SURF Final Report - Time-Ordered CHIC Simulator

Joheen Chakraborty

Columbia University

jc5110@columbia.edu

Mentors: Tzu-Ching Chang & Aaron Ewall-Wice

November 9, 2022

1 Introduction

CHIC (Cosmic Hydrogen Intensity Constellation) is a planned upcoming 21 cm cosmology experiment with the unique goal of measuring the dipole of the global 21 cm signal. The 21 cm spectral line originating from the hyperfine spin-flip transition of neutral hydrogen is optically thin and insensitive to astrophysics, thus making it an exquisite probe of cosmological history (Pritchard and Loeb, 2010). There are already multiple experiments aiming to directly measure the monopole of the "global" (i.e. full-sky averaged) 21 cm brightness temperature; CHIC takes a slightly different approach by attempting to instead measure the dipole of this global signal, relying on the fact that differential measurements are both easier than and independent of of absolute measurements (Slosar, 2017). A successful measurement of the global 21 cm dipole signal would provide an independent confirmation of the various cosmological parameters that shape the monopole.

My project this summer has been creating a time-ordered simulation code to calculate the properties and merits of different orbit/telescope/beam patterns for the upcoming experiment. To do this, I relied on decomposition to a spherical harmonic basis, and then used the coupling matrices of the time-integrated maps to determine these properties and merits. Now that the coupling matrix pipeline has been developed, it can be used to evaluate the many different observing strategies that will be considered for CHIC.

In this report, I will (I) provide some background on the mathematical formalism used to analyze cosmological data; (II) describe the steps of the pipeline and provide documentation of the different functionalities; and (III) make some remarks on next steps and future directions for the project.

2 Formalism

2.1 The 21 cm dipole signal

The following discussion on the dipole of the global 21 cm signal is adapted closely from Slosar (2017). The motion of the Earth through the cosmic rest frame modulates the global signal monopole into a dipole, via (I) frequency-independent boosting by a factor of v/c and (III) blueshifting of photons by a factor of $1 + v/c$. If we decompose the monopole signal into is frequency dependent and independents parts:

$$T_{mono}(\nu) = T_0 + \Delta T(\nu)$$

and then consider the effect of Doppler shifting on the observed signal:

$$\begin{aligned} T_{obs}(\theta, \nu) &= (T_0 + \Delta T(\nu - \delta\nu))(1 + \delta\nu/\nu) \\ &= T_0 + \Delta T(\nu) + \left(T_0 + \Delta T(\nu) - \frac{d\Delta T}{d\nu}\nu\right)\frac{v_d}{c}\cos\theta \end{aligned}$$

where θ is the angle of the observer w.r.t. the direction of the Earth's motion through the cosmic rest frame. So, we can identify the dipole term with the final term on the RHS of the above:

$$T_{dip} = \left(T_0 + \Delta T(\nu) - \frac{d\Delta T}{d\nu}\nu\right)\frac{v_d}{c}\cos\theta$$

T_{dip} is the signal we are interested in for the CHIC experiment. Notably, T_{dip} is dominated by the $\frac{d\Delta T}{d\nu}$ term (Fig. 1)

