or IFs. In particular, spectral-line observers often use TVFLG on the pseudo-continuum "channel-0" data set, but want the resulting flags to apply to all spectral channels when copied to the spectral-line data set. They should be careful to select all channels before generating any flagging commands. Each flagging command generated is applied to a list of Stokes parameters, which *does not have to include* the Stokes currently being displayed. When you begin TVFLG and whenever you switch displayed Stokes, you should use the ENTER STOKES FLAG option to select which Stokes are to be flagged by subsequent flagging commands.

If you get some of this wrong, you can use the UNDO FLAGS option in TVFLG if the flags have not yet been applied to the *uv* data set. Or you can use tasks UVFLG, TABED or TAFLG to correct errors written into the FG table of your multi-source *uv* data set. It is rather harder to undo errors if you use TVFLG on a single-source data set since the flagging commands are applied directly (and destructively) to the data. For this reason, we normally recommend that you use TVFLG on multi-source data sets, converting single-source ones with MULTI before running TVFLG.

TVFLG displays the data, for a single IF, channel, and Stokes, as a grey-scale display with time increasing up the screen and baseline number increasing to the right. Thus baselines for the VLA run from left to right as 1–1, 1–2, 1–3, ..., 2–2, 2–3, ..., 27–27, 27–28, and 28–28. An input parameter (DPARM(3) = 1 allows you to create a master grid and display baselines both as, say 1–2 and 2–1. An interactive (switchable) option allows you to order the baselines from shortest to longest (ignoring projection effects) along the horizontal axis.

The interactive session is driven by a menu which is displayed on a graphics overlay of the TV display. An example of this full display is shown on the next page. Move the cursor to the desired operation (noting that the currently selected one is highlighted in a different color on many TVs) and press button A, B, or C to select the desired operation; pressing button D produces on-line help for the selected operation. The first (left-most column) of choices is:

OFFZOOM	turn off any zoom magnification
OFFTRANS	turn off any black & white enhancement
OFFCOLOR	turn off any pseudo-coloring
TVFIDDLE	interactive zoom, black & and white enhancement, and pseudo-color contours as in AIPS
TVTRANSF	black & white enhancement as in AIPS
TVPSEUDO	many pseudo-colorings as in AIPS
DO WEDGE ?	switches choice of displaying a step wedge
LIST FLAGS	list selected range of flag commands
UNDO FLAGS	remove flags by number from the FC table master grid
REDO FLAGS	re-apply all remaining flags to master grid

Note: when a flag is undone, all cells in the master grid which were first flagged by that command are restored to use. Flag commands done after the one that was undone may also, however, have applied to some of those cells. To check this and correct any improperly un-flagged pixels, use the `REDO FLAGS` option. This option even re-does `CLIP` operations! After an `UNDO` or `REDO FLAGS` operation, the TV is automatically re-loaded if needed. Note that the `UNDO` operation is one that reads and writes the full master grid.

Column 2 offers type-in controls of the TV display and controls of which data are to be flagged. In general, the master grid will be too large to display on the TV screen in its entirety. The program begins by loading every $n^{\rm th}$ baseline and time smoothing by m time intervals in order to fit the full image on the screen. However, you may select a sub-window in order to see the data in more detail. You may also control the range of intensities displayed (like the adverb `PIXRANGE` in `TVLOD` inside `AIPS`). The averaging time to smooth the data for the TV display may be chosen, as may the averaging time for the "scan average" used in some of the displays. Which correlators are to be flagged by the next flagging command may be typed in. All of the standard Stokes values, plus any 4-bit mask may be entered. The spectral channel and IF may be typed in. Flagging may be done only for the current channel and IF and source, or it may be done for all channels and/or IFs and/or sources. Note that these controls affect the next `LOAD`s to the TV or the flagging commands prepared after the parameter is changed. When the menu of options is displayed at the top of the TV, the current selections are shown along the bottom. If some will change on the next load, they are shown with an asterisk following. Column 2 contains

ENTER BLC	Type in a bottom left corner pixel number on the terminal
ENTER TRC	Type in a top right corner pixel number on the terminal
ENTER AMP PIXRANGE	Type in the intensity range to be used for loading amplitude images to the TV
ENTER PHS PIXRANGE	Type in the phase range to be used for loading phase images to the TV
ENTER RMS PIXRANGE	Type in the intensity range to be used for loading images of the rms to the TV
ENTER R/M PIXRANGE	Type in the value range to be used for loading rms/mean images to the TV
ENTER SMOOTH TIME	Type in the time smoothing length in units of the master grid cell size
ENTER SCAN TIME	Type in the time averaging length for the "scan average" in units of the master grid cell size
ENTER CHANNEL	Type in the desired spectral channel number using the terminal
ENTER IF	Type in, on the terminal, the desired IF number
ENTER STOKES FLAG	To type in the 4-character string which will control which correlators (polarizations) are flagged. Note: this will apply only to subsequent flagging commands. It should be changed whenever a different Stokes is displayed.
SWITCH SOURCE FLAG	To switch between having all sources flagged by the current flag commands and having only those sources included in this execution of `TVFLG` flagged. The former is desirable when a time range encompasses all of 2 calibrator scans.
SWITCH ALL-CH FLAG	To reverse the flag all channel status; applies to subsequent flag commands
SWITCH ALL-IF FLAG	To reverse the flag all IFs status; applies to subsequent flag commands

The all-channel flag remains true if the input data set has only one channel and the all-IF flag remains true if the input data set has no more than one IF.

An extra word should be said about the "scan average" to which reference was made above. This is used

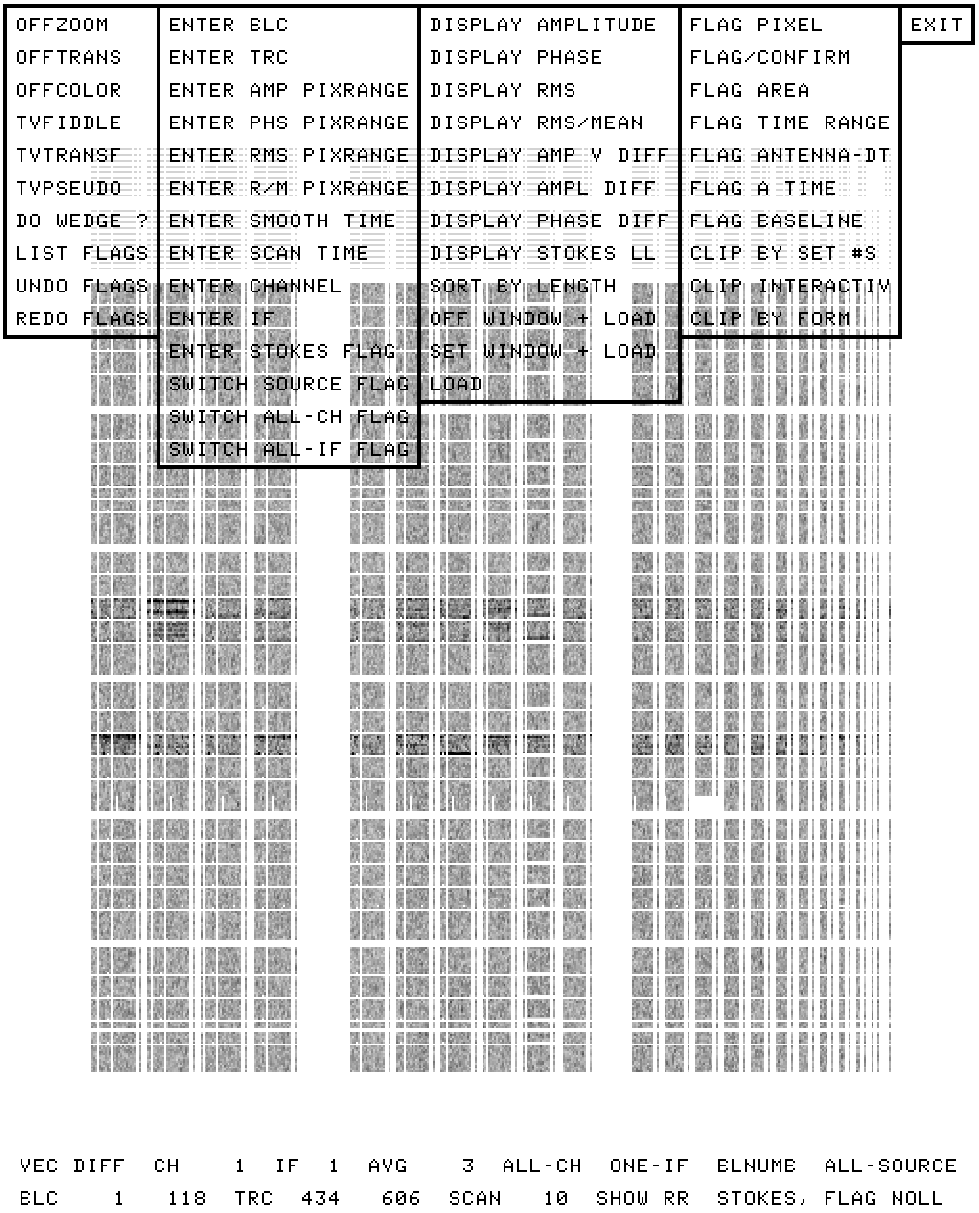

Figure 4.2: A display of a sample TV screen from **TVFLG**, made using the AIPS task **TVCPS** to produce a negative black-and-white display. The **TVFLG** menu (in the boxes) and status lines (at the bottom) are displayed in a graphics plane which is normally colored light green. The data are grey scales in a TV memory and may be enhanced in black-and-white or pseudo-colored. The particular display chosen is the amplitude of the vector difference between the sample and a running vector average of samples surrounding it. This particular parameter is sensitive to both phase and amplitude problems and may save you the extra time of looking at phase and amplitude separately. It requires that there be data to average, but does not blur the flagging by the averaging interval (as the RMS method does). The visibility data are from the VLA. All baselines are shown once only in baseline number order. Antennas 5 and 12 (at least) appear to be missing. The displayed data are the RR Stokes samples and have been windowed to exclude some times. Flag commands generated at the moment illustrated will flag all source names, all spectral channels, only one IF, and all Stokes except LL.

