E Special Considerations for EVLA Data Calibration and Imaging in $\mathcal{AIPS}$

The old VLA with its once state of the art, but now dated, correlator and electronics has been turned off. The new electronics and correlator of the EVLA[1] have been turned on and made available to users. Observing is now on the customary terms although a "Resident shared risk" program is still available for certain esoteric, not fully commissioned, capabilities. Data may have many thousands of spectral channels and up to 8 GHz of bandwidth per polarization. $\mathcal{AIPS}$ software is capable of handling these data.

The EVLA has already produced amazing scientific results although the large data volume has created difficulties. Delays are sometimes not set accurately, causing the data analysis to begin with the "VLBI" task `FRING` needed to correct large slopes in phase across the bandpass. The flagging information known to the on-line system (telescope off source and the like) is now available in the data format and so can reach $\mathcal{AIPS}$ either as an initial flag table or as already flagged data. Substantial flagging effort may therefore still be required, particularly to deal with RFI which inevitably occurs in the wide bandwidths. System temperatures and gains are now transferred, but should be applied with caution to scale the visibilities and to compute data weights. The data weights without this adjustment reflect only integration time. The weather table is available with the data so that reasonable opacities may be determined. However, the "over-the-top", table which is used in determining antenna positions, the frequency offset table, used in managing Doppler tracking, and the `CQ` table, used to correct amplitudes for spectral averaging in the presence of non-zero delays, are not available if the data come to $\mathcal{AIPS}$ via CASA. The `OT` table is now provided when the data are read using the `OBIT` package, but astrometric data used to compute accurate projected spacings are just now becoming available.

The following guide will not assume that all issues have been fixed. Steps that can be omitted or simplified when they are will be described. This appendix is written with the assumption that the reader is moderately familiar with $\mathcal{AIPS}$ as described in the preceding chapters. It is also written with the assumption that you are using current versions of the `31DEC11` or later releases of the software.

E.1 Getting your data into $\mathcal{AIPS}$

Your EVLA data are stored as an "ALMA Science Data Model" (ASDM) format file in "SDMBDF" (Science Data Model Binary Data Format) in the NRAO archive. They may be read out of the archive in that format, a CASA measurement set format, or in an $\mathcal{AIPS}$-friendly uvfits format. This last is produced by the CASA uvfits writing software. Go the the web page

`http://archive.cv.nrao.edu/`

and select the Advanced Query Tool. Fill out enough of the form to describe your data and submit the query. If the data are not yet public, you will need the Locked Project Access Key which may be obtained from the NRAO data analysts. To avoid the need for this key, you may log in to `my.nrao.edu` after which it will know if you are entitled to access particular locked projects. The query will return a list of the data sets which meet your specifications. *(Users logged in to an NRAO Socorro computer should use a variant of these instructions; see below.)* On this form, enter your e-mail address, choose AIPS Friendly names (almost certainly does not work), AIPS FITS under the EVLA-WIDAR section and choose the desired time averaging. If the delays are not accurately known, spectral averaging can be damaging to the data amplitudes. However, the data are often recorded at one-second intervals which is rather short, making the data voluminous. Judicious averaging can help with data set size and processing times without compromising the science. Choose the

[1] Now officially called the Karl G. Jansky Very Large Array, but called the EVLA here.

data set(s) you wish to receive and submit the request. You will be told an estimate of the output data set sizes and the amount of time you will need to wait for the format translation to occur. A 19 Gbyte SDMBDF file run as a test with no averaging was estimated to produce a UVFITS file of 30.26 Gbytes and to take 103 minutes to prepare for download. That time assumes that your download job is the only one being performed. If your download fails, you will probably be told erroneously by e-mail that it worked. The output file will however be missing or incomplete. Try again before contacting NRAO for help.

If you are logged into an NRAO Socorro computer — and perhaps if you are not, — you may find a better route to acquire your data with useful additional information not available via CASA. Try requesting your data in SDMBDF format instead of uvfits and also uncheck the "Create tar file" box *only if you are on a Socorro computer*. (Remote users will need to have the archive load the SDMBDF file into "tar" form in the public ftp area for copying to their home machine.) If you have enough disk space, direct the data to a directory on your machine, *after you have made that directory world writable* (`chmod 777`*directory-name*). The downloading of the SDMBDF file is quite efficient compared to having the archive computer do all the file translation. When you are told that the SDMBDF file is ready, you may run `AIPS` and load the data via the verbs `BDFLIST` and `BDF2AIPS`. These verbs run programs in the `OBIT` software package to load your data directly into *AIPS* including flag (`FG`), index (`NX`), calibration (`CL`), over-the-top (`OT`), SysPower (`SY`), and CalDevice (`CD`) tables which you will not get from CASA. If you use this approach, you may skip the `UVLOD` and `INDXR` steps described below. Note that these verbs require that `OBIT` be installed on your computer — as it is in Socorro — and that `ObitTalk` be in your `$PATH`. `OBIT` is relatively easy to install and may be obtained from `bcotton@nrao.edu`.

Unlocked files will be downloaded to the NRAO public ftp site

`ftp://ftp.aoc.nrao.edu/e2earchive/`

by default and you may then use ftp to copy the file to your computer. Locked files will go to a protected ftp site and you must use ftp to download those, even within NRAO. The instructions for downloading will be e-mailed to you. Be sure to specify *binary* for the copy. If you are located in the AOC in Socorro, you may set an environment variable to the archive location, *e.g.*,

export E2E=/home/ftp/pub/e2earchive C$_R$	for bash shells
setenv E2E /home/ftp/pub/e2earchive C$_R$	for C shells such as tcsh

and simply read unlocked data files directly from the public download area. Note that the file will be deleted automatically after 48 hours in both public and protected data areas.

SDMBDF files may be read into *AIPS* using `BDFLIST` to learn what is in your data set and then `BDF2AIPS` to translate the data. Thus

> DEFAULT 'BDF2AIPS'; INP C$_R$	to initialize all relevant adverbs.
> DOWAIT 2 ; DOCRT 1 C$_R$	to wait for the verbs to finish and to display the log file on the terminal after the `OBIT` task finishes. `DOWAIT 1` displays the messages as they are generated but is insensitive to returned error conditions from the `OBIT` tasks. Be sure to set `DOWAIT -1` after using `BDF2AIPS`.
> ASDMF(1) = '*path_to_asdm_dir* C$_R$	to set the full path name into the adverb. Note the lack of close quote so that case is preserved. If the name is too long (> 64 characters), put part of the name in `ASDMF(1)` and the rest in `ASDMF(2)`. Trailing blanks in `ASDMF(1)` will be ignored.
> BDFLIST C$_R$	to list the contents of the SDMBDF. Note particularly the "configuration" numbers.
> OUTNA '*myname*' C$_R$	to set the *AIPS* name.
> OUTCL ' ' C$_R$	to take default (`UVEVLA`) class.
> OUTDI 3 C$_R$	to write the data to disk 3 (one with enough space).

> DOUVCOMP FALSE C_R	to write visibilities in uncompressed format. There are no weights at present, so there is no loss of information in compressed format, but the conversion from compressed format costs more than reading the larger data files.
> FOR CONFIG = 0:100 ; BDF2AIPS; END C_R	to load all of the configurations in your data, terminating with error messages on the first configuration number not present in your data (when DOWAIT is 2).

There are other adverbs — NCHAN, NIF, BAND, and CALCODE — available if needed to limit which data are read. CONFIG is frequently all that is needed to select data, but these others may be needed if more complicated modes of observing were used.

The uvfits data file may be read from disk into AIPS using UVLOD or FITLD, using:

> DEFAULT 'UVLOD' ; INP C_R	to initialize and review the inputs needed.
> DATAIN 'E2E:*filename*' C_R	where *filename* is the disk file name in logical area E2E; (see § 3.10.3).
> OUTNA '*myname*' C_R	to set the AIPS name.
> OUTCL ' ' C_R	to take default (UVDATA) class.
> OUTSEQ 0 C_R	to take next higher sequence #.
> OUTDI 3 C_R	to write the data to disk 3 (one with enough space).
> DOUVCOMP FALSE C_R	to write visibilities in uncompressed format. There are no weights at present, so there is no loss of information in compressed format, but the conversion from compressed format costs more than reading the larger data files.
> INP C_R	to review the inputs.
> GO C_R	to run the program when you're satisfied with inputs.

Watch the messages from UVLOD to see where your data set goes and whether the task ran properly. When it is finished, check the output header:

> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> IMHEAD C_R	to examine the header.

Note that the header does not show the usual complement of AIPS extension files. CASA translates the on-line data into its internal format and then writes the uvfits file read by AIPS. Since CASA does not have files comparable to AIPS index and CL tables, it does not provide them. To build index and calibration tables, use;

> TASK 'INDXR' ; INP C_R	to select the task and review its inputs.
> INFILE ' ' ; PRTLEV = 0 C_R	to be sure not to use an input text file and to avoid excess messages.
> CPARM = 0 , 0 , 1/2 C_R	to make a CL table 1 with a 30-second interval.
> BPARM τ , 0 C_R	to take default EVLA gains and a zenith opacity of τ. Set $\tau = -1$ for no opacity correction. You may set $\tau = 0$, which is now recommended, to get *new* default opacities. These are based on a detailed model predicting the opacity at any frequency from that at 22 GHz. The combination of weather and seasonal model long used by FILLM and INDXR is now used solely to estimate the 22 GHz opacity.
> GO C_R	to run the task after checking the inputs.

INDXR now uses new EVLA gains information which includes some frequency dependence within the new wide bands. OBIT's application of weather and gains appears to have errors — re-create the CL table 1 and index table with INDXR even if you got your data via OBIT.

