9 REDUCING VLBI DATA IN $\mathcal{AIPS}$

This chapter describes the reduction of VLBI data in $\mathcal{AIPS}$. A Step-by-step recipe is presented. We also include some background information concerning the structure of VLBI data sets, the data reduction philosophy and a description of some of the effects for which corrections must be determined and applied. It is important to understand these aspects if you wish to reduce your data reliably. For more background information on VLBI data reduction consult *VLBI and the VLBA*, Astronomical Society of the Pacific (ASP) Conference Series No. 82, 1995.

Programs of particular interest for VLBI may be found in Chapter 13 or displayed from inside $\mathcal{AIPS}$ by typing ABOUT VLBI C_R or APROPOS VLBI C_R. Remember, the best and most complete information available on all $\mathcal{AIPS}$ verbs and tasks may be found in their EXPLAIN files. A 15APR97 or later version of $\mathcal{AIPS}$ is required to support full Space VLBI data reduction.

Most types of VLBI data, once read into $\mathcal{AIPS}$, appear very similar in structure as far as the user is concerned. We shall concentrate on describing the reduction path for data produced by the VLBA correlator, but most operations also apply to MkIII and MkII data. Where appropriate, we shall draw the reader's attention to any differences. In particular, § 9.2.2 deals with reading data from a MkIII correlator into $\mathcal{AIPS}$ and the steps necessary to prepare such data for calibration. The few extra steps necessary for calibrating phase-referencing observations are now described in this version of the chapter. Note that successful phase-referencing observations require careful planning **before** the observations.

Some of the VLBI-related tasks require the ability to read files resident outside $\mathcal{AIPS}$. To communicate to $\mathcal{AIPS}$ the directory in which these files exist it is necessary to define a logical pointer or environment variable. Please refer to § 3.10 to see how this is done.

While the majority of VLBI observations are continuum observations, more sophisticated data reduction techniques are increasingly common. Continuum VLBI observers sometimes also apply spectral-line VLBI techniques to improve the dynamic range of their data sets. For these reasons, this chapter is organised to make the discussion of data reduction techniques more uniform. The overview in this section of the steps involved for several type of VLBI data reduction is meant to guide the user through the rest of the chapter. It is strongly recommended that you read the overview carefully before proceeding.

The expected size of the output *uv* data file can be an important consideration in VLBI data reduction. The disk space required by $\mathcal{AIPS}$ for a compressed dataset is given by the relation:

$$\text{Disk Space} = 4 \text{ x } 10^{-6}\ N_{Stokes}\ N_{chan}\ N_{IF}\ \frac{N_{ant}}{2}(N_{ant}+1)\ \frac{T_{expt}}{\triangle T}\ \text{MBytes}$$

where T_{expt} is the total observing time, $\triangle T$ the correlator integration time, N_{Stokes} the number of polarization correlation pairs (RR, LL, RL, LR), N_{chan} the number of spectral channels per IF, N_{IF} the number of IF's, and N_{ant} the number of antennas in the network. SVLBI data set sizes may be much larger and can be computed using the formula given in § 9.2.1.2. In uncompressed format the same data set will require three times the disk space. Be forewarned that some tasks (including FRING) attempt to create uncompressed scratch files which may not fit into the available disk space. The amount of available free disk space can be determined using the AIPS commands FREE. The blocks referred to in the FREE output are equal to 1024 bytes.

Note that certain operating systems are still subject to a 2-Gigabyte limit for any individual file, as a result of their 32-bit file systems. Larger $\mathcal{AIPS}$ files should be possible under systems that support 64-bit file systems. This is presently supported on DEC Alpha workstations and on SGI systems running XFS, amongst others, and may be supported under Solaris (revision $\geq$ 2.6). The size of the output file can be reduced by IF selection or limited concatenation in FITLD or by time- or spectral-averaging later using UVAVG, SPLAT or AVSPC.

It is possible to construct data sets on disk that cannot be written to a single tape using `FITTP` because `FITTP` uncompresses the data when writing to tape. The new task `FITAB` is designed to address this problem. `FITAB` writes data in compressed form to tape and can write data in pieces to multiple tapes. Note that `FITAB` is only availabe in `15APR99` and later releases and that versions of `FITLD` from earlier releases cannot read such data. Packages other than *AIPS* may also be unable to understand these files.

One large point of divergence in the reduction of continuum polarization VLBI data is the question of whether or not to determine separate LL and RR phase solutions. The polarization-specific portions of the recipe given below are based upon the premise that L and R phase solutions should always be determined separately on the grounds that it is safer and should work with data from a wide variety of antennas. If the L-R phase offsets for antennas in your data set are small and constant in time, you may consider modifying the recipe in § 9.1 by determining averaged LL,RR phase solutions everywhere except in step 7.

9.1 VLBI data reduction recipe

1. LOAD THE DATA For data from the VLBA correlator, run `FITLD` (§ 9.2.1.1); if needed, follow up with `MSORT`, `USUBA`, `INDXR`, `VBGLU`, and `MERGECAL` (§ 9.2.1.3–§ 9.2.1.6). For data from a MkIII correlator, run `MK3TX`, `MK3IN`, `MSORT`, `DBCON`, `UVAVG`, `TAMRG`, `SBCOR`, and `INDXR` as needed (§ 9.2.2.1–§ 9.2.2.6). Data from the Penticton correlator should be loaded using `FITLD` (§ 9.2.3), sorted (§ 9.2.1.3), and indexed (§ 9.2.1.5). The loading of data from the Mitaka correlator will be described in the *CookBook* at a later date.

   POLARIZATION: The VLBA correlator incorrectly labels polarizations for single-polarization LL correlation and for dual parallel-hand correlation (RR and LL only). For these types of data, you must additionally run `FXPOL` (§ 9.2.1.8).

   SVLBI: See SVLBI-specific discussion of `FITLD` in § 9.2.1.2. Also, the `OB` table and the time-corrections file should be prepared for data from the VLBA correlator (§ 9.2.1.9–§ 9.2.1.10).

2. EXAMINE THE DATA It is important to familiarize yourself with the data set before proceeding further, especially if you have little experience with VLBI data. There are many *AIPS* tasks for the examination of your data (see § 9.3 for a fuller discussion). Minimally, you should at first run `LISTR`, `IMHEAD`, `EDITR`, `POSSM`, `VPLOT`, and `PRTAN`. At later stages you will probably find `SNPLT`, `PRTAB`, `DTSUM`, and `SHOUV` useful for examining data and calibration tables.

   SVLBI: Task `OBPLT` allows you to examine different aspects of the spacecraft orbit.

3. PROCESS THE CALIBRATION FILES You will have either received calibration files, or instructions on where to obtain them. Some calibration files can be automatically processed into a form suitable for use within *AIPS* using `VLOG` (§ 9.4.2). `ANTAB` is now the primary *AIPS* task for loading calibration information from log files.

   VLBA CORRELATOR: Beginning on 1 April 1999, the VLBA correlator will have attached calibration information directly to your data. This obviates the need to run `VLOG`, `ANTAB`, `PCLOD`, and `UVFLG` to process your *a priori* calibration information for VLBA antennas. Some information for non-VLBA antennas must still be processed as usual.

   POLARIZATION: Be careful to make sure that the polarization labeling of the IFs in the calibration text files is the same as the labeling in the data.

4. EDIT THE DATA Identifying and editing bad data now can save you time later. Data should first be edited using `UVFLG` to apply editing information supplied with your calibration files (§ 9.4.3). Some useful tasks for examining and editing data are `EDITR`, `UVFLG`, `TVFLG`, `SPFLG`, `EDITA`, `VPLOT`, and `QUACK`.

   POLARIZATION: You may want to edit the data consistenly in all polarizations (select `STOKES='IQUV'` within `EDITR`, `TVFLG` or `SPFLG`) — this can greatly simplify the imaging stage (see step 15).

5. APPLY *A PRIORI* CALIBRATION Use APCAL to complete the *a priori* amplitude calibration (§ 9.4.4.2) — this is called the $T_{\rm sys}$ method of amplitude calibration. APCAL can also be used to perform opacity corrections.

   POLARIZATION: For alt-az mounted antennas, a parallactic angle correction for the rotating orientation of the antenna feeds with respect to the observed source must be performed as the first step in the phase calibration using CLCOR (§ 9.4.4.3). This step should be performed no later than immediately after the $T_{\rm sys}$ calibration.

   PHASE REFERENCING: You may want to perform the parallactic angle correction described above for phase referencing observations even if you only correlated the parallel hands (RR, LL).

   VLBA CORRELATOR DATA ONLY: Corrections for sampler biases should be applied using ACCOR (§ 9.4.4.1) BEFORE the $T_{\rm sys}$ calibration. It is recommended that this step be taken for both 1-bit and 2-bit data.

   SPECTRAL-LINE: Unless the line emission is very weak, you may wish to defer amplitude calibration of your *line sources only* until step 12 below. The template method described there is much more accurate than the $T_{\rm sys}$ method.

6. CALIBRATE THE SCALAR BANDPASS RESPONSE FUNCTION Run BPASS or CPASS to determine the bandpass response function using the total-power spectra (§ 9.4.5). This step is not necessary for most continuum observations unless very high dynamic range is sought and even then may not significantly improve the calibration. You should probably skip this step initially and return to it later if you suspect that your images are limited by these effects.

   SPECTRAL-LINE: The bandpass response function should be determined using only the continuum calibrator sources. This step SHOULD NOT BE SKIPPED unless the continuum calibrator source is sufficiently weak that a good bandpass response function cannot be estimated.

7. CALIBRATE THE INSTRUMENTAL DELAYS *Phase-cal*s, or measured single band and multi-band instrumental phase errors, should be applied using PCCOR (§ 9.4.8.5). You can manually perform a phase-cal by running FRING on a limited subset of your data to account for missing phase-cal information or to refine the reported phase-cal measurements (§ 9.4.8.6).

   VLBA CORRELATOR: Phase-cal information must currently be loaded into a PC table using PCLOD (§ 9.4.8.5); some time 'real soon now', the PC table will be provided directly by the correlator.

   SPECTRAL-LINE: Delay calibration should be carried out only on the continuum sources at this stage.

   POLARIZATION: When running FRING, be certain to solve for independent left- and right-polarization delay solutions APARM(3) = 0. Run CROSSPOL after calibrating the instrumental delays, to determine a single delay offset between left and right polarization (§ 9.4.8.11).

   PHASE REFERENCING: Note that in general, you do not want to manually perform a phase-cal upon your *target*, or phase-referenced, source. However, see § 9.4.8.4 for further discussion on this topic.

8. FRINGE FIT THE DATA Estimate and remove residual delays, rates and phases using FRING or BLING (§ 9.4.8.7–§ 9.4.8.8).

   SPECTRAL-LINE: Only fringe-fit the calibrator source at this stage. Check the coherence of the target source using the resulting solutions to decide whether or not to zero the rate solutions using the 'ZRAT' option in SNCOR (§ 9.4.8.10).

PHASE REFERENCING: You should **not** fringe-fit on the *target*, or phase-referenced source. Rather, you should fringe-fit on the *cal*, or phase-reference calibrator. When you apply the solution, be sure to set the `CALSOUR` and `SOURCES` adverbs in `CLCAL` appropriately to interpolate the solutions for the cal source onto the target source (see § 9.4.1.2). If you are not interested in astrometric calibration and your target source is strong enough, you may wish to consider fringe-fitting on it to further refine the phase calibration (§ 9.4.8.4).

SVLBI: Under some circumstances, the 'ground-phasing' task `GPHAS` may prove useful if `FRING` or `BLING` cannot successfully find fringes to the space satellite (§ 9.4.8.9).

9. ESTIMATE THE INSTRUMENTAL POLARIZATION (polarization data only). Correct for the instrumental polarization terms, commonly known as 'D-terms' using `PCAL`, `LPCAL`, or `SPCAL` on the polarization calibrator (§ 9.4.8.13). This polarization calibrator should first be fully calibrated and imaged before this step can be performed.

10. CALIBRATE THE POLARIZATION POSITION ANGLE (polarization data only). If a calibration source with known polarization orientation is available, use `CLCOR` to make a final correction to adjust the polarization angles of the target source data (§ 9.4.8.11).

11. CALIBRATE THE COMPLEX BANDPASS RESPONSE FUNCTION Run `BPASS` or `CPASS` to determine a complex-valued bandpass response function. As before, this step is usually not necessary and should be skipped unless your final images are thought to be limited by these effects (§ 9.4.5).

    SPECTRAL-LINE or POLARIZATION: The bandpass response function should be determined using only the calibrator source. Unlike step 6, this step CAN be skipped.

12. APPLY THE DOPPLER CORRECTION (spectral-line data only). Run `CVEL` to compensate for the changing Doppler shifts of the antennas with respect to the source during the observation (§ 9.4.6).

13. REFINE THE AMPLITUDE CALIBRATION (spectral-line data only). Run `ACFIT` to amplitude calibrate the program source using the template spectra method (§ 9.4.7). Note that the traditional $T_{\rm sys}$ method (§ 9.4.4.2) can also be used if the line emission is too weak for the template method to work successfully.

14. DETERMINE RESIDUAL RATES (spectral-line data only). Now estimate the residual rates ONLY by running `FRING` or `BLING` on one or a few spectral points on the target source (§ 9.4.8.10).

15. APPLY CALIBRATION, AVERAGE, AND INSPECT THE FINAL DATA Run `SPLIT` to apply the calibration solutions (§ 9.5.1), `AVSPC` to frequency-average the data (§ 9.5.1), and `UVAVG` to time average the data (§ 9.5.2). You can also run `SPLAT` to combine these three operations into a single step. It is recommended that you take the time to inspect the calibrated data to see if more editing is needed, and to check that no gross calibration errors remain in the data (§ 9.5.3).

    SPECTRAL-LINE: You probably don't want to average in frequency.

16. SELF-CALIBRATE/IMAGE OR SELF-CALIBRATE/MODEL-FIT THE DATA The final complex gain corrections are determined by iterating self-calibration with imaging of the resultant data set. This is called hybrid-mapping. Alternatively, self-calibration can be iterated while fitting models directly to the data — the goal is to self-calibrate using the best model possible. The options are outlined in § 9.6.

    SPECTRAL-LINE: One final distinction remains between continuum and spectral-line data. Only one or a few spectral points are used to determine final complex gain corrections which are then applied to all spectral points in the line data. After applying these gains, the line source data can be imaged to form an image cube.

POLARIZATION: While the Stokes I and Stokes V images formed using the RR and LL visibilities will be real-valued, the Stokes Q and Stokes U images formed using LR and RL visibilities can, in principle, be complex-valued. You must use a fully complex imaging and deconvolution technique (see the HELP files for CXPOL and CXCLN) or you can simply edit the LR and RL visibilities to enforce the condition that the whenever you have a RL visibility on a baseline, you also have the LR visibility on the same baseline; this ensures that the Stokes Q and U images are real-valued and allows you to use the standard imaging tasks.

9.2 Loading and inspecting data

In theory, AIPS can process data from multiple frequency bands (FQ numbers in AIPS parlance) coexisting within the same data set. In practice, many observers prefer to separate the data for different frequency bands as soon as possible after loading the data and process each FQ number separately. If you wish to do this, you should do it immediately after performing the relevent steps in § 9.2.

9.2.1 Loading data from the VLBA correlator

9.2.1.1 Running FITLD

Data generated by the VLBA correlator are loaded from DAT (or Exabyte) tape into AIPS using FITLD. First, physically load your tape and MOUNT it (§ 3.9), then run FITLD. Often the data on your tape will be divided into a number of separate files (corresponding to separate "correlator jobs"). In this case, run FITLD with NCOUNT set equal to the number of files on the tape (or a suitably large number), as listed on the paper index which comes with the tape. Also set DOCONCAT = 1 C_R to ensure that all tape files with the same structure are concatenated into a single AIPS file. Note that standard tape handling tasks (*e.g.*, PRTTP and TPHEAD) can be used to inspect the tape contents.

Typical inputs to FITLD would be;

> TASK 'FITLD' ; INP C_R	to review the inputs.
> INTAPE n C_R	to specify the input tape number.
> NFILES 0 C_R	to skip no files on tape.
> INFILE ' ' C_R	to load from tape, not from disk.
> OUTNAME 'TEST' ; OUTCL 'FITLD ' C_R	to specify the name of the output file.
> OUTSEQ 0; OUTDI 1 C_R	to specify the sequence number and disk of the output.
> OPTY ' ' C_R	to load any type of file found.
> NCOUNT 20 C_R	to load 20 tape files.
> DOUVCOMP 1 C_R	to save disk space by writing compressed data.
> DOCONCAT 1 C_R	to concatenate files with same data structure into one disk file.
> CLINT $\triangle t$ C_R	set CL table interval to $\triangle t$ minutes (see discussion below).
> DIGICOR 1 C_R	request digital corrections.
> DELCORR 1 C_R	request delay decorrelation corrections.
> WTTHRESH 0.5 C_R	flag incoming visibilities with correlator weights less than 0.5.
> SOURCES ''; QUAL 0; CALCODE '' C_R	to accept all sources found.
> TIMERANG 0 C_R	to accept data from all times.
> BCHAN 0; ECHAN 0; BIF 1; EIF 1 C_R	to accept all channels in all IFs.
> SELBAND 0 C_R	bandwidth to select (kHz).

> SELFREQ 0; FQTOL 0 C_R   frequency to select with tolerance of 10 kHz.
> OPCODE ' ' C_R   to not copy the tape statistics table ('VT' table).
> GO C_R   to run the program.

Because the data files tend to be very large, you should usually write compressed data (DOUVCOMP=1). For most types of observations, it is better to write 'compressed' data sets. These take about 1/3 of the space of 'uncompressed' data sets, but cause information about the weights of individual spectral channels and IF channels to be lost. (See appendix Y for an expanded discussion of when to and when not to write 'compressed' data sets.) If your observation has more than one DAT or Exabyte tape, simply run FITLD for each tape. Setting DOCONCAT 1 and setting the output file name completely will ensure that the data from separate tapes with compatible observing band/data structure will be appended to existing *AIPS* files. Generally, after loading all of your data, you will have one file for each such observing band and/or observing mode. However, observations which require multiple passes through the correlator (including MkIII Modes A, B, and C observations) will have one file per observing mode *per correlation pass*. Data from separate correlator passes can be concatenated using task VBGLU.

Adverb CLINT, which specifies the CL table time sampling interval must be short compared to the anticipated coherence time; CLINT should be set such that the shortest anticipated fringe-fit interval is spanned by a few CL entries. Time sampling in the CL table that is too coarse can lead to calibration interpolation errors when applying the fringe-fit solutions at later stages of the data reduction. If the interval is made unnecessarily short the CL table may become unmanageably large (see the Lecture by Beasley and Conway in VLBI and the VLBA for guidelines on estimating the coherence times expected at different frequencies).

It is recommended that corrections for correlator scaling and quantization be performed in FITLD by setting adverb DIGICOR to one. However, in the special case of spectra with very strong narrow features, the absence of correlator zero-padding may limit the accuracy of the quantization corrections. See the FITLD help file for further information.

Adverb DELCORR enables amplitude corrections for known delay decorrelation losses in the VLBA correlator, as described in *AIPS* Memo 90 (1995, "Delay decorrelation corrections for VLBA data within *AIPS*" by A. J. Kemball). Setting DELCORR=1 will create a correlator parameter frequency (CQ) table for each file written by FITLD. The presence of this table enables the delay decorrelation correction once the residual delays have been determined in fringe-fitting. These corrections will not be applied if the data were not correlated at the VLBA correlator or if the CQ table is missing. For older FITLD files the CQ table can be generated using task FXVLB and this must be done before any changes in the frequency structure of the file are made. The CQ table is used for rate and delay amplitude decorrelation corrections after residual delay and rate errors have been determined by fringe-fitting, and are being applied to the data. The CQ table has no immediate effect on the data written by FITLD but is essential for later processing.

The WTTHRESH adverb can be applied to simply drop incoming data with weights less then the specified limit. Note that data flagged in this way are *unrecoverable* except by re-running FITLD. The data weights are normalized to unity so good data usually have weights close to 1.0. You should carefully examine your data if you use WTTHRESH to make sure that you have not discarded too much data at this stage.

The output files produced by FITLD are in standard multi-source format (as described in § 4.1) and contain data from all the target and calibrator observations in your observation. FITLD also writes a large number of extension tables including an index (NX) table, and many tables containing calibration information. A description of the VLBA correlator table types in given in § 9.8. If you are missing the CORR-ID random axis, your *AIPS* release is stale (pre-15APR97) and you are strongly encouraged to upgrade to the latest release; much of the information presented in this chapter will not be usable with pre15APR97 releases of *AIPS*. Your catalog header should be similar to the one, obtained using verb IMHEADER, given below. If you have GC, TY, FG, WX, and PC tables as in this example data header, congratulations! your data were processed with calibration transfer - see § 9.2.1.7 for more details.

```
Image=MULTI     (UV)        Filename=329        .OVLB  .   1
Telescope=VLBA              Receiver=VLBA
Observer=TM008              User #=   44
Observ. date=23-SEP-1998    Map date=06-JAN-1999
# visibilities      6567    Sort order  **
Rand axes: UU-L  VV-L  WW-L  TIME1  BASELINE  SOURCE  FREQSEL
           INTTIM  CORR-ID  WEIGHT  SCALE
----------------------------------------------------------------
Type    Pixels   Coord value     at Pixel     Coord incr   Rotat
COMPLEX      1   1.0000000E+00       1.00  1.0000000E+00     .00
STOKES       1  -2.0000000E+00       1.00 -1.0000000E+00     .00
FREQ        16   4.9714900E+09        .53  5.0000000E+05     .00
IF           8   1.0000000E+00       1.00  1.0000000E+00     .00
RA           1    00 00 0 .000       1.00        .000000     .00
DEC          1    00 00 0 .000       1.00        .000000     .00
----------------------------------------------------------------
Coordinate equinox 2000.00
Maximum version number of extension files of type HI is   1
Maximum version number of extension files of type CQ is   1
Maximum version number of extension files of type AT is   1
Maximum version number of extension files of type IM is   1
Maximum version number of extension files of type CT is   1
Maximum version number of extension files of type GC is   1
Maximum version number of extension files of type TY is   1
Maximum version number of extension files of type FG is   1
Maximum version number of extension files of type PC is   1
Maximum version number of extension files of type MC is   1
Maximum version number of extension files of type OB is   1
Maximum version number of extension files of type AN is   1
Maximum version number of extension files of type WX is   1
Maximum version number of extension files of type FQ is   1
Maximum version number of extension files of type SU is   1
Keyword = 'OLDRFQ  '  value =  4.97149000D+09
```

Note that the sort order of the output data set in listed as ** rather than TB and that there are no attached CL and NX tables. This happens (sometimes erroneously) when FITLD detects a subarray condition on reading the data or, as in this case, detects the use of multiple integration times on different baselines in the data set (common for SVLBI data, see § 9.2.1.2). The message reported by FITLD in this case takes the form:

```
**********************************************
FITLD5:  Subarray or multiple dump-rate condition found.
FITLD5:  NX/CL tables deleted.
FITLD5:  Use USUBA to set up subarrays.
FITLD5:  Rerun INDXR using CPARM(3) and (4)
FITLD5: ********************************************
```

Unless any of the following criteria are met, the data written by FITLD are immediately ready for further processing.

