C A Step-by-Step Recipe for VLBA Data Calibration in $\mathcal{AIPS}$

This appendix provides a step-by-step guide to calibrating many types of VLBA and HSA (High Sensitivity Array or Effelsberg, Arecibo, GBT, and phased VLA) experiments. Continuum strong-source or phase-referencing observations are included, as are simple spectral-line observations. This appendix applies specifically to data sets with full calibration transfer. There is an addendum (§ C.8) describing issues with flagging for non-VLBA data sets and other matters for cases in which not all calibration data are loaded by `FITLD`. It may often be used (with some modifications in loading amplitude data) for data sets containing other antennas. Simple VLBA utilities that go all the way up to and including fringe-fitting are described.

C.1 Table Philosophy

$\mathcal{AIPS}$ follows an incremental calibration process on multi-source data sets. Calibration solutions are written to `SN` ("Solution") tables, which can be inspected in various ways. `CLCAL` is used to apply an `SN` table and write a new `CL` ("Calibration") table, which stores the cumulative calibrations. The actual visibilities are not altered until the final calibration is applied using `SPLIT` (or `SPLAT`), which produces single-source (or multi-source) data sets that can be imaged. With this philosophy, it is easy to back up a step or two if errors are made in processing. Users should keep track of which tables contain which solutions and calibrations as they go through the calibration process.

A key verb to be aware of is `EXTDEST`, which can delete any unwanted table. For example, to delete `SN` version 3 from the data set cataloged as data set 1 on disk 1, type `INDISK 1; GETNAME 1; INEXT 'SN'; INVER 3; INP EXTDEST; EXTDEST`. *Beware of the fact that once a table is deleted, there is no 'undelete' function.*

C.2 Data set assumed in this Appendix

This appendix assumes a VLBA-only data set observed at several frequency bands (*e.g.*, 1.6, 2.3, and 5.0 GHz). To include data from the VLA see § C.8. It is also assumed that phase-referencing programs have been observed according to the philosophy discussed in detail in VLBA Scientific Memo No. 24. The hypothetical observation considered here contains the following sources:

- 'CAL-BAND' — fringe-search and bandpass calibrator
- 'CAL-AMP' — amplitude-check source
- 'CAL-POL' — polarization position angle calibrator
- 'STRONG' — strong target source
- 'CAL-PHASE' — phase-reference source
- 'WEAK' — weak target source, to be calibrated with CAL-PHASE

In the text below, table versions, such as `SN` version 1, are referred to as `SN` 1.

C.3 VLBA Utilities

Note that there are simple VLBA procedures ("front ends" to standard tasks) that will take the user all the way from data loading up to and including fringe-fitting. (The description below applies to the `31DEC02` and later releases, but `VLBATECR` and `VLBAEOPS` are only available in the `31DEC05` release.) These are tremendous labor-savers for those working with reasonably straightforward data sets. For spectral line, use the procedures to calibrate a lower spectral resolution version of the spectral line data and copy the final calibration to the line set. To access the utilities, type `RUN VLBAUTIL` from inside `AIPS`. The currently available procedures that simplify data reduction are

- `VLBALOAD`: loads VLBA data with simplified inputs
- `VLBAFIX`: Fixes VLBA data
- `VLBASUBS`: finds subarrays in VLBA data. Made redundant by `VLBAFIX`
- `VLBAMCAL`: removes redundant calibration data from tables. Made redundant by `VLBALOAD`
- `VLBAFQS`: copies different frequency IDs to separate files. Made redundant by `VLBAFIX`
- `VLBAFPOL`: fixes polarization labeling for common cases. Made redundant by `VLBAFIX`
- `VLBATECR`: automatically downloads and applies ionospheric corrections
- `VLBAEOPS`: automatically downloads and applies corrections to the Earth Orientation Parameters used by the correlator
- `VLBASUMM`: makes summary listings of your data set
- `VLBACALA`: determines *a-priori* amplitude calibrations
- `VLBAPANG`: determines phase corrections for parallactic angles
- `VLBAPCOR`: determines instrumental phase corrections using pulse cals
- `VLBAMPCL`: determines instrumental phase corrections using `FRING`
- `VLBACPOL`: calibrates cross polarization delays
- `VLBAFRNG`: does global fringe fit using `FRING`
- `VLBAKRNG`: does global fringe fit using `KRING`
- `VLBAFRGP`: does global fringe fit for phase referenced experiments using `FRING`
- `VLBAKRGP`: does global fringe fit for phase referenced experiments using `KRING`
- `VLBASNPL`: plots the `SN` or `CL` tables versus time
- `VLBACRPL`: plots the cross-correlation spectrum

There are two additional procedures that can make life easier, called `ANTNUM` and `SCANTIME`. `ANTNUM` will return the antenna number of the antenna corresponding to a certain character string. For example, in many data sets, typing `REFANT = ANTNUM ('BR')` will be the equivalent of typing `REFANT = 1`. `SCANTIME` will return the time range of a given scan number, for use in various programs. Typing `TIMERANG = SCANTIME(4)` will fill the eight-element array `TIMERANG` with the start and stop times of the 4^{th} scan of a given data set. (There must be an `NX` (index) table, within which scan number is row number, for this to work.)

Note 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. Also, note that the procedures do not include data editing, which should be performed at appropriate points in the calibration process. You only need to `RUN VLBAUTIL` once to access all of the procedures. If you run it again for any reason, it is a good idea to type `COMPRESS` immediately afterward to avoid overflowing `AIPS`' symbol memory.

C.4 Data Loading and Inspection

1. Load the data using `VLBALOAD` (which is a very simplified `FITLD`). Typically, the user will set `DOUVCOMP=1` to write compressed data. `CLINT` should be set so that there are several `CL` table entries for each self-calibration or fringe-fitting interval anticipated; this will minimize interpolation error during the calibration process. However, setting `CLINT` too short will result in a needlessly large table. Somewhere between `CLINT = 0.25` and `CLINT = 1.0` is about right. A `FITLD` parameter that is set automatically in `VLBALOAD` is `WTTHRESH = 0.7`, which results in irrevocable discarding of all data with playback weight less than 0.7. The only way around this is to use `FITLD` explicitly. After March 7, 2002, `VLBALOAD` will merge redundant entries in the calibration tables, making `VLBAMCAL` unnecessary.