It is a good idea to list the structure of your data set and your antenna locations on the printer and to keep those listings next to your work station for reference:

> DEFAULT LISTR ; INP C_R	to initialize the LISTR inputs and review them.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> OPTYPE 'SCAN' ; DOCRT -1 C_R	to choose a scan listing on the printer.
> GO ; GO PRTAN C_R	to print the scan listing and the antenna file contents.

Read these with care. There have sometimes been problems with antenna identifications, with the order of the IF frequencies, and even with identification of sources by scan. Task SUFIX may be used to correct the last problem and, if desired, FLOPM may be used to reverse the frequency order. Oddly ordered IFs may require UVCOP to split them apart followed by VBGLU to paste them back together. You may have to use SETJY or TABPUT to change the CALCODE of some sources if your calibration sources have a blank calibrator code or your target sources have a non-blank calibrator code.

E.2 Initial calibration — VLANT, FRING

As with the VLA, NRAO maintains text files describing any changes which are made to our estimate of the antenna locations. Users may wish to apply these changes if their data were taken between the time when antennas were moved to their current stations and the time that the corrections were entered into the on-line control data base. Task VLANT works for both VLA and EVLA data, reading these text files and performing the needed changes to the CL table, writing a new one. Thus

> DEFAULT VLANT ; INP C_R	to initialize the VLANT inputs and review them.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> GO C_R	to run the task, writing CL table 2.

We have had difficulty setting all of the delays in the EVLA to values which are sufficiently accurate. If the delay is not set correctly, the interferometer phase will vary linearly with frequency, potentially wrapping through several turns of phase within a single spectral window ("IF band"). We hope that bad delays will not arise in future, allowing you to skip this section, but use POSSM to check for phase slopes. However, even very modest delay errors may matter over the very wide bandwidths of the EVLA. Note that FRING computes its values around the center of each spectral window but then writes a phase correction appropriate to the frequency reference pixel. Use IMHEADER to see what this is in your data set. Traditionally it has been 1.0, but $N_{\rm chan}/2 + 1$ is a better choice. New task CENTR, or numerous other tasks with adverb FQCENTER, will convert your data set to this reference. Note that this conversion requires re-scaling the u, v, w values and adjusting some FQ table values; it is not just a simple change in the header.

Delay errors are problem familiar to VLBI users and *AIPS* has a well-tested method to correct the problem. Using your LISTR output, select a time range of about one minute *toward the end of a scan* on a strong point-source calibrator, usually your bandpass calibrator. Then

> DEFAULT FRING ; INP C_R	to initialize the FRING inputs and review them.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> SOLINT 1.05 * x C_R	to set the averaging interval in minutes slightly longer than the data interval (x) selected.
> TIMERANG *db* , *hb* , *mb* , *sb* , *de*, *he* , *me* , *se* C_R	to specify the beginning day, hour, minute, and second and ending day, hour, minute, and second (wrt REFDATE) of the data to be included. Too much data will cause trouble.
> DPARM(9) = 1 C_R	to fit only delay, not rate. *This is very important.*

> DPARM(4) = t CR	to help the task out by telling it the integration time t in seconds. Oddities in data sample times may cause `FRING` to get a very wrong integration time otherwise.
> INP CR	to check the voluminous inputs.
> GO	to run the task, writing `SN` table 1 with delays for each antenna, IF, and polarization.

The different IFs in current EVLA data sets may come from different basebands and therefore have different residual delays. The option `APARM(5) = 3` to force the first $N_{if}/2$ IFs to have one delay solution while the second half of the IFs has another is strongly recommended, but only when the first half all come from one of the "AC" or "BD" basebands (hardware IFs) and the second half come from the other. The 3-bit data path of the EVLA actually has four hardware IFs, so `APARM(5) = 4` produces four delay solutions, dividing the IFs in quarters. Note that, at low frequencies, the phases may also be affected by dispersion (phase differences proportional to wavelength). `FRING` now offers `APARM(10)` to enable solving for a single delay plus dispersion from the fitted single-IF delays. This `SN` table will need to be applied to the main `CL` table created by `INDXR` or `OBIT`.

> TASK 'CLCAL' ; INP CR	to look at the necessary inputs.
> TIMERANG 0 CR	to reset the time range.
> GAINUSE 0 ; GAINVER 0 CR	to select the highest `CL` table as input and write one higher as output (version 2 and 3, resp. in this case).
> SNVER 1 ; INVER 1 CR	to use only the `SN` table just created.
> INP CR	to review the inputs.
> GO CR	to make an updated calibration table.

Be sure to apply this (or higher) `CL` table with `DOCALIB 1` in all later steps.

E.3 Initial editing

You should use the tools below to flag out obviously bad data. The tasks which automatically flag data for you, however, depend on meaningful amplitudes and flat spectral shapes. Therefore, flagging and calibration are an iterative process. Do the obvious flagging without spending a lot of time on it. Then do an initial calibration of bandpass, amplitude, and phase. Use that calibration to run `RFLAG` and/or other auto-flagging tasks. Then throw away all `CL` tables following `VLANT` and begin again with `FRING`, `BPASS`, `CALIB`, etc.

There will be data validity information prepared both by the on-line control software and by the WIDAR correlator and most of this information is available as an initial flag table. The tasks above will have applied this table for you by default since `FLAGVER 0`. On-line flags may already have caused data to be flagged within your data set (but *not* deleted) by CASA. Unfortunately, a flag table is present only via the `OBIT` route. `UVFLG` will be needed to add flags for shadowing (`APARM(5)`=25 or so) to a flag table. We still need to look at the data to flag out whatever remains of the time off source not flagged using on-line flagging information. There have also been drop outs in which the visibility is pure zero, typically for all channels and IFs and a single integration. The drop outs should now be handled by `OBIT`, `UVLOD`, and `FITLD`. Note, however, that CASA and `FITLD` pass along all data samples, including those that are fully flagged. This makes the data set rather larger than one might wish. Use `UVCOP` (or `TYAPL` — see § E.6) to remove all fully flagged data samples. Before doing this, use `TVFLG` (§ 4.4.3) to look for any more data samples that might need to be flagged fully. Check especially samples at the beginnings and ends of scans. Try

> DEFAULT TVFLG ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m CR	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND -1 CR	to apply the delay calibration. If a bandpass has been determined, use `DOBAND 3` or 1 to apply it.

> BCHAN $c1$; ECHAN $c2$ C_R — to average across a range of channels — not as flexible as ICHANSEL but probably okay here.

> NCHAV ECHAN-BCHAN+1 C_R — to average all the channels into one number.

> BIF j ; EIF BIF C_R — to edit one IF only, which will suffice for problems that are not IF dependent, such as drop outs, antenna not on source, etc. Choose an IF that is reasonably free of RFI.

> CALCODE '*' C_R — to do just calibrators for the moment.

> DPARM(6) $\Delta\, t$ C_R — to do no time averaging in the work file set $\Delta\, t$ to the data interval in seconds.

> GO C_R — to start the task.

The default smoothing time shown in the display will probably be some multiple of $\Delta\, t$. Change the smoothing time to one times the basic interval in order to edit in detail; TVFLG will break up your data into overlapping "groups" which you may edit is sequence. Remember to change the initial setup so that the flags apply to all channels and all IFs. See § 4.4.3 for more information. The EVLA has shown a tendency to produce periods of data which are too low in amplitude to be normal noise, but which are not zero. Use TVFIDDLE or TVTRANSF functions to enhance the brightness of the amplitudes to make sure that apparently black regions really are black (flagged already).

We note here that some users feel that the data need to be inspected more carefully than with just an average of most of the channels. POSSM (below) may be of use to find RFI. Avoiding the worst of that, you may still wish to run TVFLG to look at the average of a few channels at a time. Use NCHAV and CHINC appropriately. Task SPFLG (§ 10.2.2) is the ultimate weapon when looking for channel-dependent difficulties and is even capable of normalizing by source flux including spectral index. However, it is onerous when there are many baselines. FTFLG is similar to SPFLG but displays all baselines together. It would be a quick way to look for wide-spread RFI. These more onerous tools should probably not be used at this preliminary stage; use them after some of the auto-flagging tasks have been run.

E.4 Basic calibration

For data taken with the new low-band receivers (frequencies rather less than 1 GHz), spectral windows may cover a frequency range so wide that the sources vary quite a bit across the individual window. Such wide windows may be convenient for taking the data, but later data reduction will be more reliable with narrower spectral windows. The task MORIF in 31DEC12 may be used to increase the number of spectral windows by a user-specified factor. To avoid having to use NOIFS ahead of MORIF, task REIFS was written. It is more flexible in placing the new IFs, including allowing them to include data from more than one input IF.

For *both* continuum and line observations, we must begin by determining which spectral channels are reliable and which are affected by the inevitable loss of signal-to-noise at band edges or are degraded by radio-frequency interference (RFI). Use POSSM to display spectra from the shorter baselines on the TV:

> DEFAULT POSSM ; INP — to set the task name and clear the adverbs.

> INDI n; GETN m C_R — to select the data set on disk n and catalog number m.

> SOURCE '*bandpass_cal*' C_R — to select the strong bandpass calibrator.

> DOTV 1 ; NPLOTS 1 C_R — to plot only on the TV, one baseline at a time.

> ANTEN $n1$, $n2$, $n3$, $n4$ C_R — to select the antennas nearest the center of the array or the maintenance areas.

> BASELINE ANTEN C_R — and only them.

> DOCAL 1 ; APARM 0 C_R	to apply the FRING solutions and display vector averaged spectra. Scalar averaged spectra will turn up at the edges reflecting the decreased signal to noise in the outer channels.
> APARM(9)=1 C_R	to plot all IFs in a single plot.
> GO C_R	to run the task. Make notes of the desirable channels IF by IF.

If there is no RFI, then you may be able to use the same channel range for all IFs. If the RFI is particularly pernicious, you may have to edit it out of your data before continuing; see § E.5. The first time through this section, you should accept but perhaps try to avoid the worst of the RFI. After the detailed editing, that RFI should be gone.

