

MIT Open Access Articles

A fast method for power spectrum and foreground analysis for 21 cm cosmology

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Dillon, Joshua S., Adrian Liu, and Max Tegmark. "A Fast Method for Power Spectrum and Foreground Analysis for 21 Cm Cosmology." *Physical Review D* 87.4 (2013). © 2013 American Physical Society

As Published: <http://dx.doi.org/10.1103/PhysRevD.87.043005>

Publisher: American Physical Society

Persistent URL: <http://hdl.handle.net/1721.1/78266>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



A fast method for power spectrum and foreground analysis for 21 cm cosmologyJoshua S. Dillon,^{1,*} Adrian Liu,^{1,2} and Max Tegmark¹¹*Department of Physics and MIT Kavli Institute, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*²*Department of Astronomy and Berkeley Center for Cosmological Physics, Berkeley, California 94702, USA*

(Received 27 November 2012; published 12 February 2013)

We develop and demonstrate an acceleration of the Liu and Tegmark quadratic estimator formalism for inverse variance foreground subtraction and power spectrum estimation in 21 cm tomography from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$, where N is the number of voxels of data. This technique makes feasible the megavoxel scale analysis necessary for current and upcoming radio interferometers by making only moderately restrictive assumptions about foreground models and survey geometry. We exploit iterative and Monte Carlo techniques and the symmetries of the foreground covariance matrices to quickly estimate the 21 cm brightness temperature power spectrum, $P(k_{\parallel}, k_{\perp})$, the Fisher information matrix, the error bars, the window functions, and the bias. We also extend the Liu and Tegmark foreground model to include bright point sources with known positions in a way that scales as $\mathcal{O}[(N \log N) \times (N \text{ point sources})] \leq \mathcal{O}(N^{5/3})$. As a first application of our method, we forecast error bars and window functions for the upcoming 128-tile deployment of the Murchinson Widefield Array, showing that 1000 hours of observation should prove sufficiently sensitive to detect the power spectrum signal from the Epoch of Reionization.

DOI: [10.1103/PhysRevD.87.043005](https://doi.org/10.1103/PhysRevD.87.043005)

PACS numbers: 95.75.-z, 95.75.Pq, 98.80.-k, 98.80.Es

I. INTRODUCTION

Neutral hydrogen tomography with the 21 cm line promises to shed light on a vast and unexplored epoch of the early Universe. As a cosmological probe, it offers the opportunity to directly learn about the evolution of structure in our Universe during the cosmological dark ages and the subsequent Epoch of Reionization (EoR) [1–4]. More importantly, the huge volume of space and wide range of cosmological scales probed makes 21 cm tomography uniquely suited for precise statistical determination of the parameters that govern modern cosmological and astrophysical models for how our Universe transitioned from hot and smooth to cool and clumpy [5–21]. It has the potential to surpass even the cosmic microwave background (CMB) in its sensitivity as a cosmological probe [20].

The central idea behind 21 cm tomography is that images produced by low-frequency radio interferometers at different frequencies can create a series of images at different redshifts, forming a three-dimensional map of the 21 cm brightness temperature. Yet we expect that our images will be dominated by synchrotron emission from our galaxies and others. In fact, we expect those foreground signals to dominate over the elusive cosmological signal by about four orders of magnitude [22,23].

One major challenge for 21 cm cosmology is the extraction of the brightness temperature power spectrum, a key prediction of theoretical models of the dark ages and the EoR, out from underneath a mountain of foregrounds and instrumental noise. Liu and Tegmark (LT) presented a method for power spectrum estimation that has many advantages over previous approaches (on which we will

elaborate in Sec. II) [24]. It has, however, one unfortunate drawback: it is very slow. The LT method relies on multiplying and inverting very large matrices, operations that scale as $\mathcal{O}(N^3)$, where N is the number of voxels of data to analyze.

The goal of the present paper is to develop and demonstrate a way of achieving the results of the LT method that scales only as $\mathcal{O}(N \log N)$. Along the way, we will also show how LT can be extended to take advantage of additional information about the brightest point sources in the map while maintaining a reasonable algorithmic scaling with N . Current generation interferometers, including the Low Frequency Array (LOFAR, [25]), the Giant Metrewave Radio Telescope (GMRT, [26]), the Murchinson Widefield Array (MWA, [27]), and the Precision Array for Probing the Epoch of Reionization (PAPER, [28]) are already producing massive data sets at or near the megavoxel scale (e.g., [29]). These data sets are simply too large to be tackled by the LT method. We expect next-generation observational efforts, like the Hydrogen Epoch of Reionization Array [30], a massive Omniscope [31], or the Square Kilometer Array [32], to produce even larger volumes of data. Moreover, as computer processing speed continues to grow exponentially, the ability to observe with increasingly fine frequency resolution will enable the investigation of the higher Fourier modes of the power spectrum at the cost of yet larger data sets. The need for an acceleration of the LT method is pressing and becoming more urgent.

Our paper has a similar objective to Ref. [33], which also seeks to speed up algorithms for power spectrum estimation with iterative and Monte Carlo techniques. The major differences between the paper arise from our specialization to the problem of 21 cm cosmology and the added

*jsdillon@mit.edu

complications presented by foregrounds, especially with regard to the basis in which various covariance matrices are easiest to manipulate. Our paper also shares similarities to Ref. [34]. Like Refs. [33,34] does not extend its analysis to include foregrounds. It differs also from this paper in spirit because it seeks to go from interferometric visibilities to a power spectrum within a Bayesian framework rather than from a map to a power spectrum and because it considers one frequency channel at a time. In this paper, we take advantage of many frequency channels simultaneously in order to address the problem of foregrounds.

This paper is organized as follows. We begin with Sec. II wherein we review the motivation for and details of the LT method. In Sec. III we present the novel aspects of our technique for measuring the 21 cm brightness temperature power spectrum. We discuss the extension of the method to bright point sources and the assumptions we must make to accelerate our analysis. In Sec. IV we demonstrate end-to-end tests of the algorithm and show some of its first predictions for the ability of the upcoming 128-tile deployment of the MWA to detect the statistical signal of the Epoch of Reionization.

II. THE BRUTE FORCE METHOD

The solution to the problem of power spectrum estimation in the presence of foregrounds put forward by LT offers a number of improvements over previous proposals that rely primarily on line-of-sight foreground information [35–43]. The problem of 21 cm power spectrum estimation shares essential qualities with both CMB and galaxy survey power spectrum estimation efforts. Like with galaxy surveys, we are interested in measuring a three-dimensional power spectrum. On the other hand, our noise and foreground contaminants bear more similarity to the problems faced by CMB studies—though the foregrounds we face are orders of magnitude larger.

The LT method therefore builds on the literature of both CMB and galaxy surveys, providing a unified framework for the treatment of geometric and foreground effects by employing the quadratic estimator formalism for inverse variance foreground subtraction and power spectrum estimation. In Sec. IIC we will review precisely how it is implemented.

The LT formalism has a number of important advantages over its predecessors. By treating foregrounds as a form of correlated noise, both foregrounds and noise can be down-weighted in a way that is unbiased and lossless in the sense that it maintains all the cosmological information in the data. Furthermore, the method allows for the simultaneous estimation of both the errors on power spectrum estimates and the window functions or “horizontal” error bars.

Unfortunately, the LT method suffers from computational difficulties. Because it involves inverting and multiplying very large matrices, it cannot be accomplished faster than in $\mathcal{O}(N^3)$ steps, where N is the number of voxels

in the data to be analyzed. This makes analyzing large data sets with this method infeasible. The primary goal of this paper is to demonstrate an adaptation of the method that can be run much faster. But first, we need to review the essential elements of the method to put our adaptations and improvements into proper context. In Secs. IIA and IIB, we describe our conventions and notation and explain the relationship between the measured quantities and those we seek to estimate. In Sec. IIC, we review the LT statistical estimators and how the Fisher information matrix is used to calculate statistical errors on power spectrum measurements. Then in IID we explain the LT model of noise and foregrounds in order to motivate and justify our refinements that will greatly speed up the algorithm in Sec. III.

A. Data organization and conventions

We begin with a grid of data that represents the brightness temperatures at different positions on the sky as a function of frequency from which we wish to estimate the 21 cm brightness temperature power spectrum. We summarize that information using a data vector \mathbf{x} , which can be thought of as a one-dimensional object of length $n_x n_y n_z \equiv N$,¹ the number of voxels in the data cube.

Although the LT technique works for arbitrary survey geometries, we restrict ourselves to the simpler case of a data “cube” that corresponds to a relatively small rectangular section of our universe of size $\ell_x \times \ell_y \times \ell_z$ in comoving coordinates.² We pick our box to be a subset of the total 21 cm brightness temperature three-dimensional map that a large interferometric observatory would produce. Unlike the LT method, our fast method requires that the range of positions on the sky must be small enough for the flat sky approximation to hold (Fig. 1). Similarly, our range of frequencies (and thus redshifts) in the data cube must correspond to an epoch short enough so that $P(k, z)$ might be approximated as constant in time. Following simulations by Refs. [20,44] argued that we can conservatively extend the redshift ranges of our data cubes to about $\Delta z \lesssim 0.5$. At typical EoR redshifts, such a small range in Δz allows a very nearly linear mapping between the frequencies measured by an interferometer to a regularly spaced range of comoving distances, $d_C(z)$, although in general $d_C(z)$ is not a linear function of z or ν . This also justifies the approximation that our data cube corresponds to an evenly partitioned volume of our Universe.

If the measured brightness temperatures, x_i , were only the result of redshifted 21 cm radiation, then each

¹While it is helpful to think of \mathbf{x} as a vector in the matrix operations below, it is important to remember that the index i in x_i , which refers to the different components of \mathbf{x} , actually runs over different values of the spatial coordinates x , y , and z .

²This restriction and its attendant approximations lie at the heart of our strategy for speeding up these calculations, as we explain in Sec. III.

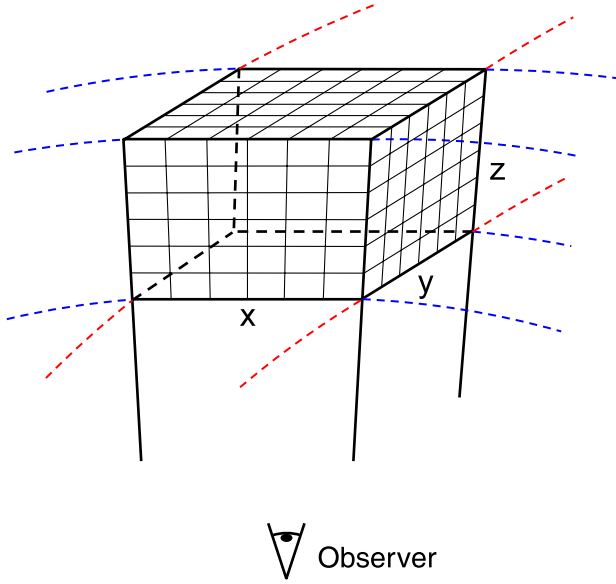


FIG. 1 (color online). This exaggerated schematic illustrates the flat sky approximation. It shows great circles (colored and dashed) approximated linearly in the region considered, with lines tracing back to the observer treated as if they were parallel. Our data cube contains the measured brightness temperatures for every small voxel.

measurement would represent the average value in some small box of volume $\Delta x \Delta y \Delta z$, centered on \mathbf{r}_i of a continuous brightness temperature field $x(\mathbf{r})$ [45],

$$x_i \equiv \int \psi_i(\mathbf{r}) x(\mathbf{r}) d^3 r, \quad (1)$$

where our discretization function ψ_i is defined as $\psi_i(\mathbf{r}) \equiv \psi_0(\mathbf{r} - \mathbf{r}_i)$, where

$$\psi_0(\mathbf{r}) \equiv \frac{\Pi(\frac{x}{\Delta x}) \Pi(\frac{y}{\Delta y}) \Pi(\frac{z}{\Delta z})}{\Delta x \Delta y \Delta z} \quad (2)$$

and where $\Pi(x)$ is the normalized, symmetric boxcar function ($\Pi(x) = 1$ if $|x| < \frac{1}{2}$ and 0 otherwise). This choice of pixelization encapsulates the idea that each measured brightness temperature is the average over a continuous temperature field inside each voxel. In this paper, we improve on the LT method by including the effect of finite pixelation. This will manifest itself as an extra $\Phi(\mathbf{k})$ term that we will define in Eq. (6) and that will reappear throughout this paper.

B. The discretized 21 cm power spectrum

Ultimately, the goal of this paper is to estimate the 21 cm power spectrum $P(\mathbf{k})$, defined via

$$\langle \tilde{x}^*(\mathbf{k}) \tilde{x}(\mathbf{k}') \rangle \equiv (2\pi)^3 \delta(\mathbf{k} - \mathbf{k}') P(\mathbf{k}), \quad (3)$$

where $\tilde{x}(\mathbf{k})$ is the Fourier-transformed brightness temperature field and where angle brackets denote the ensemble

average of all possible universes obeying the same statistics.