* If the sort code has been blanked as in this example, you must sort the data (use UVSRT or MSORT).

* If the subarray/multiple dump-rate condition is encountered, you may need to run USUBA.

* If you wish to join together data processed in multiple correlator passes, you must run VBGLU.

* If FITLD does not leave behind CL and NX tables, you must run INDXR to create them.

FITLD can also be used to load archived AIPS data previously written to tape using FITTP, as described in § 5.1.2. In this case the VLBA correlator-specific adverbs, such as those enabling digital and delay corrections, are not active.

If your data were correlated after 1 April 1999 (your IMHEADER listing shows the presence of GC, TY, WX, PC and FG tables) as in this example, you are a beneficiary of calibration transfer. Congratulations! - see § 9.2.1.7 for additional details. Note that, as this example shows, it is possible your data have calibration transfer tables even though it was correlated before 1 April 1999. If your IMHEADER does not show GC and TY tables, you do not have calibration transfer and must manually load calibration information in from text files. Also, even if you have calibration transfer, you may still have to manually load calibration information for some non-VLBA antennas (see *http://www.nrao.edu/vlba/html/OBSERVING/cal-transfer.html/cal-transfer.html* for the most up-to-date information in this regard).

9.2.1.2 Running FITLD for SVLBI data

For a compressed SVLBI data set with integration times on the ground and space baselines of $\triangle T_g$ and $\triangle T_s$ respectively, the total disk space requirement is larger than that given in § 9.1:

$$\text{Disk Space} = 4 \text{ x } 10^{-6}\ N_{Stokes}\ N_{chan}\ N_{if}\ \left[\frac{1}{2}N_{ant}(N_{ant}-1)\frac{T_{expt}}{\triangle T_g} + N_{ant}\frac{T_{expt}}{\triangle T_s}\right] \text{ MBytes}$$

The problem of large data set sizes will be significantly reduced in impact if the time filters (see below) are used to increase the integration times of the space baselines.

Space VLBI data need to be loaded with a version of FITLD that has a revision level of at least 9.0 (as shipped with the 15APR97 version of AIPS). This version supports the loading of spacecraft orbit (OB) tables and the use of time filters. The OB tables contain accurate orbit ephemeris data describing the reconstructed orbit used by the VLBA correlator. This version of FITLD will also create output *uv* data files containing a correlation id. random parameter, identified in the catalog header by the abbreviation CORR-ID. This random parameter allows time-variable changes in the time filtering used by the VLBA correlator for SVLBI data. Apart from unweighted boxcar averaging, the filters supported at present include the 32- and 64-tap filters with decimation factors of 4 and 8 respectively (32-4 and 64-8 in what follows). For further details on the time filters see VLBA Memo. XXXXX by B. Clark. The importance of these filters for SVLBI data in AIPS is that they may allow longer accumulation times on baselines to the spacecraft, with an improved signal-to-noise ratio for the larger expected fringe rates, than might be obtained with unweighted averaging. They also significantly reduce the size of the resulting output *uv* data set.

The use of a time filter can be confirmed by examining the CQ table using task PRTAB. The filter type is encoded in column 11 of the table, with column header FILTER, with *filter_type* = FILTER *modulo* 256, where *filter_type* values of (0,3,4) represent the boxcar and 32-4 and 64-8 filters respectively. Although FXVLB can be used to re-generate a missing CQ table, this should not be done if time filters were used since the resulting CQ table cannot be properly reconstructed; when time filters are used, only FITLD can properly generate CQ tables.

Use of the time filters or use of different integration times on space and ground baselines causes the correlator to write out the data slightly out of time sequence. FITLD will therefore encounter the multiple dump-rate condition described before and you must TB sort the data (see § 9.2.1.3).

The CL table time sampling interval, set via the adverb CLINT should be carefully considered. Because SVLBI data sets usually have at least a few baselines upon which the correlator integration times are extremely small, the CL table can easily become too large if CLINT is unnecessarily small.

Adverbs DIGICOR=1 and DELCORR=1 should be set when loading data generated by the VLBA correlator, as described in § 9.2.1.1.

The VLBA correlator assigns the data from the space antenna an antenna number that depends upon the tracking station used for the downlink. This is useful in data examination and calibration and can be changed at later points in the data reduction using task `RESEQ`. The antenna codes used by the VLBA correlator for tracking stations are two character codes always ending with the letter 'Z'. The tracking station code definitions as of the `15OCT97` release are: 'GZ' (Goldstone), 'NZ' (NRAO Green Bank), 'RZ' (Robledo - Madrid), 'TZ' (Tidbinbilla), and 'UZ' (Usuda).

9.2.1.3 Sorting VLBA correlator data

If time filters are used, or if multiple integration times are used on different baselines, the VLBA correlator will write data that are not in strict time-baseline (`TB`) sort order. In general, task `UVSRT` can be used to sort randomly ordered *uv* data files in *AIPS*, but has significant disk space requirements through the use of intermediate scratch files. A special task, `MSORT`, has been written which uses a direct memory sort with sufficiently large buffers to accommodate the scale over which the data deviate from true time-baseline sort order. No intermediate scratch files are used and it can be significantly faster than `UVSRT` for this special case. The inputs to `MSORT` are similar to those required by `UVSRT` and take the form:

> TASK 'MSORT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OUTDISK *n* ; OUTNAM ' ', OUTCLA ' '	to specify the output file.
> SORT ' ' C_R	to select default sort order ('TB' or time-baseline).
> GO C_R	to run the program.

Note that if the input and output file names are identical, the input file is sorted in place. In place sorting may be necessary if there is insufficient disk space for a second copy of the data set or for the intermediate scratch files required by `UVSRT`. Never abort an in-place sort in progress because you will most likely destroy the integrity of your data set.

9.2.1.4 Subarraying VLBA correlator data

If the project was observed without using subarrays (defined as times at which separate antennas are simultaneously observing different sources or at different frequencies), *this step involving* `USUBA` *is not necessary and can be skipped.*

If the observations have been scheduled in separate subarrays, defined either by source or frequency selection, the subarrays should be labeled in *AIPS* before proceeding any further. The VLBA correlator does not conserve subarray information, which in any event often has no unique characterization. This is specified in *AIPS* using a revised version of task `USUBA` which now allows subarrays to be defined through either the input adverbs, an external `KEYIN` text file, or through the use of an automatic algorithm to identify and label subarrays found in the data. The automatic algorithm is recommended.

For automatic subarray labeling by `USUBA`, representative input parameters would be:

> TASK 'USUBA' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OPCODE 'AUTO' C_R	automatic subarray identification.
> TIMERANG 0 C_R	no time range selection
> ANTENNAS 0 ; SOURCES ' ' C_R	no antenna or source selection
> FREQID -1 ; SUBA 0 C_R	no freq. id. or subarray selection
> INFILE ' ' C_R	no external subarray identifications
> GO C_R	to run the program.

Sometimes FITLD erroneously identifies a subarray condition, usually because of spurious total-power data points. In such cases, you can set OPCODE = ' ' ; SUBARRAY = 1 to force all data into the first subarray.

9.2.1.5 Indexing VLBA correlator data

If FITLD had not written NX or CL tables or it was necessary to sort the data as described in § 9.2.1.3, you must run task INDXR. INDXR will generate an NX table, and if need be a CL table. Typical parameters for INDXR are:

> TASK 'INDXR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> PRTLEV 0 C_R	to print minimal details of progress.
> CPARM 0, 0 ,$\triangle t$, 1 C_R	CL interval $\triangle t$; recalculate model.
> GO C_R	to run the program.

Note that CPARM(4) can be set to zero unless the correlator model is required in later reduction (*e.g.*, in astrometry or geodesy observations) The CL table sampling interval $\triangle t$ should be chosen subject to the same considerations given regarding adverb CLINT in the discussion of FITLD in § 9.2.1.1–§ 9.2.1.2.

9.2.1.6 Concatenating VLBA correlator data

Sometimes an observation is correlated using multiple passes through the VLBA correlator. In this context, multiple pass means different IFs/pass; this is due to data rate limitations in the correlator. FITLD will load each pass into a separate disk file. If it is desired to join together the IFs correlated on each pass, the task VBGLU should be used. VBGLU can adjoin different data sets whose structures differ only in the frequencies covered and which are otherwise identical.

The inputs to VBGLU are rather simple. Each of the input files to be glued together is specified via INNAME–IN4NAME, and an output file is specified via OUTNAME. The choice of input file 1 is important — it should be the file with the largest number of visibilities since it is the one used to set up the times of the data in the output file (see the EXPLAIN file).

> TASK 'VBGLU' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> IN2DISK *n* ; GET2N *ctn* C_R	to specify the 2nd input file.
> IN3DISK *n* ; GET3N *ctn* C_R	to specify the 3rd input file.
> IN4DISK *n* ; GET4N *ctn* C_R	to specify the 4th input file.
> OUTDISK *n* ; GETON *ctn* C_R	to specify the output file.
> GO C_R	to run the program.

9.2.1.7 Calibration transfer

Beginning on 1 April 1999, the VLBA correlator attaches calibration information for VLBA and some non-VLBA antennas directly to the output FITS files (we are not kidding). If your IMHEADER listing shows GC, TY, WX, FG, and PC tables, you are a beneficiary of this service (called *calibration transfer*). Note that projects correlated at slightly earlier dates may also have calibration transfer information. You must have 15APR99 or a patched version of 15OCT98 to take advantage of calibration transfer. Not all antennas provide the all information needed for calibration transfer to the VLBA correlator, see *http://www.nrao.edu/vlba/html/OBSERVING/cal-transfer.html/cal-transfer.html* for the latest information on this subject. For those antennas for which calibration information was not transferred by the VLBA correlator, you must process the log files in the traditional way as outlined in § 9.4.2.

The information processed by the correlator is somewhat redundant so that the calibration tables, the GC table in particular, must be merged using TBMRG. A procedure named MERGECAL is provided for this purpose with 15APR99 and as a patch for 15OCT98. You must first pre-load MERGECAL, before you can examine its inputs and run it:

> RUN MERGECAL C_R	to load the procedure and define some adverbs
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> GCVER 0 CR	to specify the input GC table version.
> TYVER 0 CR	to specify the input TY table version.
> PCVER 0 CR	to specify the input PC table version.
> OUTVERS 0 CR	to specify the output versions for all three tables.
> TIMETOL 0.1 CR	tolerance for comparing times (sec)
> BADDISK 0 CR	to specify which disks not to use for scratch
> INP MERGECAL CR	to review the inputs.
> MERGECAL CR	to run procedure.

Note that the first step, RUN MERGECAL, does not run MERGECAL; it only loads the procedure and defines necessary adverbs. The inputs above will process GC.1, TY.1, and PC.1 table into GC.2, TY.2, and PC.2 respectively. We strongly recommend that you preserve the original versions loaded by FITLD since if they are corrupted, they can only be recovered by re-running FITLD.

Although MERGECAL says it can also copy over FG tables, it does not do so at this time. However, it is also recommended that FG.1 be copied over to FG.2 using TACOP and that all further flagging operations modify FG.2.

Don't forget to set FLAGVER to take advantage of the pre-loaded and user-set flagging information.

You may also find it productive to examine/edit the entries in the MERGECAL'd tables using EDITAB.

9.2.1.8 Labeling VLBA correlator polarization data

The VLBA correlator may label the polarization of the output data incorrectly under certain limited circumstances. This is something that has been corrected over time but it is well to check the polarization labeling at this point. The cases where mis-labeling may occur are: (i) dual parallel-hand correlation (*i.e.*, RR and LL only), and (ii) single-polarization LL correlation. Full polarization correlation data (*i.e.*, RR, LL, RL, LR) have generally been labeled correctly.

In case (i), the effect is to double the number of AIPS IF channels and label them all as RR. This can be corrected using interim task FXPOL, which will correct the mislabeling within AIPS. Calibration tables are lost in using FXPOL so, if needed, this task must be run immediately after FITLD. The CL table can be re-generated using INDXR. Note that although FXPOL does not delete the CQ table, it does not properly update it so that, whenever time-filters are used for correlation, FXPOL should not be used.

Typical inputs for FXPOL are:

> TASK 'FXPOL' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> GO CR	to run the program.

In case (ii), LL data may simply be mis-labeled as RR. This does not need to be corrected within *AIPS* but the user needs to take this into account when selecting or calibrating the data, particularly in specifying the polarization in the amplitude calibration text file (§ 9.4.2). The Stokes axis can however be modified. Before running `PUTHEAD`, you should run `GETHEAD` to check which axis is the Stokes axis in the catalog header. Typical input parameters for `GETHEAD` are:

> INP GETHEAD CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> KEYWORD 'CTYPEn' CR	to select a coordinate type.
> GETHEAD CR	to get the coordinate type.
> TYPE KEYSTRNG CR	to inspect the coordinate type.

You must pick the value of `n` for which `GETHEAD` returns `KEYTYPE` with the string value of `'STOKES'`. Then, run `PUTHEAD` like this:

> INP PUTHEAD CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> KEYWORD 'CRVALn' CR	to select a coordinate type.
> KEYVALUE = -2 CR	to set the Stokes value to 'LL' (or -1 for 'RR').
> PUTHEAD CR	to set the coordinate value.

9.2.1.9 Preparing the OB table for SVLBI data

The spacecraft orbit table (`OB`) as produced by `FITLD` contains only the spacecraft position (x, y, z) and velocity (v_x, v_y, v_z) as calculated to high accuracy from the JPL reconstructed orbit using the SPICE package (developed at JPL). These quantities are calculated by the correlator on-line software and are passed directly through to *AIPS* via `FITLD` by the VLBA correlator. The orbit table is indexed on time and can include information such as the angle between the spacecraft pointing direction and the Sun, the time since the start and end of the last eclipse and the spacecraft parallactic angle. The latter quantities are not available to the correlator on-line software and, if desired, need to be computed separately for later use in *AIPS*. `OBTAB` does not take long to run and it is recommended that it be run at this point since the modifications made to the `AN` table are essential for later use in plotting or inspecting spacecraft orbit information. Sample inputs for `OBTAB`, are as follows:

> TASK 'OBTAB' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> INVERS 1 ; OUTVERS INVERS CR	process `OB` table 1 in place.
> TIMERANG 0 CR	to select all times.
> SUBARRAY 1 CR	to select subarray number.
> APARM 1, 0, 0, 0, 0, 0, 0, 0 CR	to update orbital elements in `AN` table
> GO CR	to run the program.

Note that `APARM(8)` can be used to directly specify which antenna is the orbiting antenna (this task assumes there is only one orbiting antenna).

The tabulated orbital ephemeris data are stored in the `OB` table as discussed above, but *AIPS* also makes provision for orbital elements to be stored in the `AN` table. Task `OBTAB` will determine mean orbital elements from the `OB` table and update the `AN` table under control of `APARM(1)`. The orbital elements can be examined by using `PRTAB` to review the updated `AN` table. The mean elements are used to compute *uv* coordinates for the spacecraft so that model amplitudes and closure phases can be plotted (by tasks `VPLOT` and `CLPLT`) while the *uv* coordinates generated by the correlator (or by `UVFIX`) are used for model division and subtraction, imaging, etc (*e.g.*, `CALIB` and `IMAGR`). The orbit table can be plotted using task `OBPLT`. Alternatively, the task `TAPLT` can be used to display individual columns as illustrated for the delta-T table in Figure 9.1. Use `PRTAB` to determine the names of the columns you wish plotted.

9.2.1.10 Loading the time corrections file for SVLBI data

The round-trip residual delay measurements determined by the tracking stations are supplied to the correlator in FITS format. These so-called delta-T tables are not passed to *AIPS* by the correlator at present but can be loaded indirectly. Support for the delta-T tables within *AIPS* is planned once there is further information regarding the format and availability of the proposed auxiliary time components table. This table might be used, for example, to plot the time correction as a function of time. Such information could be useful if a user suspects a loss of coherence due to a poor predicted orbit or a clock jump at the tracking station. The table contains no internal time stamps, so the row number must be used to determine the approximate time of a given entry (there are typically 10 rows per second).

The delta-T tables can be loaded at present using `FITLD` and attached to a null *uv* data file, using input parameters as follows:

> TASK 'FITLD' ; INP C_R	to review the inputs.
> OUTDISK *n* C_R	specify a separate output file.
> OUTNAM 'DUMMY', OUTCLA 'DT'	
> INFILE 'FITS:3551708.kct.a C_R	external FITS file.
> GO C_R	to run the program.

The other `FITLD` adverbs are not relevant in this instance. Note that lower case letters can be used in the `INFILE` adverb if the trailing quotation mark is omitted. `FITLD` will load the external FITS file successfully but will print an error message complaining that no array geometry table was found. This message can be ignored in this case.

The delta-T table will appear as an unknown table of type `UK`, and can be plotted using task `TAPLT`:

> TASK 'TAPLT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INEXT 'UK' C_R	to select the delta-T table.
> INVER *ukver* C_R	to select the table version.
> BCOUNT 0 ; ECOUNT 0	to select all records in table.
> XINC 1 C_R	plot each record.
> DOHIST 0 C_R	to not plot as a histogram.
> GRCHAN 0 C_R	to use graphics channel 1 on the TV.
> APARM 0 ; APARM(6) 1 C_R	only one column in the table.
> BPARM 0 ; CPARM 0 C_R	default plot control.
> LTYPE 3 C_R	standard plot layout.
> DOTV 1 C_R	for direct plotting on the TV device.
> GO C_R	to run the program.

A sample delta-T table is shown in Figure 9.1.

9.2.2 Loading data from a MkIII/MkIV correlator

9.2.2.1 Running `MK3IN`

Data from a MkIII correlator, such as that in Bonn, Germany or Haystack, Massachusetts, can also be read into *AIPS*. To do this you need to be supplied with the so called "A" tape output, also known as "type 52's." These data tapes can be read and translated by the task `MK3IN`. The process of reading MkIII correlator data into *AIPS* and preparing it for further processing is more cumbersome than the equivalent process

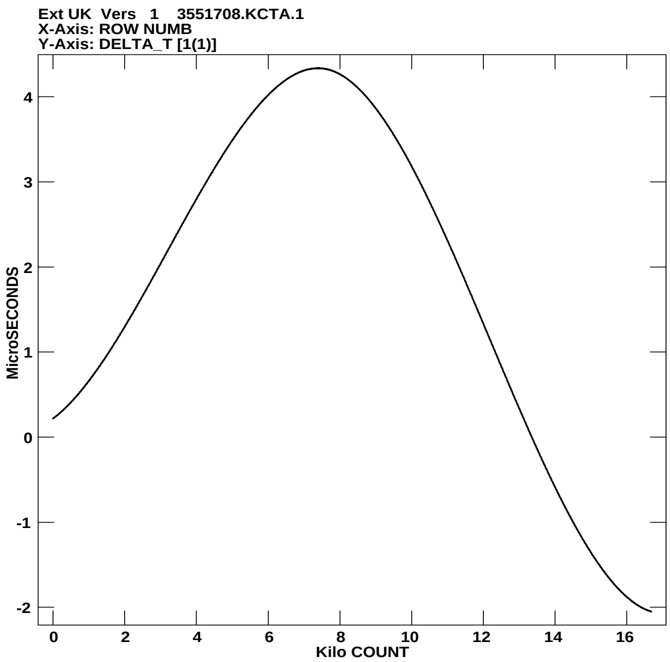

Figure 9.1: A sample delta-T table, where the recorded delta-T value is plotted against row number

for VLBA correlator data. This simply reflects the manner in which data are generated on a baseline-based correlator with a limited number of playback drives.

Before running `MK3IN`, run the task `MK3TX` to extract the text files from the MkIII archive tape. These text files contain information about the correlated scans in the data set. `MK3TX` will first provide an index of all the text files and then ask you to select files for loading onto disk. It then asks you interactively for the desired destination of the text files. It is important to load and concatenate all the "A" files, *i.e.*, those files having names like A*tttt*. The meaning of the other text files is described in the `MK3TX` Explain file. Sometimes the text files are not on the tapes, which means that you cannot select sub-sets of the data using the A-files, but is not otherwise catastrophic.

If the A-files are present and have been loaded onto the disk, use `AFILE` to sort and edit these files to produce a list of scans to be loaded by `MK3IN`. Use `APARM` settings in `AFILE` to establish criteria for selecting between any duplicate scans which may appear on the archive. If the data set contains data at multiple frequencies, you should edit the resulting output text file so that there is a version for each frequency, containing only those scans at that frequency.

The final step before running `MK3IN` is to create another text file which provides the commands for the task. This step is necessary since some information that is needed by AIPS is not present on the tape. Ideally, in this text file (as shown below), the parameter `STATIONS` should be a list of all the stations correlated, with the exact name used at correlation. If you do not have such a list, you can instead specify a list containing `STATIONS 'ANY', 'ANY'` ... Note that there must be at least as many `'ANY'` entries as there are stations in the data set or some of the stations will not be loaded. The parameters in this text file are:

NO_POL=2	the number of polarization correlations (*e.g.*, RR, LL, RL and LR), the default is 1.
STATIONS='NRAO','VLA','OVRO','FDVS','MPI'	station names.

STOKES='RR','LL'	the Stokes range of the output file. The standard abbreviations are used to select the polarization range. The largest consistent range is used. For example: `STOKES='RR','LL'` will cause only RR and LL to be written. `STOKES='LL'` will cause just LL to be written. `STOKES='RR','LR'` will cause all four circular polarization combinations to be in the output file, since RR and LR span the range of allowed *AIPS* Stokes values.
FREQCODE='R','L','r',l'	the polarization codes used by MkIII correlators are anything but standard and they need to be supplied to `MK3IN` using the parameter `FREQCODE`. The one character polarization identifiers are expected in the order RR, LL, RL, and LR. The usual correlator convention is 'R'=RR, 'L'=LL, 'r'=RL, 'l'=LR and this is the default assumed by `MK3IN`. However, other codes are possible. For example `FREQCODE='A', 'B', 'C', 'D'` will interpret 'A' as RR, 'B' as LL and so forth, while `FREQCODE='R', 'C', 'r', 'l'` will use the default abbreviations except that 'C'=LL. If `MK3IN` encounters an unidentified polarization code the task will report: `AT20XX: Unidentified Stokes parameter: '`*X*`'`. In this case, modify the `FREQCODE` parameter to include this polarization identifier. This will ensure that polarizations are not misidentified inadvertently.
/	`keyin` style delimiter.

