

1 Weights and variance of gridded visibilities

We start with the rms of a visibility,

$$\sigma_v = \frac{2k_B T_{sys}}{A_e \sqrt{df\tau}} \quad (1)$$

where k_B is Boltzmann's constant, T_{sys} is the system temperature, A_e is the effective collecting area, df is the frequency channel width, and τ is the single snapshot observation time. We can easily plug in true observational parameters to give us the rms for individual visibilities.

$$\sigma_v = 2761.3006 \left(\frac{T_{sys}}{\text{K}} \right) \left(\frac{\text{m}^2}{A_e} \right) \left(\frac{\text{Hz}}{df} \right)^{1/2} \left(\frac{\text{sec}}{\tau} \right)^{1/2} \text{Jy} \quad (2)$$

For example, if we have $T_{sys} = 440$ K, $A_e = 16$ m², $df = 40$ kHz, and $\tau = 5$ seconds, I get $\sigma_v \approx 169.8$ Jy.

Ian uses the model beam to grid the data (or the weights) into a regular uv grid. As an example, let's look at a simplified case of two overlapping visibilities, and consider the gridded value and variance of a pixel in the overlapping region.

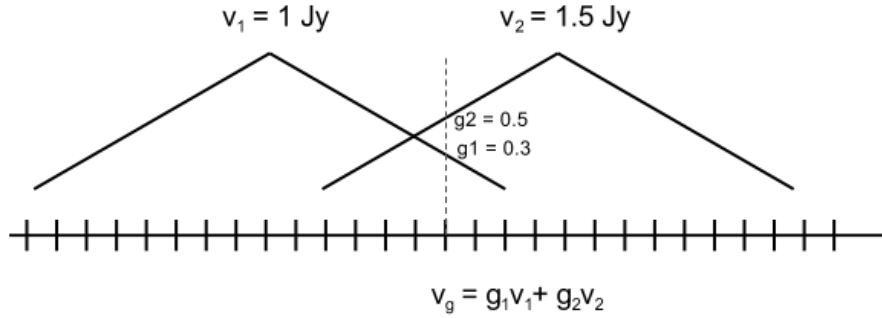


Figure 1: Gridding two visibilities

Each visibility (v_1 and v_2) is multiplied by the beam value at the pixel in question (g_1 and g_2) and added to produce the gridded visibility.

$$v_g = g_1 v_1 + g_2 v_2 \quad (3)$$

The variance on v_g can easily be calculated using error propagation.

$$\sigma_g = \sqrt{(g_1 \sigma_1)^2 + (g_2 \sigma_2)^2} \approx \sigma_v \sqrt{g_1^2 + g_2^2} \quad (4)$$

In the case that our antennas are identical, all the values in Eq. 1 are identical, so we can replace σ_1 and σ_2 with a common σ_v .

[Adding snapshots in healpix... need to figure this out...]

Ignoring the healpix business for now, we want to know what happens to our noise when we divide by the weights in (u, v, f) space. The “weights” we use are the sum of beam models in each pixel. In our simple example,

$$w_g = g_1 + g_2. \quad (5)$$

This weight is an accurate description of how well we sampled each pixel on the uv plane. Our best estimate of the apparent sky is then

$$\hat{v} = \frac{v_g}{w_g} = \frac{g_1 v_1 + g_2 v_2}{g_1 + g_2}. \quad (6)$$

Again we can simply propagate error.

$$\sigma_{\hat{v}} = \frac{\sigma_g}{w_g} = \sigma_v \frac{\sqrt{g_1^2 + g_2^2}}{g_1 + g_2} \quad (7)$$

From here we can form our covariance matrix and propagate through the frequency Fourier transform and square to power spectrum as Bryna has been doing.

Conclusion: To me it seems like we need both the weights cube ($g_1 + g_2$ in the example above) as well as a cube where the beams were squared before the gridding ($g_1^2 + g_2^2$) in order to propagate the rms correctly.

