A Easy Continuum UV-Data Calibration and Imaging

This appendix contains a step-by-step guide to calibrating simple continuum data from interferometers using $\mathcal{AIPS}$. This older guide is now preceded with information about a new "pipeline" of $\mathcal{AIPS}$ procedures which can *automagically* calibrate, image, and even self-calibrate many multi-source continuum (and line) data sets. Although this pipeline is unlikely to produce immediately publishable results in most cases, it does produce a data set with preliminary calibration and editing which can then be edited further "by hand." Following the additional editing, the pipeline may be re-run to produce improved images. When good single-source data sets are produced, self-calibration can also be performed using the methods described in § 5.4.

A.1 VLARUN

The user should read $\mathcal{AIPS}$ Memo 112, "Capabilities of the VLA pipeline in AIPS," by Lorant O. Sjouwerman dated March 19, 2007. This memo goes into details and advice that are beyond the scope of this appendix. The memo may be obtained from the $\mathcal{AIPS}$ web set `https://www.aips.nrao.edu/`.

To use the pipeline, you must first load the VLA Archive or other multi-source visibility data into $\mathcal{AIPS}$; see § O.1.1. To acquire the pipeline, including all procedures and special adverbs, enter

> RUN VLARUN C_R	to define the procedures and adverbs

Do this only once. The procedures will be remembered in your `LASTEXIT SAVE/GET` file.

The pipeline has three stages: (1) calibration and editing, (2) basic imaging, and (3) self-calibration. Study the recommendations and other advice in `EXPLAIN VLARUN` C_R. Note that `VLARUN` can also handle spectral-line data. Then, the inputs for the calibration stage include

> TASK 'VLARUN' ; INP C_R	to review the inputs needed.
> WORKDISK n C_R	to specify the disk containing the uv data.
> CATNUM $catn$ C_R	to specify the catalog number of the uv data; `INNAME` et al. may also be used.
> FASTSW 1 C_R	to correct source name peculiarities induced by fast-switching observations.
> AUTOFLAG 2 C_R	to use `FLAGR` on all data and `QUACK` on all but high frequency data sets
> PHAINT 1 ; AMPINT 5 C_R	to set the phase calibration interval to 1 minute and the amplitude calibration interval to 5 minutes.
> DOMODEL 1 C_R	to use standard flux calibrator models where available.
> NOPAUSE -1 C_R	to have the pipeline pause after `GETJY` to allow you to evaluate whether it is okay to proceed.
> AUTOPLOT -1 C_R	to make no diagnostic plot files.
> DOIMAGES -1 C_R	to do calibration and editing only.
> INP ; TPUT C_R	to double check the inputs and save them.
> VLARUN C_R	to calibrate and edit your data.

The pipeline can do imaging and even self-calibration, although the latter is not for the faint at heart. To include this in your pipeline run, enter

> DOIMAGES 1 CR	to request imaging.
> ARRYSIZE 0 CR	to let the procedure find the array dimensions.
> IMSIZE -1 CR	to image the target source over the full primary beam, with smaller areas for calibrators.
> NITER *nit* ; CUTOFF *f* CR	to set the number of iterations *nit* and Clean flux limit *f*. `NITER` must be $>$ 0 for the image to be Cleaned. Use conservative values to begin with and higher limits in subsequent runs.
> ALLIMG 1 CR	to make images of calibrators as well as target sources.
> SLFCAL 0 CR	to avoid self-calibration initially.
> INP ; TPUT CR	to double check the inputs and save them.
> VLARUN CR	to calibrate, edit, and image your data.

In the first run of `VLARUN`, it is recommended to avoid self-calibration although the making of images will help you evaluate the quality of the auto-editing. If your data are not "perfect" (*i.e.*, good enough), then consider one or more of the interactive editing tasks such as `TVFLG` (§ O.1.6), `SPFLG` (§ 10.2.2), `EDITA` (§ 4.3.11), and `EDITR` (§ 5.5.2) along with the advice given on editing in § 4.3.5, § 4.3.13, § O.1.5, and § 5.5. Then

> TGET VLARUN CR	to restore the inputs.
> AUTOFLAG -1 CR	to do no more automatic flagging.