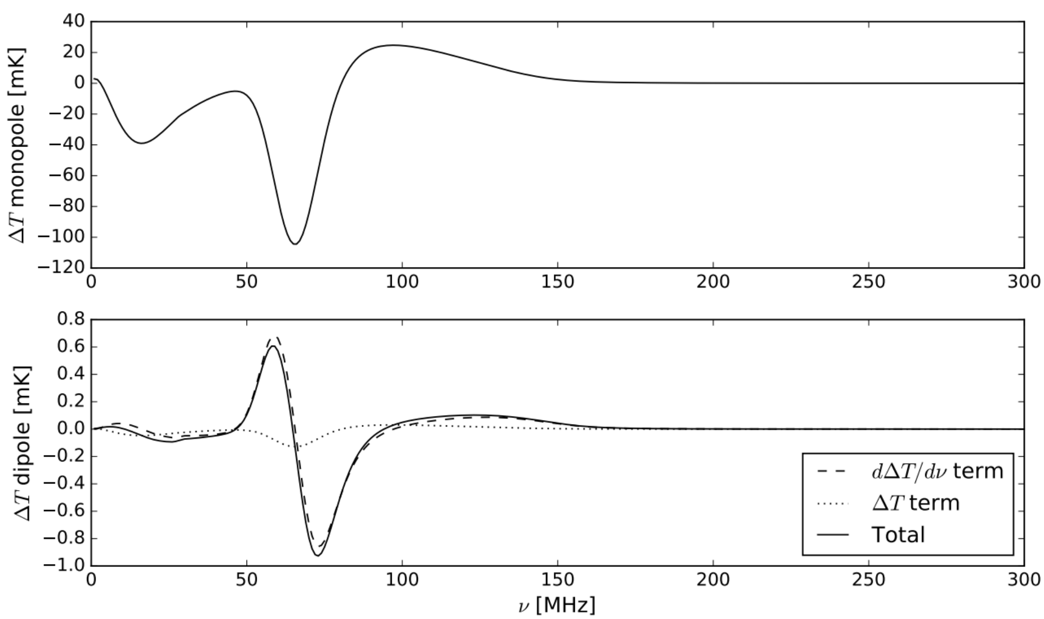

Figure 1: Epoch of Reionization monopole ("global" 21 cm signal) and dipole. Figure taken from Slosar (2017).

While measurement of this dipole signal would not provide unique information about the cosmological parameters which shape the monopole (e.g. star forming rate, efficiency of X-ray sources, cooling channels, etc.), it would provide a reliable and independent measurement source with which to compare the results of more conventional experiments targeting the global signal.

2.2 Spherical Harmonic Basis

Cosmological data is frequently analyzed by decomposing into a spherical harmonic basis. This is an attractive option, as the data is usually measured across the entire sky rather than isolated to point sources, and is also quite voluminous; therefore, an orthonormal basis defined on the surface of a sphere (the sky) is a reliable option to use. For a particular frequency ν, a telescope would measure power on the sky as:

$$P(t) = \int_\Omega B(\theta, \phi, t) I(\theta, \phi) d\Omega$$

where for a particular location on the sky (θ, ϕ), $B(\theta, \phi, t)$ is the beam pattern at a particular time during the orbit t and $I(\theta, \phi)$ is the "intensity" (brightness temperature) of the sky at that location, taken to be constant throughout the duration of the experiment. Then, if we decompose to spherical harmonics:

$$\begin{aligned} P(t) &= \int_\Omega \Big(\sum_{lm} B_{lm}(t) Y_{lm} \Big) \Big(\sum_{l'm'} a_{l'm'} Y_{l'm'} \Big) d\Omega \\ &= \sum_{lm} B_{lm}(t) a_{lm} \end{aligned}$$

where we have taken advantage of the orthonormality and completeness of the spherical harmonics to eliminate the integral over the full sky. Then, we can imagine $P(t)$ as a $t \times 1$ vector, and the sum on the RHS becomes a matrix multiplication:

$$P = BA$$

where B is the $t \times lm$ matrix whose rows contain the spherical harmonic coefficients describing the beam for t times, and A is the $lm \times 1$ vector describing the spherical harmonic coefficients of the sky.

However, time-ordered data can be prohibitively voluminous, and so we turn to the convention of cosmological map-making. Here, we eliminate the time-stream information by choosing a gridding kernel K which assigns telescope-measured power at each time step t to the map. Intuitively, the optimal gridding kernel to use is the beam (formally, the transpose of the beam B^T), which then distributes power according to the strength of the telescope beam at each location. Thus we have:

$$\begin{aligned} M &= (B^T B) A \\ &\coloneqq CA \end{aligned}$$

where we designate the quantity $B^T B$ as the "coupling matrix", thus named because it couples true spherical harmonic coefficients on the sky to those measured in our map M. M is now an $lm \times 1$ vector containing data in a spherical harmonic basis rather than a time-ordered basis, thus making analysis computational feasible. Now the expensive step of the operation is calculation of C, which only needs to be done once for a particular orbit.

