Long Wavelength Array Tutorial

May 30, 2024

1 About the Long Wavelength Array

The Long Wavelength Array or LWA is a low-frequency radio interferometer operating between 3 and 88 MHz [3]. It consists of multiple independent stations that can be combined together through interferometry to make what is called the "LWA Swarm," which is the largest radio interferometer in this frequency range in North America. As of this writing it consists of three stations currently operating, two of them with 256 dipoles located in New Mexico (located at Sevilleta National Wildlife Refuge and the center of the Very Large Array) and one in Owens Valley, California that has recently been upgraded to 352 dipoles. An additional mini-station with 64 dipoles is constructed and undergoing commissioning activities at the end of the north arm of the Very Large Array (VLA). Two more mini-stations, one in Comanche Springs, Texas and one located in Arizona, are currently in the planning stages and expected to start construction soon.

1.1 LWA Data Products

The Long Wavelength Array has different observing modes that it operates in. Some of these modes are specific to a particular station design. Each mode is designed to

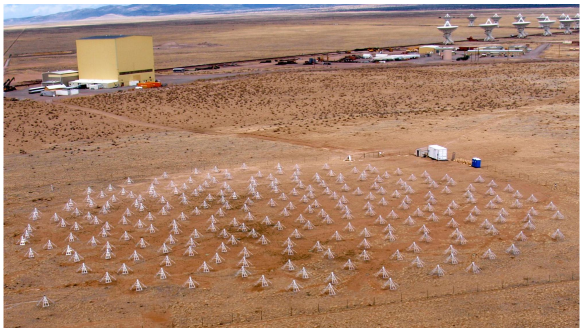


Figure 1: The first LWA station located at the Very Large Array in New Mexico

deliver a data product for certain kinds of science cases. There are various modes that can be used depending on the science goals. The modes applicable to this tutorial are the digital reciever (DRX) raw data and spectrometer modes and the Orville Wideband images made using the Transient Buffer Frequency-domain (TBF). The TBF is a mode present on the second LWA station and future stations that provides 20 MHz of bandwidth in 1-2 s dumps every 3 minutes. All stations have the DRX beamforming modes with up to 20 MHz bandwidth and two tunings. Details of the number of beams and flexibility of the tunings vary according to station.

2 LWA Tutorials

These tutorials are intended to demonstrate a selection of the kinds of science that can be done with the LWA and how to observe with the LWA. It is by no means comprehensive, notably omitted is a demonstration of lightning observations or of LWA swarm observations. What is present is a variety of tutorials involving the kinds of data products that can be generated from a single LWA station and how to generate an observation session.

These tutorials depend on the following software:

- Apptainer, formerly known as singularity, (https://apptainer.org/)
- The tutorial data found at https://lda10g.alliance.unm.edu/tutorial
- The tutorial container found at:
`https://lda10g.alliance.unm.edu/tutorial/lwa_tutorial.sif`

To start the container run:

```
apptainer shell lwa_tutorial.sif
```

Then navigate to the folder where you have placed the tutorial data. If you are using Windows, you will need to run this within Windows Subsystem for Linux, which is also required for installing apptainer.

2.1 Observing with the LWA

Space required: None

1. Start the LWA observation scheduling GUI that is part of the SessionSchedules extension with:

   ```
   sessionGUI.py --lwasv
   ```

   This will bring up a window that will help in preparing the session definition files (SDFs) used for scheduling observations at LWA-SV. Running this command without `--lwasv` will bring up the settings for generating an observation at LWA1

