AIPS Memo 89

Baseline-Oriented Fringe Searches in AIPS

Chris Flatters

November 17, 1994

Abstract

The AIPS tasks for performing global fringe searches using the Schwab-Cotton technique and applying their results have been supplemented by tasks that search for fringes on individual baselines and globalize the results in a separate step. In this memorandum I shall explain why baseline-oriented fringe-searches may be more desirable than global searches in some circumstances and describe the algorithms used by the AIPS programs. I shall also outline the differences between the current versions of these programs and earlier versions and number of enhancements which should be added to these tasks during 1995.

1 Introduction

The Schwab-Cotton technique for global fringe searches [1] used in **FRING** has two well-known advantages over straightforward baseline-oriented fringe searches.

- All baselines contribute to the solution; even those which are too insensitive for fringes to be detected without reference to other baselines. This reduces the minimum flux density required to detect fringes over a given time interval.
- Delays and rates are associated with single antennæ. This is a realistic physical model since delays and rates arise from the use of different clocks at different antennæ. It allows delay and rate corrections to be inferred for the weaker cross-polarized visibilities from the stronger parallel-hand visibilities in full-polarization experiments and allows the mapping of sources that are not small compared to the delay resolution on each baseline by the use of iterative procedures.

While the first of these can only be obtained by a fully global search, the second can also be obtained by "globalizing" the results of a baseline-oriented fringe-search by finding the antenna-based delays and rates that best reproduce the baseline-oriented results.

Such *post-detection globalization* techniques trade speed for memory reduced requirements. Since the Schwab-Cotton technique requires access to data from all baselines over a given time range its memory requirements scale as N^2 where N is the number of antennæ; single-baseline searches only need the data for for one baseline at once so that their memory requirements are independent of the number of antennæ in the array and of the order of N^2 less than that for the Schwab-Cotton method. On the other hand post-detection globalization requires that delays and rates be searched for each baseline so that number of parameters that must be found in its initial stages scales as N^2 whereas the number of parameters to be found in the Schwab-Cotton method scales as N. Since the amount of CPU time required increases with the number of parameters to be found in both cases, baseline-oriented fringe searches will usually be slower than Schwab-Cotton global fringe searches. Memory limitations can be critical for large datasets, however, so post-detection globalization may be the preferred option for some experiments.

For this reason, the *AIPS* global fringe-fitting task, `FRING`, has been supplemented by a pair of tasks that implement base-line oriented fringe searches (`BLING`) followed by post-detection globalization (`BLAPP`). Aside from reducing the amount of memory required for the fringe search these programs have a number of features that are not available in `FRING`.

- Search windows can be set differently for different baselines and may vary with time. Windows need not be centered on zero delay and rate.
- `BLING` can search for a fringe acceleration (the time derivative of the fringe rate)[1]. This is required for observations involving orbiting antennæ where the fringe rates may vary significantly within a solution interval as a result of spacecraft motion.
- `BLING` provides an estimate of the coherence obtained on each baseline over the solution interval.

Other features will be added during the course of 1995 and are summarized in an appendix.

2 Using `BLING` and `BLAPP`

In general `FRING` will give better results than `BLING` and `BLAPP` for astronomical data and will run much faster[2]. You should, therefore, only use `BLING` and `BLAPP` in cases where `FRING` can not handle the solution intervals that you need or where you need the special features available only in `BLING` (such as acceleration searches or flexible windows).

`BLING` may be run in one of 4 modes which may be selected using the `OPCODE` adverb.

`INDE` In this mode each IF is treated as a separate entity and has its own rate and delay. This is the most suitable mode for multi-IF VLBI data without phase calibration data. Since this makes the fewest assumptions about the calibration of the data, it is the default mode. It is equivalent to setting `APARM(5)` to 0 in `FRING`.

`VLBA` This assumes that one delay and rate applies to all IFs and that there is no systematic variation between IFs that needs to be fitted by a multi-band delay. This is the most suitable mode for multi-IF VLBI data where phases have been referred to a fixed point in the system using two of more phase calibration tones per IF. This should be the normal mode for VLBA data once the phase calibration data is made available in *AIPS*. It is equivalent to setting `APARM(5)` to 1 in `FRING` and is identical to the `'INDE'` mode for single-IF data.

`MK3` This assumes that all IFs have a single delay and rate and that a multi-band delay can account for systematic phase differences between IFs. This is the most suitable mode for multi-IF VLBI data where phases have been referred to a fixed point in the system using only one phase calibration tone per IF (as in Mk3 VLBI). It is equivalent to setting `APARM(5)` to 2 in `FRING` and may not be used with single IF data.

