HERA Summer Project
MIT 2022

HERA Memo #114
Received 23 Aug 2022

Spherical Harmonic Decomposition of HERA's Electromagnetic Primary Beam Simulations

Cynthia De Los Santos, Nicholas Kern, Jacqueline Hewitt

August 12, 2022

Kavli Institute for Astrophysics and Space Research,
Massachusetts Institute of Technology

Abstract

We fit spherical harmonics to electromagnetic simulations of the HERA Vivaldi primary beam (Fagnoni et al. 2021). Spherical harmonics are a natural basis for compression of the beam, and help us to understand the angular symmetries and structure of the beam. We test the accuracy of the reconstruction while varying the number of spherical harmonic modes, and perform the fitting on both the power and e-field beam models. To-date, analytic fits to HERA beam models have been limited to simplifying assumptions such as azimuthal averaging, working solely on the power beam, or extrapolating from nearby frequencies. Our complex-valued spherical harmonic fits show good agreement with the true HERA beam across a wide range of frequencies as well as for both the main lobe and far sidelobes (with errors at or below ~1%). Once the beam is compressed to a set of harmonic coefficients, we can evaluate the compressed beam at new zenith and azimuth angles fairly rapidly (i.e perform interpolation), with a run-time that is on-par (and in some cases faster) with the spline interpolation methods in pyuvdata UVBeam (accounting for the UVBeam.interp performance boost implemented in 2022). For the e-field beam, our SH interpolation is over a factor of 2 improvement in speed over UVBeam. In addition, a spherical harmonic compression significantly reduces the size of the beam in memory, which will be important for large N_antenna visibility simulations with antenna-dependent beam models.

HERA Summer Project
MIT 2022

Introduction

HERA and other 21 cm telescopes must have a precise understanding of the antenna primary beam response. The antenna primary beam is important for high dynamic range calibration, visibility simulation, and foreground subtraction. Knowledge of the beam can be derived in the field from empirical measurements of satellites or radio point sources with a known intrinsic flux. They can also be derived from electromagnetic simulations of the antenna front-end, including the feed and any surrounding material (such as the HERA dish). However, representing the beam in a discretized spherical coordinate space may be suboptimal, as we expect the beam response to be strongly correlated at neighboring angular coordinates. Therefore, compressing the beam response to a smaller dimensionality with a basis of intrinsically angularly smooth functions may prove beneficial to the applications listed above. Furthermore, our compressed basis may be a more memory-efficient way to store large quantities of beams, and may enable for faster angular interpolation.

Spherical harmonic expansion of antenna beam models has been studied in the literature for the MWA and for an SKA prototype (Sokolowski et al. 2017, Kriele et al. 2022). However, the usefulness of this technique has not yet been explored for HERA. We explore this in the context of electromagnetic simulations of the HERA antenna, outfitted with a 14-meter parabolic dish and a Vivialdi feed (Fagnoni et al. 2021). These simulations provide the complex, far-field response of the antenna for a single dipole feed for the $\hat{\theta}$ and $\hat{\phi}$ linear polarizations of incident radiation (what we call the E-field beam). These can be combined to form a “power beam” as

$$B^p = |B_1^e|^2 + |B_2^e|^2 \qquad (1)$$

where B^p is the real-valued and non-negative “power beam” and B_1^e is the complex-valued e-field beam for the $\hat{\theta}$ polarization and B_2^e is the complex-valued e-field beam for the $\hat{\phi}$ polarization. While recent works have achieved decent analytic fits to the HERA beam model, they have suffered from some simplifying assumptions such as azimuthal symmetry or extrapolation from nearby frequency channels (Choudhuri et al. 2020, Boyer et al. 2021), which have made accurate fitting of the complex beam across a wide frequency range in both the main lobe and sidelobes difficult.

Our goal in this work is twofold: 1. to test the tradeoff between accuracy and complexity when projecting HERA beam models onto spherical harmonics of increasingly large number of l and m modes, and 2. to test the speed of beam interpolation with spherical harmonics relative to existing spline methods in pyuvdata. Note that while electromagnetic modeling software such as FEKO or CST can compute and store the far-field beam pattern intrinsically as spherical harmonic modes, this is not what this work explores. Instead, we assume we are given a beam model that has been computed in discretized spherical coordinates, and we are concerned with fitting this beam model with our own spherical harmonics.

Beam Fitting Framework

Here we review the framework for generating the basis functions used to decompose the beam (the spherical harmonics), and the fitting algorithm used to solve for the harmonic coefficients of the beam. The ortho-normalized spherical harmonics can be written as

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}\, P_l^m(cos(\theta))\, e^{i\phi m} \qquad (2)$$

where P_l is the Associated Legendre polynomial, and l and m are the degree and order of the spherical harmonics, respectively. We implement this in Python code using the `scipy.special.lpmv()` function for the Legendre polynomial. For improved performance when computing the spherical harmonics for an array of l, m and an array of theta, phi, we reshape the theta and phi dimensions such that they are broadcasted across the l, m arrays, thus enabling us to compute the spherical harmonics for all l, m, theta, and phi in one function call (as opposed to setting up a FOR loop over l,m or theta,phi).

Our spherical harmonic function can be used to generate an (N_pixels, N_modes) matrix that takes the transformation from harmonic space to angular space. Calling this matrix A, we can write our linear model relating the harmonic coefficients to the beam as

$$y = A\,x \qquad (3)$$

where y is the beam in spherical coordinates of shape (N_pixels, N_frequencies), A is the spherical harmonics of shape (N_pixels, N_modes) and x are the harmonic coefficients of shape (N_modes, N_frequencies).

Given our linear model, we want to estimate the best-fit coefficients given our simulations of the beam, which we can achieve through the standard linear least squares inversion formula

$$\hat{x} = \left(A^H A\right)^{-1} A^H y \qquad (4)$$

where A^H denotes the conjugate transpose of A. Note that the inverse of $A^H A$ is not strictly possible, so instead we use a pseudo-inverse. Given our best-fit guess for the harmonic coefficients, we can forward model our best-fit beam in angular coordinates as

$$\hat{y} = A\hat{x} \qquad (5)$$

Note that in what follows, we only generate spherical harmonics for non-negative m modes (e.g. Figure 2). This is because we always cast the beam to real values before fitting it, meaning the negative m modes are redundant. To compensate for this, when taking the forward transform (Eq 5) we multiply all m > 0 modes by a factor of 2 before taking the transform. Note that we even do this for the complex-valued E field beam because we fit its real and imaginary components separately. Once we've compressed the beams into their harmonic coefficients $\hat{x}$, we can take the forward transform back to angular space at the same coordinates as the original data, or we can choose to evaluate them at entirely new angular coordinates. Doing this requires the construction of a new A matrix at the new locations on the sphere, which amounts to an interpolation scheme using spherical harmonics.