Consider raising `NITER` and/or lowering `CUTOFF` in this second pass, plus

> SLFCAL *snit* CR	to also do self-calibration with interactivity on the TV ($snit >$ 0) or in a batch-like process ($snit < 0$) for $\lvert snit \rvert$ self-cal iterations.
> INP ; TPUT CR	to double check the inputs and save them.
> VLARUN CR	to calibrate, edit, and image your data.

Repeat as needed. Note that `VLARUN` has been used on many of the data sets in the VLA archive, with the results available via `https://www.vla.nrao.edu/astro/nvas/`. These results include both images and calibrated *uv* data sets.

A.2 Basic calibration

From VLA Archive Tape to a UV FITS Tape
AIPS Memo No. 76 Updated
Glen Langston

The Gentle User enters the Computer room with a VLA archive tape containing a scientific breakthrough. The user's sources are named SOURCE1 and SOURCE2. The interferometer phase is calibrated by observations of CAL1 and CAL2. The flux density scale is calibrated by observing 3C48 (=0137+331) and polarization is calibrated with observations of 3C286 and/or 3C138. Mount the tape on drive number *n*, log in and start *AIPS*. Example input: AIPS NEW. Mount the tape: INTAPE=*n*; DENS=6250; MOUNT.

PRTTP Find out what is on the tape, get project number and bands. TASK='PRTTP'; PRTLEV=-2; NFILES=0; INP; GO; WAIT; REWIND.

FILLM Load your data from tape. Select only one band at a time to process. TASK='FILLM'; VLAOBS='?'; BAND=' '; NFILES=?; DOWEIGHT 1; INP; GO (Replace all ?'s with appropriate values.) FILLM will load your visibilities (*uv*-data) with weights suitable for calibration into a large file for each band. It also creates 6 *AIPS* tables each; these tables have two letter names which are:

HI Human readable history of things done to your data. Use PRTHI to read it.

AN Antenna location and polarization tables. Antenna polarization calibration is placed here.

NX Index into visibility file based source name and observation time. Not modified by calibration.

SU Source table contains the list of sources observed and indexes into the frequency table. The flux densities of the calibration sources are entered into this table.

FQ Frequencies of observation and bandwidth with index into visibility data. Not modified.

CL Calibration table describing the antenna based gains. Version 1 should never be modified. The CL table contains entries at regular time intervals (*i.e.*, 2 minutes) for each antenna. **The ultimate goal of calibration is to create a good CL version 2.** Use PRTAB to read tables.

PRTAN Print out the antenna locations. TASK='PRTAN'; PRTLEV=0; INP; GO. Choose a good *Reference* antenna (called *R*) near the center of the array (REFANT=R). Check the VLA operator log to make sure the antenna was OK during the entire observation.

QUACK Flag the bad points at the beginning of each scan, even the ones with good amplitudes could have bad phases. Creates a Flag Table (FG). You want to use FG table version 1 for all tasks. TASK='QUACK'; FLAGV=1; OPCOD=' '; APARM=0; SOUR=' '; INP; GO deletes the first six seconds of each scan, which may not be enough.

FG A flag table marks bad data. FG tables contain an index into the UV data based on time range, antenna number, frequency and IF number.

LISTR Lists your UV data in a variety of ways. Make a list of your observations. TASK='LISTR'; OPTYP='SCAN'; DOCRT=-1; SOUR=' '; CALC='*'; TIMER=0; INP; GO. NOTE: IF you have observed in a way as to create more than one FREQID, you must run through the entire calibration once for EACH FREQID. For new users, it is better to use UVCOP to copy each FREQID into separate files and calibrate each file separately. This is required if you are doing polarization calibration.

UVCOP Skip this step if your data consists of only one FREQID. Copy different FREQIDs into separate files. TASK='UVCOP'; FREQID=?; CLRON; OUTDI=INDI; INP; GO. The result will be a ??.UVCOP file.

SETJY Sets the flux of your flux calibration source in the SU table. TASK='SETJY'; SOUR='3C48',' '; OPTYP='CALC'; FREQID=1; INP; GO. Adjust flux density for partial resolution following the rules in the VLA Calibration Source Manual or the *AIPS CookBook*.

TASAV As insurance, make a copy of all your tables. TASK='TASAV'; CLRON; OUTDI=INDI; INP; GO.