Our choice of pixelization determines the relationship between the continuous power spectrum, $P(\mathbf{k})$, and the 21 cm signal covariance matrix, which we call \mathbf{S} for Signal. It is fairly straightforward to show, given Eq. (1) and the definition of the power spectrum [45], that

$$S_{ij} \equiv \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle = \int \tilde{\psi}_i(\mathbf{k}) \tilde{\psi}_j^*(\mathbf{k}) P(\mathbf{k}) \frac{d^3 k}{(2\pi)^3}, \quad (4)$$

where $\tilde{\psi}_i(\mathbf{k})$ is the Fourier transform of $\psi_i(\mathbf{r})$,

$$\tilde{\psi}_i(\mathbf{k}) \equiv \int e^{-i\mathbf{k} \cdot \mathbf{r}} \psi_i(\mathbf{r}) d^3 \mathbf{r}. \quad (5)$$

Separating this integral into each of the three Cartesian coordinates and integrating yields

$$\tilde{\psi}_i(\mathbf{k}) = e^{i\mathbf{k} \cdot \mathbf{r}_i} \Phi(\mathbf{k}), \quad (6)$$

$$\text{where } \Phi(\mathbf{k}) \equiv j_0\left(\frac{k_x \Delta x}{2}\right) j_0\left(\frac{k_y \Delta y}{2}\right) j_0\left(\frac{k_z \Delta z}{2}\right),$$

where $j_0(x) = \sin x / x$ is the zeroth spherical Bessel function. Because we can only make a finite number of measurements of the power spectrum, we parametrize and discretize $P(\mathbf{k})$ by approximating it as a piecewise constant function,

$$P(\mathbf{k}) \approx \sum_{\alpha} p^{\alpha} \chi^{\alpha}(\mathbf{k}), \quad (7)$$

where the ‘‘band power’’ p^{α} gives the power in region α of Fourier space,³ specified by the characteristic function $\chi^{\alpha}(\mathbf{k})$ which equals 1 inside the region and vanishes elsewhere.

Combining Eqs. (4) and (7), we can write down S_{ij} ,

$$S_{ij} = \sum_{\alpha} p^{\alpha} Q_{ij}^{\alpha}, \quad (8)$$

$$\text{where } Q_{ij}^{\alpha} \equiv \int \tilde{\psi}_i(\mathbf{k}) \tilde{\psi}_j^*(\mathbf{k}) \chi^{\alpha}(\mathbf{k}) \frac{d^3 k}{(2\pi)^3}.$$

We choose these $\chi^{\alpha}(\mathbf{k})$ to produce band powers that reflect the symmetries of the observation. Our Universe is isotropic in three dimensions, but due to redshift space distortions, foregrounds, and other effects, our measurements will be isotropic only perpendicular to the line of sight [20,46–49]. This suggests cylindrical binning of the power spectrum; in the directions perpendicular to the line of sight, we bin k_x and k_y together radially to get a region in k -space extending from $k_{\perp}^{\alpha} - \Delta k_{\perp} / 2$ to $k_{\perp}^{\alpha} + \Delta k_{\perp} / 2$, where $k_{\perp}^{\alpha} \equiv k_x^2 + k_y^2$. Likewise, in the direction parallel to the line of sight, we integrate over a region of k -space both

³In contrast to lowered Latin indices, which we use to pick out voxels in a real space or Fourier space data cube, we will use raised Greek indices to pick out power spectrum bins, which will generally each run a range in k_{\parallel} and in k_{\perp} .

from $k_{\parallel}^{\alpha} - \Delta k_{\parallel}/2$ to $k_{\parallel}^{\alpha} + \Delta k_{\parallel}/2$ and, because the power spectrum only depends on $k_{\parallel} \equiv |k_z|$, from $-k_{\parallel}^{\alpha} + \Delta k_{\parallel}/2$ to $-k_{\parallel}^{\alpha} - \Delta k_{\parallel}/2$. Therefore, we have

$$Q_{ij}^{\alpha} = \frac{1}{(2\pi)^3} \left[\int_{k_{\parallel}^{\alpha} - \Delta k_{\parallel}/2}^{k_{\parallel}^{\alpha} + \Delta k_{\parallel}/2} - \int_{-k_{\parallel}^{\alpha} + \Delta k_{\parallel}/2}^{-k_{\parallel}^{\alpha} - \Delta k_{\parallel}/2} \right] \times \int_{k_{\perp}^{\alpha} - \Delta k_{\perp}/2}^{k_{\perp}^{\alpha} + \Delta k_{\perp}/2} |\Phi(\mathbf{k})|^2 e^{i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)} k_{\perp} d\theta dk_{\perp} dk_{\parallel}. \quad (9)$$

Without the factor of $|\Phi(\mathbf{k})|^2$, the LT method was able to evaluate this integral analytically. With it, the integral must be evaluated numerically if it is to be evaluated at all. This is of no consequence; we will return to this formula in Sec. III B to show how the matrix \mathbf{Q}^{α} naturally lends itself to approximate multiplication by vectors using fast Fourier techniques.

C. 21 cm power spectrum statistics

In order to interpret the data from any experiment, we need to be able to estimate both the 21 cm brightness temperature power spectrum and the correlated errors induced by the survey parameters, the instrument, and the foregrounds. The LT method does both at the same time; with it, the calculation of the error bars immediately enables power spectrum estimation.

1. Inverse variance weighted power spectrum estimation

The LT method adapts the inverse variance weighted quadratic estimator formalism [50,51] for calculating 21 cm power spectrum statistics. The first step towards constructing the estimator \hat{p}^{α} for p^{α} is to compute a quadratic quantity, called \hat{q}^{α} whose relationship⁴ to \hat{p}^{α} we will explain shortly,

$$\hat{q}^{\alpha} \equiv \frac{1}{2} (\mathbf{x} - \langle \mathbf{x} \rangle)^{\top} \mathbf{C}^{-1} \mathbf{Q}^{\alpha} \mathbf{C}^{-1} (\mathbf{x} - \langle \mathbf{x} \rangle). \quad (10)$$

Here \mathbf{C} is the covariance matrix of \mathbf{x} , so

$$\mathbf{C} \equiv \langle \mathbf{x} \mathbf{x}^{\top} \rangle - \langle \mathbf{x} \rangle \langle \mathbf{x} \rangle^{\top}. \quad (11)$$

For any given value of α , the right-hand side of Eq. (10) yields a scalar. Were both our signal and foregrounds Gaussian, this estimator would be optimal in the sense that it preserves all the cosmological information contained in the data. Of course, with a non-Gaussian signal, the power spectrum cannot contain all of the information, though it still can be very useful [50].

Our interest in the quadratic estimators \hat{q}^{α} lies in their simple relationship to the underlying band powers. In Ref. [50], it is shown that

⁴Unlike the notation in LT, we do not include the bias term in \hat{q}^{α} but will later include it in our power spectrum estimator. The result is the same.

$$\langle \hat{\mathbf{q}} \rangle = \mathbf{F} \mathbf{p} + \mathbf{b}, \quad (12)$$

where each b^{α} is the bias in the estimator and \mathbf{F} is the Fisher information matrix, which is related to the probability of having measured our data given a particular set of band powers, $f(\mathbf{x}|p^{\alpha})$. The matrix is defined [52] as

$$F^{\alpha\beta} \equiv - \left\langle \frac{\partial^2 \ln f(\mathbf{x}|p^{\alpha})}{\partial p^{\alpha} \partial p^{\beta}} \right\rangle. \quad (13)$$

The LT method employs the estimators by calculating both \mathbf{F} and \mathbf{b} using relationships derived in [50],

$$F^{\alpha\beta} = \frac{1}{2} \text{tr}[\mathbf{C}^{-1} \mathbf{Q}^{\alpha} \mathbf{C}^{-1} \mathbf{Q}^{\beta}] \quad (14)$$

and

$$b^{\alpha} = \frac{1}{2} \text{tr}[(\mathbf{C} - \mathbf{S}) \mathbf{C}^{-1} \mathbf{Q}^{\alpha} \mathbf{C}^{-1}]. \quad (15)$$

We want our \hat{p}^{α} to be unbiased estimators of the true underlying band powers, which means that we will have to take care to remove the biases for each band power, b^{α} . We construct our estimators⁵ as linear combinations of the quadratic estimators \hat{q}^{α} that have been corrected for bias,

$$\hat{\mathbf{p}} = \mathbf{M}(\hat{\mathbf{q}} - \mathbf{b}), \quad (16)$$

where \mathbf{M} is a matrix which correctly normalizes the power spectrum estimates; the form of \mathbf{M} represents a choice in the trade-off between small error bars and narrow window functions, as we will explain shortly.

How do we expect this estimator to behave statistically? The only random variable on the right-hand side of Eq. (16) is $\hat{\mathbf{q}}$, so we can combine Eqs. (12) and (16) to see that our choice of $\hat{\mathbf{p}}$ indeed removes the bias term,

$$\langle \hat{\mathbf{p}} \rangle = \mathbf{M} \mathbf{F} \mathbf{p} + \mathbf{M} \mathbf{b} - \mathbf{M} \mathbf{b} = \mathbf{M} \mathbf{F} \mathbf{p} = \mathbf{W} \mathbf{p}. \quad (17)$$

We have defined the matrix of ‘‘window functions’’ $\mathbf{W} \equiv \mathbf{M} \mathbf{F}$ because Eq. (17) tells us that we can expect our band power spectrum estimator, $\hat{\mathbf{p}}$, to be a weighted average of the true, underlying band powers, \mathbf{p} . That definition imposes the condition on \mathbf{W} that

$$\sum_{\beta} W^{\alpha\beta} = 1, \quad (18)$$

which is equivalent to the statement that the weights in a weighted average must add up to one. The condition on \mathbf{W} constrains our choice of \mathbf{M} , though as long as \mathbf{M} is an invertible matrix,⁶ the choice of \mathbf{M} does not change the information content of our power spectrum estimate, only the way we choose to represent our result.

⁵Here we differ slightly from the LT method in the normalization, which does not have the property from Eq. (18). We instead follow Ref. [53].

⁶None of the choices of \mathbf{M} involve anything more computationally intensive than inverting \mathbf{F} . This is fine, since \mathbf{F} is a much smaller matrix than \mathbf{C} .

2. Window functions and error bars

In this paper, we choose a form of $\hat{\mathbf{p}}$ where $\mathbf{M} \propto \mathbf{F}^{-1/2}$. Two other choices for \mathbf{M} are presented in Ref. [53]: one where $\mathbf{M} \propto \mathbf{I}$ and another where $\mathbf{M} \propto \mathbf{F}^{-1}$. The former produces the smallest possible error bars, but at the cost of wide window functions and correlated measurement errors. The latter produces δ -function windows, but large and anticorrelated measurement errors. This choice of $\mathbf{M} \propto \mathbf{F}^{-1/2}$ has proven to be a happy medium between those other two choices for \mathbf{M} . It produces reasonably narrow window functions and reasonably small error bars, which have the added advantage of being completely uncorrelated so that each measurement contains a statistically independent piece of information. Because $\mathbf{W} \equiv \mathbf{M}\mathbf{F}$ and because of the condition on \mathbf{W} in Eq. (18), there is only one such \mathbf{M} ,

$$M^{\alpha\beta} \equiv \frac{(\mathbf{F}^{-1/2})^{\alpha\beta}}{\sum_{\gamma} (\mathbf{F}^{1/2})^{\alpha\gamma}}. \quad (19)$$

With this choice of \mathbf{M} we get window functions of the form

$$W^{\alpha\beta} = \frac{(\mathbf{F}^{1/2})^{\alpha\beta}}{\sum_{\gamma} (\mathbf{F}^{1/2})^{\alpha\gamma}} \quad (20)$$

which we can use to put ‘‘horizontal error bars’’ on our power spectrum estimates.

Using Eq. (16) and the fact derived in Ref. [50] that an equivalent formula for \mathbf{F} is given by

$$\mathbf{F} = \langle \hat{\mathbf{q}}\hat{\mathbf{q}}^{\mathbf{T}} \rangle - \langle \hat{\mathbf{q}} \rangle \langle \hat{\mathbf{q}} \rangle^{\mathbf{T}}, \quad (21)$$

we can see that the covariance of $\hat{\mathbf{p}}$ takes on a simple form,

$$\langle \hat{\mathbf{p}}\hat{\mathbf{p}}^{\mathbf{T}} \rangle - \langle \hat{\mathbf{p}} \rangle \langle \hat{\mathbf{p}} \rangle^{\mathbf{T}} = \mathbf{M}\mathbf{F}\mathbf{M}^{\mathbf{T}}. \quad (22)$$

This allows us to write down the ‘‘vertical error bars’’ on our individual power spectrum estimates,

$$\Delta \hat{p}^{\alpha} = [(\mathbf{M}\mathbf{F}\mathbf{M}^{\mathbf{T}})^{\alpha\alpha}]^{1/2} = \frac{1}{\sum_{\gamma} (\mathbf{F}^{1/2})^{\alpha\gamma}}. \quad (23)$$

As in LT, we can transform our power spectrum estimates and our vertical error bars into temperature units,

$$\hat{T}^{\alpha} \equiv \left[\frac{(k_{\perp}^{\alpha})^2 k_{\parallel}^{\alpha}}{2\pi^2} p^{\alpha} \right]^{1/2} \quad (24)$$

and likewise,

$$\Delta \hat{T}^{\alpha} = \left[\frac{(k_{\perp}^{\alpha})^2 k_{\parallel}^{\alpha}}{2\pi^2 (\sum_{\gamma} (\mathbf{F}^{1/2})^{\alpha\gamma})} \right]^{1/2}. \quad (25)$$

This makes it easier to compare to theoretical predictions, which are often quoted in units of K or mK.

