The early universe was a hot dense plasma where photons could not easily travel. At the photons in the early universe bounced around and entered thermal equilibrium with their surroundings until 380,000 years after the big bang, when the universe had expanded and cooled down enough for protons and electrons to combine to make neutral Hydrogen. At this point, the universe became transparent and the photons were free to move about. These photons make up what we call the Cosmic Microwave Background (CMB). The CMB gives us a thermal picture of what the early universe looked like. We can also see from the CMB how all the light in the early universe was cooled down enough for protons and electrons to combine to make neutral Hydrogen.

Cosmic inflation theory postulates that there was a period in the evolution of the universe in which extremely fast expansion occurred. This expansion would have sent gravitational waves through space, which would leave a B-mode signature in the CMB. By looking for B-polarized light in the CMB, we can establish evidence for or against inflation theory.

One way to get around the issue of not being able to write down the filtering matrix is to introduce a “messenger field”, η, such that the covariance of the messenger field, η, is proportional to the identity matrix. This allows us to write down η in any basis, and thus we can write the equation for a wiener filtered map, s, in an iterative way as:

\[
\left(K_{\ell}^{-1} + (\Delta T)^{-1}\right)^{-1} = \left(K_{\ell}^{-1} + (\Delta T)^{-1}\right)^{-1} s
\]

where \( K_{\ell} = \mathbf{N} - \mathbf{T} \). \( \mathbf{T} \) is a scalar which we use to speed up convergence, and \( \mathbf{N} \) is the input data. The first equation above is solved for \( \mathbf{s} \) in the pixel basis and the second is solved for \( \mathbf{r} \) in the spherical harmonic basis. Iteratively solving these equations for \( \mathbf{r} \) while decreasing \( \mathbf{\Delta T} \) from some large value down to 1 allows us to filter out both noise and E-modes.

In order to find the E to B leakage, we simulated E-mode only CMB maps containing power at a given f and measured how many E-modes had leaked to B-modes. B-mode only maps were simulated for BB2BB band window functions. This work is summarized in the plots below. These band power window functions tell us, for a particular bin, how much power is allowed through the filter at each f. So, multiplying the EE2BB band power window functions by the theoretical EE CMB power spectrum, then summing over all f’s, gives us the expected EE2BB leakage in each bin from the CMB. The ratio of BB2BB band powers to EE2BB band powers tells us about the purity of the filter.

The basic idea of our work is to apply a Wiener filter to a CMB data set. The Wiener filter serves as a method of noise filtering in data sets. A Wiener filtered map is defined to be \( m_{\ell} = m_{\ell} + \mathbf{N}_{\ell} \), where \( m \) represents the noise covariance. \( m_{\ell} \) represents the noise covariance of the data set and \( m \) represents the noisy data. If signal and noise are both Gaussian (which seems to be the case for the CMB), the Wiener filter maximizes the a-posteriori probability of a map given \( m, s, \) and \( w \). Any kind of light has what we call an E-mode and a B-mode component (shown in figures below). To find just the B-mode signal in the CMB, we treat the E-mode part of the signal covariance as if it was noise covariance so the wiener filter equation becomes:

\[
\mathbf{N}_{\ell} \approx (\Delta T)^{-1} \mathbf{T}_{\ell}
\]

We cannot write down the filtering matrix directly, however, because the signal and noise covariance matrices are sparse in difference bases. Our work implements the Messenger Method. Another common way to represent data is with a power spectrum. There are a few different ways we can choose to represent CMB data. The most straightforward way is to represent it in the pixel domain, with a map. We can also transform this map into the spherical harmonic domain using algorithms from the python package Healpy. We can also transform this map into the spherical harmonic domain using algorithms from the python package Healpy.