C contains a great deal of interesting information about the merit of an observing strategy. For example, if C is not full-rank or has a high condition number, then the observing strategy contains incomplete information about the chosen spherical harmonic basis. If two rows (equivalently columns, as using B^T as the gridding kernel results in a symmetric coupling matrix) have a high correlation, then they will have a similar effect on the resulting map M, and thus the observing strategy will have a poor ability to disentangle these two spherical harmonic coefficients in the sky. For instance, if we operate in a coordinate system such that the cosmological dipole is exactly in the $+z$ direction (i.e. corresponding solely to the a_{10} term), then one important metric for merit will be that the a_{10} term has a low correlation with all other spherical harmonic coefficients which have non-negligible components in the sky.

3 Simulator design and documentation

In this section, I will describe the time-ordered simulator I developed during the summer to calculate the spherical harmonic coupling matrix for a given orbit. The code for this simulator is available at github.com/joheenc/chic_coupling.

3.1 Design

To simulate the sky coverage of a given observing strategy, I start by assuming a gaussian beam pattern, which is easy to describe analytically rather than resorting to working with numerical rotation of HEALpix maps. Recall a 2D gaussian is given by the equation:

$$f = \frac{1}{2\pi\sqrt{|\mathbf{\Sigma}|}} \exp\left\{\left\{ -\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}\right\} \tag{1}$$

For us, the "data" vectors $\mathbf{x}$ will be the angular coordinates on the sphere $\theta \in [0, \pi]$, $\phi \in [-\pi, \pi]$. One critical simplification we can make to greatly improve computational cost is realizing that, wlog, our $\mathbf{\Sigma}$ can always be chosen to be diagonal. Any beam with off-diagonal covariance terms can be represented as a rotation about the beam axis of a diagonal $\mathbf{\Sigma}$. Then we can rewrite (1), for some (θ, ϕ) coordinate i, as:

$$
\begin{aligned}
f_i &= \frac{1}{2\pi\sqrt{\Sigma_{\theta\theta}\Sigma\phi\phi}} \exp\left\{\left\{ -\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) \right\}\right\} \\
&= \frac{1}{2\pi\sqrt{\sigma_\theta\sigma_\phi}} \exp\left\{\left\{ -\frac{1}{2}\begin{pmatrix}\theta_i & \phi_i\end{pmatrix}\begin{pmatrix}\sigma_\theta^{-1} & 0 \\ 0 & \sigma_\phi^{-1}\end{pmatrix}\begin{pmatrix}\theta_i \\ \phi_i\end{pmatrix} \right\}\right\} \\
&= \frac{1}{2\pi\sqrt{\sigma_\theta\sigma_\phi}} \exp\left\{\left\{ -\frac{1}{2}\left(\frac{\theta_i^2}{\sigma_\theta} + \frac{\phi_i^2}{\sigma_\phi}\right) \right\}\right\}
\end{aligned}
$$

Now that the θ_i and ϕ_i terms have been separated, this equation can be easily vectorized in NumPy for a speedup of 2600x.

To test different orbits for a given beam, I use the simplified perturbations model SGP4 to calculate orbital state vectors, the errors of which remain small on the timescales of months to a year. Input for the model is given in the form of a satellite two-line element (TLE) set, which contains information about the classical orbital elements (eccentricity, argument of periastron, etc.) of a given orbit around the Earth. The whole pipeline, starting from the satellite TLE and ending with computation of the coupling matrix, takes the following steps:

