Ay190 – Final Project
Scott Barenfeld
Dónal O Sullivan
Date: March 22, 2014

1 Introduction

Radio interferometers allow information from multiple radio antennae to be combined, giving the interferometer as a whole much greater resolution than a single dish on its own. Because radio telescopes measure the incoming electric field itself (as a voltage), instead of just its intensity, the amplitude and phase of the electric field is preserved. In an array, the time-averaged product of the voltages measured by each pair of antennae is reported as a *visibility*. Between antennas i and j, the visibility is defined as:

$$\mathcal{V} = \int_{4\pi} A(\hat{r}) I(\hat{r}) e^{-2\pi i \nu \vec{B}\cdot\hat{r}/c} \, \mathrm{d}\Omega, \tag{1}$$

where A is the beam pattern of an individual antenna (in this case, a Gaussian with σ=1 arcmin), I is the intensity of the sky, ν is the frequency of the radiation, $\vec{B}$ is the baseline vector between i and j ($\vec{r}_i - \vec{r}_j$), and $\hat{r}$ is the direction along the line of sight. In the limit of a narrow field of view, Equation 1 can be rewritten using a change of variables as:

$$\mathcal{V}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{A(l,m) I(l,m)}{\sqrt{1-l^2-m^2}} e^{-2\pi i (ul+vm)} \, \mathrm{d}l \, \mathrm{d}m, \tag{2}$$

where (u,v) is the baseline vector between the two antennae in units of wavelengths and l and m are the angles (in radians) on the sky relative to the line of sight. Equation 2 is simply a two-dimensional Fourier transform. Thus, an image of the intensity pattern on the sky can be generated from the inverse Fourier transform of the measured visibilities.

$$\frac{I(l,m)A(l,m)}{\sqrt{1-l^2-m^2}} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{V}(u,v) e^{2\pi i (ul+vm)} \, \mathrm{d}u \, \mathrm{d}v, \tag{3}$$

2 Setup

For this project, Michael provided us with a list of antenna positions and a list of the amplitude and phase of the visibility for each antenna pair, such that the visibility is defined as:

$$\mathcal{V} = A e^{i\phi}. \tag{4}$$

After reading in these files, our first step was to turn the antenna positions into (u,v) baseline vectors, which we did by dividing the difference in x and y positions for each antenna pair by the wavelength of the observations (1 cm). We then combined the (u,v) baselines for each pair of antennas with the corresponding visibilities into a single array, `uvvis`.

3 Imaging with a DFT

3.1 Equations and Code Layout

Using a Discrete Fourier Transform is relatively straight forward. We compute an intensity map $I(l, m)$ using the discretized version of Equation 3, which is implemented by the function `DFT` in our code.

$$I(l,m) = \frac{\sqrt{1-l^2-m^2}}{A(l,m)} \sum_u \sum_v \mathcal{V}(u,v) e^{2\pi i(ul+vm)} \tag{5}$$

Here, the summation is over the baselines u and v connecting a subset of the antennae (e.g. the 10 closest/farthest antennae). The following definitions are used to simplify the layout of the code:

$$a = \sqrt{1-l^2-m^2} \tag{6}$$

$$DFT_{rhs} = \sum_u \sum_v \mathcal{V}(u,v) e^{2\pi i(ul+vm)} \tag{7}$$

such that:

$$I(l,m) = \frac{a}{A(l,m)} DFT_{rhs} \tag{8}$$

Finally, the antenna beam mentioned earlier is implemented as a Gaussian with a standard deviation of σ=1 arcmin, as in Equation 9 which is really only inserted here for reference and comparison with the code.

$$A(l,m) = \frac{1}{\sigma\sqrt{\pi}} exp \frac{-(l^2+m^2)}{2\sigma^2} \tag{9}$$

The method `part_one` implements this DFT construction of $I(l, m)$ for a given subset of antennae defined by the parameters `N` and `order` where `N` is the number of antennae to be used and `order` may have the values 'asc' (ascending) or 'desc' (descending); meaning that the N closest/farthest antennae will be used, respectively. This selection is implemented by the method `get_selection`, which indicates which rows of the `uvvis` array may be used given the selected antennae and also returns the indices of the antennae themselves. Once this selection is made, the rest is a straight forward implementation of Equations 7 and 8.