CALIB CALIB is the heart of the *AIPS* calibration package. RUN VLAPROCS, an *AIPS runfile*, to create procedures VLACALIB, VLACLCAL and VLARESET. The procedure VLACALIB runs CALIB. Set UVRANGE and ANTENNA to zero to allow use of models for 3C48. For L, C and X band 5% and 5 degree errors are OK; for other bands the limits are higher. CALIB places antenna amplitude and phase corrections into an SN table for the time of observation of phase calibration sources.

SN Solution table contains antenna based amplitude and phase corrections for the time of observations of the calibration sources. These SN table results are latter interpolated for all times of observation and placed in a CL table. Only the CL table corrections will be applied to the program sources.

TASK='VLACAL'; CALS='3C48',' '; CALCODE='*'; REFANT=*R*; UVRA=0; SNVER=1; DOCALIB=-1; DOPRINT=1; MINAMP=10; MINPH=10; INP; VLACAL. VLACALIB will load and use a source model for 3C48, making UVRANGE unnecessary. The task CALIB lists antenna pairs which deviate significantly from the solution. If you have lots of errors, then carefully examine your data using TVFLG or LISTR. (See *AIPS CookBook* for a lengthy discussion on flagging.) If one antenna is bad over a limited time range, use UVFLG to flag that antenna for the time from just after the previous good CAL observation to before the next good CAL observation.

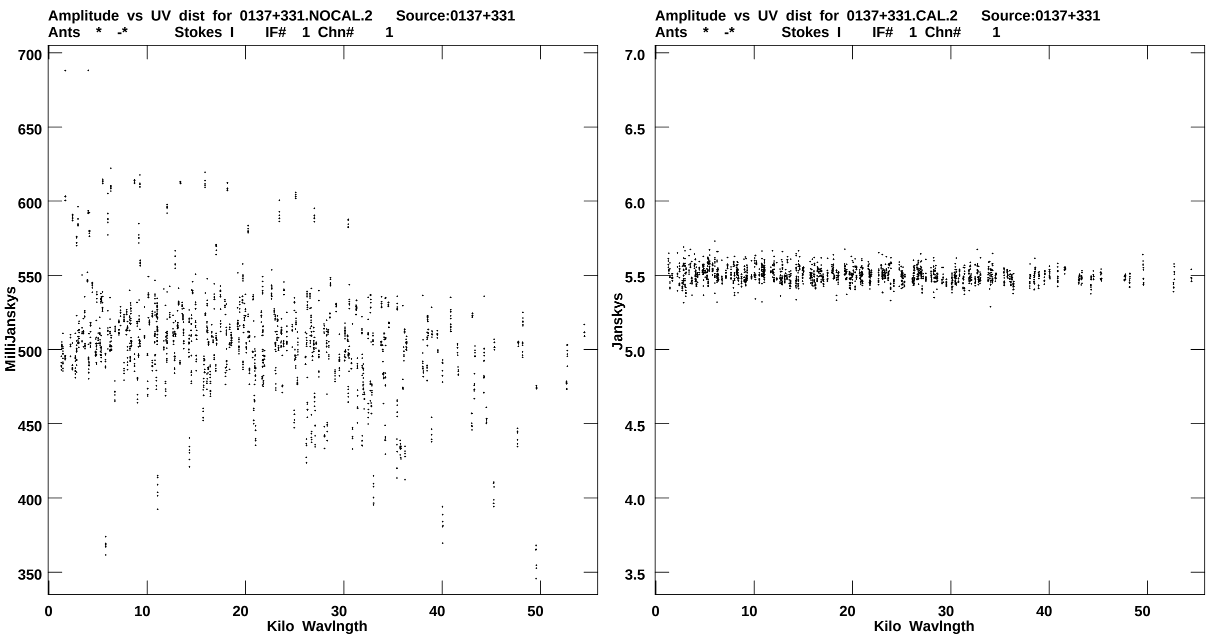

Figure A.1: *(left)* Un-calibrated *uv*-data and *(right)* calibrated *uv*-data from a C-band snapshot of 3C48. Default VLA gains are a tenth of the actual gains and can show significant scatter. Only wild *uv* points ∼50% greater than the average can be detected before calibration.

UVFLG Flag bad UV-data. TASK='UVFLG'; ANTEN=?,0; BASELI=?,0; TIMER=?; OUTFGVER=1; SOUR=' '; OPCOD=' '; INP; GO. If in doubt about any data, FLAG THEM! If you have flagged the primary calibrator, return to CALIB above and try again.