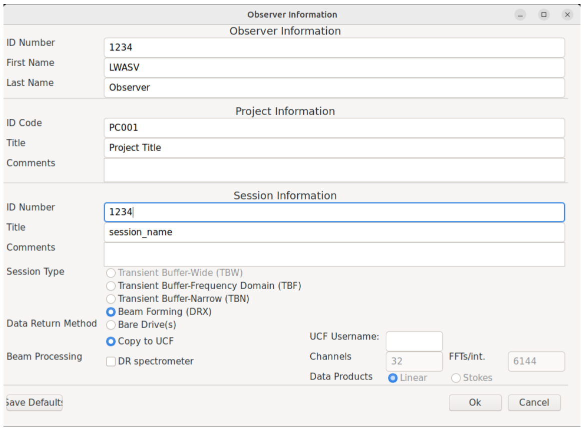
2. Define a new set of observations using the `File -> New` menu or by clicking the "New Project" icon . This will open up the observer and project information window shown in Figure 2. In this window you can define your name, the project, the LWA-SV mode to be used to collect the data, whether or not to use the data recorder spectrometer, and how the data are to be returned to you. The nomenclature is that a project is made of one or more sessions. Each session uses one of the LWA-SV outputs (two DRX beams and TBF or TBN) and consists of multiple sequential observations. Note that TBN may be deprecated in the near future. For DRX observations you can choose to observe a static position or you may track an RA/Dec. position, the Sun, or Jupiter. It is a good idea to check the current list of observing constraints posted at http://www.phys.unm.edu/ lwa/astro/currentissues.html to make sure that there are no issues that would interfere with the observations. For the observations of B0329+54 you will want to select the DR spectrometer mode. For the purpose of this tutorial you should select "Bare Drives(s)," however for a real observation you will likely want to choose "Copy to UCF" instead. The default values for the number of channels is too small and the integration time, the number of FFT windows per integration, is too long to resolve the pulse. You should specify 1024 channels of this observation and reduce the number of FFT windows per integration down to a value that is equivalent to about 40 ms. The currently supported modes for the data recorder spectrometer can be found at http://www.phys.unm.edu/ lwa/astro/scheds/spec.html.

3. After defining the observer information, create an observation to look at the pulsar B0329+54. This can be done through the `Observations -> Add -> DRX - RA/Dec` menu or the "New RA/Dec Target" icon R.This will add a new blank observation to the window. To find the coordinates of B0329+54, enter the source "PSR B0329+54" into the target field, select the observation via the check box on the left side of the screen, and choose `Observations -> Resolve Selected`.

4. We will now check the visibility of the source. First, change the duration to 24 hours. Then select `Observations -> Session at a Glance`. This will plot the elevation of the source throughout the observation.

5. Define the desired parameters of the observation (observing time, tuning, bandwidth, etc.). For the DRX mode each pointing has two independent tunings that can be tuned anywhere in the 10 to 88 MHz band of LWA-SV. To get the list of filter codes, use the `Help -> Filter Codes` dialog. For pulsar observations it is good to space the tunings by the bandwidth so that the two tunings can be combined together and jointly analyzed, e.g., if you are using filter #7, space the tunings by 19.6 MHz.

6. Check the data volume of your observations using the `Data -> Estimated Data Volume` feature. This will display a window with the raw data volume for each of the observations as well as the session total.



Observer Information

Observer Information

ID Number: 1234
First Name: LWASV
Last Name: Observer

Project Information

ID Code: PC001
Title: Project Title
Comments:

Session Information

ID Number: 1234
Title: session_name
Comments:

Session Type:
○ Transient Buffer-Wide (TBW)
○ Transient Buffer-Frequency Domain (TBF)
○ Transient Buffer-Narrow (TBN)
● Beam Forming (DRX)

Data Return Method:
○ Bare Drive(s)
● Copy to UCF

UCF Username:

Beam Processing: □ DR spectrometer

Channels: 32
FFTs/int.: 6144
Data Products: ● Linear ○ Stokes

Figure 2: The sessionGUI.py observer and project information window. Here you can define your name, the project, the LWA-SV mode to be used to collect the data, whether or not to use the data recorder spectrometer, and how the data are be to returned to you

7. Once you are happy with the observation setup, save the observation to disk. Before the file is saved it will be validated. If any of the observations have invalid parameters, they will be highlighted in red. The terminal window will also list the invalid parameters. Be sure to correct any problems before you exit `sessionGUI.py`.