POSSM may reveal extensive ringing in your spectra due to narrow RFI signals. Try SMOOTH=1,0 to apply Hanning smoothing. If this proves beneficial, you should apply this SMOOTH, plus the initial flag table and calibration to the data once and for all with SPLAT. Using SMOOTH in all operations can produce errors in bandpass functions (if you forget it once in a while) and will produce especially strange results when you use the channel-dependent auto-flagging routines such as RFLAG..

For polarization calibration, it is assumed that the phase difference between the right and left polarizations of your calibration is stable with time. Thus, if polarization is important, it is *critical* to find a reference antenna with a stable right minus left phase. Use CALIB with SOLMODE 'P' and as short a time interval as possible on your strongest calibration sources. Use SNPLT with STOKES 'DIFF' and OPTYPE 'PHAS' to look at the right minus left phases in the SN table produced by CALIB. Find the one that is the most stable and use that as REFANT henceforth. To avoid later confusion, delete the SN used for this determination with EXTDEST.

The basic EVLA calibration is much like that described in detail in Chapter 4 except that bandpass calibration is now *required* rather than merely *recommended*. Having chosen those channels which may be reliably used to normalize the bandpass functions,

> DEFAULT BPASS ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> DOCAL 1 C_R	to apply the delay calibration — *very important*.
> CALSOUR '*bandpass_cal*' C_R	to select the strong bandpass calibrator.
> SOLINT 0 C_R	to find a bandpass solution for each scan on the BP calibrator.
> ICHANSEL $c11, c12, 1, if1, c21, c22, 1, if2, c31, c32, 1, if3, \ldots$ C_R	
	to select the range(s) of channels which are reliable for averaging in each IF. Use the central 30% of the channels if your calibrators are all very strong or more like 90% if they are not. Remember these values — you will use them again.
> BPASSPRM(5) 1 ; BPASSPRM(10) 3 C_R	to normalize the results only after the solution is found using the channels selected by ICHANSEL. Use BPASSP(5)=-1 if your phases are not stable within each scan.
> GO C_R	to make a bandpass (BP) table.

Do not use spectral smoothing at this point unless you want to use the same smoothing forever after. Apply the flag table. A model for the calibrator may be used; see § 4.3.3.1.

BPASS now contains the adverbs SPECINDX and SPECURVE through which the spectral index and its curvature (to higher order than is known for any source) may be entered. For the standard amplitude calibrators 3C286, 3C48, 3C147, and 3C138, these parameters are known and will be provided for you by BPASS. For other sources, you may provide these parameters, but BPASS will fit the fluxes in the SU table for a spectral index (including curvature optionally) if you do not. Note that, if no spectral index correction is applied, the spectral index of the calibration source will be frozen into the target source. Bandwidths on the EVLA are wide enough that this is a serious problem. If you do not know the spectral index of your calibration source, BPASS itself or the new task SOUSP may be used to determine the spectral indices from the SU table. Of course, that means that GETJY must already have been run. Since BPASS must usually be run before

CALIB and hence GETJY, this suggests that one may have to iterate this whole process at least once. SOUSP now offers the option of correcting one or more SN tables after it adjusts the source fluxes for the spectral index it determined. This may reduce the need for further iterations.

Note that the bandpass parameters shown above assume that the phases are essentially constant through each scan of the bandpass calibrator. This may not be true, particularly at higher frequencies. In this case, you have two choices. One is to set BPASSPRM(5) to 0 which will determine the vector average of the channels selected by ICHANSEL at every integration and divide that into the data of that integration. This will remove all continuum phase fluctuations, but runs a risk of introducing a bias in the amplitudes since they do not have Gaussian statistics. BPASSPRM(5) = -1 now applies a phase-only correction on a record-by-record basis. A better procedure, which is rather more complicated, is as follows. Use SPLIT to separate the bandpass calibrator scans into a separate single-source file applying any flags and delay calibration and the like. Then run CALIB on this data set with a short SOLINT to determine and apply a phase-only self-calibration. On the uv data set written out by CALIB, run BPASS using the parameters described in the previous paragraph. Finally, use TACOP to copy the BP table back to the initial data set.

In 31DEC16, there is a new TV graphical editing task called BPEDT. It looks a lot like EDITA, displaying multiple antennas at a time, but the horizontal axis is spectral channel rather than time and the data displayed are the solutions from the BP table. This task is particularly suited to identifying and flagging residual RFI in the bandpass calibrator scans after the first pass(es) of RFLAG. Repeat BPASS if you generate any flags with BPEDT.

You now need to run SETJY with OPTYPE 'CALC' and SOURCES set to point at your primary flux calibration sources. You should load the models for these sources that apply to your data with CALRD; see § 4.3.3.1. Then run CALIB with the model once for each primary flux calibrator:

> DEFAULT CALIB ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> IN2DI $n2$; GET2N $m2$ C_R	to select the model image on disk $n2$ and catalog number $m2$.
> DOCAL 1 ; DOBAND 3 C_R	to apply the delay and bandpass calibration — very important.
> SOLINT 0 ; NMAPS 1 C_R	to compute a solution for each calibration scan and use the source model.
> CALSOUR '*flux_cal*' C_R	to select the primary flux calibrator by whatever form of its name appears in your LISTR output.
> ICHANSEL $c11, c12, 1, if1, c21, c22, 1, if2, c31, c32, 1, if3, \ldots$ C_R	to select the range(s) of channels which are reliable for averaging in each IF. These *must* be the same values that you used in BPASS.
> SNVER 2 C_R	to put all CALIB solutions in solution table 2.
> GO C_R	to find the complex gains for the flux calibrator.

Read the output closely. If solutions fail, examine your data closely for bad things. The primary flux calibrator should work without failure. After you have done each primary flux calibrator for which you have models, run CALIB on the remaining calibration sources:

> CALSOUR '*other_cal1*', '*other_cal2*' C_R	to select the secondary calibrators by whatever names appear in your LISTR output.
> CLR2NAME ; NMAPS 0 C_R	to do no models.
> GO C_R	to find the remaining complex gains.

Again, examine the output messages closely. There may be a few failures but there should not be many in a good data set. The RUN file procedure VLACALIB (see § 4.3.3.1) may be used but it does not offer the ICHANSEL option which may be required by your data. It also does a scalar averaging for the amplitudes. This averaging is now a vector average of the spectral channels followed by a scalar average over time.

Scalar averaging suffers from Ricean bias in the amplitudes and so should be used only when the calibration source is very strong or when the atmospheric phases are very unstable.

At this point it is necessary to calibrate the fluxes of the secondary calibration sources using your SN table:

> TASK 'GETJY' ; INP C_R	to set the task name without changing other adverbs.
> SOURCE CALSOUR C_R	to select the secondary sources by the list of name you just used.
> CALSOUR '*flux_cal*' C_R	to select the primary flux calibrator by whatever form of its name appears in your LISTR output.
> INP C_R	to check the inputs closely; remember to do all times, IFs, etc. with SNVER 2.
> GO C_R	to adjust the gains in the SN table and the fluxes in the SU (source) table.

Look at the messages with care — the fluxes in the various IFs should be consistent and the error bars should be reasonably small ($< 10\%$ at high frequencies, smaller at low frequencies). If not, look at your SN table with SNPLT to see if there are bad solutions. If there are, delete SN table 2, do more flagging with TVFLG or SPFLG, and repeat the process.

Use the interactive TV task EDITA to examine the values in your SN table. There may be bad solutions which will require additional flagging of your calibration data. If there is a significant amount of flagging, you should repeat the calibration process to avoid the influence of bad data on the gains and GETJY results. Rather than edit each antenna/IF tediously, use EDITA or SNPLT to determine allowed ranges of amplitude and phase gains and then use SNFLG with OPTYPE 'A&P' to do the flagging.

Although it may be better to wait until after detailed flagging, you may wish to iterate at this point, determining the spectral indices of your bandpass calibrators with SOUSP and re-doing BPASS, CALIB, and GETJY. If the result is a seriously changed spectral index for your secondary sources you may have to iterate further.

If your calibration source consists of a single spectral line, you may use CALIB to determine the gains in that IF, limited to the appropriate channels. The new task SNP2D will convert this SN table from a valid phase in only one IF into delays and phases in all IFs on the assumption that the observed phase is in fact a delay.

Finally, apply the gain solutions to your calibration table:

> DEFAULT CLCAL ; INP C_R	to clear the adverbs.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> CALCODE '*' C_R	to select all calibration sources.
> SNVER 2; INVERS SNVER C_R	to select your solution table from CALIB. Do *not* include the SN table from FRING a second time!
> GO C_R	to apply SN table 2 to CL table 3, creating CL table 4.

Check the result using POSSM and/or VPLOT.

E.5 Detailed flagging

The calibration you have done to this point has been degraded by RFI which has not yet been flagged. However, you need to do the above in order to bring all spectral channels and all antennas and sources into the same flux scale. Now automatic tasks may be used — and they are needed for the large volumes of data produced by the EVLA.

In 31DEC17, the task FGSPW may be used to flag IFs (aka 'spectral windows") that have been rendered useless by RFI or other problems. It performs a scalar average of the amplitudes in all spectral channels on a per IF per polarization per basdeline per scan basis. Those IFs where amplitudes have overflowed the hardware due to RFI tend to stand out with this statistic. LISTR with OPTYPE='MATX' DPARM=1,1 will also compute and print this statistic to let you determine what are reasonable values and what are not.

A very promising relatively new tool flags RFI on the assumption that it is either quite variable in time or in frequency. This task, called RFLAG, computes the rms over short time intervals in each spectral channel and IF individually and flags the interval whenever the rms exceeds a user-controlled threshold. Optionally, it will also use a sliding median window of user-specified width over the spectral channels to the real and imaginary parts of the visibility separately. Any channel deviating from the median in either part by more than a user-specified amount will also be flagged. If DOPLOT>0, RFLAG will make plots of normal and cumulative histograms and of the mean and rms of the time and spectral computations as a function of channel. It will also make a flag table only if requested (DOFLAG>0). These plots will suggest threshold parameters and allow you to choose values to use. A flag table is made for any value of DOFLAG if no plots are requested (DOPLOT≤ 0).