2 Covariance between grid points

Because the same data is being gridded to multiple grid points, this gridded data is highly covariant. Here I will quantize this covariance.

Consider two grid points, indexed with i and j , with n visibilities contributing. In general the visibilities contributing to both grid points do not need to be identical (we are interested in cases where there is some overlap), but let n represent the union of the two sets of visibilities. Then we have the gridded visibilities,

$$v_i = \sum_n g_{n,i} v_n \quad (8a)$$

$$v_j = \sum_n g_{n,j} v_n \quad (8b)$$

We wish to calculate the covariance.

$$C_{ij} = \langle (v_i - \bar{v}_i)(v_j - \bar{v}_j)^* \rangle \quad (9)$$

Angled brackets ($\langle \dots \rangle$) represent an expectation value, and a bar is shorthand for the same. We can simplify this equation slightly before proceeding.

$$C_{ij} = \langle v_i v_j^* - v_i \bar{v}_j^* - \bar{v}_i v_j^* + \bar{v}_i \bar{v}_j^* \rangle \quad (10a)$$

$$= \langle v_i v_j^* \rangle - \bar{v}_i \bar{v}_j^* - \bar{v}_i v_j^* + \bar{v}_i \bar{v}_j^* \quad (10b)$$

$$= \langle v_i v_j^* \rangle - \bar{v}_i \bar{v}_j^* \quad (10c)$$

Now we plug in Eq. 8, and group terms appropriately.

$$C_{ij} = \left\langle \sum_n g_{n,i} v_n \sum_m g_{m,j}^* v_m^* \right\rangle - \sum_n g_{n,i} \bar{v}_n \sum_m g_{m,j}^* \bar{v}_m^* \quad (11a)$$

$$= \sum_n \sum_m (g_{n,i} g_{m,j}^* \langle v_n v_m^* \rangle - g_{n,i} g_{m,j}^* \bar{v}_n \bar{v}_m^*) \quad (11b)$$

$$= \sum_n \sum_m g_{n,i} g_{m,j}^* C_{nm} \quad (11c)$$

We have now expressed our gridded covariance matrix in terms of the raw visibility covariance matrix. Thus far we have made no assumptions about the covariance of the visibilities, however, in order to simplify further I will assume that this covariance matrix is diagonal and all visibilities have the same variance (as we assumed above). Then we have the result,

$$C_{ij} = \sigma_v^2 \sum_n g_{n,i} g_{n,j}^*. \quad (12)$$

Now the sum over n can be restricted to the intersection of visibilities that contribute to *both* grid points. It is nice to see that in the case $i = j$, this result conforms to Eq. 4. Additionally, this is exactly the Holographic Mapping function used in FHD, which Ian already computes for every snapshot.

Conclusion: The covariance matrix between grid points for a single snapshot is given simply by the Holographic Mapping function used in FHD. In principle we should be able to transform this function to the image space, add in healpix, and transform back in the same way we treat the data. The difficulty is the size of this matrix. It is sparse, which should help greatly in the transformation to image space, but it is not sparse there, and the transform back may be difficult. In addition, this adds to the work needed for the Fourier transform over frequency when calculating power spectra.

3 Error propagation to power spectrum

In the next steps we wish to arrive at a power spectrum measurement, using two interleaved data cubes along with their variances. I will assume we performed the frequency Fourier transform on the data and propagated the covariance cubes by applying the Fourier transform matrix on both sides. In this space, we have cubes $a_1(u, v, \eta)$ and $a_2(u, v, \eta)$ where 1 and 2 correspond to the two interleaved cubes. We also have b_1 and b_2 for the orthogonal mode (e.g. sine and cosine), but they are separated in most of the math, so I will focus on a 's for now.