Then, from inside `AIPS`, mount the tape (§ 3.9) and run `MK3IN`:

> TASK 'MK3IN' ; INP C_R	to review the inputs.
> INFILE 'MYVLB:PARAM.LIS' C_R	to define the text control file.
> IN2FILE 'MYVLB:AFILE.LIS C_R	to point to a file containing a list of scans to be loaded as produced by `AFILE`
> INTAPE 4 C_R	to specify the tape drive number.
> NFILES 0 C_R	to skip no files on tape.
> OUTNA 'EXP 86-34' C_R	to select the output file name.
> OUTCL 'MK3IN' C_R	to select the default output class name.
> REFDATE '12/11/89' C_R	to tell `MK3IN` the start date of the observations — get this right or you may get negative times.
> SOURCES " C_R	to accept all sources found.
> TIMERANG 0 C_R	to accept data from all times found.
> DOUVCOMP 1 C_R	to write data on disk in compressed format.
> APARM 1, 0 C_R	to set the time increment in the `CL` table entries in minutes (to 1 here).
> APARM(7) 1 C_R	to separate sidebands into separate *AIPS* IFs; the default is to store both USB and LSB in the same IF.
> GO C_R	to run the program.

If the data are contained on more than one Exabyte or DAT tape, load the second tape and re-run `MK3IN`, setting DOCONCAT = 1 C_R so that the data are appended to the previous output file. Before running `MK3IN` a second time, it is important to set the list of `STATIONS` in the control file to exactly those found when loading the first tape; use `PRTAN` on the output file to obtain this list. Also leave additional `'ANY'` entries after the list for any stations that are on the second tape but which were not on the first tape. The use of `DOUVCOMP = 1` is recommended for most data sets, see Appendix C.

9.2.2.2 Sorting MkIII/IV data

The *AIPS* data files created by `MK3IN` will be in an arbitrary sort order. Use `UVSRT` to sort them into time-baseline order:

> TASK 'UVSRT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to select the input file.
> OUTNA INNA ; OUTCL 'TBSRT' C_R	to specify the output file.
> SORT 'TB' C_R	to sort to time-baseline order.
> GO C_R	to make the sorted *uv* file.

9.2.2.3 Concatenating MkIII/IV data

If you did not set `DOCONCAT=1` when running `MK3IN` and several files were loaded from tape for one observation, use `DBCON` to concatenate them together. In order to have the concatenated data all appear in a single subarray, both input files for `DBCON` must have the same reference day number and identical antenna numbers. That is, the antennas extension (`AN`) files with each input *uv* data file must be the same.

You may list the contents of `AN` files using `PRTAN`. To run `DBCON`:

> TASK 'DBCON' ; INP C_R	to review the inputs.
> INDISK *n1* ; GETN *ctn1* C_R	to select the 1st input file.
> IN2DISK *n2* ; GET2N *ctn2* C_R	to select the 2nd input file.
> OUTNA INNA ; OUTCL 'DBCON' C_R	to specify the output file.
> DOARRAY 1 C_R	to force `DBCON` to mark the output data records as being in the same sub-array. For this to work properly, both of the input files must have the same reference day and have identical antennas files.
> GO C_R	to concatenate the two files.

9.2.2.4 Merging MkIII/IV data

MkIII VLBI correlators usually produce redundantly correlated data. You must merge the data using `UVAVG`:

> TASK 'UVAVG' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OUTNA INNA ; OUTCL 'UVMRG' C_R	to specify the output file.
> YINC 4.0 C_R	to set the averaging interval of the input data records (in seconds).
> OPCODE 'MERG' C_R	to direct the task to perform the merge operation.
> GO C_R	to run the program.

The `CL` table should only contain one entry for each antenna at each time stamp. But, due to the merging process described above and the fact that redundant correlations may have been performed, there is one step to follow before you have consolidated your database fully. You must run `TAMRG` to remove the redundant `CL` entries:

> TASK 'TAMRG' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INEXT 'CL' C_R	to specify the table type to merge.

> INVER 1; OUTVER INVER C_R	to process the input table in place.
> APARM 4, 1, 4, 0, 1, 1, 1, 0 C_R	to control the merging: don't ask why, just do it!
> BPARM 1, 4 C_R	to set compared columns — again, don't ask.
> CPARM 1.157e-5, 0.2 C_R	to set degree of equality — ditto.
> GO C_R	to run the program.

9.2.2.5 Correcting MkIII/IV sideband phase offsets

If your observation contains a mixture of VLBA and non-VLBA antennas and you have not stored the sidebands as separate IFs, there will be a phase offset of about 130^{deg} between the upper and lower sidebands on baselines from VLBA to non-VLBA antennas. A correction for this offset is achieved using the task `SBCOR`:

> TASK 'SBCOR' ; INP C_R	to review the inputs.
> INDISK *n1* ; GETN *ctn* C_R	to specify the input file.
> OUTNA INNA ; OUTCL 'SBCOR' C_R	to specify the output file.
> BCHAN 1 ; INP C_R	to specify the lowest channel of lower sideband.
> ECHAN 4 ; INP C_R	to specify the highest channel of lower sideband.
> APARM(1) 0 ; INP C_R	to apply the default phase offset (*i.e.*, -130^{deg}.)
> ANTENNAS *n1* , *n2* , *n3* ,... ; INP C_R	to specify the VLBA antenna numbers; use `PRTAN` to identify VLBA antennas.
> GO C_R	to run the program.

9.2.2.6 Indexing MkIII/IV data

Next, you must index your data. The `NX` table is useful as a summary of the file for you, and is also used by the calibration programs to provide quick access for reading data. Create this file with `INDXR`:

> TASK 'INDXR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CPARM 0, 30, -1 C_R	to allow $\leq$ 10-minute time gaps within scans, to limit scans to $\leq$ 30 minutes, and to not create a new `CL` table.
> GO C_R	to run the program.

Other than these initial loading and merging steps, the reduction of MkIII and MkIV correlator data is identical to that of VLBA correlator data.

9.2.3 Loading SVLBI data from the Penticton correlator

The task `FITLD` should be used to load data from the Penticton correlator into *AIPS*. Other than options `DIGICOR` and `DELCORR` which should not be requested, `FITLD` is to be run as described in § 9.2.1.2. The adverb `CLINT` is irrelevant as `INDXR` must be run afterwards to generate the initial `CL` table.

The data must be sorted into time-baseline order (`TB`) using `UVSRT` or `MSORT` (§ 9.2.1.3). Run `INDXR` to index the *uv*data and create the intial `CL` table (§ 9.2.1.5).

The calibration text files required by `ANTAB` must be created by hand for Penticton correlator data.

`PCLOD` is not necessary for Penticton data. `PCCOR` is now thought to process the `PC` table produced by the Penticton correlator, but has had problems with it in the past.

9.3 Tools for data examination

Before proceeding further it is important to examine the data, to make sure they are all loaded, and (especially if this is the first time you have reduced VLBI data) to familiarize yourself with the data structure. As processing continues it is also important to inspect the data periodically to check on the progress of the calibration process. Use the verb `IMHEAD` regularly to check the *uv*-data header, particularly the list of tables (as seen in § 9.2.1.1).

Some tasks that can be used to examine the data and the associated tables are `LISTR`, `DTSUM`, `POSSM`, `VPLOT`, `CLPLT`, `EDITR`, `TVFLG`, `SPFLG`, `SNEDT`, `SNPLT`, `PRTAB`, `FRPLT`, `PRTAN`, `COHER`, `OBPLT`, and `SHOUV`. Some of these tasks are described in the next few pages.

9.3.1 Textual displays

As a first step, the task `LISTR` can be used to give a listing of the scans, with source names, time ranges, frequency ID's and total number of visibilities per scan for each of your output files. It is often useful to print out a paper copy of this to facilitate later data plotting/editing. Use:

> TASK 'LISTR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OPTYP 'SCAN' C_R	to request printing of scan summaries.
> DOCRT -1 C_R	to direct the output to a printer.
> OUTPRINT ' ' ; C_R	to have the output printed immediately.
> GO C_R	to run the program.

Note that at the end of the above `LISTR` output is useful information about the frequency structure of your data set. `LISTR` is also a good way to look for phase coherence. If `LISTR` fails at this point, you may have forgotten to run `INDXR` and/or `MSORT` (see § 9.2.1.5 and § 9.2.1.3).

Other verbs/tasks for inspecting your data include;

1. `IMHEAD` lists the file header including information on the number of frequency channels and Stokes parameters in the data and gives a list of all the extension tables.

2. `PRTAB` can be used to print the contents of any of these extension tables, for instance the `SU` or Source table contains information about each of the sources observed.

3. `PRTAN` provides a listing of the antennas names and their associated antenna numbers. This is useful because in antennas are generaly specified by their antenna numbers.

4. `DTSUM` produces a matrix showing the number of visibilities on each baseline for each scan allowing a check to be made that all baselines have been loaded. It also tells you the data integration time. `DTSUM` also has a mode (triggered by setting `APARM(1) = 1`) that will produce a useful matrix summary of your whole data set. `DTSUM` does not properly report the integration times when there are multiple integration times in the data set.

9.3.2 Spectral displays: `POSSM` and `SHOUV`

Your data file will probably contain a number of IFs, observed at different frequencies, corresponding to the separate "IF channels" used during the observations. While the VLA has a maximum of two such IF channels (four polarizations are counted separately), VLBA antennas have up to eight IFs (and effectively

16 if double sideband recording was used). MkIII Mode A observations can produce up to 28 IF channels. Within each IF channel are a number of equally spaced and contiguous "spectral channels" generated from each IF data stream by the correlator. For continuum applications, the correlator will generally produce 8 or 16 such spectral channels; in the spectral-line case there may be as many as 1024 such spectral channels. For VSOP data, the VLBA correlator will generally produce 256 spectral channels per IF. Use IMHEAD to find the number of IF channels and the number of spectral channels per IF channel, or examine the LISTR output. The reason that the data must be stored in narrow spectral channels, even for continuum applications, is that, in VLBI, the geometrical and propagation errors affecting the data can be large enough to cause significant phase changes across an IF channel bandwidth, preventing a coherent integration over the full bandwidth.

The frequency structure of the data can be inspected using POSSM, which provides a plot of visibility data as a function of frequency as integrated over a specified time interval. Optionally, data from up to nine baselines can be plotted on a single plot page. Initially it may be interesting to view the frequency structure of data on a bright calibrator source, as in the example below. Because, prior to calibration, the phases in each IF channel are likely to vary rapidly with time, it is important to average data coherently only over a short time interval. In general, you will see phase slopes and offsets affecting the data; these phase errors must be determined and removed before the data can be averaged in frequency and/or time. See § 9.4.8 for more information and a sample plot. To display the visibility spectrum of a source on the TV, use:

> TASK 'POSSM' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCE 'OQ208' C_R	to specify a single source name.
> TIMER 1 2 15 0 1 2 15 30 C_R	to define a time range.
> ANTENNAS 8 ; BASELINE 0 C_R	to plot all baselines to antenna 8.
> DOCAL -1 C_R	to not apply any calibration.
> APARM 1 , 1 , 0 , 0 , -180 , 180 , 0 , 0 , 1 , 0 C_R	to control the plot as APARM(1)=1 to use vector averaging, APARM(2)=1 to use fixed scale plots, APARM(5) and APARM(6) to set phase range, APARM(9)=1 to plot all IFs together in one diagram.
> BIF 0 ; EIF 0 ; BCHAN 0 ; ECHAN 0 C_R	to include all IFs and spectral channels.
> CODETYPE ' ' ; POLPLOT ' ' C_R	to plot amplitude and phase.
> SOLINT 0 C_R	to average over the full time range.
> NPLOTS 9 ; BPARM 0 ; OUTFILE ' ' C_R	to have 9 plots per page without division by "channel 0" and without writing the spectrum to a file.
> DOTV 1 C_R	to plot on the TV, else create plot extension.
> BADDISK 0 C_R	to use all disks.
> GO C_R	to run the program.

Note that the amplitudes are totally uncalibrated at this stage and are in units of "correlation coefficients"; these will generally appear on plots mislabeled as mJy (representing multiples of 10^{-3} correlation coefficients).

Task SHOUV is similar to POSSM but produces text output instead of plots. When multiple integration times are present, DPARM(4) must be specified in order to run SHOUV.

9.3.3 Time displays: VPLOT and CLPLT

The task VPLOT can be used to view the visibility data as a function of time (or other variables). Again, data from several baselines can appear on one plot page. Plots of amplitudes and phases and several other quantities can be made, although, to view closure phase, you must use task CLPLT. Note that VPLOT will plot data from a single IF and spectral channel or can coherently average the data over many spectral and IF channels. Calibration can be applied before averaging these channels. Also, if desired, a model can be plotted against the data. The model can either be displayed at the times of the data samples or, with somewhat less accuracy, continuously, even at times for which there is no associated data or recorded *uv* coordinate values.

The following parameters will display uncalibrated amplitudes and phases from a single spectral channel of a single IF channel for a short scan on a bright calibrator:

> TASK 'VPLOT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CLR2NAME C_R	to ensure no model is plotted.
> SOURCE 'OQ208' C_R	to specify the source name.
> BIF 4 C_R	to give first included IF channel.
> EIF 4 C_R	to give last included IF channel;, if EIF > BIF then IFs are averaged.
> BCHAN 8 C_R	to set the lowest spectral channel to include in average prior to plotting.
> ECHAN 8 C_R	to set the highest spectral channel to include in average prior to plotting; no averaging in this case.
> TIMER 1 2 15 0 1 2 25 00 C_R	to define a 10-minute time range.
> DOCAL -1 ; DOBAND -1 C_R	to apply no calibration tables or bandpass tables to data.
> OPTYP ' '	to display cross-correlations; 'AUTO' to get auto-correlations.
> SOLINT 0	to do no time averaging of the data before plotting.
> XINC 1 C_R	to plot every record.
> BPARM 0 , -1 C_R	to set x-axis type (BPARM(1) = 0 plots time (Hrs, min, sec)), y-axis type (BPARM(2)= -1 plots both amplitude and phase), and to use self-scaling. See EXPLAIN VPLOT C_R for other options.
> NPLOT 4 C_R	to plot 4 baselines/page.
> GO C_R	to run the program.

An example VPLOT output appears as Figure 9.2. It may be useful to make a plot of data "weight" versus time on the autocorrelation data from each antenna (set BPARM(2)=16 and OPTYP = 'AUTO'). The weight depends on the number of valid bits correlated and is a good indication of tape playback quality.

9.3.4 EDITR

Another task which can display data as a function of time is EDITR. This program is used primarily to edit data interactively (see § 5.5.2), but its interactive aspects (*i.e.*, allowing the user to "zoom" in on certain time periods) make it useful for pure data inspection. EDITR has been much improved in recent releases to offer options reminiscent of difmap, the VLBI data reduction package from CalTech. It is well worth exploring the abilities of this powerful program. Tasks TVFLG (§ 4.4.3) and SPFLG (§ 10.2.2) are also useful in this way.

9.3.5 SNPLT

Supplied with your VLBI data will be a number of important tables used for calibration and many more are generated as calibration proceeds. § 9.8 summarizes the contents of each of these tables. Two of the most important tables are the calibration or CL table and the solution or SN tables.

The task SNPLT should be used periodically to inspect the contents of the latest CL and SN tables. The example below plots the antenna-based amplitude corrections stored in CL table 2.

> TASK 'SNPLT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OPTYPE 'AMP' C_R	to plot amplitudes.
> OPCODE 'ALST' C_R	to plot both polarizations in each plot; selected IFs are plotted separately.
> INEXT 'CL' ; INVER *n* C_R	to choose CL table version.
> DOTV 1 C_R	to plot on the TV screen; otherwise, create plot extension files.
> NPLOTS 5 C_R	to plot 5 antennas/IFs per page.
> GO C_R	to run the program.

SNPLT can also be used to plot quantities from other tables generated by the calibration process including the contents of SN ("solution tables"), TY ("system temperature tables"), and PC ("phase-cal tables").

9.3.6 COHER

The task COHER can be used to determine the coherence time in a *uv* data set broken down both on time and by antenna and baseline. The coherence time is estimated by comparing vector and scalar averaged amplitudes over increasing time averaging intervals. Averaging is not performed over source scan boundaries. The coherence time is defined as the averaging interval over which the ratio of vector and scalar amplitudes falls below a pre-assigned level. This can be set under user control. In addition, data that fall below a specified signal-to-noise ratio can be excluded from the coherence time estimates. Provision is made for selection by source name, time range, IF channel, antenna and frequency id., and the ability to average over a subset of individual frequency channels. The input parameters to task COHER take the form:

> TASK 'COHER' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the *uv* input file.
> APARM 0 C_R	SNR cutoff 5; vector to scalar cutoff 0.8.
> TIMERANG 0, 10, 5, 0, 0, 10, 15, 0 C_R	time range selection.
> BIF 5 ; BCHAN 1 ; ECHAN 8 C_R	IF and channel selection.
> SOURCES 'DA193' , ' ' C_R	source selection.
> FREQID 1 C_R	Frequency id. selection.
> GO C_R	to run the program.

Be warned that COHER can take quite a long time to run. A simpler, though less rigorously correct method for determining coherence intervals is to examine the data on different baselines using EDITR.

9.3.7 FRPLT

A preliminary examination of the coherence of individual scans or time segments in the data can also be performed using task FRPLT, which allows time series or fringe-rate spectra to be plotted for a selected baseline. This task allows data to be selected by the usual criteria, including time range, source name, and frequency parameters, amongst others, and will then plot the individual time series in amplitude and phase or the associated fringe-rate spectrum. Provision is made for averaging over frequency channels within each IF, for varying degrees of padding in the FFT, and for division by the pseudo-continuum average "channel zero" before plotting. Typical input parameters to FRPLT are:

> TASK 'FRPLT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.

> SOURCES 'DA193' , ' ' C_R	source selection.
> TIMERANG 0, 10, 5, 0, 0, 10, 15, 0 C_R	time range selection (required).
> STOKES 'LL' C_R	Stokes selection.
> BIF 1 ; EIF 0 C_R	IF and frequency channel selection.
> BCHAN 3 ; ECHAN 12 C_R	
> ANTENNAS 3, 4 C_R	select baseline 3-4 (required).
> DOCAL -1 ; DOPOL -1 C_R	to apply no continuum calibration.
> DOBAND -1 C_R	to do no bandpass calibration.
> APARM(1) 1 ; APARM(2) 0 C_R	integration time 1 sec; self-scale the plots.
> APARM(7) 0 C_R	plot fringe-rate spectra with no padding.
> APARM(8) 10 C_R	pre-average all 10 frequency channels.
> BPARM 0 C_R	no division by "channel zero".
> DOTV 1 C_R	plot directly to the TV device.
> GO C_R	to run the program.

The underlying time series in amplitude and phase can be plotted by setting APARM(7)=1. Note that the baseline and time range must be selected using ANTENNAS and TIMERANG and the existing integration time in the data set using APARM(1). A separate plot will be produced for each IF channel in all cases.

9.4 Calibration strategy

As noted at the beginning of § 9.2, if you have multiple FQ data, you may want to separate the data for different FQ before performing any calibration. Use UVCOP or SPLAT, not SPLIT, now to do this (you may also need to re-run INDXR at this point).

We can now begin the process of calibrating VLBI data. As the calibration process proceeds, both amplitude and phase corrections are incorporated into the CL tables. VLBI correlator output is in terms of dimensionless "correlation coefficients." To convert to Janskys, large amplitude correction factors have to be entered into the CL tables. In addition, phase correction factors must be entered into the CL table to correct for phase offsets and ramps as functions of frequency and time. These corrections must be made so that the data can be averaged over frequency and time without loss of coherence. The process of determining the phase corrections is known as "*fringe-fitting*" in VLBI; see § 9.4.8.

Unlike VLA users, VLBI users normally do not attempt to calibrate the absolute phase of the data using external calibrator sources. VLBI users just calibrate out the phase derivatives with respect to time and frequency. The absolute phases are normally left uncalibrated and "self-calibration" methods are then used to generate images (see § 9.6). The alternative of absolute-phase calibration using an external calibrator, known as "phase-referencing" in VLBI circles, is a little more difficult in VLBI than for the VLA. In general terms, "phase-referencing" VLBI data in AIPS is accomplished by similar methods as used for VLA data in AIPS , be sure to read § 9.4.1.2 and § 9.4.8.4 for details on how to calibrate phase-referenced observations.

At some future date, astrometric data reduction methods in AIPS will be outlined in this chapter. For now, the reader is referred to the guide to AIPS astometric data reduction available from within AIPS by typing HELP ASTROMET.

Optimum fringe-fitting results are obtained if amplitudes are calibrated first, since, in this case, the data will be weighted appropriately (the AIPS task FIXWT may be used to adjust the weights in the data to reflect the scatter of the actual data). We therefore describe the process of amplitude calibration first in § 9.4.4.2. Then, in § 9.4.8, we describe the calibration of residual phase using "fringe-fitting" techniques. Note however, that for observations of strong sources or observations using only VLBA antennas (where the sensitivities on all baselines are roughly the same), the order of the two calibration steps may be reversed.

9.4.1 Incremental calibration philosophy

The general strategy adopted by AIPS for calibration is, starting with the lowest version of the CL table, to incorporate step-by-step amplitude and phase corrections for a number of different effects. At each stage either an existing CL table is modified or a new version is created from a lower version by a task which applies a certain type of calibration. Note that the actual visibilities are not changed until you are satisfied that you have the best possible calibration file; at this point the task SPLIT can be used to apply the calibration information of the best CL table to the data. However at each point along the way the effect of a particular CL table on the data can be viewed using POSSM or VPLOT by setting DOCAL=1 and GAINUSE equal to the chosen CL table. Since many CL tables may be produced in the course of calibrating a VLBI data set it is important to keep a note of which effects are included in each one. Ideally one should delete CL tables which are judged incorrect and ensure that the accumulated corrections lie in the highest numbered CL table. It is recommended that version one of the CL table, as produced by FITLD, be copied to CL.2 using TACOP before any calibration is begun, and that CL.2 be used as the starting point in the calibration sequence. An effort is made within AIPS to insure that CL.1 is not deleted inadvertently. If this does occur however it can be re-generated using task INDXR (described in § 9.2.1.5). (Note that INDXR cannot re-generate some types of information, *e.g.*, the phase-cals inserted by the MKIII correlator so that it is important to try to preserve the first CL table.) Note that the task TASAV can be used to copy all the tables to a dummy file containing no data. This can be used to save a “snapshot” of the tables at various points in your data processing for insurance purposes. The tables can be copied back into your data file if necessary using TACOP.