2. *If you used a version of* `VLBALOAD` *later than March 7 2002, then this step has already been performed.* Merge redundant VLBA gain curve (`GC`), pulse cal (`PC`) and system temperature tables (`TY`) using `VLBAMCAL`. Multiple VLBA correlator jobs create multiple entries in the `GC`, `PC` and `TY` tables; these must be merged before continuing with data reduction. Another, slightly more general procedure which can be used is `MERGECAL` (which is loaded by typing `RUN MERGECAL`). Both these procedures use `TAMRG`, which is a very general, therefore complicated task.

3. After March 7, 2002, correct data with `VLBAFIX`. *If necessary*, `VLBAFIX` sorts (with `MSORT`), splits into different frequencies (with `UVCOP`), fixes the polarization structure (with `FXPOL`), and indexes (with `INDXR`) the data. `VLBAFIX` will also correct for subarrays (with `USUBA`), but you must tell it to do so. There are only 2 inputs of interest in `VLBAFIX`, `CLINT`, the CL table interval, and `SUBARRAY` which should be set to 1 if there are subarrays and 0 if not. The steps in these procedures are very similar to `VLBASUBS, VLBAFQS` and `VLBAFPOL`, which can be run individually instead of `VLBAFIX`. This is a very benign procedure, it can be run on every data set read into `AIPS` and will only perform the necessary fixes. Note that, if the data are split into different frequencies, the flag table is applied and deleted. *If you run* `VLBAFIX` *there is no need to run* `VLBASUBS, VLBAFQS` *or* `VLBAFPOL`.

4. *If you ran* `VLBAFIX` *this step in not necessary.* Deal with subarrays, if needed, using `VLBASUBS`. Note that this should *only* be done if you know there are real subarrays. `FITLD` will err on the side of caution and print messages saying that the data may contain subarrays. When a true subarray condition is found, `FITLD` will print a detailed message listing the source numbers and antennas causing the subarray condition. If needed, `VLBASUBS` will sort the data (with `MSORT`), correct the subarray nomenclature (with `USUBA`), and/or have the index (`NX`) table and calibration (`CL`) version 1 table rebuilt (with `INDXR`). The only user-controllable input for `VLBASUBS` is the `CL` table interval; therefore for more options run `MSORT`, `USUBA`, and `INDXR` separately.

5. *If you ran* `VLBAFIX` *this step in not necessary.* For multi-frequency data sets, separate frequencies into single-frequency data sets using `VLBAFQS`. The procedure `VLBAFQS` will run `UVCOP` to separate the different `FREQID`s, deleting the data flagged by the correlator with `FLAGVER=1`, and then re-index the data set to generate new `CL` and `NX` tables. The only user-controlled input is `CLINT`, the `CL` table interval. Again, for more control, you may use `UVCOP` separately; `INDXR` should not be required. To determine which frequency ID corresponds to which band, run `LISTR` with `OPTYP = 'SCAN'`, or just run `IMHEAD` on each output file. Note that, if the data are split into different frequencies, the flag table is applied and deleted.

6. *If you ran* `VLBAFIX` *this step in not necessary.* Fix polarization labeling, if needed, with `VLBAFPOL`. 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 R and L polarizations are observed but RL and LR are not correlated, either within the same band or in different bands. Each VLBA correlator band is loaded into AIPS as a separate IF and is assigned the same polarization. For the simplest cases of VLBA-only data, the procedure `VLBAFPOL` attempts to determine which polarization case applies and creates a new data set with correct IF and polarization assignments using `FXPOL`. `VLBAFPOL` assumes that all of your `FREQID`s have similar polarization setups. For this reason, you should normally run `VLBAFPOL` after copying each frequency ID to a separate file

using `VLBAFQS`. This strategy also reduces the amount of disk space needed for `VLBAFPOL`. `VLBAFPOL` also will recommend a course of action for more complicated situations. If you want to run `FXPOL` on your own, below are examples of 2 common cases.

(a) If dual polarization (RR and LL, no cross-hands) was used, run `FXPOL` with `BANDPOL = '*RL'` for normal VLBA setups. For Mark IV setups (probably not used for a VLBA-only data set) you may need to run `FXPOL` with `BANDPOL = '*LR'`.

(b) If multiple bands were used, standard setup files probably caused 2.3 and 8.4 GHz to be observed in RCP, while others were observed in LCP. Therefore, when RCP and LCP observations occur in the same program, the polarizations are almost certainly mislabeled. Identify the polarization that you know was used for a given frequency band (*e.g.*, from the schedule file). Then, run `FXPOL` with `BANDPOL = '*L'` to change to LCP, or `BANDPOL = '*R'` to change to RCP. The result can be checked using `IMHEAD` to show the data-set header, which will contain `STOKES = -1` for RCP and `STOKES = -2` for LCP.

7. At this point it is a good idea to get a listing of the antennas and scans in your data by running `VLBASUMM`. `VLBASUMM` runs `PRTAN` over all antenna tables and `LISTR` with `OPTYPE='SCAN'` and gives a choice of writing a text file to disk or sending the listing to a printer.

8. Apply ionospheric corrections, if desired, with `TECOR` or `VLBATECR`. This task uses Global Positioning System (GPS) models of the electron content in the ionosphere to correct the dispersive delays caused by the ionosphere. It is particularly important for phase referencing experiments at low frequency. We recommend `TECOR` for all experiments at 8 GHz or lower. *After November 2005* there is a procedure, called `VLBATECR`, which automatically downloads the needed GPS files and runs `TECOR` with a minimum of intervention. However, to use `TECOR`, you must `ftp` the GPS files; see `EXPLAIN TECOR`. Run `TECOR` with `GAINVER=1, GAINUSE=2, APARM=1,0`, if you have only one `CL` table. Note that `TECOR` had a problem with experiments which approached or crossed midnight, which has been fixed in the `31DEC01` version of AIPS. See `EXPLAIN TECOR` for instructions on what files to download. `TECOR` is only as good as the ionospheric model, so it is a very good idea to compare the corrected and uncorrected phases using `VPLOT`. To inspect the phases using `VPLOT`, use options `BPARM = 0, 2; APARM=0; DOCAL=1; GAINUSE=`*highest* `CL` *table*. The phases should not wind as much (although they will probably not be completely flattened), when the corrected `CL` table is applied. To see the corrections themselves, use `SNPLT` on the new `CL` table setting `OPTYPE = 'DDLY'`.

9. Apply corrections to the Earth Orientation Parameters (EOPs). 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. Even outside these dates the EOPs could be significantly off so it is now recommended that all phase-referencing experiments be corrected for this possible error. `CLCOR` (`OPCODE='EOPS'`) and `VLBAEOPS` will do this correction. The procedure `VLBAEOPS` automatically downloads a file with correct EOPs and runs `CLCOR`. To run `CLCOR` independently you must download a file with correct EOPs by hand, see `EXPLAIN CLCOR` for instructions, and run `CLCOR` setting `OPCODE='EOPS'`.