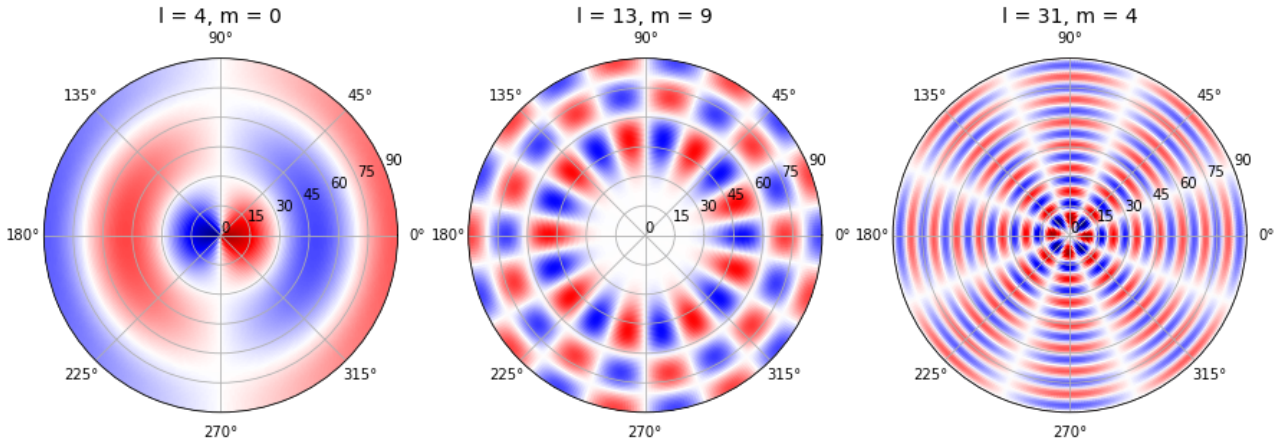

Figure 1 | This shows an example of some of the spherical harmonics we use to fit the beam. Note we only plot the real component but the functions are intrinsically complex. Also note that the left plot shows one of the modified m=0 spherical harmonic modes discussed below when fitting the E-field beams (which purposely breaks its inherent azimuthal symmetry), whereas for the power beams we use the standard m=0 spherical harmonic modes described in Eqn 2.

Fitting HERA Beams

We first load the HERA beam data from Fagnoni et al. 2021 using the pyuvdata.UVBeam interface. The beam data are 6 dimensional with the following axes: Naxes, Nspws, Nfeeds, Nfrequency, Ntheta, Nphi. Before fitting, we index the data for all frequencies, theta, and phi, and choose the zeroth index for Nspws and Nfeeds. The beam data are simulated on a uniform grid in zenith and azimuth with a 1 degree spacing and span 50 - 250 MHz with a 1 MHz spacing. Next we generate a spherical harmonic transformation matrix (A) for all angular coordinates of the beam for a set of l, m out to Lmax = 35. We chose Lmax = 35 after trying Lmax = 25 and finding that it was not quite sufficient to model the beam.

Power Beams

For the power beam, there is only one Naxes entry in the beam data. The power beam data are real-valued and non-negative. Note that before fitting the power beam we take its square root. We then use Eqn 4 to solve for the harmonic coefficients for each frequency channel. Figure 2 shows the log10 of the amplitude of these coefficients for two frequency channels.

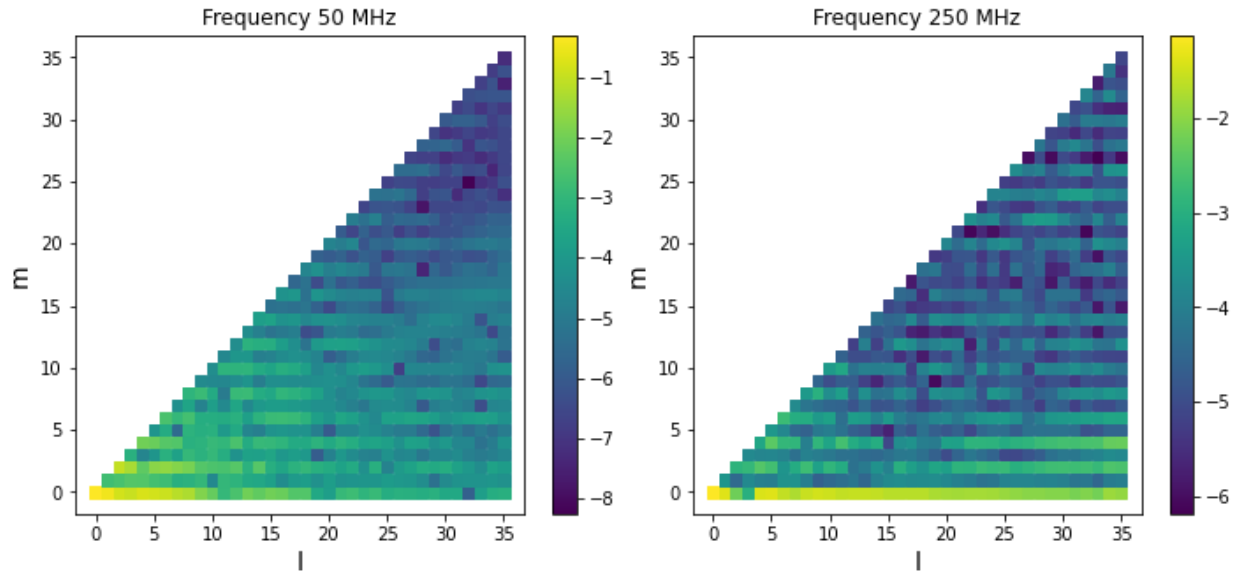

Figure 2 | Log10 of the harmonic coefficient amplitude for the power beam fits. Here we use an Lmax of 35. The left side graph is for 50 MHz frequency and the right is for 250 MHz frequency.

Suppression of the odd m modes implies a 180-degree rotational symmetry in the beam. While Figure 2 shows hints of this structure, the existence of non-zero odd m modes shows that there is not a true rotational symmetry in the HERA Vivaldi feed, which is likely due to the triangular support structure holding the feed that breaks the otherwise perfect 180-degree symmetry of the feed (Fagnoni et al. 2021). The dominance of the power in the bottom and lower-left portions suggests a natural way to reduce the dimensionality of the beam while retaining the majority of the features in the beam. This will motivate our choice for l, m cuts below.

To assess the goodness of the fit, we compare the true beam data against the best-fit beam data (Eqn 5) side-by-side in Figure 3. The left panel shows the true beam, the center shows the fitted beam, and the right shows the difference between the two.