CALIB Now calibrate the antenna gain based on the rest of the cal sources. Look in the Calibrator manual for UV limits; if there are limits, VLACAL must be run separately for these sources. TGET VLACAL; CALS='CAL1','CAL2',' '; ANTEN=0; BASELI=0; UVRANGE=?,?; INP; VLACAL. Flag bad antennas listed. Each execution of CALIB replaces previous corrections in the SN table or appends new corrections. If unsatisfied with a VLACAL execution, all effects of it are removed by running VLACAL again for the same sources (but different ADVERBS or after flagging bad data).

GETJY Sets the flux of phase calibration sources in the SU table. TASK 'GETJY'; SOUR='CAL1,'CAL2',' '; CALS='3C48',' '; BIF=0; EIF=0; INP; GO. GETJY over-writes existing SU table entries, and is not affected by previous executions.

TASAV Good time to save your tables. TGET TASAV; INP; GO.

CLCAL Read the antenna amplitude and phase corrections from the SN table and interpolate the corrections into a new CL table. CLCAL applies calibration source corrections to the program sources. Each execution of CLCAL adds to output CL table version 2. CLCAL is run using the procedure VLACLCAL. TASK='VLACLC'; SOUR='SOURCE1','CAL1',' '; CALS='CAL1',' '; OPCODE='CALI'; TIMER=0; INTERP='2PT'; INP; VLACLC. Run CLCAL for the second source using the second calibrator. TGET VLACLC; SOUR='SOURCE2','CAL2',' '; CALS='CAL2',' '; INP; VLACLC. Move the SN table corrections for 3C48 into the CL table. TGET VLACLC; SOUR='3C48',' ';CALS='3C48',' '; INP; VLACLC. (3C48 could also be calibrated with CAL1 or CAL2.)

LISTR Make a matrix listing of the Amplitude and RMS of calibration sources with calibration applied. Look for wild points. TASK='LISTR'; OPTYP='MATX'; SOUR='CAL1','CAL2',' '; DOCAL=2; DOCRT=-1; DPARM=3,1,0; UVRA=0; ANTEN=0; BASELI=0; BIF=1; INP; GO. If only a few points are bad, flag them and continue. If too many are bad, delete CL table 2 and the SN tables using VLARESET. Then return to the first CALIB step. If the data look good, run LISTR again for IF two. TGET LISTR; BIF=2; INP; GO

UVPLT Plot the *uv*-data in a variety of ways. Make a Flux versus Time plot first. Choose XINC so the plot will have no more than 1000 points. TASK='UVPLT'; SOUR='SOURCE1',' '; XINC=10; BPARM(1)=11; DOCAL=2; BIF=1; INP; GO. Look at the plot with LWPLA, TKPL, TVPL or TXPL. Plot other IF . Flag wild points. Plot Flux versus baseline. TGET UVPLT; BPARM=0; INP; GO.

Calibration is now complete for continuum, un-polarized observations. Write the calibrated data to tape with FITTP if you don't want to calibrate the polarization. To create images from the *uv*-data use SPLIT to calibrate the multi-source data and create a single source *uv*-data set. (FITTP and SPLIT are described at the end of the polarization calibration process.)

A.3 Polarization calibration

For polarization observations, the following steps are required. For 21cm or longer wavelength observations, ionospheric Faraday rotation corrections may be needed. See FARAD in the *AIPS CookBook*, but don't expect much help anymore.

TASAV As added insurance, save your tables again. TGET TASAV; INP; GO.

LISTR Print the parallactic angles of the calibration sources. TGET LISTR; SOUR=' '; CALC='*'; OPTYP='GAIN'; DPARM=9,0; INP; GO

PCAL Intrinsic antenna polarization calculation. PCAL will be successful only if cal. sources are observed at several parallactic angles. PCAL will modify the AN and SU tables. TASK='PCAL'; CALS='CAL1','CAL2',' '; BIF=1; EIF=2; DOCAL=2; REFANT=*R*; INP; GO

RLDIF Now determine the absolute linear polarization angle. Make a matrix listing of the angle of 3C286. TASK 'RLDIF'; SOUR='3C286',' '; DOCAL=2; BIF=1; EIF=0; DOPOL=1; GAINUSE=2; DOCRT=-1; INP; GO. The observed angles are different for each frequency and IF . This task returns the average angles for all IFs in CLCORPRM.