10. For a simple spectral-line data set, or any data set with high spectral resolution (*i.e.*, more than 16 or 32 channels per IF), it is a very good idea to average the data set to 16 or 32 channels before deriving the calibration parameters. Otherwise, the calibration tasks may take forever to run. It is recommended that you quickly inspect the channels of interest for your line data (*e.g.*, with `UVPLT`) for high points. Remove obviously high amplitudes with `CLIP` (or *e.g.*, `UVMLN`) before averaging. Inspect the full resolution data also for high delays and fringe rates. Spectral averaging in such cases may not be acceptable. Continue calibration on the averaged data set as if it were a continuum set. There is a better method to calibrate spectral line data described in § 9.4.7 and § 9.4.8.12, but the one used here is simpler and will usually give acceptable results. To reduce the data-set size, run the task `AVSPC` with

AVOPTION = 'SUBS'. For example, to average IFs with $N_{\rm chan}$ down to 16 channels, set the adverb CHANNEL = $N_{\rm chan}/16$ (*e.g.*, to average from 1024 to 16 channels, use CHANNEL = 64).

C.5 Amplitude Calibration

Amplitude calibration uses measured antenna gains and system temperatures ($T_{\rm sys}$), as well as finding a correction for voltage offsets in the samplers.

1. Before amplitude calibration is done there must be information for all antennas in the gain curve (GC), system temperature (TY), and weather (WX) tables. There may be missing data; this will be the case for VLBA data before April 1999, or data from non-VLBA antennas. Beginning during November 2003, these tables will normally include information for the VLA and GBT telescopes when they are used. § C.8 has details on how to incorporate the VLA $T_{\rm sys}$ and gain curves for earlier observations or if they are omitted. A similar procedure may be followed for other non-VLBA antennas. Otherwise consult § 9.4.2.5.

2. Correct sampler offsets and apply amplitude calibration by running VLBACALA. The procedure VLBACALA runs several tasks, ACCOR, SNSMO, CLCAL, APCAL and CLCAL. ACCOR (run with SOLINT=2) uses the autocorrelation to correct the sampler voltage offsets. This should always be run for data from the VLBA correlator, since it is significant for 2-bit data and may be important for 1-bit data. After ACCOR creates an SN table, SNSMO smooths the table in order to remove any outlying points. Then the SN table is applied to the highest CL table using CLCAL (using INTERPOL='2PT'), and a new CL table is created. To apply the amplitude calibration, APCAL is run on the highest TY and GC tables, and a new SN table is created. Adverb DOFIT controls whether APCAL also uses the weather tables to fit and correct for opacity. It may be desirable to perform an atmospheric opacity correction at high frequencies, particularly if very accurate source fluxes are needed. See § 9.4.4.2 for a more detailed discussion of APCAL. Lastly, VLBACALA runs CLCAL to apply the amplitude calibration SN table to the CL created by the last run on CLCAL. After running this procedure you will have two new SN tables and two new CL tables. The highest numbered CL table contains all the calibration up to this point. VLBACALA will print messages telling you about the new tables it has created. To keep track of your tables, it is important to copy these messages.

3. At this point it is a *very* good idea to examine your data.

   (a) Run the task SNPLT or procedure VLBASNPL (which is a very simplified SNPLT) to examine the tables created by ACCOR. Use INEXT='CL'; OPTYPE= 'AMP'; INVERS=CL-*table-with-sampler-offsets*; DOTV=1 (to display to the TV; for a hardcopy use DOTV=-1 and LWPLA to print the plot files). The solutions that SNPLT plots should be close to 1000 milligain or 1 gain. Some IFs may be ~ 5% lower than other IFs due to the VLBA system design; application of the ACCOR solutions will (among other things) give proper relative calibration among the IFs.

   (b) Run SNPLT or VLBASNPL to examine the amplitude calibration. This time look at the SN table that APCAL created. Use INEXT='SN'; OPTYPE= 'AMP'; INVERS=*highest* SN *table*; DOTV=1 for VLBASNPL; or to inspect IF m, use SNPLT and BIF = m; EIF = m; OPTYP = 'AMP'; INVER = 1; INEXT = 'SN'; OPCODE = ' '; NPLOT = 10; DOTV = 1; GO SNPLT. For a hardcopy, use DOTV = -1; GO SNPLT; GO LWPLA. Plotted amplitudes are the square-roots of the system-equivalent flux densities (SEFDs), in Jansky, where the SEFD is the flux density of a source that would double the system temperature. (Low numbers are good!) At centimeter wavelengths, VLBA antennas have SEFDs near 300 Jy, so gains above 30° elevation should be near 17–18 and should vary slowly and smoothly with time (*i.e.*, change in elevation) for an individual source. To look at the input system temperatures, run SNPLT with OPTYP = 'TSYS'; INEXT = 'TY'; INVER = 0. On rare occasions, you might find clearly discrepant points that have leaked in from a different frequency band. In that case, you can use task SNEDT, or the clipping option of SNSMO, to get rid of the bad points. You may notice that at low elevations the gains on individual antennas are high. All data below a given elevation can be flagged by running UVFLG; *e.g.*, to flag all data below 10°, run UVFLG with APARM(4)=0 and APARM(5) = 10. Elevation *vs.* time can be listed with LISTR, using OPTYP = 'GAIN'; INEXT = 'SN'; INVER = 1; DOCRT = 1; DPARM(1) = 11. Note that FG tables are not applied to other tables, so flagged data still may have points plotted by SNPLT. The $T_{\rm sys}$ measurements are also a very good diagnostic of bad data from poor weather, equipment failures, *etc.*. If there are time ranges of unusually high or low $T_{\rm sys}$ you may consider flagging those time ranges using UVFLG. Be particularly suspicious of patches of unusual gains at only one IF or STOKES of an antenna. Remember, one of the best things you can do for your final result is to get rid of bad data.

