2003

Astronomy 322

AIPS Laboratory:

Making, Cleaning and Analyzing a VLA Map

I. Introduction

These sheets are to help you navigate a software package called AIPS (Astronomical Image Processing Software). You will "learn by doing." That is, most of the learning about AIPS will be as you use it to construct a map of the sky from u-v data obtained several years ago using the VLA. Embedded in this write-up are a number of tasks for you to complete and turn in. You will also be asked to print out some of the images you construct and turn those in as well.

II. What Went Before…

The data I'm providing you has already been considerably processed. Here, I want to list the steps that it has gone through before you received it; you may want to look up these AIPS tasks just to see what has already been done for you. Here and elsewhere, I'll specify task names with all caps, like TVFLG. You can type lower or upper case.

1. Load the data into AIPS: relevant program…UVLOD
2. Sort the data " UVSRT
3. Calibrate the data " VLACAL
4. Remove bad data " TVFLG

The second of these programs arranges the u-v data in increasing distance from the center of the u-v plane, the form in which it's easiest to analyze for the making of "maps," or radio images of the sky.

III. Entering AIPS

To get into AIPS, log in to Urania (our Dell workstation) or one of the XTerms in the Observatory, using your name and password, and when you get the prompt, type "start_aips." The program will ask you which printer to use—select 1, the Observatory printer. It will then ask you your AIPS user number—be sure to use your number and *only your number*. In all your lab work, supply your AIPS user number as well as your name. You are now ready to look for data, or to carry out various operations on it. The data will be in two forms, either u-v data (visibilities) or images of the sky, called maps. To see what you've got, select a particular disk by typing "INDI n" where n is the number of the disk. I'll set you up with the data on disk 1. Then to see what u-v data you've got, type "UCAT" and return; for maps, type "MCAT."

To fetch data into a program, you call it by its catalog number on the disk using the command "GETN n".

IV. A Primer of AIPS Tasks and Variables

The programs in AIPS are considered either tasks or verbs. To specify a particular task, like making a map, you would type TASK 'UVMAP' for instance. Note the single quotes. That calls up the task called UVMAP. Once you've called up a task, you have to specify how it is to be carried out. That is done by specifying a number of *inputs*. For instance, you may need to get some data into the program. You would do that by using GETN (see above). [In order to change the inputs, of course, you have to be able to view them—that is done by typing “INP” once you’ve selected a task.] In the instructions that follow, I’ll generally tell you which of the input variables you have to worry about. They can be changed by typing the name of the variable (for instance OUTNA) followed by the quantity or name. In some cases the quantity has to be enclosed in single quotes; in other cases it does not—look to see how each quantity is specified on the inputs.

Once the inputs are all set, you make the task operate by typing “go” and the task name, this time with no single quotes around it.

You may have noticed that AIPS opens a second window on the screen, which is a message window—this will tell you how your programs are doing. One thing you should look for is the dreaded phrase, “dies of unnatural causes,” in which case you need to seek help. A happier outcome is “ended successfully.”

V. **Making a Map**

The task you want to use here is UVMAP. It basically takes a set of u-v data, correctly sorted, and then fast Fourier transforms it into an image of the sky. You will need to set the parameter INNAME to your u-v data; the easiest way to do this is to type “GETN n” with n the appropriate catalog number once you’re in the task UVMAP. You will also need to think about the size of the image you wish to construct. Given the speed of our system, it’s okay to experiment with either 512 or 1024 maps; the final product should probably be a 1024 map, that is 1024 pixels on a side. You will be asked to specify a UVWTFN—use ‘NA’—note the single quotes.

You will also need to work out the size of the pixels in the map. The rule of thumb is that there should be four or more pixels across the synthesized main beam. So you will need to work out roughly how big the synthesized beam will be. That will be determined by the size of the array (d), which I will give you, and the wavelength.

Cell size = __________

This is the first of the pieces of information I want you to turn in.

The program UVMAP also constructs an image of the beam. Recorded in the data I’ve given you is information about the position of each of the 27 elements of the VLA. The computer uses this to construct a gray scale image of what the synthesized beam looks like. To find the catalog numbers of the map and the beam produced by UVMAP, use MCAT.