D. Foreground and noise models

The structure of the matrix \mathbf{C} that goes into our inverse variance weighted estimator depends on the way we model our foregrounds, noise, and signal. We assume that those contributions are the sum of five uncorrelated components,

$$\begin{aligned} \mathbf{C} &= \sum_{c \in \text{components}} \langle \mathbf{x}_c \mathbf{x}_c^{\mathbf{T}} \rangle - \langle \mathbf{x}_c \rangle \langle \mathbf{x}_c \rangle^{\mathbf{T}} \\ &\equiv \mathbf{S} + \mathbf{R} + \mathbf{U} + \mathbf{G} + \mathbf{N}. \end{aligned} \quad (26)$$

These are the covariance matrices due to 21 cm Signal, bright point sources Resolved from one another, Unresolved point sources, the Galactic synchrotron, and detector Noise, respectively. This deconstruction of \mathbf{C} is both physically motivated and will ultimately let us approximate $\mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ much more quickly than by just inverting the matrix.

Following LT, we neglect the small cosmological \mathbf{S} because it is only important for taking cosmic variance into account. It is straightforward to include the \mathbf{S} matrix in our method, especially because we expect it to have a very simple form, but this will only be necessary once the experimental field moves from upper limits to detection and characterization of the 21 cm brightness temperature power spectrum.

In this paper, we will develop an accelerated version of the LT method using the models delineated in LT. That speed-up relies on the fact that all of these covariance matrices can be multiplied by vectors $\mathcal{O}(N \log N)$ time. However, our techniques for acceleration will work on a large class of models for \mathbf{C} as long as certain assumptions about translation invariance and spectral structure are respected. In this section, we review the three contaminant matrices from LT, \mathbf{U} , \mathbf{G} , and \mathbf{N} . When we discuss methods to incorporate these matrices into a faster technique in Sec. III D, we will also expand the discussion of foregrounds to include \mathbf{R} , which is a natural extension of \mathbf{U} .

1. Unresolved point sources

For a typical current generation or near future experiment, the pixels perpendicular to the line of sight are so large that every one is virtually guaranteed to have a point source in it bright enough to be an important foreground to our 21 cm signal. These confusion-limited point sources are taken into account using their strong correlations parallel to the line of sight and weaker correlations perpendicular to the line of sight, both of which are easily discerned in Fig. 2.

Following LT we split \mathbf{U} , into the tensor product of two parts, one representing correlations perpendicular to the line of sight and the other parallel to the line of sight,

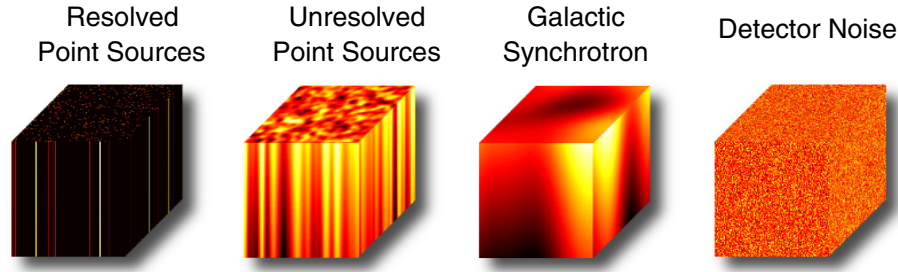


FIG. 2 (color online). These example data cubes (with the line of sight drawn vertically) illustrate the strong or weak correlations between different voxels in the same cube. In Sec. III F we explain how these simulated data cubes are generated quickly. The addition of resolved point sources, which is not included in LT, is discussed in Sec. III D 1. To best exemplify the detailed structure of the models, the color scales are different for each of the cubes.

$$\mathbf{U} \equiv \mathbf{U}_\perp \otimes \mathbf{U}_\parallel \quad (27)$$

Covariance perpendicular to the line of sight is modeled as an unnormalized Gaussian,

$$(U_\perp)_{ij} \equiv \exp\left[-\frac{((\mathbf{r}_\perp)_i - (\mathbf{r}_\perp)_j)^2}{2\sigma_\perp^2}\right] \quad (28)$$

where σ_\perp represents the correlation length perpendicular to the line of sight. Following LT, we take this to be a comoving distance corresponding to 7 arcminutes, representing the weak clustering of point sources.

The covariance along the line of sight assumes a Poisson distributed number of point sources below some flux cut, S_{cut} , which we take to be 0.1 Jy, each with a spectral index drawn from a Gaussian distribution with mean $\bar{\kappa}$ and standard deviation σ_κ . Given a differential source count [54] of

$$\begin{aligned} \frac{dn}{dS} &= (4000 \text{ Jy}^{-1} \text{ sr}^{-1}) \\ &\times \begin{cases} \left(\frac{S}{0.880 \text{ Jy}}\right)^{-2.51} & \text{for } S > 0.880 \text{ Jy} \\ \left(\frac{S}{0.880 \text{ Jy}}\right)^{-1.75} & \text{for } S \leq 0.880 \text{ Jy,} \end{cases} \end{aligned} \quad (29)$$

we get a covariance parallel to the line of sight of

$$\begin{aligned} (U_\parallel)_{ij} &= (1.4 \times 10^{-3} \text{ K})^2 (\eta_i \eta_j)^{-2-\bar{\kappa}} \left(\frac{\Omega_{\text{pix}}}{1 \text{ sr}}\right)^{-1} \\ &\times \exp\left[\frac{\sigma_\kappa^2}{2} (\ln(\eta_i \eta_j))^2\right] I_2(S_{\text{cut}}), \end{aligned} \quad (30)$$

where we have assumed a power law spectrum for the point sources where $\eta_i \equiv \nu_i/\nu_*$, $\nu_* = 150 \text{ MHz}$, and $\bar{\kappa}$ and σ_κ are the average value and standard deviation of the distribution of spectral indices of the point sources. We define $I_2(S_{\text{cut}})$ as

$$I_2(S_{\text{cut}}) \equiv \int_0^{S_{\text{cut}}} S^2 \frac{dn}{dS} dS \quad (31)$$

Following LT, we take $\bar{\kappa} = 0.5$ and $\sigma_\kappa = 0.5$, both of which are consistent with the results of Ref. [29]. In

Sec. III D 2, we will return to Eq. (31) and show how it can be put into an approximate form that can be quickly multiplied by a vector.

2. Galactic synchrotron radiation

Following LT, we model Galactic synchrotron emission in the same way that we model unresolved point sources. Fundamentally, both are spatially correlated synchrotron signals contributing to the brightness temperature of every pixel in our data cube. However the Galactic synchrotron is much more highly correlated spatially, which can be clearly seen in the sample data cube in Fig. 2. This leads to our adoption of a much larger value of σ_\perp ; we take σ_\perp to be a comoving distance corresponding to 30° on the sky. Following LT, we take $\bar{\kappa} = 0.8$ and $\sigma_\kappa = 0.4$.

This is an admittedly crude model for the Galactic synchrotron, in part because it fails to take into account the roughly planar spatial distribution of the Galactic synchrotron. A more sophisticated model for \mathbf{G} that incorporates a more informative map of the Galactic synchrotron can only produce smaller error bars and narrower window functions. However, such a model might involve breaking the assumption of the translational invariance of correlations, which could be problematic for the technique we use in Sec. III to speed up this algorithm. In practice, we expect very little benefit from an improved spatial model of the Galactic synchrotron due to the restriction imposed by the flat sky approximation that our map encompass a relatively small solid angle.

3. Instrumental noise

Here we diverge from LT to adopt a form of the noise power spectrum from Ref. [55] that is more readily adaptable to the pixelization scheme we will introduce,

$$P^N(\mathbf{k}, \lambda) = \frac{\lambda^2 T_{\text{sys}}^2 y d_M^2}{A f_{\text{cover}} \tau} B^{-2}(\mathbf{k}, \lambda). \quad (32)$$

Here T_{sys} is the system temperature (which is sky noise dominated in our case), A is the total effective collecting area of the array, and τ is the total observing time.

$B(\mathbf{k}, \lambda)$ is a function representing the uv coverage, normalized to peak at unity, which changes with wavelength. Lastly, y is the conversion from bandwidth to the comoving length of the box parallel to the line of sight and d_M is the transverse comoving distance,⁷ so $yd_M^2 \Omega_{\text{pix}} \Delta\nu = \Delta x \Delta y \Delta z$ with Ω_{pix} being the angular size of our pixels and $\Delta\nu$ being the frequency channel width. This form of the noise power spectrum assumes that the entire map is observed for the same time τ , which is why the ratio of the angular size of the map to the field of view does not appear.

We use Eq. (4) to discretize the power spectrum and get \mathbf{N} ,

$$N_{ij} = \int e^{i\mathbf{k}\cdot\mathbf{r}_i} e^{-i\mathbf{k}\cdot\mathbf{r}_j} |\Phi(\mathbf{k})|^2 P^N(\mathbf{k}, \lambda) \frac{d^3k}{(2\pi)^3} \quad (33)$$

Instead of evaluating this integral, we will show in Sec. III D 3 that it can be approximated using the discrete Fourier transform.

E. Computational challenges to the brute force method

For a large data cube, the LT method requires the application of large matrices that are memory intensive to store and computationally infeasible to invert. However, we need to be able to multiply by and often invert these large matrices to calculate our quadratic estimators [Eqs. (9) and (10)], the Fisher matrix [Eq. (14)], and the bias [Eq. (15)]. A 10^6 voxel data cube, for example, would take $\mathcal{O}(10^{18})$ computational steps to analyze. This is simply infeasible for next-generation radio interferometers, and we have therefore endeavored to find a faster way to compute 21 cm power spectrum statistics.

III. OUR FAST METHOD

To avoid the computational challenges of the LT method, we seek to exploit symmetries and simpler forms in certain bases of the various matrices out of which we construct our estimate of the 21 cm power spectrum and its attendant errors and window functions. In this section, we describe the mathematical and computational techniques we employ to create a fast and scalable algorithm.

Our fast method combines the following six separate ideas:

- (1) A Monte Carlo technique for computing the Fisher information matrix and the bias (Sec. III A).
- (2) An FFT-based technique for computing band powers using the \mathbf{Q}^α matrices (Sec. III B).

⁷The transverse comoving distance, $d_M(z)$, is the ratio of an object's comoving size to the angle it subtends, as opposed to the angular diameter distance, $d_A(z)$, which is the ratio of its physical size to the angle it subtends. It is sometimes called the "comoving angular diameter distance," and it is even sometimes written as $d_A(z)$. See Ref. [56] for a helpful summary of these often confusingly named quantities.

- (3) An application of the conjugate gradient method that eliminates the need to invert \mathbf{C} (Sec. III C).
- (4) A Toeplitz matrix technique for multiplying vectors quickly by the constituent matrices of \mathbf{C} (Sec. III D).
- (5) A combined FFT and spectral technique for preconditioning \mathbf{C} to improve converge of the conjugate gradient method (Sec. III E).
- (6) A technique using spectral decomposition and Toeplitz matrices for rapid simulation of data cubes for our Monte Carlo (Sec. III F).

In this section, we explain how all six are realized and how they fit into our fast method for power spectrum estimation. Finally, in Sec. III G, we verify the algorithm in an end-to-end test.

A. Monte Carlo calculation of the Fisher information matrix

In order to turn the results of our quadratic estimator into estimates of the power spectrum with proper vertical and horizontal error bars, we need to be able to calculate the Fisher information matrix and the bias term. Instead of using the form of \mathbf{F} in Eq. (14) that the LT method employs, we take advantage of the relationship between \mathbf{F} and $\hat{\mathbf{q}}$ in Eq. (21) that $\mathbf{F} = \langle \hat{\mathbf{q}} \hat{\mathbf{q}}^T \rangle - \langle \hat{\mathbf{q}} \rangle \langle \hat{\mathbf{q}} \rangle^T$. If we can generate a large number of simulated data sets \mathbf{x} drawn from the same covariance \mathbf{C} and then compute $\hat{\mathbf{q}}$ from each one, then we can iteratively approximate \mathbf{F} with a Monte Carlo. In other words, a solution to the problem of quickly calculating $\hat{\mathbf{q}}$ also provides us with a way to estimate \mathbf{F} . What's more, the solution is trivially parallelizable; creating artificial data cubes and analyzing them can be done by many CPUs simultaneously.