To understand the tradeoff between fit accuracy and computational speed, we explore six different scenarios where we increase the number of modes in the spherical harmonic expansion. Case 1 has the lowest number of modes and therefore yields the most compression, is the fastest to evaluate, but is the least accurate in its beam reconstruction. Case 6 has the highest number of modes and is therefore the least amount of compression, slowest to evaluate, but is the most accurate (and corresponds to our fiducial example of all modes out to Lmax=35). For each case we set up an interpolation problem where we generate a new

spherical harmonic matrix for the theta, phi values of an NSIDE 64 HEALpix map (for theta < 90 degrees, yielding Npix=25000) given the unique set of l, m modes for the specified scenario. We use Eqn 4 and 5 to generate the best-fit beam at these new angular coordinates, and record the run-time (averaged over 7 runs). Profiling was performed on the NRAO CPU cluster.

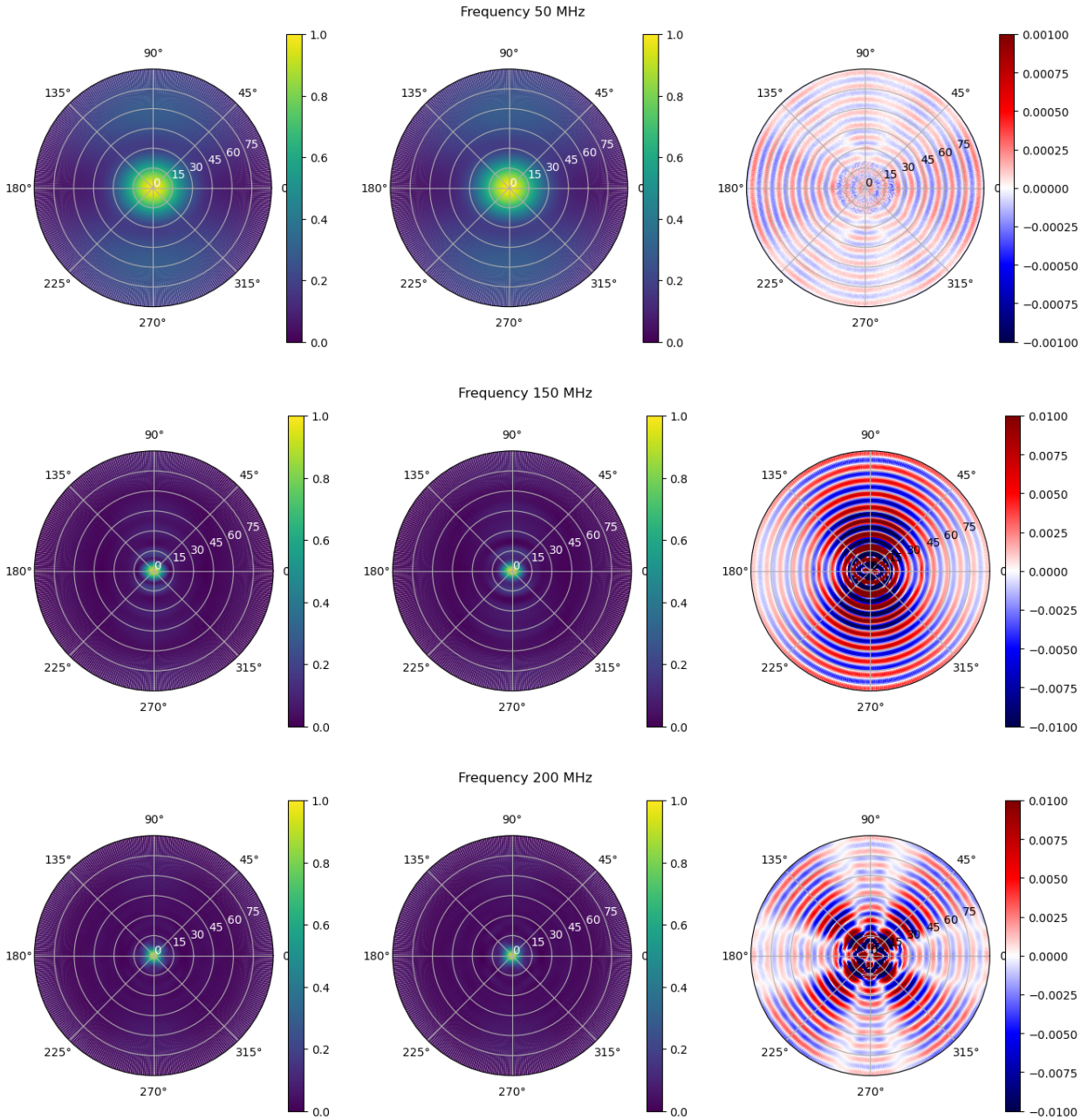

Figure 3 | The true power beam (left), fitted power beam (center) and their difference (right) for three different frequency channels. All fits are done with the same Lmax=35. Given that the beam is smaller at higher frequency, it is not surprising that a fixed Lmaxl leads to a poorer fit relative to the beam at lower frequencies. Nonetheless, an Lmax=35 gives a fairly good fit to the beam.

Case #	Run time [seconds]	(Lmax, Mmax, even M)	Nmodes
1.	1.6 s	(35, 10, True)	186
2.	2.2 s	(35, 15, True)	232
3.	3.3 s	(35, 25, True)	312
4.	3.9 s	(35, 35, True)	342
5.	4.2 s	(35, 15, True)	456
6.	7.6 s	(35, 35, False)	666

Table 1 | This is a table for Case 1- 6 variation of Lmax and Mmax conditions for interpolating the power beam. Each Case # also has a measurement of the Run time in seconds. Column 3 is the different Lmax, Mmax and even M conditions. Column 4 is the Nmodes for each case. Even M means we keep only even m modes and truncate odd m modes from the fit.

Table 1 shows the run-time and number of modes for each case. In each scenario we use all l modes out to Lmax=35, but truncate m to be m < Mmax. In Case 1 - 5, we also truncate all odd m modes. Note that Case 6 is simply our fiducial case from before where we use all modes out to Lmax=35. We profile the run-time of the interpolation in Appendix A and show that it is dominated by the construction of the A matrix, meaning we expect run-time to scale directly with Nmodes, demonstrated in Figure 4.