(c) At this point, you may wish to use your favorite method of inspecting data for flagging (*e.g.*, EDITR, TVFLG, IBLED). On-line flags are already included in FG 1 unless they were applied as the data were split into separate frequencies. Use OUTFGVER = 2 initially to copy FG 1 to FG 2, then work with FG 2, so the data do not have to be loaded again if mistakes are made. (But be careful to use the proper FLAGVER and OUTFGVER in various programs.) For example, run EDITR with inputs SOURCES= 'CAL-BAND',' ' (do each source separately); DOCAL=1; GAINUSE=*highest* CL *table*; FLAGVER=2; OUTFGVER=2; DOTWO=1; ANTUSE=1,2,3,4,5,6,7,8,9,10. Once you gain experience you might want to set CROWDED=1 which allows plots of all polarizations and IFs in one plot; this can speed up editing significantly. At this point be concerned about anomalously high or low amplitudes, remember there can be a slow change in amplitude with time due to source structure. Some people do no additional flagging at this stage, but later use the results of fringe-fitting and visibility plots of calibrated data to point the way to bad data, or they do their flagging in the Caltech program DIFMAP.

(d) Run POSSM or VLBACRPL (a simplified version of POSSM) on a calibrator to check that the CL table with the gain corrections has appropriate values. Plot cross-correlation amplitude and phase for a short time period, and examine calibrator flux and phase coherence within each IF. The phase will show a slope *vs.* frequency, indicating an uncalibrated (so far) residual delay. Sample inputs for VLBACRPL are SOURCE = 'CAL-BAND'; REFANT = n; GAINUSE = 2; SOLINT = -1; DOTV = 1. Use STOKES = 'RR' or 'LL' as appropriate. For a weak phase-referencing calibrator, the flux density may look too high due to scalar averaging of the amplitudes, which are dominated by noise. If the data are coherent over the desired time range, using POSSM with APARM(1) = 1 (a vector average), will provide a more realistic estimate of the source flux density. At this point, you may want to note a time with good fringes on all antennas, to use when instrumental phase corrections are made. The only important input to VLBACRPL is the reference antenna REFANT, which it plots. A good choice for the reference antenna is one in the center of the array (PT, LA, FD, or KP for the VLBA) that performed well according to the log, the PI letter, and the initial amplitude calibration and was around for most of the experiment. Hereafter, this is denoted as antenna n.

4. For spectral-line experiments needing velocity accuracy better than 1 km/s, a Doppler correction should be performed. Use CVEL; see § 9.4.5 and § 9.4.6 for details.

5. This is a useful time to run TASAV to save all your ancillary tables to another file. If you foul up the calibration, the relevant tables can be copied back using TACOP.

C.6 Delay, Rate, and Phase Calibration

Now that the data have calibrated amplitudes, the next step is to do the calibration of the antenna delays, rates, and phases. This section describes that process.

1. Correct the antenna parallactic angles, if desired, using VLBAPANG. The RCP and LCP feeds on alt-az antennas will rotate in position angle with respect to the source during the course of the observation

(all VLBA and VLA antennas are alt-az). Since this rotation is a simple geometric effect, it can be corrected by adjusting the phases without looking at the data. You *must* do this correction for polarization experiments. This correction is also important in phase referencing experiments, because the parallactic angle difference between calibrator and target is different at different stations which leads to an extra phase error which can be corrected. `VLBAPANG` copies the highest numbered `CL` table with `TACOP` and then runs `CLCOR` (`OPCODE = 'PANG'; CLCORPRM = 1,0`). `VLBAPANG` has no inputs that require discussion. Be sure to correct the parallactic angles before any of the following steps. Again keep track of which `CL` tables add which correction.

2. Next, the instrumental delay residuals must be removed. These offsets or "instrumental single-band delays" are caused by the passage of the signal through the electronics of the VLBA baseband converters or MkIII/MkIV video converter units. There are two different methods to remove these instrumental delays, one for the case where you have pulse-cal information for some, but not necessarily all, of your antennas; and one for the case where you have no pulse-cal information at all. Note that the preferred method for continuum experiments is to use the pulse-cals, since they correct the instrumental delay over the whole experiment, rather than on a short scan which is "pretty good" for the rest of the experiment. Spectral-line observers would have switched off the pulse-cals as they interfere with line observations, so they are forced to use the second (strong source) method. For VLBA continuum experiments before April 1999, you can load the pulse-cal data using `PCLOD`; consult the § 9.4.2.

    (a) For the case where you have some pulse-cal information, run `VLBAPCOR`. `VLBAPCOR` is another procedure which runs quite a few tasks, `PCCOR`, `CLCAL`, `FRING` (sometimes) and `CLCAL` again (sometimes). `PCCOR` extracts pulse-cal information from the `PC` table and creates an `SN` table. Then `CLCAL` is run to apply that `SN` table to the highest `CL` table, creating a new `CL` table. If there are antennas that do not have information in the `PC` table, or their `PC` entries are wrong, then `VLBAPCOR` runs `FRING` on a short calibrator scan (input `TIMERANGE`). The `SN` table from `FRING` contains corrections for the antennas left out of the `PC` table, and is applied to `CL` table without corrections from `PCCOR`, and added to the `CL` table with the `PC` corrections. For the simplest case of all VLBA antennas, the inputs for `VLBAPCOR` should be `TIMER=`*time range on* `CAL-BAND` *with good fringes for all baselines*; `REFANT=`n; `SUBARRAY=0; CALSOUR='CAL-BAND''; GAINUSE=0; OPCODE=''; ANTENNAS=0`. For the case where you have the VLA (in this example antenna 11), which does not have pulse-cals, your inputs should be the same as above except `OPCODE='CALP'; ANTENNAS=11,0`. For the second case it is important that there are no "Failed" solutions from the run for `FRING`; if there are failed solutions, then you should delete the tables that were created and find another `TIMERANG` with good fringes to the `REFANT`, particularly on the antenna(s) in the `ANTENNAS` list. Also see `EXPLAIN VLBAPCOR` for a detailed description of the steps involved with using pulse-cals and `FRING` without using `VLBAPCOR`.

    (b) The alternate method is to use solve for the phase cals manually with `VLBAMPCL`. This method uses the fringes on a strong source to compute the delay and phase residuals for each antenna and IF. `VLBAMPCL` runs `FRING` to find the corrections and then `CLCAL` to apply them. If there is no calibrator scan that includes all antennas then there is an option to run `FRING` and `CLCAL` again on another source in order to correct the antenna(s) not corrected by the first scan. For the simplest case where all antennas have strong fringes to `CAL-BAND`, set `TIMERANG =` *time range of scan on calibrator with strong fringes to all antennas*; `REFANT =` n; `CALSOUR = 'CAL-BAND'; GAINUSE =` *highest* `CL` *table*; `OPCODE = ''`. If there are no scans that have fringes to all antennas, then you should select a second scan with good fringes to the antennas missing from the first scan (in this example antennas 1 and 9), *plus the reference antenna*. In this case set `OPCODE = 'CALP'; TIME2 =` *time range of second scan*; `SOURCES =` *second calibrator*; `ANTENNAS = 1, 9`.