In calculating \mathbf{F} , we can get b^α out essentially for free. If we take the average of all our $\hat{\mathbf{q}}$ vectors, we expect that

$$\begin{aligned} \langle \hat{q}^\alpha \rangle &= \left\langle \frac{1}{2} (\mathbf{x} - \langle \mathbf{x} \rangle)^T \mathbf{C}^{-1} \mathbf{Q}^\alpha \mathbf{C}^{-1} (\mathbf{x} - \langle \mathbf{x} \rangle) \right\rangle \\ &= \text{tr}[\langle (\mathbf{x} - \langle \mathbf{x} \rangle) (\mathbf{x} - \langle \mathbf{x} \rangle)^T \mathbf{C}^{-1} \mathbf{Q}^\alpha \mathbf{C}^{-1} \rangle] \\ &= \text{tr}[\mathbf{Q}^\alpha \mathbf{C}^{-1}] = b^\alpha \end{aligned} \quad (34)$$

in the limit where \mathbf{S} is negligibly small. This implies that $\hat{\mathbf{p}}$ can be written in an even simpler way,

$$\hat{p}^\alpha = \frac{1}{\sum_\gamma F^{\alpha\gamma}} (\hat{q}^\alpha - \langle \hat{q}^\alpha \rangle), \quad (35)$$

where, recall, \mathbf{F} is calculated as the sample covariance of our $\hat{\mathbf{q}}$ vectors. We therefore can calculate all the components of our power spectrum estimate and its error bars using a Monte Carlo.

In Sec. III G we will return to assess how well the Monte Carlo technique works and its convergence properties. But first, we need to tackle the three impediments to

computing \hat{q}^α in Eq. (10) quickly: generating a random \mathbf{x} drawn from \mathbf{C} , computing $\mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$, and applying \mathbf{Q}^α .

B. Fast power spectrum estimation without noise or foregrounds

If we make the definition that

$$\mathbf{y} \equiv \mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle) \quad (36)$$

to simplify Eq. (10) to

$$q^\alpha = \mathbf{y}^\top \mathbf{Q}^\alpha \mathbf{y}, \quad (37)$$

we can see that even if we have managed to calculate \mathbf{y} quickly, we still need to multiply it by a $N \times N$ element \mathbf{Q}^α matrix for each band power α . Though each \mathbf{Q}^α respects translation invariance that could make multiplying by vectors faster, there exists an even faster technique that can calculate every entry of $\hat{\mathbf{p}}$ simultaneously using fast Fourier transforms.

To see that this is the case, we substitute Eq. (9) into Eq. (37), reversing the order of summation and integration and factoring the integrand,

$$\begin{aligned} \hat{q}^\alpha &= \frac{1}{2} \left[\int_{k_{\parallel}^\alpha - \Delta k_{\parallel}/2}^{k_{\parallel}^\alpha + \Delta k_{\parallel}/2} - \int_{-k_{\parallel}^\alpha + \Delta k_{\parallel}/2}^{-k_{\parallel}^\alpha - \Delta k_{\parallel}/2} \right] \\ &\times \int_{k_{\perp}^\alpha - \Delta k_{\perp}/2}^{k_{\perp}^\alpha + \Delta k_{\perp}/2} \left(\sum_i y_i e^{i\mathbf{k} \cdot \mathbf{r}_i} \right) \left(\sum_j y_j e^{-i\mathbf{k} \cdot \mathbf{r}_j} \right) \\ &\times |\Phi(\mathbf{k})|^2 \frac{k_{\perp} d\theta dk_{\perp} dk_{\parallel}}{(2\pi)^3}. \end{aligned} \quad (38)$$

The two sums inside the integral are very nearly discrete, three-dimensional Fourier transforms. All that remains is to discretize the Fourier space conjugate variable \mathbf{k} as we have already discretized the real space variable \mathbf{r} .

In order to evaluate the outer integrals, we approximate them as a sum over grid points in Fourier space. The most natural choice for discretization in \mathbf{k} is one that follows naturally from the FFT of \mathbf{y} in real space. If our box is of size $\ell_x \ell_y \ell_z$ and broken into $n_x \times n_y \times n_z$ voxels,⁸ we have that

$$\begin{aligned} \mathbf{r}_j &= \left(\frac{j_x \ell_x}{n_x}, \frac{j_y \ell_y}{n_y}, \frac{j_z \ell_z}{n_z} \right), \\ \text{where } j_x, j_y, j_z &\in \left\{ -\frac{n_{x,y,z}}{2}, \dots, 0, \dots, \frac{n_{x,y,z}}{2} - 1 \right\}. \end{aligned} \quad (39)$$

The natural three-dimensional Fourier space discretization is

⁸For simplicity and consistency we assume that n_x , n_y , and n_z are all even and we take the origin to be the second of the two center bins.

$$\mathbf{k}_m = \left(\frac{2\pi m_x}{\ell_x}, \frac{2\pi m_y}{\ell_y}, \frac{2\pi m_z}{\ell_z} \right)$$

$$\text{where } m_x, m_y, m_z \in \left\{ -\frac{n_{x,y,z}}{2}, \dots, 0, \dots, \frac{n_{x,y,z}}{2} - 1 \right\} \quad (40)$$

with a Fourier space voxel volume

$$(\Delta k)^3 = \frac{2\pi}{\ell_x} \times \frac{2\pi}{\ell_y} \times \frac{2\pi}{\ell_z}. \quad (41)$$

With this choice of discretization, we will simplify our integrals by sampling Fourier space with delta functions, applying the approximation in the integrand of Eq. (38) that

$$1 \approx \sum_m \frac{(2\pi)^3 \delta^3(\mathbf{k} - \mathbf{k}_m)}{\ell_x \ell_y \ell_z}. \quad (42)$$

This simplifies Eq. (39) considerably,

$$\begin{aligned} \hat{q}^\alpha &= \frac{1}{2} \sum_m \left(\sum_i y_i e^{i\mathbf{k}_m \cdot \mathbf{r}_i} \right) \\ &\times \left(\sum_j y_j e^{-i\mathbf{k}_m \cdot \mathbf{r}_j} \right) \chi^\alpha(\mathbf{k}_m) |\Phi(\mathbf{k}_m)|^2 / (\ell_x \ell_y \ell_z). \end{aligned} \quad (43)$$

If we define $\tilde{y}_m \equiv \sum_j y_j e^{-i\mathbf{k}_m \cdot \mathbf{r}_j}$, then we can write \hat{q} as

$$\begin{aligned} \hat{q}^\alpha &\approx \frac{1}{2\ell_x \ell_y \ell_z} \sum_m \tilde{y}_m^* \tilde{y}_m \chi^\alpha(\mathbf{k}_m) |\Phi(\mathbf{k}_m)|^2 \\ &= \frac{1}{2\ell_x \ell_y \ell_z} \sum_m |\tilde{y}_m|^2 \chi^\alpha(\mathbf{k}_m) |\Phi(\mathbf{k}_m)|^2. \end{aligned} \quad (44)$$

This result makes a lot of sense; after all, the power spectrum is—very roughly speaking—the data Fourier transformed, squared, and binned with an appropriate convolution kernel.

This is a very quick way to calculate \hat{q} because we can compute \tilde{y} , in $\mathcal{O}(N \log N)$ time (if we already have \mathbf{y}), and then we simply need to add $|\tilde{y}_m|^2$ for every m , weighted by the value of the analytic function $|\Phi(\mathbf{k}_m)|^2$ to the appropriate band power α .⁹ Each value of $|\tilde{y}_m|^2$ gets mapped uniquely to one value of α , so there are only N steps involved in performing the binning. Unlike in the LT method, we perform the calculation of \hat{q}^α for all values of α simultaneously.

However, the FFT approximation to Q_{ij}^α from Eq. (42) does not work very well at large values of $(\mathbf{r}_i - \mathbf{r}_j)$ because the discrete version of \mathbf{Q}^α does not sample the continuous

⁹For simplicity, we choose band power spectrum bins with the same width as our Fourier space bins (before zero padding). This linear binning scheme makes plotting, which is typically logarithmic in the literature, more challenging. On the other hand, it better spreads out the number of Fourier space data cube bins assigned to each band power.

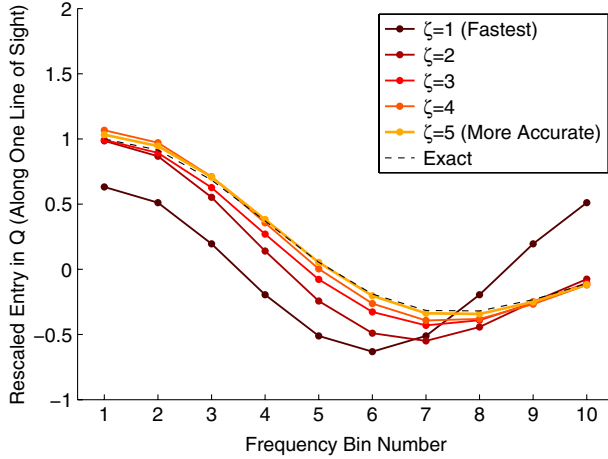


FIG. 3 (color online). We use an FFT-based technique to approximate the action of the matrix \mathbf{Q}^α that encodes the Fourier transforming, binning, and pixelization factors. In this figure, we show how the approximation improves with different factors of the zero padding parameter, ζ , while varying a single coordinate of one of the \mathbf{Q}^α matrices. For a fairly small value of ζ , the approximation is quite good, meaning that the binning and Fourier transforming step contributes subdominantly to the complexity of the overall algorithm.

version of \mathbf{Q}^α very finely. This can be improved by zero padding the input vector y_i , embedding it inside of a data cube of all zeros a factor of ζ^3 larger. For simplicity, we restrict ζ to integer values where $\zeta = 1$ represents no zero padding. By increasing our box size, we decrease the step size in Fourier space and thus the distance between each grid point in Fourier space where we sample k with delta functions. Repeating the derivation from Eqs. (39) through (44) yields

$$\hat{q}^\alpha \approx \frac{1}{2\ell_x \ell_y \ell_z \zeta^3} \sum_m |\tilde{y}_m|^2 \chi^\alpha(\mathbf{k}_m) |\Phi(\mathbf{k}_m)|^2, \quad (45)$$

where $\tilde{\mathbf{y}}$, has been zero padded and then Fourier transformed. This technique of power spectrum estimation scales as $\mathcal{O}(\zeta^3 N \log N)$, which is fine as long as ζ is small.¹⁰ In Fig. 3 we see how increasing ζ from 1 to 5 greatly improves accuracy.

C. Inverse variance weighting with the conjugate gradient method

We now know how to calculate \hat{q}^α quickly provided that we can also calculate $\mathbf{y} \equiv \mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ quickly. The latter turns out to be the most challenging part of the problem; we will address the various difficulties that it presents in this section through Sec. III E. We take our inspiration for a solution from a similar problem that the WMAP team

¹⁰Though not the computational bottleneck, this step is the most memory intensive; it involves writing down an array of $\zeta^3 N$ double-precision complex numbers. This can reach into the gigabytes for very large data cubes.

faced in making their maps. They employed the preconditioned conjugate gradient method to great success [57,58].

The conjugate gradient method [59] is an iterative technique for solving a system of linear equations such as $\mathbf{C}\mathbf{y} = (\mathbf{x} - \langle \mathbf{x} \rangle)$. Although directly solving this system involves inverting the matrix \mathbf{C} , the conjugate gradient method can approximate the solution to arbitrary precision with only a limited number of multiplications of vectors by \mathbf{C} . If we can figure out a way to quickly multiply vectors by \mathbf{C} by investigating the structure of its constituent matrices, then we can fairly quickly approximate \mathbf{y} . We will not spell out the entire algorithm here but rather refer the reader to the helpful and comprehensive description of it in Ref. [60].

Whenever iterative algorithms are employed, it is important to understand how quickly they converge and what their rates of convergence depend upon. If we are trying to achieve an error ε on our approximation \mathbf{y}_{CGM} to \mathbf{y} where

$$\varepsilon \equiv \frac{|\mathbf{C}\mathbf{y}_{\text{CGM}} - (\mathbf{x} - \langle \mathbf{x} \rangle)|}{|\mathbf{x} - \langle \mathbf{x} \rangle|} \quad (46)$$

and where $|\mathbf{x}| \equiv (\sum_i x_i^2)^{1/2}$ is the length of the vector \mathbf{x} , then the number of iterations required to converge (ignoring the accumulation of round-off error) is bounded by [60],

$$n \leq \frac{1}{2} \sqrt{\kappa} \ln\left(\frac{2}{\varepsilon}\right), \quad (47)$$

where κ is the condition number of the matrix (not to be confused with κ used elsewhere as a spectral index), defined as the ratio of its largest eigenvalue to its smallest,

$$\kappa(\mathbf{C}) \equiv \frac{\lambda_{\max}(\mathbf{C})}{\lambda_{\min}(\mathbf{C})} \quad (48)$$

Because n only depends logarithmically on ε , the convergence of the conjugate gradient method is exponential. In order to make the algorithm converge in only a few iterations, it is necessary to ensure that κ is not too large. This turns out to be a major hurdle that we must overcome, because we will routinely need to deal with covariance matrices with $\kappa(\mathbf{C}) \approx 10^8$ or worse. This dynamic range problem is unavoidable; it comes directly from the ratio of the brightest foregrounds, typically hundreds of kelvin, to the noise and signal, typically tens of millikelvin. That factor, about 10^4 , enters squared into the covariance matrices, yielding condition numbers of roughly 10^8 . In Sec. III E we will explain the efforts we undertake to mitigate this problem.

