9 Reducing VLBI Data in $\mathcal{AIPS}$

This chapter describes the reduction of VLBI data in $\mathcal{AIPS}$. A Step-by-step recipe, covering both simple and more difficult situations, is presented. See Appendix C for simpler and shorter recipes suitable for straightforward observations. Procedures to simplify some of the VLBI reduction steps are mentioned and become available to you after you enter the command RUN VLBAUTIL C_R. There is also a VLBA pipeline, VLBARUN, which is useful for simple datasets, see § 9.2. If you have very old VLBI data, in Mk III or Mk IV format, see § O.2. We also include here 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 MkII, MkIII, and MkIV data. See § O.2 where we draw the reader's attention to any differences. The few extra steps necessary for calibrating phase-referencing observations are described in § 9.5.7.4. Note that successful phase-referencing observations require careful planning **before** the observations. See VLBA Scientific Memo No. 24 by J. Wrobel, C. Walker, J. Benson, and A. Beasley.

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 organized 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. Space VLBI (SVLBI) data can have different integration times for the ground and space baselines of $\triangle T_g$ and $\triangle T_s$, respectively, and therefore the total disk space requirement is larger. If N_{ant} is the number of ground telescopes,

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

In uncompressed format the same data set will require two-three times the disk space. Be forewarned that some tasks 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 command 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 AIPS files are supported on DEC Alpha, SGI (running XFS), HP, Solaris (revision $\geq$ 2.6), and Linux (kernel $\geq$ 2.4.2). The last three require all AIPS′ C code to be compiled with an additional option. 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 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 available 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 calibration recipe

See Appendix C for simpler and shorter recipes suitable for straightforward observations.

1. LOAD THE DATA For data from the VLBA correlator, run FITLD (§ 9.3.1.1); if needed, follow up with MSORT, USUBA, INDXR, VBGLU, VBMRG, and MERGECAL (§ 9.3.1.4–§ 9.3.1.7). For data from a MkIII correlator, see § O.2. Data from the Penticton correlator should be loaded using FITLD, sorted (MSORT, § 9.3.1.4), and indexed (INDXR, § 9.3.1.6).

   POLARIZATION: The combination of the VLBA correlator and FITLD incorrectly labels polarizations for dual parallel-hand correlation (RR and LL only), even if RR and LL are in different frequency bands (*e.g.*, LL at 5 GHz and RR at 8.4 GHz). For these types of data, you must run FXPOL (§ 9.3.1.8).

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 § 9.4 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 (§ 9.5.2). ANTAB is now the primary AIPS task for loading calibration information from log files.

   VLBA CORRELATOR: The VLBA correlator will usually attach calibration information directly to your data for all VLBA and some other antennas. 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 usually still be loaded from text files.

   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. CORRECT FOR THE IONOSPHERE For low frequency experiments ($<$ 3GHz), TECOR should be run to remove at least part of the ionospheric contribution to the phase offsets. This should also be considered for higher frequencies (*e.g.*, 8 GHz) for abnormal ionospheric conditions.

5. CORRECT FOR THE EARTH ORIENTATION PARAMETERS For phase referencing experiments correlated at the VLBA correlator, particularly between 5-May-2003 and 2-August-2005, CLCOR (OPCODE='EOPS') should be run. This will correct the possibly inaccurate Earth Orientation Parameters used by the VLBA correlator. This is particularly important for astrometry experiments but can effect any phase referencing experiment including those correlated outside the above range of dates.

6. 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 (§ 9.5.3). Some useful tasks for examining and editing data are EDITR, UVFLG, TVFLG, SPFLG, EDITA, BPEDT, FLAGR, FINDR, VPLOT, and QUACK.

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

7. POLARIZATION: ADD PARALLACTIC ANGLE CORRECTION 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 (§ 9.5.4.1). This step should be performed no later than immediately after the T_{sys} calibration.

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

8. APPLY SAMPLER CORRECTION **We now advocate a new amplitude calibration strategy based on VLBA Scientific Memo #37 (Walker 2015). This strategy interleaves the classic *a priori* calibration with instrumental delay and bandpass calibration to improve the calibration of data from the new Roach Digital Backend (RDBE) on the VLBA (see VLBA Observational Status Summary for a description of the RDBE and VLBA Scientific Memo #37 for a discussion of amplitude problems when using the RDBE).** With data from before the RDBE you can use either the old or new strategy. First corrections for sampler biases should be applied using ACCOR (§ 9.5.4.2) for data from the VLBA and some other correlators. Some correlators apply this correction to the data before writing them out — notably the EVN JIVE correlator and correlators used for the Australian LBA. The VLBA hardware and software (DiFX) correlators do *not* apply this correction to the data. Therefore, ACCOR is required for the VLBA correlators and any others that do not apply the correction. ACCOR should be benign (do nothing) for those correlators that do apply the correction prior to reading the data into *AIPS*. Note that you can always run ACCOR and look at the SN table produced with PRTAB or SNPLT to see if it was benign or not.

9. CALIBRATE THE INSTRUMENTAL DELAYS *Phase-cal*s, or measured single-band and multi-band instrumental phase errors, should be applied using PCCOR (§ 9.5.4.3). 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 (§ 9.5.4.4).

   SPECTRAL-LINE: Delay calibration should be carried out only on the continuum sources at this stage. Since there should be no pulse-cals, the "manual" phase-cal method should be used.

   POLARIZATION: If running FRING to determine the instrumental delays, be certain to solve for independent left- and right-polarization delay solutions APARM(3) = 0. Run RLDLY after calibrating the instrumental delays, to determine a single delay offset between left and right polarization (§ 9.5.7.11).

10. CALIBRATE THE COMPLEX BANDPASS RESPONSE FUNCTION Run BPASS or CPASS to determine the bandpass response function using the *cross-power* spectra (§ 9.5.4.5). The normalization

should be over the full bandwidth, be careful because the default channel selection is the inner 75% of the band. If you use the inner 75%, it can lead to amplitude errors of up to 15%.

SPECTRAL-LINE: The bandpass response function should be determined using only the continuum calibrator sources.

11. APPLY ONE MORE AUTOCORRELATION CORRECTION AND THEN FINISH *A PRIORI* CALIBRATION After determining the bandpass calibration the autocorrelations are probably different from unity by a few percent. To correct this run the task `ACSCL`, applying the previous calibration and bandpass correction. Then finally use `APCAL` to complete the *a priori* amplitude calibration (§ 9.5.4.6) — this is called the T_{sys} method of amplitude calibration. `APCAL` can also be used to perform opacity corrections.

    SPECTRAL-LINE: Unless the line emission is very weak, you may wish to defer amplitude calibration of your *line sources only* until step 16 below. The template method described there is much more accurate than the T_{sys} method.

12. FRINGE FIT THE DATA Estimate and remove residual delays, rates and phases using `FRING` or `BLING` and `CLCAL` (§ 9.5.7.7–§ 9.5.7.8).

    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` (§ 9.5.7.10). It is probably better to force the rates to be zero in `FRING` rather than zeroing them after the fitting.

    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 § 9.5.1.2). 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 (§ 9.5.7.4).

13. POLARIZATION: ESTIMATE THE INSTRUMENTAL POLARIZATION . Correct for the instrumental polarization terms, commonly known as 'D-terms' using `PCAL` or `LPCAL` on the polarization calibrator (§ 9.5.7.13). This polarization calibrator should first be fully calibrated and imaged before this step can be performed.

14. POLARIZATION: CALIBRATE THE POLARIZATION POSITION ANGLE 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 (§ 9.5.7.11).

    SPECTRAL-LINE or POLARIZATION: The bandpass response function should be determined using only the calibrator source. Unlike step 7, this step cannot be skipped.

15. 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 and between the different observations (§ 9.5.5).

16. SPECTRAL-LINE: REFINE THE AMPLITUDE CALIBRATION Run `ACFIT` to amplitude calibrate the program source using the template spectra method (§ 9.5.6). Note that the traditional T_{sys} method (§ 9.5.4.6) can also be used if the line emission is too weak for the template method to work successfully.

17. SPECTRAL-LINE:DETERMINE RESIDUAL RATES Now estimate the residual rates ONLY by running `FRING` or `BLING` on one or a few spectral points on the target source (§ 9.5.7.10).

18. APPLY CALIBRATION, AVERAGE, AND INSPECT THE FINAL DATA Run SPLIT or SPLAT to apply the calibration solutions and to average the data in frequency if appropriate (§ 9.6.1), and UVAVG to average the data in time (§ 9.6.2). 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 (§ 9.6.3).

19. 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 § 9.7.

    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, CXCLN, and CXIMAGR) or you can simply edit the LR and RL visibilities to enforce the condition that 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 Pipeline for the VLBA

VLBAPIPE and VLBARUN are two similar pipeline procedures which use the VLBA calibration procedures (from VLBAUTIL) and some logic to calibrate and image VLBA data. They both attempt to make intelligent decisions on defaults, so they can be run fairly automatically, if the names of the sources are known. VLBAPIPE and VLBARUN have been given options to flag data on elevation, to flag excessive fluxes with CLIP, and to apply auto-flagging with RFLAG. The option to clip out bad system temperatures, replacing them with good ones, with TYSMO was also added. The two pipelines differ primarily in the order in which they carry out their operations. Most importantly, VLBAPIPE runs FRING on all calibration scans before computing the bandpass. If the primary calibrator was onserved with multiple scans, this order should help to make the scans phase coherent. If desired, VLBAPIPE and VLBARUN will produce diagnostic plot files and write them to disk creating an HTML file to ease examination of these files. Images will be produced, optionally including self-calibration of the calibration sources. Since only one self-cal, phase only iteration is run, the images should be considered diagnostic in nature.

The two piplines use essentially the same input adverbs, except for the meaning of the DOFLAG parameter. Sample inputs for procedure VLBAPIPE are:

> RUN VLBAUTIL C_R	to acquire the procedures used by VLBAPIPE.
> RUN VLBAPIPE C_R	to acquire VLBAPIPE.
> DATAIN */dirname/data.fits* C_R	to set data file to load from disk.
> OUTDISK *n* C_R	to set disk for output.
> OPTYPE 'CONT' C_R	to say this is a continuum data set.
> CLINT 0 C_R	to use default.
> CHREFANT 'FD' C_R	to set reference antenna to Fort Davis.
> TIMERANG 0 C_R	to have VLBARUN determine a good instrumental delay calibration scan.

> INVER 0 C_R	PC table to use, -1 if you want to force manual phase cal.
> CALSOUR 'CAL-BAND', 'CAL-PHASE', 'CAL-AMP' C_R	to list calibrators, bandpass calibrator *must* be first.
> SOURCES 'CAL-PHASE', 'TARGET', . . . C_R	to list phase referencing and target pairs.
> CHINC -1 C_R	to determine channel increment based on the data channel width.
> INFILE '' C_R	set if you want to apply a zenith delay file produced by DELZN.
> SOLINT 0 C_R	to use default.
> IMSIZE 512 C_R	to make images and specify size of target images.
> FACTOR 0 C_R	to make calibrator images 128x128.
> DOCLIP fp, fc C_R	to run CLIP to flag a sample when the parallel-hand flux is above $\lvert fp\rvert$ or the cross-hand flux is above $\lvert fc\rvert$. If fp is positive, the clip is done after amplitude calibration; if negative, it is done before the bandpass and amplitude calibration. The values for fp and fc are different in these two cases.
> DOFLAG 1 C_R	to run RFLAG on each source individually.
> TYSMOLEV 100 C_R	to clip system temperatures above 100 K with median-window filtering at 10 K and replacing the clipped values with nearby good ones.
> ELEVLIM 15 C_R	to flag out all data for which the elevation is < 15 degrees.
> GET2N *ctn2* ; NMAPS=1 C_R	to use a model for the bandpass calibration.
> SELFCAL 1 C_R	to do one pass of self-cal, phase only on each calibrator prior to the final SPLIT¿
> BPASS5 -2 C_R	to normalize bandpass scans before the fit. Default is +1 to normalize the bandpass functions themselves without normalization of the data.
> BPSOLINT 0 C_R	to make a solution for each calibration scan. BPSOLINT=-1 averages all calibration scans.
> DOBAND 3 C_R	to interpolate the bandpass over time. DOBAND=1 (the default) averages all bandpass records.
> CALTASK 'CALIBS' C_R	to use FRING for the initial time series calibration but use CALIB for any self-cal. Other choices are 'CALIB' and 'FRING' (the default) to use the specified task for all time series calibrations.
> DOPLOT 1 C_R	to make some diagnostic plots; level 2 makes an awful lot of plots.
> OUTFILE '*/directory/name* C_R	to specify directory for output plots. If this is set, plots are written out from AIPS and organized in an HTML file for easy viewing. The plots, HTML file, and message file contents are put in a dated subdirectory below the specified directory.
> OUTTEXT '*email@somewhere.edu* C_R	to specify an e-mail address if the user wants to be notified when the pipeline is done.
> BADDISK 0 C_R	to specify which disks to avoid for scratch.
> VLBAPIPE C_R	to run the procedure.

VLBAPIPE or VLBARUN will then run and produce the requested number of diagnostic plots. For details on the plots produced for each level of DOPLOT, see EXPLAIN VLBAPIPE. If an e-mail address is specified then a VLBAPIPE DONE or VLBAPIPE FAILED message will be sent. However, the VLBAPIPE FAILED message will only be sent if the pipeline failed because of problems with the inputs, if it failed because a task it was running failed this message is not sent. It is highly recommended that the user read the explain file for

VLBAPIPE before use.

The only adverb that differs between the two pipeline scripts is DOFLAG. In VLBAPIPE, the pipeline will shut down after running RFLAG if $0 <$ DOFLAG < 10 to allow you to rerun the pipeline using the RFLAG flag tables. Presumably, the calibration in the second pass should be better. If DOFLAG ≥ 10 in VLBAPIPE or DOFLAG> 0 in VLBARUN the pipeline invokes RFLAG and then continues to the optional self-cal step followed by SPLIT and IMAGR. VLBARUN uses the value of DOFLAG as an initial clip level in RFLAG but only if it is greater than 100. VLBAPIPE does not use the initial clip option in RFLAG, leaving that to the DOCLIP adverb.

Note that editing with CLIP is always a good idea. RFLAG appears to work well at low frequencies but to be overly enthusiastic at C-band and above. Note that the flagging is done after all calibration steps, so you may examine your data with and without the new flags to determine their correctness. RFLAG is run using baseline-dependent clip levels and so should work with arrays containing high-sensitivity telescopes as well as the VLBA. You may wish to use VLBAPIPE without RFLAG (or to discard the flags produced by RFLAG) and then run SPFLG to flag the visibility data interactively. Then use VLBAPIPE over again applying the manually produced flags. Procedure R_RESTART (P_RESTART in VLBARUN) will clear away all plot and calibration tables from your data set, leaving any flag tables. Do not delete version one of the CL and FG tables since they contain the initial calibration and on-line flagging information.

A similar RUN file named VLBAMPHC is available to calibrate, flag, and image data sets for secondary phase stopping centers when DiFX has been used to correlate multiple centers from a single pointing.

In 31DEC21, a similar pipeline was constructed to inspect autocorrelation data for RFI. VLBARFI reduces the data set using many of the steps of VLBARUN. After the amplitude calibration, the pipeline writes out the autocorrelation data with SPLAT and then runs VBRFI to analyze the statistics of the autocorrelations and to write then into a text file and optionally to plot them. Task PLRFI may be used later to plot one or more of these output text files. In 31DEC22, VLBARFI may be run with no calibration and statistics determined with medians using VLBRF rather than VBRFI.

9.3 Loading, fixing and inspecting data

In theory, *AIPS* can process data from multiple frequency bands (FQ numbers in *AIPS* parlance) coexisting within the same data set. However, it is recommended that the data be separated into 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 relevant steps in § 9.3. The VLBAUTIL procedures VLBAFQS and VLBAFIX do this automatically (VLBAFIX is recommended and does other "fixing" tasks like fixing subarrays etc.). If you want to do this by hand use the task UVCOP.