solely for displaying the difference of the data at time T and the average of the data at times near T. This average is computed with a "rolling buffer." Thus, for a scan average time of 30 seconds and data at 10-second intervals, the average for a set of 7 points is as follows:

```
time        average of times
 00            00    10    20
 10            00    10    20
 20            10    20    30
 30            20    30    40
 40            30    40    50
 50            40    50    60
 60            40    50    60
```

The third column of options is used to control which data are displayed and to cause the TV display to be updated. The master grid must be converted from complex to amplitude, phase, the rms of the amplitude, or the rms divided by the mean of the amplitude for display. It may also be converted to the amplitude of the vector difference between the current observation and the "scan average" as defined above or the absolute value of the difference in amplitude with the scalar-average amplitude or the absolute value of the difference in phase with the vector scan average. Furthermore, the baselines may be reordered in the TV display by their length rather than their numerical position. This column has the options:

DISPLAY AMPLITUDE	To display amplitudes on the TV
DISPLAY PHASE	To display phases on the TV
DISPLAY RMS	To display amplitude rms on the TV
DISPLAY RMS/MEAN	To display amplitude rms/mean on the TV
DISPLAY AMP V DIFF	To display the amplitude of the difference between the data and a running (vector) "scan average"
DISPLAY AMPL DIFF	To display the abs(difference) of the amplitude of the data and a running scalar average of the amplitudes in the "scan"
DISPLAY PHASE DIFF	To display the abs(difference) of the phase of the data and the phase of a running (vector) "scan average"
DISPLAY STOKES *xx*	To switch to Stokes type *xx* (where `xx` can be RR, LL, RL, LR, etc as chosen by the `STOKES` adverb).
SORT BY *xxxxxxxx*	To switch to a display with the x axis (baseline) sorted by ordered by `LENGTH` or by `BASELINE` number
OFF WINDOW + LOAD	Reset the window to the full image and reload the TV
SET WINDOW + LOAD	Interactive window setting (like `TVWINDOW`) followed by reloading the TV
LOAD	Reload TV with the current parameters

`SET WINDOW + LOAD` is "smarter" than `TVWINDOW` and will not let you set a window larger than the basic image. Therefore, if you wish to include all pixels on some axis, move the TV cursor outside the image in that direction. The selected window will be shown.

The fourth column is used to select the type of flagging to be done. During flagging, a TV graphics plane is used to display the current pixel much like `CURVALUE` in `AIPS`. Buttons A and B do the flagging (except A switches corners for the area and time-range modes). Button C also does the flagging, but the program then returns to the main menu rather than prompting for more flagging selections. Button D exits back to the menu without doing any additional flagging. Another graphics plane is used to show the current area/time/baseline being flagged. All flagging commands can create zero, one, two, or more entries in the flagging list; hit button D at any time. There are also two clipping modes, an interactive one and one in which the user enters the clip limits from the terminal. In both, the current image computed for the TV (with user-set windows and data type, but not any other windows or alternate pixels etc. required to fit the image on the TV) is examined for pixels which fall outside the allowed intensity range. Flagging commands are prepared and the master file blanked for all such pixels. In the interactive mode, buttons A and B switch between setting the lower and upper clip limits, button C causes the clipping to occur followed by a return to the main menu, and button D exits to the menu with no flagging. The options are

FLAG PIXEL	To flag single pixels
FLAG/CONFIRM	To flag single pixels, but request a yes or no on the terminal before proceeding
FLAG AREA	To flag a rectangular area in baseline-time
FLAG TIME RANGE	To flag all baselines for a range of times
FLAG ANTENNA-DT	To flag all baselines to a specific antenna for a range of times
FLAG TIME	To flag all baselines for a specific time
FLAG BASELINE	To flag all times for a specific baseline
CLIP BY SET #S	To enter from the terminal a clipping range for the current mode and then clip high and low samples
CLIP INTERACTIV	To enter with the cursor and LUTs a clipping range for the current mode and then clip data outside the range.
CLIP BY FORM	To clip selected channels/IFs using the "method" and clipping range of some previous clip operation

The last operation allows you to apply a clipping method already used on one channel/IF to other channels and/or IFs. `CLIP BY FORM` asks for a command number (use `LIST FLAGS` to find it) and applies its display type (amp, phase, rms, rms/mean, differences), averaging and scan intervals and clip levels to a range of channels, IFs and Stokes (as entered from the terminal). To terminate the operation, doing nothing, enter a letter instead of one of the requested channel or IF numbers. To omit a Stokes, reply, if requested for a flag pattern, with a blank line. You may watch the operation being carried out on the TV as it proceeds.

The right-most column has only the option:

EXIT	Go resume `AIPS` and, optionally, enter the flags in the data

Before the flags are entered in the data, `TVFLG` asks you whether or not you actually wish to do this. You must respond yes or no. Note that, if the master grid is to remain cataloged, there is no need to enter the flagging commands every time you decide to exit the program for a while. In fact, if you do not enter the commands, you can still undo them later, giving you a reason not to enter them in the main *uv* data set too hastily.

The two most useful data modes for editing are probably amplitude and amplitude of the vector difference. The former is useful for spotting bad data over longer time intervals, such as whole scans. The latter is excellent for detecting short excursions from the norm. For editing uncalibrated data, rms of two time intervals is useful, but the rms modes require data to be averaged (inside `TVFLG`) and therefore reduce the time resolution accuracy of the flagging. If you edit by phase, consider using the pseudo-coloration scheme that is circular in color (option `TVPSEUDO` followed by button B) since your phases are also circular.

Using `TVFLG` on a workstation requires you to plan the real estate of your screen. We suggest that you place your message server window and your input window side-by-side at the bottom of the screen. Then put the TV window above them, occupying the upper 70–90% of the screen area. (Use your window manager's tools to move and stretch the TV window to fill this area.) Instructions and informative, warning and error messages will appear in the message server window. Prompts for data entry (and your data entry) appear in the input window. Remember to move the workstation cursor into the input window to enter data (such as IF, channel, antenna numbers, and the like) and then to move the cursor back into the TV area to select options, mark regions to be flagged, adjust enhancements, and so on.

4.4.4 Baseline corrections

Sometimes, *e.g.*, during an array re-configuration, your observations may have been made when one or more of the antennas had their positions poorly determined. The positional error is usually less than a centimeter, but even this may affect your data significantly. The most important effect is a slow and erroneous phase wind which is a function of source position and time. Since this error is a function of source position, it cannot be removed exactly using observations of a nearby calibrator, although the error will be small if the target source is close to the calibrator. In many observations, the target sources and calibrators are sufficiently close to allow this phase error to be ignored. Self-calibration will remove this error completely *if* you have enough signal-to-noise to determine the correction during each integration.

The maximum phase error introduced into the calibrated visibility data by incorrect antenna coordinates $\Delta\phi_B$, in radians, by a baseline error of ΔB meters is given by

$$\Delta\phi_B \approx 2\pi\Delta\theta\Delta B/\lambda$$

where $\Delta\theta$ is the angular separation between the calibrator and the target source in radians and λ is the wavelength in meters.

Note, however, that the error due to the phase-wind is not the only error introduced by incorrect antenna positions. A further, but much smaller effect, will be incorrect gridding of the data due to the erroneous calculation of the baseline spatial frequency components u, v and w. This effect is important only for full primary beam observations in which the antenna position error is of the order of a meter. It is highly unlikely that such a condition will occur. Note too, that this error *cannot* be corrected by the use of self-calibration. The maximum phase error in degrees, $\Delta\phi_G$, caused by incorrect gridding of the u,v,w data is

$$\Delta\phi_G \approx 360\Delta\epsilon\Delta\Theta$$

where $\Delta\epsilon$ is the antenna position error in antenna diameters and $\Delta\Theta$ is the angular offset in primary beams.