In detail, RFLAG is run using

> DEFAULT RFLAG ; INP C_R	to clear and review the adverbs.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> SOURCES '*source_1*', '*source_2*', ... C_R	to select sources of similar flux level.
> DOCALIB 1 ; DOBAND 1 C_R	to apply continuum and bandpass calibration.
> STOKES 'FULL' C_R	to examine all polarizations.
> DOPLOT 15 ; DOTV 1 C_R	to examine all kinds of plots on the TV.
> FPARM 3 , x , -1, -1 C_R	to examine spectral rms over 3 time intervals each a bit longer than x seconds. The -1's cause the program to use other adverbs for the cutoffs and to do a spectral solution as well as the time one.
> FPARM(9) = 4.0 ; FPARM(10) = 4. C_R	to set the cutoff values as 4 times the median rms plus deviation found in the spectral plots as a function of IF. The default is 5.
> FUNCTYPE 'LG' C_R	to plot the histograms on a log scale.
> NBOXES 1000 C_R	to use 1000 boxes in the histograms.
> INP C_R	to re-examine the inputs. VPARM will let you control aspects of the plotting.
> GO	to run the program.

This will produce plots and set cutoff levels in adverbs NOISE and SCUTOFF. Another run, with DOPLOT = 0 will apply these cutoffs and create a new flag table. Note that the flux cutoff levels may depend on the source flux, calling for different levels for strong calibrators, weak calibrators, and very weak target sources. Different cutoff levels for STOKES='RRLL' and STOKES = 'RLLR' may also be needed. A strong, resolved target source may require different levels for different UVRANGEs. If so, you will need to break up wide bandwidth data into separate files each containing only one IF so that UVRANGE is applied properly. VBGLU may be used later to put the IFs back together. RFLAG is a new task, so experiment a bit. Note that, if you set DOFLAG=1, the creation of a new flag table will happen after the plots in the same execution of RFLAG. If a channel is found bad at a time in any one polarization, all polarizations are flagged. If you have a significant spectral line signal in your data, use DCHANSEL to have the affected channels ignored throughout RFLAG.

There are a lot of adverbs to RFLAG. FPARM(5) allows you to speed up the spectral part of the flagging by testing more than just the central channel in the sliding median filter. FPARM(6) allows you to expand all flags to adjacent channels. FPARM(7), 8, 11, and 12 control the extending of flags to additional channels, baselines, or antennas if too large a fraction of channels, baselines, or baselines to an antenna are flagged in the basic time and spectral operations. Similar adverbs also occur in the new task REFLG whose job it is to compress the enormous flag tables generated generated by RFLAG. REFLG does not handle flags generated by CLIP, TVFLG, and SPFLG since they vary with polarization. REFLG can extend a flag to all times if too large a fraction of time is flagged for a given channel, baseline, etc. REFLG may not reduce your flag table enough, although it is inexpensive to run and so worth the effort. The application of 10 million flag entries to a data set repetitively is rather expensive. Copying the data, applying the flags once and for all, is the best solution. UVCOP has been the traditional method to do this. However, TYAPL which needs to be run next and must make a new copy of the data has been given the option of applying a large flag table to avoid having to copy the data set twice. Task FGCNT lets you see how much of your data is flagged by any particular flag table.

A new tool which may help identify bad data at this early stage is the task REWAY described in § E.8. It checks the data for spectral windows with particularly low and particularly high rms levels. It must copy the selected data to a new file, so it is not particularly recommended at this point. Run it with no flagging of the output for bad values of the spectral rms. Then plot the weights with VPLOT or ANBPL to look for weights that are seriously abnormal (high or low). Those data may need to be flagged. High weights mean that the data are of abnormally low amplitude, whilst low weights mean that the data are very noisy. REWAY uses robust methods to find the rms and so a few channels of RFI may not cause very low weights, but lots of RFI or receiver failures will make the weights abnormally low. REWAY now displays statistical information to help you assess what weights are "high" and "low".

POSSM may be used again to see if serious RFI remains after RFLAG and it may be appropriate to run TVFLG to look at groups of a small number of spectral channels (or even every channel) on your calibration source. Task FLGIT (§ 8.1) is an older task that attempts to flag RFI that is both channel- and time-dependent in a non-interactive fashion. SPFLG (§ 10.2.2) is labor and time intensive but would be the most reliable method to deal with the problem. FTFLG is like SPFLG except that all baselines are added together in a single plot (per polarization). It probably is not good for much flagging, but will provide a quick and sensitive way to look for widely distributed RFI. CLIP will flag particularly high amplitudes and is capable of running in a normalized fashion dividing by source flux including spectral index (and curvature if desired). (UVPLT also has the DOSCALE option, letting you see the normalized visibilities prior to flagging with CLIP.) However, RFLAG should get the high fluxes in most realistic cases and even has a clip option (FPARM(13)).

The auto-editing task FLAGR is also of some use here. It averages the spectral channels to get an estimate of the mean and rms and uses those numbers evaluated over time and baseline in a variety of algorithms to further flag the data. RFI which is rather wide spectrally and long lived may be found in this way.

E.6 Calibration with the SysPower table

Having done a more careful job with your editing, it is now time to discard with EXTDEST the bandpass (BP) tables and all CL tables after the one written by VLANT. Discard all SN tables, but keep the highest numbered flag (FG) table.

Because of its wide dynamic range, the EVLA does not normalize its output visibilities. To calibrate gains it records the total power when the switched noise tube is on and when it is off. These data, taken in synchronism with the visibilities, are recorded in the SysPower table of the ASDM. The OBIT program BDFIn, available to AIPS users in the new verb BDF2AIPS, reads this table and creates an AIPS SY table. The columns of this table contain POWER DIF ($Gain \times (P_{on} - P_{off})$), POWER SUM ($Gain \times (P_{on} + P_{off})$), and POST GAIN ($Gain$) columns for right and left polarizations with values for each IF.

This table is accessible to AIPS users with a number of tasks. Begin with task PRTSY to view the table statistically over time on a per IF, per antenna basis or to view scan or source median averages of one or more of the SY table parameters. Then to examine its contents in more detail in various ways, use SNPLT with OPTYPEs 'PDIF', 'PSUM', 'PGN', 'PON', 'POFF', 'PSYS', 'PDGN', or 'PSGN'. You could use OPTYPE = 'MULT' to examine more than one of these at one time, comparing any oddities in *e.g.*, Psum and Pdif. Note that PSYS is especially interesting since $P_{sum}/P_{dif}/2 * T_{cal} = T_{sys}$, the system temperature. It should reflect changes in elevation and strength of the observed source, but should be immune to adjustments to the gain of the telescope. It determines data weights in TYAPL while $\sqrt{P_{dif_i} P_{dif_j}}$ divides into the visibilities. You may use EDITA (§ 4.4.2) to edit your uv data on the basis of the contents of the SY table. Editing may be based on Psum, Pdif, Pgain, Tsys, and on the differences between these parameters and a running median of these parameters. One may also edit the SY table itself with SNEDT; the same parameters are available. LISTR can even display the SY table Psum, Pdif, system temperatures, and gain factors with OPTYPE 'GAIN' and DPARM(1) set to 17, 18, 15, or 16, respectively.

More importantly, the SY table can be used to do an initial calibration of the visibility data. Use the display programs to decide if your SY table is fine as is or needs editing. The tasks TYSMO and TYAPL (§ 4.1.1.1) may be used with EVLA data having an SY table. TYSMO flags SY samples on the basis of Pdif, Psum, Pgain, and Tsys and then smooths Psum, Pdif and Pgain to replace the flagged samples and/or reduce the noise. You may want to do this to remove outlying bad points and to reduce the jitter in these measurements. TYSMO even applies a flag table to the SY before its clipping and smoothing operations. Be sure to plot the results to make sure that the task did what you wanted. Then use TYAPL to remove a previously applied SY table (if any) and to apply the SY table you have prepared. The result should be data scaled nearly correctly in Jy and weights in $1/\mathrm{Jy}^2$ in all IFs. The CUTOFF option allows you to use obviously good values from the SY table while passing the data from antennas with poor SY values along unchanged. If the SY values from some polarizations or IFs are bad due to RFI while others are good, you may copy the good values to replace the bad values using task TYCOP. The wide-band, 3-bit mode of the EVLA has serious non-linearities in the Pdif measurements which are not yet fully understood. For such data, use OPTYPE='PGN' to apply only the post-detection gains to the data. This will remove abrupt jumps due to changes in those gains (which will be more common with 3-bit data). Note that TYSMO and TYAPL also require a table of the Tcal values which OBIT provides in an *AIPS* CD table. Amplitude calibrations are not applied to EVLA data weights until they have been made meaningful by TYAPL or REWAY. Set FLAGVER in TYAPL if you want to apply your flag table once and for all.

If you have observations of the Sun, do not use TYAPL; use SYSOL instead. This task does the usual TYAPL operation on non-Solar scans, but, for Solar scans, it must determine the average gain and weight factors on those antennas having solar Tcals and then apply those averages to all antennas not equipped with Solar Tcals. Use a well-edited SY table if possible. This Solar capability was made available in OBIT (version 567) and *AIPS* in mid-June 2017.

Now return to § E.4 to repeat the bandpass and continuum calibrations with correctly scaled data with most of the RFI removed.