9.3.1 Loading data from the VLBA correlator

9.3.1.1 Running FITLD

The information below applies to data from the VLBA Correlator in what the VLBA archive calls "raw" format, more formally known as FITS-IDI format. The archive now also contains data called "calibrated" which have been run through a few of the simplest VLBA procedures but which are nowhere near calibrated. These files are recommended under normal circumstances. They may also be loaded with FITLD, but the DOCONCAT option does not work with the FITS table format used in the archive.

Data generated by the VLBA correlator are loaded from DAT (or Exabyte) tape (or from disk files) 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. The adverb ANTNAME allows the user to control the antenna numbering if desired. Also set DOCONCAT = 1 CR 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.

Note that antennas, sources, frequency IDs, and other things may be numbered differently in different correlator jobs. FITLD fixes all this for you, but only if you set DOCONCAT = 1 and, better still, load as many files as possible in each execution of FITLD. FITLD can load VLBA correlator data from multiple disk files so long as they have the same name plus a consecutive post-pended number beginning with 1. If you forget to put all the related data together with FITLD you can use MATCH to align the antenna numbers followed by DBCON later.

Typical inputs to FITLD would be:

> TASK 'FITLD' ; INP CR	to review the inputs.
> INTAPE *n* CR	to specify the input tape number.
> NFILES 0 CR	to skip no files on tape.
> DATAIN ' ' CR	to load from tape, not from disk.
> OUTNAME 'TEST' ; OUTCL 'FITLD ' CR	to specify the name of the output file.
> OUTSEQ 0; OUTDI 1 CR	to specify the sequence number and disk of the output.
> OPTY ' ' CR	to load any type of file found.
> NCOUNT 20 CR	to load 20 tape files.
> DOUVCOMP 1 CR	to save disk space by writing compressed data.
> DOCONCAT 1 CR	to concatenate files with same data structure into one disk file.
> CLINT $\triangle t$ CR	set CL table interval to $\triangle t$ minutes (see discussion below).
> DIGICOR 1 CR	to request digital corrections (usually VLBA correlator only).
> DELCORR 1 CR	to request delay decorrelation corrections (VLBA correlator only).
> WTTHRESH 0.65 CR	flag incoming visibilities with correlator weights less than 0.65.
> DOWEIGHT 1 CR	to have the data weights include channel bandwidth and integration time as well as correlator weights.
> SOURCES ' '; QUAL 0 CR	to accept all sources found.
> TIMERANG 0 CR	to accept data from all times.
> BCHAN 0; ECHAN 0; BIF 1; EIF 0 CR	to accept all channels in all IFs.
> SELBAND 0 CR	bandwidth to select (kHz).
> SELFREQ 0; FQTOL 0 CR	frequency to select with tolerance of 10 kHz.
> OPCODE ' ' CR	to not copy the tape statistics table ('VT' table).
> GO CR	to run the program.

This may seem a bit formidable. For straightforward VLBI observations, there is a collection of procedures to simplify matters including the loading of data. Enter

> RUN VLBAUTIL CR	to acquire the procedures; this need be done only once since they will be remembered.
> INTAPE *n* CR	to specify the input tape number.
> NCOUNT 20 CR	to load 20 tape files.
> OUTNAME 'TEST' ; OUTDI 1 CR	to specify the name and disk of the output file.
> DOUVCOMP 1 CR	to save disk space by writing compressed data.

> CLINT $\triangle t$ C$_R$	to set the CL table interval to $\triangle t$ minutes (see discussion below).
> INP VLBALOAD C$_R$	to review the inputs.
> VLBALOAD C$_R$	to run the procedure.

Because the data files tend to be very large, you will usually write compressed data (DOUVCOMP=1). These files take about 1/3 of the space of 'uncompressed' data sets, but cause information about the weights of individual polarizations, spectral channels, and IFs to be lost. There is some loss in dynamic range and sensitivity when the weight information is (partially) compromised. (See Appendix F 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 will have one file per observing mode *per correlation pass*. Data from separate correlator passes can be concatenated using task VBGLU and/or merged with task VBMRG.

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.

It is recommended that corrections for digital representation of the correlated signals be performed in FITLD under control of adverb DIGICOR, but only for data from the VLBA correlator. DIGICOR should be set to one for all continuum and nearly all spectral line experiments. Set DIGICOR to 3 or 4 if the digital corrections are desired for a non-VLBA correlator, *e.g.*, some versions of the DiFX correlator. 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. The details of digital correction for FX correlators can be found in *Radio Science* **33**, 5, 1289–1296, "Correction functions for digital correlators with two and four quantization levels", by L. Kogan.

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. Do this for the VLBA correlator only. 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 drop incoming data with playback weights less than 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 correlator weights close to 1.0. If DOWEIGHT > 0, the actual data weights will be scaled by twice the product of the channel bandwidth in Hz and the integration time in seconds. This should make the weights approximately one over the rms squared. You should examine your data carefully if you use WTTHRESH to make sure that you have not discarded too much data at this stage. Typically 0.8 or higher is good for the VLBA, but for non-VLBA stations a lower value such as 0.6 or 0.7 may be appropriate.

Calibration data have been transferred from the correlator with your data if your data include VLBA antennas and were correlated after 1 April 1999 and before late 2009, when the DiFX correlator came on line and your IMHEADER listing shows the presence of GC, TY, WX, PC and FG tables, as in the example below. If you loaded more than one tape file, you must merge the calibration tables. VLBALOAD does the merging for you. See § 9.3.1.2 for additional details. Note that, as this example shows, it is possible your data have calibration transfer tables even though they were 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 https://science.nrao.edu/facilities/vlba/calibration-and-tools/caliblogs for some information in this regard).

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 is 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, your data were processed with calibration transfer - see § 9.3.1.2 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 is listed as ** rather than TB and that there are no attached

CL and NX tables. This happens when FITLD detects what might be a sub-array condition (two frequency IDs or two sources observed at the same time) on reading the data. In clear cases, the actual simultaneous frequency IDs and sources will be reported. In this case, FITLD detected the use of multiple integration times on different baselines in the data set; this is common for SVLBI data. 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 source subarray condition is encountered, you may need to run USUBA *after* doing any ANTAB that may be needed..
- If the frequency ID subarray condition is encountered, you must separate the frequency IDs into separate data sets; procedure VLBAFQS will do this for you.
- If FITLD does not leave behind CL and NX tables, you must run INDXR to create them. Procedure VLBAFIX will do all of the above for you.
- If you wish to join together data processed in multiple correlator passes, you must run VBGLU and/or VBMRG.

FITLD can also be used to load archived AIPS data previously written to tape using either FITTP or FITAB, 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.

9.3.1.2 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. If your IMHEADER listing shows GC, TY, WX, FG, and PC tables, then the correlator has provided calibration information; this service is called *calibration transfer*. Note that projects correlated at slightly earlier dates may also have calibration transfer information. You must have 15APR99 or later version of AIPS to take advantage of calibration transfer. Not all antennas provide all the information needed for calibration transfer to the VLBA correlator, see

https://library.nrao.edu/public/memos/vlba/ops/VLBAO_34.pdf

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 § 9.5.2. Calibration for the VLA and the GBT began to be transferred with the FITS files in November 2003.

Between April 1999 and late 2009 the information processed by the correlator was somewhat redundant so that the calibration tables, the GC table in particular, must be merged using TAMRG, a very general and hence complicated task. There are a couple procedures to do this for you in the VLBAUTIL package, VLBAFIX and VLBAMCAL, if you have previously run VLBAFIX your tables have been merged:

> RUN VLBAUTIL C$_R$ — to acquire the procedures; this should be done only once since they will be remembered.

> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INP VLBAFIX C_R	to review the inputs.
> VLBAFIX C_R	to run the procedure.

X

You should use VLBAFIX after you have finished loading the data from tape, but before you either change the polarization structure of the data with FXPOL, load any calibration data for non-VLBA telescopes, or apply the calibration data.

At this point it is a good idea to save the tables that were loaded with your data with TASAV. This protects you from having to reload the uv data from scratch if one of the original tables is damaged in some way. If you need to copy a table from the TASAV'ed file use TACOP.

9.3.1.3 Repairing VLBA data after FITLD

As listed above, there are a variety of reasons why VLBA data may need some repair after FITLD has been run. They may need to be sorted into strict time order, to have the subarray nomenclature corrected, to be split into different frequencies, to have the polarization structure fixed, and/or to have the original index (NX) table and calibration ((CL) recreated. These repairs can all be done by the procedure VLBAFIX, which will examine the data and perform any of the necessary fixes. If the data contain subarrays then the procedure must be told to split the data into multiple subarrays (SUBARRAY=2), otherwise it will assume no subarrays and force all the data into one subarray.VLBAFIX is intended to replace VLBASUBS, VLBAFQS, VLBAMCAL and VLBAFPOL, all of which can be run individually instead. Also we have recommended in the last few sections that VLBAFIX be run, if you have already run it, it does not need to be run again.

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CLINT $\triangle t$ C_R	to set the CL table interval to $\triangle t$ minutes (see discussion above in § 9.3.1.1).
> OUTDISK *m* C_R	to specify the output disk when needed.
> INP VLBAFIX C_R	to review the inputs.
> VLBAFIX C_R	to run the procedure.

Remember that all of the VLBAUTIL procedures have HELP files with good discussions about when to use the simple procedures and when to use the tasks directly.

9.3.1.4 Sorting and indexing VLBA correlator data

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. VLBAFIX (§ 9.3.1.3) will sort your data if needed, if you want to do the sorting by hand do the following. 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. MSORT competes with UVSRT in performance even in other cases, particularly when the individual visibility records are large due to many spectral channels and/or IFs. 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 is dangerous, but 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 destroy the integrity of your data set.*

9.3.1.5 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 should 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 task USUBA which 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, *but its results should be checked closely*. Note that ANTAB tables do not know about subarrays that will be assigned by USUBA, so you must run all ANTABs before running USUBA.

If you have subarrays, they need to be sorted, have the subarray nomenclature corrected, and/or have the index (NX) table and calibration (CL) version 1 table rebuilt. In this case, there is a simplified procedure to combine the three repair operation, VLBASUBS. Only use this procedure if you know you have subarrays.

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CLINT $\triangle t$ C_R	to set the CL table interval to $\triangle t$ minutes.
> INP VLBASUBS C_R	to review the inputs.
> VLBASUBS C_R	to run the procedure.

The only user-controllable input is the CL table interval; see discussion above. VLBAFIX will perform this operation if requested (§ 9.3.1.3).

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	to identify subarrays automatically.
> TIMERANG 0 C_R	to include all times.
> ANTENNAS 0 ; SOURCES ' ' C_R	to include all antennas and sources.
> FREQID -1 ; SUBA 0 C_R	to include all frequency IDs and subarrays.
> INFILE ' ' C_R	to use no external file for 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.

In circumstances requiring USUBA, one often wants the calibration in one subarray to apply to other subarrays. USUBA will make the subarray column have value 0 which means all. Other tasks may not be so obliging so you may need to use TABED or verb TABPUT to change tables from subarray-specific to subarray-general.

9.3.1.6 Indexing VLBA correlator data

If FITLD had not written NX or CL tables or it was necessary to sort the data as described in § 9.3.1.4, you must run task INDXR. If VLBAFIX (§ 9.3.1.3) was run this is done automatically. 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	to set the CL interval to $\triangle t$ and recalculate the 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 § 9.3.1.1. VLBAFIX will perform this operation if needed (§ 9.3.1.3).

9.3.1.7 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. Be careful to have FITLD load each pass into a separate disk file; otherwise a very confused data set will be produced. If it is desired to join together the IFs correlated on each pass, the task VBGLU should be used. VBGLU can only handle data sets that have the same polarization and spectral channel parameters. In 31DEC21, it can join data sets from different days while previously it required the oservation date to be the same. If the IFs are not observed at the same time, the file becomes rather larger due to the many flagged IFs. But, this option does allow rather different frequencies to be imaged together correctly. Task MATCH may be used to make the antenna, source, and frequency ID numbers in one data set the same as those in another data set so that they may be used as inputs to VBGLU. Note that MATCH is very fussy about details. You may use RESEQ to renumber antennas and RESOU to renumber sources in order to bring data sets into agreement.

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 no longer important. No data are lost in the revised version of this task.

> 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 2^{nd} input file.
> IN3DISK *n* ; GET3N *ctn* C_R	to specify the 3^{rd} input file.
> IN4DISK *n* ; GET4N *ctn* C_R	to specify the 4^{th} input file.
> OUTDISK *n* C_R	to specify the output disk.
> GO C_R	to run the program.

With the changes in recording technology, it is also possible that a correlator will not have enough playback units for all antennas in an experiment. In this case, multiple correlations will also have to be done in order to correlate every possible baseline. But, inevitably, certain baselines will appear more than once in these correlations. FITLD will load all passes into a single data set (if DOCONCAT=1) or separate disk files which may be concatenated, after MATCH, with DBCON. Note that MATCH is very fussy about details. You may use RESEQ to renumber antennas and RESOU to renumber sources in order to bring data sets into agreement. Sort the data set into 'BT' order with UVSRT. Then task VBMRG may be used to discard any duplicate data. Task DBAPP may be used to avoid the 2^n proliferation of files, but only if the files are fairly similar in antennas, subarrays, and frequency IDs.

9.3.1.8 Labeling VLBA correlator polarization data

The VLBA correlator does not preserve polarization information unless it is operating in full polarization mode. This results in polarizations not being labeled correctly when both RR and LL polarizations are observed without RL and LR. Each VLBA correlator band is loaded into 𝒜ℐ𝒫𝒮 as a separate IF and is assigned the same polarization. FXPOL takes a data set from the VLBA correlator and produces a new data set that has the correct IF and polarization assignments. Unfortunately, there is no reliable way to determine the polarization of each IF from the input data set and you must specify the polarization assignments using the BANDPOL adverb.

Most VLBA setups assign odd-numbered bands to RCP and even-numbered bands to LCP. In this case BANDPOL should be set to '*(RL) ' (the default) and FXPOL will generate a new data set that is of equal size to the input data set, but has two polarizations and half the number of IFs. This case normally applies if LISTR shows pairs of IFs with the same frequency and QHEADER shows one pixel on the STOKES axis with coordinate value RR, but there may be exceptions to this rule when non-VLBA antennas are used.

There is a procedure for use with VLBA-only data that attempts to determine which of the above cases applies and then runs FXPOL for you, if you ran VLBAFIX (§ 9.3.1.3) this has already been done:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INP VLBAFPOL C_R	to review the inputs.
> VLBAFPOL C_R	to run the procedure.

Use VLBAFPOL to check whether you need to relabel the polarizations in your data after loading the data, looking for subarrays, and merging redundant calibration data, but before reading any calibration data from non-VLBA stations. VLBAFPOL assumes that all of your FREQIDs have similar polarization setups. For this reason, you should normally run VLBAFPOL after copying each frequency ID to a separate file using VLBAFQS (§ 9.5). This strategy also reduces the amount of disk space needed for VLBAFPOL.

To use FXPOL directly, typical inputs are:

> TASK 'FXPOL' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> BANDPOL '*(RL)' C_R	to specify the normal VLBA polarization structure.
> GO C_R	to run the program.

Consult HELP FXPOL for further information about more complicated cases. Note that FXPOL has to write a new output file since the structure of the data is being changed. All standard extension files are also converted, but it is still a good idea to run FXPOL before running the calibration tasks.

In single-polarization observations, LL data may simply be mis-labeled as RR or vice-versa. 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 (§ 9.5.2). The Stokes axis can however be modified. Before running PUTHEAD, you should run IMHEAD to check which axis is the Stokes axis in the catalog header.

> INP PUTHEAD CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> KEYWORD 'CRVAL*m*' CR	to select the Stokes or m^{th} axis in the header.
> KEYVALUE = -2 CR	to set the Stokes value to 'LL' (or -1 for 'RR').
> PUTHEAD CR	to set the coordinate value.