If baseline errors are significant they need to be removed from your data before calibration. If your observations are affected by antenna positional errors, you should be informed in a covering letter for your observations. This letter, issued only when there are large errors, will advise you of the affected antennas and the time range through which the antenna position corrections must be made. The appropriate corrections are in the covering letter. Since smaller errors are not reported by covering letters, you may wish to check for antenna position corrections using the local utility "`vlais`" on *zia* (`zia.aoc.nrao.edu`). This on-line utility summarizes recent observational details for the VLA and VLBA. It can be invoked by typing (at system level)

`% vlais` C_R

and following the menu. For information on how to get to system level, see Appendix Z. Try the commands `vla`, `bl`, and `bl94` in sequence to see for 1994 the date and time that the antenna was moved to its station, the baseline changes in meters, and the date and time at which the changes were installed. Noting these, and the time of observation, you can determine, using the relationships given above, whether you need to make corrections. If no corrections are required for your data set, go on to the next section.

To apply the antenna position corrections, first copy the first version of the `CL` table to version 2 with task `TACOP`.

> TASK 'TACOP' C_R	
> INDISK *m* ; GETN *n* C_R	to select the data set, $n = 3$ and $m = 1$ above.
> CLRONAME C_R	to copy to the input file by default.
> INEXT 'CL' ; INVERS 1 ; OUTVER 2 C_R	to copy `CL` table version 1 to version 2.
> NCOUNT 1 C_R	copy only one version.
> KEYWO ' ' ; KEYVAL 0 ; KEYSTR ' ' C_R	these adverbs not required here.

> GO	to run task TACOP.

This will create a higher version of the CL table 1 and append it to your multi-source data set. Next, use CLCOR to enter the antenna position corrections (in meters) in the new version of the CL table. This must be done for each affected antenna in turn.

> TASK 'CLCOR' C_R	
> INDISK *m* ; GETN *n* C_R	to get the correct data set. Note that you don't have to keep doing this unless you switch between different input data files.
> SOURCES ' ' ;STOKES ' ' C_R	to do all sources, all Stokes,
> BIF 0 ; EIF 0 C_R	and all IFs.
> SUBARRAY *x* C_R	to choose the correct sub-array.
> OPCODE 'ANTP' C_R	to select the antenna position correction mode.
> GAINVER 2 C_R	choose the correct version of the CL table to modify.
> ANTENNA *k* C_R	to select antenna.
> CLCORPRM $\Delta b_x, \Delta b_y, \Delta b_z, 0, 0, 0, 1$ C_R	to add the appropriate antenna corrections in meters; the 1 in CLCORPRM(7) indicates VLA phase conventions rather than VLB conventions.
> GO C_R	to run CLCOR.

The program will need to be run as many times as there are antennas for which positional corrections must be made. Note that subsequent calibration must be applied to CL table 2 to create higher versions of the calibration table. This new CL table (version 2) will replace version 1 in all of the subsequent sections on calibration. Thus, in subsequent executions of CALIB, you must apply these corrections by specifying DOCALIB TRUE ; GAINUSE 2 (or higher).

4.5 Antenna-based complex gain solutions

At this point, we assume that you have removed the worst of the bad calibrator data (if any) and have run CALIB over as large a UVRANGE as possible for each calibrator. The resulting gain tables can be brought to a consistent amplitude scale, bootstrapping the unknown fluxes of the secondary calibrators. Final pass(es) of CALIB are done if needed and then the solution tables are merged into a full calibration (CL) table.

4.5.1 Bootstrapping secondary flux-density calibrators

Task GETJY can be used to determine the flux density of the secondary flux calibrators from the primary flux calibrator based on the flux densities set in the SU table and the antenna gain solutions in the SN tables. The SU and SN tables will be updated by GETJY to reflect the calculated values of the secondary calibrators' flux densities. This procedure should also work if (incorrect) values of the secondary calibrators' flux densities were present in the SU table when CALIB was run. Bad or redundant SN tables should be deleted using EXTDEST before running GETJY, or avoided by selecting tables one at a time with adverb SNVER.

To use GETJY:

> TASK 'GETJY' ; INP C_R	
> SOURCES '*cal*1' , '*cal*2' , '*cal*3' ... C_R	to select secondary flux calibrators.
> CALSOU '3C286' , ' ' C_R	to specify primary flux calibrator(s).
> CALCODE ' ' C_R	to use all calibrator codes.
> BIF 1 ; EIF 2 C_R	to do both IFs.
> FREQID 1 C_R	to use FQ number 1.

> ANTENNAS 0 C_R	to include solutions for all antennas.
> TIMERANG 0 C_R	to include all times.
> SNVER 0 C_R	to use all SN tables.
> INP C_R	to review inputs.
> GO C_R	to run the task when the inputs are okay.

GETJY will give a list of the derived flux densities and estimates of their uncertainties. If any of the uncertainties are large, then reexamine the SN tables as described above and re-run CALIB and/or GETJY as necessary. Multiple executions of GETJY will not cause problems as previous solutions for the unknown flux densities are simply overwritten.

4.5.2 Full calibration

Once you have determined the flux densities of all your gain calibrators, you are ready to complete the first pass of the calibration. At this point, many observers take a conservative viewpoint and delete their existing SN table(s) with

> INEXT 'SN' C_R	to specify the SN table.
> INVERS -1 C_R	to delete all versions.
> EXTDEST C_R	to do the deletion.

This step forces you to re-run CALIB for all your gain calibration sources and is not required if the previous bootstrapping calibrations included all antennas and most correlators, for these calibrators.

Procedure VLACALIB may be used for your gain calibration sources as you did previously.

> INDI *n* ; GETN *m* C_R	to select the data set, $n = 3$ and $m = 1$ above.
> CALSOUR = '3C286' , ' ' C_R	to name primary flux calibrator(s) individually, or
> CALCODE = '*' C_R	to select any non-blank calibrator code.
> UVRANGE *uvmin uvmax* C_R	*uv* limits, if any, in kiloλ.
> ANTENNAS *list of antennas* C_R	antennas to use for the solutions, see discussion above.
> REFANT *n* C_R	reference antenna number.
> MINAMPER 10 C_R	display warning if baseline disagrees in amplitude by more than 10% from the model.
> MINPHSER 10 C_R	display warning if baseline disagrees by more than 10 deg of phase from the model.
> FREQID 1 C_R	use FQ number 1.
> INP VLACALIB C_R	to review inputs.
> VLACALIB C_R	to make the solution and print results.

If there are different *uv* ranges for different sources, then re-run the procedure with changed parameters, such as:

> CALSOUR = '*cal1*' , '*cal2*' , '*cal3*' C_R	to name secondary flux calibrator(s).
> ANTENNAS 0 C_R	solutions for all antennas.
> UVRANGE 0 C_R	no *uv* limits, or range if any, in kiloλ.
> INP VLACALIB C_R	to review inputs.
> VLACALIB C_R	to process the secondary calibrators.

At this time, you should use as many antennas and as large a UVRANGE as you can for each calibrator, consistent with its spatial structure.

4.5.3 Final (?) initial global calibration

At this point you should have gain and phase solutions for the times of all calibration scans, including the correct flux densities for the secondary calibrators. The next step is to interpolate the solutions derived from the calibrators into the CL table for all the sources. CLCAL may be run multiple times if subsets of the sources are to be calibrated by corresponding subsets of the calibrators, unless you limit it to one table with SNVER, CLCAL assumes that all SN tables contain only valid solutions and concatenates all of the SN tables with the highest numbered one. Therefore, any bad SN tables should be removed before using CLCAL. For polarization calibration, it is essential that you calibrate the primary flux calibrator (3C48 or 3C286) also so that you can solve for the left minus right phase offsets and apply PCAL.

To use CLCAL:

> TASK CLCAL ; INP C_R	to review the inputs.
> SOURCES '*sou1*' , '*sou2*' , '*sou3*' , ... C_R	sources to calibrate, ' ' means all.
> CALSOUR '*cal1*' , '*cal2*' , '*cal3*' , ... C_R	calibrators to use for SOURCES.
> FREQID *n* C_R	use FQ number *n*.
> OPCODE 'CALI' C_R	to combine SN tables into a CL table.
> GAINVER 1 C_R	to select the input CL table; 1 for first calibration, 2 if there are baseline corrections.
> GAINUSE 2 C_R	to select the output CL table; 2 is normal, 3 if there are baseline corrections.
> REFANT *m* C_R	to select the reference antenna; needed only if REFANT reset since CALIB was run.
> INTERP '2PT' C_R	for linear interpolation, no smoothing, *or*
> INTERP 'BOX' C_R	for boxcar smoothing, then interpolation.
> INTPARM *n* C_R	for *n*-hr smoothing if BOX selected.
> INP C_R	to check inputs.
> GO C_R	to run CLCAL.

Calibrator sources may also be selected with the QUAL and CALCODE adverbs; QUAL also applies to the sources to be calibrated. Note that REFANT appears in the inputs because *AIPS* references all phases to those of the reference antenna. If none is given, it defaults to the one used in the most solutions.