E.7 Polarization calibration — RLDLY, PCAL, and RLDIF

You may skip this section unless you have cross-hand polarization data and wish to make use of them. Although there have been major improvements in *AIPS* polarization routines, they still do not correct parallel hand visibilities for polarization leakage. Thus you need to calibrate polarization only if you wish to make images of target source Q and U Stokes parameters. Polarization calibration is discussed extensively in § 4.6; we will discuss changes made because of the wide bandwidths and other aspects of the EVLA.

Frequently, the delay difference between right- and left-hand polarizations must be determined even if FRING was not required for the parallel-hand data. Use POSSM to plot the RL and LR spectra to see of there are significant slopes in phase. If so, use a calibration source with significant polarization, although the

EVLA D terms are often large enough to provide a usable signal in the absence of a real polarized signal. Note that 3C286 is significantly polarized and is likely to be the best source to use for this purpose. Then

> TASK 'RLDLY' ; INP CR	to look at the necessary inputs.
> BCHAN c_1 ; ECHAN c_2 CR	to select channels free of edge effects.
> DOCAL 1 ; GAINUSE 0 CR	to apply the FRING results and all other current calibrations.
> REFANT n_r CR	to select a reference antenna - only baselines to this antenna are used so select carefully. Alternatively, REFANT 0 will loop over all possible (not necessarily good) reference antennas, averaging the result.
> DOIFS j CR	to set the adverb to the value of APARM(5) used in FRING (§ E.2. The IFs are done independently (≤ 0), all together ($= 1$), in halves ($= 2$), or more generally in N groups ($= N$).
> TIMERANG *db* , *hb* , *mb* , *sb* , *de*, *he* , *me* , *se* CR	to specify the beginning day, hour, minute, and second and ending day, hour, minute, and second (wrt REFDATE) of the data to be included. Use an interval not unlike the one you used in FRING.
> INP CR	to check the inputs.
> GO CR	to produce a new SN table with a suitable left polarization delay.

Note that RLDLY now always creates an SN table and may be run with multiple calibrator scans. If there is only one calibrator scan and APARM(2)≤ 0, it will also copy the CL table which was applied to the input data through GAINUSE to a new CL table applying the correction to the L polarization delay. For all other cases, you must apply the added L polarization delays with CLCAL.

It is probably better to determine a continuum solution for source polarization and antenna D terms before doing the lower signal-to-noise spectral solutions. To find an average solution for each IF:

> DEFAULT PCAL ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m CR	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND 3 CR	to apply the delay, complex gain, and bandpass calibration.
> CALSOUR '*pol_cal1*', '*pol_cal2*' CR	to select the polarization calibrator(s) by whatever form of their names appears in your LISTR output. These sources must have I polarization fluxes in the source table.
> ICHANSEL $c11, c12, 1, if1, c21, c22, 1, if2, c31, c32, 1, if3, \ldots$ CR	to select the range(s) of channels which are reliable for averaging in each IF. These probably should be the same values that you used in BPASS.
> DOMODEL -1; SPECTRAL -1 CR	to solve for source polarization in a continuum manner.
> PRTLEV 1 CR	to see the answers and uncertainties on an antenna and IF basis.
> CPARM 0,1 CR	to update the source table with the calibrator source Q and U found.
> INP CR	to review the inputs.
> GO CR	to find the antenna leakage terms and the source Q and U values on an IF-dependent basis.

PCAL will write the antenna leakage terms in the antenna file and the source Q and U terms in the source table (if CPARM(2) > 0). DOMODEL may be set to true only if the model has Q $=$ U $= 0$ since PCAL cannot solve for the right minus left phase difference. If SOLINT=0, PCAL will break up a single scan into multiple intervals, attempting to get a solution even without a wide range of parallactic angles.

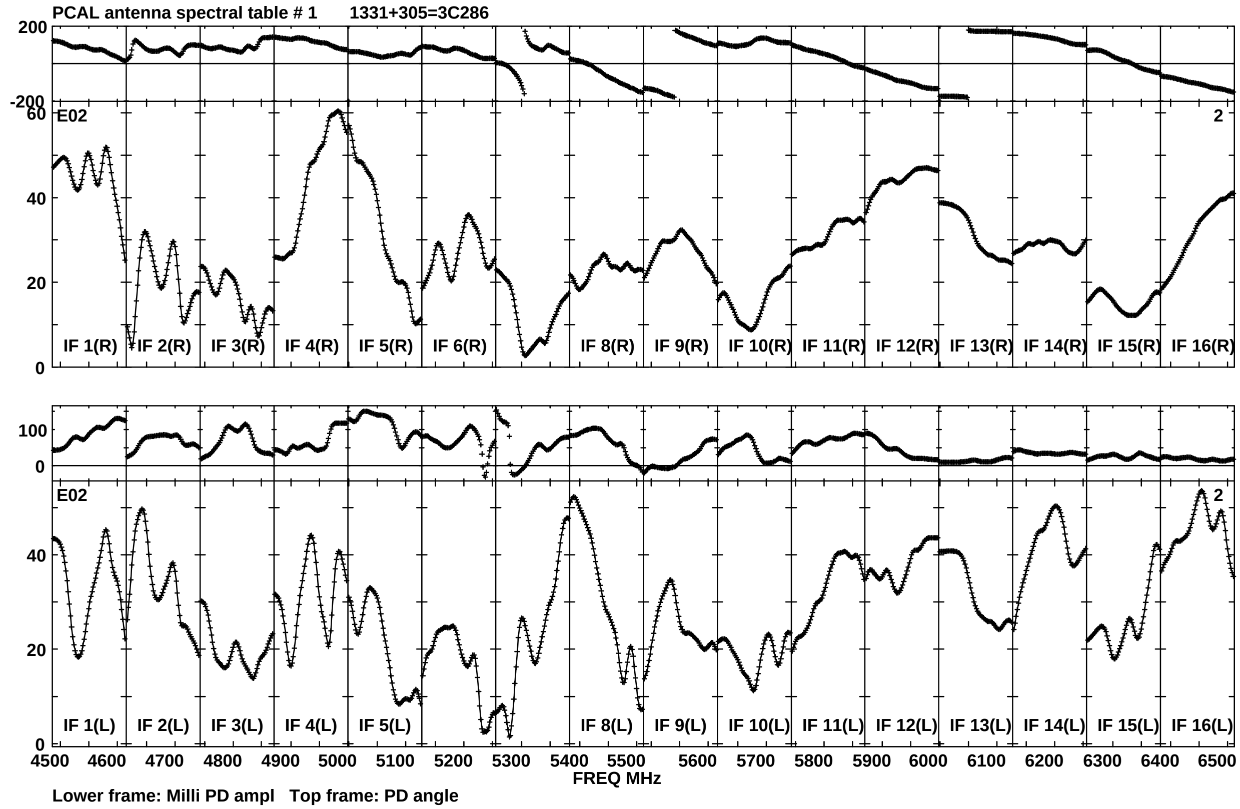

Figure E.1: Example spectrum showing D term solutions for one antenna in right and left polarizations covering about 2 GHz at C band

Having prepared a continuum solution for Q and U, you must also correct it for the difference in phase between R and L polarizations which normally varies considerably between IFs. The task RLDIF will correct the antenna, source, and calibration tables for this difference using observations of a source with known ratio of Q to U. 3C286 is by far the best calibrator for this purpose.

> DEFAULT RLDIF ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND 3 C_R	to apply the delay, complex gain, and bandpass calibration.
> DOPOL 1 C_R	to apply the polarization calibration.
> BCHAN c_1; ECHAN c_2 C_R	to average data from channels c_1 through c_2 only.
> SOURC '*pol_cal1*', '*pol_cal2*' C_R	to select the polarization calibrator(s) by whatever form of their names appears in your LISTR output. These sources must have known polarization angles.
> SPECTRAL 0 C_R	to do the correction in continuum mode.
> DOAPPLY 1 C_R	to apply the solutions to a CL table (making a new modified one) and to the AN and SU tables, updating them in place.
> DOPRINT 0 C_R	to omit all the possible printing.
> INP C_R	to review the inputs.
> GO C_R	to determine and apply the corrections.

The EVLA polarizers appear to be very stable in time, but to have significant variation with frequency. See Figure E.1. Serious polarimetry with the EVLA will *require* solving for the antenna polarization leakage as a function of frequency. To compute a spectral solution, assuming you already did the process in the preceding paragraph:

> TGET PCAL C_R	to retrieve the PCAL adverbs.
> SPECTRAL 1 C_R	to do the channel-dependent mode.
> DOMODEL 0 C_R	to solve for Q and U as a function of frequency. Because PCAL does not solve for a right-left phase difference and that difference is a function of spectral channel, you must solve for a source polarization.
> INTPARM p_1, p_2, p_3 C_R	to smooth the data after all calibration has been done while honoring ICHANSEL.
> SPECPARM 0 C_R	to determine the calibration source I, Q, and U spectral indices from fluxes in the source table. If you use PMODEL you must provide spectral indices for the model that apply in the frequency range of the data (curvature cannot be specified).
> INP C_R	to review the inputs, the task will take a while to run.
> GO C_R	to run the task writing a PD table of spectral leakages ("D terms") and, if DOMODEL ≤ 0, a CP table of source Q and U spectra.

If the combination of flagging, ICHANSEL, and INTPARM results in no solutions for some channels, the solutions from nearby channels will be interpolated or extrapolated so that all channels get solutions.

After running PCAL in spectral mode, you may examine the resulting PD (polarization D terms) table with POSSM using APARM(8)=6 and BPLOT using INEXT = 'PD'. If a CP table (calibrator polarization) was written, you may also use POSSM with APARM(8) = 7 or 8 and BPLOT with INEXT = 'CP' to examine the results.