The first thing we need to do is add and subtract the two cubes, accounting for the weights. From here on out, I will assume the “weights” are the inverse of the variance, rather than the gridded constants from the previous section.

$$a_{sum} = \frac{w_{a1} a_1 + w_{a2} a_2}{w_{a1} + w_{a2}}; \quad a_{sub} = \frac{w_{a1} a_1 - w_{a2} a_2}{w_{a1} + w_{a2}} \quad (13)$$

The variance of the results are

$$\sigma_{sum}^2 = \sigma_{sub}^2 = \frac{1}{w_{a1} + w_{a2}}. \quad (14)$$

Because all we have done thus far is scale by weights and add, I believe we can perform this step in the (u, v, f) space due to the linearity of the Fourier transform. This should allow us to avoid translating two weights cubes to (u, v, η) space.

The next step is to perform a magnitude squared on both a_{sum} and a_{sub} . The math is a little messy, so I will demonstrate it for an arbitrary complex number drawn from a symmetric gaussian distribution. We start with a probability distribution function for a number c .

$$\Pr(c) = \frac{1}{\pi\sigma^2} e^{-|c-\mu|^2/\sigma^2} \quad (15)$$

Here μ is the mean complex value of the distribution, and σ is the standard deviation of the *complex number* c . It is important to distinguish this σ from the standard deviation of the real and imaginary parts separately (which would be a factor of $\sqrt{2}$ smaller). Also note that the visibility rms given at the beginning of the last section is actually the real/imaginary version. So things aren't exactly perfect from here...

Next we find the pdf of $|c|$ which is found by integrating a ring of constant magnitude,

$$\Pr(|c|) = \int_0^{2\pi} d\phi \Pr(c)|c| = \frac{2|c|}{\sigma^2} e^{-(|c|^2+|\mu|^2)/\sigma^2} I_0\left(\frac{2|c||\mu|}{\sigma^2}\right) \quad (16)$$

where I_0 is the 0th modified Bessel function. A change of variables takes us to the pdf for $|c|^2$.

$$\Pr(|c|^2) = \frac{d|c|}{d|c|^2} \Pr(|c|) = \frac{1}{\sigma^2} e^{-(|c|^2+|\mu|^2)/\sigma^2} I_0\left(\frac{2|c||\mu|}{\sigma^2}\right) \quad (17)$$

The stats we care most about here are the expectation value of $|c|^2$ and the variance of $|c|^2$. They are

$$\langle |c|^2 \rangle = |\mu|^2 + \sigma^2 \quad (18a)$$

$$\text{var}(|c|^2) = \sigma^2 (2|\mu|^2 + \sigma^2). \quad (18b)$$

Back to the sum and difference cubes, we can now determine the expected values and variances of the magnitudes squared (taking S to be the signal, and σ^2 is the inverse of the respective weight).

$$\langle |a_1 + a_2|^2 \rangle = |S|^2 + \sigma_{sum}^2 \quad (19a)$$

$$\langle |a_1 - a_2|^2 \rangle = \sigma_{sub}^2 \quad (19b)$$

$$\text{var}(|a_1 + a_2|^2) = \sigma_{sum}^2 (2|S|^2 + \sigma_{sum}^2) \quad (20a)$$

$$\text{var}(|a_1 - a_2|^2) = \sigma_{sub}^4 \quad (20b)$$

We are almost there. Because our sum term has noise in its expected value, we must subtract the pure noise term to arrive at a best estimate of the signal.

$$|\hat{S}_a|^2 = |a_1 + a_2|^2 - |a_1 - a_2|^2 \quad (21)$$

Note that even though the two terms can have very different variances, this should *not* be a weighted sum, or the noise would not subtract properly. The variance is

$$\sigma_{S,a}^2 = \text{var}(|a_1 + a_2|^2) + \text{var}(|a_1 - a_2|^2) = 2\sigma_{sum}^2 (|S|^2 + \sigma_{sum}^2) \quad (22)$$

where I have used the fact that $\sigma_{sum} = \sigma_{sub}$ to simplify slightly. At first look I thought these terms would be covariant and the error propagation would not be so easy. However, simple simulations have shown that the covariant terms are negligible. I have not fully understood this, but I think it has to do with the particular linear combinations being independent (similar to changing bases in vector space).