3.2 Results

The images obtained by using the 10 closest/farthest antennae are shown in Figure 1. Immediately obvious is the difference in resolution between the two; the 10 closest antennae have a much smaller baseline on average and so provide a more blurred image while the 10 farthest have a relatively large baseline and provide more fine detail image. What is perhaps not immediately obvious is the cause of the asymmetry, given that the FFT (see next section) is symmetric. We explain this in the discussion section at the end. Finally, timing will also be discussed in the last section.

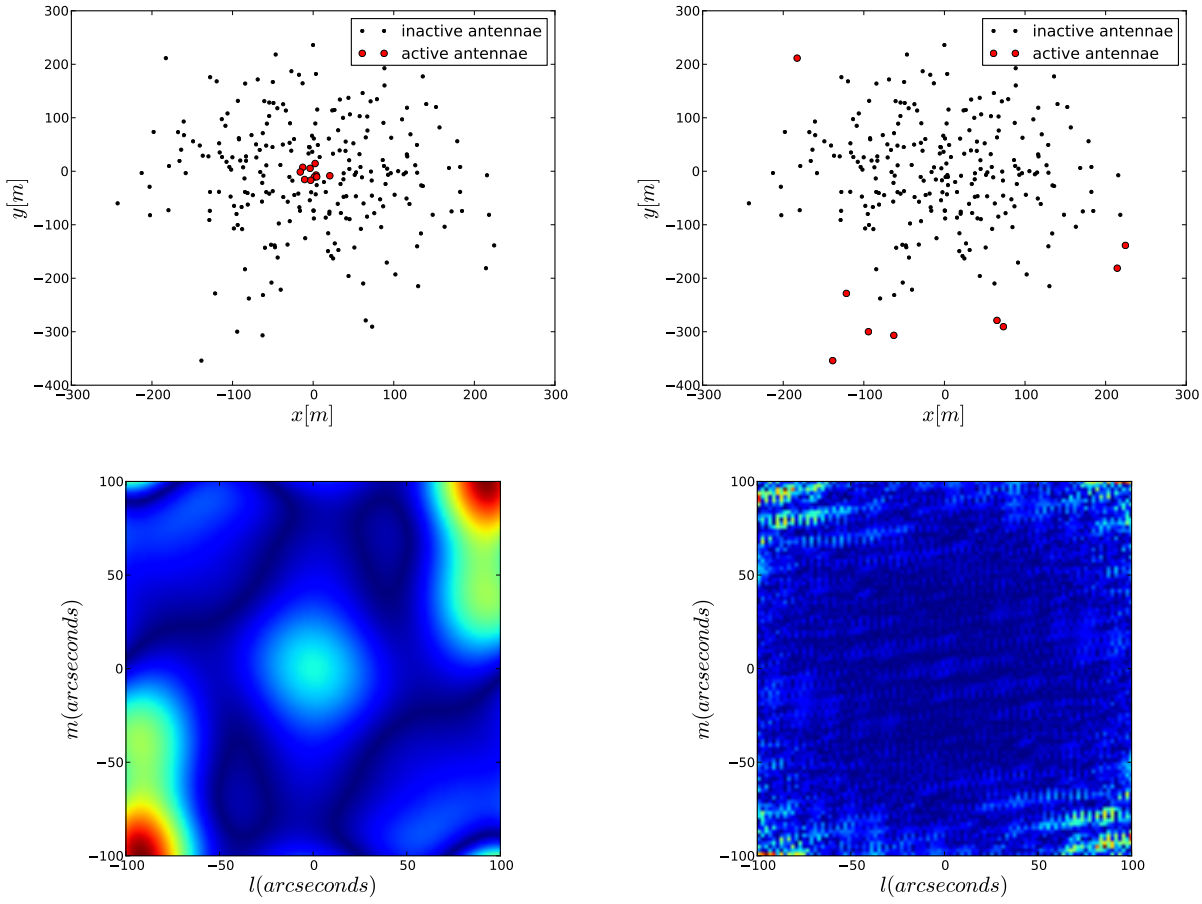

Figure 1: Image created with a DFT using (left) 10 closest and (right) 10 farthest antennae. The top row shows the antennae positions, while the bottom row shows the corresponding image created.