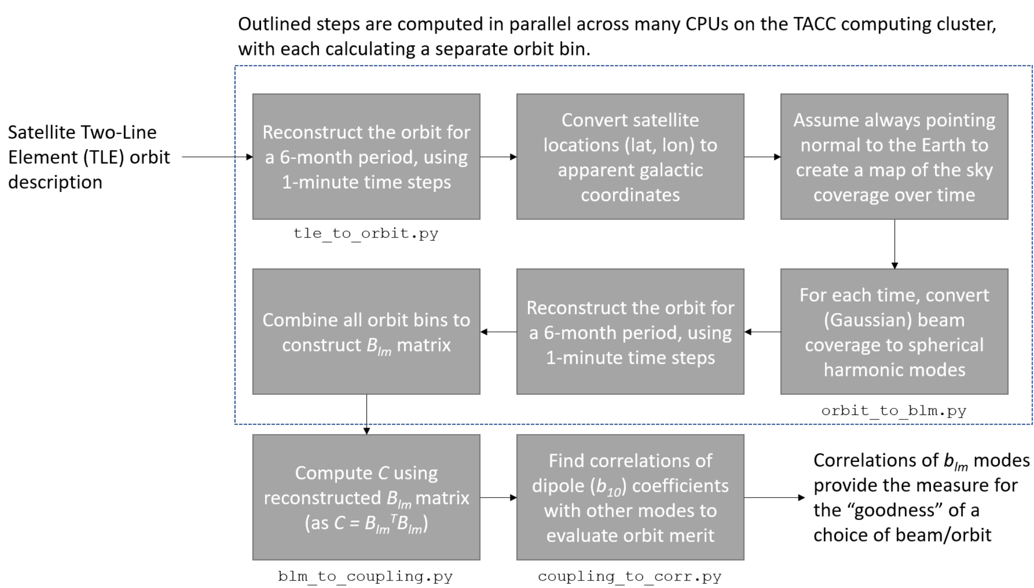

Figure 2: Flowchart visualization of the coupling matrix code pipeline. Corresponding python scripts (located in the GitHub repository linked above) are given along with the steps they compute. CPU parallelization is used for a speedup of 300x; with Healpix NSIDE=64, angular resolution (for Gaussian beam calculation) of 1°, and $l_{\max} = 100$, the entire pipeline takes 1.5 hrs.

3.2 Documentation

In this subsection I will list each of the functions available in the simulation code and give an example of how to run it on a multi-CPU computing cluster.

`compute_B`

Convert the sky coverage of a HEALpix map from pixel space to a spherical harmonic basis. Usage is `compute_B(skycov, lmax=LMAX)` where `skycov` is the HEALpix map containing the time-integrated sky coverage. Returns the matrix of spherical harmonic coefficients as a 2D complex NumPy array.

`compute_corr`

Compute the correlations of every row (equivalently, every column) within a matrix with a designated "reference" row. Usage is `compute_corr(C, lref=1, mref=0, lmax=LMAX)` where C is the matrix and lref, mref give the spherical harmonic indices of the reference row. Returns the correlations as a 1D NumPy array.

`gal_to_dipole`

Rotate a HEALpix map (in pixel space) from the galactic coordinates frame to the "dipole frame", i.e. a frame with the cosmic velocity dipole in the +z direction. Usage is `gal_to_dipole(hpxmap)`, where hpxmap is the HEALpix map to be rotated. Returns the rotated HEALpix map in pixel space as a 1D NumPy array.

`orbit_from_tle`

Use the satellite two-line element (TLE) set to calculate the satellite's positions at each discrete time-step during an interval, in equatorial coordinates. Uses the SGP4 simplified perturbations model. Usage is `orbit_from_tle(tle1, tle2, starttime, stoptime, tstep=6.944e-4)` where `tle1`and `tle2` are the first and second lines of the TLE set to be calculated between `starttime` and `stoptime` (given in MJD). Returns two 1D NumPy arrays corresponding to the RA and Dec of the satellite at each time step.

`point_beam`

Generate a HEALpix map (in pixel space) of a gaussian beam pointed at a designated location on the sphere. Usage is `point_beam(theta, phi, beamthetastd=0.3, beamphistd=0.3, nres=400, nside=32, invert=True, savefile=None)` where (theta, phi) give the colatitude and azimuthal angle in radians describing the gaussian centroid of the pointed beam; `(beamthetastd, beamphistd)` are the respective standard deviations of the gaussian beam in radians; `nres`**2 is the angular resolution (i.e. number of bins) used to calculate the value at each angular coordinate; and `invert` is a switch which toggles whether `phi` runs increasing from left-to-right (TRUE) or right-to-left (FALSE). Returns `beam, indices, normhpxmap` where `beam` contains the value of the beam in angular coordinates, `indices` contains the mapping from angular coordinates to HEALpix map indices, and `normhpxmap` contains the HEALpix map describing the pointed beam normalized from 0 to 1.