Note that CLCAL uses both the GAINUSE and GAINVER adverbs. This is to specify the input and output CL table versions, which should be different. CL table version 1 is intended to be a "virgin" table, free of all injury from any calibration you do using the *AIPS* package. It may not always be devoid of information, as "on-line" corrections may be made and recorded here by some telescope systems, *e.g.*, the VLBA. (The VLA, through task FILLM, currently puts no special information in this file.) CLCAL and most other *AIPS* tasks are forbidden to over-write version 1 of the CL table. This protects it from modification, and keeps it around so that you may *reset* your calibration to the raw state by using EXTDEST to destroy all CL table extensions with versions higher than 1. Be careful doing this, since you rarely want to delete CL version 1. (All past versions of AIPS have allowed you to do this, but, beginning with the 15JUL94 release, AIPS will ask for special confirmation before allowing you to delete CL version 1.) Should you destroy CL table version 1 accidentally, you may generate a *new CL* table version 1 with the task INDXR. This new *CL* table will not contain any on-line calibration, however. If you have made baseline corrections — or any of the many other sorts of corrections allowed by CLCOR — then you will probably want to protect (and use for input) CL table version 2 as well. In that case, GAINVER = 3 is recommended.

If you have any reason to suspect that the calibration has gone wrong — or if you are calibrating data for the first time — you should examine the contents of the output CL table. LISTR with OPTYPE = 'GAIN' will print out the amplitudes and phases in the specified CL or SN table. Note that these tables can be vary large. Use the SOURCES and TIMERANG adverbs to limit the output, or look at it on your terminal (DOCRT

= 132) so that you can stop the display whenever you have had enough. Task **SNPLT** will provide you with a graphical display which may be easier on the eye.

The most important step in the calibration is your verification that everything has gone according to plan. To check this, you should produce matrix listings for all your calibrator sources. For simplicity in interpretation, limit each listing to the **UVRANGE** to which you limited the calibrator during calibration. Thus:

> TASK 'LISTR' C_R	
> DOCRT -1 C_R	to direct output to the printer.
> SOURCES '*cal1*' , '*cal2*' , '*cal3*' , . . . C_R	to list all selected calibrators by name.
> UVRANGE *uvmin uvmax* C_R	*uv* limits, if any, in kiloλ.
> OPTYP 'MATX' C_R	to get the matrix form of listing.
> DOCALIB TRUE C_R	to list with calibration applied.
> GAINUSE 2 C_R	or 3, to point to the *new* gain table.
> FREQID *n* C_R	list data for **FQ** *n*.
> DPARM = 5 , 1 , 0 C_R	to have amplitude and phase using scalar scan averaging.
> BIF 1 C_R	to specify the "AC" IFs; only one can be done at a time.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.
> BIF 2 C_R	to specify the "BD" IFs.
> WAIT ; GO C_R	to run the program again after the first job is done.

The matrix average amplitudes for the calibrators in this listing should be very close to the values that you entered with **SETJY** (or which were derived by **GETJY**) and the phases in all rows and columns for these sources should be very close to zero.

If some rows and columns of the amplitude matrices are systematically different from the mean, the amplitude calibration for the associated antennas is imperfect. The reasons for this should be investigated. More flagging of visibilities, scans, or antennas, may be indicated. If the phase matrices have all elements near zero, then the phase calibration is in good shape. If some calibrators have discrepant phases and others do not, the discrepant calibrators are probably resolved. Note that you will not be able to detect errors in the assumed positions of your calibrators at this stage if you have used the usual 2-point interpolation of the calibration. Position errors in the calibrators have now become phase and position errors in the target sources.

If the previous steps indicate serious problems and/or you are seriously confused about what you have done and you want to start the calibration again, you can use the procedure **VLARESET** from the RUN file **VLAPROCS** to reset the **SN** and **CL** tables.

> INP VLARESET C_R	to verify the data set to be reset.
> VLARESET C_R	to reset **SN** and **CL** tables.

4.6 Polarization calibration

The calibration of visibility data sensitive to linear polarization involves two distinct operations: (1) determining and correcting the data for the effects of imperfect telescope feeds and (2) removing any systematic phase offsets between the two systems of orthogonal polarization. These two components of polarization calibration will be considered separately.

The effective feed response is parameterized most generally by its polarization ellipticity and the orientation of the major axis of that ellipse. For the VLA, it appears to be adequate to make the simpler assumption that each polarization is corrupted by a small complex gain times the orthogonal polarization.

In general, the polarization of the calibrator(s) to be used to determine the feed parameters will not be known *a priori* and must be determined along with the feed parameters. Observations of a given source (or sources) over a wide range ($\geq$ 90 deg) of parallactic angles is necessary to separate calibrator polarization from the feed parameters. Task LISTR may be used to determine the parallactic angles at which data have been taken:

> TASK 'LISTR' C_R	
> SOURCES '*cal1*' , '*cal2*' , '*cal3*' , ... C_R	list all calibrators to be used.
> INEXT 'CL' C_R	to determine parallactic angle at times in CL table.
> INVER 1 C_R	CL version 1.
> FREQID *n* C_R	to use FQ number *n*.
> OPTYPE 'GAIN' C_R	to use gain table rather than visibility data.
> DPARM = 9 , 0 C_R	to display parallactic angle.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

Multiple calibrators may be used in determining the feed polarization, but the data from them must be accurately calibrated. In particular, the phase calibration of any calibrator used to determine antenna polarizations should be determined from that calibrator itself (*i.e.*, the source should be self-calibrated). Note that this will normally have occurred for all gain calibrators if the procedure described in the previous sections was followed.

The normal phase calibration technique treats parallel-hand visibilities in the two orthogonal polarizations independently. Thus, there will be a systematic phase difference between the two polarizations systems. This difference may be due to differences in instrumental phase offset for the two systems or due to the propagation medium (*i.e.*, Faraday rotation) or both. Faraday rotation effects are particularly bothersome as they may be time variable and increase rapidly with wavelength. For data at L band or longer wavelengths, *AIPS* should be given an estimate of the ionospheric Faraday rotation measure using task FARAD. This task computes the ionospheric rotation measure using either total electron content from a nearby ionospheric monitoring station (Boulder Colorado for the VLA) or an empirical model that uses the monthly mean Zurich sunspot number (R1) as a measure of solar activity. If monitoring data are available, they should be used in preference to the model. FARAD enters the ionospheric Faraday rotation measure into the CL table. This is used by PCAL when determining antenna polarization parameters and is used by other calibration tasks to de-rotate the data when polarization corrections are applied. FARAD may be run any number of times with different parameters before PCAL is run; each run of FARAD over-writes the values written by the previously.

Data from Boulder for the total electron content for year *mm* is contained in the file TECB.*mm* in the directory with logical name AIPSIONS. Currently, data are available for 1980 onwards through part of 1992. Some gaps in the TEC data occur, particularly for years 1980 and 1988 and for times when solar activity has been high and no reliable estimate of the total electron content could be made. Unfortunately, we are no longer able to get the TEC data from Boulder. If your VLA data are more recent than early 1992, you can consider using the ionospheric model in FARAD, but you should be aware that the model is crude and should check that it improves matters before using it in your final calibration.

The phase offsets between the right-hand and left-hand polarizations at a given time may be determined using, from a source with a known angle of linear polarization, data which have had the effects of imperfect feeds removed. The phase of the right-left correlations or the conjugate of the left-right correlations indicates the phase difference between the two polarizations.

The (initial) need for ionospheric corrections can be bypassed if either (1) you use unpolarized sources in PCAL, and/or (2) the ionosphere was well behaved during your observations. Typical rotation measures are only a few and therefore affect only L and longer-wavelength bands. The ionosphere is almost always well enough behaved to be ignored at shorter wavelengths and is usually able to be ignored even at L band.

Changes of around 10 deg in the relative phases of R and L polarizations are not enough to disrupt a `PCAL` solution seriously. And, fortunately, calibrators at long wavelengths, such as P band, tend to be unpolarized. In general, if the ionosphere is well behaved, it can be ignored. If it is bad, no simple model is able to correct it and you may simply have to forget about polarization for that observing run. Note that an apparent position angle variation of 3C286 with time probably indicates that ionospheric rotation is significant. But, if 3C286 shows large rotations, it does not follow that *its* rotation can be applied to other directions in the sky. All it implies reliably is that a *model* is needed.