3. Now you *must* check the results of correcting your instrumental delays using `VLBACRPL` or `POSSM`. Set `GAINUSE=`*highest* `CL` *table*, and plot cross-correlations (`VLBACRPL` will do this for you). The plotted cross-correlations should show the phase slope removed from each IF and there should no longer be a phase jump between IFs, although the phase will not usually be at 0°. If you do a "manual" instrumental delay correction (*i.e.*, you used `FRING`, not `PCCOR`); then the phases far in time from the

scan on which FRING was performed may have a small slope and a small phase jump between the IFs. Also non-zero phase slopes still may be seen at low elevations, where the atmosphere causes additional delay residuals, or for low-frequency observations where the ionospheric delay varies. If POSSM or VLBACRPL is run on a scan of the phase-reference source, CAL-PHASE, there may be more phase noise than for CAL-BAND, because the source is likely to be much weaker. Try APARM(1) = 1 (vector averaging) to get a true measure of the source flux density. If you see significant phase slopes, or phase jumps between IFs on *any* baseline, then the instrumental phase corrections have not worked and you need to figure out why and start again. You can also inspect your new CL table with SNPLT or VLBASNPL. Choose OPTYP = 'DELA' or OPTYP = 'PHAS' and for a given antenna and IF, SNPLT should show a single value of delay repeated for the entire length of the data set, while phase will vary slightly due to the parallactic angle correction and/or the pulse-cal application. If only a portion of the observation appears, there may have been a problem, such as specifying only a certain time range or source in CLCAL. If some antenna was not operating at the beginning (typically MK or BR) or end (typically SC or HN) of the observation, some CL entries will be missing; this is okay. If there appears to be a problem, use SNPLT to look at the previous CL tables to see where things went wrong, delete any erroneous tables, back up to the appropriate stage of the calibration, and move forward from there.

4. Now you must remove global frequency- and time-dependent phase errors using either FRING or KRING or one of the procedures which use these programs, VLBAFRNG, VLBAKRNG, VLBAFRGP and VLBAKRGP. This cannot be done simply for spectral-line sources, so the practice here is to determine delay and rate solutions from the (continuum) phase-reference sources and interpolate them over the spectral line observations. The procedures run either FRING or KRING along with CLCAL. VLBAFRNG and VLBAFRGP use FRING, with VLBAFRGP specifically for phase referencing. Similarly, VLBAKRNG and VLBAKRGP use KRING, with VLBAKRGP specifically for phase referencing. For all these procedures, if the SOURCES adverb is set, then CLCAL is run once 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. These procedures will produce new (highest numbered) SN and CL tables. Since it is probably best to run CLCAL on each source separately, SOURCES should always be set. To use VLBAFRGP for a simple phase referencing experiment (remember that CAL-PHASE is the phase reference calibrator), set CALSOUR='CAL-PHASE','CAL-BAND','CAL-AMP','CAL-POL','STRONG'; GAINUSE=*highest* CL *table*; REFANT=*n*; SEARCH 9 4 1 3 5 6 7 8 10; SOLINT=*coherence time*; DPARM(7)=1 *(if a polarization experiment)*; SOURCES='CAL-PHASE'; 'CAL-BAND'; 'CAL-AMP'; 'CAL-POL'; 'STRONG','WEAK'; INTERPOL='SIMP'. For this example, FRING will be run on the sources in CALSOUR and then CLCAL will be run 6 times, with all of the sources except WEAK referenced to themselves and WEAK referenced to CAL-PHASE, using interpolation method SIMP. For a non-phase-referencing experiment you would use VLBAFRNG with inputs the same as above except for SOURCES, which would not contain WEAK. The results will be the highest SN and CL tables. The INTERPOL to use is a personal preference, AMBG is usually recommended but can cause spurious phase wraps if the rates are very low, which is common with VLBA antennas; SIMP is a simpler interpolation method and therefore more robust. You might want to restrict the channel range slightly using BCHAN and ECHAN, since the channels at the high end of each IF will have lower SNR, due to the cutoffs in the bandpass filters. For a data set with 16 channels per IF, numbered from 1 to 16, setting ECHAN to 14 or 15 may be worth trying. Note that some people like to run CALIB rather than FRING or KRING for this stage of phase-referencing observations, but fringe fitting is recommended, as it solves for rates, which CALIB doesn't do.

   (a) The above fringe fit may take a bit of time, depending on the computer and the spectral resolution. Then, use SNPLT or VLBASNPL to inspect the solutions in the SN table. It's not totally out of the question that some data will be found that need flagging, which can be done with UVFLG. In that case, it's a good idea to delete the last SN and CL table and re-run VLBAFRGP or VLBAFRNG.

   (b) This fringe-fitting stage is the most likely place where things can go wrong, for reasons that are not immediately apparent to the observer. Below, a few common examples are listed.

      - **Many solutions failed.** The source may be too weak, or the coherence time too short. Try increasing or decreasing SOLINT. Or narrow the search window. For most VLBA data, DPARM(2) = 400 and DPARM(3) = 60 should be a good first step, though the rate window specified in DPARM(3) is proportional to the observing frequency, and may need to be larger at 22 GHz and above. Try setting ECHAN so that the top one or two spectral channels in each IF are not used. For more options you could try running FRING and reduce the SNR threshold with APARM(7). Also, if the phase-reference source was too weak, you might try restricting solutions to the shorter baselines with UVRANGE, but it also might be that you're out of luck!