9.3.1.9 Ionospheric corrections

At low frequencies (2 GHz and lower) the ionosphere can cause large unmodeled dispersive delays, seen as rapid phase wrapping. This can be of particular importance in phase referencing observations, where phases must be interpolated over weak sources. Even at high frequencies (*e.g.*, 8 GHz) the ionosphere can be important, depending on the experiment and the condition of the atmosphere during the observation. One way to remove at least some of the ionospheric phase offsets is by applying a global ionospheric model derived from GPS measurements. The *AIPS* task `TECOR` processes such ionospheric models that are in standard format known as the IONEX format. These models are available from the Crustal Dynamics Data Information System (CDDIS) archive. There is a procedure which is part of `VLBAUTIL`, called `VLBATECR` that automatically downloads the needed IONEX files from CDDIS and runs `TECOR`. It will examine the header and the `NX` table and figure out which dates need to be downloaded, so the observation date in the header must be correct and an `NX` table must exist. See `EXPLAIN VLBATECR` for other requirements. `TECOR` also determines the ionospheric Faraday rotation which is applied in the calibration of the cross-hand visibilities.

> RUN VLBAUTIL CR	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> INP VLBATECR CR	to review the inputs.
> VLBATECR CR	to run the procedure.

You can also download the files manually from the CDDIS archive through anonymous ftp and run `VLBATECR` or `TECOR`, see `EXPLAIN TECOR` for detailed instructions on how to retrieve the models. `TECOR` interpolates between the maps of electron content in the ionosphere; therefore IONEX files must be retrieved to cover the entire experiment. Presently, each IONEX file contains maps every 2 hours from hours 00:00 to 24:00. Before November 2002, they contained maps every 2 hours from hours 1:00 through 23:00. Therefore, for example, if an experiment prior to November 2002 started at 0:00 then files must be retrieved for the day of the experiment and the previous day so the times between 0:00 and 0:59 can be interpolated. More recent experiments require two or more files only if they occurred in two or more days.

Typical inputs to `TECOR` are:

> TASK 'TECOR' ; INP CR	to review the inputs.
> INDISK *n* ; GETN *ctn* CR	to specify the input file.
> INFILE 'FITS:JPLG1230.01I' CR	to set the name of the IONEX file. If there is more than one file, this name *must* be a standard format and be the first file. See `EXPLAIN TECOR` for more details.
> NFILES *n* CR	to set number of IONEX files to be read.
> SUBARRAY 0 CR	to process all subarrays. This option allows you to process subarrays used on different dates.
> ANTENNAS 1 2 3 4 6 7 8 9 10 CR	to find corrections for all antennas except antenna 5 in a ten antenna experiment. This is important because if the IONEX models do not cover an antenna and it is not excluded here then all the solutions for that antenna will be undefined and the data flagged when the `CL` table is applied.

> GAINVER 1 C_R	to apply corrections to the first CL table.
> GAINUSE 2 C_R	to create CL table 2 with the corrections.
> APARM 1 0 C_R	to correct for dispersive delay; otherwise only the ionospheric Faraday rotation will be corrected.
> GO C_R	to run TECOR, correcting for the ionosphere in a new CL table.

The dispersive delays should be checked using EDITA (option DODELAY=3), SNPLT (options INEXT 'CL'; INVERS 2; OPTY 'DDLY') and VPLOT (options BPARM 0; APARM 0; DOCAL 1; GAINUSE 2). TEPLT plots parameters from the TE table written by TECOR, in particular the Faraday rotation, total electron content, and dispersive delay. It can also plot the difference of two TE tables in multiple parameters. MFIMG can be used to make an image cube of the IONEX data after which TVMOVIE and other image display tools will let you examine the data given to TECOR. TECOR is only as good as the models, which at this time are quite rough. Therefore, it is a very good idea to compare the corrected and uncorrected phases using VPLOT.

Extensive studies have revealed that TECOR usually over-corrects data. It removes the time-dependent IFRM, but leaves behind an extra rotation measure that changes some with obsering date and very slightly with source. A package of software called ALBUS has been found to give somewhat better solutions for the IFRM. In 31DEC24 a task also named ALBUS was written to invoke this package to correct uv data including dispersive delay which is important to VLBI data. Unfortunately, the complications of this package require the Linux operating system and an outside program called apptainer. See EVLA Memo 235 for details.[1]

CLCOR has a OPCODE = 'IONO' to make delay corrections for the ionosphere, similar to the 'ATMO' operation which is for the atmosphere (§ 9.5.7.5).

9.3.1.10 Corrections to the Earth Orientation Parameters

This correction is only useful for experiments correlated at the VLBA correlator. VLBI correlators must use measurements of the Earth Orientation Parameters (EOPs) to take them out of the observations. These change slowly with time and therefore the EOPs used by the correlator must be continually updated. From 5-May-2003 to 9-Aug-2005 the VLBA correlator used old predicted EOPs which could be significantly wrong and will effect all phase referencing experiments. Incorrect EOPs can both move the position and possibly smear the target of a phase referencing experiment. Self-calibration can improve the smearing. Even outside the above quoted period of particularly bad EOPs, the EOPs can be off so it is recommended that all phase-referencing experiments, particularly astrometry experiments should have their EOPs corrected. CLCOR (OPCODE='EOPS') can do this correction. It uses the CT table which is only produced by the VLBA correlator, so at the moment CLCOR can only correct experiments processed at the VLBA correlator. CLCOR also uses a file of measured EOPs, which can be downloaded from NASA (see EXPLAIN CLCOR for details). There is a procedure which is part of VLBAUTIL, called VLBAEOPS, which downloads the file automatically and runs CLCOR. To run the procedure:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INFILE ' ' C_R	to automatically download file.
> INP VLBAEOPS C_R	to review the inputs.
> VLBAEOPS C_R	to run the procedure.

The procedure will correct the highest CL version while copying it to a version one higher.

If multiple days have been concatenated, you should run CLCOR manually. Download the file using the instructions in CLCOR's explain file. Sample inputs are as follows:

[1] "Testing Ionospheric Faraday Rotation Measure Models", Perley R., Greisen, E., Tremou, L., Butler, B., Willis, T., October 2024

> TASK 'CLCOR' ; INP C_R	to review the inputs.
> INDISK *n1* ; GETN *ctn* C_R	to specify the input file.
> OPCODE 'EOPS' C_R	to select correction of the EOPs.
> GAINVER *clin* C_R	`CL` table to read, new default is the current highest version.
> GAINUSE 0 C_R	`CL` table to write; version highest+1 is written unless `GAINUSE = GAINVER`.
> INFILE 'FITS:usno_finals.erp C_R	to specify file with correct EOPs — note missing close quote to retain lower case letters.
> CLCORPRM(2) *dd* C_R	to use *dd* days worth of data following the start date from the USNO file. Set `dd` to about 4 days more than your last day of data.
> GO C_R	to run the program.

In case the automatic download fails, `VLBAEOPS` can use the file which you have downloaded manually.

9.3.1.11 Preparing the `OB` table for SVLBI data

The spacecraft orbit table (`OB`) as produced by `FITLD` contains 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 by task `OBTAB`. Additionally `OBTAB` stores orbital elements in the `AN` table; these are essential for later use in plotting or inspecting spacecraft orbit information. Sample inputs for `OBTAB`, are as follows:

> TASK 'OBTAB' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INVERS 1 C_R	to process `OB` table 1.
> SUBARRAY 1 C_R	to select subarray number.
> APARM 1, 0 C_R	to update orbital elements in `AN` table
> GO C_R	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).

Task `OBTAB` determines mean orbital elements from the `OB` table using the spacecraft positions and velocities and updates 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 and amplitudes can be plotted (by tasks `VPLOT`, `CLPLT`, and `CAPLT`). The orbit table can be plotted using task `OBPLT`. Alternatively, the task `TAPLT` can be used to display individual columns. Use `PRTAB` to determine the names of the columns you wish plotted.

9.3.1.12 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 but can be loaded indirectly. 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	to specify a separate output file.
> OUTNAM 'DUMMY', OUTCLA 'DT'	
> DATAIN 'FITS:3551708.kct.a C_R	to specify the 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 `DATAIN` 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`

9.4 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. Use the verb `IMHEAD` regularly to check the *uv*-data header, particularly the list of tables (as seen in § 9.3.1.1).

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

9.4.1 Textual displays

As a first step, use the procedure `VLBASUMM` to print out the essential contents of your data set:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> DOCRT -1 C_R	to direct the output to the line printer.
> INP VLBASUMM C_R	to review the inputs.
> VLBASUMM C_R	to run the procedure.

This will make a listing of the scans, sources, frequency structure, and antennas found in your data set. You should run this procedure after “fixing” the data with `VLBAMCAL`, `VLBAFQS`, `VLBASUBS`, and `VLBAFPOL`, but you may also find it useful on the initial dataset.

`VLBASUMM` runs the task `LISTR` 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. If you did not do `VLBASUMM`, 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. In 31DEC23, LISUN produces a similar listing plus the angle of each scan to the Sun with fewer adverbs to consider. LISUN also prints a summary of the antennas used. LISTR with OPTYP = 'LIST' and DPARM(1) = 1 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 § 9.3.1.6 and § 9.3.1.4).

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. Some tables have very many columns. You can use input parameter BOX to select the list of the columns you want to print.

3. PRTAN, run by VLBASUMM, provides a listing of the antenna names and their associated antenna numbers. This is useful because antennas are generally specified by their antenna numbers. (Procedure ANTNUM in the VLBAUTIL set allows you to translate station names into 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. Verb GETITIME returns the integration time as INTTIME but has the same problem with multiple integration times.

9.4.2 Spectral displays: POSSM

Your data file will probably contain a number of IFs, observed at different frequencies, corresponding to the separate "IF channels" used during the observations. Use IMHEAD to find the number of IF channels and the number of spectral channels per IF channel, or examine the LISTR or LISUN 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 § 9.5.7 for more information and a sample plot. There is a procedure simplifying the use of POSSM:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCES ' ' C_R	to plot all sources.
> TIMERANG 0 C_R	to plot all times.

> SUBARRAY 0 C_R	to plot all subarrays.
> REFANT *n* C_R	to plot the cross-power spectrum for baselines with antenna *n*.
> STOKES 'I' C_R	to plot Stokes I.
> GAINUSE *CLin* C_R	to apply CL table *CLin* to the data before plotting.
> DOTV 1 C_R	to plot the data on the TV; -1 to make a plot file.
> VLBACRPL C_R	to plot the data.

To use POSSM directly 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.
> SOURCES '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 plot the data without calibration.
> BIF 0 ; EIF 0 ; BCHAN 0 ; ECHAN 0 C_R	to include all IFs and spectral channels.
> STOKES 'HALF' C_R	To plot RR and LL separately.
> CODETYPE ' ' ; POLPLOT ' ' C_R	to plot amplitude and phase.
> SOLINT 0 C_R	to average over the full time range.
> APARM 1 , 1 , 0 , 0 , -180 , 180 , 0 , 0 , 3 , 0 C_R	to control the plot: 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)=3 to plot all IFs and polarizations together in one diagram.
> NPLOTS 9 ; BPARM 0 ; OUTTEXT ' ' 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 for scratch.
> 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} in correlation coefficient). POSSM can produce text output into the file given by OUTTEXT.

Sample POSSM displays are given in Figure 9.1, Figure 9.2, and Figure 10.2.

Task SHOUV with OPTYPE 'SPEC' will display the data from a number of channels on the printer with optional time averaging.

9.4.3 Time displays: VPLOT, CLPLT, and CAPLT

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 and amplitude, you must use tasks CLPLT and CAPLT. Note that VPLOT can average spectral channels or plot them individually under control of AVGCHAN and can plot spectral channels and IFs in separate panels or all together under control of CROWDED. Data points are plotted with a user-selected SYMBOL and may be connected by lines under control of FACTOR. Calibration can be applied to the data before they are plotted. The data can be averaged in time and the max/min within the interval plotted along with the average. 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 are 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.
> SOURCES '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 9.1. 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.

9.4.4 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` now offers options reminiscent of `difmap`, the VLBI data reduction package from CalTech. In particular, the `CROWDED` adverb allows displays of all IFs and/or all polarizations at the same time (and color may be used to separate them visually) and the `FLAG QUICKLY` run-time option allows fast sample deletion with only quick mouse clicks. The new `CLIP EVERYTHING` option is paricularly efficient in deleting high values. The new coherence parameter (scalar average divided by vector average minus one) offers interesting possibilities as well. It is well worth exploring the abilities of this powerful program. Tasks `TVFLG` (§ O.1.6), `SPFLG` (§ 10.2.2), and `WIPER` are also useful in this way.

9.4.5 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 tables and the solution or SN tables.

The task SNPLT should be used periodically to inspect the contents of the latest CL and SN tables. There is a simplified procedure for making these plots:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INEXT 'CL' C_R	to plot a CL table.
> INVERS 0 C_R	to plot the highest version.
> SOURCES ' ' C_R	to plot all sources.
> TIMERANG 0 C_R	to plot all times.
> STOKES ' ' C_R	to plot both R and L solutions.
> SUBARRAY 0 C_R	to plot all subarrays.
> OPTYPE 'AMP' C_R	to look at amplitudes; 'PHAS', 'DELA', and 'RATE' are other useful choices.
> DOTV 1 C_R	to plot the data on the TV; -1 to make a plot file.
> VLBASNPL C_R	to plot the data.

Using SNPLT directly, the example below plots the antenna-based amplitude corrections stored in CL table *m*.

> 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 *m* C_R	to choose CL table version.
> DOBLANK 1 ; DOTV 1 C_R	to plot on the TV screen including failed solutions; otherwise, create plot extension files.
> NPLOTS 5 C_R	to plot 5 antennas/IFs per page.
> DO3COL 2 C_R	to separate the calibration sources by color.
> 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 TY ("system temperature"), and PC ("phase-cal") tables. OPTYPE = 'MULT' allows you to compare values of two or more parameters, plotting them at the same time in separate panels. Task SNIFS is similar and may be used to compare values across IFs. SNBPL is similar to SNPLT, but plots calibartion parameters on a baseline basis. Unstable phases may be stable on some baselines, but not others. The antenna-based plots of SNPLT may not differentiate these adequately.

9.4.6 COHER

The task COHER can be used to determine the coherence time in a *uv* data set broken down both in 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.

9.4.7 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 one or more baselines and time intervals. 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.
> SRCNAME 'DA193' C_R	to select a source.
> TIMERANG 0, 10, 5, 0, 0, 10, 15, 0 C_R	to select a time range, strongly recommended.
> SOLINT 2 C_R	to plot fringe-rate spectra for every 2-minute interval in TIMERANG.
> NPLOTS 6 C_R	to do 6 plots per page.
> STOKES 'LL' C_R	to select a single Stokes.
> BIF 1 ; EIF 0 C_R	to select the IFs included.
> BCHAN 3 ; ECHAN 12 C_R	to select the range of frequency channels averaged within each IF.
> ANTENNAS 3, 4 C_R	to select baseline 3-4, 3-5, and 4-5.
> BASELINE 4, 5 C_R	to select baseline(s) plotted in the familiar way.
> 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.
> BPARM 0 C_R	to do no division by "channel zero."
> DOTV 1 C_R	to plot directly on 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; otherwise the fringe-rate spectrum is plotted. Note that the baseline(s) shown are selected with ANTENNAS and BASELINE in the usual way and a time range must be selected with TIMERANG. APARM(1) sets the integration time to be used before doing the FFT over the selected time range. SOLINT may be used to break the time range into intervals. Separate plots are produced for each IF, baseline, and time interval.

9.5 Calibration strategy

If you have multiple frequency IDs in your data, you may want to separate the data for different FREQID before performing any calibration. Use UVCOP to do this and take advantage of the opportunity to delete data flagged by the correlator with FLAGVER=1. You no longer need to re-run INDXR on the output files. Again there is a procedure to do this for you:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CLINT $\triangle t$ C_R	to set the CL table interval to $\triangle t$ minutes (see discussion above in § 9.3.1.1).
> INP VLBAFIX C_R	to review the inputs.
> VLBAFIX C_R	to run the procedure.