Polarization calibration may be performed on amplitude- and phase-calibrated VLA data using the following five-step procedure:

Step 1: Run `PCAL` on one or more phase calibrator sources observed with a wide range of parallactic angles:

> TASK 'PCAL' C_R	
> CALSOUR '*cal1*' , '*cal2*' , '*cal3*' , ... C_R	list all calibrators to be used.
> TIMERANG 0 C_R	to use all times.
> ANTENNAS 0 C_R	to solve for all antennas.
> UVRANGE *uvmin uvmax* C_R	to set *uv* limits, if any, in kiloλ.
> BIF 1 ; EIF 2 C_R	to do both IFs.
> DOCALIB TRUE C_R	to apply the calibration to the sources (very important!).
> GAINUSE 2 C_R	or 3 or whatever, to use the latest `CL` table.
> CLR2N C_R	to clear `IN2NAME` *etc.* since there is no Clean-image model.
> FREQID *n* C_R	to use `FQ` value *n*; only one polarization solution can be stored.
> PMODEL 0 C_R	to use polarization parameters in the source table.
> SOLINT 2 C_R	to use a 2-minute solution interval; scan averages are usually sufficient.
> SOLTYPE 'APPR' C_R	to use linear approximation model.
> PRTLEV 1 C_R	to display the results and some diagnostic information.
> REFANT n C_R	only if `REFANT` reset since `CALIB` run.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

`PCAL` will list the fitted values of the antenna polarization parameters and the source polarizations with estimates of the uncertainties. If these results do not appear reasonable (*e.g.*, large errors or large corrections or inconsistent solutions for the calibrator polarizations at neighboring frequencies), more editing and a rerun of `PCAL` may be necessary. `PCAL` puts the derived source polarizations in the `SU` table and the antenna feed values in the `AN` table. These values may be examined later with `PRTAN` and `PRTAB`.

Step 2:. Use `LISTR` to determine the apparent right minus left phase angle of the polarization calibrator source, *e.g.*, 3C286 or 3C138:

> TASK 'LISTR' C_R	
> SOURCE '3C286' , ' ' C_R	to view only the polarization angle calibrator.
> TIMERANG 0 C_R	to check all times.
> ANTENNAS *list of antennas* C_R	antennas to use; the list used for `CALIB`.
> UVRANGE *uvmin uvmax* C_R	to limit *uv*, if appropriate.
> BIF 1 C_R	to view the first ("AC") IF.
> FREQID *n* C_R	to view the current `FQ` value (*n*).
> DOCALIB TRUE C_R	to list with calibration applied.
> DOPOL TRUE C_R	to correct for feed polarization and Faraday rotation.
> GAINUSE 2 C_R	to use the latest `CL` table.
> OPTYPE 'MATX' C_R	to get matrix listing.

> STOKES 'POLC' C_R	to display cross-polarized data.
> DPARM 1,0 C_R	to display phases.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.
> BIF 2 C_R	to view the second ("BD") IF.
> WAIT ; GO C_R	to wait for the first display and then run the second.

The matrix of scan-averaged averaged right minus left phase angles (actually RL and conjugate of LR polarizations) will be printed. Check that none of the phases differ from the mean by more than a few degrees. If any do, then use `UVFLG` to edit these data and go back to step 1. After the matrix of phases, the average over the matrix of the right minus left phases is displayed. This is the number to be used in step 4.

This method will fail if the calibrator source (3C286 or 3C138, usually) is heavily resolved and the atmospheric phase stability is poor. (These two are frequently coupled!) Under these conditions, the self-calibration of the calibrator will have failed and will have to be done especially for the polarization calibration. In the steps below, you may safely relax the *uv* limits by about 20%, but should solve only for phases using `SOLMODE = 'P'`. The process consists of:

2.1 Apply `CALIB` to the inner (short-baseline) antennas on the calibrator source using the rules in the table found in § 4.3.3 but relaxed a bit. Set DOCALIB = 1 ; GAINUSE = 2 ; SOLMODE = 'P'.

2.2 Use `CLCAL` to apply these solutions to the calibrator source using GAINVER = 2 ; GAINUSE = 3.

2.3 Run `LISTR` for cross-hand phases using *only* the antennas used with `CALIB`.

2.4 Use `EXTDEST` to delete `CL` table 3, a most important step.

After correcting the calibration, repeat steps 2.1 and 2.2 and the special calibration until satisfactory results are obtained.

Step 3: Use `TASAV` to copy all your table files to a dummy *uv* data set, saving in particular the `CL` table with the results of the amplitude and phase calibration. This step is not essential, but it reduces the magnitude of the disaster if the the next step is done incorrectly. (Note - this may be a good idea at several stages of the calibration process!)

> TASK 'TASAV' C_R	
> CLRO C_R	Use default output file file name.
> INP C_R	to review the (few) inputs.
> GO C_R	to run the program.

The task `TACOP` may be used to recover any tables that get trashed during later steps. The notes below also assume that we make a copy of the current `CL` table, so as not to destroy it. Thus,

> TASK 'TACOP' C_R	
> CLRO C_R	To use the default output file, which is the input file.
> INEXT 'CL' C_R	To copy `CL` table.
> INVER 2 C_R	Copy table 2.
> NCOUNT 1 C_R	Copy only one table.
> OUTVER 3 C_R	Copy to 3.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

Step 4: The right minus left phase offset corrections are made using task `CLCOR`. The phase offset correction is the expected value (*twice* the source polarization angle) minus the observed phases from step 2. The expected value is 66 degrees for 3C286, -18 for 3C138 (at L band, perhaps -24 at higher frequencies), and -140 for 3C48 (at 6-cm or shorter wavelengths).

> TASK 'CLCOR' C_R	
> SOURCE ' ' ; ANTENNAS 0 C_R	to correct all sources and all antennas.
> TIMERANG 0 C_R	to correct all times.
> BIF 1 ; EIF 2 C_R	to correct both IFs.
> FREQID *n* C_R	to correct only the current FQ value.
> GAINVER 3 C_R	to modify the CL table just produced by TACOP.
> OPCODE 'POLR' C_R	to do right minus left phase offset correction.
> STOKES 'L' C_R	correction applied to left circular polarization.
> CLCORPRM $cor1, cor2$ C_R	to set the phase correction for "AC" and "BD" IFs.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

This will cause CLCOR to apply appropriate corrections to the CL and AN tables. If the CL table becomes hopelessly corrupted, delete it and return to Step 3. If the AN table is corrupted, then PCAL must be re-run. If more than one CL table needs to be corrected, use the OPCODE='POLR' option only once; other CL tables must be corrected using OPCODE='PHAS' and correcting 1 IF at a time. CLCOR (with OPCODE = 'POLR') may be applied multiple times to the *same* CL table, in order to get the R-L phases "right." But you must *not* apply CLCOR in succession to different CL tables of the same database. If there is any doubt, rerun PCAL. The best way to judge if all is well in the final polarization solution is to look at the spread in the cross-hand phases for 3C286 or 3C138 (step 5 below). If the spread ("eyeball rms") is less than 3 degrees, then all is well. If more than ten, then there is definitely something wrong.

Step 5: Use LISTR to verify the polarization corrections:

> TASK 'LISTR' C_R	
> SOURCE '*cal1*' , '*cal2*' , . . . C_R	to list the calibrators to be checked.
> TIMERANG 0 C_R	to display all times.
> ANTENNAS *list of antennas* C_R	to list the antennas to use.
> UVRANGE *uvmin uvmax* C_R	to set *uv* limits, if appropriate.
> BIF 1 C_R	to list the first ('AC") IF.
> FREQID *n* C_R	to use the current FQ value (*n*).
> DOCALIB TRUE C_R	to list with calibration applied.
> DOPOL TRUE C_R	to correct for feed polarization and Faraday rotation.
> GAINUSE 3 C_R	to use CL table written by CLCOR.
> OPTYPE 'MATX' C_R	to get matrix listing.
> STOKES 'POLC' C_R	to display cross-polarized data.
> DPARM 1,0 C_R	to display phases.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.
> BIF 2 C_R	to list the second ("BD") IF.
> WAIT ; GO C_R	to wait for the first job and then list other IF.

Note well: all of this calibration process must be done with only one FQ at a time. PCAL with FQID = 2 will over-write solutions done for any other FQID.

The phases produced should be consistent. Significant deviations of the phase may indicate that further editing is needed or that residual atmospheric phase errors are still present. If this display appears okay, then the polarization corrections may be applied in SPLIT (see below) by specifying DOPOL = 1 when applying the calibration to produce single-source files.

4.7 Spectral-line calibration

The calibration of spectral-line data is very similar to that of continuum data with the exception that the antenna gains have to be determined and corrected as a function of frequency as well as time. The model used by *AIPS* is to determine the antenna gains as a function of time using a pseudo-continuum ("channel-0") form of the data. Then the complex spectral response function ("bandpass") is determined from observations of one or more strong continuum sources at or near the same frequency as the line observation. In general, the channel-0 data are calibrated using the recipes in the previous sections of this chapter. The sub-sections below are designed to bring out the few areas in which spectral-line calibration differs from continuum.