8. SDFs are submitted to LWA station for observation through the SDF validator at http://fornax.phys.unm.edu/lwa/validator/index.html. This page accepts multiple SDFs or a tarball containing multiple SDFs and validates each file. Any errors in the SDFs will also be noted as part of the validation process. Valid files are sent to the observing queue to be run by the LWA station operator.

2.2 Pulsars at Low Frequencies

2.2.1 Observing a Known Pulsar in Spectrometer Mode

Space required: 918 MB

1. Take a look at the power and clipping levels in the spectrometer data file using scripts in the Commissioning repository. You can get an overview of the power levels and clipping via:

   ```
   drspecFileCheck.py 056770_000044687
   ```

   which will report both quantities. Optimal values of power are between seven and ten. Below seven some dynamic range is lost because the signal strength is not enough to populate many bit levels in the data. Above ten the clipping starts to become severe and the dynamic range is also compromised. However, high values for the power and clipping may be transient, e.g., radio frequency interference (RFI), and do not necessarily mean that the observation is useless. If the power or clipping is high for either tuning, convert the data to an HDF5 file and interactively plot the results using:

```
drspec2hdf.py 056770_000044687
plotHDF.py 056770_000044687-waterfall.hdf5
```

Inside plotHDF.py you can view both tunings and polarizations in a variety of ways and look at how the data evolve in time and frequency.

2. Now convert the file into PSRFITS format in preparation for folding the pulsar using PRESTO.

   ```
   writePsrfits2FromDRSpec.py -s B0329+54 056770_000044687
   ```

   This will create two files:

   ```
   drx_56770_B0329+54_b2t1_0001.fits
   drx_56770_B0329+54_b2t2_0001.fits
   ```

   One for each tuning. If the tunings are spaced by the sample rate the two PSRFITS files can be combined with combine lwa and jointly analyzed in PRESTO:

   ```
   combine_lwa2 -o drx_56770_B0329+54_b2 \
   drx_56770_B0329+54_b2t2_0001.fits \
   drx_56770_B0329+54_b2t1_0001.fits
   ```

   This will create a new file named “`drx_56770_B0329+54_b2_0001.fits`” that contains the data from the two input files.

3. Although the LWA1 site is relatively free of RFI it is often a good idea to create a mask to remove any RFI that may be present in the data. To create this mask use the PRESTO rfifind utility:

   ```
   rfifind -time 2.0 -o \
   mask_B0329+54_b2 drx_56770_B0329+54_b2_0001.fits
   ```

   The resulting mask includes a postscript file that can be viewed with evince or ghostscript.

4. Finally, incoherently dedisperse the PSRFITS file and fold the data at the pulsar’s period of ≈715 ms using the PRESTO prepfold utility:

   ```
   prepfold -psr 0329+54 -mask mask_B0329+54_b2_rfifind.mask \
   -nsub 128 drx_56770_B0329+54_b2_0001.fits
   ```

After folding, prepfold will generate an analysis plot showing the pulse strength over time, the constraints on the period and period rate of change, and the constraints on the dispersion measure (DM). Press the ”enter” key in the terminal window when finished.

2.2.2 Observing an Unknown Pulsar in Raw Data Mode

Space required: 18 GB

1. Start by converting the raw DRX data file to PSRFITS via:

   ```
   writePsrfits2.py 056227_000024985_DRX.dat
   ```

   and then create RFI masks for the two PSRFITS files similar to before. For this example we will name our two rfi masks “mask1” and “mask2”.

2. Since the DM of the pulsar is only constrained between 5 and 15 pc cm^2, search both files with PRESTO by first incoherently dedispersing files at a collection of trial DMs.