`rotate_beam`

Rotate a given beam in angular space, then represent it in HEALpix pixel space, from a starting direction (θ_1, ϕ_1) to a final direction (θ_2, ϕ_2). Optionally rotate the beam about its centroid/boresight axis through an angle ψ. Usage is `rotate_beam(beam, indices, hpxmap, theta1, phi1, theta2, phi2, psi=0, nres=400,` `nside=32, onlybeam=False)` where `beam, indices, hpxmap` are the respected beam in angular coordinate space, mapping from angular coordinates to HEALpix map indices, and HEALpix map returned by `point_beam` corresponding to the starting configuration of the beam; `theta1, phi1, theta2, phi2` are the starting and ending colatitudes and azimuthal angles in radians to be rotated through; `psi` is the angle about the beam centroid axis in radians to rotate through; and `nres`**2 is the angular resolution (i.e. number of bins) used to calculate the value at each angular coordinate. `onlybeam` should be ignored, as it is an internal parameter used for the recursive step of the function. Returns `beam, normhpxmap` where `beam` contains the beam in angular coordinate space and `normhpxmap` contains the beam in HEALpix pixel space normalized from 0 to 1.

`sky_coverage`

Calculate the time-integrated sky coverage of a given orbit and beam configuration. Usage is `sky_coverage(lons, lats, psis, incoords='c', outcoords='g', beamthetastd=0.3, beamphistd=0.3, nres=400, nside=32, display=False)` where `lons, lats` are 1D NumPy arrays containing the longitude and latitude pointing directions at each time step in the astronomical coordinate system given by `incoords`; `beamthetstd, beamphistd` parametrize the size of the gaussian beam; and `display` is a switch which toggles whether the HEALpix map at each time step is shown (NOTE: setting `display=TRUE` will significantly increase the time taken and memory consumed by the function). Output is a 1D NumPy array containing the HEALpix map in pixel space corresponding to the time-integrated sky coverage of the combined orbit/beam pattern in the coordinate system given by `outcoords`. Two coordinate systems are supported: `['celestial', 'equatorial', 'c', 'C']` are all equivalent, as are `['galactic', 'g', 'G']`.

3.3 Usage

To use the pipeline starting from scratch in an empty directory, one should first clone the GitHub repository linked at the beginning of Section 3. Then, create the following empty directories used to store intermediate data produced by the pipeline: `blms_dipole, blms_eq, blms_gal, orbitfiles, skycov_eq, output`. Then create a `.txt` file containing the desired satellite TLE set, and change the `TLEFILE` parameter in `tle_to_blm.sh` to point to the file. Also in this script, you can modify the start and stop times and maximum spherical harmonic resolution to use. Note that if you change LMAX within `tle_to_blm.sh`, you must change it to match in `blm_to_corr.sh`. The pipeline can then be run on a multi-CPU cluster using e.g 300 concurrent CPUs as:

```
sbatch --share -a 0-299 tle_to_blm.sh
# Now WAIT until all 300 jobs finish running
sbatch --share blm_to_corr.sh # This step will take a long time
```

At the end of the pipeline ($\sim$ 90 minutes for LMAX 100), the `output` directory will be populated with the coupling matrix, correlations with a_{10}, resulting sky coverage map, and various other useful data about the observing configuration.

4 Results & Conclusion

Some preliminary results have been obtained using this pipeline to showcase its functionality, but the bulk of the work of testing different observing strategies and evaluating their relative merits remains for future work.