- **Some antenna has low SNR, and may cause an entire set of solutions to go bad.** This typically happens because an antenna should have been flagged. A common cause is when OV is looking at the White Mountains, and neither the on-line system nor the astronomer has flagged the data. Then, you need to run UVFLG and re-run VLBAFRGP or VLBAFRNG.
- **The task fails with some message related to memory allocation.** This may happen if there are lots of spectral channels, or a long SOLINT. Possible solutions are to run AVSPC, to reduce the size of the search window with DPARM(2) and DPARM(3), or to reduce SOLINT. This is much less likely to occur in 31DEC00 and should never occur beginning with 31DEC07.
- **There are discrepant delay/rate solutions.** Look at the solutions you believe, and try VLBAFRGP or VLBAFRNG again with DPARM(2) and DPARM(3) specified appropriately. Full widths are specified, so if the good solutions fall between +15 mHz and −15 mHz, use DPARM(3) = 30. (Actually, you should use a value somewhat larger to allow some margin.) It may be that an antenna is suffering from radio-frequency interference, so some channels and/or IFs will need to be flagged.
- **Some solutions are outside the specified delay/rate range.** This can happen because the initial coarse fringe search uses the range specified by DPARM(2) and DPARM(3), but the least-squares solution can take off from there and go elsewhere.
- **Delays and rates for some station change rapidly near the beginning or end of the observation.** This may be caused by low elevation at the relevant station. Depending on how desperate you are to include low-SNR data, you may wish to flag some time range, or flag all data at elevations below 5° or 10° (particularly at high frequencies).
- **Phases wrap rapidly, particularly on the phase-reference source, CAL-PHASE.** There may not be a lot you can do about this initially, because it's possible that the tropospheric delay just changed too fast for the cycle time used in the observation, especially at low elevation. However, you may wish to note the times and antennas when the phase connection is best (typically the southwestern antennas near transit). Later, when imaging the program source, it can be helpful to image with a subset of antennas and time ranges, then use that initial image to self-calibrate the rest of the data.

5. Use SNPLT or VLBASNPL to inspect the interpolation of the phases in the CL. When you inspect the CL table notice any phase wraps that seem out of place. The human eye is better at pattern matching than a computer and these phases may be in error. If so you might want to run CLCAL independently and try another interpolation method or you might want to edit the CL table. Remember that this is your last calibration table; you want to get rid of any bad calibration now before applying it to the data. Getting rid of spurious wraps in the final CL table using SNEDT will improve your final image more consistently than anything else, *particularly* for phase referencing.

C.7 Final Calibration Steps

1. If you used AVSPC to reduce the size of the data set used in determining calibration, you must copy your final calibration tables back to the full-size data set. This can be done with task TACOP. For bookkeeping purposes, it may be best to copy over all the CL tables with the same table numbers in both the averaged and un-averaged data sets. Copy the FG table as well, since any data which are bad in the averaged dataset will be bad in the full resolution dataset. After inspecting the data with UVPLT

or `VPLOT`, run `EDITR`, `TVFLG`, `CLIP`, `SPFLG`, or other data editor to edit the bad data from the calibrated spectral-line dataset.

2. In some cases (spectral-line observations and continuum experiments seeking dynamic ranges of a few thousand or more, or large fields of view), it is important to calibrate the bandpass shapes. To do this, run `BPASS` on the bandpass calibrator, `CAL-BAND`. Make sure that the spectral line data for the bandpass calibrator is clean and devoid of high points, using `UVPLT` or `SPFLG`. Inputs for `BPASS` are `CALSOUR = 'CAL-BAND'; DOCALIB = 1; GAINUSE =` *highest* `CL` *table*`; SOLINT = 0; BPVER = -1; BPASSPRM(5)=1; BPASSPRM(9) = 1; BPASSPRM(10) = 1; REFANT =` *n*. If the phases vary rapidly duiring the bandpass calibrator scan, then the results using these adverbs will not be satisfactory. Try `BPASSPRM(5)=0; BPASSPRM(10)=4` instead. It is a good idea to do a vector averaged `POSSM` and to look at the bandpass calibrator with `DOCAL=1; GAINUSE=`*highest* `CL` *table* before and after making the bandpass (first with `DOBAND=-1`, then with `DOBAND=1`) to check the result. You can also examine the `BP` table using `POSSM` by setting `APARM(8)=2`.

3. After you have made the `BP` table for spectral line, you may want to correct for the change in frequency by the motion of the antennas with respect to the Sun *etc.*. This is done with `CVEL`, after the source velocities are entered in the `SU` table with `SETJY`. For a detailed description see § 9.4.6.

4. Polarization calibration still remains, if desired, and if all the appropriate calibration sources were observed. This can be done in a variety of ways; see § 9.4.8 for details.

5. Finally, apply the calibration to the visibility data and make single-source data sets using `SPLIT`. (Some people might wish to use `SPLAT` to average over time as well as spectral channel.) Inputs for a continuum observation are `SOURCES = ' '; BIF = 0; EIF = 0; DOPOL = -1` (or 1 if polarization calibration was attempted)`; DOBAND = -1` (or 1 if bandpass calibration was done)`; DOUVCOMP = 1; NCHAV = 0; APARM = 1,0; DOCALIB = 1; GAINUSE =` *highest* `CL` *table*. For a spectral-line observation, set `APARM = 0`, because you don't want to average over frequency. Use `OUTDISK` and `OUTCLASS` as appropriate for your computer and record-keeping purposes.

The single-source data sets are now ready for imaging and possible self-calibration. At this point, it is a good idea to look at the amplitude check source 'CAL-AMP' using tasks such as `UVPLT` or `VPLOT` in order to see if there are any antenna gain calibrations that must be adjusted. Doing a `UVPLT` for each target source is a good idea also, because there may be discrepant amplitude points due to interference or poor fringe fits (among other things). `UVFLG` and `CLIP` are useful tasks to deal with these bad points.

C.8 Incorporating non-VLBA antennas

Beginning in November 2003, calibration data from the VLA and the GBT are incorporated in the tables loaded by `FITLD` *except for flag tables*. See the next section on dealing with flagging. More recently, calibration data from Arecibo and Effelsberg are also incorporated. We retain the following sections for observations made before these dates and for observations made with other telescopes. Note that for other "foreign" (non-VLBA and non-HSA) antennas, a procedure similar to that in § C.8.2 can be followed.

The observation being calibrated may have incorporated either a single VLA antenna or the phased VLA, but the amplitude calibration parameters for the VLA were not transferred automatically. (See VLBA Operations Memo No. 34 for some details.) You will need to create an input text file for the VLA, then run `ANTAB` before `APCAL`. The gains and system temperatures for this file, in an appropriate format, are supplied in a file called *xxxxx*`cal.y.gz'`, where `'`*xxxxx*`'` is the observation code (*e.g.*, 'bm120'), located at `http://www.vlba.nrao.edu/astro/VOBS/astronomy/mmmyy/xxxxx/`. That file contains instructions on editing the file to get correct inputs. For a phased array or a 1.3-cm observation in which 3 antennas are used, follow the instructions in § C.8.3; for a single antenna, use § C.8.2.