   ```
   for i in {1..2}
   do
   prepsubband -lodm 5 -dmstep 0.1 -numdms 100 \
   -nsub 128 -mask mask${i}_rfifind.mask -numout ##### \
   -o band${i} drx_56227_None_b4t${i}_0001.fits
   done
   ```

   This script will create a collection of dedispersed files with a base name of ”`band${i}`” where `${i}` is either 1 or 2. You can figure out the value used for `-numout` by looking at the number of points that rfifind reports when it starts, and then determining the closest power of 2. The next largest power of two can be found by running the following lines in python:

   ```
   import math
   print(2**math.ceil(math.log(######/math.log(2))))
   ```

3. For each trial DM file search for a pulsar using:

   ```
   accelsearch -zmax 0 -numharm 16 <name_to_search>
   ```

   Since prepsubband generates many files it is convenient to process them all at once with the xargs command:

   ```
   ls <output_base_name>*.dat | xargs -n1 accelsearch \
   -zmax 0 -numharm 16
   ```

which will run accelsearch on each file found by the ls command. Sift through the various pulsar candidates found by accelsearch to find the approximate DM and period for the pulsar using the PRESTO utility ACCEL_sift.py:

```
python $PRESTO/examplescripts/ACCEL_sift.py *.cand
```

The pulsar should show up across several closely spaced DMs at the same period

4. Once the approximate DM and period are known, fold the original data at the pulsar's period. This searches over a small range of DM and period to get the best fit. Note: prep fold takes DM values in pc cm^{-3} and periods in seconds. Note that you may need to convert the period from milliseconds into seconds.

   ```
   prepfold -dm ##.# -p ##.# <psrfitsfilename> -nsub 128 \
   -mask <maskname>
   ```

   Where you would replace `<maskname>` and with the appropriate maskname from above, such as `mask1_rfifind.mask` and you would replace `psrfitsfilename` with the psrfits filename created in step 1. Note the final DM and period.

2.2.3 Improving the Detection Using Coherent Dedispersion

Space required: 5.8 GB

1. Up to this point only incoherent dedispersion has been used for the pulsar analysis. However, the dispersion introduced by the interstellar medium can be removed from the raw data more precisely using coherent dedispersion if the DM is known ahead of time. To apply coherent dedispersion to LWA1 DRX data use the writePsrfits2D.py script:

   ```
   writePsrfits2D.py ##.# 056227_000024985_DRX.dat
   ```

   where the first argument to the script is the DM found above. After the file has been converted, create RFI masks for the two PSRFITS files similar to sections 3.1 and 3.2.

2. Fold the data at the pulsar's period using prepfold. This will also search over a small range of DM and period to get the best fit.

   ```
   prepfold -dm ##.# -p ##.# <psrfitsfilename> -nsub 128 \
   -mask <maskname>
   ```

   Note the final DM and period. Is the detection stronger with coherent dedispersion?

2.2.4 Measuring Faraday Rotation

Space required: 1.4 GB

1. The data for this exercise originally come from the LWA Pulsar Archive[1]. This archive contains observations of the 107 pulsars detected using LWA1 and has data that spans several years for some pulsar. The data on PSR B0950+08 comes from January of 2018 and is at a frequency of 64.5 MHz. All of additional data available for this pulsar on this date can be found at http://lda10g.alliance.unm.edu/PulsarArchive/B0950+08/58132/.

2. Since many of the pulsar analysis tools used in this exercise default to writing in the same directory as the data it is convenient to create a local copy of data. Do this by copying all the contents of the https://lda10g.alliance.unm.edu/tutorial/PulsarRM/ directory. Now make median_6.psh and plot_pol executable:

   ```
   chmod +x median_6.psh
   chmod +x plot_pol
   ```

3. Using your local copy of the data, flag obvious radio frequency interference (RFI) using the median 6.psh script.

   ```
   ./median_6.psh 58132_B0950+08_64.5MHz.0000.ar
   ```

   This script finds and flags data that are more than 6σ from the median and writes the results to a new file with a .zz extension

4. Next, integrate over all the periods (time-bins) of the pulsar. This can be done using the PSRCHIVE command pam:

   ```
   pam -T -e tscrunch 58132_B0950+08_64.5MHz.0000.zz
   ```

   Here, the `-T` flag tells pam to integrate over times and the `-e` flag sets the output extension to `.tscrunch`.