You can also use the verb HINOTE to add comments into the history file. This can be very useful for checkpointing progress during calibration.

The task SPLAT may now be used instead of SPLIT. SPLAT can also time-average the data and produces multi-source data sets on output if requested.

9.4.1.1 Smoothing and applying corrections in SN and CL tables

The various stages of calibration described below produce SN tables which are then used to create CL tables using CLCAL. The ancillary tasks SNCOR, SNSMO, SNEDT, and CLCOR can be used to modify the SN and CL tables directly. It is important to choose the proper methods of interpolation in these tasks.

SN tables can be smoothed using tasks SNEDT and SNSMO and before being used to update CL tables using CLCAL.

Typical inputs for SNSMO would be:

> TASK 'SNSMO ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCES ' ' C_R	use specify sources whose solutions are to be modified.
> SELBAND -1; SELFREQ -1; FREQID 0 C_R	do not select on frequency.
> BIF 0; EIF 0 C_R	do not select by IF.
> TIMERANG 0; ANTENNAS 0 C_R	to not select in time or by antennas
> SUBARRAY 0 C_R	to select the first subarray; NB, SNSMO works only on one subarray at a time.
> INTERPOL 'MWF' C_R	to use the median window filter method.
> BPARM 0.5,0 C_R	to use a 30 minute filter time.
> DOBLANK 1 C_R	to interpolate blanked values.
> SMOTYPE 'AMPL' C_R	to smooth amplitudes only.
> CPARM 0.5, 0, 0, 0, 0, 0.02, 0 C_R	to set ranges of allowed values.

> INVER *snin* ; OUTVER *snout* CR	to read in the SN table and to create a new SN table.
> REFANT 0 ; BADDISK 0 CR	to keep current reference antenna; to specify disks no to use for scratch files.
> INP CR	to check the inputs.
> GO CR	to run the task.

Typical inputs for CLCAL would be:

> TASK 'CLCAL' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> CALSOUR ' ' CR	use all corrections in the SN table.
> SOURCE ' ' CR	apply corrections to all sources.
> OPCODE 'CALI' CR	to apply SN tables to a CL table.
> INTERPOL ' ' CR	to use linear vector interpolation with no SN table smoothing. see below for more discussion of INTERPOL.
> SMOTYPE ' '; INTPARM 0 CR	to clear interpolation and smoothing parameters.
> SNVER *snin* CR	to select the SN table containing solutions to be interpolated.
> GAINVER *clin* CR	to select the CL version to which solutions are to be applied.
> GAINUSE *clout* CR	to select the output CL version, containing updated calibration information.
> REFANT 0 ; BADDISK 0 CR	Try to consistently use a single reference antenna if possible during all steps of calibration.
> INP CR	to check the inputs.
> GO CR	to run the task.

Note well that SNVER=0 means here to combine the solutions from all SN tables, GAINVER=0 means to apply the solutions to CL table 1, and GAINUSE=0 means to write to CL table 2 — using these defaults unintentionally can be the source of serious errors. If two SN tables contain two similar attempts at finding corrections and SNVER=0 then, effectively, CLCAL will apply the solutions twice!

You will almost never want to use these defaults for VLBI data reduction.

The parameter INTERPOL allows the user to choose between several different methods of phase interpolation. Use EXPLAIN CLCAL CR to view all the options. The default interpolation option is INTERPOL = 'BOX', in which the SN table is smoothed with a boxcar function before being linearly interpolated onto the CL table. The boxcar smoothing times for delay, rate etc. are specified in parameter INTPARM. However it is recommended that SN smoothing be done prior to CLCAL using task SNSMO.

With good quality data, the 'AMBG' option should work well. Note, however, that this option uses the SN solutions immediately before and after a CL entry to make the interpolation and it uses any SN solution found for any source specified in CALSOUR. Therefore, if CALSOUR is left blank (allowing all sources) and delay and rate solutions were significantly different for different sources, then inappropriate solutions may be applied for a few minutes before or after a source change.

One way of avoiding this problem is to run CLCAL with INTERPOL = 'AMBG' several times, once for each source, setting both SOURCE and CALSOUR to the name of the desired source with all other inputs remaining unchanged. Another way of avoiding the problem is to use INTERPOL = 'SELF'. In this option, only solutions found on a given source are used to calibrate that source and the single SN table entry closest in time for that source is used without any interpolation. This is not as good as doing multiple runs with the INTERPOL = 'AMBG' option because there can be jumps in phase at points equidistant from two SN table entries.

If there are bad SN solutions, INTERPOL = 'POLY' is used to fit a polynomial to the rate solutions and then integrate this polynomial to determine the phase corrections to be entered into the CL table.

A final note on CLCAL. It is sometimes the case that *a priori* information is not available for all antennas in a single format. For example, you may have system temperature information for VLBA antennas in your SN.2 table and for non-VLBA antennas in SN.3. You can merge this information by running CLCAL twice with the same GAINVER and GAINUSE; each time you should explicitly set the SN version number and list the antennas to be processed using the ANTENNAS adverb. If you leave the ANTENNAS adverb blank, the final CL table will contain information only for antennas present in the last SN table processed.

9.4.1.2 Running CLCAL for phase referencing obsrvations

In particular, this example illustrates how to set the inputs for CLCAL for the specific case when phase corrections determined for the cal source 'J1636-36' are to be transferred to the target source 'P1643-12' in a phase referencing experiment:

> TASK 'CLCAL' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> CALSOUR 'J1636-36',″ CR	use the corrections determined for the cal source.
> SOURCE 'J1636-36','P1643-12',″ CR	apply corrections to both the cal source and the target source.
> OPCODE 'CALI' CR	to apply SN tables to a CL table.
> INTERPOL 'AMBG' CR	to use linear vector interpolation with no SN table smoothing and simple phase ambiguity removal. See above for more discussion of INTERPOL.
> SMOTYPE ' '; INTPARM 0 CR	to clear interpolation and smoothing parameters.
> SNVER *snin* CR	to select the SN table containing solutions to be interpolated.
> GAINVER *clin* CR	to select the CL version to which solutions are to be applied.
> GAINUSE *clout* CR	to select the output CL version, containing updated calibration information.
> REFANT 0 ; BADDISK 0 CR	Try to consistently use a single reference antenna if possible during all steps of calibration.
> INP CR	to check the inputs.
> GO CR	to run the task.

It's a good idea to always apply the calibration information to both the cal and target sources when running CLCAL for phase-referencing observations. This allows you to monitor the cal source data to check the progress of the phase calibration procedure.

9.4.2 Processing observing log and calibration information

As of 1 April 1999, the VLBA correlator provides calibration transfer information, as described in § 9.2.1.7 for VLBA antennas. Consequently, you can skip § 9.4.2 entirely unless you have non-VLBA data, space-VLBI data, VLA data, or you wish to process the log files manually for other reasons.

This section describes the processing of external calibration information, as supplied in ASCII log files. The information that may be used by AIPS includes T_{sys} or related total power measurements, edit flags as written by the tracking stations or on-line monitor control system, weather information, and pulse-calibration data. These external data can be read into AIPS by tasks ANTAB, UVFLG, APCAL, and PCLOD respectively, as described in § 9.4.2.5, § 9.4.3, § 9.4.4.2 and § 9.4.8.5.

You should have received information about where to obtain your calibration data. VLBA calibration log files may be obtained by ftp as described below. Similar calibration files for other participating antennas and VSOP should be obtained from the appropriate sites.

9.4.2.1 Automatic formatting of VLBA and VLBA-like log files

For VLBA antennas, the external calibration file for a given experiment can be obtained by ftp to **aspen.aoc.nrao.edu (146.88.1.124)** under user account **vlbiobs**. The files are named as *xxxxx***cal.vlba.gz**, where *xxxxx* is the project code (*e.g.*, bz199), and are stored in gzipp'ed format. Gain curve information can be obtained for the VLBA antennas by anonymous ftp to host **ftp.aoc.nrao.edu (146.88.1.10)** to retrieve file **vlba_gains.key** in the **pub** sub-directory. The gain curve file should be concatenated to the external calibration file.

The external calibration file, if suitably close to the standard VLBA format, concatenated with the gain curve file, can be automatically subdivided and re-formatted to comply with *AIPS* requirements using task `VLOG`. Typical inputs would be:

> TASK 'VLOG' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the `FITLD` output file.
> SUBARRAY 1 C_R	select the required subarray.
> INFILE 'FITS:bz199cal.vlba C_R	external calibration file.
> OUTFILE 'FITS:BZ199' C_R	prefix and directory for output files.
> FQTOL 1000 ; CPARM 0 C_R	tolerance for frequency match (kHz).
> LTYPE 3 C_R	standard plot layout.
> PRTLEV 0 C_R	do not echo calibration file to the screen.
> GO C_R	to run the program.

A sequence of output text files will be created in the `$FITS` directory named `BZ199.*` with suffixes:

1. `.TSYS:` $T_{\rm sys}$ calibration data, including gain curves, suitable for direct use by `ANTAB`. An `INDEX` record is constructed for each frequency id. if possible. If no match can be made to the FQ data a warning message is printed and the `INDEX` keyword is omitted; the `INDEX` keyword must then be inserted by hand (see the HELP file for `ANTAB`). Gain curve entries for the frequency and time range in the *uv* data are copied to this output file. It is recommended to insert $T_{\rm sys}$ values at the end of scans immediately before source changes to avoid interpolation problems for sources of greatly differing flux density. Occasionally spurious data from previous observing runs or system startup files will end up in the `.TSYS` file and must be edited out by hand.

2. `.FLAG:` Flag data, suitable for direct use by `UVFLG`. In the older format "antennas=vlba_xx" the appropriate antenna and day numbers are inserted.

3. `.WX:` Weather data, as used by `APCAL` if performing an opacity solution. This file is altered to conform with `APCAL` requirements (*e.g.*, `WEATHER` keyword) and lines with bad entries (*) are commented out.

4. `.PCAL:` Pulse-calibration data, for input to `PCLOD`. No editing is performed.

Files with suffixes `.SCAN` and `.MKIII` contain scan summaries and MkIII information and are for information purposes only.

9.4.2.2 VLA log files

VLA calibration files, named xxxxxcal.y.gz and stored in gzipped format, can be obtained from the same server as for VLBA files. The VLA file starts with an explanatory preamble, including minor editing instructions.

9.4.2.3 SVLBI log files

HALCA calibration files may be obtained by pointing a web browser to *http://www.vsop.isas.ac.jp* and following the link to "HALCA Calibration". You will need to get the `.TSM` files which are in a VLBA-like log format and the halca_gains.key file.

9.4.2.4 Manual formatting of log files

Partitioning of the calibration file can and must be done by hand if the calibration file format is sufficiently distinct from the standard VLBA format. If your observation used non-VLBA antennas you will need to edit the calibration text file manually to add any log information supplied for these antennas. The necessary steps are as follows:

Extract the flagging and Tsys information from the calibration text file. You will also need to prepend gain curves to the $T_{\rm sys}$ file. Try `EXPLAIN ANTAB` to see an example file in the proper format. The parameter `TIMEOFF` should be set to zero for each station since both flag information and data are stored with UTC times. A new keyword `DTIMRANG` is also supported which pads each flagged time interval to insure that even very short flag intervals are applied.

While older VLBA format calibration files were supplied with the `ANTENNA` keyword, newer VLBA format calibration files are supplied with the `ANT_NAME` keyword. In the former case the file should be edited to insert the antenna numbers as listed in the `AN` table (use `PRTAN` on your *AIPS* file to find these) and the absolute day numbers must be replaced by relative day numbers with respect to the *AIPS* reference date. In the latter case, no adjustments to either day numbers or antenna numbers are necessary.

9.4.2.5 Loading calibration log information

The calibration information in the external text files such as $T_{\rm sys}$ and gain curve measurements are read into `TY` and `GC` tables using `ANTAB`. These tables are then used by `APCAL` to generate an amplitude solution (`SN`) table, allowing an optional solution for atmospheric opacity. The user is advised to read the `ANTAB` help file closely and check the syntax of the text file carefully.

The INDEX keyword is used to assign the tabulated $T_{\rm sys}$ data to individual *AIPS* IF channels and polarizations. Up-to-date information on the usage of the `INDEX` keyword may be found by typing `EXPLAIN ANTAB`. Be careful to match up the proper polarization labels for the tabulated $T_{\rm sys}$ information. The frequency and polarization association for each IF channel in the *AIPS* file can be compared (use `LISTR` with `OPTYPE = 'SCAN'`) with that at the head of the calibration text file.

The CONTROL group at the head of the calibration file is used only to specify a default index mapping. If the IF channel orders in the calibration file and the *uv* file are identical it is not required.

The default *b*-factor and source flux densities are not specified in the `ANTAB` input file. If source flux densities are required by `APCAL`, the source (`SU`) table will be searched, into which flux densities can be inserted using `SETJY` if necessary.

`ANTAB` permits specification of IF-dependent and tabulated gains; the format description may be found by typing `EXPLAIN ANTAB`.

`ANTAB` can be run multiple times to append to the same `TY` and `GC` tables. Also, calibration files from separate antennas (*e.g.*, VLA) which have $T_{\rm sys}$ data tabulated in a different format can be concatenated and processed in one run. In this case the INDEX keyword must be specified for each antenna to fix the data format.

Note that ANTAB will (usually) ignore calibration data for which there are no corresponding *uv* data (see the help file for ANTAB). There is one exception however: calibration data for an antenna that does not appear in the AN table will cause ANTAB to crash. If ANTAB crashes under such circumstances, you have two choices. You can edit the calibration text file, removing all reference to the missing antennas; or you can use the input adverb SPARM to specify explicitly the names of antennas for which there are calibration data, but which do not appear in the AN table.

*** The use of SPARM is no substitute for careful inspection of the calibration text files. ***

Having created the input text file, typical inputs for ANTAB would be:

> TASK 'ANTAB' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INFILE 'MYVLB:BC25CAL.VLBA' C_R	to specify the text file.
> SUBARRAY 1 C_R	select subarray one.
> TYVER 0 ; GCVER 0 C_R	create new TY and GC tables.
> BLVER 0 C_R	create new BL table for any specified baseline factors.
> PRTLEV 1 C_R	select print level.
> GO C_R	to run the program.

PRTLEV = 2 will echo the calibration file as it is processed, which can be useful in locating format errors.

9.4.2.6 Loading phase cal information

For data from the VLBA correlator, phase-cal measurements for the VLBA antennas and a few others can be extracted from the .CAL file using VLOG (see § 9.4.2.1) and then loaded using task PCLOD into a PC table.

9.4.3 Data editing

Before proceeding to calibrate, you should first flag any obviously bad data. In summary, initial editing is based on the flagging information supplied by the on-line antenna monitor systems, which is applied using UVFLG. This information may be extracted to a .FLAG file as outlined in the previous section or subsequent editing based on the station report logs, or elevation limits can also be performed using UVFLG. Finally, graphical editing tasks such as IBLED and EDITR may be used for interactive baseline-based editing. Until the data are converted into single-source data format, flagging information is stored in the FG table instead of being used to directly discard data. This allows for the possibility of 'un-flagging' using OPCODE='UFLG' in UVFLG. Flag tables may also be used with single-source files in many cases.

To edit *uv* data by reading a text file listing periods of known errors, *e.g.* the .FLAG text file created by VLOG, run UVFLG with the following inputs:

> TASK 'UVFLG' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCES '' C_R	generally none is desired for this operation.
> SUBARRAY 0 C_R	select the required subarray.
> FREQID -1 C_R	apply to all frequency IDs.
> APARM 0 C_R	ignored.
> FLAGVER 1 C_R	specify output flag table.
> INFILE 'MYVLB:BC25CAL.FLAG' C_R	to specify the text file.
> GO C_R	to run the program.

This will generate a FG table with entries read from $MYVLB:BC25CAL.FLAG.

To edit *uv* data based on elevation limits, UVFLG can be used with input parameters:

> TASK 'UVFLG' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> SOURCES 'DA913' CR	source selection.
> SUBARRAY 1 CR	select the required subarray.
> ANTENNAS 3, 4, 5, 8 CR	antenna selection.
> APARM(4) 10 CR	elevation limit of 10 degrees.
> FLAGVER 1 CR	specify output flag table.
> GO CR	to run the program.

This will generate a FG table flagging time ranges that fall below the specified elevation limit.

Note that in both of these examples, the flagging information is incorporated into a FG table instead of irrevocably deleting your data. In order to apply these flags, you must set FLAGVER to the appropriate FG table version.

Another editing task which may be useful is QUACK, which can edit, say, the first 30 to 60 seconds after each source change. This might be needed because (at non-VLBA antennas) the telescopes were still slewing or system temperature measurements were being made. The first 20 seconds or so of a scan, for baselines to the VLA, are often unusable while the VLA correlator phases up at the new source position. QUACK can be used to flag specific antennas, non-VLBA or VLA only, for these and other reasons.

Tasks EDITA and EDITR can be used to inspect and edit the data interactively (EDITR is similar to the older task IBLED which is no longer recommended). These tasks are similar in many respects to TVFLG (*q.v.*) but are more suited to interferometers with small numbers of baselines. It is also possible to use TVFLG to perform your editing although the TV display is somewhat confusing on sparse arrays of data, especially if there is significant source structure. Read § 4.4 through § 4.4.2 and § 5.5 for more details on the editing of data.

If you flag a large fraction of the data set, you may wish to run UVCOP to discard the flagged data and conserve disk space.

9.4.4 *a priori* calibration

The calibration steps described in this section use *a priori* information about the performance of the antennas. These should be performed before any other *a posteriori* calibration. The order of these different calibrations is not particularly important, but probably they should be done in the order listed here.

The voltage threshold levels in the digital samplers at the antennas may differ from their optimum theoretical values, and this may vary from antenna to antenna. This sampler bias, which is usually only significant in two bit quantization, introduces an antenna-based amplitude offset. In full polarization observations this appears as an amplitude offset between RR and LL. The amount of the offset depends upon the detailed workings of the correlator used. For data from the VLBA correlator, if FITLD was run with DIGICOR = 0, 1 or 2, task ACCOR can be used to remove these digital sampler biases.

Starting with the 15JUL95 version of *AIPS* the recommended method of amplitude calibration using *a priori* $T_{\rm sys}$ and antenna gain information is to use the task APCAL. This task offers greater flexibility in amplitude calibration, including corrections for atmospheric opacity, than ANCAL which is now no longer supported.

The RCP and LCP feeds on each antenna will rotate in position angle with respect to the source during the course of the observation for Alt-Az antennas (which probably constitute a majority of the antennas in your observation). Since this rotation is a simple geometric effect, it can be corrected by adjusting the phases without looking at the data. This correction must be performed before any phase calibration which actually examines the data is executed. This correction is important for polarization observations. This correction can also be important for some phase referencing observations, depending upon the distribution of calibrators and targets on the sky.

9.4.4.1 Digital sampler bias corrections for VLBA correlator data

*** ACCOR is only intended to run on data from the VLBA correlator. ***

Typically the amplitude offsets for the VLBA correlator are of order $5-10\%$ but values as high as 20% have been observed. These can be corrected by examination of the autocorrelation spectra using task ACCOR. Since ACCOR computes the necessary correction by examining the total-power spectra, it must be run immediately after FITLD in the sense that nothing else should be run that actually modifies the total-power spectra directly. Although ACCOR ignores any SN or CL tables that are present, it is essential to correct for the sampler biases before performing any *a posteriori* calibration, *e.g.*, fringe-fitting or self-calibration.

Typical inputs to ACCOR for this correction would be:

> TASK 'ACCOR' ; INP C_R	to review the inputs.
> INDISK *n*; GETN *ctn* C_R	to specify the input file.
> TIMERANG 0 C_R	select all data.
> SOLINT 2 C_R	solution interval for sampler corrections (min).
> GO C_R	to run the program.

The correction factors are expected to be fairly stable over time so a few-hours long solution interval may also be used. With a shorter solution interval, such as the two minutes indicated here, it is well to examine the solution (SN) table generated by ACCOR using SNPLT for any bad points or inconsistent values. One approach is to inspect the SN table and then run SNSMO with clipping to get rid of discrepant points. Alternatively, the new interactive table editing task SNEDT can be used.

For one-bit quantization, no significant sampler bias correction is expected. In this case if the ACCOR gain factors deviate from unity this may point to overall scaling or *b*-factor errors. In this respect it is recommended that ACCOR be run on one bit data as a consistency check.

9.4.4.2 Continuum amplitude calibration

The TY table can be examined using SNPLT or LISTR with OPTYPE='TSYS' or 'TANT', and the GC table using PRTAB. Inspection of the TY table may reveal anomalously high system temperatures, indicating possible bad data (*e.g.*, due to weather). Note that flagging bad data will not change the appearance of the TY table since no flags are applied in plotting this table. APCAL can then be used to derive an amplitude calibration (SN) table. Typical inputs to APCAL are:

> TASK 'APCAL' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> ANTENNAS 0 ; SUBARRAY 0 C_R	default selection adverbs.
> SOURCES ' ' ; STOKES ' ' C_R	
> BIF 1 ; EIF 0 ; FREQID 1 C_R	
> TIMERANG 0 ; OPCODE ' ' C_R	no opacity solution required.
> TYVER 0 ; GCVER 0 C_R	use latest created TY and GC tables.
> SNVER 0 C_R	create new SN table.

> GO C_R	to run the program.

The resulting solution (SN) table can be smoothed/clipped using SNSMO and applied using CLCAL as described in § 9.4.1.1). Substantial smoothing of the TY table, especially for VLBA-only observations, is not generally recommended since variations of the system temperature often reflect a real response to the weather. Smoothing can be useful for data from the phased-VLA, when the amplitude calibration information reflects low-signal-to-noise. For non-VLBA antennas, it is important to check with SNEDT that the TSYS information is associated with the correct source.