C.8.1 Pointing Flags

The VLA, GBT and AR on-line systems produce only recorder-related flags, not pointing flags. Thus, for example, there are no flags for when the telescope is not on-source. This can lead to a large amount of bad data especially if you are changing source frequently. However, the VLBI scheduling program `SCHED` creates a `*.flag` file which contains estimates of how long it takes these antennas to get on source. The `*.flag` file will also contain flags for all antennas in the experiment, so it is best to remove the flags that pertain to the VLBA antennas. The flag file is in a format that can be read in with `UVFLG` using the `INFILE` adverb.

C.8.2 Single VLA Antenna

Beginning in June 2003, the `INDEX`, `GAIN`, and `TSYS` information in this table are reformatted to be directly acceptable to *AIPS*. You should check the times in the text file to make sure that your observation has been properly described. Only a few special cases will require editing of the file; in most cases you are able to invoke `ANTAB` with no editing.

In the input text file, add an `INDEX` entry within the `TSYS` card (**Do not separate the `INDEX` entry from the `TSYS` entry by a "/" !!!**), uncomment the `GAIN` line for your particular observing frequency, and uncomment the `TSYS` line. There are examples of `INDEX` entries in the comments at the head of the file. Then, place this text file in an appropriate directory to read it in with `ANTAB`. The most straightforward step is to place it in directory `$FITS` with filename `VLA.ANTAB`. At the AOC in Socorro, the `$FITS` directory for a given computer, *e.g.*, 'laguna', is located at `/DATA/LAGUNA_1/FITS`.

1. After merging the calibration tables for the VLBA antennas, prevent confusion and any chance of having to re-run `FITLD` by first copying the VLBA `TY` and `GC` tables. If you used `VLBAMCAL`, the VLBA parameters will be in `TY` 1 and `GC` 1, and they should be copied to `TY` 2 and `GC` 2. If you used `MERGECAL`, the VLBA parameters will be in `TY` 2 and `GC` 2, which should be copied to `TY` 3 and `GC` 3. For the example used here, then run `ANTAB` with `INFILE = 'FITS:VLA.ANTAB'`, setting `TYVER` and `GCVER` to their highest numbered values (either 2 or 3). If `ANTAB` fails, it is most likely caused by having an incorrect format for the input file. Perhaps you forgot to add the `INDEX` entry within the `TSYS` card, or gave it the wrong format, or failed to uncomment the `GAIN` or `TSYS` lines.

2. Now, run `VLBACALA` as described in § C.5 to combine the gain and system temperature information for all antennas into the appropriate `SN` table. Therefore, the `APCAL` inputs should include `TYVER = 0; GCVER = 0; ANTENNAS = 0`. If `VLBACALA` fails while running `APCAL`, it is possible that your input file was not properly set up for `ANTAB`. Perhaps the `INDEX` line gives polarizations that are inconsistent with those specified in the headers of the VLBI data set, which could mean you forgot to run `FXPOL`. Then, use `SNPLT` or `VLBASNPL` as described in § C.5 to make sure that the resulting `SN` now contains amplitude calibration for all VLBA antennas and the VLA. At most frequencies, the VLBA antennas should perform slightly better than a VLA antenna, so the amplitude gains plotted for the VLA antenna will be slightly higher. **Make sure that all antennas and IFs are included!**

C.8.3 Phased VLA

The VLA may be phased on a program source ('STRONG'), or may be phased on a phase-reference source ('CAL-PHASE'), with the resulting solutions applied to the program source ('WEAK'). Rather than recording a system temperature, the VLA system will record a ratio of antenna temperature to system temperature, which will vary as the array phases up. In order to convert the ratio of antenna and system temperatures to a usable gain, the flux density of some source will be needed.

1. Load and calibrate the VLA data by standard means (see Chapter 4). Determine the flux density of a relevant strong source, usually either 'STRONG' or 'CAL-PHASE'. Then, on the VLBI data

set, insert the flux density of this source into the `SU` table using `SETJY`. For example, if the source is 'CAL-PHASE' and its flux density is 0.432 Jy, run `SETJY` with `SOURCES = 'CAL-PHASE'; BIF = 0; EIF = 0; ZEROSP = 0.432,0; OPTYPE = ' '`.

2. Edit the input file as indicated above for a single VLA antenna. Again, an `INDEX` line, a `GAIN` line, and a `TSYS` line must be checked (after June 2003) or be created or uncommented. The `GAIN` line is independent of observing band (the source flux is used to determine the gain), and the `TSYS` line should include the parameter `'SRC/SYS'`, indicating that the ratio of antenna temperature to system temperature is being supplied.

3. Run `ANTAB` to read in the input file of amplitude calibration parameters. Then run `VLBACALA` to put this in an `SN` table. Both steps are essentially the same as for a single VLA antenna (see § C.8.2). The most likely problem is that `APCAL` in `VLBACALA` will fail because you forgot to enter a source flux density using `SETJY`, although the error message may not always make this obvious.

4. Run `VLBASNPL` or `SNPLT` to inspect the resulting `SN` table, as for the single VLA antenna. In this instance, you should see that the phased VLA is very sensitive. If the phasing worked well at centimeter wavelengths, the amplitude should be near 4 or 5 instead of the value of 17 or 18 seen for a single VLBA antenna. At the start of scans where the VLA is being phased, you may see a rapid change in the amplitude gain (toward smaller numbers) as the antenna phases are brought into alignment. The `SN` table should be inspected very carefully, because there may be data that should be flagged when the VLA phasing did not work well. Three possible reasons for poor phasing are (1) the source is too weak; (2) the troposphere is misbehaving; or (3) there was radio-frequency interference at the VLA.

C.8.4 Summary

Following the insertion of the amplitude solutions for the VLA, you can return to follow the standard path for calibration of VLBA data. Although the procedures from here on are identical to the VLBA-only case, the observer may wish to pay attention to several issues.

1. The phased VLA is far more sensitive than a single VLBA antenna, so it is often a good idea to use the phased VLA as the reference antenna for fringe-fitting.

2. The phased VLA has a large delay offset which should have been taken into account by use of a GPS file during correlation. Still, the user should pay close attention to the fringe fits, and be aware of the possibility that the VLA may have larger residual delays and rates than a VLBA antenna.

3. The VLA does not slew as rapidly as the VLBA. The `FG` table supplied by calibration transfer includes back-end flags only, and does not incorporate information about the pointing of the VLA antennas, and when they arrive on source. Therefore, some judicious flagging by the user may be necessary. It is very likely you will have low amplitudes on VLA baselines for the first 10 or so seconds of each scan; the AIPS task `QUACK` can be used to flag these low amplitudes. For example, if the VLA is antenna 11, use inputs `ANTENNAS=11,0; FLAGVER=1; OPCODE='BEG'; REASON='QUACK:VLA'; APARM=0, 0.2` in `QUACK`.