4 Imaging with an FFT

Python's built-in FFT and inverse-FFT functions expect the input to be on an evenly spaced grid. So to form an image from the measured visibilities using an FFT, we first had to place the visibilities on such a grid. We created a 100×100 grid in the (u,v) plane. For each measured u and v value, we found the grid u and v it was closest to using Python's `bisect_left` function. This function returns the location of where a given value would be inserted in a sorted array. So, given a measured u or v, the closest grid u or v is one of the elements on either side of where the measured value would be inserted into the grid. Subtracting the measured value from each of these two elements gives which grid element element is closest. Once each measured (u,v) is assigned to a gridpoint in this way, the visibility at each gridpoint is defined to be the sum of all the measured visibilities assigned to that gridpoint.

The gridded visibilities are then use to form an image. From Equation 3, it is easy to see that intensity is given by:

$$I(l,m) = \frac{\hat{\mathcal{V}}(l,m)\sqrt{1-l^2-m^2}}{A(l,m)}, \tag{10}$$

where $\hat{\mathcal{V}}(l, m)$ is the inverse-FFT of the gridded visibilities. The resulting image is shown in Figure 2.

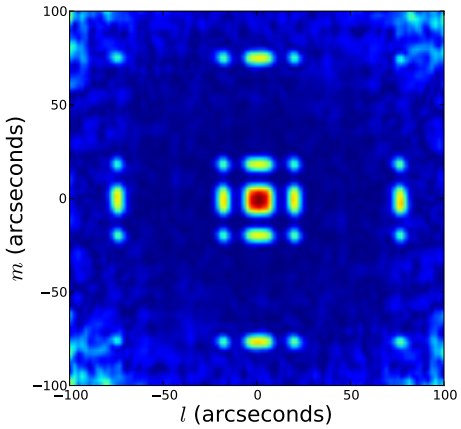

Figure 2: Image created with an FFT using all 256 antennae.

5 Discussion

5.1 DFT Asymmetry

Two things that confused us for a while was the asymmetric form of the images produced by the DFT method and the higher intensities towards the corners of the image. Given that the intensity map calculated using all 256 antennae with the FFT was symmetric and had highest readings in the center, it didn't seem to make much sense at an intuitive level. Luckily, Tony Readhead came to the the rescue! We sat down with Tony and figured it out.

Firstly, the higher intensities at the center come from the fact that $I(l, m)$ is inversely proportional to the antenna's (Gaussian) beam, as seen in Equation 8. Figure 3 shows a map of the contribution to the intensity from the antenna beam and the a term from Equation 6.

This explains the high intensity at the edges: as the Gaussian drops to values near zero, the inverse of the Gaussian explodes and amplifies the intensity significantly. However, this raises two questions; (a) why don't we see this in the FFT using all 256 antennae and (b) how does the center become brighter if the beam has this shape?

These questions are answered by considering the effects of increasing the number of antennae used. Figure 4 shows the intensity maps produced by 2, 4, and 6 antennae in the center of the array when ignoring the beam effects entirely. It is immediately clear from this where the asymmetry comes from. The low number of antennae produces a fringe pattern which is oriented in one particular direction. Because of this fringe pattern, the bottom-left and top-right corners are more intense than the top-left and bottom-right, and these corners then get amplified by the inverse Gaussian. Figure 5 shows the same maps with beam effects included. The next most obvious thing is that as more antennae are added, the intensity at the edges (caused by the sidelobes of the

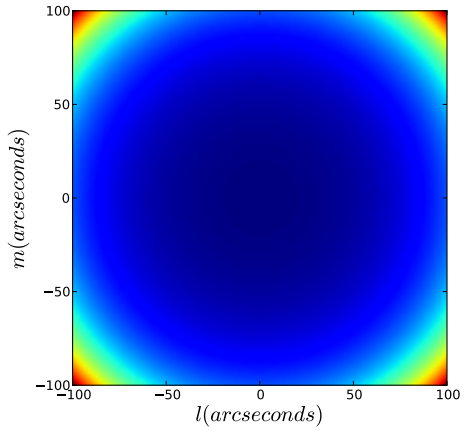

Figure 3: Contribution to intensity from the Gaussian antenna beams and Equation 6.

antennae) gets reduced and the signal becomes more focused on the center. What's more is that as more antennae are added, the summation term in Equation 5 becomes larger and so the overall intensity is increased until such a point that it dominates over the inverse Gaussian term. These effects explain why we see an asymmetric edge-focused pattern when using only 10 antennae, yet as we increase to 256 antennae we still get the expected result.