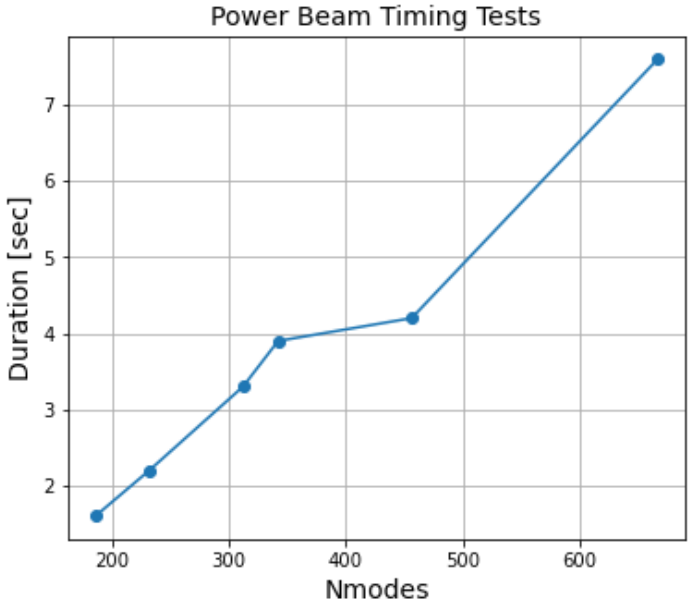

Figure 4 | This shows the run-time of each case in Table 1, showing the run-time scaling against Nmodes of the spherical harmonic matrix when interpolating the power beam. The run-time is dominated by the construction of the spherical harmonic A matrix, thus showing a direct correlation with Nmodes.

Figure 4 plots the measured run-time against Nmodes for each case in Table 1, showing the direct correlation between run-time and Nmodes. Note that the existing spline interpolation method in `pyuvdata.UVBeam` for the same HEALpix setup has a run-time of 2.07 seconds (even after the UVBeam interpolation performance boost introduced in June 2022). This means that our Case 1 interpolation run-time **improves interpolation speed by roughly 25%**; however, this is also the least-accurate fit of the various cases explored.

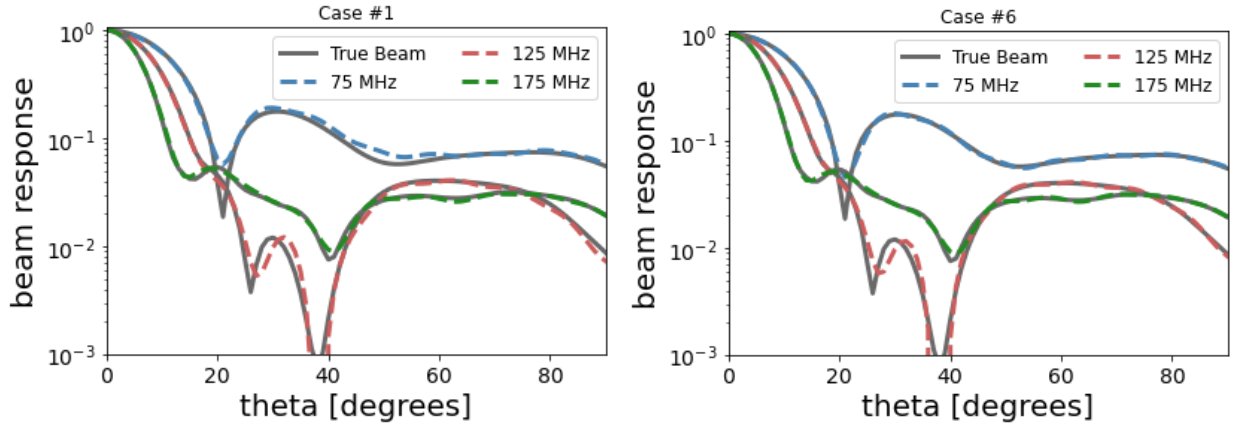

Figure 5 | This is a line plot of the best fit line of a true beam against the fit of at 75 MHz, 125 MHz, and 175 MHz. The true beam values are in gray, the best fits are dashed lines. Case #1 shows a less accurate but faster model (left). Case #6 is a more accurate but slower model (right). While the main-lobe is fit well in both cases, case 6 does a better job at fitting the far sidelobes.

To show more clearly the accuracy of the fits, Figure 5 compares the true power beam (gray lines) against the fitted power beam (dashed lines) for case 1 and case 6 for a handful of frequency channels. We see that case 1 provides a good fit to the main lobe, and provides a decent but not exact fit to the sidelobes. Case 6 improves upon this in most regions, except for the first sidelobe at 125 MHz which is still not perfectly captured (and likely requires a higher Lmax).

E-field beams

Next we repeated the same process for the E-Field beam data. This beam is made of complex numbers and is no longer non-negative. Furthermore, the Naxes dimension now holds two elements, which can be thought of as the two terms in Eqn 1 (i.e. a row of the 2x2 direction-dependent Jones matrix). We use the same spherical harmonic setup as before, but to accommodate the complex beam we fit the real and imaginary beam components separately.