VLBAFIX will normally be run after loading the data.

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 § 9.5.7.

In order to calibrate the absolute phase of their data, VLBI users use either "phase-referencing" or "self-calibration". If the self-calibration route is chosen, then the phase derivatives with respect to time and frequency are calibrated and the absolute phases are normally left uncalibrated. Then self-calibration methods are then used to generate images (see § 9.7). For phase-referencing, an absolute-phase calibration is done using an external calibrator. Phase-referencing in VLBI is similar to, but slightly more complicated than, phase calibration on the VLA. In general terms, phase-referencing VLBI data is accomplished by similar methods as used for VLA data in AIPS; be sure to read § 9.5.1.2 and § 9.5.7.4 for details on how to calibrate phase-referenced observations.

For astrometric data reduction methods in AIPS the reader is referred to the guide to AIPS astrometric 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 § 9.5.4.6. Then, in § 9.5.7, 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, particularly at centimeter wavelengths, the sensitivities on all baselines are roughly the same), the order of the two calibration steps may be reversed.

9.5.1 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 suggested that version 1 of the CL table, as produced by FITLD, be copied to CL version 2 using TACOP before any calibration is begun, and that CL version 2 be used as the starting point in the calibration sequence. An effort is made within AIPS to insure that CL version 1 is not deleted inadvertently. If this does occur however it can be re-generated using task INDXR (described in § 9.3.1.6). (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.) The task TASAV can be used to back up your tables by copying all of them 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 check-pointing progress during calibration.

When you are ready to apply the calibrations, you run either SPLIT or SPLAT. Both tasks can average data in the spectral domain if appropriate. SPLAT can also time-average the data and produces multi-source data sets on output if requested.

9.5.1.1 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 before being used to update CL tables using CLCAL. SNSMO uses superior smoothing methods to those available in CLCAL and should always be used to do any smoothing of VLBI data, *i.e.*, data with non-zero delays and rates. The adverb DOBLANK now controls which data are actually altered by the smoothing; use it carefully.

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	to modify the solutions for all sources.
> SELBAND -1; SELFREQ -1; FREQID 0 C_R	to do all frequency IDs.
> BIF 0; EIF 0 C_R	to include all IFs.
> TIMERANG 0; ANTENNAS 0 C_R	to include all times and antennas.
> SUBARRAY 0 C_R	to select the first subarray; NB, SNSMO works only on one subarray at a time.
> SAMPTYPE 'MWF' C_R	to use the median window filter method.

> SMOTYPE 'AMPL' C_R	to smooth amplitudes only.
> BPARM 0.5,0 C_R	to use a 30-minute filter time for amplitude.
> DOBLANK 1 C_R	to replace blanked values with interpolated values *only*.
> CPARM 0.5, 0, 0, 0, 0, 0.02, 0 C_R	to set ranges of allowed values.
> INVER *snin* ; OUTVER *snout* C_R	to read in the SN table version *snin* and to write SN table version *snout* (which should be a new table).
> REFANT 0 ; BADDISK 0 C_R	to keep the current reference antenna and to allow all disks to be used for scratch files.
> INP C_R	to check the inputs.
> GO C_R	to run the task.

Typical inputs for CLCAL would be:

> TASK 'CLCAL' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR ' ' C_R	to use all corrections in the SN table.
> SOURCE ' ' C_R	to apply corrections to all sources.
> OPCODE 'CALP' C_R	to apply SN tables to a CL table.
> INTERPOL ' ' C_R	to use linear vector interpolation ('2PT').
> SAMPTYPE ' ' C_R	to do no further smoothing of the merged SN tables..
> SNVER *snin* ; INVERS 0 C_R	to select the one SN table containing solutions to be interpolated.
> GAINVER *clin* C_R	to select the CL version to which solutions are to be applied.
> GAINUSE *clout* C_R	to select the output CL version, containing updated calibration information.
> REFANT 0 ; BADDISK 0 C_R	to try to use use a single reference antenna if possible during all steps of calibration.
> INP C_R	to check the inputs.
> GO C_R	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 the highest numbered CL table and GAINUSE=0 means to write a new CL table. If two SN tables contain two similar attempts at finding corrections and SNVER=0 then, effectively, CLCAL will be inconsistent in applying the solutions from these two tables. Basically, CLCAL simply concatenates all SN tables and then merges apparently identical records (same time and antenna) to eliminate blanked solutions and to complain about otherwise non-identical solutions.

The parameters INTERPOL and SAMPTYPE allow the user to choose between several different methods of smoothing the SN files followed by interpolation to the times in the CL table. Use EXPLAIN CLCAL C_R to view all the options. The default interpolation option is INTERPOL = '2PT', in which the SN table is linearly interpolated between the measurements in the SN table. Using SAMPTYPE = 'BOX' causes the SN table to be smoothed with a boxcar function before being interpolated onto the CL table. The smoothing times for delay, rate etc. are specified in parameter BPARM. DOBLANK controls how both failed and good solutions are handled when smoothing. DOBLANK ≥ 0 replaces failed solutions with smoothed ones, while DOBLANK ≤ 0 replaces good solutions with smoothed ones. However it is recommended that SN smoothing be done prior to CLCAL using task SNSMO. When smoothing delays, use the new SMOTYPE = 'VLDE' method whenever the IFs have been grouped to find single solutions (APARM(5)> 0 in FRING) across multiple IFs. Be sure to set NPIECE to indicate how many groups of IFs were used in FRING, *e.g.*, 1 for a multi-band delay, 2 for a delay for IFs 1 — $Nif/2$ amd a delay for IFs $Nif/2+1$ — Nif.

With good quality data, the INTERPOL = '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 SOURCES 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 SN table entries closest in time for that source are used with 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 version 2 table and for non-VLBA antennas in SN version 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. Note that you must use OPCODE = 'CALI' when building up a CL table in pieces.

9.5.1.2 Running CLCAL for phase referencing observations

In particular, this example illustrates how to set the inputs for CLCAL for the specific case when phase corrections determined for the cal source 'J1636-16' are to be transferred to the target source 'P1643-12' in a phase referencing experiment:

> TASK 'CLCAL' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR 'J1636-16', ' ' C_R	to use the corrections determined for the cal source.
> SOURCES 'J1636-16', 'P1643-12', ' ' C_R	to apply corrections to both the cal source and the target source.
> OPCODE 'CALP' C_R	to apply SN tables to a CL table passing all data.
> INTERPOL 'AMBG' C_R	to use linear vector interpolation with no SN table smoothing and simple phase ambiguity removal. See above for more discussion of INTERPOL.
> SAMPTYPE ' '; BPARM 0 C_R	to clear smoothing parameters.
> SNVER *snin* C_R	to select the SN table containing solutions to be interpolated.
> GAINVER *clin* C_R	to select the CL version to which solutions are to be applied.
> GAINUSE *clout* C_R	to select the output CL version, containing updated calibration information.
> REFANT 5 ; BADDISK 0 C_R	Use a single reference antenna if at all possible during all steps of calibration.
> INP C_R	to check the inputs.
> GO C_R	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.

9.5.2 Processing observing log and calibration information

As of 1 April 1999, the VLBA correlator provides calibration transfer information, as described in § 9.3.1.2 for VLBA antennas. Experiments correlated after November 2003 also have full calibration transfer for the VLA, the GBT, Arecibo and the Bonn 100m. Consequently, **you can skip § 9.5.2 entirely** unless you have data from non-VLBA telescopes (*e.g.*, the VLA before November 2003, Space, other European telescope) or other correlators 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 § 9.5.2.3, § 9.5.3, § 9.5.4.6 and § 9.5.4.3.

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.

9.5.2.1 Automatic formatting of VLBA and VLBA-like log files

For VLBA antennas, the external calibration file for a given experiment can be downloaded from
`https://www.vlba.nrao.edu/astro/VOBS/astronomy/`*mmmyy*`/`*xxxxx*`cal.vlba.gz`
where *mmmyy* is the month and year (*e.g.*, `aug03`) and *xxxxx* is the project code (*e.g.*, `bz199`). The calibration file is named as above and is in GNU-zipped format. Gain curve information can be obtained from the same web site in the `astronomy` directory (*i.e.*, two directories up from the project directory). The gain curve 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	to select the required subarray.
> CALIN 'FITS:bz199cal.vlba C_R	to specify the input external calibration file.
> OUTFILE 'FITS:BZ199' C_R	to define the directory and prefix for the output files.
> FQTOL 1000 C_R	to set the tolerance for frequency match in kHz; one channel width is recommended.
> PRTLEV 0 C_R	to limit output; in particular to avoid echoing the calibration file to the screen.
> GO C_R	to run the program.

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

1. `.TSYS`: T_{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_{sys} values at the end of scans immediately before source changes to avoid interpolation problems for sources of greatly

differing flux density or use INTERPOL = 'SELF' in CLCAL. Occasionally spurious data from previous observing runs or system start-up 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.

9.5.2.2 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 or, possibly, if it contains multiple frequency bands. 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 T_{sys} information from the calibration text file. You will also need to append gain curves to the T_{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. The 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.

9.5.2.3 Loading calibration log information

The calibration information in the external text files such as T_{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_{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_{sys} information. The frequency and polarization association for each IF channel in the AIPS file can be compared (use LISTR with OPTYPE = 'SCAN' or LISUN) 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.

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. Use SETJY to insert flux densities if necessary.

The parameter TIMEOFF in the input file adds a time offset to the all entries. Non-VLBA stations sometimes measure the system temperature between, rather than during, scans causing ANTAB to be unable to match the measurements with the source and frequency ID. The ANTAB input parameter OFFSET serves the same purpose, but is more successful since the scan times are expanded at both ends.

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 fail. If ANTAB quits 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.
> CALIN 'MYVLB:BC25CAL.VLBA' C_R	to specify the text file.
> SUBARRAY 1 C_R	to select subarray one.
> TYVER 0 ; GCVER 0 C_R	to create new TY and GC tables.
> BLVER 0 C_R	to create new BL table for any specified baseline factors.
> PRTLEV 1 C_R	to 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.

9.5.2.4 Generating VLA amplitude calibration with VLAMP

Before February 2013, the VLA's *a priori* calibration was done with calibration files, named *xxxxx*cal.y.gz (stored in gzipped format). These were obtained from the same server and disk directory as for VLBA files. This VLA file started with an explanatory preamble, including minor editing instructions. See § C.11 for more detailed instructions.

After February 2013 and the advent of the phased-EVLA, either an ANTAB style file was provided to the observer or the calibration information is in the data file. This calibration information is generated using the EVLA switched power (SY table, see § 4.3.15), the known gain curve (GC table), and the first CL table using the task VLAMP. Sometimes there are problems with the values in the SY, GC, and/or CL tables and the observer might want to recreate the amplitude calibration of the phased-EVLA after editing one or more of them. To do this the phased-EVLA only data (not the VLBI data) must be loaded into *AIPS* with BDF2AIPS, see § 4.1.1. Then the SY and/or GC tables can be edited. For instructions on how to edit or smooth the SY table see § 4.3.15. Then run VLAMP, which will create an output text file that can be loaded into the VLBI data with ANTAB. Typical inputs for VLAMP would be:

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

> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> FLAGVER 1 C_R	to apply flags to the input tables.
> INVER 1 C_R	to specify the SY table.
> GAINVER 1 C_R	to specify the CL table.
> IN2VERS 1 C_R	to specify the GC table.
> VPARM 5 6 11 12 C_R	to select the EVLA spectral windows used by the VLBA.
> OUTTEXT '*filename*' C_R	to specify the name of the output text *filename*.
> GO C_R	to run the program.

The RUN file and procedure named DOVLAMP can be used to run the full process, reading the data into AIPS with BDF2AIPS, smoothing and applying the SY tables with TYSMO and TYAPL, correcting standard calibration source amplitudes with special usage of CALIB and CLCAL, and finally running VLAMP to produce the needed file for ANTAB.

9.5.3 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 EDITR and IBLED may be used for interactive baseline-based editing. In both single- and multi-source data formats, flagging information is stored in the FG table instead of being used to discard data directly. One may also undo a flag operation using OPCODEs 'UFLG', 'REAS', and 'WILD' in UVFLG. Note that this operation works *only* when the operation being undone is in the current flag table. (Note that many tasks delete fully flagged data when copying a data set.)

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	to flag all sources, which is usually desired.
> SUBARRAY 0 C_R	to select the required subarray, all in this case.
> FREQID -1 C_R	to flag all frequency IDs.
> OUTFGVER 2 C_R	to specify the output flag table; use 2 *only* if you copied FLAGVER 1 to 2 as suggested in § 9.3.1.2.
> INTEXT 'MYVLB:BC25CAL.FLAG' C_R	to specify the input 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 C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCES 'DA913' , ' ' C_R	to select one source.
> SUBARRAY 1 C_R	to select subarray one.
> ANTENNAS 3, 4, 5, 8 C_R	to flag only certain antennas.
> APARM 0, 0, 0, 0, 10 C_R	to flag data between elevations 0 and 10 degrees (APARM(4) to APARM(5)), with no flagging on amplitude or weight.
> OUTFGVER 2 C_R	to specify the output flag table.

> GO C_R   to run the program.

This will generate FG table flagging time ranges that fall between the specified elevation limits.

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 the selected portion of the scan at its beginning and/or end. 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. The first second or so after a scan may also need to be flagged (OPCODE 'TAIL') since some antennas may not leave the previous source promptly, leaving bad data marked good. QUACK can be used to flag specific antennas for these and other reasons.

Tasks EDITA and EDITR can be used to inspect and edit the data interactively. IBLED is no longer of much use since EDITR now can edit on a coherence parameter. These tasks are similar in many respects to TVFLG, 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 § 4.3.5 through § 4.3.11 and § 5.5 for more details on the editing of data. Also VPLOT may be used to edit data which deviates excessively from the mean amplitude over a specified averaging interval; see HELP VPLOT C_R for details. Task WIPER is a dangerous but powerful editing tool as well.

The task DEFLG generates a flag table to delete data having coherence less than a user-specified limit. It must be run only on sources that should be strongly coherent although it may be used to flag data from other sources in between the strong-source scans. The task SNFLG is also used to flag data whenever the phase jumps in an SN or CL table are excessive on a baseline-by-baseline basis.

The VLBA correlator may produce a lot of samples which it knows to be bad and which are flagged by the transferred flag table. For this reason alone, or if you also flag a noticeable fraction of the data set, you may wish to run UVCOP to discard the flagged data to conserve disk space and processing time.

9.5.4 Amplitude and instrumental delay calibration

We now advocate a new amplitude calibration strategy base on VLBA Scientific memo #37 (Walker 2015). This strategy interleaves classic *a priori* calibration with instrumental delay and bandpass calibration to improve the amplitude calibration of data from the new Roach Digital Backend (RDBE) on the VLBA (see the VLBA Observations Status Summary for a description of the RDBE and VLBA Scientific Memo #37 for a discussion of amplitude problems with the RDBE). With data from before the RDBE you can use either the old or new strategy.

9.5.4.1 Parallactic angle correction

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 and phase-referencing observations, so it should probably be applied to all cases.. Task CLCOR is used for this purpose; see § 9.5.4.1.

There is a procedure which assists you in running CLCOR to correct phases for parallactic angle:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SUBAR *ss* C_R	to select the subarray number — only one per execution.
> INP VLBAPANG C_R	to review the inputs.
> VLBAPANG C_R	to run the procedure.

VLBAPANG should be run before applying any other phase corrections.