Finally we can include the b terms to help our estimate slightly.

$$|\hat{S}|^2 = \frac{w_{S,a} |\hat{S}_a|^2 + w_{S,b} |\hat{S}_b|^2}{w_{S,a} + w_{S,b}} \quad (23)$$

With a variance on our final estimate being

$$\sigma_S^2 = \frac{1}{w_{S,a} + w_{S,b}}. \quad (24)$$

4 PDF after averaging bins

In this section I will describe the probability distribution function when we square complex numbers, then average bin (rings for 2D PS, shells for 1D PS). As in the last section, I will treat the two orthogonal modes in a k bin separately (e.g. sines and cosines). They are combined in the averaging step with all the other modes in the bin.

We start with the PDF of a symmetric complex gaussian distributed number. Some of this is repeated from above, but I wish to put everything together. To conform notation to what Bryna uses in her code, σ will now represent the uncertainty on the real and imaginary parts individually. I will also drop the mean value because we are currently particularly interested in the noise cubes.

$$\text{Pr}(c) = \frac{1}{2\pi\sigma^2} e^{-|c|^2/2\sigma^2} \quad (25)$$

Taking magnitude squared of c gives us an exponential distribution with parameter $\lambda = 1/2\sigma^2$.

$$\Pr(|c|^2) = \frac{1}{2\sigma^2} e^{-|c|^2/2\sigma^2} \quad (26)$$

The expectation value and variance of this distribution are $1/\lambda = 2\sigma^2$ and $1/\lambda^2 = 4\sigma^4$ respectively.

The next step is to perform a weighted average in either rings or spherical shells. We assign the above distribution for each voxel i and weight by the inverse variance to sum. Here we include the sine and cosine modes in the sum.

$$\overline{|c|^2} = \frac{\sum \lambda_i^2 |c_i|^2}{\sum \lambda_i^2} \quad (27)$$

The expectation value and variance of the weighted sum are given by

$$\langle \overline{|c|^2} \rangle = \frac{\sum \lambda_i^2 \langle |c_i|^2 \rangle}{\sum \lambda_i^2} = \frac{\sum \lambda_i}{\sum \lambda_i^2} \quad (28a)$$

$$\text{var}(\overline{|c|^2}) = \frac{1}{\sum \lambda_i^2}. \quad (28b)$$

The sum of exponentially distributed numbers with equal parameters yields an Erlang distributed number. The weighted sum does not exactly yield an Erlang distribution, but if the weights are reasonably uniform (within an order of magnitude of one another), we can well approximate the resulting distribution as Erlang with effective parameters. The PDF of an Erlang is given by

$$\Pr(z) = \frac{z^{n-1} e^{-z/\beta}}{\beta^n \Gamma(n)} \quad (29)$$

where n would be the number of voxels that went into the average if all weights were the same, and β would be $1/\lambda$. Because our weights are not all the same some voxels contribute more than others, which leads to an effective number that went into the average less than the true number, and a variance more heavily weighted by the small variance voxels. We can solve for the effective parameters by writing down the mean and variance of this distribution, and setting them equal to those for our weighted average.

$$\langle z \rangle = n_{\text{eff}} \beta_{\text{eff}} = \frac{\sum \lambda_i}{\sum \lambda_i^2} \quad (30a)$$

$$\text{var}(z) = n_{\text{eff}} \beta_{\text{eff}}^2 = \frac{1}{\sum \lambda_i^2} \quad (30b)$$

Solving...

$$n_{\text{eff}} = \frac{(\sum \lambda_i)^2}{\sum \lambda_i^2} \quad (31a)$$

$$\beta_{\text{eff}} = \frac{1}{\sum \lambda_i} \quad (31b)$$

We can test this parameterized fit with some simulated data. In Figure 2 we plot a histogram of simulated data obtained from squaring and averaging complex random numbers (ie. starting at the beginning of this section).