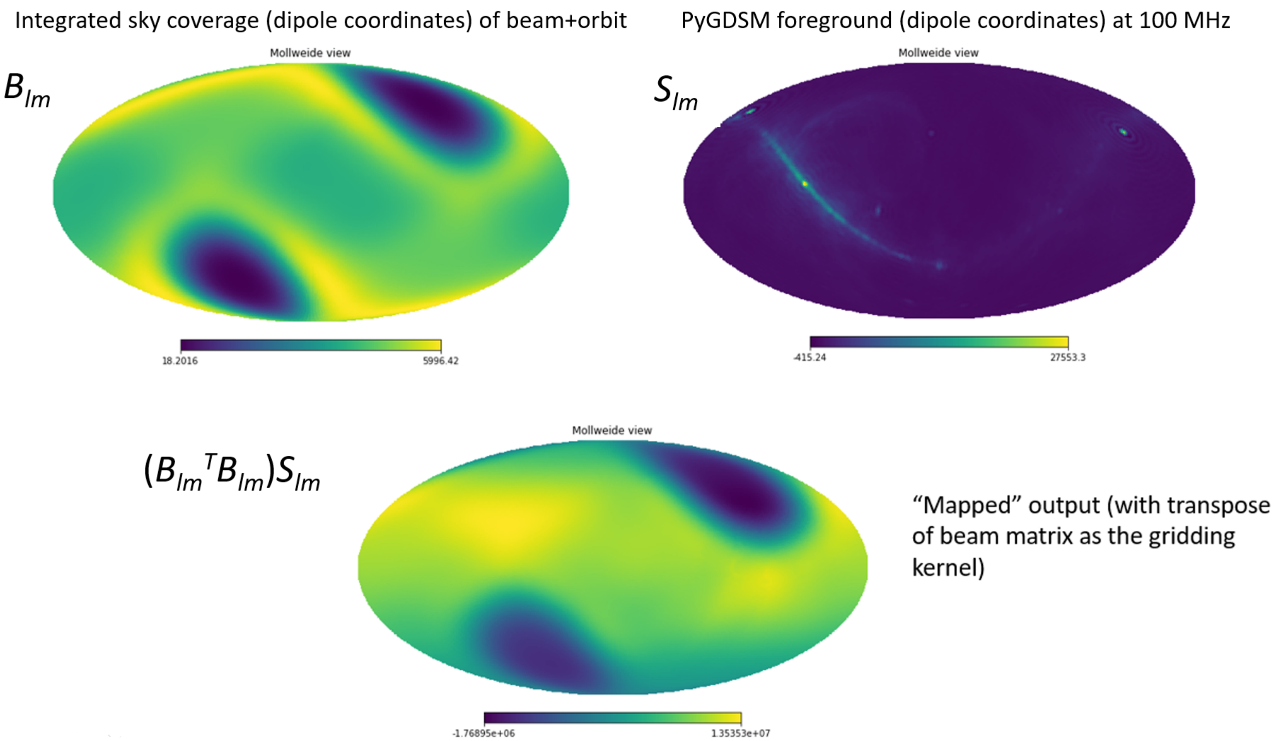

Figure 3: **Upper left:** time-integrated sky coverage of a Starlink-2172 orbit over a 6-month period, rotated to cosmic velocity dipole coortdinates. **Upper right:** Global Sky Model foreground (ν = 100 MHz) in the same frame. **Bottom middle:** "mapped" output, i.e. application of the coupling matrix to the sky in spherical harmonic space.

Fig. 3 contains some sample Mollweide-projected HEALpix maps of relevant quantities from the pipeline. The top left shows the time-integrated sky coverage over 6 months of a baseline very-low-Earth orbit tracing the Starlink-2172 satellite using a symmetric two-dimensional gaussian beam of std=0.3 rad. This map can be interpreted as the B_{lm} matrix, which we left-multiply by its transpose to get the coupling matrix. The top right then shows the PyGDSM foreground model of the sky at a frequency of 100 MHz. Both these maps are rotated to the "velocity dipole frame", i.e. a frame s.t. the cosmic velocity dipole points exactly in the +z direction. Then the dipole of interest is isolated to the a_{10} term, further demonstrated in Fig. 4. The bottom map of Fig. 3 shows the final mapped output which an experiment would measure; this is the ideal case, where we have applied the coupling matrix directly to the sky model. In reality, we would start from this map, and apply a pseudoinverse of the coupling matrix (constructed based on our appproximate knowledge of the beam and orbit patterns) to calculate our approximation of the sky.