9.5.4.2 Digital sampler bias corrections for VLBA correlator data

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 and from polarization to polarization. This sampler bias, which is usually significant only in two-bit quantization, introduces an antenna/polarization-based amplitude offset. In full polarization observations this appears as an amplitude offset between RR and LL. The cross-correlation amplitudes may be corrected if the auto-correlation spectra have been measured. See VLBA Scientific Memo No. 9 (1995, "Effect of digitizers errors on the cross and auto correlation response of an FX correlator", by L. Kogan). Task ACCOR can be used to remove these digital sampler biases from VLBA correlator data, if FITLD was run with DIGICOR = 0, 1, or 2. ACCOR corrects these offsets by examining the autocorrelation spectra. You can check your autocorrelation data with task ACIMG which makes an image cube with axes of frequency, time, and antenna number. Use the image display options describe in Chapter 6 to inspect the image. There may be data that needs flagging. 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. If the ACCOR gain factors deviate from unity, there may be overall scaling or *b*-factor errors. For 2-bit quantization, however, the amplitude offsets for the VLBA correlator are typically of order $5-10\%$, but values as high as 20% have been observed.

For one-bit quantization, no significant sampler bias correction is expected. Nonetheless, it is recommended that ACCOR be run on one-bit data as a consistency check. VLBACCOR is a procedure which runs ACCOR, smooths the results with SNSMO, and applies the solutions using CLCAL. Typical inputs would be:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> FREQID *ff* ; SUBAR *ss* C_R	to select the frequency ID and subarray numbers — only one of each per execution.
> INP VLBACCOR C_R	to review the inputs.
> VLBACCOR C_R	to run the procedure.

VLBACCOR will normally be run after "fixing" the the data with VLBAFIX (if needed). Use this procedure only on data from the VLBA correlator.

If you do not want to use VLBACCOR, 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	to select data from all times.
> SOLINT 2 C_R	to set the solution interval for sampler corrections (min).
> GO C_R	to run the program.

The correction factors were expected to be fairly stable over time, but they have been found to vary over times less than an hour. With a solution interval of a few minutes, 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 interactive table editing task SNEDT can be used.

9.5.4.3 Instrumental phase corrections

If you run POSSM on a short (≈ 1 minute) section of data on a strong calibrator using the inputs described in § 9.4.2 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 system can inject narrow band signals ("phase-cals") into the data recorded at each antenna from which the IF channel phase offsets, and the instrumental single-band delays, can be determined.

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 § 9.5.4.3). If you have other antennas 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 PCCOR without throwing away data for the antennas with missing phase-cal information.

If external phase-calibration data (pulse cals) 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 § 9.5.4.4). Even when phase-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 inspect the data using POSSM at times different from the time used to determine the manual phase calibration. Note that time-dependent delays may still be seen because of low elevation and ionospheric effects.

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. If you are unsure whether your VLBI data has phase-cal information, use IMHEAD to list the extension tables and look for a PC extension.

Phase-calibration data in a PC table can be used by task PCCOR to generate an SN table which corrects for the single-band instrumental phase and delay offsets (note that PCCOR uses two phase-cals in each IF). A short calibrator scan must be specified to be used by PCCOR to resolve any 2π phase ambiguities in the phase-calibration data. The specified time range must include at least one PC table entry for each antenna appearing in the PC table. Having resolved the 2π phase ambiguities, PCCOR uses the whole PC table to calculate entries in an SN table for all times (not just the TIMERANG used to resolve the ambiguities). The procedure VLBAPCOR runs PCCOR, FRING (if needed because of missing pulse-cals for some antennas), and CLCAL for you. Inputs are:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> TIMERANG *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to specify a short scan on a calibrator. There is no default.
> REFANT *m* C_R	to select a particular reference antenna.
> SUBARRAY 0	to do all subarrays.
> CALSOUR '*cal1*', ' '	to specify the calibrator source name.
> GAINUSE *CLin* C_R	to indicate the CL table with all calibration up to this point.

> OPCODE 'CALP' C_R — to indicate that there are antennas with no usable pulse cals; use `OPCODE ' '` if all antennas have pulse cals.
> ANTENNAS *a1 a2 a3* C_R — to solve for antennas *a1*, *a2*, *a3* "manually" (using `FRING`).
> VLBAPCOR C_R — to run the procedure.

This should be done after the sampler correction (`VLBACCOR`) and before the complex bandpass, The `CALP` option requires the data in the specified `TIMERANG` to include strong fringes for those antennas lacking phase-cal data.

Running the tasks individually, 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 — to isolate a short scan on a calibrator; *no default*.
> SNVER 0 C_R — to create a new `SN` table.
> INVER 1 C_R — to specify the input `PC` table version.
> REFANT 5 C_R — to specify reference antenna.
> SUBARRAY 1 ; FREQID 1 C_R — to set subarray and freqid.
> CALSOUR '3C345',' ' C_R — to 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 linear gradient with frequency due to residual delay errors. Unless these conditions hold for all baselines, you should proceed to § 9.5.4.4.

The DiFX correlator has the capability of writing a large number (1 per MHz typically) of pulse cals into each IF. AIPS can handle this with new tasks `PCFLG` (flags `PC` data with a `SPFLG`-like TV display), `PCEDT` (flags `PC` data with a `BPEDT`-like TV graphical display), `PCPLT` (plots `PC` table data all antennas with one time per page or all times with one antenna per page), `PCASS` (finds *amplitude* bandpass), and `PCFIT` (fits delays and phases to `PC` data, writing an `SN` table of the changes in these as a function of time). `PCLOD` can now read `PCAL` files written by DiFX. Other tasks include `PCAVG` to average pulse-cal tables in time and `PCRMS` to edit pulse-cal tables automatically from internal statistics. `POSSM` can also now plot `PC` table data (`APARM(8) = 9`. This package of routines remains experimental, but it has been shown do good things with the data. It is now documented in AIPS Memo 123.[2] New tasks in `31DEC21` include `VLBSY` to read VLBA total power files, `VLBCD` to read noise tube values from a VLBA cal file, `PCABL` to copy cable cal values from the old `PC` table to ones created by `PCLOD`, and `PCDIV` to scale pulse-cal amplitudes by the system temperatures.

9.5.4.4 "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

[2]Greisen, E. W. 2025 (revised), "New Pulse-cal Capabilities for VLBI in AIPS" AIPS Memo 123,
`https://www.aips.nrao.edu/aipsdoc.html`

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* ; GAINUSE 0 C_R	to specify which `CL` table to modify after copying it to a new table.
> 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 an SVLBI data set, you may wish to run `CLCOR` with `OPCODE = 'CLOC'` and `CLCORPRM = 0,` *offset*`, 0, 0, 0, 0, 1, 0.`

Now you can determine the manual phase correction for one or two strong calibrator scans for which all the antennas are present using `VLBAMPCL`. Task `BSCAN` may be used to select the scan(s) to be used at this stage. `VLBAMPCL` runs `FRING` and `CLCAL` once or twice, depending on whether one or two scans are used. Choose a scan with strong fringes to all antennas; if none exists, find a second scan that has strong fringes to the antennas missing from the first. Note that if you use 2 scans the `REFANT` must have good fringes in both scans. Typical inputs are:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> TIMERANG *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to specify a short scan on a calibrator. There is no default.
> REFANT *m* C_R	to select a particular reference antenna.
> SUBARRAY 0	to do all subarrays.
> CALSOUR '*cal1*', ' '	to specify the calibrator source name.
> GAINUSE *CLin* C_R	to indicate the `CL` table with all calibration up to this point.
> OPCODE 'CALP' C_R	to indicate that there are antennas with no fringes for the scan in `TIMERANG`; use `OPCODE ' '` if all antennas will be corrected by the first scan.
> TIME2 *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to specify a short second scan on a calibrator.
> CALSOUR '*cal2*', ' '	to specify the calibrator source name for the second scan.
> ANTENNAS *a1 a2 a3* C_R	to solve for antennas *a1*, *a2*, *a3*.
> VLBAMPCL C_R	to run the procedure.

If you wish to run `FRING` separately, 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 § 9.5.7.7. Note that it is simplest to choose a single short section of data within a single scan using `TIMERANG` and to set `SOLINT` equal to the scan length 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.
> TIMERANG 0, 16, 0, 0, 0, 16, 2, 0 C_R	to select a single short scan.

> DOCALIB 1 ; 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 the most recent flag table.
> SMODEL 0 C_R	to use the null (point-source at the origin) source model.
> REFANT 5 C_R	to specify a reference antenna that will give fringes to most other antennas.
> SOLINT 0 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	to use only 1 baseline in the initial (coarse) fringe search.
> DPARM(9)=1 C_R	to suppress fitting of rates; rates will be 0 in 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 was 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 (the same REFANT would be best) 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 § 9.5.1.1. To apply the calibrator solutions to the other sources in the data file, set CALSOUR to the calibrator source used when running FRING (in the example above, CALSOUR '0954+658') and set SOURCES = ' ' for all sources. 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 ; if you do this, you must first be careful to delete all SN tables except those generated by FRING. 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 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, although it may not be centered on zero. Run POSSM on another scan containing a strong calibrator to check that the assumption of constant IF phase offset holds.

9.5.4.5 Bandpass calibration

Full bandpass response calibration should be performed for all observations. 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. For this amplitude calibration scheme the cross-correlation (complex) bandpass must be used. 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 cross-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.
> DOCALIB 1 C_R	to apply calibration.
> SOLINT 0 C_R	to average data over whole scans before determining the bandpass.
> BPASSPRM 0 C_R	to use the cross-spectra.
> BPASSPRM(5) 1 C_R	to not divide by "channel 0."
> BPASSPRM(9) 1 C_R	to interpolate over flagged channels.
> BPASSPRM(10) 6 C_R	to normalize the amplitude solutions using the *power* solutions rather than the actual solutions which are voltages.
> ICHANSEL $chan_{beg}$, $chan_{end}$, 1, IF_{num}, ... C_R	to set channels for entire bandwidth for normalization, if this left 0 then the inner 75% is used and can cause up to 15% error in the amplitude, so unfortunately ICHANSEL must be explicitly set to all the channels in the IF, for every IF.
> GO C_R	to run the program.

Be careful with the adverb SMOOTH. If you smooth, or do not smooth, the data while finding a bandpass solution, then you must apply the same SMOOTH adverb values whenever you apply that bandpass solution to the data. The only exception is that you may smooth the data after applying the bandpass solution with SMOOTH(1) values 5 through 8 when you did no smoothing in BPASS.

This will produce a BP table containing the antenna-based bandpass functions to be applied to the data. You should check your results very carefully. The BP tables can be plotted with POSSM or printed with PRTAB. Note that this task merely creates a BP table. To use this BP table, set BPVER and DOBAND as described in § 4.3.8 when running any later *AIPS* tasks.

Note, if your bandpass calibrator is not stable or strong enough during the observation (your bandpass calibrator data should have a better S/N than the data you're trying to correct with the bandpass), you could consider using the phase reference source (if there is one) and use SOLINT = -1 (include all scans to make one bandpass solution). If the bandpass calibrator is strong enough, but the average phase varies through the scan, then divide each record by "channel 0" by setting BPASSPRM(5) = -1 to adjust phase only. You should select a range of channels that have similar phases to be averaged as channel 0 using adverb ICHANSEL

The normalization of the bandpass correction can be checked by running ACSCL after BPASS. This task is a version of ACCOR which applies the current calibration tables to the data before forcing the autocorrelations to unity. Be sure to set DOBAND to apply the bandpass by your preferred method. ACSCL will write an SN table with amplitude corrections which should be very close to one. Apply that table with CLCAL. However if you run VLBAAMP, ACSCL and CLCAL will be run for you, see § 9.5.4.6.

The interactive data editing task BPEDT lets one flag visibility data based on bad bandpass solutions. It can edit the BP table instead and even interpolate over the flagged values. It is a graphical editor much like EDITR and EDITA except that it displays bandpass solutions as a function of spectral channel. It is a quick way to check your bandpass table and, if necessary, to flag bad channels in your calibration source. If you do generate new data flags, you should run BPASS over again to generate a corrected BP table. Set BPASSPRM(9) = 1 to interpolate over any fully flagged channels. For really large BP tables, you may wish to use BPFLG instead after determining clip levels with BPEDT although BPEDT has been made much faster for such BP tables.

9.5.4.6 Continuum amplitude calibration

After bandpass calibration is determined, the auto-correlations can be slightly offset from unity. To correct this offset ACSCL can be run, applying all the previous amplitude and bandpass calibration. Described below is a procedure that is part of VLBAUTIL called VLBAAMP that runs ACSCL as well as the final amplitude calibration steps.

The TY table can be examined using SNPLT with OPTYPE='TSYS' or 'TANT' or LISTR with OPTYPE='GAIN'; the GC table is examined with PRTAB. SNEDT may also be used to inspect the TY table and even used to delete or smooth some of the measurements. SNRMS may be used to examine statistics of the TY table. TYSMO can be used to delete discrepant system temperatures and replace the bad and/or good values with time-smoothed values. Note, however, that anomalously high system temperatures may indicate possible bad data (*e.g.*, due to weather) rather than bad measurements of the system temperature. On occasion, RFI in some IFs can render the values for those IFs in the TY table unusable. Task TYCOP will copy TY data between IFs and polarizations in a variety of ways in an attempt to replace the bad data with something reasonable even if not exactly correct for that antenna and IF. Note that flagging bad *uv* data will not change the appearance of the TY table since no flags are applied in plotting this table. EDITA may also be used to examine the TY data interactively and, if bad system temperatures are found, to flag the associated *uv* data. (This task does flag displayed $T_{\rm sys}$ when the data are flagged.) Tasks UVCOP and SPLAT will apply a flag table not only to the *uv* data, but also to the TY and SN tables. APCAL can then be used to derive an amplitude calibration (SN) table.

Atmospheric opacity becomes significant at high frequencies ($\geq$ 15 GHz). APCAL can fit for an opacity correction, if needed, using weather information and system temperature. The weather information can be taken from the WX table by setting INVERS (after October 7, 2003) or loaded from disk by setting CALIN. In order to have APCAL fit for opacity set OPCODE to 'GRID', 'OPAC' or 'LESQ' and DOFIT to 1 (both of these are necessary for an opacity fit). OPCODE 'GRID' and 'OPAC' need an initial guess for the receiver temperature (TRECVR and zenith opacity (TAU0). APCAL will estimate the initial guesses from the data if TRECVR and/or TAU0 are 0. If OPCODE is set to 'GRID', 'OPAC' or 'LESQ' and DOFIT ≤ 0 then the opacity correction is applied using the provided TRECVR. This is a good option if you have a reliable measurement of the receiver temperature. In 31DEC20, use OPCODE 'PLOT' to plot the data against the solution you wish to apply to make sure it is reasonable. The fits are fairly robust, but the plots that APCAL makes should be examined. Note: a large number of bad Tsys values can make the fits unreliable. APCAL will warn you if fits appear to be incorrect. The procedure VLBAAMP runs ACSCL, smooths the results with SNSMO, runs APCAL, and applies the solutions using CLCAL.:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> FREQID *ff* ; SUBAR *ss* C_R	to select the frequency ID and subarray numbers — only one of each per execution.
> INP VLBAAMP C_R	to review the inputs.
> DOFIT 1 C_R	to enable the opacity correction; ≤ 0 disables it.
> VLBAAMP C_R	to run the procedure.

VLBAAMP will normally be run after correcting the bandpass and loading any gain curves or system temperature data for non-VLBA antennas using ANTAB.

If you have not used VLBAAMP and do not want to correct for atmospheric opacity, 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~ to select all antennas and subarrays.
> SOURCES ' ' ; STOKES ' ' C~R~ to select all sources and Stokes.
> BIF 1 ; EIF 0 ; FREQID 1 C~R~ to select all IFs of frequency ID 1.
> TIMERANG 0 ; OPCODE ' ' C~R~ to select all times and use no opacity solutions.
> TYVER 0 ; GCVER 0 C~R~ to use the latest TY and GC tables.
> SNVER 0 C~R~ to create a new SN table.
> GO C~R~ to run the program.

If atmospheric opacity correction is desired, set the following inputs as well as the above:

> OPCODE 'GRID' C~R~ do a grid search.
> DOFIT 1 C~R~ to fit the opacities.
> INVERS 1 C~R~ to use WX table 1.
> TAU0 0 ; TRECVR 0 C~R~ to let APCAL estimate initial values.
> GO C~R~ to run the program.