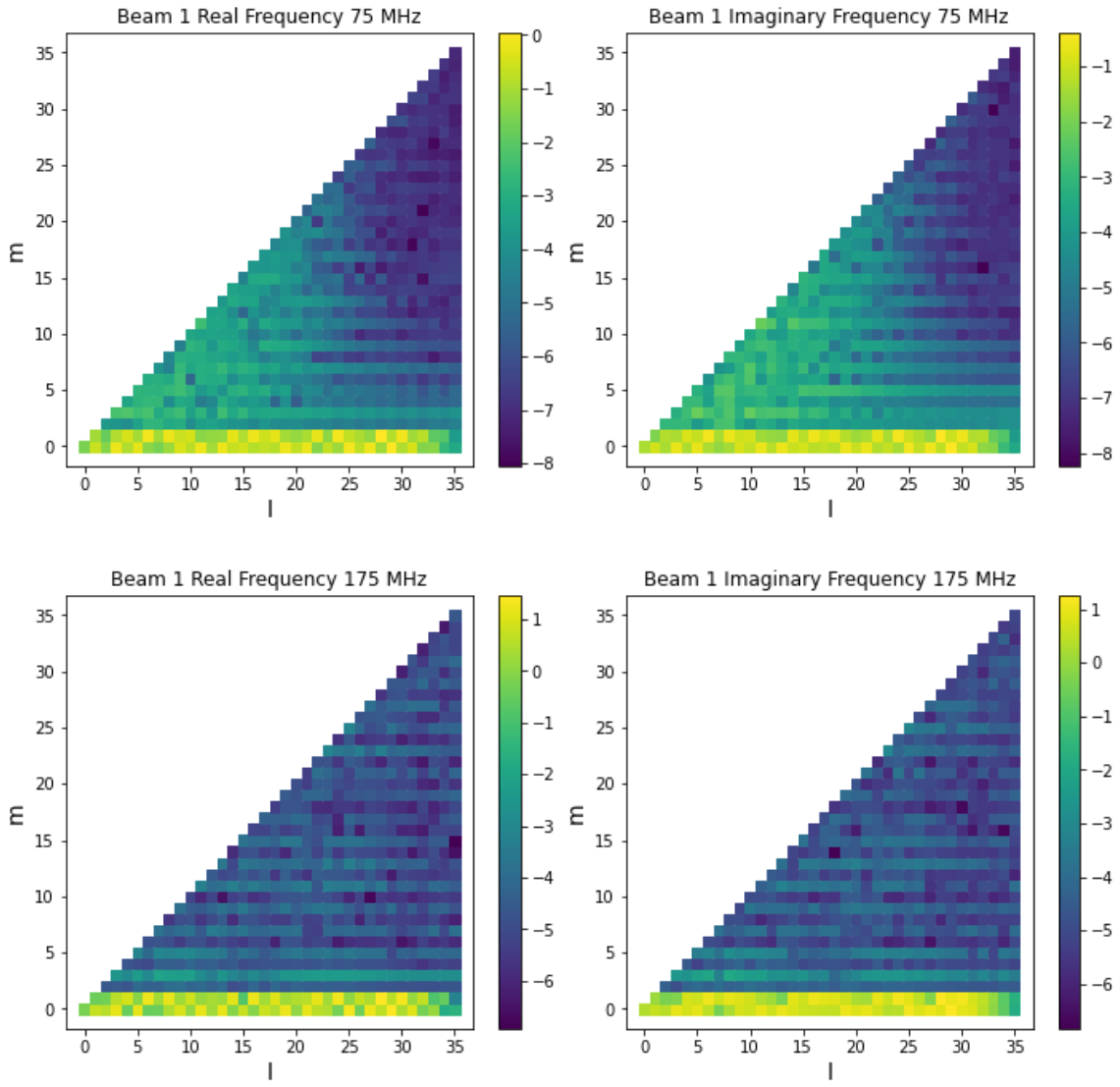

Figure 6 | This shows the amplitude of the harmonic coefficients of the real and imaginary component of the E-field beam for one element of the Naxes dimension. Relative to the power beam, the power of the e-field beam coefficients is more tightly confined to low m modes.

One crucial change was made to the spherical harmonics to allow us to fit them to the e-field beams: for all modes with m = 0, we multiplied the spherical harmonics by $exp[i\phi]$. This was done because the e-field beams do not have any azimuthally-constant structure, which needs to be reflected in the basis functions as well. This means when we take the forward transform (Eqn 5) we now multiply all harmonic coefficients (m >= 0) by 2 before taking the forward transform.

Figure 6 shows the amplitude of the harmonic coefficients of the real and imaginary components of the E-Field beam for one of the elements in the Naxes dimension (which we call Beam 1 and Beam 2). We can see that relative to the power beam, the e-field beam is better described by the low m modes of the spherical harmonics.

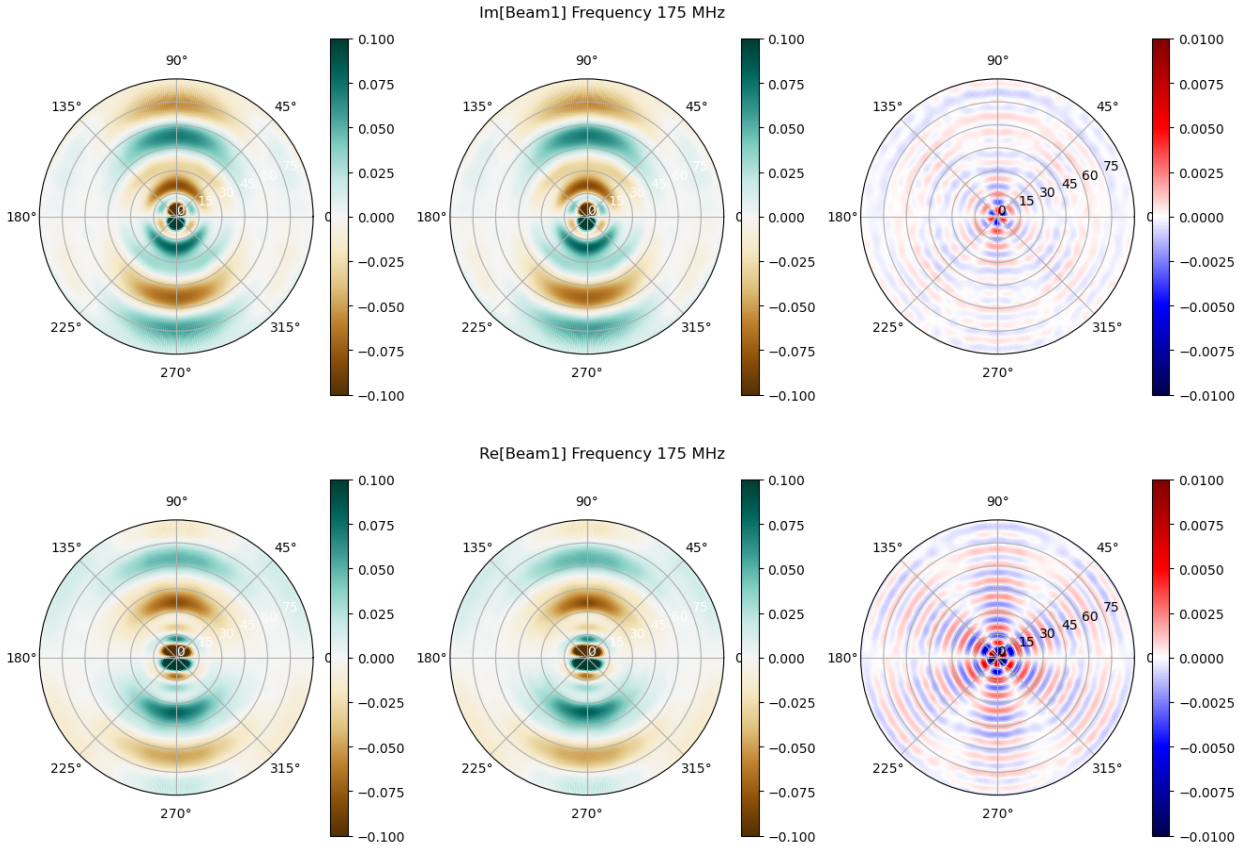

Figure 7 | This is a comparison of the E-field beam 1 at 175MHz, showing the true beam (left), the fitted beam (center) and their residual (right). For the same Lmax, we can see that the E-field beam fits have significantly lower residual than the power beam.