The fit accurately predicts the shape of the distribution for a few voxels as well as for many. Where the fit breaks down is when the weights that go into the average are allowed to vary significantly. In the figure titles, the quantity ‘sigma SNR’ refers to the average sigma divided by the standard deviation of the sigmas that went into the original simulation. As this number decreases, the predicted distribution (red line) deviates from the simulated data. In particular, we see in the last plot, when the sigma SNR is about 3 that the predicted distribution is not as peaked as the simulation.

More investigation is needed to determine if the voxels going into our averages will satisfy the large sigma SNR condition. However, this method can be tentatively used to predict the noise power and uncertainty in EoR data.

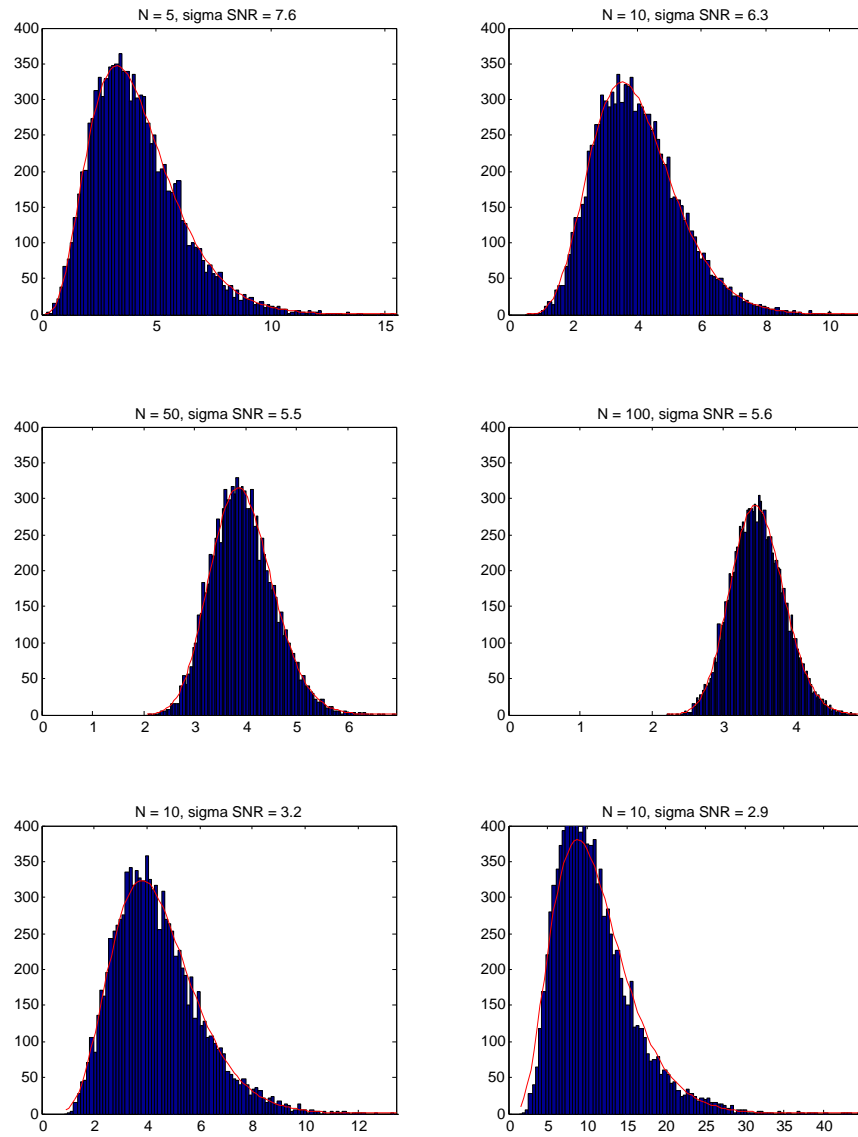


Figure 2: Simulated noise data against fit Erlang distributions