You are almost, but not quite done. The combination of CALIB and BPASS has produced a good calibration for everything except the phase difference between right and left polarizations. This is now a function of spectral channel and needs to be corrected. The task RLDIF has been modified to determine a continuum or spectral right minus left phase difference and to modify the CL or BP table, respectively, to apply a phase change to the left polarization on an IF or channel, respectively, basis. Thus

> DEFAULT RLDIF ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND 3 C_R	to apply the delay, complex gain, and bandpass calibration.
> DOPOL 1 C_R	to apply the polarization calibration, spectral if present.
> INTPARM p_1, p_2, p_3 C_R	to smooth the data after all calibration has been done.
> BCHAN c_1; ECHAN c_2 C_R	to use solutions from channels c_1 through c_2 only, extrapolating solutions to channels outside this range.
> SOURC *'pol_cal1', 'pol_cal2'* C_R	to select the polarization calibrator(s) by whatever form of their names appears in your LISTR output. These sources must have known polarization angles.
> POLANGLE p_1, p_2 C_R	to provide the task with the source polarization angle(s) in degrees in source number order. The phase correction will be twice this value minus the observed RL phase. Do not provide values for 3C286, 3C147, 3C48, and 3C138. These are known to RLDIF including rotation measures and other spectral dependence.
> SPECTRAL 1 C_R	to do the correction in spectral mode.
> DOAPPLY 1 C_R	to apply the solutions to a BP table (making a new modified one) and to the PD and CP tables.

> DOPRINT -1 ; OUTPRI '*file_name*' C_R	to write the phase corrections applied to a text file suitable for plotting by PLOTR.
> INP C_R	to review the inputs.
> GO C_R	to determine and apply the corrections.

Use UVPLT, LISTR or POSSM to check that the expected RL and LR phases now appear with all calibrations turned on. Following these steps, you apply the polarization calibration in any task offering DOPOL. A value of 1 will apply the spectral solution if present or the continuum one is there is no PD table. A value of 6 for DOPOL requests the continuum solution despite the presence of a spectral solution. Use POSSM to plot the calibration sources in RL, LR, Q, and U polarization to make sure that all has functioned correctly (these are newly revised tasks).

If you do not have good observations of a polarization angle calibrator, it may be possible to correct the data in some circumstances. See the help files for QUOUT and QUFIX for possible methods.

If you think that the right minus left phase of your reference antenna was not stable over time, but you have produced good model images of Q and U ignoring that, then it may be possible to improve the polarization calibration with the task RLCAL. It does a self-calibration of Q and U to find a time-variable right minus left phase to apply to the data.

E.8 Target source data — edit and SPLIT

At this point, your calibration should be finished. You should now do an initial editing on the target sources, much like that done above for the calibration sources. Run RFLAG on the target source(s) and perhaps use:

> DEFAULT TVFLG ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND 3 C_R	to apply the delay, complex gain, and bandpass calibration.
> BCHAN $c1$; ECHAN $c2$ C_R	to average across a range of channels — not as flexible as ICHANSEL but probably okay here.
> BIF j ; EIF BIF C_R	to edit one IF only, which will suffice for problems that are not IF dependent, such as drop outs, antenna not on source, etc. Choose an IF that is reasonably free of RFI.
> NCHAV ECHAN-BCHAN+1 C_R	to average all the channels into one number.
> CALCODE '-CAL' C_R	to do just target sources now.
> DPARM(6) $\Delta\ t$ C_R	to do no time averaging in the work file set $\Delta\ t$ to the data interval in seconds.
> GO C_R	to start the task.

Again, remember to set it to flag all channels and IFs. You may have to select sub-windows and force the averaging to one times $\Delta\ t$ to edit in detail, or perhaps the default time averaging will be beneficial. In general, the DISPLAY AMP V DIFF is a powerful way to catch bad amplitudes and phases. It will catch drop outs either as bright lines for strong sources or dark grey ones for weak sources.

Since EVLA data sets tend to be large and unwieldy, it is recommended that you separate the data into the separate target sources, applying the current calibration and flagging once and for all. The imaging task IMAGR can do this on the fly, but, especially for observations of spectral-line sources, this is excessively expensive.

> DEFAULT SPLIT ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.

> DOCAL 1 ; DOBAND 3 C_R	to apply the delay, complex gain, and bandpass calibration.
> CALCODE '-CAL' C_R	to do just target sources now.
> GO C_R	to write out separate calibrated data sets for each target source.

A task new to `31DEC13` will help calibrate data sets which contain a strong, *e.g.*, maser, line in one or a few channels with interesting but weaker signals in other channels and/or the continuum. After applying the best standard calibration described above, you can split out the maser channel(s) with `UVCOP`. Make images and self-calibrate following the processes described in Chapter 5 making an `SN` table containing the full calibration needed to apply to the maser. Then use task `SNP2D` to convert this into delays and phases to apply to the full, multi-IF data set. This will work so long as the residual phases found in the self-cal are small and is actually required to do the best calibration possible over very wide bandwidths.

Unless `TYAPL` has been used, EVLA data sets have weights which only reflect the integration time in seconds. Calibration routines do not change these weights when changing the data amplitudes. There is a task called `REWAY` which computes a robust rms over spectral channels within each IF and polarization. It can simply base the weights on these on a record-by-record, baseline-by-baseline basis. Alternatively, it can use a scrolling buffer in time so that the robust rms includes data for a user-specified number of records surrounding the current one. A third choice is to average the single-time rmses over a time range and then convert them to antenna-based rmses. In all three modes, the task can then smooth the rmses over time applying clipping based on user adverbs and the mean and variance found in the rmses. A flag table (extension file) may be written to the input data file removing those data found to have rmses that are either too high or too low. For these weights to be meaningful, the bandpass and spectral polarization calibration must be applied and it helps to omit any RFI or other real spectral-line signal channels from the rms computation. For the weights to be correctly calibrated, all amplitude calibration must also be applied. For these reasons, `REWAY` might well be used instead of `SPLIT` — when `TYAPL` was not used — running it one source at a time. Thus,

> DEFAULT REWAY ; INP	to reset all adverbs and choose the task.
> INDI n; GETN m C_R	to select the data set on disk n and catalog number m.
> DOCAL 1 ; DOBAND 3 C_R	to apply the delay, complex gain, and bandpass calibration.
> SOURCE '$target_1$' , ' ' C_R	to do one target source.
> APARM 11, 30, 12, 0, 10, 4 C_R	to use a rolling buffer of 11 times separated by no more than 30 seconds and then smoothed further with a Gaussian 12 seconds in FWHM. Data are flagged if the rms is more than 4 times the variance away from the mean averaged over all baselines, IFs, and polarizations. Flagging on the variance of the rms from the mean on a baseline basis is essentially turned off by the 10.
> GO C_R	to write out a calibrated, weighted data set for the first target source.

Then, when that finishes

> SOURCE '$target_2$' , ' ' ; GO C_R	to do another target source.

It is not clear that this algorithm is optimal, but it certainly should be better than using all weights 1.0 throughout. It will be interesting to compare data weights found with `TYAPL` to those found with `REWAY`. The task now displays statistical information to assist the user in determining which weights are excessively high or low.

E.9 Spectral-line imaging hints

Some spectral-line data sets contain only a few spectral windows, each with a great many spectral channels. The continuum subtraction task UVLSF requires that the continuum signal be a linear function of channel across each spectral window. Converting the data to more spectral windows with task MORIF or REIFS may make the data come closer to this requirement. Having more spectral windows with fewer channels may also speed up imaging in the LINIMAGE procedure mentioned below.

In many spectral-line observations you will want to separate the continuum signal from the channel-dependent signals. This is discussed in some detail in § 8.3. The larger number of channels from the EVLA does mean that continuum may be estimated with greater accuracy than when there were rather few channels which were both free of edge effects and spectral-line signal. The wider total bandwidth may, however, invalidate the assumption that the continuum signal at each visibility point can be represented by a polynomial of zero or first order. If there is a single dominant continuum source offset from the phase center, the assumption may be rendered valid by shifting the data with UVLSF to center the continuum source temporarily in order to subtract it. To examine this assumption and to determine which channels appear safe to use as "continuum" channels, use POSSM.

> DEFAULT POSSM ; INP	to set the task name and clear the adverbs.
> INDI Tn; GETN Tm C_R	to select the calibrated line data set on disk Tn and catalog number Tm.
> DOTV 1 ; NPLOTS 1 CR	to plot only on the TV, one baseline at a time.
> ANTEN $n1$, $n2$, $n3$, $n4$ CR	to select the antennas nearest the center of the array
> BASELINE ANTEN CR	and only them.
> BIF j ; EIF BIF CR	to plot one IF at a time.
> APARM 0 CR	to display vector averaged spectra. Scalar averaged spectra will turn up at the edges reflecting the decreased signal to noise in the outer channels which will assist in determining channels that should be omitted.
> GO CR	to run the task. Make notes of the desirable channels IF by IF.

Note also whether the continuum appears to be a linear function of channel. If so, then use UVLSF to fit the continuum signal, writing a continuum only and a spectral-line only data set:

> DEFAULT UVLSF ; INP	to set the task name and clear the adverbs.
> INDI Tn; GETN Tm CR	to select the calibrated line data set on disk Tn and catalog number Tm.
> ICHANSEL $c11, c12, 1, if1, c21, c22, 1, if2, c31, c32, 1, if3, \ldots$ CR	to select the range(s) of channels which are reliable for fitting the continuum. For a multi-IF data set, you will need to select the channel ranges carefully by IF.
> ORDER 1 CR	to select fitting the continuum in real and imaginary parts with a first order polynomial in channel number. UVLSF offers orders up to four, but they are not for the faint at heart and will give bad results if there are large ranges of channels left out of the fit due to line signals.
> DOOUTPUT 1 CR	to have the continuum which was fit written as a separate data set. This may be used to image the continuum.
> SHIFT $\Delta x, \Delta y$ CR	to shift the phase center to the dominant continuum source temporarily for the fitting.
> GO CR	to run the task.

Imaging the continuum output may, in addition to any scientific value of the continuum image, provide additional flagging and even self-calibration information which may be applied to the line data.