`RATE` This assumes that all delays are zero and fits for rate only. This mode may only be used if a single channel and IF have been selected and is equivalent to selecting a single channel and IF in `FRING`. It is normally used to determine rates for spectral line experiments using continuum calibrators.

Different polarizations are treated separately in all 4 modes.

The basic mode of operation is the same in each mode. The data are divided into solution intervals with the maximum solution interval being determined by the `SOLINT` adverb or the scan duration in the index table. A windowed FFT search is carried out for each solution interval and the results of this are refined by

[1] Other *AIPS* programs do not yet make use of this information, however

[2] On a nine-antenna VLBA polarization data set with 4 IFs and 16 channels `FRING` is approximately 3 times faster than `BLING`.

minimizing the χ^2 statistic[3]. The delay and rate windows for the FFT search are set using the `DPARM` array and may be overridden for individual baselines and time ranges by specifications in an auxiliary text file (see the explain file for details). Tighter delay and rate windows will result in slightly better performance than wider windows. Note that the calculation of the χ^2 statistic requires that the data be correctly weighted and that it is therefore necessary to apply *a-priori* amplitude calibrations before running `BLING`; if the amplitudes are not calibrated `BLING` will almost certainly fail to find solutions.

Acceleration is treated separately in the FFT search phase. The search is carried out for different values of the acceleration between `DPARM(7)` and `DPARM(8)` microseconds per second with an increment of `DPARM(9)` microseconds per second and the acceleration for which the peak amplitude in delay-rate space is greatest is used as the starting point in the optimization step. If `DPARM(9)` is negative or zero then the acceleration search is not carried out. You should normally set DPARM(9) to zero since searching acceleration can degrade the quality of the solutions for the other fringe parameters. You may enable acceleration searches for specific baselines using the acceleration fields in the auxiliary text file. This should only be necessary for baselines with orbital antennæ. The acceleration fields in the text file are used in the same way as `DPARM(7)` through `DPARM(9)`.

`BLING` is a slow program[4] and will usually take several hours to run so it is usually worthwhile to experiment with different solution intervals and coherence thresholds on a small amount of data and then run the full `BLING` overnight. You should experiment with at least one weak source and one strong source and at least one sensitive and one insensitive baseline. In the test runs you can set the coherence threshold to a very low value (0.1, say) and use a value that is just slightly below the coherence obtained for the weakest plausible fringes for the full run.

Short solution intervals usually give better coherence while long intervals usually give better SNR ratios. Five minutes is often a good compromise setting. If necessary, you can run `BLING` separately with short solution intervals for strong sources and long solution intervals for strong sources and combine the resulting `BS` tables afterwards using `TAMRG`. After you have found an optimum setting for `SOLINT` you should re-index your data set using `INDXR` and setting a maximum scan length approximately equal to `SOLINT`. If you do not re-index your data then `BLING` will need to repeatedly read data from earlier solution intervals in a scan to find the start of the later solution intervals. This can slow `BLING` down by large factor; depending on the relative lengths of the scans and the solution intervals, `BLING` may be more than an order of magnitude slower than it would be if the data were re-indexed.

After running `BLING`, you should then run `BLAPP` to reduce the baseline-based terms to antenna-based quantities. `BLAPP` will run very quickly and can be used to generate an `SN` table or to directly apply the antenna-based quantities to a `CL` table.

The next two sections will describe the workings of `BLING` and `BLAPP` in more detail.

3 A Closer Look at `BLING`

`BLING` begins by generating a list of required baselines from the `ANTENNAS` and `BASELINE` adverbs and by reading the window definitions from the auxiliary input file (if any). The default window set using the `DPARM` array is appended to the end of the window list. `BLING` then processes each baseline in turn.

For each baseline, `BLING` looks at each of the scans defined by the `NX` table that fall in or partially in the requested time range and that meet the source selection criteria. Each scan is progressively divided into even intervals until the length of the intervals is no longer than `SOLINT` minutes. `BLING` then checks that the data for a single solution interval will fit into its internal buffers; if not, `BLING` will continue dividing the scan until the data will fit.

---

[3] This is explained in more detail in the following sections.

[4] BLING takes approximately 2.5 hours to process 10 minutes of VLBA data (9 antennae, both parallel-hand polarizations, 4 IFs and 16 channels) using OPCODE 'INDE' on a SPARCstation IPX. The time taken will be roughly proportional to the number of polarizations and IFs and to the length of the experiment and roughly inversely proportional to the AIPSmark rating of the machine you are running it on (a SPARCstation IPX rates about 1.0 AIPSmarks). The solution interval has only a small effect on the run-time.