D. Foreground and noise covariance matrices

Before we can go about ensuring that the conjugate gradient method converges quickly, we must understand the detailed structure of the constituent matrices of \mathbf{C} . In particular, we will show that these matrices can all be multiplied by vectors in $\mathcal{O}(N \log N)$ time. We will first

examine the new kind of foreground we want to include, resolved point sources, which will also provide a useful example for how the foreground covariances can be quickly multiplied by vectors.

1. Resolved point sources

Unlike LT, we do not assume that bright point sources have already been cleaned out of our map. Rather we wish to unify the framework for accounting for both resolved and unresolved foregrounds by inverse covariance weighting. This will allow us to directly calculate how our uncertainties about the fluxes and spectral indices of these point sources affect our ability to measure the 21 cm power spectrum.

In contrast to the unresolved point sources modeled by \mathbf{U} , we model N_R bright resolved point sources as having known positions¹¹ with different fluxes S_n (at reference frequency ν_*) and spectral indices κ_n , neither of which is known perfectly. We assume that resolved point source contributions to \mathbf{x} are uncorrelated with each other, so we can define an individual covariance matrix \mathbf{R}_n for each point source. This means that our complete model for \mathbf{R} is

$$\mathbf{R} \equiv \sum_n \mathbf{R}_n. \quad (49)$$

Following LT, we can express the expected brightness temperature in a given voxel along the line of sight of the n th point source by a probability distribution for flux, $p_{S_n}(S')$, and spectral index, $p_{\kappa_n}(\kappa')$, that are both Gaussians with means S_n and κ_n and standard deviations σ_{S_n} and σ_{κ_n} , respectively. Following the derivation in LT, this yields

$$\begin{aligned} \langle x_i \rangle_n &= \delta_{ii_n} (1.4 \times 10^{-3} \text{ K}) \left(\frac{\Omega_{\text{pix}}}{1 \text{ sr}} \right)^{-1} \eta_i^{-2} \\ &\quad \times \int_{-\infty}^{\infty} \left(\frac{S'}{1 \text{ Jy}} \right) p_{S_n}(S') dS' \int_{-\infty}^{\infty} \eta_i^{-\kappa'} p_{\kappa_n}(\kappa') d\kappa' \\ &= \delta_{ii_n} (1.4 \times 10^{-3} \text{ K}) \left(\frac{S_n}{1 \text{ Jy}} \right) \left(\frac{\Omega_{\text{pix}}}{1 \text{ sr}} \right)^{-1} \\ &\quad \times \eta_i^{-2-\kappa_n} \exp \left[\frac{\sigma_{\kappa_n}^2}{2} (\ln \eta_i)^2 \right], \end{aligned} \quad (50)$$

where again $\eta_i \equiv \nu_i/\nu_*$. Here δ_{ii_n} is a Kronecker delta that forces $\langle x_i \rangle$ to be zero anywhere other than the line of sight corresponding to the n th resolved point source. Likewise, we can write down the second moment,

¹¹If the data cube is not overresolved, this assumption should be pretty good. If a point source appears to fall in two or more neighboring pixels, it could be modeled as two independent point sources in this framework. An even better choice would be to include the correlations between the two pixels, which would be quite strong. Modeling those correlations could only improve the results, since it would represent including additional information about the foregrounds, though it might slow down the method slightly. Not accounting for position uncertainty will cause the method to underestimate the ‘‘wedge’’ feature [7,61–64].

$$\begin{aligned} \langle x_i x_j \rangle_n &= \delta_{ii_n} \delta_{jj_n} (1.4 \times 10^{-3} \text{ K})^2 (\eta_i \eta_j)^{-2} \\ &\quad \times \left(\int_{-\infty}^{\infty} \left(\frac{S'}{1 \text{ Jy}} \right)^2 p_{S_n}(S') dS' \right) \times \left(\frac{\Omega_{\text{pix}}}{1 \text{ sr}} \right)^{-2} \\ &\quad \times \left(\int_{-\infty}^{\infty} (\eta_i \eta_j)^{-\kappa'} p_{\kappa_n}(\kappa') d\kappa' \right) \\ &= \delta_{ii_n} \delta_{jj_n} (1.4 \times 10^{-3} \text{ K})^2 (\eta_i \eta_j)^{-2-\kappa_n} \\ &\quad \times \left(\frac{S_n^2 + \sigma_{S_n}^2}{(1 \text{ Jy})^2} \right) \left(\frac{\Omega_{\text{pix}}}{1 \text{ sr}} \right)^{-2} \times \exp \left[\frac{\sigma_{\kappa_n}^2}{2} (\ln \eta_i \eta_j)^2 \right], \end{aligned} \quad (51)$$

where we assume $\sigma_{S_n} \approx 5\%$ of S_n and $\sigma_{\kappa_n} \approx 0.2$.

We know that $\langle x_i \rangle_n \langle x_j \rangle_n$ can be quickly multiplied by a vector because it is a rank 1 matrix. Therefore, in order to show that all of \mathbf{R} can be quickly multiplied, we recast $\langle x_i x_j \rangle_n$ as the product of matrices that can be multiplied by a vector in $\mathcal{O}(N \log N)$ or faster. If we then ignore the constants and just look at the parts of this matrix that depend on coordinates, we have that

$$\begin{aligned} \langle x_i x_j \rangle_n &\propto \delta_{ii_n} \delta_{jj_n} (\eta_i \eta_j)^{-2-\kappa_n} \exp \left[\frac{\sigma_{\kappa_n}^2}{2} (\ln \eta_i \eta_j)^2 \right] \\ &= \delta_{ii_n} (\eta_i)^{-2-\kappa_n} \exp \left[\frac{\sigma_{\kappa_n}^2}{2} (\ln \eta_i)^2 \right] \\ &\quad \times \exp \left[-\frac{\sigma_{\kappa_n}^2}{2} \left(\ln \frac{\eta_i}{\eta_j} \right)^2 \right] \times \delta_{jj_n} (\eta_j)^{-2-\kappa_n} \\ &\quad \times \exp \left[\frac{\sigma_{\kappa_n}^2}{2} (\ln \eta_j)^2 \right]. \end{aligned} \quad (52)$$

This matrix can be separated into the product of three matrices: one diagonal matrix that only depends on η_i , an inner matrix that includes the logarithm of a quotient of η_i and η_j in the exponent, and another diagonal matrix that only depends on η_j . The diagonal matrices can be multiplied by a vector in $\mathcal{O}(n_z)$. Moreover, because our cubes have redshift ranges $\Delta z < 0.5$, the frequencies at i and j are never very far apart, and we can make the approximation that

$$\begin{aligned} \ln \left(\frac{\eta_i}{\eta_j} \right) &= \ln \left(\frac{\nu_i}{\nu_j} \right) \\ &\approx \frac{\nu_0 + \Delta \nu_i}{\nu_0 + \Delta \nu_j} - 1 = \frac{1}{\nu_0} (\Delta \nu_i - \Delta \nu_j), \end{aligned} \quad (53)$$

where $\Delta \nu_i \equiv \nu_i - \nu_0$ and ν_0 is a constant reference frequency close to both ν_i and ν_j . We choose the center frequency of the data cube to be ν_0 . We can see now by combining Eqs. (52) and (53) that the inner matrix in our decomposition of the second moment depends only on the magnitude of the difference between ν_i and ν_j . In the approximation that the physical size of the data cube is small enough that frequencies map linearly to distances, this shows that \mathbf{R}_n respects translational invariance along the line of sight.

Because the entries in this inner part of \mathbf{R}_n only depend on differences in frequencies, the inner matrix is a diagonal-constant or ‘‘Toeplitz’’ matrix. Toeplitz matrices have the fortuitous property that they can be multiplied by vectors in $\mathcal{O}(N \log N)$, as we explain in Appendix A. Therefore, we can multiply \mathbf{R}_n by a vector in $\mathcal{O}(n_z \log n_z)$, and we can multiply \mathbf{R} by a vector faster than $\mathcal{O}(N \log N)$.

We can understand this result intuitively as a consequence of the fact that the inner part of \mathbf{R}_n is translationally invariant along the line of sight. Matrices that are translationally invariant in real space are diagonal in Fourier space. That we need to utilize this trick involving circulant and Toeplitz matrices is a consequence of the fact that our data cube is neither infinite nor periodic.

2. Unresolved point sources and the galactic synchrotron radiation

Let us now take what we learned in Sec. III D 1 (and Appendix A) to see if \mathbf{U} , can also be quickly multiplied by a vector. Looking back at Eq. (28), we can see that our job is already half finished; $(U_\perp)_{ij}$ only depends on the absolute differences between $(r_\perp)_i$ and $(r_\perp)_j$. Likewise, we can perform the exact same trick we employed in Eqs. (52) and (53) to write down the relevant parts of $(U_\parallel)_{ij}$ from Eq. (30) with the approximation that $\Delta\nu_i$ is always small relative to ν_0 ,

$$(U_\parallel)_{ij} \propto \exp\left[-\frac{\sigma_{\kappa_n}^2}{2\nu_0^2} (\Delta\nu_i - \Delta\nu_j)^2\right]. \quad (54)$$

In fact, we can decompose \mathbf{U} , as a tensor product of three matrices sandwiched between two diagonal matrices,

$$\mathbf{U} = \mathbf{D}_U[\mathbf{U}_x \otimes \mathbf{U}_y \otimes \mathbf{U}_z]\mathbf{D}_U. \quad (55)$$

where all three inner matrices are Toeplitz matrices. When we wish to multiply \mathbf{U} by a vector, we simply pick out one dimension at a time and multiply every segment of the data by the appropriate Toeplitz matrix (e.g., every line of sight for \mathbf{U}_z). All together, the three sets of multiplications can be done in $\mathcal{O}(n_x n_y n_f \log n_f) + \mathcal{O}(n_x n_f n_y \log n_y) + \mathcal{O}(n_y n_f n_x \log n_x) = \mathcal{O}(N \log N)$ time.

TABLE I. Due to the symmetries or approximate symmetries our models for the foreground and noise covariance matrices, they can all be multiplied by a vector in $\mathcal{O}(N \log N)$ time or faster. Summarized above are the reasons why each matrix can be quickly multiplied by. Either the matrices respect translation invariance and thus Toeplitz symmetry, their components are uncorrelated between lines of sight or frequencies, making them diagonal in real space, or they are uncorrelated and thus diagonal in Fourier space. In each case, the symmetries rely on the separability of the modeled covariance matrices into the tensor product of parts parallel or perpendicular to the line of sight.

Covariance Matrix	Parallel to the Line of Sight	Perpendicular to the Line of Sight
R : Resolved Point Sources	Toeplitz symmetry	Diagonal in real space
U : Unresolved Point Sources	Toeplitz symmetry	Toeplitz symmetry
G : Galactic Synchrotron Radiation	Toeplitz symmetry	Toeplitz symmetry
N : Instrumental Noise	Diagonal in real space	Diagonal in Fourier space

Moreover, since \mathbf{G} has exactly the same form as \mathbf{U} , albeit with different parameters, \mathbf{G} too can be multiplied by a vector in $\mathcal{O}(N \log N)$ time by making the same approximation that we made in Eq. (53).

3. Instrumental noise

Lastly, we return now to the form of \mathbf{N} we introduced in Sec. II D 3. To derive a form, we combine Eqs. (32) and (33). The details are presented in Appendix B, so here we simply state the result,

$$\mathbf{N} = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}} \mathbf{F}_\perp, \quad (56)$$

where \mathbf{F}_\perp and \mathbf{F}_\perp^\dagger are the unitary discrete two-dimensional Fourier and inverse Fourier transforms and where

$$\tilde{N}_{lm} = \frac{\lambda^4 T_{\text{sys}}^2 j_0^2(k_{x,l} \Delta x / 2) j_0^2(k_{y,l} \Delta y / 2)}{A_{\text{ant}}^2 (\Omega_{\text{pix}})^2 n_x n_y \Delta \nu} \frac{\delta_{lm}}{t_l}. \quad (57)$$

Here, A_{ant} is the effective area of a single antenna, $\Delta\nu$ is the frequency channel width, l and m are indices that index over both uv cells and frequencies, and t_l is the total observation time in a particular uv cell at a particular frequency.

Because this matrix is diagonal, we have therefore shown that \mathbf{N} , along with \mathbf{R} , \mathbf{U} , and \mathbf{G} , can be multiplied by a vector in $\mathcal{O}(N \log N)$. We have summarized the results for all four matrices in Table I.