If UVLSF cannot be used, flag the channels at the edges and those with spectral signals using UVFLG. Construct a continuum image with IMAGR on this flagged, spectral-line data set. Note that you might want to reduce the size of the data set with time averaging (UVAVG) and/or channel averaging (SPLIT or AVSPC) before beginning the imaging. Imaging is discussed in detail in § 5.2 through § 5.3.6 and will not be discussed here. You may find that additional editing is needed and that iterative self-calibration is of use. Be sure to copy those flags (but not the edge and spectral-signal flags) and final SN table back to the line data set. Apply them with SPLIT and then subtract the final continuum model with UVSUB. It you have had to use the spectral index options of IMAGR, you may do the proper subtraction including these options with OOSUB rather than UVSUB.

At present, the EVLA observing setup allows you to select the initial frequency of observation based on a desired LSRK velocity in the central channel. From there, however, the observations are conducted at a fixed frequency. Furthermore, the information about rest frequencies, source velocities, and even more fundaemtal parameters such as reference frame (LSRK or barycentric) and type of velocity (radio or optical) are lost. SETJY allows you to correct this. First use SETJY to set the desired rest frequencies (note that they are allowed to be a function of IF) and the VELTYP and VELDEF. Then use OPTYEP='VCAL' over all sources and IFs. This will compute the velocities at which you observed for the first time you observed each source and enter the values in the source table.

Having done this, the task CVEL may be used to shift the visibility data to correct for the rotation of the Earth about its axis as well as the motion of the Earth about the Solar System barycenter and the motion of the barycenter with respect to the Local kinetic Standard of Rest. CVEL works on multi-source as well as single-source data sets. It applies any flagging and bandpass calibration to the data before shifting the velocity (which it does by a carefully correct Fourier transform method). Note, the use of Fourier-transforms means that one *must not* use CVEL on data with channel separations comparable to the widths of some of the spectral features. Furthermore, narrow EVLA bands apparently have sharp cutoffs at the edges which cause any continuum signal to generate sine waves in amplitude after the FFT. Therefore, UVLSF *must be run before* CVEL. The velocity information used by CVEL must be correct. Use LISTR and SETJY to insure this before using SPLIT and CVEL. A special version of CVEL has been written to correct not only for the Earth's motion but also for planetary motion to observe a line at rest with respect to a planet; see PCVEL.

Spectral-line imaging of EVLA data will resemble that for the old VLA except for the increased number of spectral channels and the consequent increase in the data set size. Since IMAGR must read the full data set to select the data for the next channel to be imaged, it is important that the data set be small enough to fit in computer memory if at all possible. OSRO data sets may not need this operation and skipping it may simplify any continuum imaging that you wish to do. However, separating the IFs into separate files will not interfere with spectral imaging and will help with the data set size problem:

> DEFAULT UVCOP ; INP	to reset all adverbs and choose the task.
> INDI Tn; GETN Tm C_R	to select the calibrated target data set on disk Tn and catalog number Tm.
> DOWAIT 1 C_R	to have the task resume AIPS only after it has finished.
> OUTSEQ 0 ; OUTDISK INDISK C_R	to avoid file name issues and select the output disk.
> FOR BIF = 1 TO N; EIF = BIF ; GO; END C_R	to make separate files of each of the N IFs.
> DOWAIT -1 C_R	to turn off waiting.

Doing this UVCOP step on large data sets will be worth any extra trouble it may cause. A RUN file and procedure called LINIMAGE has been written to assist in this process. It even runs FLATN and reassembles the images from the separate IFs into cubes with MCUBE or FQUBE if necessary. Note that you could perform the separation into separate IFs before UVLSF which will speed up POSSM and UVLSF. However, the continuum output would then have to be assembled using VBGLU, which is why the steps above were shown in the present order.

Spectral-line imaging is discussed in § 8.4 as well as throughout Chapter 5. With large numbers of spectral channels, you may wish to have `IMAGR` find appropriate Clean boxes for you. Set `IM2PARM(1)` through `IM2PARM(6)` cautiously. `IM2PARM(7)` controls whether the boxes of channel n are passed on to channel $n+1$. The default does not pass the boxes along when auto-boxing which is probably the correct decision. The end result of the imaging will be one image "cube" for each IF since each IF has to be imaged separately even with a multi-IF input data set. (If you set `BIF = 1; EIF = 0` and try to image channel 103, you will actually image the average of channel 103 from each of the IFs.) To put the individual cubes together into one large cube, use `MCUBE` (§ 8.5.1).

The wide bandwidths of the EVLA have revealed an error in the old code. `IMAGR` and `MCUBE` now control the units of data cubes carefully, making sure that each plane is in units of Jy per *header* beam. The actual restoring beamwidths used are now maintained in a `CG` table. This allows the best resolution to be used in each plane and the approximate match between the units of the residual image and the restored components to be maintained, while still returning correct fluxes when integrating brightness over area. `CONVL` with `OPCODE = 'GAUS'` will now use the `CG` table to find the exact Gaussian needed to produce a constant resolution in each plane. Use of `BMAJ = 0` in `IMAGR` followed by `CONVL` is now the best way to insure a constant resolution with correct image units throughout.

E.10 Continuum imaging hints

The first problem that continuum observers will notice with their EVLA data is that the spectral and time resolution of the data, by default anyway, will be rather more than their science requires. It will be possible to instruct the software which extracts data from the archive to do some averaging in both frequency and time. However, detailed editing for RFI and other issues may require excellent resolution in both these domains. After the data have been edited, you can average data in both domains so long as you are careful not to average so much that you produce radial (bandwidth) and/or transverse (time) smearing within the image area. Note that the increased sensitivity of the EVLA will increase the area over which non-negligible astronomical objects may be found while the wide bandwidth will mean that lowest frequency part of your band will be sensitive, because of its larger primary beam, to a much larger area on the sky than the highest frequency part. The spectral averaging can be done with `SPLIT`; use `APARM(1)=1` and set `NCHAV`, `CHINC`, and perhaps `SMOOTH` appropriately. Similarly, `AVSPC` can be used with `AVOPTION='SUBS'`, setting `CHANNEL` and `SMOOTH` suitably. You will almost certainly wish to retain some spectral separation, so do not use the "channel 0" option.

Time averaging should be done with `UVAVG`:

> DEFAULT UVAVG ; INP	to reset all adverbs and choose the task.
> INDI Sn; GETN Sm C_R	to select the calibrated target data set on disk Sn and catalog number Sm.
> YINC Δt C_R	to average to Δt seconds.
> OPCODE 'TIME' C_R	to have all samples in a given interval written with the same time; this helps with self-cal. Other options are available.
> GO C_R	to produce the averaged data set.

`UBAVG` will do a more aggressive averaging, using baseline-dependent time intervals appropriate for the desired field of view. Do not use `UBAVG` if you are planning to use self-calibration since it destroys the time regularity in the data on which `CALIB` depends. `IMAGR` may now do this extra averaging for you on the fly to reduce the size of the work file it uses. Set `IM2PARM(11)` and `(12)`.

Imaging of the continuum is discussed at great length in Chapter 5 and those details will not be repeated here. Bandwidth-synthesis imaging, which will be the only form of continuum imaging with the EVLA, will make certain adverbs more important. Set `BCHAN` and `ECHAN` to avoid the noisier edge channels. Set `NCHAV`

= ECHAN - BCHAN + 1 and CHINC = NCHAV. This will then image all of your IFs and spectral channels into a single image, positioning each channel correctly in the uv plane. With the EVLA, you will be imaging a wider field of view than you did with the VLA. Use SETFC with IMSIZE 0 ; CELLSIZE 0 to see if you should image with a single facet or with multiple facets. If using multiple facets and trying for significant dynamic range, start imaging with OVERLAP 2 ; ONEBEAM -1, but consider OVRSWTCH = -0.05 or so to switch into faster methods of Cleaning when the dynamic range in the residual is small enough.

It has been widely noted that the noise in the outer channels of each IF (spectral window) is higher than in the more central channels. You may wish to experiment with down-weighting channels by a function of the bandpass correction amplitude. In 31DEC13, task BPWGT offers several options for doing just this, using WEIGHTIT to control what exponent of the bandpass amplitude is used in the correction. A solution, which should be better, is availble in 31DEC14 with task BPWAY. This task operates on ether a scan-by-scan or a source-by-source basis to evaluate the rms as a function of time on each channel individually. It determines the rms on short intervals, like RFLAG, normalizes them over each spectral window, and then smooths the normalized rmses over time functions which can be quite long. The results are then applied to the data to modify the weights on a channel-by-channel, baseline-by-baseline basis. This process should allow data from the edges of the spectral windows to be used appropriately, especially in bandwidth synthesis. Note however, that channel 1 in each window includes signal from both that channel and from channel $N_{\rm chan}+1$ (due to aliasing) and so is probably best ignored.

IMAGR allows you to request automatic finding of the Clean boxes (IM2PARM of 1 through 6). In cases with low sidelobes, this works rather well, but you should probably keep an eye on what it does with DOTV 1 in any case. IM2PARM(12) controls the baseline-dependent time averaging while specifying the maximum field of view you expect. This allows you to reduce the size of the work file considerably which will at least reduce the time required for many of the steps in the imaging proportionally. It may be rather better than that if the work file is very large otherwise, requiring actual reading of the disk every time the data are accessed. Note, however, that the uniform weighting of your data will be affected. This averaging reduces the number of samples at short spacings disproportionally and so appears to reduce their weight in the imaging. Some UVTAPER could be used to compensate for this.