5.2 Timing

To test the performance of our DFT and FFT routines, we used Python's built-in timing capabilities to determine how long each routine takes for different numbers of antennae N. The results are shown in Figures 6 and 7. The DFT routine has to loop through all $N(N-1)$ baselines to perform the Fourier transform. So, for large N, the time required to run the DFT routine should go as N^2. This matches what we observe. On the other hand, Python's FFT algorithm runs in $\mathcal{O}(N \log N)$ time. Figure 7 shows that, while there is some scatter to the fit, the time to run the routine follows an $N \log N$ proportionality reasonably well. The scatter may be due to small variations in the computer's performance in each run (note the scale), or due the the additional step in the FFT of gridding the data. More important than the exact scaling is the fact that the FFT is much faster than the DFT. In the interest of time, we only timed the DFT for up to 20 antennae. Even using only 20 antennae, the DFT was nearly five times slower than the FFT using 250 antennae.

5.3 Gridding

One of the key steps in performing an FFT of the visibilities is gridding. For this project, by adding all measured visibilities to the gridpoints to which they were assigned, we are giving equal weight to each measured visibility when computing the FFT. This seems like a natural thing to do, but it does introduce possible problems. If the radio array provides non-uniform (u, v) coverage, extra weight will be given to certain gridpoints just because they have more measured visibilities around them. For example, the Owens Valley LWA has 256 antennae within a 200 meter radius, providing over 30,000 relatively short baselines. To increase angular resolution, several new antennae are being built with baselines of 2 km. If equal weighting were given to