4. Eliminating unobserved modes with the psuedoinverse

In our expression for the noise covariance in Eq. (57), we are faced with the possibility that t_l could be zero for some values of l , leading to infinite values of N_{ij} . Fourier modes with $t_l = 0$ correspond to parts of the uv plane that are not observed by the instrument, i.e., to modes containing no cosmological information. We can completely remove these modes by means of the ‘‘psuedoinverse’’ [45], which replaces \mathbf{C}^{-1} in the expression $\mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ and optimally weights all observed modes (this removal can itself be thought of as an optimal weighting—the optimal weight being zero). The psuedoinverse involves $\mathbf{\Pi}$, a projection matrix ($\mathbf{\Pi}^\dagger = \mathbf{\Pi}$ and $\mathbf{\Pi}^2 = \mathbf{\Pi}$) whose eigenvalues are 0 for modes that we want to eliminate

and 1 for all other modes. It can be shown [45] that the quantity we want to calculate for inverse variance weighting is not $\mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ but rather the quantity where

$$\mathbf{C}^{-1} \rightarrow \mathbf{\Pi}[\mathbf{\Pi}\mathbf{C}\mathbf{\Pi} + \gamma(\mathbf{I} - \mathbf{\Pi})]^{-1}\mathbf{\Pi}. \quad (58)$$

In this equation, γ can actually be any number other than 0. The term in brackets in the above equation replaces the eigenvalues of the contaminated modes of \mathbf{C} with γ . The outer $\mathbf{\Pi}$ matrices then project those modes out after inversion. In this paper, we take $\gamma = 1$ as the convenient choice for the preconditioner we will develop in Sec. III E.

The ability to remove unobserved modes is also essential for analyzing real data cubes produced by an interferometer. Interferometers usually produce so-called ‘‘dirty maps,’’ which are corrected for the effects of the primary beam but have been convolved by the synthesized beam, represented by the matrix \mathbf{B} ,

$$\mathbf{x}_{\text{dirty map}} = \mathbf{B}\mathbf{x}. \quad (59)$$

To compute \mathbf{x} for our quadratic estimator, we need to invert \mathbf{B} . Since the synthesized beam matrix is diagonal in Fourier space, this would be trivial were it not for unobserved baselines that make \mathbf{B} uninvertable. This can be accomplished with the psuedoinverse as well, since the modes that would have been divided by 0 when inverting \mathbf{B} are precisely the modes that we will project out via the psuedoinverse. We can therefore comfortably take

$$\mathbf{x} = \mathbf{F}_{\perp}^{\dagger} \mathbf{\Pi}[\mathbf{\Pi}\tilde{\mathbf{B}}\mathbf{\Pi} + \gamma(\mathbf{I} - \mathbf{\Pi})]^{-1} \mathbf{\Pi}\mathbf{F}_{\perp} \mathbf{x}_{\text{dirty map}}, \quad (60)$$

where $\mathbf{B} \equiv \mathbf{F}_{\perp}^{\dagger} \tilde{\mathbf{B}} \mathbf{F}_{\perp}$ and $\tilde{\mathbf{B}}$ is diagonal.

The psuedoinverse formalism can be usefully extended to any kind of mode we want to eliminate. One especially useful application would be to eliminate frequency channels contaminated by radio frequency interference or adversely affected by aliasing or other instrumental issues.

E. Preconditioning for fast conjugate gradient convergence

We have asserted that the quantity $\mathbf{y} \equiv \mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ can be estimated quickly using the conjugate gradient method as long as the condition number $\kappa(\mathbf{C})$ is reasonably small. Unfortunately, this is never the case for any realistic data cube we might analyze. In Fig. 4 we plot the eigenvalues of \mathbf{C} and its constituent matrices for a small data cube (only $6 \times 6 \times 8$ voxels) taken from a larger, more representative volume. In this example, $\kappa(\mathbf{C}) \approx 10^8$, which would cause the conjugate gradient method to require tens of thousands of iterations to converge. This is typical; as we discussed in Sec. III C, values of around 10^8 are to be expected. We need to do better.

1. The form of the preconditioner

The core idea behind ‘‘preconditioning’’ is to avoid the large value of $\kappa(\mathbf{C})$ by introducing a pair of preconditioning matrices \mathbf{P} and \mathbf{P}^{\dagger} . Instead of solving the linear system $\mathbf{C}\mathbf{y} = (\mathbf{x} - \langle \mathbf{x} \rangle)$, we solve the mathematically equivalent system,

$$\mathbf{C}'\mathbf{y}' = \mathbf{P}(\mathbf{x} - \langle \mathbf{x} \rangle), \quad (61)$$

where $\mathbf{C}' \equiv \mathbf{P}\mathbf{C}\mathbf{P}^{\dagger}$ and $\mathbf{y}' \equiv (\mathbf{P}^{\dagger})^{-1}\mathbf{y}$. If we can compute $\mathbf{P}(\mathbf{x} - \langle \mathbf{x} \rangle)$ and, using the conjugate gradient method on

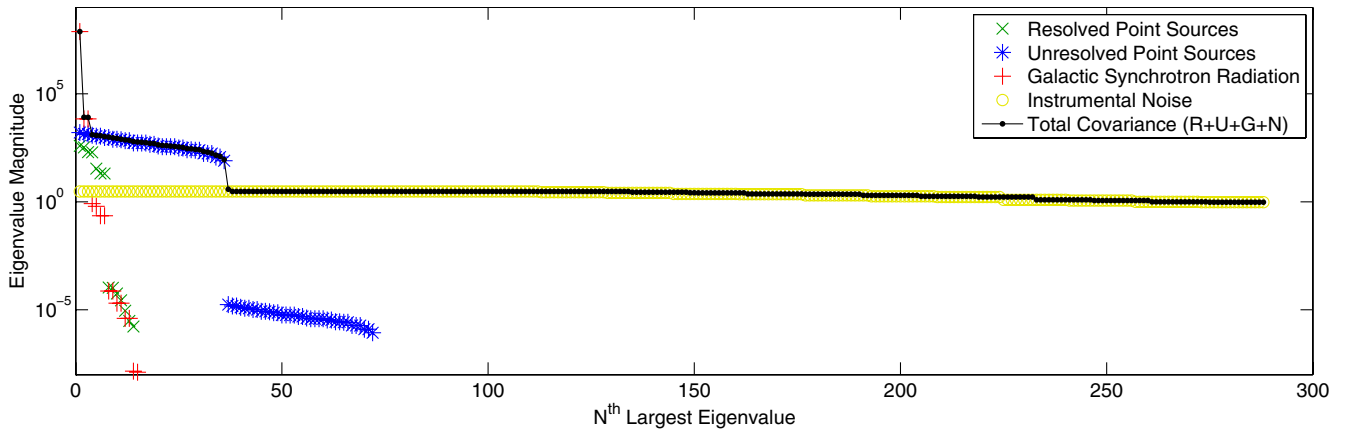


FIG. 4 (color online). The distinct patterns in the eigenvalue spectrum of our covariance matrix provide an angle of attack for making the calculation of $\mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$ numerically feasible via preconditioning. The plotted eigenvalue spectra of the covariance for a very small data cube exemplifies many of the important characteristics of the constituent matrices. First, notice that the noise eigenvalue spectrum, while flatter than any of the others, is not perfectly flat. The condition number of N is related to the ratio of the observing times in the most and least observed cell in the uv plane. Sometimes this factor can be 10^3 or 10^4 . Another important pattern to notice are the fundamental differences between the eigenvalue spectra of \mathbf{U} , \mathbf{G} , and \mathbf{R} . First off, \mathbf{R} has mostly zero eigenvalues, because \mathbf{R} is a block diagonal matrix with most of its blocks equal to zero. Second, despite the fact that \mathbf{U} and \mathbf{G} have nearly identical mathematical forms, \mathbf{U} has stair-stepping eigenvalue spectrum while that of \mathbf{G} is a much clearer exponential falloff. This is due to the much stronger correlations perpendicular to the line of sight in \mathbf{G} .

C' , we can solve for \mathbf{y}' and thus finally find $\mathbf{y} = \mathbf{P}^\dagger \mathbf{y}'$. If \mathbf{P} and \mathbf{P}^\dagger are matrices that can be multiplied by quickly and if $\kappa(C') \ll \kappa(C)$, then we can greatly speed up our computation of $\mathbf{y} = \mathbf{C}^{-1}(\mathbf{x} - \langle \mathbf{x} \rangle)$. Our goal is to build up preconditioning matrices specialized to the forms of the constituent matrices of \mathbf{C} . We construct preconditioners for $\mathbf{C} = \mathbf{N}$, generalize them to $\mathbf{C} = \mathbf{U} + \mathbf{N}$, and then finally incorporate \mathbf{R} and \mathbf{G} to build the full preconditioner.

The result is the following:

$$\mathbf{C}' = \mathbf{F}_\perp^\dagger \mathbf{P}_U \mathbf{P}_\Gamma \mathbf{P}_N(\mathbf{C}) \mathbf{P}_N^\dagger \mathbf{P}_\Gamma^\dagger \mathbf{P}_U^\dagger \mathbf{F}_\perp. \quad (62)$$

Where \mathbf{P}_U , \mathbf{P}_Γ and \mathbf{P}_N and preconditioners for \mathbf{U} , $\mathbf{\Gamma} \equiv \mathbf{R} + \mathbf{G}$, and \mathbf{N} respectively. A complete and pedagogical explanation of this preconditioner and the motivation for its construction and complex form can be found in Appendix C. The definitions of the matrices can be found in Eqs. (C13), (C25), and (C2), respectively.

Despite its complex form and construction, the procedure reduces $\kappa(C)$ by many orders of magnitude. In Fig. 5, in explicit contrast to Fig. 4, we see a demonstration of that effect.

2. Computational complexity of the preconditioner

In Appendix C, we briefly discuss how the different steps in computing and applying this preconditioner scale with the problem size. If any of them scale too rapidly with N , we can quickly lose the computational advantage of our method over that of LT.¹²

First, let us enumerate the complexity of setting up the preconditioner for each matrix. \mathbf{P}_N requires no setup since it only involves computing powers of the diagonal matrix $\tilde{\mathbf{N}}$ (see Appendix C 1). \mathbf{P}_U requires the eigenvalue decomposition of \mathbf{U}_z , the component of \mathbf{U} along the line of sight, which takes $\mathcal{O}(n_z^3)$ time (see Appendix C 2).

We need the eigensystems of \mathbf{R} and \mathbf{G} to compute the eigensystem of $\mathbf{\Gamma}$ for \mathbf{P}_Γ (see Appendix C). \mathbf{R} requires performing one eigenvalue decomposition of an $n_z \times n_z$ matrix for every resolved point source; that takes $\mathcal{O}(N_R n_z^3)$ time. \mathbf{G} simply requires three eigenvalue decompositions: one for each matrix like those that appear for \mathbf{U} in Eq. (C3) whose total outer product is \mathbf{G} . Thus, the complexity is $\mathcal{O}(n_x^3) + \mathcal{O}(n_y^3) + \mathcal{O}(n_z^3)$.

Next, we need to compute the eigenvalues of $\mathbf{\Gamma}_{\perp,k}$, the components of $\mathbf{\Gamma}$ perpendicular to the line of sight corresponding to each of the “relevant” (i.e., much bigger than the noise floor) eigenvalues of $\mathbf{\Gamma}$ along the line of sight (see Appendix C 3 for a more rigorous definition). Using the notation we develop in Appendix C 2, we denote the number of relevant eigenvalues of a matrix \mathbf{M} as $m(\mathbf{M})$. The number of times we need to decompose an $n_x n_y \times n_x n_y$ matrix is generally equal to the number of

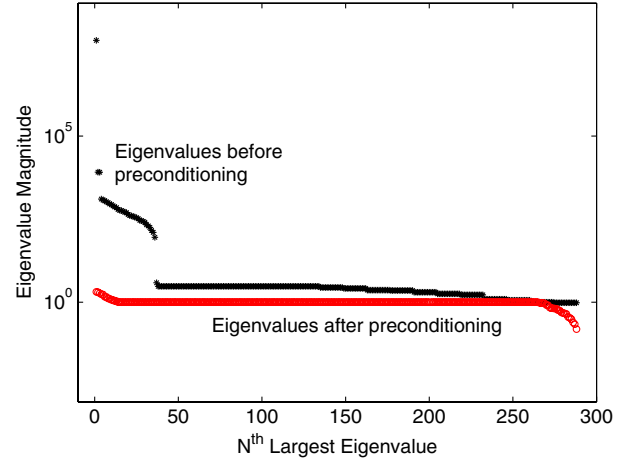


FIG. 5 (color online). The preconditioner for the conjugate gradient method that we have devised significantly decreases the range of eigenvalues of \mathbf{C} . Our preconditioner attempts to whiten the eigenvalue spectra of the constituent matrices of \mathbf{C} sequentially, first \mathbf{N} , then \mathbf{R} and \mathbf{G} together, and finally \mathbf{U} . By preconditioning, the condition number $\kappa(C)$, the ratio of the largest to smallest eigenvalues, is reduced from over 10^8 to about 10^1 .

relevant eigenvalues of \mathbf{G}_z , since the number of relevant eigenvectors is almost always the same for \mathbf{G} and \mathbf{R} . So we have then a computational complexity of $\mathcal{O}(m(\mathbf{G}_z) \times (n_x n_y)^3)$. Given the limited angular resolution of the experiment and the flat sky approximation, we generally expect n_x and n_y to be a good deal smaller than n_f , making this scaling more tolerable. All these scalings are summarized in Table II.

Until now, all of our complexities have been $\mathcal{O}(N \log N)$ or smaller. Because these small incursions into bigger complexity classes are only part of the set-up cost, they are not intolerably slow as long as $m(\mathbf{G}_z)$ is small. This turns out to be true because the eigenvalue spectra of \mathbf{R}_n and \mathbf{G}_z fall off exponentially, meaning that we expect the number of relevant eigenvalues to grow only logarithmically. This is borne out in Fig. 6 where we see exactly how the number of eigenvalues that need to be preconditioned scales with the problem size.