5. In order to convert to Stokes parameters we will again make use of the pam command, this time with the `-S` flag:

   ```
   pam -S -e tscrunchS 58132_B0950+08_64.5MHz.0000.tscrunch
   ```

   The four Stokes parameters (I, Q, U, and V) can be plotted to view the effects of Faraday Rotation using:

   ```
   ./plot_pol 58132_B0950+08_64.5MHz.0000.tscrunchS
   ```

[1] http://lda10g.alliance.unm.edu/PulsarArchive/

The Faraday Rotation manifests as frequency dependent banding in the Stokes Q (second plot from left) and U (third plot from left) plots of frequency vs. pulsar phase.

6. Now attempt to fit a rotation measure (RM) to the data. We know the RM of PSR B0950+08 is ≈2, but the ionosphere can change that up to ±1.5, so we will look at RMs from –5 to 5 and test 256 values of RM to get a good fit:

   ```
   rmfit -D -K 58132_B0950+08_64.5MHz.0000_rmfit.eps/CPS \
   -m -5,5,256 58132_B0950+08_64.5MHz.0000.tscrunch
   ```

   This will create a plot of trial RMs versus total polarization and attempt to fit a Gaussian to it. The plot is stored as a .eps file that can be viewed with evince.

7. Using the best-fit value for the RM, apply it to the data and re-generate the plot of the Stokes parameters:

   ```
   pam -R #.### -e tscrunchRM 58132_B0950+08_64.5MHz.0000.tscrunch
   pam -S -e tscrunchRMS 58132_B0950+08_64.5MHz.0000.tscrunchRM
   ./plot_pol 58132_B0950+08_64.5MHz.0000.tscrunchRMS
   ```

   In the first command of the above "`#.###`" is the best-fit RM value found with rmfit.

2.3 Studying the Transient Universe

2.3.1 Jovian Bursts

Space required: 273 GB depending on selection in step 2. For step one only: 91 MB

1. The data being analyzed here are part of a commissioning test observation of Jupiter. Unlike most LWA1 beam formed data, this file contains a single 19.6 MS/s tuning. Since reducing the raw data, even at a reduced spectral and temporal resolution, takes about two hours, a pre-reduced file named

   ```
   055917_000007389_Jupiter-waterfall-low.hdf5
   ```

   has been provided in https://lda10g.alliance.unm.edu/tutorial/Jupiter/. This file has a temporal resolution of five seconds and a spectral resolution of 4.8 kHz. Display this file using:

   ```
   plotHDF.py 055917_000007389_Jupiter-waterfall-low.hdf5
   ```

   and look for evidence of bursts during the file. For each burst note the time offset, UTC start time, and duration. The plotting script also allows for the color map, stretch, and scale to be interactively adjusted which may help identify events.

2. After you have identified any bursts in the file, produce higher resolution versions of the event using hdfWaterfall.py. Since the bursts are highly circularly polarized try processing the file to Stokes parameters:

   ```
   hdfWaterfall.py -s ##.# -d ##.# -a 0.01 \
   -k 055917_000007389_Jupiter.dat
   ```

   where "`-s`" sets the relative offset from the beginning of the file, "`-d`" specifies the duration to process, "`-a`" sets the integration time to 10 ms, and "`-k`" sets the processing mode to Stokes I, Q, U, and V. Note that the larger the time selection, the more disk space, processing time, and memory will be required for this and the following step.

3. Plot the higher resolution files with plotHDF.py to get a better view of the burst structure. Depending on the size of the higher resolution file you may need to display a subset of the data

2.3.2 Solar Bursts

Space required: 4.2 GB

1. The data being analyzed here are part of a program to search for prompt emission from gamma ray bursts (GRB). The data are from Fermi Trigger 4053677423 which turned out not to be a GRB but, in fact, a solar burst. The data are taken at a temporal resolution of 40 ms which makes displaying the full file difficult. Start by creating a lower resolution version of the HDF5 file in https://lda10g.alliance.unm.edu/tutorial/Sun/ using the decimateHDF.py script:

   ```
   decimateHDF.py -s 5 20 056601_000024625-waterfall.hdf5
   ```

   This will create a new HDF5 file that has one-fifth the spectral resolution and one-twentieth the temporal resolution which will be more suitable for displaying.