4.7.1 Reading the data

If your data are on a VLA archive tape then they should be read into *AIPS* using `FILLM`, as described in § 4.1.1. `FILLM` will fill a typical line observation into two files, a large one containing the line data only, and a smaller file containing the "channel-0" data. The standard calibration and editing steps are performed on channel 0 and the results copied over to the line data set. *You must be careful with the tolerance you allow* `FILLM` *to use in determining the* `FQ` *numbers. If you desire all of your data to have the same* `FQ` *number, so that you can calibrate it all in one pass, then set* `CPARM(7)` *in* `FILLM` *to an appropriately large value.*

By default for the VLA, the channel-0 data are generated by the vector average of the central 3/4 of the observing band. If this algorithm is not appropriate for your data, you may generate your own channel-0 data set by averaging only selected channels. To do this, use the task `AVSPC`:

> TASK 'AVSPC' C_R	
> INDI *n* ; GETN *m* C_R	to specify line data set.
> OUTDI *i* ; OUTCL 'CH 0' C_R	to specify output "channel-0" data set disk and class.
> CHANSEL 10, 30, 1, 31, 55, 2 C_R	for example, to average every channel between 10 and 30 and every other channel between 31 and 55.
> GO C_R	to create a new channel-0 data set.

You might find this necessary when observing neutral hydrogen at galactic velocities. Most calibrator sources have some absorption features at these frequencies.

4.7.2 Editing the data

You should follow the steps outlined in § 4.4 to edit the calibrator data using the channel-0 data set. Even though channel-0 data is continuum, be careful to have `TVFLG` and `UVFLG` generate the flagging commands for all channels, not just channel 1. Then, copy the resulting `FG` table to the line file. Use `TACOP`:

> TASK 'TACOP' C_R	
> INDI *n* ; GETN *m* C_R	to specify channel-0 data set.
> OUTDI *i* ; GETO *j* C_R	to specify the line data set.
> INEXT 'FG' C_R	to copy the `FG` table.
> INVER 1 C_R	to copy table 1.
> NCOUNT 1 C_R	to copy only one table.
> OUTVER 1 C_R	to copy it to output table 1
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

Specifying the "`ALL-CH`" setting in `TVFLG` and specifying BCHAN 1 ; ECHAN 0 C_R in `UVFLG` cause all channels to be flagged when the `FG` table is copied to the line data set.

Spectral-line observers should also use **SPFLG** (§ 10.2.2) to examine and, perhaps, to edit their data. This task is very similar to **TVFLG** described in § 4.4.3, but **SPFLG** displays spectral channels on the horizontal axis, one baseline at a time. If you have a large number of baselines, as with the VLA, then you should examine a few of the baselines to check for interference, absorption (or emission) in your calibrator sources, and other frequency-dependent effects. Use the **ANTENNAS** and **BASELINE** adverbs to limit the displays to a few short spacings and one or two longer ones as well. If there are serious frequency-dependent effects in your calibrators, use **SPFLG** and **UVFLG** to delete them. (You might wish to delete the **FG** table with **EXTDEST** to begin all over again.) Then use **AVSPC** to build a new channel-0 data set and repeat the continuum editing. Note that you should not copy the **FG** table from the spectral-line data set to the new continuum one. The reason for this is the confusion over the term "channel." If you have flagged channel 1, but not all channels, in the spectral-line data set — a very common occurrence — then a copied **FG** table would flag all of the continuum data since it has only one "channel." When you have flagged the channel-0 data set, you can merge the new flags back into the spectral-line **FG** table with task **TABED**.

> TASK 'TABED' C_R	
> INDI *n* ; GETN *m* C_R	to specify channel-0 data set.
> OUTDI *i* ; GETO *j* C_R	to specify the line data set.
> INEXT 'FG' C_R	to copy the **FG** table.
> INVER 1 C_R	to copy table 1.
> OUTVER 1 C_R	to copy it to output table 1.
> BCOUNT 1 ; ECOUNT 0 C_R	to copy from the beginning to the end.
> OPTYPE 'COPY' C_R	to do a simple copy appending the input table to the output table.
> TIMER 0 C_R	to copy all times.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

If the channel-0 data set is meaningful for your program sources, you might consider doing a first-pass editing of them along with your calibrators before copying the **FG** table back to the line data set. If your program sources contain significant continuum emission, then this is a reasonable operation to perform. If they do not, then the standard channel-0 data set is not useful for editing program sources. You can use **SPFLG** to edit all channels, or if the signal is strong in a few channels, you could run **TVFLG** on those channels from the spectral-line data set or average those channels alone to a special "channel-n" data set.

4.7.3 Bandpass calibration

The task **BPASS** is designed to take visibility data from specified calibrator(s) to determine the antenna-based complex bandpass functions. It does this in a manner analogous to self-calibration in that the data are divided by a source model or the so-called "channel 0" before the antenna gains are determined as a function of frequency. These are written to a BandPass (**BP**) table. The bandpass calibration is the first operation that should be performed on the line data. So long as one uses the mode in which the data are divided by the so-called "channel 0," it is not necessary to calibrate the data before estimating the bandpasses.

> TASK 'BPASS' C_R	
> INDI *i* ; GETN *j* C_R	to specify line data set.
> CALSOUR '*cal1*' , '*cal2*' , . . . C_R	to specify bandpass calibrators.
> FREQID 1 C_R	to select which **FQ** value to use.
> ANTENNAS 0 C_R	to solve for all antennas.
> REFANT *n* C_R	to set the reference antenna number.
> DOCALIB FALSE C_R	to avoid applying calibration.

> BCHAN 1 ; ECHAN 0 CR	to use all channels.
> BPASSPRM 0 CR	to turn off all "parameters."
> BPASSPRM(5) 0 CR	to divide by channel 0 before determining antenna-based bandpasses
> IN3DI *a* ; GET3N *b* CR	to specify the channel 0 data file, or
> CLR3NAME CR	to have channel 0 found from the input data themselves.
> CHANSEL 20 50 1	to use the average of all channels from 20 through 50, for example, to determine channel 0, when the third input file name is empty.
> FLAGVER 1 CR	to apply flag table 1.
> SOLINT 0 CR	to use scan averages.
> BPVER 1 CR	to select the output `BP` table number.
> INP CR	to review the inputs.
> GO CR	to run the program when inputs set correctly.

The spectral quality of the final images has been found to be determined in part by the quality of the bandpass solutions. In particular, for reasons which are not yet known, the bandpasses are not exactly antenna dependent especially in the edge channels. This "closure error" may be measured in individual and statistical ways by `BPASS` and reported to you. To check on this problem for your data set, set

> MINAMPER *a* CR	to count and, if `BPASSPRM(2)`>1, to report amplitude closure failures $> a$ per cent. Note that closure errors are accumulated as logarithms so that 0.5 and 2.0 are both errors of 100%.
> MINPHSER *p* CR	to count and, if `BPASSPRM(2)`>1, to report phase closure failures $> p$ degrees.
> BPASSPRM(2) 1 CR	to report statistics of amplitude and phase closure failures without reporting individual failures.
> BPASSPRM(6) *a* CR	to report all channels in which the average amplitude closure error $> a$ per cent.
> BPASSPRM(7) *p* CR	to report all channels in which the average phase closure error $> p$ degrees.

It is probably a good idea to set `MINAMPER` and `MINPHSER` fairly high (*i.e.*, 20 and 12) to make a big deal only about major excursions, but to set `BPASSPRM(6)` and `(7)` fairly low (*i.e.*, 0.5 and 0.5) to view the spectrum of closure errors (which will look a lot like the spectrum of noise on your final Clean images). There is even a task called `BPERR` which will summarize and plot the error reports generated by `PBASS` and written to text files by `PRTMSG`.

The bandpass solutions are calculated at each bandpass calibrator scan. As a consequence, they are likely to be unevenly spaced in time and may even have times (due to on-line or later editing) at which there are solutions for some IFs and polarizations but not all. When the latter happens, program source data will be lost unless the missing solutions are filled in. The task `BPSMO` may be used for this purpose or to create a new `BP` table at regular time intervals using one of a number of time-smooting functions. Set `APARM(4) = -1` for the "repair" mode or set `APARM(4)` to the desired `BP` interval.

After the bandpasses have been generated, you can examine them using tasks `BPLOT` and `POSSM`. You can obtain an average from all antennas with

> TASK 'POSSM' CR	
> INDI *i* ; GETN *j* CR	to specify the line data set.
> SOURCES '*cal1*' , '*cal2*' , ... CR	to specify the bandpass calibrators.
> ANTENNAS 0 CR	to include all antennas.
> TIMER 0 CR	to average over all times.

> BCHAN 1 ; ECHAN 0 C_R	to display all channels.
> BPVER 1 C_R	to select the BP table.
> FREQID 1 C_R	to set the FQ value to use.
> APARM 0 C_R	to do a scalar average and have the plot self-scaled and labeled in channels.
> APARM(8) 2 C_R	to plot BP table data.
> NCOUNT 0 C_R	to make one plot only, averaging all included data.
> INP C_R	to review the inputs — check closely.
> GO C_R	to run the program when inputs set correctly.
> GO LWPLA C_R	to send the plot to the (PostScript) printer/plotter.

To view each antenna individually, using the TV to save paper

> DOTV TRUE C_R	to use the TV.
> NCOUNT 1 C_R	to plot one antenna per page/screen.
> GO C_R	to display the bandpasses, averaged over time, on the TV with one antenna per screen.