Figure 7 shows the best-fit beams for the real and imaginary components of beam 1. We see that the fit is visually very good, fitting the main-beam and far-sidelobes quite well, and inspecting the residuals we see that the difference is much smaller than when we fit the power beams directly. As we will see below, this is because the intrinsic e-field beams have angular structures smoothly oscillating from negative to positive, and when we take the ABS of this to form the power beam we insert a sharp discontinuity that is not well fit by our Lmax=35 spherical harmonics. To further drive this point home, we compare the true power beam against the reconstructed power beams of the best-fit E-field beams in Figure 8, showing residuals that are significantly smaller than Figure 3. In all, this suggests that beam fitting and interpolation with spherical harmonics should be done on the e-fields, which can then used to form the power beam if needed.

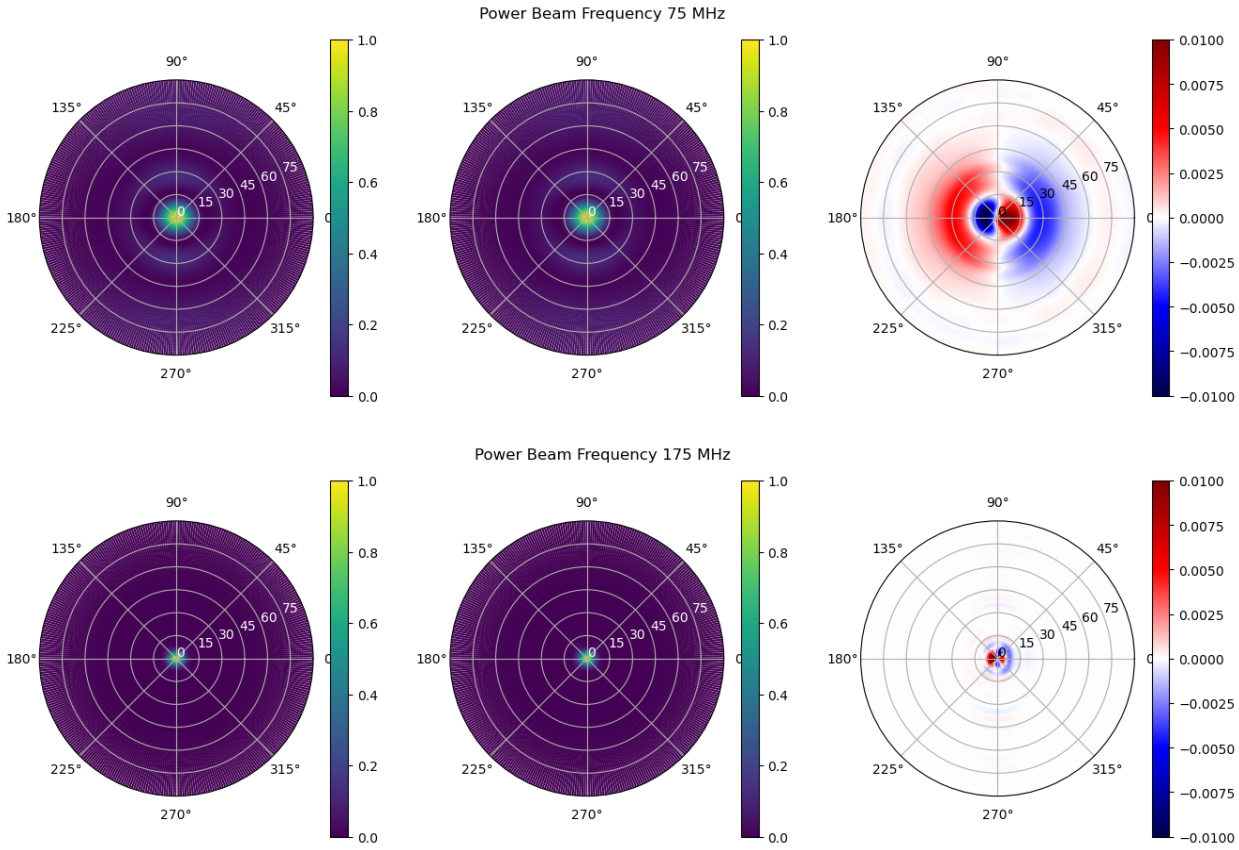

Figure 8 | This shows the true power beam (left) the reconstructed power beam from the best-fit e-field beams (center) and their residual (right). We can see that the residuals are much smaller when we fit the e-fields and then form the power beam, as opposed to when we fit the power beams directly (Figure 3).

Similar to before, we compare a couple of scenarios where we truncate our spherical harmonic Nmodes to compare the run-time of beam interpolation and the accuracy of the resultant fits. The interpolation setup is the same from before. Table 2 records the run-time of each case, where again we order the cases from fewest Nmodes (Case 1, fastest to run but least accurate fit) to most Nmodes (Case 6, slowest to run but most accurate fit). Note that the UVBeam interpolation run-time for the efield beam was 7.44 seconds, meaning our Case 1 interpolation run-time is a factor of 7 in speed-up. Even for Case 3 (which is a more accurate fit) we get >2x speedup.

Case #	Run time [seconds]	Lmax, Mmax	Nmodes
1.	1.1 s	(35, 2)	105
2.	1.6 s	(35, 4)	170
3.	3.0 s	(35, 8)	288
4.	5.6 s	(35, 16)	476
5.	8.8 s	(35, 32)	660
6.	9.0 s	(35, 35)	666

Table 2 | The interpolation run time for Case 1- 6 interpolation of the e-field beam. Here we only truncate Mmax and keep Lmax = 35, and we do not truncate odd Nmodes.

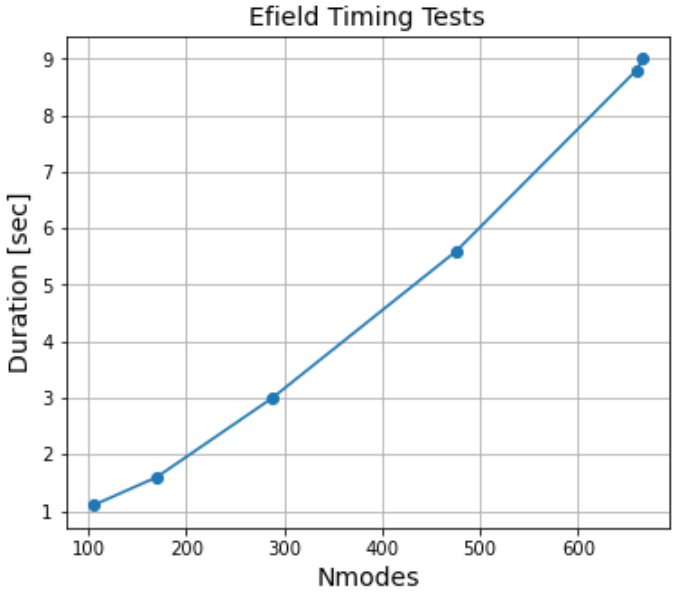

Figure 10 | This shows the spherical harmonic e-field beam interpolation run-time for each case against the number of spherical harmonic Nmodes.