2. Examine the file using the plotHDF.py script and check both tunings for evidence of bursts.

   ```
   plotHDF.py 056601_000024625-waterfall-decim.hdf5
   ```

   For each burst you find, note the time range of the burst relative to the start of the observation. Once again, the plotting script also allows for the color map, stretch,and scale to be interactively adjusted which may help identify events.

3. For each of the bursts identified, display the burst in the full-resolution data in plotHDF.py. The script supports displaying a subset of the data via the "`-s`" and "`-d`" flags for "skip" and "duration":

   ```
   plotHDF.py -s ## -d ## 056601_000024625-waterfall.hdf5
   ```

2.3.3 Giant Pulses from the Crab Pulsar

Space required: 25 GB

1. The data being analyzed here are part of a project to characterize the low frequency giant pulsed emission (CGPs) from the Crab pulsar . The data consist of tunings at 60 and 76 MHz with a sample rate of 19.6 MS/s. To begin searching for the CGPs, first convert the first 100 seconds of data in /data/network/SummerSchool/CGPs/ to an HDF5 file with 16,384 channels and 40 ms integrations:

   ```
   hdfWaterfall.py -a 0.04 -l 16384 -d 100 056598_000003517
   ```

   The large number of channels will improve the dedispersion by reducing the amount of DM smearing per channel.

2. After the file has been converted, display it using plotHDF.py to get an idea of the overall interference environment and to check for any obvious pulses. The CGPs will appear as curves moving from the upper left to the lower right of the waterfall due to dispersion.

   ```
   plotHDF.py 056598_000003517-waterfall.hdf5
   ```

   Masking RFI will help identify pulses. The calculateSK.py utility provides a simple way to mask RFI based on spectral kurtosis. To create and apply a basic mask run:

   ```
   calculateSK.py -g 056598_000003517-waterfall.hdf5
   ```

   and examine the updated file in plotHDF.py. You can also manually create and edit the mask stored in an HDF5 file with plotHDF.py. To do this use the Mask menu. You may want to try different masking parameters via the `Mask -> Adjust` Masking Parameters dialog, particularly try adjusting the number of sections used for spectral kurtosis. You can also manually adjust the mask by clicking on points to be removed with the right mouse button. After creating an acceptable set of masks save them to the HDF5 file via `File -> Save.`

3. Since the Crab pulsar has a DM of ≈56.79 pc cm^{-3} the giant pulses will be more obvious if the data are first incoherently dedispersed and then spectrally averaged. To perform the dedispersion use the dedisperseHDF.py utility:

   ```
   dedisperseHDF.py 56.79 056598_000003517-waterfall.hdf5
   ```

   and then average using decimateHDF.py:

   ```
   decimateHDF.py -s 256 1 056598_000003517-waterfall-DM56.7900.hdf5
   ```

This will reduce the number of frequency channels to 64.

4. Display the dedispersed and decimated data using plotHDF.py:

   ```
   plotHDF.py 056598_000003517-waterfall-DM56.7900-decim.hdf5
   ```

   The CGPs will appear in the waterfall as horizontal bursts. To more easily identify the CGPs, you may need to turn on bandpassing via the “Bandpass” menu and flag channels that are dominated by RFI.

2.4 All-Sky Imaging

2.4.1 Detecting Meteor Radio Afterglows in Orville Images

Space required: 4.7 GB

1. The Orville imager is a wideband all-sky imager that creates 100 kHz bandwidth images over a continuous 19.8 MHz bandwidth every 5 seconds. These images are then averaged down to six channels for long-term archival storage. Plasma trails from meteors can create radio emission observable by the LWA. Varghese et al. [4] found meteor radio afterglows (MRAs) in the 198 channel unaveraged data. For this tutorial, we will look at an averaged 6 channel version available at https://lda10g.alliance.unm.edu/tutorial/MRAs/, which is the format that is available in the Orville archival images[2].