POSSM shows each screen for 30 seconds before going ahead. You can speed it up by hitting TV buttons A, B, or C, or tell it to quit by hitting button D. If DOTV = -1, then POSSM makes multiple plot extension files, which can be sent to the printer (individually) by LWPLA. You might want to use a larger value of NCOUNT to reduce the number of executions of LWPLA and pieces of paper.

BPLOT is used to create one or more plots (on the TV or in plot files) of the selected bandpass table. The plots will be a set of profiles separated on the vertical axis by an increment in time or antenna number (depending on the sort selected). More than one plot for more than one antenna or more than one time may be generated. Multiple IFs and polarizations will be plotted along the horizontal axis if they are present in the BP table and selected by the adverbs. Thus, BPLOT is useful for plotting the change in bandpass shape as a function either of time or of antenna.

The BP tables are applied to the data by setting the adverb DOBAND $>$ 0 and selecting the relevant BP table with the adverb BPVER. There are three modes of bandpass application. The first (DOBAND 1) will average all bandpasses for each antenna within the time range requested, generating a global solution for each antenna. The second mode (DOBAND 2) will use the antenna bandpasses nearest in time to the data point being calibrated. The third mode (DOBAND 3) interpolates in time between the antenna bandpasses and generates the correction from the interpolated data. This mode has been found to be required for VLA data. If BPSMO was used to make a fairly finely sampled BP table, then DOBAND 2 may be used.

4.7.4 Amplitude and phase calibration

The channel-0 data set should be calibrated as described above for continuum data (§ 4.4 and § 4.5). When you are satisfied with your results, you should copy the relevant CL table over to the line data set with TACOP:

> TASK 'TACOP' C_R	
> INDI n ; GETN m C_R	to specify the channel-0 data set.
> OUTDI i ; GETO j C_R	to specify the line data set.
> INEXT 'CL' C_R	to copy a CL table.
> INVER 2 C_R	to copy table 2 from CLCAL step.
> NCOUNT 1 C_R	to copy only one table.
> OUTVER 0 C_R	to create new output table.
> INP C_R	to review the inputs.

> GO C_R	to run the program when inputs set correctly.

At this point it is often useful to examine your fully calibrated data using `POSSM`:

> TASK 'POSSM' C_R	
> INDI *i* ; GETN *j* C_R	specify line data.
> SOURCES '*source*1' , ' ' C_R	to specify the source of interest.
> ANTENNAS 0 C_R	to plot all antennas.
> BCHAN 10 ; ECHAN 55 C_R	to plot spectrum for this channel range only.
> DOCALIB TRUE C_R	to apply the antenna gain calibration.
> GAINUSE 2 C_R	to use `CL` table 2.
> DOBAND 3 C_R	to apply the bandpass calibration time smoothed.
> BPVER 1 C_R	to use `BP` table 1.
> FREQID 1 C_R	to use only one `FQ` value.
> APARM 0 C_R	to do scalar averaging of amplitudes and self-scale the plots.
> SMOOTH 1 , 0 C_R	to apply Hanning smoothing in the spectral domain.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.
> GO LWPLA C_R	to send the plot to the (PostScript) printer/plotter.

If you have multiple `FQ` entries in your data set, you should repeat the calibration for each additional `FQ` entry. Bookkeeping is simplified if you eliminate all extant `SN` tables before calibrating the data associated with each frequency identifier. However, it is not essential to do this.

4.8 Solar data calibration

The calibration of solar *uv* data differs from normal continuum and spectral-line calibration in one critical respect: the system temperature correction to the visibility data is applied by the observer in AIPS. See Lecture 21 in *Synthesis Imaging in Radio Astronomy* for a discussion of the system temperature correction as it applies to VLA solar visibility data. The system temperature correction is embodied in a quantity referred to as the "nominal sensitivity," an antenna-based numerical factor normally applied in real time to the scaled correlation coefficients before they are written on the VLA archive tape. With the exception of X and L band, only a handful of VLA antennas are equipped with so-called "solar CALs." The nominal sensitivity is only computed for those antennas so-equipped, namely antennas 5, 11, 12, and 18 (at K, U,and C bands) and antennas 7, 12, 21, and 27 (at P band). The system-temperature correction for those antennas without solar CALs must, therefore, be bootstrapped from those antennas which do. This is accomplished through two tasks. `FILLM` fills the uncalibrated visibility data to disk and places the nominal sensitivities in a `TY` extension table. Then, `SOLCL` applies the nominal sensitivities to calibration parameters in the `CL` table.

4.8.1 Reading solar data from a VLA archive tape

To load a solar *uv*-data file to disk from a VLA archive tape follow the general instructions given above (§ 4.1.1 and § 4.7.1) with the following addition:

> VLAMODE 'S ' C_R	to indicate solar mode observing.

If your experiment involved observing active solar phenomena, (*e.g.*, flares), you may wish to update the system-temperature correction every integration time. For example, if you observed a flare with an integration time $\tau = 1.67$ seconds, choose

> CPARM(8) 1.67 / 60 C_R	for 1.67 sec CL and TY table intervals.

Loading an entire solar *uv*-data set to disk with the minimum integration time results in very large disk files which make all subsequent programs take a long time to run. A useful strategy is to load the data with relatively low time resolution (20–30 seconds for observations of active solar phenomena) and to proceed with the usual continuum data calibration, deferring the system temperature correction. When a satisfactory calibration is obtained, the relevant SN table may be saved using TASAV. (Note that you must save the SN table, before running CLCAL rather than the final CL table.) Then run CLCAL and inspect the data for interesting periods of activity — try UVPLT with BPARM = 11, 1 for plots of amplitude versus time or TVFLG, displaying amplitudes as a function of baseline length and time. Use FILLM to load the relevant time ranges of solar *uv* data to disk with no averaging. The saved SN table is then copied to each high-time resolution data set. Assess, and possibly edit, the nominal sensitivities (§ 4.8.2) and then apply the system-temperature corrections (§ 4.8.3). Finally, apply the saved/copied SN table to the CL table 2 of each using CLCAL.

4.8.2 Assessment of the nominal sensitivities using SNPLT and LISTR

When solar *uv* data are written to disk, FILLM writes the nominal sensitivities of those antennas equipped with solar CALs into the TY table. Before bootstrapping the system temperature correction for antennas without solar CALs from those which do, it is always wise to examine the nominal sensitivity for each of the solar CAL antennas for each of the IFs. There are three tools available for this purpose: SNPLT, which plots the nominal sensitivities in graphical form, LISTR or PRTAB, which allow one to inspect the values directly, and EDITA, which provides an interactive display of the TY data and allows you to edit the data. To make plots:

> TASK 'SNPLT' ; INP C_R	to review the inputs needed.
> IND *m* ; GETN *n* C_R	to specify the input *uv* file.
> INEXT 'TY' C_R	to plot data from TY extension table.
> INVERS 0 C_R	to use the highest version number.
> SOURCES 'SUN' , ' ' C_R	to plot solar source only.
> TIMERANG 0 C_R	to select all times.
> ANTENNAS 5 11 12 18 C_R	to select only CAL-equipped antennas; this sample list for K, U, or C band.
> PIXRANGE 0 C_R	to self-scale each plot.
> NCOUNT 4 C_R	to do 4 plots on a page.
> XINC 1 C_R	to plot every XINC$^{\text{th}}$ point.
> OPTYPE 'TSYS' C_R	to plot nominal sensitivities.
> INP C_R	to review the inputs.
> GO C_R	to run the program when you're satisfied with inputs.

SNPLT produces a PL extension file which may be plotted using LWPLA, TKPL, or TVPL — or you could set DOTV TRUE in SNPLT and get the display directly (and temporarily) on the TV. Then to inspect the values over some limited time range in detail, run LISTR (assuming the name, SOURCES, and ANTENNAS adverbs are set as above):

> TASK 'LISTR' ; INP C_R	to review the inputs needed.
> OPTYPE 'GAIN' C_R	to list quantities in a calibration file.
> INEXT 'TY' C_R	to select the sensitivities.
> TIMER *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to select by suspect time range.
> DOCRT -1 C_R	to route output to the printer.
> DPARM 10 0 C_R	to list nominal sensitivities.
> INP C_R	to review the inputs.

> GO CR	to run the program when you're satisfied with inputs.

The use of EDITA with TY tables is described extensively in § 4.4.2 and need not be described further here.

4.8.3 Applying the system-temperature correction with SOLCL

Once you have identified the appropriate subset of reference solar CAL antennas for each source and IF you are ready to bootstrap the system-temperature correction of the remaining antennas. It is recommended that you run SOLCL before applying any other calibration to the CL table. In this way, you can easily verify that the appropriate corrections have been made to each antenna. If you are using a version of AIPS older than 15JUL94, you must copy CL table version 1 into CL table version 2 before running SOLCL (which does the copy for you in later releases). Then you apply the system-temperature correction to version 2 and correct mistakes by deleting and recreating version 2. To run SOLCL:

> TASK 'SOLCL' ; INP CR	to review the inputs needed.
> SOURCES '*' CR	to correct all sources.
> STOKES ' ' CR	to correct both polarizations.
> TIMERANG 0 CR	to correct all times.
> ANTENNAS 5 11 12 18 CR	to use the listed antennas as references.
> SUBARRAY 1 CR	to modify sub-array 1.
> GAINVER 2 CR	to write corrected entries to CL table version 2.
> INP CR	to review the inputs.
> GO CR	to run the program when you're satisfied with inputs.