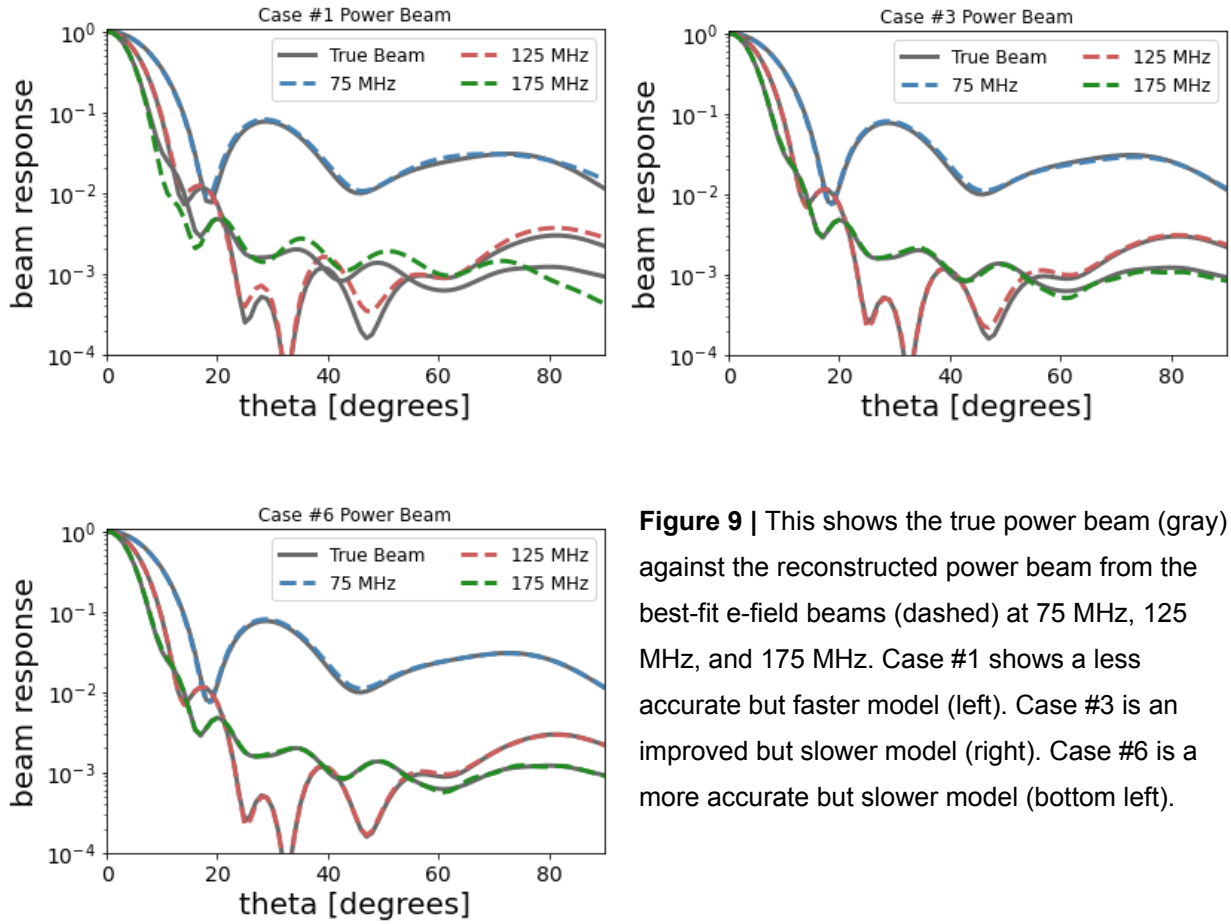

Figure 9 | This shows the true power beam (gray) against the reconstructed power beam from the best-fit e-field beams (dashed) at 75 MHz, 125 MHz, and 175 MHz. Case #1 shows a less accurate but faster model (left). Case #3 is an improved but slower model (right). Case #6 is a more accurate but slower model (bottom left).

To more closely compare the accuracy of the fits in each case, Figure 9 shows the power beam reconstruction from the best-fit e-field beams for cases 1, 3 and 6. This more clearly shows the improved accuracy of the fit relative to fitting the power beams directly.

Conclusion

We demonstrate fitting spherical harmonic functions to electromagnetic simulations of the HERA primary beam response. We explore run-time and accuracy of the fits on both the power beams and complex e-field beams, while varying the number of spherical harmonic modes in the fit. Relative to existing methods of representing the HERA beam that have made simplifying assumptions about the beam (e.g. Choudhuri et al. 2020, Boyer et al. 2021), our method operates on the full angular structure of the beam, and fits each frequency mode independently. As a consequence, our fits are considerably more accurate, with residual errors on the e-field beam fits of roughly 1% or less (Figure 8). When comparing decomposition of the power beams and e-field beams, we see that it is best to decompose the e-field beams first, interpolate if needed, and then reconstruct the power beams. This has multiple advantages driven largely by the fact that the e-field beam is more compactly represented in harmonic space than the power beam (due to the sharp discontinuities created when taking the ABS of the e-field beam). This means we need less modes to accurately represent the beam, which then translates to faster and more memory efficient interpolation because we don't need to store as many harmonic modes in the A matrix. In conclusion, we show that analysis pipelines that require the beam response can be made faster and more memory efficient by using a spherical harmonic representation of the beam. While this has been shown before for other telescopes (e.g. Sokolowski et al. 2017), we show the effectiveness specifically for the HERA Vivaldi beam model for the first time.

Below we summarize the important take-aways from this work.

- **Memory-efficient beam storage:** An (N_modes = 500, N_pix = 65160) complex128 spherical harmonic matrix uses 0.52 GB of memory, whereas an (N_freq = 128, N_pix = 65160) complex128 e-field beam model uses 0.26 GB of memory. However, if we want a suite of perturbed beam models we would naively need to store a separate beam for each model, bringing the cost storage cost to 0.26 x N_beams GB. For the harmonic decomposition, we only need a single spherical harmonic matrix, and we instead store a separate coefficient vector for each beam, which is of negligible size, keeping the storage requirement to ~0.5 GB.
- **Fast and accurate beam interpolation:** For the power beam, our Case 1 scheme sees a slight speed-up (factor of 1.25) over existing methods, but is likely of poorer accuracy. For the e-field beam, however, we get over a factor of 7 improvement in speed for our

Case 1 scheme, and over a factor of 2 improvement in speed for our more accurate Case 3. This is compounded in the regime of interpolating multiple beam models (e.g. when simulating antenna-dependent beams for large N_antenna visibility simulations), where, with current spline interpolation schemes, each beam must be interpolated separately (each beam requires its own UVBeam object). This incurs a linear N_antenna scaling of the beam interpolation, which will be prohibitive for large N_antenna simulations. However, our spherical harmonic interpolation has a nearly flat scaling with N_antenna because as demonstrated in Appendix A, the bottleneck is the construction of the Ylm matrix, which can be used for all antennas. This brings our e-field Case 3 interpolation speed up relative to existing methods to a factor of 2 x N_antenna.