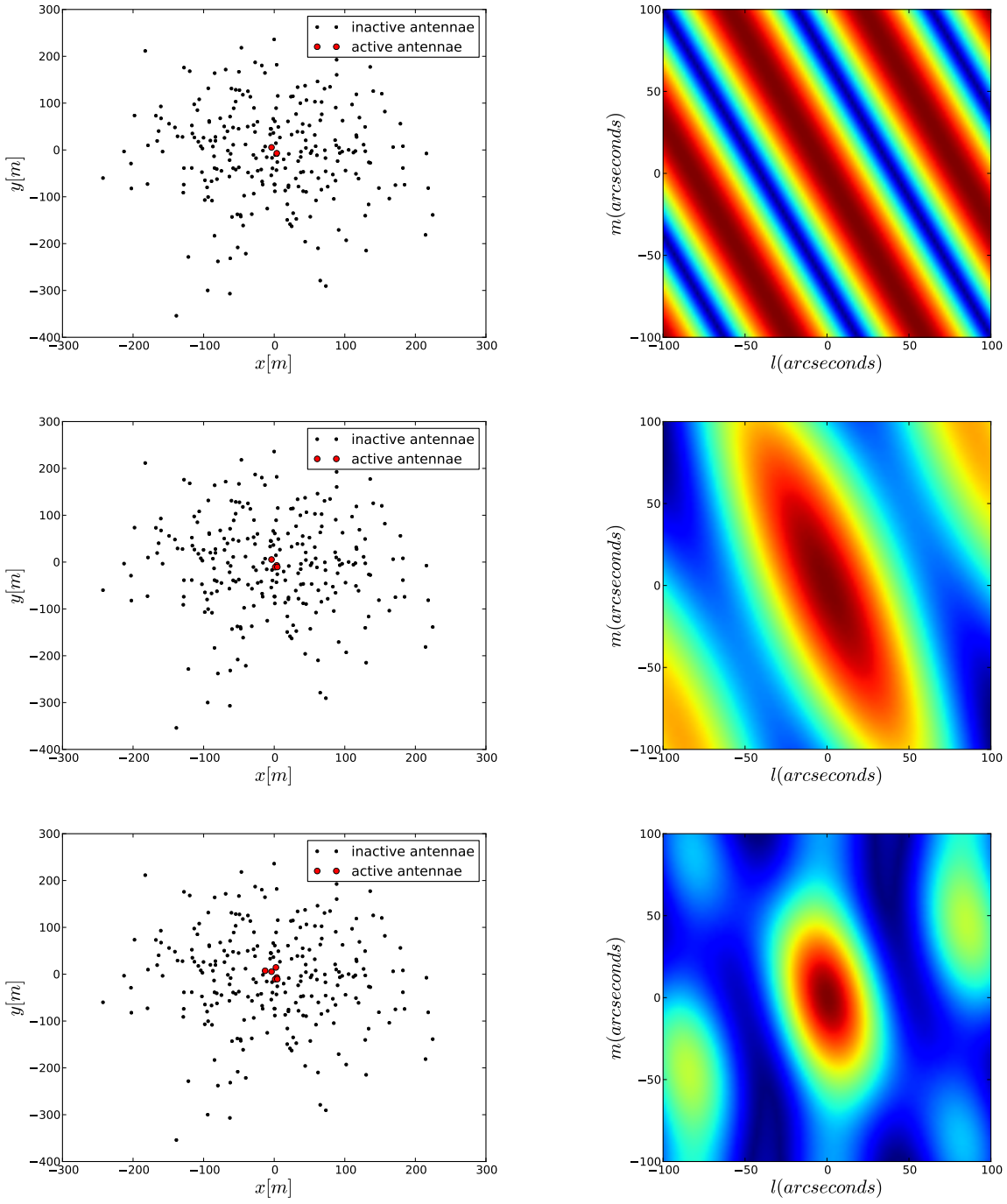

Figure 4: Evolution of the intensity map produced by using 2, 4, and 6 antennae in the center of the array, ignoring beam effects. The left hand side shows the antennae used to produce the image on the right.

all measured visibilities, the few long baselines would be completely dominated by the 30,000 shorter baselines, negating the increased resolution. One way around this problem is to build additional antennae to more uniformly cover the (u, v) plane. However, this is often expensive