4. The VLA elevation limit is 8°, while the VLBA antennas can go much lower. This means that a source may set at the VLA well before it sets at Pie Town or Los Alamos, for example.

5. The VLA observing time is allocated in Local Sidereal Time rather than UTC. Therefore, it may start or finish observing as much as 15 or 20 minutes before/after the VLBA, even if the same amount of time is allocated.

C.9 Some Useful References

1. Chatterjee, S., "Recipes for Low Frequency VLBI Phase-referencing and GPS Ionospheric Correction," VLBA Scientific Memo No. 22, May 1999.
`http://www.vlba.nrao.edu/memos/sci/`

2. Ulvestad, Jim, "VLBA Calibration Transfer with External Telescopes, Version 1.1," VLBA Operations Memo No. 34, July 30, 1999. `http://www.vlba.nrao.edu/memos/vlba/vba.oper.txt`

3. Ulvestad, Jim, "A Step-By-Step Recipe for VLBA Data Calibration in AIPS, Version 1.3" VLBA Scientific Memo No. 25 (the basis of this appendix), January 2, 2001.
`http://www.vlba.nrao.edu/memos/sci/`

4. Ulvestad, Jim, Greisen, Eric W., Mioduszewski, Amy "AIPS Procedures for Initial VLBA Data Reduction, Version 2.0" AIPS Memo No. 105 April 26, 2001.
`http://www.aaips.nrao.edu/aipsdoc.html # MEMOS`

5. Wrobel, J. M., Walker, R. C., Benson, J. M., & Beasley, A. J., "Strategies for Phase Referencing with the VLBA," VLBA Scientific Memo No. 24, June 2000.
`http://www.vlba.nrao.edu/memos/sci/`

C.10 Additional recipes

C.10.1 Chewy banana split dessert

1. Prepare and bake one package (19.8 0z) chewy fudge (or other favorite) **brownie mix**. Allow to cool thoroughly, four hours or more.
2. Peel 2 large ripe **bananas** and place very thin slices on top of brownie.
3. Cover bananas evenly with one 12-oz. container of **whipped topping** (thawed) and drizzle 1/2 cup **chocolate syrup** over that.
4. Refrigerate to chill completely. Cut into squares to serve.

C.10.2 Orange baked bananas

1. Mix in a saucepan 1/2 cup firmly packed **brown sugar**, 1 tablespoon **cornstarch**, 1/8 teaspoon **cinnamon**, and a few grains **salt**.
2. Add gradually, blending in 3/4 cup boiling water.
3. Bring rapidly to boiling and cook about 5 minutes or until sauce is thickened, stirring constantly.
4. Remove from heat and blend in $1\frac{1}{2}$ teaspoons grated **orange peel**, 1/4 cup **orange juice**, 1 teaspoon **lemon juice**, and 2 tablespoons **butter**.
5. Peel and cut into halves lengthwise 6 **bananas** with all-yellow or green-tipped peel.
6. Arrange halves cut side down in baking dish and brush with about 2 tablespoons melted **butter**.
7. Sprinkle 1/2 teaspoon **salt** over bananas and then pour the orange sauce over bananas.
8. Bake at 375° F for 10 to 20 minutes.

C.10.3 Little banana cream tarts

1. Preheat oven to 325° F.
2. Combine 6 tablespoons **margarine** or butter (softened), 1/4 cup packed **brown sugar**, 1/4 cup **powdered sugar**, and 1/2 teaspoon **vanilla extract**.
3. Stir in 2/3 cups crushed **cereal** (2 cups un-crushed Multi-Bran Chex suggested), 1/2 cup all-purpose **flour**, and 1/3 cup finely chopped **nuts** (optional).
4. Divide dough evenly into 12 balls. Place each ball in 2.5-inch muffin cup; press into sides. Bake 8 to 10 minutes.
5. Let stand in pan 15 minutes. Use knife to remove each tart carefully from pan. Tarts will be very soft. Let cool completely.
6. Melt 2 tablespoons **margarine** in skillet over low heat.
7. Stir in 2 tablespoons **heavy cream**, 4 tablespoons packed **brown sugar**, and 1/8 teaspoon **allspice**. Cook until sugar is dissolved, stirring occasionally.
8. Stir in 3 medium **bananas**, sliced. Divide filling evenly among the cooled tarts and garnish with whipped cream.

Thanks to Ralston Purina Company.

C.10.4 Banana mandarin cheese pie

1. In large mixer bowl, beat 8 ounces softened **cream cheese** until fluffy.
2. Gradually beat in 8 ounces **sweetened condensed milk** until smooth.
3. Stir in 1 teaspoon **lemon juice** and 1 teaspoon **vanilla extract**.
4. Slice 2 medium **bananas**, dip in lemon juice, and drain.
5. Line 8(?)-inch **graham cracker pie crust** with bananas and about 2/3 of an 11-ounce can (drained) **mandarin oranges**.
6. Pour filling over fruit and chill for 3 hours or until set.
7. Garnish top with remaining orange segments and 1 medium **banana** sliced and dipped in lemon juice.

C.10.5 Banana Dream Pizza

1. Preheat oven to 400° F. In a large bowl, combine 2 1/2 cups **all-purpose flour**, 2 tsp **baking powder**, and a pinch of **salt**. Add 4 Tsp softened **sweet cream butter** and blend. Add 3/4 cup warm **milk** and mix well. If the dough is still sticky, add a small amount of flour.
2. Form the dough into a ball. Knead it on a floured surface until it is smooth. Roll out the dough and place it in an oiled, 16-inch pizza pan. Bake for 15-20 minutes, or until the crust is light brown.
3. In a nonmetallic bowl, mash 4 **bananans**. Add 1 teaspoon **lime or lemon juice** and 6 tablespoons **honey**; mix well.
4. Slice 2 **bananas** horizontally and place the slices in water to cover. Add 1 teaspoon **lime or lemon juice** to prevent discoloration.
5. Spread the banana mixture on the crust.
6. Drain the sliced bananas and blot them with paper towels. Place them in a circular pattern on the banana mixture. Baste the banana slices with 3 tablespoons **melted butter**.
7. Bake for 20–30 minutes at 400° F until the crust is golden brown.
8. Remove from the oven and top with 1 quart **vanilla ice cream** and 1/2 cup chopped **macadamia nuts** while still hot. Serve immediately.