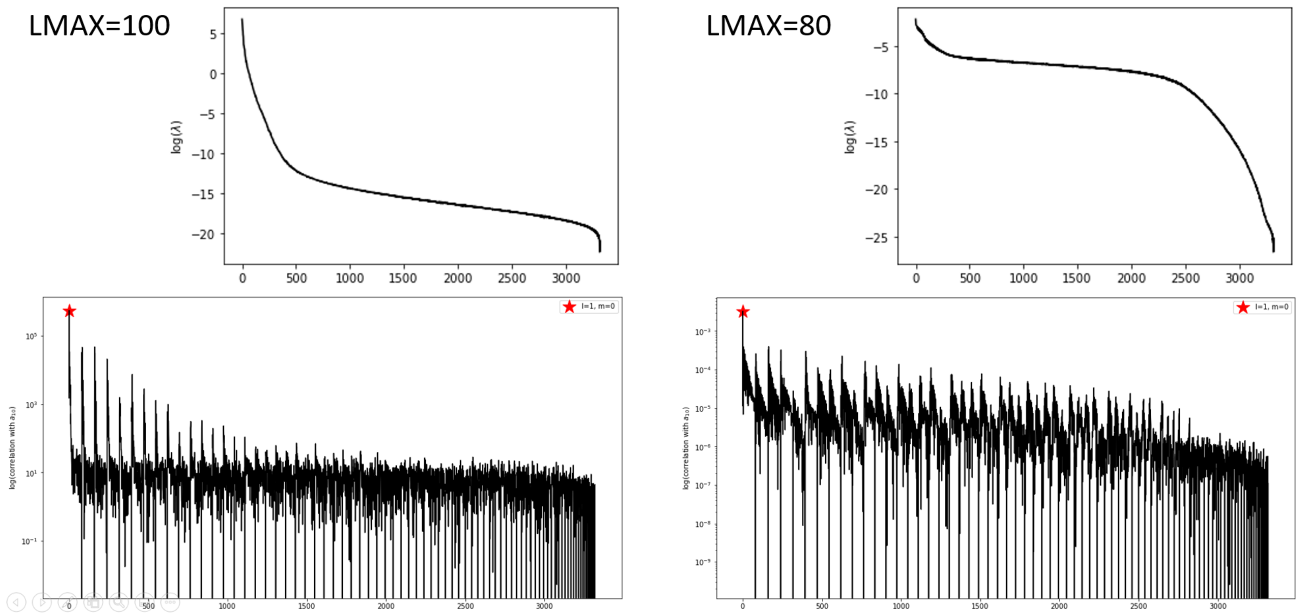

Figure 4: **Left:** $\log_{10}$ eigenspectrum and $\log_{10}$ correlations with a_{10} of the coupling matrix for LMAX=100. **Right:** the same quantities for an identical observing strategy and foreground model, but with LMAX=80.

Fig. 4 demonstrates some preliminary "metrics for merit", i.e. quantities that may be used to evaluate and compare different observing strategies. Of interest here are (I) the eigenspectrum of the coupling matrix and (II) the correlations with a_{10} of the other spherical harmonic coefficients in the coupling matrix. As we can see, decreasing LMAX from 100 to 80 dramatically alters both quantities, in a sort of "win-lose" scenario: the trade-off for achieving a more complete eigenspectrum (i.e. fuller rank/lower condition number denoting an observing strategy is better suited to reconstruct higher-order l-modes) is that there is greater "leakage" with the dipole coefficient, and the observing straegy is poorer at constraining the contribution from just the kinematic dipole. Such trade-offs must be carefully considered when evaluating and ultimately choosing an observing strategy for CHIC.

Future work on this project would progress in this direction, using the pipeline developed in this project to analyze the properties of various observing setups. Ultimately, this time-ordered simulator will prove crucial in determining the optimal parameters the CHIC experiment will utilize.

References

J. Pritchard and A. Loeb. Constraining the unexplored period between the dark ages and reionization with observations of the global 21 cm signal. *Physical Review D*, 82, 2010.

A. Slosar. Dipole of the epoch of reionization 21-cm signal. *Physical Review L*, 118, 2017.