The resulting solution (SN) table can be smoothed and clipped using SNSMO and applied using CLCAL as described in § 9.5.1.1). Substantial smoothing of the TY table (task TYSMO), especially for VLBA-only observations, is not generally recommended since variations of the system temperature often reflect a real response to the weather. Note, however, that TYSMO is very useful for removing bad system temperatures before they are applied. 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 TY information is associated with the correct source and SNEDT can be used to delete occasional bad system temperature measurements before they are applied to the data. When opacities are fit, APCAL generates plots of the receiver temperature versus zenith angle and the opacity versus time. APCAL will warn you about bad fits, but the plots should be checked for problems with the data or the fits.

On occasion, it has been noted that plots of system temperature show unreasonable variation between IFs. This suggests that the T_{cal} values may have systematic errors as a function of frequency. APCAL has an option to bring the T_{sys} values into better agreement with either their average or with a user-chosen IF, thought to be probably accurate. APARM(6) > 0 selects this function while APARM(7) may be used to specify a specific IF to calibrate the others. Note that a single value is used for each antenna/IF over all times.

The DiFX correlator in now capable of correlating multiple phase-stopping positions within a single antenna beam (pointing position). There is task called CLVLB designed to apply corrections to a CL table (writing a new one) for the offset of phase-stopping and pointing position. The task has internal tables for the beam shape and squint of the VLBA antennas and can take user input for these parameters for other antennas. This is a new task and should be used with caution — early results with it were less than satisfying — although the corrections it applies are certainly in the right direction.

9.5.5 Spectral-line Doppler correction

Normally, when observing, you will have kept the frequency constant throughout the run for ease of observing. 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. The VLBA correlator compensates for the revolution of the antennas relative to the center of the Earth. In this case, the only movement which CVEL should compensate is the rotation of the antennas together with the Earth around the Sun. This movement should be the same for all antennas and gives a smaller effect than the rotation relative to the center of the Earth. Nonetheless, CVEL is required for the VLBA correlator at least to be able to compare observations made at different times. Without CVEL such comparisons will show the velocity shift of the Earth's orbit about the Sun.

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

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

Then run CVEL:

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

9.5.6 Spectral-line amplitude calibration

The calibration strategy suggested in AIPS for spectral-line VLBI data utilizes the total-power spectra method described in Lecture 12 of *VLBI, Techniques and Applications*, eds. Felli and Spencer, published by Kluwer Academic Publisher, 1988. The continuum method using T_{sys} values (see § 9.5.4.6) 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 C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> SOURCE 'OH127.8', ' ' C_R	to write the program source.
> BCHAN 0 ; ECHAN 0 C_R	to write all spectral channels.
> DOCALIB -1 C_R	to avoid applying any calibration.
> DOBAND -1 C_R	to skip the bandpass correction since the data will have been corrected already in CVEL.
> TIMERANG 0 22 0 0 0 22 30 0 C_R	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 C_R	to pass only self-spectra.
> GO C_R	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-dependent amplitude gain correction factors into an SN table.

> TASK 'ACFIT' ; 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 template file.
> CALSOUR 'OH127.8 ' C_R	to select the source to use for calibration.
> DOCALIB -1 C_R	to avoid applying any previous calibration.
> DOBAND -1 C_R	to skip the bandpass correction since it was done when CVEL was run.
> SOLINT *n* C_R	to average the self-spectra over *n* minutes (*e.g.*, 10) before doing the least-squares fit.
> REFANT 1 C_R	to select the desired reference antenna from the template file.
> BCHAN 50 ; ECHAN 70 C_R	to set the range of spectral channels over which the fit is performed.
> 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_{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.
> APARM 0, 0, 50, 0, 0.72 C_R	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.
> 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.

9.5.7 Phase calibration

After carrying out amplitude calibration, the remaining calibration steps involve correcting the phase between the different integration periods. This will allow averaging of the data over both frequency and time without loss of coherence. The phase offsets may be corrected using *a priori* 'phase-cal' measurements if available and/or by directly fitting to the data.

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 § 9.5.7.7. Two alternatives to `FRING` are the tasks `BLING` and `BLAPP` and the experimental version of `FRING` named `KRING`. `BLING` and `BLAPP` are discussed briefly in § 9.5.7.8. `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`. 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. For this reason it is probably wise to run through the fringe-fitting procedure described in § 9.5.7.7 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. Note that, if your spectral channels are narrow, the use of the `CHINC` parameter can speed up `FRING`.

9.5.7.1 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 problems specific to SVLBI fringe-fitting. The primary SVLBI fringe-fitting tasks in *AIPS* are `BLING` and `FRING` and are discussed in § 9.5.7.7–§ 9.5.7.9. The tasks `COHER` and `FRPLT`, described in § 9.4.6 and § 9.4.7, 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 boundaries 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.

SVLBI data often contain tracking passes three or four hours in length, for which fringes are mostly (or wholly) not apparent. Typically, the space-ground baselines will have the highest correlation coefficients near perigee, when those baselines are shortest. However, the imperfectly known orbit will cause high fringe rates and short coherence times. Near apogee, the coherence time is longer, and may be limited by the atmosphere above the ground telescopes, but the correlated flux also is much lower. Sometimes, it may be possible to find fringes for only 15 or 20 minutes, but that's better than nothing.

If no fringes are seen anywhere during a tracking pass, a useful trick is to set `APARM(7) = 0.01` to let through the highest SNR for each solution interval, and set `DPARM(5) = 1` to turn off the least squares solution. Then run `FRING` for an entire tracking pass, and use `SNPLT` with `OPTYP = 'DELA'` and `OPTYP = 'RATE'` to look for repeating values (usually easier to see in delay than in rate). Also, make plots with `OPTYP = 'SNR'` to see if slightly higher SNRs are found at a time when there seems to be some consistency in delay values. It may be necessary to try this process with several values of `SOLINT` in order to arrive at a guess for the fringe location. Use `VPLOT` to plot *uv* distance versus time for the space-ground baselines, then search using the `TIMERANG` and `REFANT` that provide the shortest projected baselines. Another possibility is to set `REFANT = ANTNUM('MK')`, since the atmospheric coherence time should be longest at Mauna Kea, and increase `SOLINT` to a fairly large value in hopes something will show up. (Note, however, that a large `SOLINT` with a wide-open search window in delay and rate may require a large-memory computer or great inefficiency due to page faulting.)

If fringes are found somewhere, use `CLCOR` to center the fringes (see § 9.5.4.4), then run `FRING` again with small delay and rate windows (*e.g.*, `DPARM(2) = 200` to 400 and `DPARM(3) = 40` to 80 at 1.6 GHz, or ~ 200 at 5 GHz). Set low SNR thresholds with `APARM(7) = 3.5`, and turn the least-squares solutions back on with DPARM(5)=0. Usually, it's best to turn on the exhaustive antenna search with `APARM(9) = 1`, since only a few space-ground baselines may show fringes. It can be helpful to use `SEARCH` to order the search from shortest to longest space-ground baselines.

It generally doesn't work well to use one tracking station to predict the results of another, because clock initialization offsets are typically relatively large and have unrelated errors, and fringe-rate errors also may be unrelated.

9.5.7.2 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 § 9.5.7.10 for more details.

9.5.7.3 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 § 9.5.7.11. After fringe-fitting, all parallel-hand fringe solutions need to be re-referenced to the same antenna.

9.5.7.4 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 A. Beasley and J. Conway in "VLBI and the VLBA", 1995, (ASP), and VLBA Scientific Memo No. 24 (2000, by J. Wrobel, C. Walker, J. Benson, and A. Beasley) for more details on how to design phase-referencing observations.

In "phase referencing," the phase calibration for your target source is derived from a calibrator, or phase referencing, source observed for that purpose. First, you 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 so determined to the target source. In practice this is done by specifying the target as well as the cal source in the `SOURCES` adverb list whenever an `SN` table containing phase corrections for the cal source is applied using `CLCAL`. See § 9.5.1.2 for a specific example of how to run `CLCAL` to transfer phase corrections determined using a cal source to a target source.

Certain "instrumental" corrections such as those for unmodeled zenith delay may have subtle but significant effects on phase referencing. See § 9.5.7.5 for a discussion.

You can perform a "hybrid" form of phase referencing in some instances. It may be that your target source is too weak for initial 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. Then, after averaging in frequency, the target source data may have adequate signal-to-noise to allow rate corrections to be determined for it by fringe-fitting. 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 determining 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.

9.5.7.5 Correcting for atmospheric delays

VLBI correlators remove some estimate of the non-dispersive atmospheric delay at the elevation and frequency of the observation from the data. These a priori models are usually fairly good, but careful observations can improve upon them. AIPS offers a number of options to deal with this problem. `DELZN` uses multi-band delay (see § 9.5.7.6) in an `SN` table to fit for the zenith tropospheric delay and the clocks as a function of time. It works best if the observations include data on a variety of calibrators well distributed around the sky. `DELZN` applies a correction to a `CL` table or writes a file to disk with zenith atmospheric delays and possibly clock offsets that can be used by `CLCOR, OPCODE='ATMO'` to correct a CL table. The second option is for the situation where the data used by `DELZN` is in a different file from the data that needs to be corrected.

To correct an attached `CL` table, the typical inputs for `DELZN` would be:

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

> SNVER *snin* C_R	to select the SN table containing multi-band delays.
> GAINVER *clin* C_R	to select the CL version to which solutions are to be applied.
> APARM(4) 1 C_R	to create a new CL table.
> APARM(5) 1 C_R	to solve for atmosphere and clocks
> SOURCES 'DA913' , ' ' C_R	to specify the sources to be corrected.
> CALSOUR '0103+337', '0140+412', '0150-334', '0159+418', '0202+319', '0244-297', '0358+210', '0425+174', '0641+392' C_R	to specify the calibrator sources observed at a large variety of elevations
> OPTYPE 'MDEL' C_R	to use multi-band delay
> DOTV -1 C_R	to make PL files
> GO C_R	to run the program.

This will generate a CL table and several PL files that show the data and the fitted model. You are **strongly encouraged** to examine these.

To create an output file rather than correct a CL table use the same inputs as above except:

> APARM(4) 0 C_R	to create no CL table.
> SOURCES ' ' C_R	to correct no sources
> CALSOUR '*' C_R	to use all calibrator sources.
> OUTFILE 'MYVLB:BZ199.DELZN' C_R	output file name

Again, you are advised to examine the resulting plot files which show both the data and the fitted model. The output file can be read in with CLCOR (OPCODE='ATMO'; INFILE='MYVLB:BZ199.DELZN') to correct a CL table.

There is another task designed to deal with the effects of zenith delay in phase-referencing observations. Phases for the target source in phase referencing are corrected by the phases at the calibrator which usually is at a different elevation. Task DFCOR is a special version of CLCOR which applies the 'ATMO' operation to correct the CL table for the difference in elevation between the target source and adjacent calibration sources without applying the full atmospheric delay correction.

If the SN table contains dispersions as well as multi-band delays, DELZN may also be run to fit a zenith angle model to the dispersions (OPTYPE = 'DISP'). Like the 'MDEL' operation, the zenith and time function found by DELZN can be applied to a CL table or written to a text file for later application by CLCOR with OPCODE = 'DISP'.

9.5.7.6 Finding multi-band delays

For astrometric and geodetic experiments and to use DELZN (see § 9.5.7.5), the multi-band delay must be determined. The multi-band delay is the delay caused by errors in the station positions and the difference between the correlator model and reality for clocks and the troposphere. The multi-band delay is best determined over IFs which are widely spaced in the frequency band. After the instrumental phases have been corrected (§ 9.5.4.3–§ 9.5.4.4), the multi-band delay can be determined in one of two ways. For strong sources do a global fringe fit as described in § 9.5.7.7 setting APARM(5)=0, and then run MBDLY on the resulting SN table.

Typical inputs for MBDLY would be:

> TASK 'MBDLY' ; INP C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> INVERS SN *table from FRING*	to select input SN table

> OUTVERS 0 C_R	make new SN table
> BIF 0; EIF 0 C_R	to select all IFs.
> SUBARRAY 0 C_R	to select all subarrays.
> APARM 0 C_R	the defaults are generally O.K.
> GO C_R	to run the program.

This will produce a new SN table with the MBDELAY columns filled in.

For weak sources, use APARM= 2 0 0 1 1 in FRING. This averages each IF and fits a multi-band delay across them. Note that this *does not* solve for single-band delays, unlike the previous method. This will also produce a new SN table with the MBDELAY columns filled in.

Note that MBDLY can fit the single-band delays for a multi-band delay plus a dispersion (phase varies with wavelength rather than frequency) with OPTYPE = 'DISP'. FRING can do a very similar fit, after finding the single-band delays, if you set APARM(10) = 1. Wide bandwidth observations at low frequencies may benefit from this.

9.5.7.7 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 § 9.5.4.4). 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 points are illustrated in Figure 9.1.

You will want to run FRING to correct for these residual rates. You can help these tasks by making sure that the reference pixel in frequency is in the center of the band (Nchan/2+1 is best). Use task CENTR or any of the tasks with the FQCENTER adverb to fix data sets for which this is not true.

FRING and KRING use a global fringe-fitting algorithm described by Schwab and Cotton, 1983, *Astron. J.*, **88**, 688. Unfortunately, these are large and complicated tasks. The procedure VLBAFRNG is available to simplify access to FRING and VLBAKRNG access to KRING. Versions of these procedures for phase-referencing experiments are called VLBAFRGP and VLBAKRGP. For all these procedures, if the SOURCES adverb is set, then CLCAL is run once to apply the results of FRING (or KRING) for each source in SOURCES. For the phase-referencing procedures (VLBAFRGP and VLBAKRGP), any source that is in the SOURCES list that is *not* in the CALSOUR list will be phase referenced to the *first* source in the CALSOUR list. Note that, if every source in the SOURCES list occurs in the CALSOUR list, VLBAFRNG and VLBAKRNG will run identically to VLBAFRGP and VLBAKRGP, respectively. If the SOURCES list is empty, VLBAFRNG and VLBAKRNG will run CLCAL once over all sources, while VLBAFRGP and VLBAKRGP will run CLCAL once referencing all the sources to the first source in CALSOUR. These procedures will produce new (highest numbered) SN and CL tables.

Sample inputs for procedure VLBAKRNG are:

> RUN VLBAUTIL C_R	to acquire the procedures; this should be done only once since they will be remembered.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> TIMERANG 0	to include all times.

> BCHAN 0 ; ECHAN 0 C~R~	to use all frequency channels.
> GAINUSE *CLin* C~R~	to use the `CL` table with all the calibration up to this point.
> REFANT *n* C~R~	to specify an antenna that is present most of the time as the reference antenna.
> SUBARRAY 0 C~R~	to use all subarrays.
> SEARCH 0 C~R~	to try all antennas as a reference antenna if fringes cannot be found using `REFANT`. *This is different from* `FRING`*; in* `FRING` *this must be set to try other reference antennas*.
> OPCODE ' ' C~R~	to leave all solutions in the output `SN` table.
> CPARM 0 C~R~	to use defaults for `KRING` steering parameters; this is okay for strong sources.
> CPARM(1) *x* C~R~	to specify the minimum integration time in seconds.
> CPARM(8) 1 C~R~	to avoid re-referencing solutions; do this *only* for polarization experiments.
> CALSOUR '*src1*', '*src2*' C~R~	to specify the sources to fringe fit using `KRING`.
> SOURCES '*src1*', '*src2*' C~R~	to have `CLCAL` run for each source using the interpolation method given below.
> INTERPOL 'AMBG' C~R~	to use the "`AMBG`" interpolation method (linear phase connection using rates to resolve phase ambiguities).
> BADDISK 0 C~R~	to use all disks for scratch files.
> VLBAKRNG C~R~	to run the procedure.