9.4.4.3 Polarization calibration: parallactic angle corrections

For full polarization experiments, the parallactic angle contribution to the phase should be removed at the very start of the phase calibration. The parallactic angle correction is made using task CLCOR (Note that CLCOR directly modifies the specified version of the CL table so that you may wish to first make a backup copy of the CL table using TACOP:

> TASK 'CLCOR' ; INP C_R	to review the inputs.
> INDISK *n1* ; GETN *ctn* C_R	to specify the input file.
> OPCODE 'PANG' C_R	to select the parallactic angle correction.
> CLCORPRM 1,0 C_R	to specify the sense of parallactic angle removal.
> GAINVER *clin* C_R	CL table to update, NO DEFAULT!
> GO C_R	to run the program.

This will directly modify the specified version of the CL table.

9.4.5 Bandpass calibration

Full bandpass response calibration is usually only performed for spectral-line observations although it can be important for certain types of continuum obsrvations.

A quick and dirty bandpass calibration can be carried out for most types of continuum observations by throwing away the outer channels in the data set since these are the most affected by the bandpass response function. The number of channels to throw away should be determined by examining the bandpass effects on the total-power spectra and carefully weighing the competing considerations of signal-to-noise loss by tossing channels versus non-closing errors by keeping channels. This procedure of tossing channels may not be as good as performing an actual bandpass, but it is simpler and easier to carry out.

It is suggested that integrating over a variable bandpass function is one of the most significant sources of non-closing errors in continuum VLBI data. By calibrating the bandpass before averaging over frequency, these effects can be avoided. In VLBA test observations, a dynamic range of 28,000:1 was achieved on the source DA193 (Briggs *et al.* 1994, VLBA Memo No 697A, "High Dynamic Range Imaging with the VLBA") after applying bandpass calibration.

Bandpass calibration is carried out using the task BPASS using either the auto- or cross-correlation data. The output is a BP or bandpass table. The derived bandpass solutions can be plotted using POSSM by setting APARM(8)=2. The effect of applying these bandpass solutions to your data can also be viewed using POSSM by setting DOBAND=1 and BPVER. An example of the inputs to produce bandpass spectra from the auto-correlation data would be:

> TASK 'BPASS' ; INP C_R	to review the inputs.
> INDISK *n1* ; GETN *ctn1* C_R	to select the multi-source visibility data as the input file.
> CALSOUR 'BLLAC','DA193' C_R	to specify the continuum source(s) which were observed for the purpose of bandpass calibration.

> BCHAN 0 ; ECHAN 0 CR	to use the whole spectrum.
> DOCALIB -1 CR	to not apply any calibration.
> SOLINT 0 CR	to average data over whole scans before determining the bandpass.
> SMOOTH 1, 0, 0 CR	to Hanning-smooth the spectrum.
> BPASSPRM 1, 0 CR	to use the self-spectra.
> GO CR	to run the program.

This will produce a `BP` table containing the antenna-based bandpass functions to be applied to the data. Since only the self-spectra were used, the phase response of the bandpasses is not determined. If you wish to correct for phase errors across the band, then you must first fringe-fit the calibrator data (see below), then set `DOCALIB` to 1 and `BPASSPRM(1)` to 0, and run the task. However, you should check your results very carefully. The `BP` tables can be plotted with `POSSM` or printed with `PRTAB`. Note that this procedure merely creates a `BP` table. To use this `BP` table, set `BPVER` and `DOBAND` as described in § 4.7.3 when running any later *AIPS* tasks.

9.4.6 Spectral-line doppler correction

Normally, when observing, you will have kept the frequency constant throughout the run for ease of observing and also because most modern video-converters (BBCs) can only be set to a precision of 10 kHz. Therefore, although your data will have the same frequency, the center velocity of your spectrum will change with time and the spectral-line signals will wander backwards and forwards through the spectrum. To ensure that the velocity is constant throughout the data you should run `SETJY` and then `CVEL`.

`SETJY` will insert the velocity information required in the `SU table`:

> TASK 'SETJY' ; INP CR	to review the inputs.
> INDISK *n1* ; GETN *ctn1* CR	to select the data.
> SOURCE 'OH127.8' , ' ' CR	to specify the line source whose velocity is to be specified.
> OPTYPE ' ' CR	to switch off flux modification.
> SYSVEL -66.0 CR	to specify the velocity of the "center" of the band in km/s.
> APARM 65, 0 CR	to specify which spectral pixel is the "center" of the band, actually the pixel to which `SYSVEL` refers.
> RESTFREQ 1612e6,231.09e3 CR	to give the rest-frequency in Hz, *e.g.*, that of the OH transition.
> VELTYP 'LSR' CR	to select the rest frame of the velocity.
> VELDEF 'OPTICAL' CR	to define velocities by the optical convention.
> GO CR	to run the program.

Then run `CVEL`:

> TASK 'CVEL' ; INP CR	to review the inputs.
> INDISK *n1* ; GETN *ctn1* CR	to select the data.
> OUTDISK 3 ; OUTCLASS 'CVEL' CR	to specify the output file.
> SOURCE 'OH127.8' , ' ' CR	to select the source(s) to be shifted, all others will be passed un-shifted.
> DOBAND 1 CR	to apply the bandpass correction — important.
> BPVER 1 CR	to specify the version of `BP` table to use.
> GO CR	to run the program.

After applying the `BP` table, `CVEL` will not copy it to the output file to protect you from applying it twice. Although `CVEL` allows you to select which sources are to be shifted, the `BP` table, if `DOBAND` is set appropriately, will be applied to all sources found.

9.4.7 Spectral-line amplitude calibration

The calibration scheme suggested in *AIPS* for spectral-line VLBI data utilizes the total-power spectra method[1], although the continuum method using $T_{\rm sys}$ values (see § 9.4.4.2) can also be used. The first step is to generate a so-called template spectrum. This is a high quality spectrum from the most sensitive antenna in the array that has been corrected for the effects of the bandpass filter. For example:

> TASK 'SPLIT' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> SOURCE 'OH127.8', ' ' CR	to write the program source.
> BCHAN 0 ; ECHAN 0 CR	to write all spectral channels.
> DOCALIB -1 CR	to avoid applying any calibration.
> DOBAND -1 CR	to skip the bandpass correction since the data will have been corrected already in CVEL.
> TIMERANG 0 22 0 0 0 22 30 0 CR	to select the data from a range of times when the antenna elevation was high and the source spectrum of high quality.
> APARM 0, 0, 0, 0, 1 CR	to pass only self-spectra.
> GO CR	to run the program.

You should then run ACFIT to do a least-squares fit of the template total-power spectrum to the total-power spectra of all other antennas and to write the resulting time-dependant amplitude gain correction factors into an SN table.

> TASK 'ACFIT' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> IN2DISK *n* ; GET2N *ctn* CR	to specify the template file.
> CALSOUR 'OH127.8 ' CR	to select the source to use for calibration.
> DOCALIB -1 CR	to avoid applying any previous calibration.
> DOBAND -1 CR	to skip the bandpass correction since it was done when CVEL was run.
> SOLINT *n* CR	to average the self-spectra over *n* minutes (*e.g.*, 10) before doing the least-squares fit.
> REFANT 1 CR	to select the desired reference antenna from the template file.
> BCHAN 50 ; ECHAN 70 CR	to set the range of spectral channels over which the fit is performed.
> APARM 0, 0, 50, 0, 0.72 CR	program control: APARM(1) and APARM(2) specify the orders of the polynomial spectral baseline to remove from the source and template spectra; APARM(3) and APARM(4) specify the sensitivity of the template antenna (in Jy/deg) in the first and second (if needed) polarizations; APARM(5) and APARM(6) specify the minimum and maximum relative antenna gains allowed, with defaults to allow all positive values; APARM(7) specifies the maximum allowed gain error, with 0 meaning all; APARM(8) specifies the print level, with 0 providing minimal information, 1 providing useful information on the gains determined for each antenna and solution interval and 2 giving the gory details for each fit; APARM(9) specifies that the fits are done after subtracting a spectral baseline (0) or without a baseline (1); and APARM(10) controls whether baseline-subtracted spectra are written to an output file.

[1] see Lecture 12 in *VLBI, Techniques and Applications*, eds. Felli and Spencer, published by Reidel

> BPARM 80, 120 C_R	to set up to 5 pairs of start and stop channels to use in determining the baseline polynomial to be removed from the source spectra. The order of the polynomial is specified in APARM(1).
> CPARM 80, 120 C_R	to set up to 5 pairs of start and stop channels to use in determining the baseline polynomial to be removed from the template spectra. The order of the polynomial is specified in APARM(2).
> XPARM 45.1, 48.0, 50.1, 49.5 C_R	to specify $T_{\rm SYS}$ values in the first polarization for each IF for the template scan of the reference antenna. YPARM provides an equivalent list for the data, if any, from a second polarization.
> SNVER 0 C_R	to create a new SN table into which the solutions are to be written.
> GO C_R	to run the program.

ACFIT will generate an SN table, which has to be applied to the CL table.

If needed, run SNSMO to smooth the amplitude correction factors determined by ACFIT before running CLCAL. A 30 minute smoothing interval for SNSMO (set using CPARM 0.5,0) should be sufficient.

9.4.8 Phase calibration

After carrying out amplitude calibration, the remaining calibration steps are involved with calibrating the phase of the data within and between the separate IF channels and between the different integration periods to the integrate the data over both frequency and time without loss of coherence. Such phase offsets may be corrected using *a priori* 'phase-cal' measurements if available and/or by directly fitting to the data.

For those antennas for which phase-cal measurements are available, task PCCOR can be used to incorporate the phase-cals into an SN table. (see § 9.4.8.5). If you have other anntennas for which phase-cal measurements are not available, you can run CLCAL using the OPCODE='CALP' option to incorporate the incomplete SN table information loaded by PCLOD without throwing away data for the antennas with missing phase-cal information.

For MkIII data from the BONN correlator, phase-cal measurements are incorporated directly into the first CL table produced by MK3IN — this is another strong reason to protect the first CL table.

If external pulse-calibration data are not available then directly fringe-fitting a short scan of data to measure the phase and single-band delay offsets may be applicable under limited circumstances (see § 9.4.8.6).

Even when pulse-calibration information is available, performing a manual phase-cal can be a good idea to confirm that the IF-dependent delays and phases have been successfully estimated and removed. One good check is to carefully inspect the data using POSSM at times different from the time used to determine the manual phase calibration.

Even after removing instrumental phase offsets from each IF, the data will in general still contain frequency and time dependent phase variations. The purpose of "fringe-fitting" is to determine these phase errors and then remove them from the data.

The primary *AIPS* task for fringe-fitting is FRING. This task estimates time variable station-based delays (phase derivatives wrt frequency) and rates (phase derivatives wrt time) using a self-calibration-like algorithm. Once these delays and rates are determined, the task CLCAL is used to produce the phase correction that should be applied to each integration period and spectral channel to correct for delay and rate effects. This use of `FRING` and `CLCAL` is discussed in detail in § 9.4.8.7. Two alternatives to `FRING` are the tasks `BLING` and `BLAPP` and the experimental version of `FRING`, `KRING`. `BLING` and `BLAPP` are discussed briefly in § 9.4.8.8. `KRING` provides a superset of the functionality in `FRING` with numerous enhancements such as: the use of extremely small scratch files, a parsimonious use of memory, possible solution extrapolation both backwards and forwards in time and a rationalized definition of SNR. For more information about `KRING`, type `HELP KRING` from within *AIPS*. `KRING` is generally recommended instead of `FRING` but especially should be used if `FRING` crashes due to lack of disk space or memory. When fringe-fitting to many small scans, `KRING` can be substantially slower than `FRING`. When fringe-fitting data sets with large numbers of spectral channels and long solution intervals, `KRING` can be substantially faster than `FRING`.

*** The process of fringe-fitting, and then interpolating the solutions using `CLCAL`, can be a very time consuming process. Although it depends a lot on the size and structure of the data set, the fringe-fitting time can equal or exceed the observing time for a large data set. For this reason it is probably wise to run through the fringe-fitting procedure described in § 9.4.8.7 on a small amount of data first (say 30 minutes' worth) before attempting to process the whole data set. This is especially true if this is your first time processing multi-IF, multi-channel VLBI data. It is probably simplest to use `UVCOP` to copy out a short time range of data from your main file and to work only on this initially. Doing so also avoids the possible confusion of having many versions of extension tables. ***

9.4.8.1 Special considerations: SVLBI

The existing fringe-fitting tasks within *AIPS* have been enhanced to improve their performance when dealing with SVLBI data. In addition, several new tasks have been written to address several problems specific to SVLBI fringe-fitting. The primary SVLBI fringe-fitting tasks in *AIPS* are `BLING`, `GPHAS`, and `FRING` and are discussed in § 9.4.8.7–§ 9.4.8.9. The tasks `COHER` and `FRPLT`, described in § 9.3.6 and § 9.3.7, may be of particular interest when reducing SVLBI data.

There may be delay discontinuities in the recorded data for a variety of reasons such as tracking station handoffs, clock glitches, etc. The recommended method for dealing with such discontinuities is to force scan broundaries at such events. The task `INDXR` can be used to generate a new `NX` table with scan boundaries at desired locations using an input text file. In practice, either `INDXR` will do the right thing by design, or your P.I. information letter should have contained instructions on how to construct a text file in the proper format for `INDXR`.

A new task, `OBEDT`, is available which allows selection of specific orbital parameter ranges, through the creation of an output flag (`FG`) table. This can be used to constrain initial fringe searching.

If you have large, known, clock offsets for the spacecraft, you may wish to run `CLCOR` as described in § 9.4.8.6.

9.4.8.2 Special considerations: spectral-line

Delay and fringe-rate calibration of spectral-line VLBI data must be handled differently. The residual delay cannot be estimated from the source itself because, due to the very nature of the source, the delay is a rapidly varying function of frequency. The continuum calibrator, observed for this purpose, is first used to determine residual delays and fringe-rates which are then applied to the spectral-line source. A suitable channel or range of spectral-line channels "on-source" is then used to determine the residual fringe-rates. It is very important to note that in some situations, the residual fringe-rates determined from the calibrator may not be applicable to the line source because the fringe-rate residuals towards the two sources may be quite distinct. In such situations, the residual fringe-rates determined from the continuum source should not be applied to the line source. See § 9.4.8.10 for more details. It is not clear that VSOP will observe a continuum calibrator.

9.4.8.3 Special considerations: polarization

In addition to phase calibrating the LL and RR data separately, for polarization data the R-L phase and delay offsets must also be determined. This is outlined in § 9.4.8.11. After fringe-fitting all parallel-hand fringe solutions need to be re-referenced to the same antenna.

9.4.8.4 Special considerations: phase-referencing

The process of phase referencing for VLBI data is conceptually very simple. Unfortunately, the technical difficulties in conducting a successful phase-referencing observation are primarily in setting up the schedule. So by the time you get around to reading this section, your project is either guaranteed to succeed or guaranteed to fail, depending upon how well your observations were designed. See the lecture by Beasley and Conway in "VLBI and the VLBA", 1995, (ASP) for more details on how to design phase-referencing observations.

The key principle of phase referencing is that the phase calibration for your target source is derived either from *a priori* information, such as phase-cals, or from a cal, or phase referencing, source especially observed for that purpose. The procedure is to first apply any available *a priori* phase-cals to both the target and cal source. Next, you fringe-fit, self-calibrate, and/or hybrid-map the cal source — whatever is needed to complete the phase calibration for that source. Finally, you apply the phase corrections determined by fringe-fitting, self-calibration, and/or hybrid mapping the cal source to the target source. In practice this is done by, whenever an `SN` table containing phase corrections for the cal source is applied using `CLCAL`, appropriately specifying the target in the `SOURCES` adverb list. See § 9.4.1.2 for a specific example of how to run `CLCAL` to transfer phase corrections determined using a cal source to a target source.

You can perform a 'hybrid' form of phase referencing in some instances. It may be that your target source is too weak for intial fringe-fitting. In this case, you can fringe-fit the cal source data to determine the phase corrections to be applied to the target source data, and then average up the latter data in frequency. At this point, the target source may be strong enough to determine rate corrections by fringe-fitting on it. In this mode, you may or may not wish to zero the rate corrections determined on the cal source. If the cal source is 'far' from the target source, the rate corrections may do more harm than good for the target source and should be zero'ed. On the other hand, your target source may be entirely too weak to fringe-fit on at all. In this case, you must rely on determing the phase corrections solely using your cal source.

If you are attempting phase-referenced astrometry, you may have a target source that is brighter than your cal source(s). In this case, you simply fringe-fit on the target source and transfer the solutions to the cal source(s).

Be careful, if your goal is to extract absolute positional information, not to independently self-calibrate the cal and target sources. For more information on various types of phase-referencing strategies, see *http://www.nrao.edu/vlba/html/phase-referencing.html*.

9.4.8.5 Instrumental phase corrections

If you run `POSSM` on a short (≈ 1 minute) section of data on a strong calibrator using the inputs described in § 9.3.2 and set `DOCAL = -1` C_R, you will see that each individual IF channel has its own independent phase offset and its own phase gradient against frequency. These phase offsets and instrumental "single-band delays" are caused by passage of the signal through the electronics of the VLBA baseband converters (or MkIII/MkIV video converter units). The VLBA and MkIII systems can inject narrow band signals ("phase-cals") into the data recorded at each antenna from which the IF channel phase offsets, and possibly the instrumental single-band delays, can be determined.

Data produced by a MkIII correlator are provided with the phase offsets in version 1 of the CL table (see § 9.2.2). To check whether your data from a MkIII correlator contains phase-cal data, use SNPLT on CL table version 1 to plot the phases. If the values are non-zero, phase-cal data are present.

As of 1 April 1999, the VLBA correlator is capable of transferring phase-cal information directly to a PC table for some antennas. For other antennas, the phase-cal information may be read into the PC table using the task PCLOD. Type EXPLAIN PCLOD for further information.

Pulse-calibration data in a PC table can be used by task PCCOR to generate a SN table which corrects for the single-band instrumental phase and delay offsets (note that PCCOR uses two pulse-cal's in each IF). A short calibrator scan must be specified which is used by PCCOR to resolve any 2π phase ambiguities in the pulse-calibration data. The specified time range must include at least one PC table entry for each antenna appearing in the PC table. Typical inputs to PCCOR are:

> TASK 'PCCOR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> TIMERANG 1 2 15 0 1 2 20 0 C_R	short scan on a calibrator.
> SNVER 0 C_R	to create a new SN table.
> INVER 1 C_R	input PC table version.
> REFANT 5 C_R	specify reference antenna.
> SUBARRAY 1 ; FREQID 1 C_R	set subarray and freqid.
> CALSOUR '3C345',' ' C_R	set calibrator source name.
> GO C_R	to run the program.

The resulting solution table is applied using CLCAL. If you are missing phase-cal information for some antennas, you must use the 'CALP' mode of CLCAL; this mode allows calibration information for some antennas to be incorporated into the CL table while passing other antennas through without modification. Examine the corrected data using POSSM to determine if the instrumental phase and delay offsets between the IF channels have been removed correctly.

To check that the applied phase-cals are valid, run POSSM on a short section of data containing a strong source setting DOCAL = 1, GAINUSE to the version number of the CL table containing the phase-cal calibration, and APARM(9) = 1 to place all IFs on the same plot. The phase as a function of frequency on each baseline should be smoothly varying, with no sharp jumps between different IF channels. There may be an overall gradient between channels and small gradients within each channel (which should be approximately the same from IF channel to IF channel, but not necessarily the same as the gradient between channels). Unless these conditions hold for all baselines, you should proceed to § 9.4.8.6.

9.4.8.6 "Manual" instrumental phase corrections

If your file does not have phase-cal information, or if these phase-cals do not successfully remove frequency phase offsets, you can use observations of a bright calibrator source and the task FRING to correct for these effects. If you attempted phase-cal calibration, it is best to avoid possible confusion by first deleting any partial or erroneous phase-cal information that already exists. Using CLCOR:

> TASK 'CLCOR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> GAINVER *clin* C_R	to specify which CL table to modify.
> OPCODE 'PCAL' ; CLCORPRM 0 C_R	to set phase-cals to zero.
> INP C_R	to check the inputs.
> GO C_R	to run the task.

If you have large known delay offsets you may also wish to run CLCOR using OPCODE='SBDL' to shift the center of the fringe search window.

If you have a known clock offset, as may be common for VSOP data, you may wish to run CLCOR with OPCODE = 'CLOC' and CLCORPRM = 0,*offset*,0,0,0,0,1,0.

Now you can determine the phase offsets/single band delays by running FRING on a short section of calibrator data where all or most of the antennas are present. Suitable inputs for FRING for this purpose are shown below; for more details of some of the FRING input parameters see § 9.4.8.7. Note that it is simplest to choose a single short section of data using TIMERANGE and to set SOLINT equal to this interval so that a single solution is achieved. The interval chosen must be less than than an atmospheric coherence time, but long enough that high signal-to-noise is achieved. At centimeter wavelengths with Jansky-level calibrators, solution intervals of a few minutes will work well. For example:

> TASK 'FRING' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR '0954+658 ' C_R	to select a strong calibrator source.
> TIMERANGE 0,16,0,0,0,16,2,0 C_R	to select a single short scan.
> DOCALIB 2 ; GAINUSE 2 C_R	to apply the amplitude calibration from CL table 2 to the weights as well as the visibilities.
> FLAGVER 0 C_R	to apply most recent flag table.
> SMODEL 0 C_R	to use the null (point-source at the origin) source model.
> REFANT 1 C_R	to specify a reference antenna that will give fringes to most other antennas.
> SOLINT 2 C_R	to set the solution interval in minutes; do not exceed the atmospheric coherence time.
> APARM 0; DPARM 0 C_R	to initialize FRING options to defaults.
> APARM(1)=2 C_R	to require at least 2 antennas.
> APARM(6)=1 C_R	to solve for the rate, single-band delay and phase of each IF separately.
> DPARM(1)=1 C_R	use only 1 baseline in the initial (coarse) fringe search.
> DPARM(8)=1 C_R	to zero the fringe rates before writing the SN table.
> SNVER 0 C_R	to create a new SN table.
> ANTWT 0 C_R	to apply no additional weights to the antennas before doing the solutions.
> INP C_R	to check the inputs.
> GO C_R	to do the fit.

If there were no single scan where all the antennas were present, you can run FRING again for another scan setting REFANT to be one of the antennas found in the first run and ANTENNAS to this antenna plus all of the antennas *not* found in the first run. FRING will generate a new SN table each time; be careful to keep track of which SN tables you wish to use.