VI. **Viewing Images**

Once UVMAP has worked successfully, you’d like to look at the image of the beam and the image of the sky you have created. That can be done with the task TVLOD. First, use GETN n to bring in the appropriate image, then “task ‘TVLOD’” and INP to check the inputs. When they are okay, type “go tvlod” and fiddle with the TV window until you’ve got a nice image. You may need to adjust the transfer function (the brightness and contrast) of the image; that is done with a separate task called TVFID. Once you start TVFID, it will instruct you as to how to proceed.

VII. Manipulating, Analyzing and Printing Images

Here I list a series of tasks useful for analyzing and improving radio images displayed on the TV, using AIPS.

TVFID—changes *contrast* (be sure to have cursor in image window when you push buttons A-D)

Note: "buttons A-D" ≡ F3-F6 function keys on our keyboards.

TVWIN--draws boxes on screen, and sets BLC and TRC for later tasks.

TVSLICE—draws a line on screen.

TVRES—"restores" and cleanses image.

(If you need to reload, use TVLOD again.)

IMSTAT—takes statistics within box set by TVWIN.

IMFIT—fits 2-d Gaussian to a source in a box set by TVWIN; output should appear on printer.

SLICE—takes a profile of intensity along a line previously set by TVSLCE.

Using SLICE produces a graphic image (sketched below). How do you display such images? Plots like this and other graphic images (as opposed to radio images of the sky) appear in a separate (fourth) window, the TK window. To display such images, type go TKSL. To make a hard copy, type go SL2PL; then go LWPLA and look for your output on the printer.

VIII. Task to Perform on This Map (the "Dirty" Map) and the Beam

1. Take a slice through the center of the *beam* image using TV and print out this cross-section of the beam. Is the response of the VLA unity on axis, as expected? By playing around with various slices, get one that you like, and use it to estimate the relative amplitude of the first interference side lobes of the VLA beam.

   side lobe level = __________%

   I also want you to turn in the beam image when you're happy with it. Finally, I'd like you to comment on the particular pattern you see in the image. Can you relate this to the
   Y-shaped arrangement of the VLA antennas?

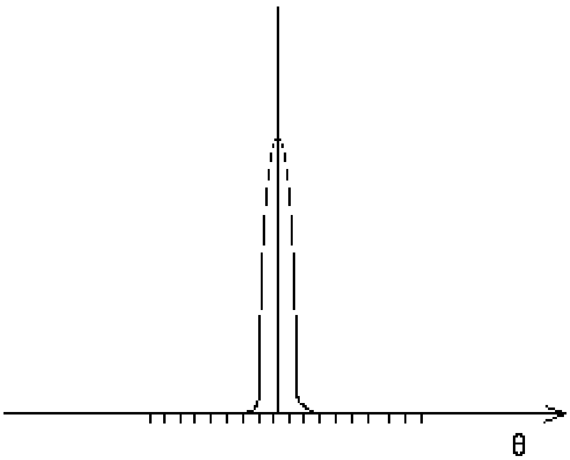
   Now use TVLOD to look at the image of the sky, the so-called "dirty map." It will look quite messy. In particular, you should see a bright source or two with strong side-lobe patterns attached. When you have the transfer function (brightness and contrast) to your liking, make an image of this dirty map. Also use the programs TVWIN and IMSTAT to measure the rms noise in a portion of the image away from bright sources. Record this value.

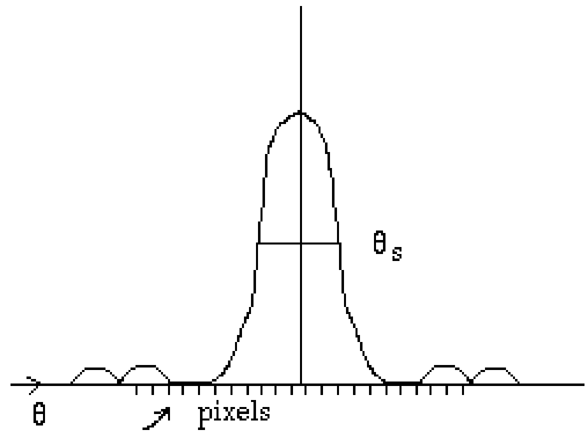
   rms noise in the dirty map = __________.