The data for each solution interval will then be read into the internal buffers using `UVGET` and a fringe search will be performed. `UVGET` first consults the index table to find the index entry immediately preceding the starting time for the solution interval and positions the file pointer at this visibility. It then begins to read the data sequentially, discarding records that have time stamps earlier than the starting time. This is the reason that having scans in the index table that are much longer than the solution interval slows `BLING` down so dramatically[5].

3.1 The Fringe Search

There are two ways in which a fringe search can be carried out. The first is to transform the data into the delay-rate domain and search for the peak while the second is to model the phase as a function of time and frequency with delay and rate as model parameters and find the parameter values which give the best fit to the actual data. The first method is fast but provides only a coarse approximation to the desired delay and rate. This can be improved by padding the Fourier transform but such techniques are limited. Furthermore the FFT search method gives little help in evaluating error bounds on the fringe parameters which are required for astrometry and geodesy; the only error bounds available are the cell spacings in delay and rate and these do not reflect the quality of the fringe detection. Model-fitting gives results that are more precise and can provide error bounds but is computationally more expensive and requires a starting approximation that is near the global minimum of the penalty function (usually χ^2).

`BLING` combines the two techniques: a windowed FFT search is used to obtain provisional values for the delays and rates which are then refined using a χ^2-fit.

3.1.1 The FFT Search

`BLING` first copies the data to a grid in AP memory, normalizing the amplitudes as it does so. The grid has up to three dimensions corresponding to time, channel frequency and IF frequency. It then consults the list of windows for one that matches the baseline definition and time. The first match in the list is taken and the windows are rounded to pixel coordinates in the FFT grid. If any dimension of the window is zero or negative then the window is set to the full ambiguity range (sometimes inaccurately referred to as the Nyquist range).

Each axis is then Fourier transformed in turn to yield rate, single-band delay and multi-band delay axes. After each axis transformation is done, data that lie outside the window for the corresponding axis are discarded. This has two effects. Firstly, the time required for the subsequent Fourier transforms is reduced since the number of transforms to be done is reduced. Secondly, all of the data that lies inside the search window lies in a contiguous region of AP memory when the transforms have been completed so that the peak amplitude inside the window can be found using a single call to `QCVMMA`. Once the peak has been found, its grid coordinates are converted to a delays and rates in seconds and Hz.

If an acceleration search is requested this procedure is carried out for several different values of the acceleration parameter, rotating the phases of the data points to remove the effects of the acceleration. The amplitude of the peak in transformed data is recorded for each trial and the trial that yields the greatest amplitude is taken to give the nearest approximation to the true acceleration.

3.1.2 Refining the Results

The delays, rates and accelerations from the FFT search are used as the starting approximation for a non-linear χ^2-fit. That is, BLING will attempt to minimize

$$\sum_i \frac{1}{2}\frac{A_i^2}{\sigma_i^2}(e^{j\phi'_i} - e^{j\phi_i})^2 \tag{1}$$

[5]This isn't an issue for most *AIPS* programs which read the data sequentially and keep the file open. Some of the future requirements for `BLING` make it desirable for `BLING` to work baseline-by-baseline and have the capability of moving backwards in time. The rather unusual use of time ranges in `UVGET` is a consequence of these requirements.

where A_i and ϕ_i are the measured amplitude and phase of the i-th data point, σ_i is the estimated error and ϕ_i' is the corresponding model phase. In the `'INDE'` and `'VLBA'` modes the model phase for a data point at time t and frequency ν is calculated as

$$\phi_i' = \phi_0 + 2\pi\tau(\nu - \nu_0) + 2\pi\dot{\tau}(t - t_0) + \pi\ddot{\tau}(t - t_0)^2 \tag{2}$$

where t_0 is the reference time for the solution, ν_0 is the reference frequency, ϕ_0 is a reference phase, τ is the delay, $\dot{\tau}$ is the rate and $\ddot{\tau}$ is the acceleration; ν_0, τ, $\dot{\tau}$ and $\ddot{\tau}$ are the model parameters. The only difference between `'INDE'` mode and `'VLBA'` mode is that the latter combines the data from all selected IFs while the former treats the IFs separately. The model for `'RATE'` mode is the same with τ constrained to be zero while `'MK3'` mode uses

$$\phi_i' = \phi_0 + 2\pi\tau_m(\nu_i - \nu_0) + 2\pi\tau_s(\nu - \nu_i) + 2\pi\dot{\tau}(t - t_0) + \pi\ddot{\tau}(t - t_0)^2 \tag{3}$$

where ν_i is the reference frequency for the IF from which the data point is taken, τ_m is the multi-band delay and τ_s is the single-band delay.