The phase solutions in the SN table(s) are interpolated onto a calibration or CL table using task CLCAL as described in § 9.4.1.1. Be sure to set CALSOUR to the calibrator source used when running FRING (in the example above, CALSOUR '0954+658''') and to set SOURCE = '' — this is how the calibrator solutions are applied to the other sources in the data set. Also, set REFANT to whatever was used when running FRING. If multiple runs of FRING were required, you can set SNVER=0 so that all SN tables are combined before being applied (you must first be careful to delete all SN tables except those generated by FRING if you do this). Or you can run CLCAL multiple times specifying each SN table in turn, with the specific antenna numbers, while keeping the same CL table versions.

To check the output CL table, run POSSM for the scan used for the FRING solution with DOCAL = 1 and

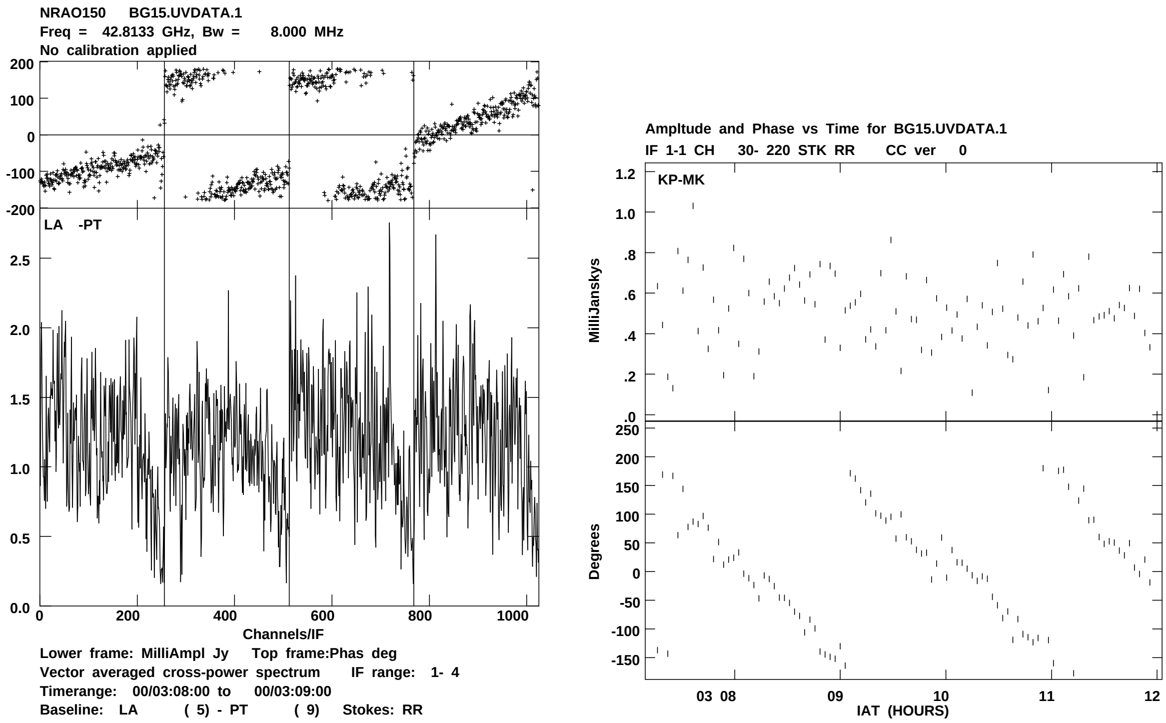

Figure 9.2: *left:* A POSSM plot of the 43-GHz spectrum of the quasar NRAO150 on the Los Alamos to Kitt Peak baseline. The plot shows that the observation was performed with 4 IFs, with 256 spectral channels within each IF. The upper frame shows the phase variation with frequency; within each IF, the small phase slope is caused by a residual delay error. The phase offsets between the IFs can be clearly seen as well. Both the residual delay error and the phase offsets must be determined and removed before the data can be spectrally averaged (see § 9.4.8). *right:* A VPLOT plot of the amplitude (upper frame) and phase (lower frame) as a function of time for the source NRAO150 at 43 GHz on the baseline Kitt Peak to Mauna Kea. Note how the phase varies as a function of time; this variation is equivalent to a residual fringe rate of 8.3 mHz. Unless the fringe rate is determined and removed (see § 9.4.8), the data cannot be averaged in time.

GAINUSE = 3 (*i.e.*, the output CL table from the CLCAL above). The phase should be flat as a function of frequency on all baselines. It is also useful to run POSSM on another scan containing a strong calibrator to check that the assumption of constant IF phase offset holds.

9.4.8.7 Antenna-based fringe-fitting

To see an example of the residual phase errors in your data, use POSSM to view the phase on a short calibrator scan (at some time other than that used to solve for the phase-offsets in § 9.4.8.6. In general, there will be a gradient in phase between the IFs (due to the "multi-band" delay) and also small gradients within each IF (caused by small residual "single-band" delays). These time-variable phase gradients are mainly due to inaccuracies in the geometrical time delays that the correlator assumed for the time of arrival of the wavefront at each antenna. These inaccuracies arise from propagation effects through the troposphere and ionosphere, inaccurate Earth geometry, etc. and give *phase* errors which are proportional to frequency. Such phase errors prevent integration of the data over frequency (or cause a loss of coherence if you do). Similarly, VPLOT will show, on any single IF and spectral channel, phases which change rapidly with *time*. Again, these are due to unavoidable inaccuracies in the correlator model; such large "phase rates" prevent integration over time. Both of these plots are illustrated in Figure 9.2.

You will want to run FRING to correct for these residual rates. FRING uses a global fringe-fitting algorithm described by Schwab and Cotton, 1983, *Astron. J.*, **88**, 688. Suitable inputs for the fringe-fitting task FRING are:

> TASK 'FRING' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR ' ' C_R	to find solutions for all sources.
> TIMER 0 C_R	to find solutions for all times.
> DOCALIB 2 C_R	to apply the most complete calibration file including amplitude calibration and IF and channel phase offsets to both the visibilities and the weights.
> GAINUSE 3 C_R	to use CL table 3.
> SNVER 2 C_R	to write solutions into SN table 2.
> BCHAN 0; ECHAN 0 C_R	to use all spectral channels within each IF channel.
> SMODEL 0 C_R	to use a point-source at the origin model for the sources.
> CLR2NAME C_R	to not use a Clean-component model.
> FLAGVER 0 C_R	to apply the most recent flag table.
> REFANT 1 C_R	to choose an antenna that will give fringes for most of the scans. This is important: FRING will search for fringes to this antenna first. If it fails for some reason, it will select another reference antenna, based on the ANTWT data, and, if it still fails, give up (see however the discussion of the new SEARCH adverb below). In this case, you should look for scans with no fringes or a bad reference antenna may be causing the problem. A big, sensitive antenna is often used as REFANT (*e.g.*, Effelsberg). Occasionally, it may be helpful to split your data set up into 2 or 3 sections, which are fringe-fitted with different REFANT (*e.g.*, a "European" and a "US" part of the observations). Changes in reference antenna should, in general, not cause problems.
> SEARCH *ref2, ref3, ...* C_R	if APARM(9) is set, the antennas specified via SEARCH augment the antenna specified via REFANT allowing FRING to search for fringes more exhaustively (see the HELP file under APARM(9)). The use of APARM(9) and SEARCH makes FRING much more likely to find fringes to weak antennas and is highly recommended.
> ANTWT 0 C_R	to apply no additional weights to the antennas before doing the solutions. If the amplitude calibration was incorrect, you can use this option to force antenna weights up or down to control the weight FRING gives to data to each station when making the global solutions. Unlesss the SEARCH option described above is chosen, ANTWT also controls the order in which antennas are tried as secondary reference antennas after failing to find fringes on the REFANT. Give higher weight to antennas you want to see used as secondary references.
> SOLINT 3 C_R	to set the solution interval in minutes; do not exceed the atmospheric coherence time (see below). Setting SOLINT to 0 sets solution intervals equal to scan lengths.
> APARM(1) 2 C_R	to accept solutions when only 2 antennas are present; default is 6.

> APARM(2) 0 C_R	to have the data divided by the model before fitting fringes; `APARM(2)` > 0 tells `FRING` that the data have already been divided by a model.
> APARM(3) 0 C_R	to treat polarizations separately; `APARM(3)` > 0 averages RR and LL.
> APARM(4) 0 C_R	to not average frequencies within each IF before the solution.
> APARM(6) 1 C_R	to get some useful, but limited, messages, including the SNR.
> APARM(7) 9 C_R	to avoid false detections by setting a moderately high minimum for the SNR accepted. You may wish to use a lower threshold (especially for SVLBI) although APARM(7) less than about 3 is probably not useful.
> APARM(8) 0 C_R	to set the maximum number of antennas; used only if no `AN` table is present.
> APARM(9) 1 C_R	to enable the exhaustive search mode.
> APARM(5) 0 C_R	to do separate least-squares fits for single- and multi-band delays. `APARM(5)` ≤ 0 means to solve separately for the rate, single band delay and phase of each IF. 1.5 > `APARM(5)` > 0 means to solve for one single rate and multi-band delay affecting all IFs. If, as here, > 1.5, the task additionally solves for the difference between the multi-band delay and single-band delay, *i.e.*, it allows for a different gradient of phase versus frequency within an IF than between IFs. Note, however, that unlike `APARM(5)=0`, this option assumes that the single-band delay is the same in each IF; it therefore solves for a single value for the difference between multi-band and single-band delay affecting all IFs. Normally users should use `APARM(5)=2` for multi-IF data.
> ANTENNAS 0; DOFIT 0 C_R	Only baselines between antennas listed in the `ANTENNAS` adverb are used in the fringe search. If any antennas are specified in `DOFIT` however, a solution is made only for those antennas; all other selected antennas are assumed to be already calibrated and are passed through with no additional corrections. See the `HELP` file for `FRING` under `DOFIT`. **`DOFIT` is only active if `APARM(9)` is set.**
> DPARM(1) 1 C_R	to use one baseline combination in the initial coarse (FFT) fringe search. This provides a starting guess for the least-squares solution. If you are searching for weak fringes you should consider using two and three baseline combinations in the search; this can improve sensitivity in the initial fringe search. See Chapter 19 in *Synthesis Imaging in Radio Astronomy*, for an explanation of how this global multi-baseline searching works. Note that if your source structure is complex and you have not divided the data by an accurate source model, then setting this parameter to one is safest. FYI, `DPARM(1)>1` is implemented for unequal integration times.
> DPARM(2) 0 C_R	to set the full width of the delay window in nsec, centered around 0, to search; the default, chosen here, is to use the full Nyquist range defined by the frequency spacing. A smaller search window can improve the detected SNR but can also result in lost data due to failed fringe searches. For the VLBA, a 1000 ns window (`DPARM(2) = 1000`) is usually adequate.

> DPARM(3) 0 CR	to set the full width of the fringe-rate window in mHz, centered around 0; the default, chosen here, is to use the full Nyquist range defined by the integration time. A smaller search window can improve the detected SNR but can also result in lost data due to failed fringe searches. For the VLBA, a 200 mHz window (`DPARM(3) = 200`) is usually adequate.
> DPARM(4) 2 CR	to specify the correlator integration time in seconds; use `DTSUM` to find the correct value. For data from the VLBA correlator, `DPARM(4)=0` will cause `FRING` to determine the correct integration time by examining the data file directly.
> DPARM(5) 0 CR	to do both the coarse and the least squares solutions; set to 1 if you require only FFT solutions.
> DPARM(6) 1 CR	to keep, for single source files, frequencies separated in the output file; the default is to average frequencies within IFs. This parameter does not affect multi-source files.
> DPARM(7) 0 CR	to re-reference solutions to a common reference antenna; when processing polarization data, set this to 1 to avoid the re-referencing.
> DPARM(8) 0 CR	to disable the zero'ing options — see the `HELP` file. **WARNING, `DPARM(8)` > 0 will discard parts of the final solution. Be sure to use this option with extreme care.**
> INP CR	to check the inputs.
> GO CR	to do the fit — finally.

Note that `FRING` finds solutions in two steps. First approximate solutions are found in the FFT step using combinations of one, two or three baselines (see `DPARM(1)` above). Then, as long as `DPARM(5)` < `1`, a least-squares algorithm uses these approximate values as a starting point for refining the solutions.

Note that the `SOLINT` interval should be selected with consideration of the atmospheric coherence time, but must be long enough that high signal-to-noise-ratio solutions are achieved. For observations between 1.6 and 15 GHz, solution intervals of 3–6 minutes (and often longer) should be fine. At other frequencies shorter solution intervals may be required. In these cases, consult an expert or experiment with different length solution intervals on short sections of data. Note that solution intervals greater than the scan length will never be used; the scan lengths are listed in the `NX` table and may be examined using `PRTAB`.

If the source is complex, and especially if the visibility phase of the source changes during `SOLINT`, it is useful to divide the data by a Clean model derived from previous observations or from an earlier attempt at processing the data. This Clean model can be specified by filling in `IN2NAME` *et al.* Situations where this is useful include observations of equal doubles (where there are zeros in the amplitude and, hence, rapidly changing phases) or very large sources (of order arc-seconds). If you are using multi-baseline searching (*i.e.*, `DPARM(1) > 0`, then solutions may be more sensitive to source structure and an input model may be useful if the structure phases are larger than one radian on many baselines.

You should check the SNRs found by `FRING` carefully; they are printed if `APARM(6) > 0`. The SNRs estimated during the FFT search are used to determine if the SNR of a solution is ≥ the threshold set in `APARM(7)`. If they are not, then that solution is flagged before being passed to the more accurate least-squares routine. Users should check that the SNRs found in the LSQ routine match those expected. If the detected SNRs are too low, `SOLINT` may be too long or other parameters may be set wrongly.

Be warned that proper scaling of the SNRs by FRING depends upon whether or not the data weights have been properly calibrated. Task `FIXWT` may be used to calibrate the weights, but changing the SNR threshold for `FRING` directly (`APARM(7)`) usually produces satisfactory results.

The final delay and rate solutions and their SNRs should be inspected using `LISTR`:

> TASK 'LISTR' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OPTYPE 'GAIN' C_R	to list gain solutions.
> INEXT 'SN' ; INVER 2 C_R	to list `SN` table 2 (as above).
> DPARM 6, 0 C_R	to list delay; use 7 for rate, 1 for phase, and 8 for SNR.
> GO C_R	to run the program.

Alternatively, the solutions can be plotted against time to make sure they are sensible. Use `SNPLT`:

> TASK 'SNPLT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OPTYPE 'RATE' C_R	to plot rate solutions, `OPTYPE 'DELA'` for delay solutions.
> OPCODE ' ' C_R	to plot each polarization and IF in separate plots.
> NPLOTS 5 C_R	to plot five antennas/IFs/polarizations per page.
> INEXT 'SN' ; INVER 2 C_R	to plot `SN` table 2 (as above).
> GO C_R	to run the program.

It is a good idea to plot out the solutions for `OPTYP 'RATE'`, `'DELA'` (single-band delay), and `'MDEL'` (multi-band delay) as well as the associated `'SNR'`s. They should be smoothly varying functions. Delays and rates should be found only within your specified windows. Check for suspicious detections at the limits of the search windows — for instance, they could be detections of side lobes of the main fringe. If you used windows smaller than Nyquist, you may want to check your detections using bigger windows. Gaps in detections with time can occur if, *e.g.*, tapes were bad, antennas were off source, the source visibility is in a minimum, or the reference antenna choice was bad. It pays to investigate such problems at this point before proceeding.

Discrepant `SN` solutions can be removed using the interactive task `SNEDT` or the non-interactive task `SNSMO`. If, in the latter, the `CPARM` values are set, then the `SN` solutions will be clipped if they differ from the running mean by amounts which you can specify. If the `BPARM` values are set, the solutions are then smoothed and clipped entries can be replaced with mean values based on a boxcar- or median-window-filter average. See the explain file for details. Although `SNSMO` allows the option of modifying the input `SN` table, it is safest to have it create a new one. If you have a lot of discrepant `SN` values, you should also consider using option `INTERPOL = 'POLY'` in task `CLCAL` (see below).

If there isn't enough disk space to run `FRING` on all the data at once (because of the large scratch file that `FRING` currently insists on creating), you can run `FRING` multiple times specifying time ranges and explicitly setting `SNVER` to the same `SN` table.

If, for some reason, you set the parameters of `FRING` or `SNSMO` wrongly and the resulting `SN` table is unusable, it is wise to avoid confusion by deleting it using `EXTDEST` and starting over again.

Once a valid `SN` table has been produced, the next step is to interpolate the solutions found onto the finer grid of entries in a `CL` table using the task `CLCAL` with `INTERPOL = 'AMBG'` as described in § 9.4.1.1.

Once a final `CL` table is generated, its effect on the data can be viewed using tasks `VPLOT` and `POSSM` by setting `DOCAL=2` and `GAINUSE` to the version number of the final `CL` table; see § 9.3. Optionally, in `VPLOT`, one can average over spectral channels and/or IF channels before plotting. Use `VPLOT` to plot a time range covering a few `FRING` solution intervals on a strong source. Phase variations should be small with no jumps. If this is not the case, check the inputs to `FRING` (especially `SOLINT`) and `CLCAL`. A comparison of before and after phases is shown in Figure 9.3.

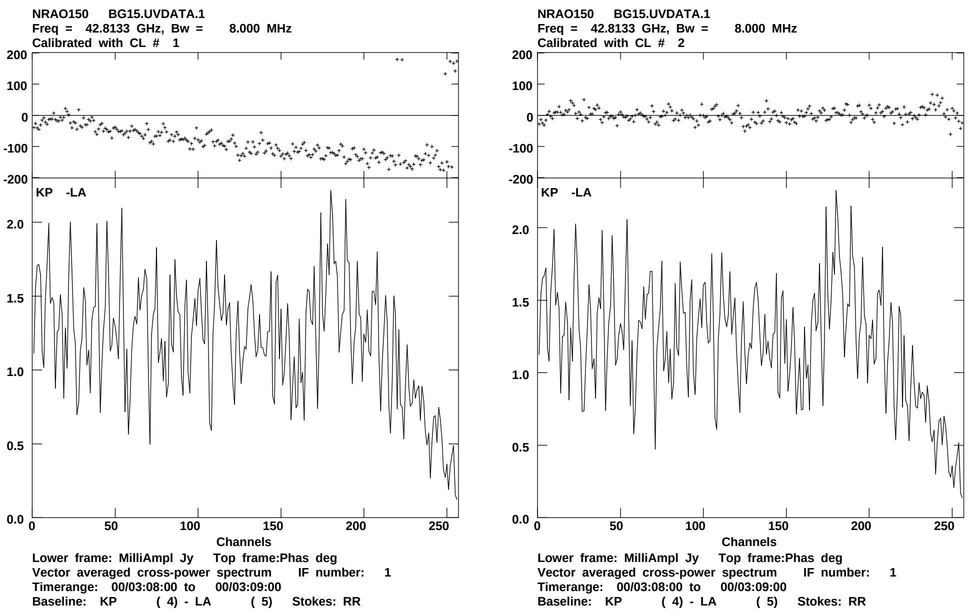

Figure 9.3: *left:* A POSSM plot of the uncalibrated spectrum of NRAO150 at 43 GHZ on the baseline Kitt Peak to Los Alamos. The plot shows the spectrum for IF 1 to show the effects of the residual delay error more clearly. The phase slope as a function of frequency is clear evidence for a small delay error in the correlator model. *right:* The same data as shown on the left, but corrected for a delay error of -55 nanosec and a residual fringe-rate of -2.0 milliHz. Note how the phase as a function of frequency is now flat and centered around zero degrees. These data can now be averaged in frequency, if desired.

9.4.8.8 Baseline-based fringe-fitting

Baseline-based fringe-fitting, implemented in BLING and BLAPP, is an alternative to using FRING and CLCAL described above. Whereas FRING searches and solves for station rates and delays globally, BLING makes independent fits to each baseline for delays and rates, creating a BS table of baseline-based solutions.

In most cases, the global fringe-fitting described in § 9.4.8.7 should be used since FRING should be able to fringe-fit weaker sources more reliably. However, there are some instances in which baseline-based fringe-fitting is to be preferred. Amongst the advantages of the baseline-based fringe-fitting are:

1. BLING may be more robust than FRING even in the absence of an accurate source structure model.
2. Fringe solutions can be found for cross-polarized fringes without editing the *uv* header.
3. As presently implemented, for a given number of IF and spectral channels, BLING can solve for longer scans than can FRING.
4. BLING has the option of adjustable, non-zero centered fringe-search windows, which can be controlled from an external file. This option will be especially important in fringe-fitting Space VLBI data.

You may wish to leave the acceleration search in BLING (DPARM(7) to DPARM(9)) off. This is a special option for space VLBI and will slow down BLING considerably.

This section concerns the use of BLING for baseline-based fringe-fitting. Associated tasks include BLAPP, which interpolates BLING solutions, and BSPRT, which prints solutions found by BLING.

BLING is distinguished from FRING by the ability to directly control the fringe search on each separate baseline, and by the ability to solve for fringe acceleration if required. Separate fringe windows in delay, rate and acceleration, and different solution intervals can be set for each individual baseline. The fringe windows may have non-zero offsets and can be specified using the input adverbs or, more flexibly, by drawing up an external ASCII control file of fringe-prediction windows and BLING control parameters. The latter option allows the specification of time-variable fringe-search parameters across the observing file. The general algorithm follows that described by Alef and Porcas, 1986, (*Astron. Astrophys.*, **168**, 365); BLING also allows the stacking of data from different baselines, as discussed by Schwab and Cotton, 1983, *Astron. J.*, **88**, 688. In addition, model division is possible before the fringe-fitting is performed and cross-polarized fringe searches can also be conducted without editing the *uv* header. BLING writes the results to a BS table. These baseline-based solutions can be factorized into antenna-based corrections using the separate task BLAPP, and applied to the data.

Acceleration is solved for by conducting a coarse search in the specified acceleration window. The results of this search are then interpolated to estimate the final solution. Fringe acceleration searches can considerably increase the amount of CPU time it takes to run BLING.