Procedure `VLBAKRGP` sets the same adverbs as `VLBAKRNG` *except*

> SOURCES '*src1*', '*src2*', '*src3*' C~R~	to have `CLCAL` run for each source using the interpolation method given by `INTERPOL`. Any source here that is not in the `CALSOUR` list will be phase referenced to the first source in the `CALSOUR` list. In this example, *src3* is phase referenced to *src1*.
> VLBAKRGP C~R~	to run the procedure.

`VLBAFRNG` and `VLBAFRGP` are identical except there is no `OPCODE` (it is equivalent to `DPARM(8)`) and `DPARM(4)` and `DPARM(7)` in `FRING` are the same as `CPARM(1)` and `CPARM(8)` in `KRING`, respectively. Also note the different use of `SEARCH` in `FRING` and `KRING`.

Suitable inputs for the fringe-fitting task `FRING` are, in detail:

> 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 1 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 ; CHINC 1 C~R~	to use all spectral channels within each IF channel.
> FLAGVER 0 C~R~	to apply the most recent flag table.
> SMODEL 0 ; CLR2NAME C~R~	to use a point-source at the origin model for the sources, rather than a Clean-component model.

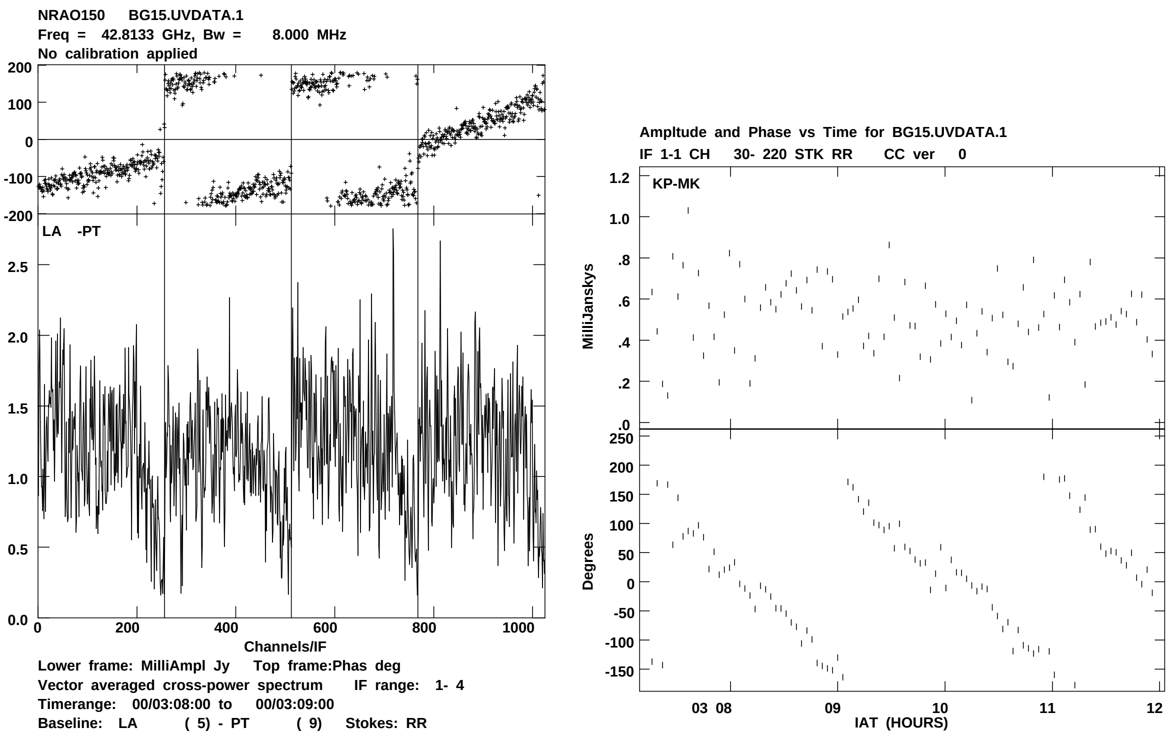

Figure 9.1: *left:* A `POSSM` plot of the 43-GHz spectrum of the quasar NRAO150 on the Los Alamos to Kitt Peak baseline. The plot shows that the observation was performed with 4 IFs, with 256 spectral channels within each IF. The upper frame shows the phase variation with frequency; within each IF, the 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 § 9.5.7). *right:* A `VPLOT` plot of the uncalibrated 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 § 9.5.7), the data cannot be averaged in time.

> 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.
> SEARCH *ref2*, *ref3*, ... C_R	to specify the order over which antennas are searched for fringes when the exhaustive search is requested. If `APARM(9)` is set, all antennas will be searched for fringes with `SEARCH` controlling the order of the search. Note, `REFANT` should be `SEARCH(1)`. The use of `APARM(9)` and `SEARCH` makes `FRING` much more likely to find fringes to weak antennas and is highly recommended.
> 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 use the frequencies individually within each IF; `APARM(4) > 0` causes the frequencies within each IF to be averaged 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 (if no `AN` table).

> APARM(9) 1 C_R — to enable the exhaustive search mode.

> APARM(5) 0 C_R — to do least-squares fits in each IF. `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 $2.5 >$ `APARM(5)` > 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)=0` for multi-IF data. Primarily for the VLA, a fourth option, `APARM(5) =` M has been added, with $M > 2$. It divides the IFs into $M - 1$ groups and solves for one delay per group.

> REFANT 5 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 above). 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.

> 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. Unless 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.

> ANTENNAS 0; DOFIT 0 C_R — to allow all antennas to be fit. 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 Lecture 19 in *Synthesis Imaging in Radio Astronomy*, edited by R. Perley, F. Schwab, and A. Bridle, 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. `DPARM(1)>1` works even when the integration times are not equal.

> 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 permit a lower SNR threshold to be set, but can also result in lost data due to failed fringe searches. For the VLBA, a `DPARM(2) = 1000` window is usually adequate.

> DPARM(3) 0 C_R — 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 permit a lower SNR threshold to be set, but can also result in lost data due to failed fringe searches. For the VLBA, a `DPARM(3) = 200` window is usually adequate.

> DPARM(4) 2 C_R — to specify the correlator integration time in seconds; use `GETITIME` or `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 C_R — to do both the coarse and the least squares solutions; set to 1 if you require only FFT solutions.

> DPARM(6) 1 C_R — 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 C_R — 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 C_R — 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.*

> DPARM(9) 0 C_R — to allow fitting for rate. `DPARM(9)` $>$ 0, causes the task to suppress rate fitting entirely rather than zeroing it after the fact as in `DPARM(8)`.

> INP C_R	to check the inputs.
> GO C_R	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. Especially for weak sources, the least-square solution may wander outside narrow constraints set by DPARM(2) and DPARM(3).

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, 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 or LISTR with OPTYPE='SCAN', or LISUN.

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 arcseconds). If you are using multi-baseline searching (*i.e.*, DPARM(1) > 1), 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. When using a model, convergence may be improved by weighting the data by $1/\sigma$ rather than $1/\sigma^2$; set WEIGHTIT = 1.

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 $\geq$ 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 too short 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' ; DOBLANK 1 C_R	to plot rate solutions including failed ones, 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 the solutions for OPTYP 'RATE' and 'DELA' (single-band delay) as well as the

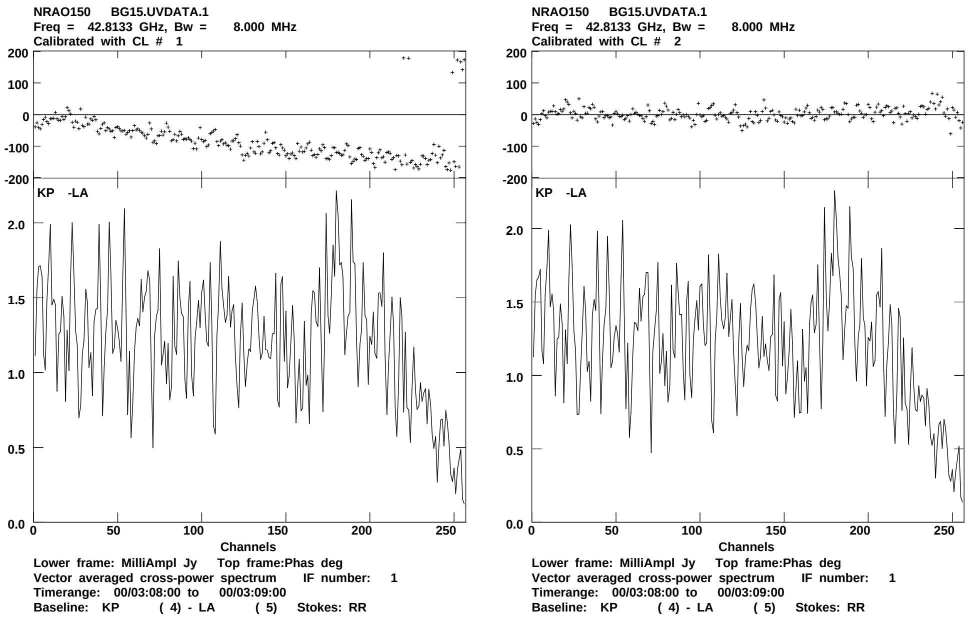

Figure 9.2: *left:* A POSSM plot of the uncalibrated spectrum of NRAO150 at 43 GHz on the baseline Kitt Peak to Los Alamos. The plot shows the spectrum for a single IF to show the effects of the residual delay error more clearly. The phase slope as a function of frequency is clear evidence for a small 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.

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 insists on creating), you can run FRING multiple times specifying time ranges and explicitly setting SNVER to the same SN table. Judicious use of CHINC when your spectral channels are narrow (< 0.5 MHz) may also help with disk space and compute time.

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 § 9.5.1.1.

Once a final CL table is generated, its effect on the data can be viewed using tasks VPLOT and POSSM by setting DOCAL=1 and GAINUSE to the version number of the final CL table; see § 9.4. 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 9.2.

9.5.7.8 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 § 9.5.7.7 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. Note that FRING may be used to do baseline-based fringe-fitting by running it many times, each time specifying only 2 antennas. BLING has not been actively maintained or used, and so may be less reliable. 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 may be important in fringe-fitting Space VLBI data.

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 converted into antenna-based corrections using the separate task BLAPP and then applied to the data.

Acceleration may be 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. Therefore, you may wish to turn off the acceleration search (DPARM(7) to DPARM(9)) unless you need it for space VLBI.

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	to specify the calibrator source.
> STOKES 'LL' C_R	to select the Stokes.
> TIMERANG 0, 10, 5, 0, 0, 11, 0, 0 C_R	to limit the time range.
> ANTENNAS 3 ; BASELINE 0 C_R	to select all baselines to antenna 3.
> SUBARRAY 1 C_R	to use subarray 1.
> FREQID 1 ; BIF 1 ; EIF 0	to use frequency ID 1 with all IFs.
> BCHAN 1 ; ECHAN 0 C_R	to use all spectral channels.
> DOCAL 1 ; GAINUSE *clin* C_R	to apply amplitude calibration before fringe fitting.
> CLR2N ; NMAPS 0 C_R	to do no model division.
> SOLINT 0.5 C_R	to use a 30-second fringe solution interval.
> INFILE ' ' C_R	to use the adverbs rather than an external control file.
> APARM(1) 2 ; APARM(2) 0 C_R	to set the integration time; no model division.
> APARM(3) 0 C_R	to do no stacking of baselines.
> APARM(4) 0 ; APARM(5) 0 C_R	to set minimum acceptable SNR to 5 and accept coherence of 20%
> DPARM 0 C_R	to use the default fringe windows and no acceleration search.
> DOUVCOMP 1 C_R	to use compressed scratch files.
> BADDISK 0	to use all disks for scratch files.
> 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 specified using APARM(4) and APARM(5) are important parameters. The use of compressed scratch files is recommended and is not believed to have a significant impact on precision. Note, if model division is required, APARM(2) should be set and the source should be entered explicitly.

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	to specify the BS table version number.
> DOCRT -1 C_R	to send output to an external file.
> OUTPRINT 'FITS:BSPRT.LIS' C_R	to define the 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 changes in FFT interpolation the errors may be an overestimate. 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	to specify the input BS table version number.
> SOURCES ' ' ; STOKES 'LL' C_R	to do all sources and a specific Stokes.
> FREQID 1 C_R	to select frequency ID 1.

> TIMERANG 0 ; ANTENNAS 0 C_R	to do all times and antennas.
> SUBARRAY 1 ; REFANT 3 C_R	to do subarray 1 with reference antenna 3.
> ANTWT 0 C_R	to use equal antenna weights.
> OPCODE 'SOLV' C_R	to solve for a SN table.
> GAINVER clin C_R	to specify the input CL table which defines the times for which solutions are desired.
> BADDISK 0 C_R	to use all disks for scratch files.
> 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 output CL table also needs to be specified using GAINUSE. If an SN table is generated, it can be smoothed or clipped using task SNSMO and applied using CLCAL as described in § 9.5.1.1.

9.5.7.9 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. It was shown in VLBA Scientific Memo No. 13 (1996, "Global ground VLBI network as a tied array for space VLBI", by L. Kogan) that the method of phasing a group of ground-based antennas and the method using global fringe fitting with baseline stacking give the same minimum detectable flux density. Therefore, baseline stacking with DPARM(1)=3 in FRING should yield the best possible sensitivity. There are other options which also may be explored. 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. The exhaustive baseline search mode, used by default in BLING and KRING and activated in FRING by setting APARM(9)=1, allows more baselines to the spacecraft to be searched.

9.5.7.10 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.

If computer memory is limited, 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 may force FRING to require too short a solution interval. Judicious use of CHINC in FRING reduces this problem for spectral channels < 0.5 MHz in width. (FRING now allocates memory dynamically and may be able to handle large cases so long as the computer is adequately equipped with real and swap memory.) This limit can be overcome by 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 1 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 5 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 acceptable SNR. *Warning:* solutions with lower SNR will be flagged as bad which will ultimately flag the affected data.
> 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 250 kHz-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 calibrators had been extracted to a separate data set, use TACOP to copy the resultant SN table back to the line data set. SNCOP may be used when the number of IFs in the two data sets is different. This is more likely to arise when you use a strong line signal in one IF to solve for the delays of a larger data set.

Run CLCAL to apply the delay and fringe-rate solutions to all sources, as is described in § 9.5.1.1, 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 made the phase coherence worse. 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 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 C_R	to review the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> CALSOUR 'OH127.8' , ' ' C_R	to select the spectral-line source.
> DOCALIB 1 C_R	to apply the previous amplitude and delay/rate calibration.

> GAINUSE *clin*	to specify which CL table to use.
> FLAGVER 0 C_R	to apply the most recent flag table.
> SNVER 0 C_R	to write a new solution table.
> BCHAN 72 ; ECHAN 72 C_R	to select the strongest and/or simplest spectral channel as a reference.
> REFANT 5 C_R	to try to use the same reference antenna as in the previous run of FRING.
> 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 useful, but limited printout.
> APARM(7) 9 C_R	to avoid false detections by setting the minimum acceptable SNR. *Warning:* solutions with lower SNR will be flagged as bad which will ultimately flag the affected data.
> DPARM(1) 1 C_R	to use one-baseline combination in initial, coarse fringe search (FFT).
> DPARM(2) = -1 C_R	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 C_R	to search for fringes over the full Nyquist fringe-rate window since we don't know where the fringes are.
> DPARM(4)= 0 C_R	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 C_R	to do the least-squares solution.
> GO C_R	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.

9.5.7.11 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 a task named RLDLY is 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 and LPCAL. SPCAL was designed for polarization calibration of spectral-line data sets, but PCAL is now more capable of doing a channel-dependent polarization solution. 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 § 4.3.16) using task RLDIF 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' or task RLCOR (see

§ 4.3.16). See also VLBA Scientific Memo No. 26, "Polarization Angle Calibration Using the VLA Monitoring Program," by G. Taylor and S. Myers, October, 2000.

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.

9.5.7.12 R-L delay calibration

RLDLY is a task which replaces RUN file procedures VLBACPOL and CROSSPOL. It determines R-L delay differences and produces an SN table which corrects for these effects (after CLCAL). 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 should be averaged but RL or LR fringes (or both) must be detectable on each baseline to the reference antenna. This task should leave a single R-L phase difference that must be determined from a calibrator of known polarization angle.