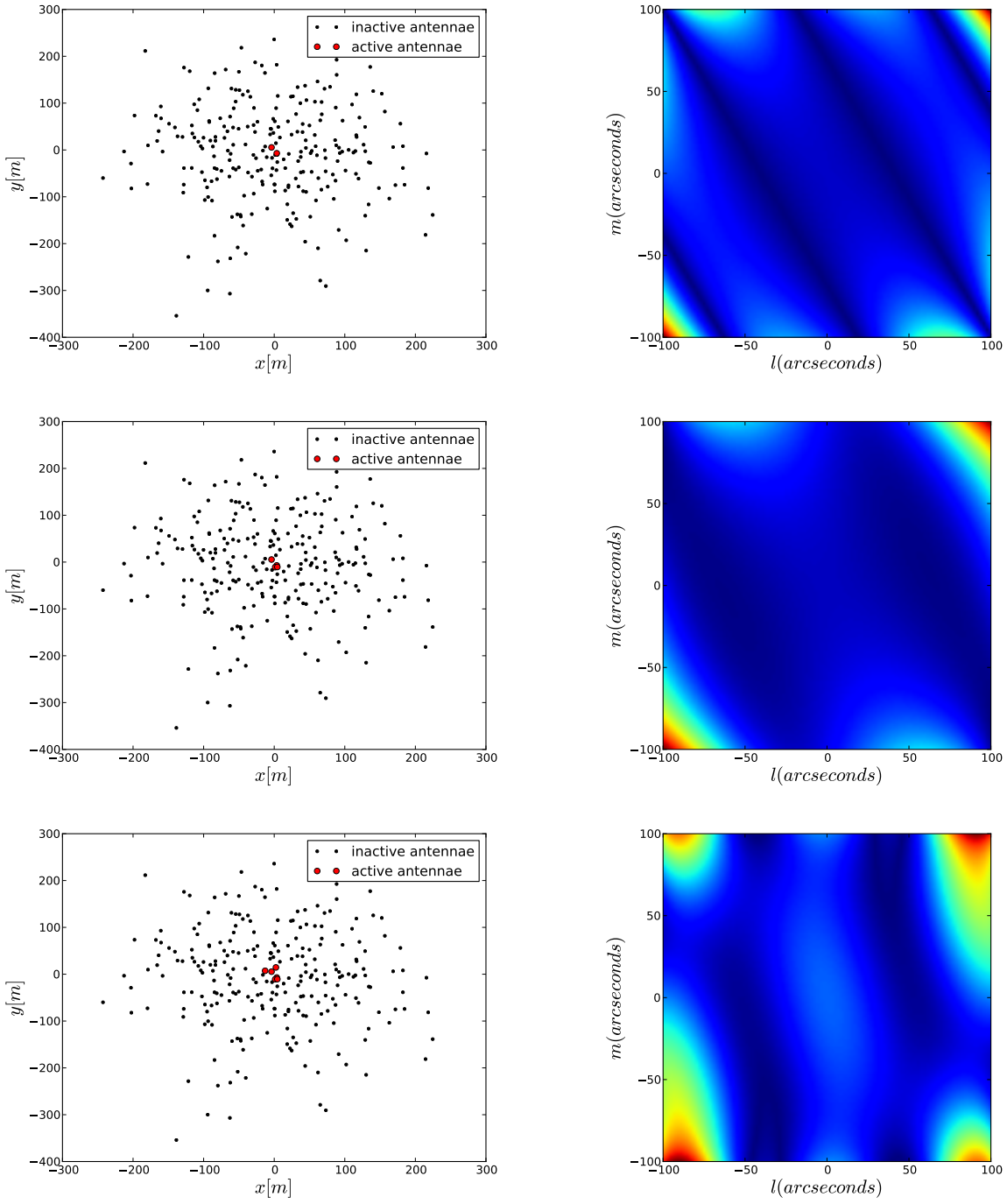

Figure 5: Evolution of the intensity map produced by using 2, 4, and 6 antennae in the center of the array, ignoring beam effects. The left hand side shows the antennae used to produce the image on the right.

and impractical. Instead, the visibilities from the shorter baselines can be given less weight than ones from the longer baselines. This allows the longer baselines to increase the resolution of the array. However, its overall sensitivity decreases due to the contribution of so many visibilities

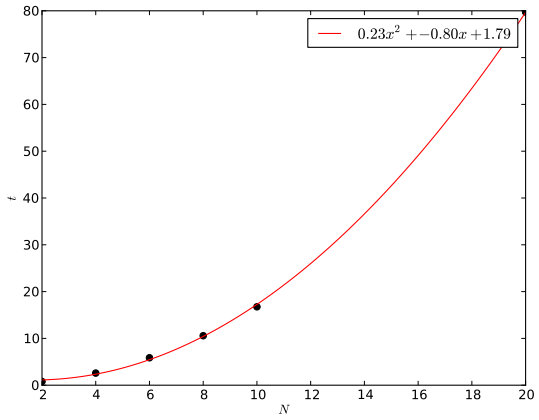

Figure 6: Time (in seconds) required to run our DFT routine as a function of antenna number. As expected, $t \propto N^2$.

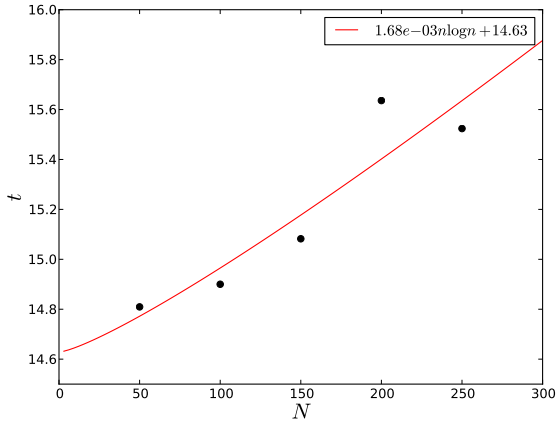

Figure 7: Time (in seconds) required to run our FFT routine as a function of antenna number. As expected, $t \propto N \log N$. Also note that the FFT runs much more quickly than the DFT.

being decreased. This is a trade-off that must be considered when designing an array. The proper weight should reflect the scientific goals the array is intended to achieve.