Full details concerning BLING input parameters can be found by typing EXPLAIN BLING. This includes information concerning the format required for the external control file. Earlier versions of BLING are discussed in AIPS Memo 89 (1994, "Baseline-Oriented Fringe Searches in AIPS" by Chris Flatters). Typical input parameters to BLING are given below:

> TASK 'BLING' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR 'DA193' C_R	source selection.
> STOKES 'LL' C_R	Stokes selection.
> TIMERANG 0, 10, 5, 0, 0, 11, 0, 0 C_R	time range selection.
> ANTENNAS 3 ; BASELINE 0 C_R	select all baselines to antenna 3.
> SUBARRAY 1 C_R	subarray selection.
> FREQID 1 ; BIF 1 ; EIF 0	frequency selection.
> BCHAN 1 ; ECHAN 0 C_R	
> DOCAL 2 ; GAINUSE *clin* C_R	apply amplitude calibration.
> CLR2N ; NMAPS 0 C_R	no model division.
> SOLINT 0.5 C_R	30-second fringe solution interval.
> INFILE ' ' C_R	no external control file specified.
> APARM(1) 2 ; APARM(2) 0 C_R	integration time; no model division.
> APARM(3) 0 C_R	no stacking.
> APARM(4) 0 ; APARM(5) 0 C_R	set minimum acceptable SNR to 5; accept coherence of 20%
> DPARM 0 C_R	default fringe windows; no acceleration search.
> DOUVCOMP 1 C_R	use compressed scratch files.
> BADDISK 0	scratch file selection.
> GO C_R	to run the program.

Note that baseline-stacking (APARM(3)) is not implemented for data sets with unequal integration times. Also, note that the fringe-rejection criteria are specified using APARM(4) and APARM(5) and these are important parameters. The use of compressed scratch files is recommended and is not believed to have a significant impact on precision. Note also that APARM(2) should be set and that the source should be entered explicitly if model division is required.

BLING will execute with a summary line marking the start of each baseline processed. The resulting BS table can be examined using task BSPRT, with input parameters:

> TASK 'BSPRT' ; INP C_R	to review the inputs.

> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INVERS *bsin* C_R	BS table version number.
> DOCRT -1 C_R	select output to an external file.
> OUTPRINT 'FITS:BSPRT.LIS' C_R	output file name.
> GO C_R	to run the program.

For printing to the screen, select DOCRT=1.

The BS table output includes the estimated fringe parameters and their associated errors. Note that due to recent changes in FFT interpolation the errors may be an overestimate at this time. The BLING solutions are interpolated and factorized into antenna-based gain solutions using BLAPP. This task either writes a solution (SN) table which can be applied using CLCAL, or allows a CL table to be updated with the calibration information directly. These options are selected using OPCODE='SOLV' or OPCODE='CAL' respectively. BLAPP can interpolate solutions with unequal time sampling and includes the acceleration term in interpolation if it is available. Typical inputs to BLAPP are:

> TASK 'BLAPP' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INVERS *bsin* C_R	input BS table version number.
> SOURCES ' ' ; STOKES 'LL' C_R	source and Stokes selection.
> FREQID 1 C_R	frequency id. selection
> TIMERANG 0 ; ANTENNAS 0 C_R	time range and antenna selection.
> SUBARRAY 1 ; REFANT 3 C_R	subarray and reference antenna selection.
> ANTWT 0 C_R	unit antenna weights.
> OPCODE 'SOLV' C_R	solve for a SN table.
> GAINVER clin ; GAINUSE clout C_R	specify CL tables to read in and write out.
> BADDISK 0 C_R	scratch file selection.
> GO C_R	to run the program.

The resulting solution (SN) table can be plotted using SNPLT in the standard fashion. For OPCODE='CAL' the input and output CL tables need to be specified using GAINVER and GAINUSE. If an SN table is generated it can be smoothed or clipped using task SNSMO and applied using CLCAL as described in § 9.4.1.1.

9.4.8.9 SVLBI-specific techniques

An alternative approach to direct fringe detection of each individual baseline to the orbiting antenna is to first calibrate the ground array using conventional fringe-fitting techniques, then coherently combine all ground antennas to improve the fringe detection sensitivity to the spacecraft. Several incarnations of this approach exist within *AIPS*. The *AIPS* tasks FRING, BLING, and KRING all allow baseline stacking which can be used to fringe fit the space baseline using composite baselines as discussed in a variety of VLBA scientific memos (#2,#3, #4, and #7). The adverb DOFIT in FRING and KRING can be used to solve for subsets of the available antennas in order to find good solutions for the ground antennas in a dual round of fringe-fitting (see the description of DOFIT in § 9.4.8.7). The exhaustive baseline search mode, activated in FRING by setting APARM(9)=1 and on by default in BLING and KRING, allows more baselines to the spacecraft to be searched.

The task GPHAS implements more formally a "phasing-up" of the ground-array into a single synthesized baseline to the spacecraft that can be fringe-fit'ted using standard techniques. The synthesized baseline can also be displayed and edited in *AIPS* using standard tasks. Because FRING, KRING, or BLING have proven capable of analyzing most existing data sets (as of 1 April 1999), GPHAS is not well tested. One method for reducing SVLBI data using GPHAS is presented here to give some idea of how GPHAS is to be used.

The usefulness of this method may depend on the accuracy of the source model as derived from the ground array imaging. This model correction is applied before the data are combined in GPHAS to form the synthesized baseline to the spacecraft. The ground array calibration and imaging can proceed by normal methods discussed elsewhere in the Cookbook. In this initial phase, the data on baselines to the spacecraft are excluded. We assume in what follows that this process has been completed successfully, with a resulting image file and *AIPS* solution (SN) table containing the ground array calibration. This SN table will lack any calibration information for the spacecraft. Users are reminded that CLCAL can merge solution tables using OPCODE='MERG'.

An important point to note is that the general *AIPS* calibration philosophy requires that each antenna, polarization and IF channel in any given solution table have complete calibration information. If calibration information is missing for one or more antennas, these antennas will be flagged by gain in the output CL table produced by CLCAL when the solution table is applied. A special mode in CLCAL has been introduced to allow incomplete calibration information to be applied. In this mode, the CL entries that have no associated SN table information are passed through to the output CL table unmodified. This mode is activated in CLCAL using OPCODE='CALP'. This mode is identical to OPCODE='CALI',INTERPOL='SELF' in all other respects, and the other input adverbs have their customary meaning in this case. This CLCAL mode should be used in applying the ground-array SN table containing the calibration information. The resulting CL table will correctly calibrate the ground-array antennas but leave the spacecraft uncalibrated. At this point it is possible to combine the ground baselines using GPHAS:

> TASK 'GPHAS' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> IN2DISK *n* ; GET2N *ctn* C_R	to specify the input image.
> CMODEL ' ' ; CMETHOD ' ' C_R	model division defaults.
> SMODEL 0 ; BCOMP 0 ; NCOMP 0 C_R	
> INVERS 1 C_R	input CC table version number.
> OUTDISK *n* C_R	specify the output file.
> OUTNAM 'VSOP', OUTCLA 'UV'	
> CALSOUR 'DA193' C_R	must be specified for model division.
> STOKES ' ' C_R	Stokes selection
> BCHAN 1 ; ECHAN 0 C_R	default frequency selection.
> BIF 1 ; EIF 0 C_R	
> DOCAL 2 ; GAINUSE *clin* C_R	apply ground-array calibration.
> ANTENNAS 0 ; REFANT 0 C_R	select all antennas.
> SOLINT 0 C_R	no averaging.
> APARM 11 1 C_R	to combine all baselines to antenna 11 and to reference output baseline to most commonly occuring reference antenna.
> BADDISK 0 C_R	scratch file selection.
> GO C_R	to run the program.

Other standard selection and calibration adverbs are also available. Type EXPLAIN GPHAS for further details. The resulting output file can be examined using standard tasks such as VPLOT and fringe-fit using FRING or BLING. The incremental calibration for the spacecraft can then be applied using similar methods to those used to calibrate the ground array only, as discussed above.

9.4.8.10 Spectral-line fringe-fitting

The determination of the delay and fringe-rate calibration is a two or three-step process for spectral-line VLBI data. First, the residual delay and fringe-rates are estimated for each antenna from the continuum calibrators. Then, residual fringe-rates must be determined again for the line source using a "strong" channel or range of channels. As an intermediate step, the phases of the line source should be examined to check that the calibrator's residual fringe-rates haven't destroyed phase coherence; if so, then the calibrator's residual fringe-rates should not be applied to the line source.

Because of limited computer memory, FRING must trade off the number of spectral channels against the length of the solution interval. For continuum calibrators in a spectral-line dataset, the large number of spectral channels often forces FRING to require too short a solution interval. This can be overcome be running UVCOP to extract the continuum calibrators into a separate data file and then running AVSPC with AVOPTION 'SUBS' to average spectral channels coherently within each IF. INDXR should be run to regenerate an NX table. FRING will then allow more reasonable solution intervals.

Run FRING as follows only on the continuum calibrators to determine residual delays and fringe-rates:

> TASK 'FRING' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR 'BLLAC','DA193' C_R	to select continuum calibrator sources.
> DOCALIB 2 C_R	to apply the current calibration.
> GAINUSE *clin*	to specify which CL table to use.
> SMODEL 0 C_R	to use the null source model (points at the origin).
> FLAGVER 0 C_R	to apply the most recent flag table.
> BCHAN 10 ; ECHAN 115 C_R	to exclude the edges of the band; normally the data in these channels are corrupted by the bandpass filters.
> REFANT 1 C_R	to select a reference antenna (see continuum discussion).
> SOLINT 6 C_R	to set the solution interval in minutes. It should not exceed the coherence time.
> APARM(6) 1 C_R	to get some useful, but limited, printout; gives SNR.
> APARM(7) 7 C_R	to avoid false detections by setting the minimum SNR acceptable. *Warning:* data with lower SNR will be flagged as bad.
> DPARM(1) 1 C_R	to use one-baseline combination in initial, coarse fringe search (FFT). This provides starting points for the least-squares solutions.
> DPARM(2) 10000 C_R	to select a delay window in nsec, centered around 0(!). The default is to use the Nyquist range. For a 250kHz-bandwidth observation, setting this value to 10000 nsec is equivalent to setting the search window to 5 delay channels, which is usually sufficient.
> DPARM(3) 200 C_R	to select a fringe-rate window in mHz.
> DPARM(4)= 1 C_R	to tell FRING the correlator integration time.
> DPARM(5) 0 C_R	to do the least-squares solution.
> SNVER 0 C_R	to write solutions in a new SN table.
> GO C_R	to do the fit.

If the calibrator had been extracted to a separate data set, use TACOP to copy the resultant SN table back to the line data set.

Run CLCAL to apply the delay and fringe-rate solutions to all sources, as is described in § 9.4.1.1, and then carefully examine the phase coherence of the line source in a suitable line channel (or group of channels) using POSSM or COHER before and after applying the new CL table. It may be that the fringe-rate solutions have worsened the phase coherence. In this case, you must run SNCOR using the 'ZRAT' option to zero the fringe-rates and then re-run CLCAL.

After this point, the calibrator data is usually of little or no interest. But, if you do plan to use the calibrator data further, remember to be careful to juggle the SN and CL tables correctly. Even if you decide not to apply the calibrator fringe-rates to the line-source, they are still applicable for the calibrator itself. The CL table created using the un'ZRAT'-ed SN table contains the proper corrections for the calibrator data while the CL table created using the 'ZRAT'-ed SN table contains the proper corrections for the the line source.

Now re-run FRING to determine the residual fringe-rates for the line source, this time selecting a suitable line channel or group of channels:

> TASK 'FRING' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> CALSOUR 'OH127.8' , ' ' CR	to select the spectral-line source.
> DOCALIB 2 CR	to apply the previous amplitude and delay/rate calibration.
> GAINUSE *clin*	to specify which CL table to use.
> FLAGVER 0 CR	to apply the most recent flag table.
> SNVER 0 CR	to write a new solution table.
> BCHAN 72 ; ECHAN 72 CR	to select the strongest and/or simplest spectral channel as a reference.
> REFANT 1 CR	try to use the same reference antenna as in the previous run of FRING.
> SOLINT 6 CR	to set the solution interval in minutes. It should not exceed the coherence time.
> APARM(6) 1 CR	to get useful, but limited printout.
> APARM(7) 9 CR	to avoid false detections by setting the minimum SNR acceptable. *Warning:* data with lower SNR will be flagged as bad.
> DPARM(1) 1 CR	to use one-baseline combination in initial, coarse fringe search (FFT).
> DPARM(2) = -1 CR	to prohibit a search in the delay domain by setting the delay window to a negative number. Remember setting this to 0 means to use the Nyquist value, which is not what we want.
> DPARM(3) 0 CR	to search for fringes over the full Nyquist fringe-rate window since we don't know where the fringes are.
> DPARM(4)= 0 CR	to let FRING automatically determine the correlator integration time (if this fails, you will have to set the proper value here — see DTSUM).
> DPARM(5) 0 CR	to do the least-squares solution.
> GO CR	to do the fit.

Then run CLCAL again to apply these solutions to the previous calibration tables. You have then generated a full set of calibration tables and data. Remember that the calibration information for the continuum and line sources are stored in different CL tables.

9.4.8.11 Polarization-specific fringe-fitting

The phase and delay corrections obtained using RR and LL data can only remove R-R and L-L offsets between different antennas. There still may be R-L phase and multi- and single-band delay offsets. The R-L delay corrections can be determined using the RL and LR data and two runfiles, CROSSPOL and CRSFRING, are available for determining these effects. This still leaves an overall R-L phase offset.

Once the cross-hand calibration has been completed, the instrumental polarization (otherwise known as the feed D-terms), can be determined. Several different methods are available for this purpose, implemented in the tasks PCAL, LPCAL and SPCAL. The latter task is designed for polarization calibration of spectral-line datasets. PCAL is discussed briefly below and further details for each of these tasks can be found in the appropriate EXPLAIN files.

The determination of the absolute polarization position angle is equivalent to the determination of the absolute R-L phase difference. For an unresolved source this can be measured in much the same way as for VLA data (see § 4.6) using task LISTR to determine the cross-polarized phase on the polarization calibrator. Alternatively, a source with known polarization properties can be used, *e.g.*, 3C286 or 3C279. For a resolved polarization calibrator a sum of Clean components is required to compare to the integrated polarization position angle as measured by the VLA or single-dish observations nearby in time. Once the R-L phase correction is known it is applied using task CLCOR with OPCODE='POLR' (see § 4.6).

After feed calibration and the determination of the absolute polarization position angle, the final Stokes Q and U images can be formed directly. Note that task PCNTR, which is used for displaying polarization images, allows an arbitrary rotation of all polarization vectors under control of input adverb ROTATE (this is only necessary if CLCOR wasn't used to correct the R-L phase; also the polarization angle specified for PCNTR is half the R-L phase difference).

9.4.8.12 R-L delay calibration

CROSSPOL (see EXPLAIN CROSSPOL) can determine multi- and single-band R-L delay differences and produce an SN table which corrects for these effects. It is best run on a set of calibrator data for which the source is at least moderately polarized (source polarization dominates instrumental polarization). Several baselines can be averaged but RL or LR fringes (or both) must be detectable on each baseline to the reference antenna. OPCODE='AVIF' will cause the single band R-L delays to be averaged. This procedure should leave a single R-L phase difference that must be determined from a calibrator of known polarization angle. The SN table produced by CROSSPOL should be applied to the CL table using CLCAL. The results can be checked using POSSM or SHOUV. The RL and LR phases should be continuous across the bandpass on each baseline and be flat if the RR and LL phases are flat (no residual delays).

9.4.8.13 Feed D-term calibration

The feed D-terms, or instrumental polarization terms, can be determined using PCAL. SOLTYPEs 'ORI-' and 'RAPR' are appropriate for VLBI data. 'ORI-' uses a non-linear orientation-ellipticity feed model and is appropriate if instrumental polarization exceeds a few percent (*e.g.*, EVN, or VLBA at 18 or 13 cm). 'RAPR' uses a linearized "D-term" model — this is faster but less accurate.

Typical inputs for PCAL would be:

> TASK 'PCAL' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR 'DA193' , ' ' C_R	to select the source.
> TIMERANG 0 C_R	to use all times.
> SELBAND 0; SELFREQ 0; FREQID -1 C_R	to select all frequencies.
> BIF 0; EIF 0 C_R	to select all IFs.
> ANTENNAS 0; UVRANGE 0 C_R	to select all antennas and baselines.
> SUBARRAY 0 C_R	to select all subarrays.
> FLAGVER 0 C_R	to apply the most recent flag table.

> DOCALIB 2 C_R	to apply the previous amplitude and delay/rate calibration.
> GAINUSE *clin*	to specify which CL table to use.
> IN2DISK *n*; GET2N *ctn* C_R	to specify a clean map.
> INVERS 1 C_R	to select a CC table version.
> REFANT 1 C_R	to use the same reference antenna as in the previous run of CALIB.
> NCOMP 77; NMAPS 1 C_R	to specify the number of clean components to use and to explicitly set the number of clean maps supplied.
> SOLINT 6 C_R	to set the solution interval in minutes. It should not exceed the coherence time.
> SOLTYPE 'RAPR' C_R	specify the type of feed model.
> PRTLEV 0 C_R	to print minimal information
> BPARM(1) 0 C_R	use the initial feed model if found in the AN table.
> BPARM(3) 0 C_R	to not fit for R-L phase difference.
> BPARM(4) 0 C_R	to not specify the initial R-L phase.
> BPARM(5) 0 C_R	to not solve for Vpol.
> BPARM(6) 0; BPARM(7) 0 C_R	to solve for the orientations of both polarizations of the reference antenna.
> BPARM(8) 0 C_R	to solve for all orientations.
> BPARM(9) 0 C_R	to solve for all ellipticities.
> BPARM(10) 0 C_R	to fit for the source polarization model parameters.
> CPARM(1) 0 C_R	to find separate solutions for each IF.
> CPARM(8) 0 C_R	to not limit the number of iterations.
> CPARM(9) 0 C_R	to use the default convergence tolerance.
> CPARM(10) 0 C_R	use default convergence criterion.
> BADDISK 0 C_R	to specify which disks to avoid for scratch.
> GO C_R	to run the program.

The instrumental polarization may vary rapidly with frequency and independent solutions may be necessary in each IF. Both 'ORI-' and 'RAPR' model the source Q and U as scaled versions of I which is generally only true in the limit of unresolved or unpolarized sources. The D-terms can be determined iteratively by subtracting estimates of source polarization (Q,U clean components) in UVSUB. Note: this should be done on data for which instrumental polarization corrections have *not* been applied.

Tasks LPCAL and SPCAL also can be used to compute D-term corrections for continuum and spectral-line polarization data respectively. Be forewarned that both of these tasks use linearized D-term models.

9.4.9 Baseline-based errors

Baseline-based non-closing phase and amplitude errors can limit the dynamic range of the final maps. One way to proceed is to try to solve directly for the non-closing effects using bright, point-like calibrator observations and the task BLCAL. This task writes a BL table containing the estimated non-closing baseline-based errors which can later be applied in SPLIT, or any of the other calibration tasks. To use BLCAL the noise in the calibrator-source images should approach the theoretical limit. Furthermore, the signal-to-noise ratio in the visibility data must be at least 100:1 on baseline-averaged data. BLCAL will divide your data by your best model and then write a BL table containing the baseline-based corrections. Use this task carefully only after reading the EXPLAIN file thoroughly. As an example:

> TASK 'BLCAL' ; INP C_R	to review the inputs.

> INDISK *n1* ; GETN *ctn1* C_R	to select the multi-source visibility data as the input file.
> IN2DISK *n2* ; GET2N *ctn2* C_R	to select your best image as the input model file.
> SOURCE 'DA193' C_R	to select your calibration source.
> DOCALIB 2 ; GAINUSE *clin* C_R	to select the CL table to use.
> BLVER 1 C_R	to create BL table version 1.
> SOLINT 1440 C_R	to determine one complex gain correction per baseline for the whole observation.
> GO C_R	to run the program.

9.5 After initial calibration

9.5.1 Applying calibration

Having obtained your best possible calibration CL table (and BP and BL tables if bandpass or baseline-dependent errors were found), you finally get to make a calibrated data set. This is done with SPLIT, which applies the calibration and splits the database into separate files, one for each source observed.

> TASK 'SPLIT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCE ' ' C_R	to write all sources.
> DOCALIB 2 ; GAINUSE clin C_R	to specify which CL table to use.
> DOBAND 1; BPVER 1 C_R	to apply BP table 1 if present.
> BLVER *blin* C_R	to apply BL table 1 if present.
> APARM(1) 1 C_R	to have all spectral channels within each IF averaged.
> APARM(2) 2 C_R	to have an amplitude correction made for the correlator integration time (in seconds)
> GO C_R	to run the program.

The options for SPLIT given above will apply calibration and then average the spectral channels within each IF, but not average IF channels together . To average over IF channels as well, set APARM(1)=2 in SPLIT or use task AVSPC:

> TASK 'AVSPC' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OUTDISK *n* ; OUTNAM ' ', OUTCLA ' '	to specify the output file.
> AVOPTION 'AVIF' C_R	to average over IFs.
> BIF 0 ; EIF 0 C_R	to select all IF channels
> GO C_R	to run the program.

At this point, it is well worth spending time to examine your output visibility data carefully. You may plot the data against time with VPLOT, IBLED, EDITR, or UVPLT, and list them with LISTR, PRTUV, or UVPRT. POSSM is now no longer useful since you have averaged your data in frequency space.

9.5.2 Time averaging

It is now convenient to time average the data using UVAVG both to reduce the bulk of the data and to increase the signal-to-noise for subsequent iterations of the self-calibration/mapping cycle. However, it is important to realize that the fringe-fitting process to this point has only removed gradients of phase over the fringe-fitting solution interval. There will still be stochastic atmospheric (and clock) phase errors affecting the data on short time scales. These phase errors can be significant over minutes at frequencies of 22 GHz and above (and possibly even at 15 GHz) and a reduction in amplitude can occur if data are directly averaged. Self-calibration should remove such phase errors.

For data at frequencies below 15 GHz (and ≈ minute integrations), it should be safe to proceed with `UVAVG` (see below). For higher frequency data, it may be worth your while to examine the phase coherence of the data first. `VPLOT` can be used to examine your target or calibrator data to see directly the level of residual phase error over your chosen averaging time. Alternatively, the task `IBLED` allows you to view the degree of coherence of data averaged over different averaging times. If there are coherence problems (and the target data has enough SNR), `CALIB` can be run to align the phases prior to coherent averaging. Try:

> TASK 'CALIB' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the single-source input file.
> OUTNA INNA ; OUTCL 'ALIGN' C_R	to specify the output file.
> CALSOUR ' ' ; SMODEL 1, 0 C_R	to use the source with a point-source model.
> DOCALIB -1 ; GAINUSE 0 C_R	to not apply any tables to the input data.
> SOLTYPE ' ' C_R	to use normal least squares.
> SOLMODE 'P' C_R	to solve for phase.
> SOLINT (10.0/60.0) C_R	to solve for phase in 10-second intervals. This should probably be set as low as the strength of the source will allow. The limit is the integration time that gives a SNR > 2 on most baselines.
> APARM(1) 3 C_R	to require 3 antennas present for solution.
> APARM(6) 0 C_R	to skip diagnostic printout.
> APARM(7) 1 C_R	to set the minimum allowed SNR. This limit should be low since the SNR is calculated as a phase difference from model and this can be large. Start with a value ≤ 1.
> ANTWT 1 C_R	to use weights from calibration with no additional weights applied to the antennas. For the purposes of phase alignment, it is appropriate to use the data weights; this allows the noise in the solution to be distributed over the noisiest baselines. This may not be the case when using `CALIB` for self-calibration in the hybrid mapping sense (see § 9.6).
> GO C_R	to run the program.