- **Understanding the structure in the beam:** The harmonic coefficients tell us about the intrinsic structure of the beam and its symmetries. Figure 2 tells us that the beam has a slight 180 degree rotational asymmetry, which was not observed in previous dipole beams but is expected in the Vivaldi feed due to the non-symmetrical elements in the Vivaldi support structure.

Next we summarize some areas for future work.

- Run-time benchmarking for large N_frequencies. In this work we benchmark on N_frequency = 201, which is what is provided in the beam. But we are curious to see if our speed-ups over UVBeam are made better or worse for much large N_frequencies.
- Exploring larger Lmax could lead to better fits while marginal run-time cost. Future work could explore increasing Lmax and evaluating tradeoffs.
- Exploring beam decomposition of perturbed beams and mutual coupling power patterns. How much does this scatter power to different harmonic modes in the beam? Is there a good low-dimensional spherical harmonic representation?

Appendix A: Interpolation Profiling

Power Beam Interpolation for Case #1: Profiled on NRAO computers

```
Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     1                                           def interp(xhat, l, m, theta, phi, lmax=25, mmax=25, even_m=False):
     2         1        125.0    125.0      0.0      lms = list(zip(l, m))
     3         1        794.0    794.0      0.1      larr, marr = generate_lm(lmax=lmax, mmax=mmax, even_m=even_m)
     4         1         50.0     50.0      0.0      coeff = np.ones_like(larr)
     5         1         16.0     16.0      0.0      coeff[marr > 0] = 2.0
     6         1         48.0     48.0      0.0      lmarr = list(zip(larr, marr))
     7         1     795281.0 795281.0     87.0      Y = Ylm(larr, marr, theta.reshape(-1, 1), phi.reshape(-1, 1))
     8         1       1030.0   1030.0      0.1      cut = [i for i in range(len(lms)) if lms[i]in lmarr]
     9         1        401.0    401.0      0.0      xh = xhat[cut, :] * coeff[:, None]
    10         1     116638.0 116638.0     12.8      beam_bf = Y @ xh
    11         1          3.0      3.0      0.0      return beam_bf, larr, marr
```

We see that 87% of the run-time for the interpolation function is in calling the Ylm() function for the specified number of l,m coefficients and theta, phi sky angles, meaning that the construction of the Ylm is the main bottleneck.

E-Field Beam Interpolation for Case #1

```
Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
     1                                           def interp(xhat1_real, xhat1_imag, xhat2_real, xhat2_imag, l, m, theta, phi, lmax=35, mmax=35):
     2         1        166.0    166.0      0.0      lms = list(zip(l, m))
     3         1       1343.0   1343.0      0.1      larr, marr = generate_lm(lmax=lmax, mmax=mmax)
     4         1        102.0    102.0      0.0      coeff = np.ones_like(larr)[:, None] * 2
     5         1         66.0     66.0      0.0      lmarr = list(zip(larr, marr))
     6         1    1213540.0 1213540.0     65.1      Y = Ylm(larr, marr, theta.reshape(-1, 1), phi.reshape(-1, 1))
     7         1      24527.0  24527.0      1.3      Y[:, marr==0] *= np.exp(1j * phi.reshape(-1, 1))
     8         1       2492.0   2492.0      0.1      cut = [i for i in range(len(lms)) if lms[i] in lmarr]
     9         1        295.0    295.0      0.0      xh1_real = xhat1_real[cut, :] * coeff
    10         1        209.0    209.0      0.0      xh1_imag = xhat1_imag[cut, :] * coeff
    11         1        206.0    206.0      0.0      xh2_real = xhat2_real[cut, :] * coeff
    12         1        206.0    206.0      0.0      xh2_imag = xhat2_imag[cut, :] * coeff
    13         1     144149.0 144149.0      7.7      beam1_real_bf = (Y @ (xh1_real))
    14         1     109340.0 109340.0      5.9      beam1_imag_bf = (Y @ (xh1_imag))
    15         1     110183.0 110183.0      5.9      beam2_real_bf = (Y @ (xh2_real))
```

```
  16      1    109918.0 109918.0     5.9      beam2_imag_bf = (Y @ (xh2_imag))

  17      1    147001.0 147001.0     7.9      power_bf = beam1_real_bf**2 + beam1_imag_bf**2 + beam2_real_bf**2 +
beam2_imag_bf**2

  18      1         3.0      3.0     0.0      return beam1_real_bf, beam1_imag_bf, beam2_real_bf, beam2_imag_bf, power_bf, larr,
marr
```

For the E-field beam interpolation, calling the Ylm() function is still the bottleneck (65% of run-time), but is not as severe due to the fact that we need to forward transform 4 separate beams now.

```
import scipy
import numpy as np
from scipy import special

def Ylm(l, m, theta, phi):
    x = np.cos(theta)

    # compute the associated legendre function
    P = special.lpmv(m, l, x)

    # compute the complex exponential
    E = np.exp(1j * m * phi)

    # compute the normalization prefactor
    N = np.sqrt(((2 * l + 1) * special.factorial(l - m))/(4 * np.pi * special.factorial(l + m)))

    # multiply them all together

    Y=(P*E*N)

    # return Y
    return (Y)
```

Figure 10 | The function we wrote to evaluate the spherical harmonics.

References

Boyer et al. 2021, “Improvements on an analytical representation of the polarized HERA beam”, HERA Memorandum #101

Choudhuri et al. 2020, “1D Basis Expansion Models for the HERA Primary Beam”, HERA Memorandum #81

Fagnoni et al. 2021, “Design of New Wideband Vivaldi Feed for the HERA Radio Telescope Phase II” ITAP 69, 8143F

Kriele et al. 2022, “Imaging the Southern Sky at 159 MHz using spherical harmonics with the engineering development array 2” PASA 39, 17K

Sokolowski et al. 2017, “Calibration and Stokes Imaging with Full Embedded Element Primary Beam Models for the Murchison Widefield Array” PASA 32, 62S