> DEFAULT RLDLY ; INP C_R	to set the task name and examine the inputs.
> INDISK *n* ; GETN *ctn* C_R	to specify the input file.
> OUTDI 1 C_R	to use disk 1 for temporary files.
> FLAGVER 0 C_R	to use the highest numbered flag table.
> GAINUSE *CLin* C_R	to use the CL table with all calibration up to this point; *no default*.
> SUBARRAY 0 C_R	to do all subarrays.
> ANTENNAS 0 C_R	to use all antennas.
> REFANT $A_{\rm ref}$ C_R	to select the reference antenna; any antenna may be used but all baselines to it should have RL and LR fringes. REFANT 0 will loop over all possible (not necessarily good) reference antennas, averaging the result.
> APARM 6, 1, 10 C_R	to use only solutions with signal-to-noise ratio above 6, to not write a CL table even if there is only one calibration scan, and to omit all solutions with rms $<$ 10 from the average over antennas.
> CALSOUR '*cal1*' , ' ' C_R	to specify the calibrator source to use.
> TIMERANG *d1 h1 m1 s1 d2 h2 m2 s2* C_R	to specify a time range with high SNR for RL and LR.
> INTTIME 0 C_R	to specify the minimum integration time in seconds; 0 causes it to be found from the data.
> GO C_R	to run the task.

RLDLY should be done after parallel-hand instrumental delays are removed (VLBAPCOR). It may be done before or, with a slight preference, after fringe fitting (VLBAFRNG, VLBAKRNG, VLBAFRGP, or VLBAKRGP). The corrections should be checked with VLBACRPL, by setting STOKES to 'RL' and/or 'LR'. 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).

9.5.7.13 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 1 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 Clean images as models for the I, Q, and U polarizations.
> 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 images supplied.
> SOLINT 6 C_R	to set the solution interval in minutes. It should not exceed the coherence time.
> SOLTYPE 'RAPR' C_R	to specify the type of feed model.
> PRTLEV 0 C_R	to print minimal information
> BPARM(1) 0 C_R	to 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	to 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 PCAL 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.

9.5.8 Complex Bandpass

For spectral line experiments and continuum observations where a high dynamic range is required, it is be a good idea to do a complex bandpass at this point. In § 9.5.4.5 a scalar bandpass was done; *i.e.*, only the amplitude was calibrated but not the phase. Now that the phases have been calibrated in time (by fringe-fitting), a complex bandpass may be solved for. This will take out any dependence of the phase on frequency. To do this, run BPASS, again applying the CL table that includes all the calibration. With the inputs we recommend here, the phases *must be stable in time over the entire bandpass calibrator scan(s).* Check this with VPLOT before proceeding. Then

> 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.
> DOCALIB 1 C_R	to apply calibration.
> GAINUSE *CLin* C_R	to indicate the CL table with all calibration up to this point.
> BPVER -1 C_R	do not apply previous bandpass table
> SOLINT 0 C_R	to average data over whole scans before determining the bandpass.
> BPASSPRM 0 C_R	to do a complex bandpass and set rest to 0.
> BPASSPRM(5) 1 C_R	to not divide by "channel 0."
> BPASSPRM(9) 1 C_R	to interpolate over flagged channels.
> BPASSPRM(10) 6 C_R	to normalize the amplitude and phase of the bandpass solutions, using *power* rather than voltage for amplitudes.
> ICHANSEL $chan_{beg}$, $chan_{end}$, 1, IF_{num}, ... C_R	to set channels for entire bandwidth for normalization, if this left 0 then the inner 75% is used and can cause up to 15% error in the amplitude.
> GO C_R	to run the program.

As recommended in § 9.5.4.5, you should look at the bandpass with POSSM and BPEDT. If necessary, use BPEDT to flag channels in your bandpass table or calibration data. In the latter case, re-make the BP table. This is the bandpass that should be applied when the data is calibrated and averaged (*e.g.*, with SPLIT).

Note, if your bandpass calibrator is not stable or strong enough during the observation (your bandpass calibrator data should have a better S/N than the data you're trying to correct with the bandpass), you could consider using the phase reference source (if there is one) and use SOLINT = -1 (include all scans to make one bandpass solution). If the bandpass calibrator is strong enough, but the average phase varies through the scan, then divide each record by "channel 0" by setting BPASSPRM(5) = -1 to adjust phase only. You should select a range of channels that have similar phases to be averaged as channel 0 using adverb ICHANSEL.

9.5.9 Baseline-based errors

Baseline-based non-closing phase and amplitude errors can limit the dynamic range of the final images. 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 1 ; 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.

Set ICHANSEL to select only the most desirable spectral channels to average. Task BLPLT may be used to plot the BL table. Task BLCHN is available to compute baseline-based errors on a channel-by-channel basis. It writes a BD table which can be plotted by POSSM and BPLOT, but the calibration itself is not applied by the standard calibration routines. BLCHN writes a calibrated output data set and BDAPL can apply the BD table to another data set. In 31DEC23, procedure BLPOLCAL uses task BLPCL to carry out a similar calibration for both parallel-hand and cross-hand polarizations.

9.6 After initial calibration

9.6.1 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 1 ; 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 *blin* if present.
> APARM(1) 1 ; NCHAV 0 C_R	to have all spectral channels within each IF averaged: read the help file closely, other useful averaging options are available.
> 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) = 3 in SPLIT.

(The task AVSPC no longer averages IFs although it is useful in averaging spectral channels in calibrated data sets.)

The task SPLAT can be used instead of SPLIT to do time averaging and different options in spectral averaging. SPLAT can be used also to assemble the selected sources into a multi-source file after applying the specified calibration and averaging. This option allows the user to continue calibration on a smaller data set.

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.

9.6.2 Time averaging

It is now convenient to average the data in time using UVAVG both to reduce the bulk of the data and to increase the signal-to-noise for subsequent iterations of the self-calibration/imaging 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. The ionosphere can cause similar problems at lower frequencies. Self-calibration should remove such phase errors.

For data at frequencies below 15 GHz (and $\approx$ 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.
> 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 § 9.7).
> 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.
> 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. There are several options controlling the averaging interval selection and reported times.
> GO C_R	to run UVAVG.

If SPLAT was used to assemble the selected sources into a multi-source file (while applying the preliminary calibration), CALIB will write an SN table which must be converted to a CL table by CLCAL. This new CL table can be applied by SPLAT with time averaging.

9.6.3 Verifying calibration

Before proceeding to image your data, it's worth checking that the calibration performed in § 9.5 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 SPFLG, EDITR, WIPER, 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.

In 31DEC20, two tasks allow you to examine selected portions of a *uv* data set statistically. UVRMS averages the selected data in SOLINT time intervals and then can plot the average values as a function of time and as a histogram. A variety of statistical parameters are computed and printed. SPRMS determines and plots the mean and standard deviation of the selected data as a function of spectral channel. Note that both tasks will combine data from multiple sources, baselines, etc. if so instructed. In 31DEC21 RIRMS computes the mean and rms of the real and imaginary parts of the visibilities. It prints these in a baseline matrix and can plot them by baseline as functions of time and as histograms. These tasls have a full set of calibration and data selection adverbs, so they may be run before you run SPLIT.

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 2,500,000 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 or SPLAT (with spectral averaging)

> 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 $45\,\mathrm{deg}$.
> 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 SRCNAME, DOCAL, GAINUSE etc. must then be set. It can handle up to 20 sources to be fit, using a channel-dependent input text file and can write the results in a compact text file form.

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 are 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', and CLCORPRM(1) set to the amplitude scale factor found in UVFIT; note that these are *voltage* gains. All higher values in the array CLCORPRM should be zero. The effect of the altered calibration can be viewed using UVPLT with DOCAL = 1. If it is satisfactory, SPLIT can be re-applied to the data. We recommend using CLCOR to perform such amplitude corrections. It now produces a new CL table each time it is used unless you specify both GAINVER and GAINUSE as having the same, non-zero version number.

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 (*not recommended!*, 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 § 9.5. 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.

9.7 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, unless phase-referencing is used, absolute phases of VLBI data are un-calibrated. 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 tasks `SCMAP` or `SCIMG` which cycle between imaging and self-calibration with an option to edit the *UV* data. If you wish to edit the Clean components or make plots, you will need to do the manual Clean/self-cal process with `IMAGR` and `CALIB`.

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). 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*, **1995**, edited by A. Zensus, P.J. Diamond, and P.J. Napier which is available on the World-Wide Web (www.cv.nrao.edu/vlbabook). 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. They do not offer several experimental algorithms found in `IMAGR` including Clean component filtering, the SDI Clean algorithm, and multi-resolution Cleaning.

9.7.1 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 extrapolated 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 ($\leq$ 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 in the output file may be flagged. This is not too important if you use the original input *uv* file as the input to `CALIB` in all iterations. You can also set `APARM(9) = 1` to leave data affected by failed solutions uncalibrated.
2. On subsequent iterations, use the Clean image as produced by `IMAGR` as your input model. 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 in `IMAGR` (`IMAGRPRM(8)` and `IMAGRPRM(9)`) or afterward 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 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 (output by CALIB) which was used to produce that Clean model. It is advantageous to use the original data set at least until you turn on amplitude calibration. At that point, you should stick with the file produced from the original data with the best phase-only solution. Amongst other things, this can prevent the telescope amplitudes from "wandering" (see below).

4. When the model contains extended structure, there may be problems with convergence when weighting the data "correctly" (*i.e.*, by $1/\sigma^2$.) The WEIGHTIT adverb allows you to use less extreme weights, such as $1/\sigma$ or $1/\sqrt{\sigma}$. If your array contains antennas that have a wide range of sensitivities, *e.g.*, the VLBA plus the phased-VLA and/or the Effelsberg 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 baseline weights 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, CAPLT, 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. Set WTUV > 0.

7. When you are finally ready to solve for amplitude corrections, you should first apply all previous phase calibration including the final phase-only self-calibration solution. Then run CALIB setting SOLMOD 'A+P', initially setting the solution interval (SOLINT) to a longer time than used for phase-only solutions (*e.g.*, 3 times). 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 to improve its usefulness for SVLBI data sets. 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 § 9.5.7.7).

9.7.2 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 Effelsberg, the phased-VLA, and/or HALCA), then it may be advantageous to alter 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 the most 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 and SCIMG also support 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. Note the BOXFILE and OBOXFILE options which allow you to retain interactively set Clean boxes for use in the next self-cal iteration.

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.

9.7.3 Non-conventional methods of imaging

Since fringe rate is a function of source position, we can use measurements of the fringe rate to estimate the positions of sources. Fringe rate imaging is widely used with maser sources which are usually widely separated point objects. The angular resolution of fringe rate images is not as good as conventional aperture synthesis imaging, but it can be used for an initial determination of the location of emitting clusters. This will help in setting field shifts and Clean windows for use by IMAGR. Having measured the fringe rate $FR(i)$ for each sample in (baseline-time) and computed the derivative of fringe rate ($UDOT(i)$) wrt to right ascension (X) and the derivative ($VDOT(i)$) wrt declination (Y), we obtain a set of equations of straight lines

$$FR(i) = UDOT(i) * X + VDOT(i) * Y$$

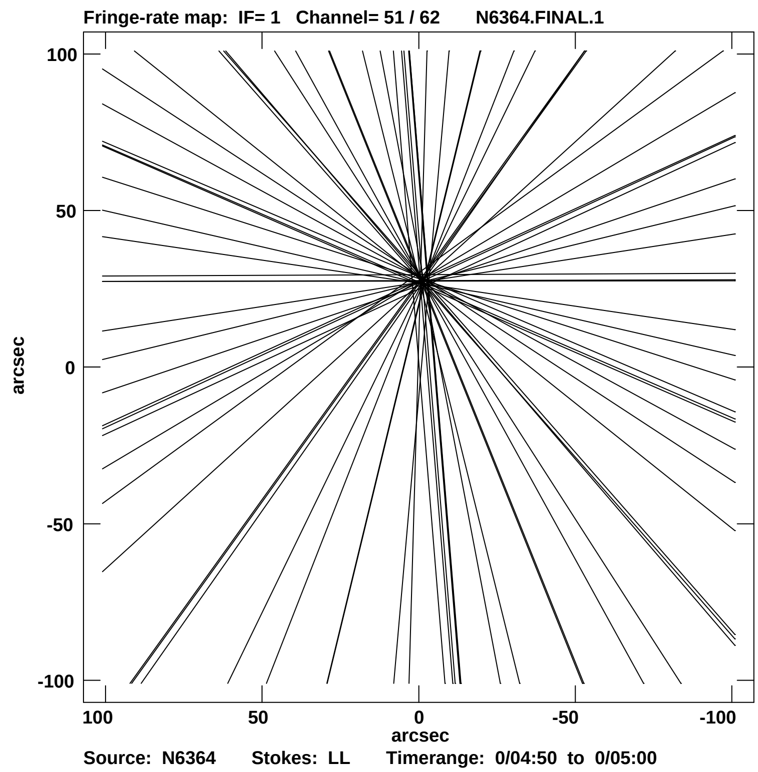

Figure 9.3: An example of a preliminary determination of the source location using fringe-rate analysis. The position of the source in channel 51 is shifted relative to the reference channel (62) by ≈ 25 asec to the north. Conventional imaging can be carried out near this position.

describing the loci of constant fringe rate for each observation. The positions of the source component(s) can be determined by finding the places of highest density of crossing lines. (Details can be read in Giuffrida,T.S, 1977 Ph.D.thesis MIT and in Walker R.C., 1981, *Astr. J.*, **86**, 9, 1323.)

This algorithm is implemented in *AIPS* as the task `FRMAP`. The task plots the straight lines on the TV or prepares a plot file. Then it determines the positions of higher density of crossing lines. The coordinates in RA and DEC of the components found are written in an output file. The fringe rate method is especially attractive in SVLBI because of the better angular resolution due to faster movement of the orbiting antenna. Figure 9.3 shows an example of a fringe rate image.

`FRMAP` can be used for more accurate definition of the source coordinates that is required by the correlator. In this case the correlator carries out two passes. After the first one (short) `FRMAP` is used to find a more accurate position of the source. The new coordinates are then used in the second pass covering the whole experiment.

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. See AIPS Memos 114 and 117, reachable via https://www.aips.nrao.edu, for complete documentation of the table and data formats used by AIPS.

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.

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 Calibration table. Version 1 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, delay, and amplitude 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, sub-array 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 orbiting 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` and other tasks.

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 temperature 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 recipes

9.9.1 Bananes rôties

1. Preheat oven to 375° F.
2. Place 6 (peeled) **bananas** in a baking dish.
3. Sprinkle bananas with juice of 1/2 **lemon**.
4. Pour 2 tablespoons melted **butter** and 2 tablespoons **dark rum** over the bananas. Sprinkle with 2 tablespoons **brown sugar**.
5. Place in oven for 10 minutes.
6. Pour on 2 more tablespoons **melted butter** and 2 more tablespoons **dark rum** and bake for 5 minutes more.
7. Serve at once, spooning some sauce over each banana.

9.9.2 Banana curried chicken

1. Fry 2 chopped **onions** in 50 ml **cooking oil** until light brown.
2. Add 1/4 cup **cake flour** and mix well. Add 1 (cup?) **chicken stock** gradually while stirring.
3. Add 1 cup **raisins**, 1 teaspoon **salt**, 2 pounds cooked, boned **chicken**, 5 sliced **bananas**, 2 grated **apples**, 2 tablespoons grated **lemon rind**, 1 tablespoon **sugar**, 1 1/2 tablespoons **curry powder**, 1 **bay leaf**, 4 **peppercorns**.
4. Cover saucepan and simmer for 20 minutes.
5. Remove bay leaf. Add 1 cup **cream** and heat just before serving.
6. Serve on a bed of rice. Decorate with pineapples if preferred.

Thanks to Turbana Corporation (`www.turbana.com`).