2. The .oims can be converted into a .fits file using the OIMS2fits.py script:

   ```
   OIMS2fits.py 58956_080000_30.100MHz_49.800MHz-avg.oims
   ```

3. The resulting multi-extension fits files can be viewed using ds9 with the -mecube option. For now just pick the 48.2 MHz image:

   ```
   ds9 -mecube 58956_080000_48.2MHz.fits
   ```

   The MRA detection from this dataset was around 8:15 UTC. We can find the image from around that time by selecting `File -> Header` and picking a header and looking for the ”DATE-OBS” keyword. Make a note of the time index number of the last image as shown in figure. Hint: If you assume that each file starts at the top of the hour, you can get close to the right index by converting 15 minutes into seconds then dividing by the duration of a single image.

4. The MRA is not clearly visible in the regular image, so we will create a difference image to find sources that have changes in brightness from one image to the next. We do this by re-running OIMS2fits.py with the `--diff` option:

   ```
   OIMS2fits.py 58956_080000_30.100MHz_49.800MHz-avg.oims --diff
   ```

[2] https://lda10g.alliance.unm.edu/Orville/

5. Use ds9 to view the difference images that were created:

   ```
   ds9 -mecube 58956_080000_48.2MHz-diff.fits
   ```

   In the difference images we see quite a few scintillating bright sources and possible RFI on the horizon, but at around 8:15 we see a bright source that looks slightly elongated. It appears and then disappears in the following images. This object is the MRA. Make a note of the time index number where the MRA appears.

2.4.2 Performing Flux Scale Calibration and Flux Measurement

In this tutorial, we'll perform a rudimentary flux calibration in order to measure the flux of the MRA we detected in the previous tutorial. For this tutorial, we'll measure the flux of Cygnus A to derive a correction to the flux scaling. A more rigorous approach is given in [4] and would involve measuring calibration sources over multiple images in order to discard measurements most affected by scintillation. Alternatively for an example of flux calibrating pulsar observations see [2] and [1].

1. First we will create a primary beam corrected image of the last image in the 48.2 MHz observation in order to measure the calibrator sources. The primary beam correction will correct for the fall off in sensitivity for sources away from the center of the field-of-view. We can use OIMS2fits.py to perform this correction:

   ```
   OIMS2fits.py -p -i 626 --channel 5 \
   58956_080000_30.100MHz_49.800MHz-avg.oims
   ```

   The `-p` indicates primary beam correction, the `-i 626` indicates that we only want the 626th image in the dataset, and the `--channel 5` indicates that we want the last, highest frequency channel

2. Now we will open the image in ds9. You may need to adjust the scaling in the `Scale` menu in order to make everything visible. Now, create a region by selecting `edit -> region.` Click the image and drag the new region into the emptiest part of the image in order to make a background measurement. In the region window titled "Circle" select `Analysis -> Statistics` and record the RMS measurement. You may want to try moving the window around to see how it affects the RMS number.

3. Edit the location of the region to contain the source Cygnus A which is located at 19:59:28 +40:44:02 (RA DEC J2000). Make the radius of the region 6 degrees.

4. Once again in the region window select `Analysis -> Statistics` and find the Max flux measurement and take a note of it.

5. Find the catalog flux of Cynus A using the VLSS Bright Source Spectral Calculator: https://lda10g.alliance.unm.edu/calspec/calspec.html. Follow the instructions on the website and use the 3C name along with the frequency to obtain the interpolated flux. Record the given interpolated flux value for Cygnus A. Some of the sources that can be used to calibrate flux from this website include:

   (a) Cassiopeia A (3C 461), 23:23:24 +58:48:54

   (b) Cygnus A (3C 405), 19:59:28 +40:44:02

   (c) Crab/Tau A (3C 144), 5:34:30.9 +22:00:53

   (d) Virgo A (3C 274), 12:30:49 +12:23:28

6. Using the information we have we willl create an averaged correction factor for the flux. This factor can be computed by taking the ratio of the model flux to the background-subtracted flux measurement:

$$corrfac = model/(measured - background) \tag{1}$$

   Calculate and record this value.