Let us now turn to a far more important scaling: that of multiplying the preconditioner by a vector. The set-up needs to be done only once per Fisher matrix calculation; the preconditioning needs to happen for every iteration of the conjugate gradient method. \mathbf{P}_N is the easiest; we only ever need to perform a Fourier transform or multiply by a diagonal matrix. The complexity is merely $\mathcal{O}(N \log N)$. \mathbf{P}_U only involves multiplying by vectors for each relevant eigenvalue of \mathbf{U}_z , so the total complexity is $\mathcal{O}(Nm(\mathbf{U}_z))$.

Finally, we need to assess the complexity of applying \mathbf{P}_Γ . When performing the eigenvalue decomposition of $\mathbf{\Gamma}_{\perp,k}$, we expect roughly the same number of eigenvalues to be important that would have been important from \mathbf{R} and \mathbf{G} separately for that k index. Each of those

¹²This section may be difficult to follow without first reading Appendix C. However, the key results can be found in Table II.

TABLE II. The computational complexity of setting up the preconditioner is, at worst, roughly $\mathcal{O}(N^2)$, though this operation only needs to be performed once. Even for large data cubes, this is not the rate-limiting step in power spectrum estimation. The computational complexity of applying the preconditioner ranges from $\mathcal{O}(N \log N)$ to $\mathcal{O}(NN_R)$. For large data cubes with hundreds of bright point sources, the preconditioning time is dominated by \mathbf{P}_Γ , which is in turn dominated by preconditioning associated with individual point sources. The computational complexity of the preconditioner therefore depends on the number of point sources considered “resolved,” which scales with both field of view and with the flux cut. Here N_R is the number of resolved point sources in our field of view, n_d is the size of the box in voxels along the d^{th} dimension, and m is the number of relevant eigenvalues of a matrix above the noise floor that need preconditioning.

Operation	Complexity
Compute \mathbf{U} eigensystem	$\mathcal{O}(n_z^3)$
Compute \mathbf{G} eigensystems	$\mathcal{O}(n_x^3) + \mathcal{O}(n_y^3) + \mathcal{O}(n_z^3)$
Compute \mathbf{R} eigensystems	$\mathcal{O}(N_R n_z^2)$
Compute $\mathbf{\Gamma}$ eigensystems	$\mathcal{O}(m(\mathbf{G}_z)(n_x n_y)^3)$
Apply \mathbf{P}_N	$\mathcal{O}(N \log N)$
Apply \mathbf{P}_U	$\mathcal{O}(Nm(\mathbf{U}_z))$
Apply \mathbf{P}_Γ	$\mathcal{O}(Nm(\mathbf{G})) + \mathcal{O}(NN_R m(\mathbf{R}_n))$

eigenvectors takes $\mathcal{O}(N)$ time to multiply by a vector. So we expect to deal with $m(\mathbf{G})$ eigenvalues from \mathbf{G} and one eigenvalue from each resolved point source for each relevant value of k , or about $N_R m(\mathbf{R}_n)$. Applying \mathbf{P}_Γ therefore is $\mathcal{O}(Nm(\mathbf{G})) + \mathcal{O}(NN_R m(\mathbf{R}_n))$. If we keep the same minimum flux for the definition of a resolved point source and if we scale our cube uniformly in all three spatial directions, then $N_R \propto N^{2/3}$.

This turns out to be the rate-limiting step in the entire algorithm. If we decide instead to only consider the brightest N_R to be resolved, regardless of box size, then applying \mathbf{P}_Γ reduces to $\mathcal{O}(N \log N)$. Likewise, if we are only interested in expanding the frequency range of our data cube, the scaling also reduces to $\mathcal{O}(N \log N)$. We can comfortably say then that the inclusion of a model for resolved point sources introduces a complexity bounded by $\mathcal{O}(N \log N)$ and $\mathcal{O}(N^{5/3})$. We can see the precise computational effect of the preconditioner when we return in Sec. IV C to assess the overall scaling of the entire algorithm. These results are also summarized in Table II.

3. Preconditioner results

Choosing which eigenvalues are “relevant” in the constituent matrices of \mathbf{C} and therefore need preconditioning depends on how these eigenvalues compare to the noise floor. In Appendix C 2, we define a threshold θ which distinguishes relevant from irrelevant eigenvalues by comparing them to θ times the noise floor. Properly choosing a value for θ , the threshold below which we do not precondition eigenvalues of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{\Gamma}}$, presents a tradeoff. We

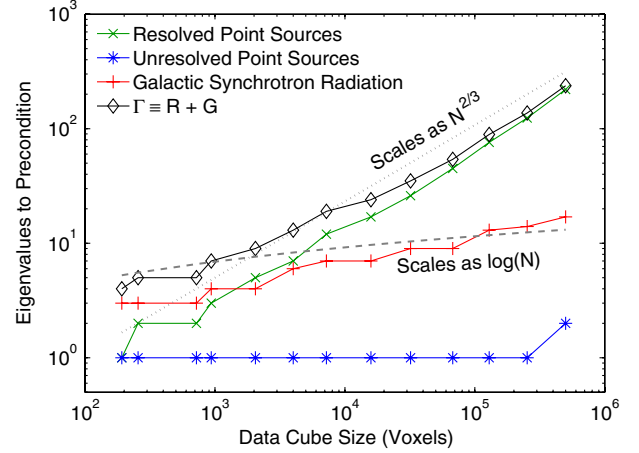


FIG. 6 (color online). For large data cubes and a fixed definition of what constitutes a “bright” point source, the complexity of preconditioning is dominated by the number of resolved point sources. Specifically, the complexity of preconditioning for $\mathbf{\Gamma}$ scales as $N^{2/3}$ because the number of resolved point sources is simply proportional to the solid angle of sky surveyed, which scales with the survey volume (and thus number of voxels, assuming fixed angular and frequency resolution) to the $2/3$ power. This also confirms our assertion that the number of important eigenvalues of \mathbf{G} and \mathbf{U} should scale logarithmically with data cube size (albeit with a different prefactor). Each of the data cubes is taken from the same survey with the same ratio of width to depth. The number of eigenvalues to precondition is computed assuming an eigenvalue threshold of $\theta = 1$.

expect that that too low of a value of θ will precondition inconsequential eigenvalues, thus increasing the conjugate gradient convergence time. We also expect that too large of a value of θ will leave some of the most important eigenvalues without any preconditioning, vastly increasing convergence time. Both of these expectations are borne out by our numerical experiments, which we present in Fig. 7.

In this work, we choose $\theta = 1$ (all foreground eigenvalues above the noise floor are preconditioned) for simplicity and to be sure that we are not skipping the preconditioning of any important foreground eigenvalues. One might also worry that more iterations of the algorithm provides more opportunity for round-off error to accumulate and prevent convergence, as has sometimes proven the case in our numerical experiments. For lengthy or repeated calculations of the Fisher matrix, it is wise to explore the performance of several levels of preconditioning, especially if it can garner us a another factor of 2 in speed.

F. Fast simulation of foregrounds and noise

We concluded Sec. III A with the fact that a Monte Carlo calculation of the Fisher matrix required the ability to compute $\hat{\mathbf{q}}$ from many different realizations of the foregrounds and noise modeled by \mathbf{C} . In Secs. III B through III E, we have shown how to quickly calculate $\hat{\mathbf{q}}$ from a data vector \mathbf{x} using Eq. (10).

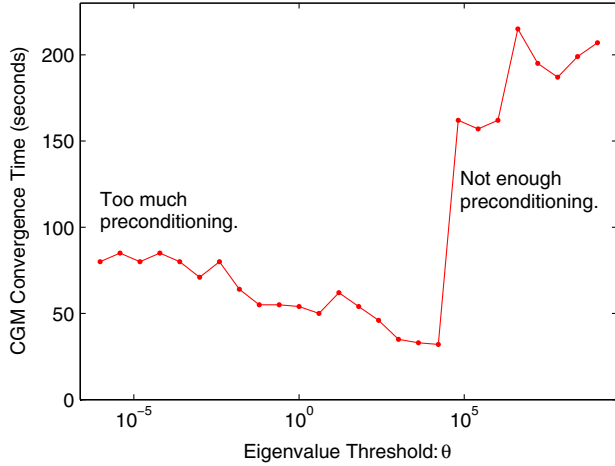


FIG. 7 (color online). This plot shows how computational time scales with θ , the threshold for preconditioning, for the conjugate gradient method performed on an $N \approx 10^4$ voxel data cube. It appears that for this particular covariance matrix, a minimum exists near $\theta = 10^4$. At the minimum, the greater number of conjugate gradient iterations are balanced by quicker individual iterations (since each iteration involves less preconditioning). We can see from this plot that there exists a critical value of θ around 5×10^4 where the preconditioning of a small number of additional eigenvalues yields a large effect on the condition number of the resultant matrix. Without preconditioning, sufficiently large values of $\kappa(\mathbf{C})$ could also lead to the accumulation of roundoff error that prevents convergence of the conjugate gradient method.

But where does \mathbf{x} come from? When we want to estimate the 21 cm temperature power spectrum of our Universe, \mathbf{x} will come from data cubes generated from real observations. But in order to calculate \mathbf{F} , which is essential both to measuring $\hat{\mathbf{p}}$ and estimating the error on that measurement, we must first be able to create many realizations of \mathbf{x} drawn from our models for noise and foregrounds that we presented in Sec. III D.

A mathematically simple way to draw \mathbf{x} from the right covariance matrix is to create a vector \mathbf{n} of independent and identically distributed random numbers drawn from a normal distribution with mean 0 and standard deviation 1. Then, it is easy to see that

$$\mathbf{x} \equiv \mathbf{C}^{1/2} \mathbf{n} \quad (63)$$

is a random vector with mean $\mathbf{0}$ and covariance \mathbf{C} . Unfortunately, computing $\mathbf{C}^{1/2}$ is just as computationally difficult as computing \mathbf{C}^{-1} .

In this last section of our presentation of our fast method for power spectrum estimation and statistics, we will explain how a vector can be created randomly with covariance \mathbf{C} . We do so by creating vectors randomly from each constituent matrix of \mathbf{C} , since each contribution to the measured signal is uncorrelated. In Sec. III F 5, we will demonstrate numerically that these simulations can be performed quickly while still being accurately described by the underlying statistics.

1. Resolved point sources

The simplest model to reproduce in a simulation is the one for resolved point sources, because the covariance was created from a supposed probability distribution over their true fluxes and spectral indices. We start with a list of point sources with positions and with a specified but uncertain fluxes and spectral indices. These fluxes can either come from a simulation, in which case we draw them from our source count distribution [Eq. (29)] and spectral indices from a Gaussian distribution, or from a real catalog of sources with its attendant error bars. The list of sources does not change over the course of calculating the Fisher matrix.

In either case, calculating a random \mathbf{x}_R requires only picking two numbers, a flux and a spectral index, for each point source and then calculating a temperature in each voxel along that particular line of sight. The latter is easy, since we assume it is drawn from a Gaussian. The former can be quickly accomplished by numerically calculating the cumulative probability distribution from Eq. (29) and inverting it. Each random \mathbf{x}_R is therefore calculable in $\mathcal{O}(N_R n_z) < \mathcal{O}(N)$ time.

2. Unresolved point sources

We next focus on \mathbf{U} , which is more difficult. Our goal is to quickly produce a vector with specified mean and covariance. LT has already established what value we want for $\langle \mathbf{x}_U \rangle$ and $\langle \mathbf{x}_G \rangle$ with a calculation very similar to Eq. (50). We need to figure out how to produce a vector with zero mean and the correct covariance.

One way around the problem of calculating $\mathbf{C}^{1/2}$ is to take advantage of the eigenvalue decomposition of the covariance matrix. That is because if $\mathbf{C} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$, where \mathbf{Q} is the matrix that transforms into the eigenbasis and $\mathbf{\Lambda}$ is a diagonal matrix made up of the eigenvalues, then $\mathbf{C}^{1/2} = \mathbf{Q} \mathbf{\Lambda}^{1/2} \mathbf{Q}^T$. We already found the few important eigenvalues of \mathbf{U} for our preconditioner (see Sec. C2), so does this technique solve our problem?

Yes and no. In the direction parallel to the line of sight, this technique works exceedingly well because only a small number of eigenvectors correspond to non-negligible eigenvalues. We can, to very good approximation, ignore all but the largest eigenvalues (which correspond to the first few “steps” in Fig. 4.) We can therefore generate random unresolved point source lines of sight in $\mathcal{O}(n_z m(\mathbf{U}_z))$ with the right covariance.

A problem arises, however, when we want to generate \mathbf{x}_U with the proper correlations perpendicular to the line of sight. Unlike the extremely strong correlations parallel to the line of sight, these correlations are quite weak. Weak correlations entail many comparable eigenvalues; in the limit that point sources were uncorrelated, $\mathbf{U}_x \otimes \mathbf{U}_y \rightarrow \mathbf{I}_\perp$ and all the eigenvalues would be 1 (though the eigenvectors would of course be much simpler too). Utilizing the

same technique as above would require a total complexity of $\mathcal{O}(Nn_xn_y)$ time, which is slower than we would like.