By default, bandwidth synthesis imaging assumes that the primary beam and all continuum sources are the same at every frequency. In fact, the primary beam size varies linearly with frequency (to first order anyway) and sources have spectral index. IMAGR will allow you to compensate for the average spectral index at almost no cost with IMAGRPRM(2). A far more accurate and expensive correction for spectral index may be made if you do the following. First image each spectral channel (or group of closely-spaced channels) separately. Combine them into a cube with FQUBE, transpose the cube with TRANS, and solve for spectral index images with SPIXR. To use these images, set IMAGRPRM(17) to a radius (> 0) in pixels of a smoothing area and put the image name parameters in the 3rd and 4th input image names. Note that this algorithm is expensive, but that it can be sped up with judicious use of the FQTOL parameter. The change of primary beam with frequency may be corrected by setting IMAGRPRM(1) = 25 for the diameter of the EVLA dishes. Note that this algorithm is expensive, but that it can be sped up with judicious use of the FQTOL parameter. These two corrections work together, so that doing both costs very little more than doing just one of them.

A RUN file and procedure named OOCAL has been written to do self-calibration using the spectral-index image(s) and primary-beam correction in the manner used in IMAGR. The procedure runs OOSUB to produce a divided data set, then CALIB to solve for complex gains, and finally TACOP to move the SN table to the real input data set. If your model is a large image rather than Clean components, the procedure IMSCAL in this RUN file will convert the model image into faceted Clean components and then run OOCAL. Similar procedures for FRING, called OOFRING and IMFRING are also available.

If you are observing a strong source and trying for very high dynamic range, you may have to correct for errors that are baseline- rather than antenna-dependent. One source of these errors is the antenna polarization leakage which affects the parallel-hand visibilities in a non-closing fashion. Task BLCAL can be used after you have as good an image as you can get without it. This task will divide the data by the model and average over a user-specified time to find baseline-dependent corrections which may then be applied to the data by setting adverb BLVER. We recommend that you average the divided data over all of the times in your data to get a single correction for each baseline (and IF and polarization). If you use shorter intervals, you run the risk of forcing your data to look too much like your model. Since the polarization leakage is probably a function of frequency, an experimental version of BLCAL called BLCHN has been released. It determines the same correction but does not average over channels. BLCHN can even adjust the resulting correction table to correct for spectral index prior to applying and storing the values. The correction is saved in a table which POSSM and BPLOT are able to display. However, the calibration routines do not know how to apply this table, so BLCHN writes out the corrected data as well as the table. A new task called BDAPL has been written to apply this BD table to another data set.

31dEC16 contains a new TV-menu driven editing task called UFLAG.[2] If your image shows stripes, indicative of residual bad data, this task allows you to view the data as it is gridded for imaging and to flag bad cells in the uv plane or even some, but not all, of the individual visibilities contributing to apparently bad cells.

E.11 Additional recipes

E.11.1 Columbian fresh banana cake with sea foam frosting

1. Open an 18.5 oz **yellow cake mix** into a large mixxing bowl; do not use a mix that contains pudding or requires oil. Combine with 1/8 teaspoon **baking soda**.
2. Stir 3/4 cup **Coca-Cola** briskly until foaming stops. Add to cake mix and blend until just moistened. Then beat at high speed for 3 minutes, scraping the bowl often.
3. Combine 2 teaspoons **lemon juice** with 1 cup mashed **bananas** and then add to batter. Add 1/3 cup finely chopped **nuts** and beat for 1 minute at medium speed.
4. Turn the batter into a well greased, lightly floured 9x13 baking dish. Bake in a preheated 350° F oven for about 40 minutes or until the cake tests done. Cool on a rack for 15 minutes, remove cake from pan and turn right side up on a rack to finish cooling.
5. In the top of a double boiler, combine 2 large **egg whites**, 1 1/2 cups packed **light brown sugar**, 1/8 teaspoon **cream of tartar** (or 1 tablespoon corn syrup), and 1/3 cup **Coca-Cola**. Beat at high speed for 1 minute with an electric mixer.
6. Place over boiling water — the water should not touch the bottom of the top half of the double boiler. Beat on high speed for about 7 minutes until the frosting forms peaks when the mixer is raised. Remove from boiling water. Empty into a large bowl.
7. Add 1 teaspoon **vanilla extract** and continue beating on high speed until thick enough to spread, about 2 minutes. Spread on the sides and top of the cold banana cake.

E.11.2 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.

[2]Greisen, E. W. 2016, "Editing on a uv grid in AIPS" AIPS Memo 121, http://www.aips.nrao.edu/aipsdoc.html

E.11.3 Banana July cocktail

1. Sprinkle 3 sliced **bananas** with 1 tablespoon **lemon juice**.
2. Mix with 1 1/4 cans drained and flaked **tuna**, 1/2 **onion** chopped, and 2 tablespoons chopped **gherkins** or **olives**.
3. Spoon into 7 cocktail shells.
4. Melt 2 tablespoons **butter** in a saucepan. Add 2 tablespoon **cake flour** and salt and pepper to taste.
5. Add 1/4 cup **chicken stock** and 1/4 cup dry **white wine**. Simmer for one minute stirring constantly.
6. Add 1/3 cup grated **cheddar cheese** and allow to cool.
7. Add 1/4 cup fresh cream to sauce and pour over banana-tuna mixture.
8. Sprinkle with 1 tablespoon grated **cheese** and **paprika**. Decorate with a slice of **gherkin** or **olive**.
9. Bake 15–20 minutes at 350° F; serve warm.

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

E.11.4 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.

E.12 Concluding remarks, early science

AIPS itself, and particularly this appendix, do not begin to cover all of the issues that will arise with EVLA data. The increased sensitivity of the EVLA will mean that imaging will no longer be able to ignore effects that are difficult to correct such as pointing errors, beam squint, variable antenna polarization across the field, leakage of polarized signal into the parallel-hand visibilities, etc., etc. These are research topics which may have solutions in *AIPS* or other software packages such as `OBIT` and `CASA` eventually.

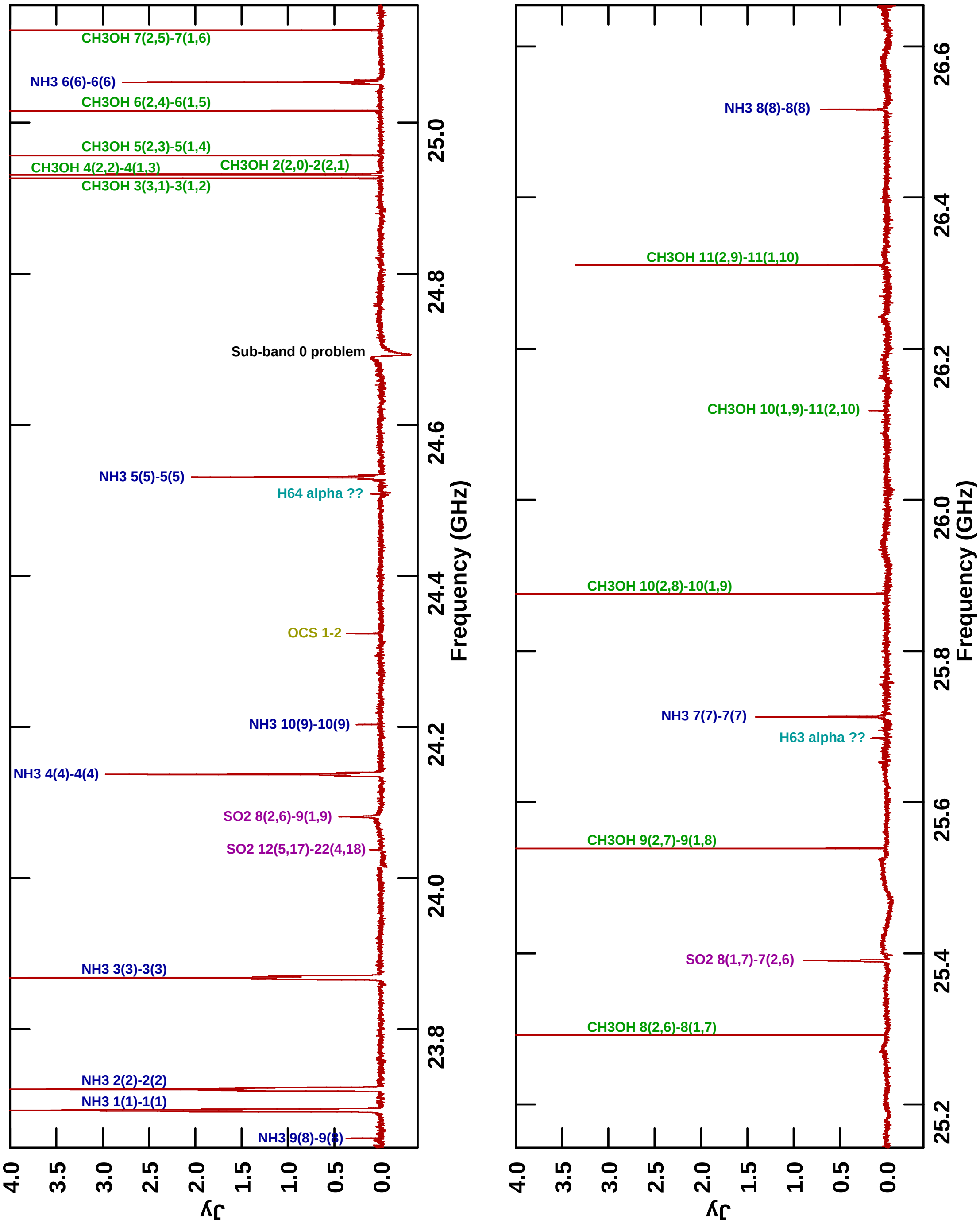

Figure E.2: **Early EVLA science:** The spectrum of the hot core of Orion A at K band. Three separate observations of 8192 channels each 0.125 MHz wide were made using 12 antennas in the D array. Two hours total telescope time went into each of the two lower thirds of the spectrum and 1 hour was used for the highest third. The plot was made using `ISPEC` over a 54 by 60 arc second area. Line identifications provided by Karl Menten.