The model fitting is done using the public-domain code for ACM algorithm 717 [2] which has been slightly modified to detect relative function convergence while the curvature matrix is not positive-definite. This modification is necessary to obtain convergence in `BLING`. The optimization package also calculates the covariance matrix for the best-fit parameters. The on-diagonal elements of the covariance matrix are estimates of the variances of the corresponding parameters if it can be assumed that the measurement errors have a Gaussian distribution. The delays, rates and accelerations are scaled to nanoseconds, mHz and μHz/s before performing the fit to avoid numerical underflows while calculating the covariance matrix.

While calculating the model data during the χ^2 fit, `BLING` also calculates the vector sum of the amplitudes for the current values of the delays, rates and errors. The ratio of this quantity to the scalar sum of the amplitudes gives an estimate of the coherence of the data. The coherence is tested against a minimum threshold to determine whether a real fringe has been found. If the coherence meets the threshold test then the results of the fit are written out to a `BS` (**B**aseline **S**olution) table.

An SNR is calculated from the error in the best-fit phase at the reference frequency and is reported to the user but is of little use in determining fringe quality since it merely indicates how well the reference phase is determined and does not reflect how well the rates and delays are determined.

4 A Closer Look at `BLAPP`

`BLAPP` is a much simpler program than `BLING`. It merely sorts the `BS` table into time order and then reads in the baseline-oriented delays and rates for each solution interval. A linear least-squares fit (`SGELSX` from the LAPACK library) is then used to distribute the delays and rates among the antennæ. The results are then written out to a `SN` table. The resulting `SN` table may optionally be applied directly or may be processed using other programs (eg. `SNSMO`) before being applied with `CLCAL`.

The amount of data in the `BS` table is relatively small so that `BLAPP` generally runs very quickly.

A Changes from Previous Versions

The versions of `BLING` and `BLAPP` described in this memorandum differ significantly from the versions available in earlier versions of $\mathcal{AIPS}$. The majority of the changes have been made to increase the robustness of the packages and to make better provisions for future development. The most significant changes are summarized below.

- `BLING` now refines the fringe parameters using the original data instead of fitting an elliptical Gaussian to the peak in delay-rate space as in earlier versions. The earlier method was prone to failure and often produced little improvement over the position from the raw FFT search leading to steps in the delay and rate in the cases where fringes were found. In addition it was difficult to assign meaningful errors to the results as required for geodetic use. The newer method is much more robust but, unfortunately, much slower.

- `BLING` now uses coherence as its quality test rather than SNR.
- `BLING` now reports on its progress to the *AIPS* printer rather than using messages. The maximum line width enforced in the message system is too restrictive for the output information to be presented neatly.
- `BLING` now has a `SOLINT` parameter.
- `BLAPP` uses `SGELSX` rather than `SGELS`; `SGELSX` can detect antennæ that are unconstrained by not having a direct or indirect baseline connection to the reference antenna while `SGELS` can not. Older versions of `BLAPP` wrote nonsensical solutions for unconstrained antennae; the latest version will discard the bad results.

B Future Changes

A number of enhancements are planned for the baseline-oriented fringe-search package during 1995. These include the following which are not necessarily listed in the order in which they will be introduced.

- Introduce a more physically realistic rate and acceleration model for `BLING` where rates and accelerations are assumed to be the same for all IFs. This should improve the quality of the solutions and may also produce a slight improvement in speed.
- Sort the `BS` table output by `BLING` into time order to make it easier to browse.
- Add a task for plotting and editing quantities in `BS` tables.
- Allow the use of fringe solutions to provide the approximate locations of undetected fringes at earlier and later times. This is a requirement for space VLBI where fringes may be most easily found when the orbiting antenna is at apogee, which may only occur part way through an experiment.
- Allow different solution intervals to be used on different baselines and over different time ranges.
- Make use of the acceleration term in `BLAPP`.
- Change the optimization engine in `BLING` to one that uses Greenstadt's modification. This should increase the speed of the program by reducing the number of iterations required to reach a solution.

References

[1] F. R. Schwab and W. D. Cotton. *Astron. J.* **88:**688

[2] D. S. Bunch, D. M. Gay and R. E. Welsch. *ACM TOMS* **19:**109