Note that `CALIB` will only give valid solutions if the signal-to-noise over the solution interval on most baselines is greater than 2 (and preferably much higher). At high frequencies on weak sources, it may not be possible to select a solution interval long enough that the signal-to-noise satisfies this criterion, yet short enough to follow the atmospheric phase variations. In such cases, it is probably best not to attempt to self-calibrate the data, but instead to use a short averaging time and to live with any coherence losses in the data.

When the data are sufficiently phase coherent, they should be averaged over time down to a reasonable size using `UVAVG`:

> TASK 'UVAVG' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the `CALIB` output file as the `UVAVG` input file.
> OUTNA INNA ; OUTCL 'UVAVG' C_R	to specify the output file.
> YINC 30.0 C_R	to set the time-averaging interval to 30 seconds.
> OPCODE ' ' C_R	to enable the averaging operation.
> GO C_R	to run `UVAVG`.

9.5.3 Verifying calibration

Before proceeding to image your data, it's worth checking that the calibration performed in § 9.4 is sensible. For each of your sources, produce a plot of the correlated flux density against *uv* distance using UVPLT. As well as identifying bad data which can then be deleted with IBLED, EDITR, or UVFLG, these amplitude versus distances plots (especially those of your calibrator sources) can be used to identify stations where the amplitudes are too high or too low. Furthermore, by fitting simple models to the calibrator data, constant correction factors can be determined for each station which can be used to correct the amplitude calibration. It is often the case that the amplitude calibration to a certain station (particularly non-VLBA stations) is out by a constant factor, either due to uncertainties in the antenna gain or in the noise calibration signal.

Most VLBI calibrator sources can be adequately described by one or two Gaussian components. The task UVFIT can be used to fit such a model while finding constant correction factors for the antenna gains. Note that UVFIT can handle no more than 25,000 visibilities, so further averaging with UVAVG may be required. The following example shows how to fit antenna gains and a single elliptical Gaussian model of known position and flux to one of the single-source data sets produced by SPLIT and AVSPC:

> TASK 'UVFIT' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input *uv* data set.
> OPCODE 'GAUS' ; NGAUS 1 C_R	to specify 1 Gaussian component.
> GMAX 1.2 ; DOMAX FALSE C_R	to fix the flux at 1.2 Jy.
> GPOS 0 ; DOPOS FALSE C_R	to hold the position fixed at the origin.
> GWIDTH 0.002 , 0.001 , 45. C_R	to provide an initial guess of the Gaussian widths as 2×1 mas at a position angle of 45 deg.
> DOWIDTH TRUE C_R	to fit for size.
> GAINERR 1 C_R	to fit for all antenna gains with initial guess = 1.
> NITER 50 C_R	to limit the fitting to 50 iterations.
> IMSIZE 0.0005 , 0.01 C_R	to limit sizes to be in the range 0.5 to 10 mas.
> DOCAT TRUE ; INVER 1 C_R	to save the solution in a CC file of version number 1.
> INP C_R	to check the inputs.
> GO C_R	to run UVFIT.

UVFIT can also be applied to the multi-source file, but SOURCE, DOCAL, GAINUSE etc. must then be set.

Another way to test your amplitude calibration is to use the task UVCRS for bright sources with long tracks. This task calculates correction factors for the amplitudes of the stations using regions of the *uv* plane where *uv* tracks cross. UVCRS can write the correction factors into an SN table.

Once the scale factors are determined, there are a number of options for correcting the data. The simplest option is to apply the correction factors to the single-source data sets using task VBCAL. Alternatively, the correction factors can be incorporated into the highest version CL table of the multi-source data file and task SPLIT run again to make new calibrated data files. The corrections to the CL table is done with CLCOR. Unlike VBCAL, CLCOR must be run separately for each antenna whose calibration you wish to alter; ANTENNA must be set to the antenna number you wish to change, OPCODE = 'GAIN' C_R and CLCORPRM(1) set to the amplitude scale factor found in UVFIT squared; all higher values in the array CLCORPRM should be zero. The effect of the altered calibration can be viewed using UVPLT with DOCAL = 2. If it is satisfactory, SPLIT can be re-applied to the data. We recommend using CLCOR to perform such amplitude corrections.

Another way of incorporating amplitude corrections is to edit the calibration text files used by ANTAB. This can be accomplished by setting the FT parameters for affected stations. For instance, if the Bonn scale factor is 1.043, set the FT parameter on the BONN TSYS card to FT=(1.043*1.043). If amplitude calibration was carried out *after* fringe-fitting, then it is only necessary to rerun ANTAB, delete the latest CL table containing amplitude calibration and rerun APCAL using the highest CL table produced in the fringe-fitting step. If however, as we have described in this chapter, the amplitude calibration was done prior to fringe-fitting, then correcting the amplitudes is more involved. It is probably best to delete all CL tables except the first one and start again at § 9.4. However, it may not be necessary to carry out the time consuming FRING solutions again. If the amplitude changes are small, the phase, rate and delay solutions will be essentially unchanged. Therefore, with care, the existing SN tables can be used in lieu of re-running FRING.

9.6 Self-calibration, imaging, and model-fitting

We are now in a position to make images. As with VLA data, we do this by iteratively, self-calibrating the data and deconvolving using Clean, MEM *et al.*. We describe below a typical self-calibration and imaging sequence for VLBI data. The tasks used are described in more detail in Chapter 5.

1. CALIB self-calibrates the *uv* data.
2. IMAGR images and cleans. IMAGR is now the preferred task for imaging VLBI data in *AIPS*.
3. SCMAP images, cleans, and self-calibrates. SCMAP is meant to provide the functionality of the popular VLBI data analysis package difmap. SCIMG is a multi-field version of SCMAP..

The main difference between the processing of VLBI and VLA data is that, initially, absolute phases of VLBI data are un-calibrated (unless phase-referencing is used). Therefore, many more iterations around the imaging loop are required for the VLBI case; dozens of self-calibration iterations are not uncommon. Given this, it may be convenient to use the procedure HYB which executes a whole cycle of hybrid mapping (*i.e.*, an MX plus CALIB). It also plots images and allows editing of Clean component files prior to self-calibration. Type HELP HYB C_R for more information. Note that it does not use the superior imaging algorithms available in IMAGR, SCMAP, and SCIMG.

Note that VLBI imaging is not an exact science and there are a number of different views on the "correct" imaging method and the "correct" software to accomplish this method. Some users take their *AIPS* data into a package written at CalTech to use the difmap program (see ftp://phobos.caltech.edu/pub/difmap/). Others use the *AIPS* tasks SCMAP and SCIMG (see § 5.4), while still others follow older HYB path. It is beyond the scope of this document to explain in detail all aspects of VLBI imaging. For more details, see Craig Walker's chapter on "Practical VLBI Imaging" in the publication *VLBI and the VLBA* which is available on the World-Wide Web. Here we make a few suggestions on how to control CALIB and IMAGR. Again, please note that the latter is preferable to all previous imaging tasks; SCIMG and to a lesser extent SCMAP offer the same improved imaging techniques.

9.6.1 CALIB

1. Start by correcting antenna phases only, *i.e.*, use SOLMODE = 'P' C_R. Switch on the amplitude correction only after you have converged to a fairly good image. On the first iteration, you will need to invent an input model. For most extragalactic continuum sources, a point-source model is a good choice. Set SMODEL(1) to the zero-spacing flux density as interpolated by eye using UVPLT and consider using a circular Gaussian model at the origin to reduce the impact of the longest spacings. Start with the so-called SNR parameter APARM(7) small (≤ 1) and gradually increase it as the image improves. If this parameter is large during early iterations, when the model used is far from correct, then large portions of your data may be flagged.
2. On subsequent iterations, use the Clean image as produced by IMAGR as your input model. Unlike VLA images, VLBI applications usually require Clean components well beyond the first negative component to be used in calculating the source model. One possibility is to use PRTCC to find the point where

a significant fraction (*e.g.*, one third) of all new Clean components are negative. An alternative is to use all of the Clean components, but to use tight windowing in IMAGR — which can now be done interactively on the TV as IMAGR progresses. Alternatively, use tight windowing and clipping of the Clean components with CCEDT or CCSEL before running CALIB. Tight windowing is especially important when *uv* coverage is poor. Editing Clean components after IMAGR, but before CALIB, can be effective in removing possibly spurious features; if they are real they will usually reappear in later iterations. IMAGR now offers a filtering option to remove weak, isolated components when requested from the TV and at the end before the components are restored to the image. This removes much of the need for CCSEL.

3. When carrying out the next CALIB iteration with the new Clean model, you can either self-calibrate the original data set or, alternatively, self-calibrate the data set which was used to produce that Clean model. It is probably advantageous to go back to the original data set periodically. Amongst other things, this can prevent the telescope amplitudes from "wandering" (see below).

4. If your array contains antennas that have a wide range of sensitivities, *e.g.*, the VLBA plus the phased VLA and/or the Bonn 100-m, it is helpful to alter the weights of the antennas in your CALIB solutions. If this is not done, then your solution will be dominated by only a few baselines and the uniqueness of the solution is not guaranteed. Use PRTUV to inspect the weights of your data. Then set the CALIB input array ANTWT, which provides multiplicative factors adjusting the weights for each antenna prior to the CALIB solution. Set these parameters so that the effective range of baselines weight is only 10 to 100. Alternatively, use WTMOD to raise the original weights to a power between 0.25 and 0.5.

5. As you iterate, keep an eye on how the model image is converging to fit the data. Use VPLOT, CLPLT and UVPLT.

6. When your source has a lot of extended structure and/or your VLBI array has relatively few short spacings, you should consider setting UVRANGE to only include the range of spacings in which the model provides a good fit to the data. However, given the relatively small number of antennas in most VLBI observations, you may need to compromise to allow in enough baselines to get good self-cal solutions.

7. When you are finally ready to solve for amplitude corrections, you should first apply all previous phase calibration (including final phase-only self-calibration solution). Then run CALIB setting SOLMOD 'A+P', initially setting the solution interval (SOLINT) to several hours. Try to prevent the antenna amplitudes from "wandering," which can sometimes happen if there is still a significant amount of short spacing flux density missing from the source model. Setting UVRANGE is useful, as is setting CPARM(2) =1 ↵ to constrain the mean amplitude solutions over all antennas to be one. You can also set SOLMODE='GCON' ↵ and the array GAINERR to the expected standard deviation of the gains for each antenna. This constrains amplitude solutions to conform to the expected statistics. Setting the gain constraint factor SOLCON to values larger than 1 will increase the importance of these gain error constraints. Finally, going back and self-calibrating starting with the original data set and the best available Clean model is useful way to prevent amplitude wander.

CALIB has been enhanced however to improve its usefulness for SVLBI datasets. This has involved the implementation of improved antenna selection and partial array calibration, through the new adverb DOFIT. The possibility of solving only for a subset of selected antennas has been implemented in this manner and may prove useful for SVLBI data. The implementation of adverb DOFIT in CALIB is analogous to its implementation in FRING (see § 9.4.8.7).

9.6.2 IMAGR, SCIMAG, and SCMAP

1. Before using IMAGR or SCMAP or SCIMG, print out and read the EXPLAIN file. They are powerful and complicated tasks with many adverbs — some of which are new — and shouldn't be used blindly.

2. The quality of images produced may depend on the type of weighting used. With VLBA-only experiments, the best quality images are often produced using natural (UVWTFN 'NA' C_R) weighting in IMAGR. These images will represent the extended structure of the source better. If the highest resolution is required, try uniform (UVWTFN 'UN' C_R) weighting. The ROBUST parameter allows weightings intermediate between these two extremes often with both good signal-to-noise characteristics and a narrow synthesized beam. It may also be worth experimenting with the UVBOX parameter to allow smoothing of weights over larger areas of the *uv* plane (*i.e.*, to use "super-uniform weighting"). If the array contains antennas with very different sensitivities, (for instance, if it includes Bonn or the phased VLA), then it may be advantageous to lower the weights of baselines to these antennas. Although this increases the thermal noise in the image, it will improve the *uv* coverage, which, otherwise, will contain effectively only the baselines to these sensitive antennas. One way of doing this is to use UVWTFN = 'UV' C_R in IMAGR. This option takes the fourth root of the input weights before applying uniform weighting. SCMAP also supports these weighting options. Another flexible (but deprecated) approach is to use task WTMOD to change the weights in the data set prior to running IMAGR.
3. After an initial self-calibration against a point-source starting model, the deconvolved image will often show spurious symmetric structure. Convergence can be speeded up by placing Clean boxes CLBOX) around the side showing the brighter structure. Note that CLBOX may be used to produced circular as well as rectangular windows for use with IMAGR. Alternatively, CCEDT can be used to edit the Clean components after IMAGR, but before the next CALIB. IMAGR, SCIMG, and SCMAP allow this to be done interactively at the start of each major Clean cycle including the first.
4. Use DOTV = 1 C_R to view the residuals and possibly modify the Clean boxes as you Clean. You can stop Cleaning if you feel that you are including spurious structure into your model or if you feel you need to reset a Clean box to include a new feature.
5. SCMAP and SCIMG, at each self-calibration cycle, offer a powerful interactive data editing tool which displays input and residual data from up to 11 baselines simultaneously. This is the same editor as found in task EDITR.

The hints outlined above are by no means the whole story when it comes to self-calibrating and imaging VLBI data. Unfortunately, it can still be somewhat of an art form. Very experienced users can produce noise-limited images, but there is no simple recipe that will enable inexperienced users to do the same.

The existing weighting schemes available in IMAGR are believed to be sufficiently flexible for SVLBI imaging requirements. Further information on this point will be obtained during the VSOP in-orbit checkout period.

9.6.3 Model-fitting

For those SVLBI datasets for which conventional aperture synthesis imaging is not possible there are several other options.

Fringe-rate mapping is available, in the form of the task FRMAP. Type EXPLAIN FRMAP within *AIPS* for further information.

An integrated model-fitting environment is provided in the form of the task SLIME (Slick Interactive Model Editor). This is an external application, written in the C language, but linked against the standard *AIPS* calibration and data selection libraries, and is distributed separately from *AIPS*. It has an X-window based graphical user interface and allows data display and interactive graphical editing of a direct representation of the fitted model. Web-based information on SLIME and the Users Guide can be found via

http://info.aoc.nrao.edu/~cflatter/slime.html .

This task can be run directly from *AIPS*; type `EXPLAIN SLIME` for further information. Note however that it is important to average the data in time and frequency before using the model-fitter (see § 9.5.2).

9.7 Data simulation within *AIPS*

In developing SVLBI data reduction support within *AIPS* the capability was required to generate fake datasets with pre-fringe-fitting errors of various forms. A task was written for this purpose, called `DTSIM`. This task, when used with an auxiliary task `DTCHK`, has also been used to test the correctness of *AIPS* tasks in general. The simulator should be understood as a software testing tool and does not have an error model designed for highly sophisticated data simulations. Support is implemented however for orbiting VLBI antennas.

The format of the output *uv* data file, calibration errors, source model and the fake schedule are specified in an external text file in `KEYIN` format. The required format is described in the `DTSIM EXPLAIN` file. The general model used in generating the fake data is of the following general form:

$$V_{ij}^{pq} = G_i^p G_j^{q*} V_{mod}^{pq} + N_{ij}^{pq}$$

where V_{ij}^{pq} is the simulated visibility data on baseline $i - j$ for polarization pq ($p, q \in$ RCP, LCP), and V_{mod} and N are the model visibility and thermal noise contribution respectively. The instrumental antenna-based complex gain terms are indicated as $G_i^{p,q}$, and are parametrized by residual delay, phase and bandpass errors. In addition, there is support for instrumental polarization errors but this is not represented in the expression above.

Task `DTSIM` is run as follows:

> TASK 'DTSIM' ; INP C_R	to review the inputs.
> OUTDISK n C_R	specify the output file.
> OUTNAM 'VSOP', OUTCLA 'UV'	
> INFILE 'FITS:DTSIM.INP' C_R	`KEYIN` input file.
> PRTLEV 0 C_R	do not echo input text file.
> GO C_R	to run the program.

The resulting fake dataset has `FQ, AN, SU`, and `HI` tables attached. The `CL` table version 1 and the index table need to be generated using `INDXR`. The *uv* data file is then in standard *AIPS* format.

This task can be used to generate fake SVLBI datasets, and has been used to test the fringe-fitting and other SVLBI software. The results of these tests are described elsewhere, but an illustrative plot of the performance of `FRING` for various signal-to-noise ratio data is shown in Figure 9.4.

9.8 Summary of VLBI calibration tables

Several different tables are supplied with a VLBI data set created by the Socorro and other correlator; various other tables (*e.g.*, `SN` tables) are created by the calibration process. A list of these tables is given below for your edification. Not all tables will be present with all files.

AN	Antenna table. Contains a list of the antenna names and station coordinates. Also contains instrumental polarization terms.
AT	Antenna Characteristics table. Contains additional information about antenna properties, including some time variable quantities.

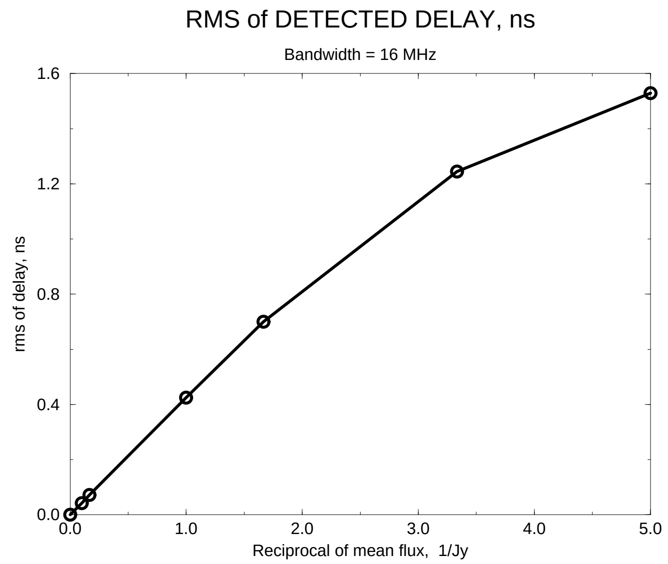

Figure 9.4: RMS of measured residual delay as determined by `FRING` for simulated 18cm VSOP data with a point double source model of 1.5 mas E-W separation and a flux density ratio of 2:1.

`BL`	Baseline offset table. Contains non-closing baseline-dependent phase and amplitude errors as determined by `BLCAL`.
`BS`	Baseline Solution table. Contains baseline delay, rate and phase solutions as determined by `BLING`.
`CL`	Default calibration table. Contains, amongst other things, the default calibration parameters for the amplitude (usually unity), phase, and single-band delay (usually zero) for each source for each IF as a function of time. It also contains polynomial coefficients allowing the correlator delay and phase models to be recomputed. As calibration proceeds, higher versions of this table are created which incorporate more and more calibration effects into the phase and delay entries.
`CQ`	Correlator parameter frequency table. Contains VLBA correlation parameters for each AIPS IF, and activates VLBA delay decorrelation corrections. Type `EXPLAIN FXVLB` for further information.
`CT`	`CALC` table. Contains the input parameters passed to `CALC` to generate the polynomials recorded in the `IM` table.
`FG`	Flag table. Contains information used to delete selected portions of the data.
`FQ`	Frequency table. Contains information about the IF frequencies, channel spacings, bandwidths, etc.
`GC`	Gain Calibration table. Contains the expected zenith gain and gain-elevation curve for each antenna. It is used for amplitude calibration.
`HF`	Haystack `FRNGE` table. Contains information generated from the AIPS tables that can be exported to the `CALC` and `SOLVE` package.

IM	Interferometer model table. Contains the actual polynomial coefficients which the VLBA correlator used to calculate the geometrical model. Unlike the coefficients in the `CL` table, these have not been re-interpolated onto the `CL` time grid, but have time stamps corresponding to the times at which the correlator computed the geometrical model.
MC	Model Components table. Contains the various components of the geometric model used in the VLBA correlator to generate the `IM` table.
NX	Index File. Contains information about the time, source, subarray and location within the data file of each observation or "scan." It is used by some AIPS tasks in accessing the main data file and subsidiary tables.
OB	Spacecraft Orbit table. Contains information about the positions and velocities used by the correlator for an orbitting antenna.
PC	Phase-calibration table. Contains phases within each IF computed from the injected phase calibration signals. It is used to determine the phase offsets and single-band delays for each IF channel.
SN	Solution table. Contains antenna delay, rate, phase, and amplitude corrections solved for by `CALIB` and `FRING`.
SU	Source table. Contains a list of the sources found within the multi-source file, including information on source positions and flux density
TY	System temperatures table. Contains the system temperature as a function of time for each antenna and IF channel. It is used for amplitude calibration.
VT	VLBA Tape table. Contains tape playback statistics for use mainly by the VLBA correlator group.
WX	Weather table. Contains weather-related information for each station.

9.9 Additional recipe

9.9.1 Banana-Rhubarb Crisp

1. Slice 2 large `bananas` into 1/4-inch rounds. Combine with $2\frac{1}{2}$ cups diced `rhubarb`, 2 tablespoon `sugar`, 1/4 teaspoon `cinnamon`, and a generous dash `nutmeg`. Spoon the mixture into a well-greased 9-inch pie plate or shallow baking dish (preferably glass or ceramic).
2. In a medium bowl, combine 1/2 cup white or whole-wheat pastry `flour`, 1/2 cup `graham cracker` crumbs, $1\frac{1}{2}$ teaspoons `baking powder`. With a pastry blender or two knives worked in a crisscross fashion, cut in 1/4 cup `butter` until the mixture is crumbly.
3. Combine 1 `egg` lightly beaten with 1/4 cup `milk` and stir into the flour mixture. Spoon the batter as evenly as possible over the fruit mixture. Sprinkle with 2 tablespoons `sugar`.
4. Bake in a pre-heated 400deg oven for 25–30 minutes.

Thanks to *Jane Brody's Good Food Book*.