After applying the system temperature correction, you may proceed with the usual AIPS data calibration procedures outlined in previous sections, including the special solar tactics described in § 4.8.1.

4.9 Completing the initial calibration

When you are satisfied with the initial calibration (pre self-calibration) of your data set, you should back up your full multi-source data set on magnetic tape. Then you can apply the calibration to the data for each program source, creating a separate single-source *uv* data set for each. These data sets are used with the imaging and self-calibration tasks to be described in the following chapters. For the impatient, there is one imaging task which reads the multi-source data set directly, applying any calibration,

4.9.1 Writing multi-source data to FITS tape with FITTP

The recommended way out of AIPS for multi-source *uv* data is to use FITTP to write a FITS-format tape. This will preserve the data and all associated calibration and editing tables in a machine-independent form. Consult § 3.9 about magnetic tapes in AIPS. That section tells you to mount your tape on the hardware device and then to do a software mount in AIPS. For example,

> INTAP *n* CR	to specify which tape drive to use.
> DENSITY 6250 CR	to set the density to 6250-bpi, if needed.
> MOUNT CR	to mount the tape in software.

This step used to be optional for some operating systems. However, in recent versions of AIPS, it is required on all operating systems.

To write the data to tape:

> TASK 'FITTP' C_R	
> IND *m* ; GETN *n* C_R	to specify the multi-source data set.
> DOEOT TRUE C_R	to write at the end of tape — if there are other data files on the tape you wish to preserve.
> OUTTA INTAP C_R	to write to tape just mounted.
> DOSTOKE FALSE C_R	to leave the data in input Stokes form.
> DOTABLE TRUE C_R	to write associated tables.
> DONEWTAB TRUE C_R	to use the latest format wrinkles — if the version of *AIPS* with which you will read the tape is recent enough. Read the help file for the latest details.
> FORMAT 3 C_R	to use IEEE floating format for data.
> BLOCKING 10 C_R	to use blocked FITS for tape efficiency.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

Most people use 8mm Exabyte or 4mm DAT tapes today. These have very large capacities. However, if you must still use half-inch reel tapes, you will find that many data sets (particularly spectral line) may be too large to fit on one 6250 bpi tape even with `BLOCKING = 10` and `FORMAT = 1` (16-bit integer). Since it is not possible to write multi-volume FITS tapes, it is recommended that you back up the single-source data sets formed after applying the calibration tables in `SPLIT` (see § 4.9.2). Alternatively, since all of the calibration information is contained in the extension tables, you may copy these to a dummy *uv* file with task `TASAV` and write this new file to tape with `FITTP`.

Be sure to run task `PRTTP` to make sure that the data were written succesfully on your tape *before* you delete your multi-source *uv* data set!

4.9.2 Creating single-source data files with SPLIT

When you are happy with the calibration and editing represented by the current set of calibration and flag tables, you can convert the multi-source file into single-source files, applying your calibration and editing tables. Remember that only one `FREQID` can be `SPLIT` at a time.

> TASK 'SPLIT' C_R	
> SOURCE '*sou1*' , '*sou2*' , . . . C_R	to select sources, ' ' means all.
> TIMERANG 0 C_R	to keep all times.
> BIF 1 ; EIF 2 C_R	to keep both IFs
> FREQID 1 C_R	to set the one `FQ` value to use.
> DOCALIB TRUE C_R	to apply calibration.
> GAINUSE 0 C_R	to use the highest numbered `CL` table.
> DOPOL TRUE C_R	to correct for feed polarization.
> DOBAND 3 C_R	to correct bandpass with time smoothing.
> BPVER 1 C_R	to select `BP` table to apply.
> STOKES ' ' C_R	to write the input Stokes type.
> DOUVCOMP FALSE C_R	to write visibilities in uncompressed format.
> APARM 0 C_R	to clear VLBA options, including the one to calibrate the data weights.
> INP C_R	to review the inputs.
> GO C_R	to run the program when inputs set correctly.

The files produced by this process should be completely calibrated and edited and ready to be imaged or further processed as described in later chapters.

It is not necessary to run `SPLIT` to make images with `IMAGR` and it is probably a good idea to make a couple of quick images to make sure that the calibration is okay. However, for serious imaging, it is probably best to run `SPLIT` and then use the single-source output files. See § 5.2 for details of the imaging process.

4.9.3 Making images from multi-source data with `IMAGR`

`IMAGR` can be used to make images from multi-source data files. It is probably a good idea to make a couple of quick images to make sure that the calibration is okay. An example set of inputs to `IMAGR` is:

> TASK 'IMAGR' C_R	
> SOURCE '*sou1*' , ' ' C_R	to choose one source to image.
> TIMERANG 0 C_R	to use data from all times.
> STOKES 'I' C_R	to image total intensity.
> FREQID 1 C_R	to select `FQ` value to image.
> BIF 1 ; EIF 0 C_R	to image all IFs — multi-channel mode images only one IF.
> BCHAN n ; ECHAN m C_R	to combine a range of channels.
> NPOINTS N C_R	to include N spectral channels in each image where $N \leq (m - n + 1)$; for each spectral channel, IFs bif through eif are also included. Note that each channel and IF included in the "average" image is handled individually at its correct frequency.
> DOCALIB TRUE C_R	to apply calibration.
> GAINUSE 0 C_R	to use highest numbered `CL` table.
> FLAGVER 1 C_R	to edit data.
> DOPOL TRUE C_R	to correct for feed polarization.
> DOBAND 3 C_R	to correct bandpass with time smnoothing.
> BPVER 1 C_R	to select `BP` table to apply.
> OUTNAME '*sou1*' C_R	to set the output file name to the source name; the default is `INNAME`.
> OUTDISK 0 C_R	to use any output disk with enough space.
> IMSIZE 512 512 C_R	to set the size in cells of image.
> CELLSIZE 0.25 , 0.25 C_R	to set the size of each image cell in arc-seconds.
> RASHIFT 0 ; DECSHIFT 0 C_R	to (not) shift image center.
> NFIELD 1 C_R	to make only one image.
> UVWTFN ' ' C_R	to use uniform weighting.
> UVBOX 0 C_R	to do uniform weighting over a single grid cell.
> ZEROSP 0 C_R	to introduce no zero-spacing flux.
> XTYPE 0 ; YTYPE 0 C_R	to use the default convolving function.
> XPARM 0 ; YPARM 0 C_R	to use default convolving-function parameters (important!).
> NITER 0 C_R	to do no Cleaning.
> INP C_R	to review the inputs.
> GO C_R	to run `IMAGR` when the inputs are set correctly.

4.10 Additional recipes

4.10.1 Going bananas with bananas

1. Garnish a baked ham or ham steak with bananas.
2. Make a quick, rich desert with bananas and cream.
3. Bananas are perfect for lunch boxes. They come in their own wrapper, are easy to eat and mess-less.
4. Slice a banana in half lengthwise, brush with melted butter and bake it until tender; serve it as a "vegetable" with roasted meats or fish. Very Caribbean.
5. Don't forget old favorites like bananas sliced over cereal, diced in pancake batter, or buried midst the ice cream in a banana split.
6. Slice and stir-fry bananas with carrots, tomatoes and ground beef for a super-quick main dish.

4.10.2 Banana colada

1. Peel and slice 1 ripe **banana**.
2. Place sliced **banana** in blender along with 6 *oz.* pineapple juice (or crushed tinned pineapple in its own juice) and 1 *oz.* rum + 1 *oz.* coconut rum *or* 2 *oz.* rum + 1 *tsp.* Coco Lopez.
3. Optionally add 1 *oz.* **banana** liqueur.
4. Blend until smooth.
5. Add crushed ice, if so desired.
6. If the mixture is too thick, add more juice (or more rum if you prefer!); if too thin, add more banana. This is a really easy recipe to adjust to one's taste.

4.10.3 Hot banana soufflé

1. Preheat oven to 375deg.
2. Select a 6-cup soufflé dish or other mold and grease it liberally with 1 tablespoon `butter`.
3. Place 6 `eggs`, 1/2 cup `cream`, juice of 1/2 `lemon`, 1 tablespoon `kirsch`, and 1/4 cup `sugar` in blender. Blend until the batter is smooth.
4. Peel 2 large `bananas`, removing any fibers and break into chunks. With blender running, add the chunks one at a time.
5. Break 11 ounces `cream cheese` into chunks and add them to the blender.
6. When all the ingredients are thoroughly mixed, run the blender at high speed for a few seconds.
7. Pour batter into prepared dish and place it in the hot oven. Bake 45–50 minutes until the top is lightly browned and puffy. You may quit when the center is still a bit soft or continue baking until the center is firm.
8. Serve at once. A whipped cream flavored with Grand Marnier makes a nice topping.