IX. Cleaning the Map

As the adjective "dirty" suggests, the map you've just constructed is full of artifacts. In particular, each bright point in the image of the sky is surrounded by the side-lobe pattern of the beam. Since we know what the beam looks like, however, we can in principle subtract out those effects, leaving you with a "clean" map. That is what the program APCLN does.

Let’s look at that in a little more detail. Suppose you were dealing with a point source, or at least a source much smaller than either the pixel size or the synthesized beam. Then it should appear in your map essentially as a delta function, as shown in fig. 1 below. But because of the shape of the synthesized beam, you will actually have the shape of the synthesized beam, as shown in fig. 2 below—recall the case of $\Omega_s << \Omega_A$ I gave you early in the course.





What the program APCLN does, in effect, is to start with the brightest pixel in the dirty map, convolve it with the synthesized beam image, then subtract 1/10 of that convolved pattern from the map. Thus, in effect, it’s subtracting off 1/10 of the amplitude of the side lobes. It then seeks out the next brightest pixel, and so on. It does this n times, where n is specified by the input parameter NITER. Hint: in a dirty map like the one you’re working with, it’s a good idea to start off with a light clean, say NITER = 500.

A couple of further remarks. First, if one source is the brightest in the field, the cleaning program will essentially deal only with it (a very bright source could contain the brightest 500 pixels in your map). Therefore, if you do have a bright source, you may need to clean more deeply. This is an art, not a science, and I’ll expect you to play around considerably with a number of iterations in APCLN.

Next, the program subtracts 1/10 of the amplitude of the beam pattern for each iteration. Pretty clearly, if it ran to its ultimate limit, it would subtract all the sources in the map. That you don't want. So you need to “restore” the clean sources, rid of their side lobes, to the map. That is done by specifying an estimate of the beam size—this rather complicated feature I’ll explain to you in real time.

Your task is to fiddle with APCLN until you have a clean map that looks like the map I made, called “first map.” When you’re happy with *your* map, print it out. I also want you to use TVWIN and IMSTAT to find the rms noise of this clean map (which has CLN as its tag in MCAT).

rms noise of clean map = __________.

If you’ve cleaned the map correctly, you should see that the rms noise is less than you found for the dirty map—that is, getting rid of all the side lobes reduces the statistical noise in the map as well as its visual clutter. You should also be able to see many more sources than in the dirty map.

X. Measuring the Position and Flux of Sources

Once you have a clean map, with lots of radio sources visible, you'd like to know what their flux density is. That can be done by using the programs TVWIN and IMFIT. First use TVWIN to draw a box fairly tightly around each source. Then simply type "go imfit" and the program will analyze the sources. It does this by fitting a two-dimensional Gaussian to the brightest source in the window you have drawn. Once the Gaussian is fit, the program integrates the flux underneath that two-dimensional Gaussian curve and determines its central position. It then reports these numbers as the flux density of the source and its position in the sky. Each time you run IMFIT, it will give you a crude map of the source, of the two-dimensional Gaussian model and of the difference between those two. Look at the third of these to see whether there are huge discrepancies—if there are, your Gaussian model is probably not a very good fit, and you can't trust the flux densities very much. We will talk about how to measure flux densities for complicated sources, several of which may appear in your map.

The next thing to turn in is a list of sources in increasing order of right ascension, giving their positions in right ascension and declination and their flux density with its associated error. I would also ask you to comment on the shape of each source, as seen in your clean map, and hence the likelihood that your flux density measurements are accurate.

XI. Primary Beam Correction

The task PBCORR corrects for the primary beam response. Recall that the Fourier Transform of V(u, v) gives the *convolution* of the true sky map and the beam pattern $F(\xi, \eta)$. PBCORR divides out $F(\xi, \eta)$. Notice that the effect is to increase the noise level towards the edge of the maps—think through why this should be so, and write down a brief statement explaining this result.

A source not at the phase and map center will be "down-weighted" by $F(\xi, \eta)$ until the image is PBCORR'd. For that reason, to measure the true flux density of a source not at the map center, you should use IMFT on the PBCORR'd map.