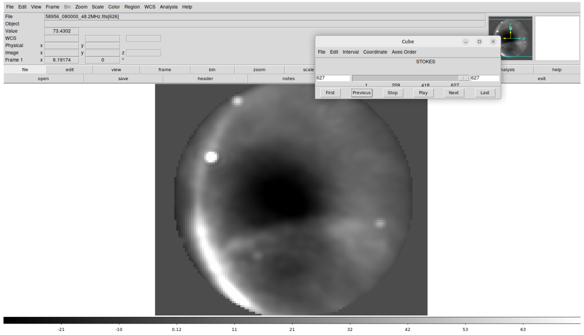
7. Create a primary beam, flux corrected image of the MRA using OIMS2fits.py with the `--background`, `--corrfac`, `--index`, `--diff`, `--channel`, and `-p` options:

```
OIMS2fits.py --background ?? --corrfac ??  --channel 5 \
--diff  -p 58956_080000_30.100MHz_49.800MHz-avg.oims
```

8. Now we can open the image we create with ds9, create a region around the MRA, and record the flux of the MRA.

3 Acquiring Your Own LWA Data

Now that you know how the LWA works, consider using the LWA for a research project of your own. The LWA offers regular calls for proposals in September every year. Typical requests are for 50 beam hours or more, using one or more stations. You can also request smaller allocations at any time through requests to the LWA Director for Director's Discretionary Time (DDT). DDT requests can be made to explore the feasibility of a larger project, or for time-critical observations. In all cases proposals should be sent to `lwa@unm.edu`. The latest call for proposals, along with lots of useful information about the LWA can be found at `https://leo.phys.unm.edu/~lwa/proposals.html`.



File 58956_080000_48.2MHz.fits[626]

Object

Value 73.4302

Frame 1 x 6.19174 0 °

Cube

STOKES

627 627

File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
WCS
Physical x y
Image x y z
file edit view frame bin zoom scale analysis help
open save header notes exit
File Edit Interval Coordinate Axes Order
1 209 418 627
First Previous Stop Play Next Last
Y
N
X
-21
-10
0.12
11
21
32
42
53
63

Figure 3: The ds9 window is shown with an image opened using the -mecube option. The index of this image is 626 as shown in the field marked ”File.” Note that the cube window shows the index number+1 because it is one indexed instead of zero-indexed.

References

[1] P. Kumar et al. “A multi-frequency census of 100 pulsars below 100 MHz with LWA: a systematic study of flux density, spectra, timing, dispersion, polarization, and its variation from a decade of observations”. *Submitted to ApJS* (2023).

[2] K. Stovall et al. “Pulsar Observations Using the First Station of the Long Wavelength Array and the LWA Pulsar Data Archive”. *ApJ* 808.2, 156 (Aug. 2015), p. 156. DOI: `10.1088/0004-637X/808/2/156`. arXiv: `1410.7422 [astro-ph.IM]`.

[3] G. B. Taylor et al. “First Light for the First Station of the Long Wavelength Array”. *Journal of Astronomical Instrumentation* 1.1, 1250004-284 (Dec. 2012), pp. 1250004–284. DOI: `10.1142/S2251171712500043`. arXiv: `1206.6733 [astro-ph.IM]`.

[4] S. S. Varghese et al. “Broadband Imaging to Study the Spectral Distribution of Meteor Radio Afterglows”. *Journal of Geophysical Research (Space Physics)* 126.10, e29296 (Oct. 2021), e29296. DOI: `10.1029/2021JA029296`. arXiv: `2103.03347 [astro-ph.EP]`.