CLCOR Now apply the angle corrections to CL table 2. The relative phase of Left and Right circular polarization produces the linear polarization angle and the phase correction is applied to L. The phase difference (twice the angle of linear polarization) for 3C286 is 66^o and for 3C138, $\phi = -18^o$ at L band, perhaps -24^o at higher frequencies. Change RLDIF's results to this form with FOR I = 1:20; CLCORP(I)=66-CLCORP(I); END. Then TASK='CLCOR'; STOKES='L'; SOUR=' '; OPCOD='POLR'; BIF=1; EIF=2; GAINVER=2; GAINUSE=0; INP; GO. Run RLDIF again to check the phases. TGET RLDIF; INP; GO. Note that CLCOR copies the CL table version 2 to 3 while applying the phase correction. If the phases are wrong, delete version 3, return to PCAL and RLDIF and then do another CLCOR.

A.4 Backup and imaging

FITTP Writes the output *uv*-data to tape. DISMOUNT your archive; MOUNT your output tape. TASK='FITTP'; DOEOT=1; OUTTAP=INTAP; INP; GO. Use DOEOT=-1 when at the beginning of a new tape.

SPLIT The *AIPS* calibration process only modifies the tables associated with the multi-source *uv*-data set. SPLIT selects individual sources, reads the CL table and multiplies the visibilities by the corrections to produce a calibrated single-source *uv*-data set. TASK='SPLIT'; SOUR=' '; CALC=' '; UVRA=0; TIMER=0; DOCAL=2; FLAGVER=1; GAINUSE=3; DOPOL=1; DOBAN=-1; BIF=0; EIF=0; STOKES=' '; BLVER=-1; APARM=0; DOUVCOM=1; ICHANSEL=0; INP; GO

Mapping Use your favorite Fourier Transform task (*e.g.*, IMAGR) to produce images from the calibrated data. Procedure MAPPR provides a simplified interface to IMAGR.

A.5 Additional recipes

A.5.1 Banana colada

1. Peel and slice 1 ripe **banana**.
2. Place sliced banana in blender along with 6 ounces **pineapple juice** (or crushed tinned pineapple in its own juice) and 1 ounce **rum** plus 1 ounce **coconut rum** *or* 2 ounce **rum** plus 1 teaspoon **Coco Lopez**.
3. Optionally add 1 ounce **banana liqueur**.
4. Blend until smooth.
5. Add crushed ice, if so desired.
6. If the mixture is too thick, add more juice (or more rum if you prefer!); if too thin, add more banana. This is a really easy recipe to adjust to one's taste.

A.5.2 Breaded chicken and bananas

1. In food processor, blend 1 can **condensed milk**, 1/3 cup **milk**, 1/2 cup flaked **coconut**, and 1/4 cup **lemon juice** until smooth. Pour into a bowl.
2. Prepare 3 cups **corn flake crumbs** in another bowl or plate.
3. Cut 6 very firm **bananas** lengthwise, dip in milk mixture, roll in corn flakes, and set aside.
4. Cut 2 **chickens** into pieces, dip in milk mixture, roll in corn flakes, and place in greased baking pans (2 13x9 pans may be required).
5. Sprinkle chicken with 1/2 cup melted **butter** and bake as 350° F for one hour.
6. Arrange bananas over the chicken. Sprinkle with 1/4 cup melted **butter**. Bake 15 minutes longer or until chicken juices run clear.
7. Garnish with sliced star and/or kiwi fruits if desired.

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

A.5.3 Banana-pineapple bread

1. Mix together 1 cup chopped **nuts**, 2-1/2 cups **sugar**, 5 cups **flour**, 1 teaspoon **salt**, 1 teaspoon **baking powder**, and 1 teaspoon **cinnamon**.
2. Mix together 1-1/2 cups **vegetable oil**, 3 **eggs**, 3 mashed **bananas**, 1 teaspoon **lemon juice**, and 1 can **crushed pineapple** (drained).
3. Combine. Bake at 350° F for one hour.

Thanks to Tim D. Culey, Baton Rouge, La. (`tsculey@bigfoot.com`).