However, the fact that both \mathbf{U}_x and \mathbf{U}_y are Toeplitz matrices allows us to use the same sort of trick we employed to multiply our Toeplitz matrices by vectors in Sec. III D 1 to draw random vectors from $\mathbf{U}_x \otimes \mathbf{U}_y$ [65]. It turns out that the circulant matrix in which we embed our covariance matrix must be positive-semidefinite for this technique to work. Although there exists such an embedding for any Gaussian covariance matrix, only Gaussians with coherence lengths small compared to the box size can be embedded in a reasonably small circulant matrix—exactly the situation we find ourselves in with \mathbf{U}_\perp . As such, we can generate random \mathbf{x}_U vectors in $\mathcal{O}(Nm(\mathbf{U}_z) \log(n_xn_y)) \approx \mathcal{O}(N \log N)$.

3. Galactic synchrotron radiation

The matrix \mathbf{G} differs from \mathbf{U} primarily in the coherence length perpendicular to the line of sight. Unlike \mathbf{U} , \mathbf{G} has only a small handful of important eigenvalues, which means that random \mathbf{x}_G vectors can be generated in the same way we create line of sight components for \mathbf{x}_U vectors, which we described above. Since $m(\mathbf{G})$ is so small (see Fig. 4) and grows so slowly with data cube size (see Fig. 6), we can create random \mathbf{x}_G vectors in approximately $\mathcal{O}(N)$.

4. Instrumental noise

Finally, we turn to \mathbf{N} , which is also mathematically simple to simulate. First off, $\langle \mathbf{x}_N \rangle = 0$. Next, because \mathbf{N} is diagonal in the Fourier basis, we can simply use Eq. (63). Because $\mathbf{N} = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}} \mathbf{F}_\perp$,

$$\mathbf{N}^{1/2} = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}}^{1/2} \mathbf{F}_\perp, \quad (64)$$

which is computationally easy to multiply by \mathbf{n} because $\tilde{\mathbf{N}}$ is a diagonal matrix. The most computationally intensive step in creating random \mathbf{x}_N vectors is the fast Fourier transform, which of course scales as $\mathcal{O}(N \log N)$.

5. Data simulation speed and accuracy

Before we conclude this section and move on to the results of our method as a whole, we verify what we have claimed in the above sections, namely, that we can quickly generate data cubes with the correct covariance properties. Figure 8 verifies the speed, showing that the algorithm is both fast and well behaved for large data cubes.

In order to show that the sample covariance of a large number of random \mathbf{x} vectors converges to the appropriate covariance matrix, we must first define a convergence statistic, ε . We are interested in how well the matrix converges relative to the total covariance matrix \mathbf{C} . For example, for \mathbf{R} we choose

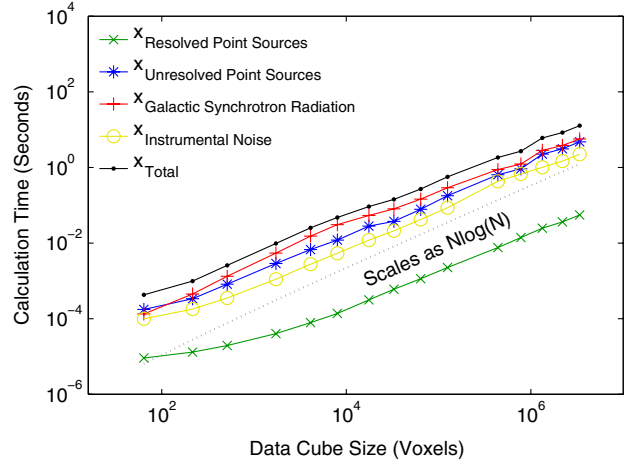


FIG. 8 (color online). In order to estimate the Fisher matrix via a Monte Carlo, we need to draw random data cubes from our modeled covariance. Here we show that we can do so in $\mathcal{O}(N \log N)$ by plotting computational time as a function of problem size for generating a random \mathbf{x} for each of the constituent sources of \mathbf{x} . In practice, generating random \mathbf{x} vectors is never the rate-limiting step in calculating \mathbf{F} .

$$\varepsilon(\hat{\mathbf{R}}) \equiv \sqrt{\frac{\sum_{ij} |\hat{R}_{ij} - R_{ij}|^2}{\sum_{ij} |C_{ij}|^2}}, \quad (65)$$

where $\hat{\mathbf{R}}$ is the sample covariance of n random \mathbf{x}_R vectors drawn from \mathbf{R} . If each \mathbf{x} is a Gaussian random vector then the expected RMS value of ε is

$$\sqrt{\langle \varepsilon(\hat{\mathbf{R}})^2 \rangle} = \frac{1}{\sqrt{n}} \left[\frac{\sum_{i,j} R_{ij}^2 + (\text{tr} \mathbf{R})^2}{\sum_{ij} C_{ij}^2} \right]. \quad (66)$$

In Fig. 9, we see that all four constituent matrices of \mathbf{C} converge like $n^{-1/2}$, as expected, with very nearly the prefactor predicted by Eq. (66). We can be confident, therefore, in both the speed and accuracy of our technique for generating random vectors.

G. Method accuracy and convergence

Before we move on to discuss some of the results of our method, it is worthwhile to check that no unwarranted approximations prevent it from converging to the exact form of the Fisher information matrix in Eq. (14). Since calculating \mathbf{F} exactly can only be done in $\mathcal{O}(N^3)$ time, we perform this test in two parts.

First, we measure convergence to the exact Fisher matrix for a very small data cube with only $6 \times 6 \times 8$ voxels. Taking advantage of Eq. (21), we generate an estimate of \mathbf{F} , which we call $\hat{\mathbf{F}}$, from the sample covariance of many independent $\hat{\mathbf{q}}$ vectors. We compare these $\hat{\mathbf{F}}$, which we calculate periodically along the course of the Monte Carlo, with the \mathbf{F} that we calculated directly using Eq. (14). As we

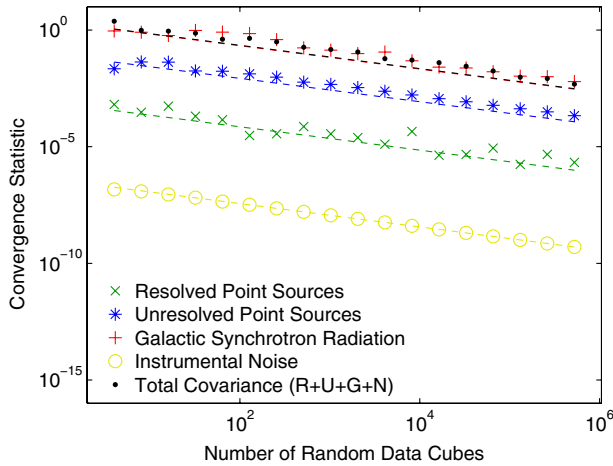


FIG. 9 (color online). We verify that our technique for quickly generating random data cubes actually reproduces the correct statistics by generating a large number of such cubes and calculating their sample covariances. Plotted here is the error statistic detailed in Eq. (65). The color-matched dotted lines are the expected convergences for correlated Gaussians from Eq. (66).

show in Fig. 10, the sample covariance of our $\hat{\mathbf{q}}$ vectors clearly follows the expected $n^{-1/2}$ convergence to the correct result.

However, we are more concerned with the accuracy of the method for large data cubes which cannot be tackled by the LT method. Unfortunately, for such large data cubes, we cannot directly verify our result except in the case where $\mathbf{C} = \mathbf{I}$. In concert with other tests for agreement with LT, we also check that the method does indeed converge as $n^{-1/2}$ by comparing the convergence of subsets of the $\hat{\mathbf{q}}^\alpha$ vectors up to $n/2$ Monte Carlo iterations to

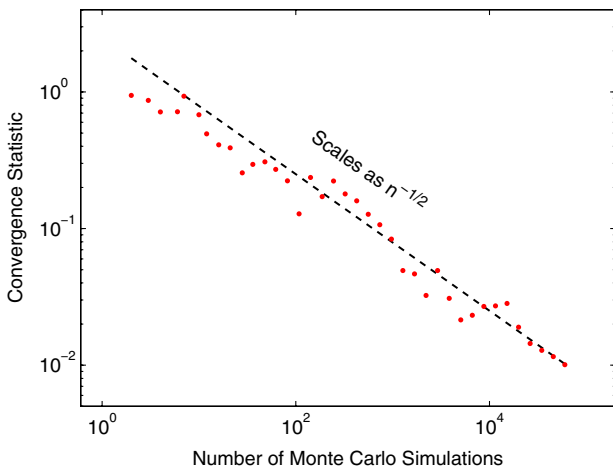


FIG. 10 (color online). Our Monte Carlo converges to the correct Fisher matrix as $n^{-1/2}$, as expected. In this plot, we compare the sample covariance of many $\hat{\mathbf{q}}$ vectors generated from small data cubes to an exact calculation of \mathbf{F} by calculating the relative error of their diagonals.

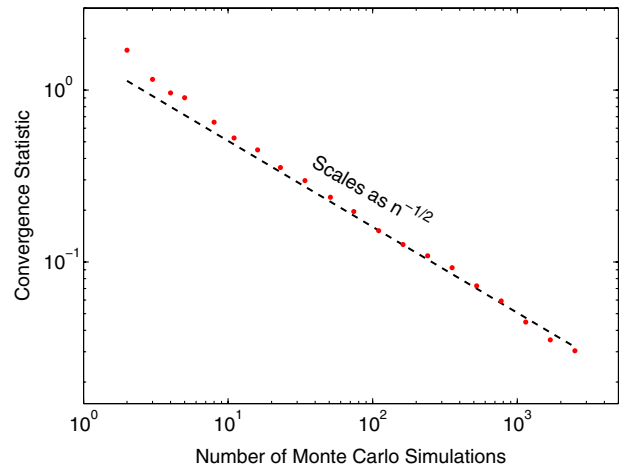


FIG. 11 (color online). For large data cubes, the convergence of the sample covariance of our $\hat{\mathbf{q}}$ vectors to \mathbf{F} also nicely follows the expected $n^{-1/2}$ scaling. We perform this analysis on a data cube with 1.5×10^5 voxels analogously to that which we performed in Fig. 10, except that we use the sample covariance of double the number of Monte Carlo iterations as our “true” Fisher matrix. This explains the artificially fast convergence we see in the last few points of the above plot.

the reference Fisher matrix, which we take to be the sample covariance of all n iterations. As we show in Fig. 11, our expectation is borne out numerically.

H. Method summary

We have constructed a technique that accelerates the LT technique to $\mathcal{O}(N \log N)$ and extends it to include bright point sources in, at worst, $\mathcal{O}(N^{5/3})$. We do so by generating random data vectors with the modeled foreground and noise covariances and calculating the Fisher information matrix via Monte Carlo. We are able to calculate individual inverse variance weighted power spectrum estimates quickly using the conjugate gradient method with a specially adapted preconditioner.

Our method makes a number of assumptions, most of which are not shared with the LT method. Our method can analyze larger data sets but at a slight loss of generality. Although we have mentioned these assumptions throughout this work, it is useful to summarize them in one place.

- (i) Our method relies on a small enough data cube perpendicular to the line of sight that it can be approximated as rectilinear (see Fig. 1).
- (ii) We approximate the natural log of the quotient of frequencies in the exponent of our point source covariance matrix by a leading-order Taylor expansion [Eq. (53)]. This assumption makes the foreground covariances translationally invariant along the line of sight and thus amenable to fast multiplication using Toeplitz matrix techniques. This is a justified assumption as long as the coherence length

of the foregrounds is much longer than the size of the box along the line of sight.

- (iii) Our ability to precondition our covariance matrix for the conjugate gradient method depends on the approximation that the correlation length of \mathbf{U} perpendicular to the line of sight, due to weak spatial clustering of point sources, is not much bigger than the pixel size. For the purposes of preconditioning, we approximate \mathbf{U}_\perp to be the identity (see Sec. C2). The longer the correlation length of \mathbf{U}_\perp , the longer the conjugate gradient algorithm will take to converge.
- (iv) Likewise, the speed of the preconditioned conjugate gradient algorithm depends on the similarities of the eigenmodes of the covariances for \mathbf{R} , \mathbf{U} , and \mathbf{G} along the line of sight. The more similar the eigenmodes are (though their accompanying eigenvalues can be quite different) the more the preconditioning algorithm can reduce the condition number of \mathbf{C} . We believe that this similarity is a fairly general property of models for foregrounds, though the introduction of a radically different foreground model might require a different preconditioning scheme.
- (v) We assume that the number of Monte Carlo iterations needed to estimate the Fisher information matrix is not so large that that it precludes analyzing large data cubes. Because the process of generating more artificial $\hat{\mathbf{q}}$ vectors is trivially parallelizable, we do not expect getting down to the requisite precision on the window functions to be an insurmountable barrier.

One common theme among these assumptions, especially the last three, is that the approximations we made to speed up the algorithm can be relaxed as long as we are willing to accept longer runtimes. This reflects the flexibility of the method, which can trade off speed for accuracy and vice versa.

IV. RESULTS

Now that we are confident that our method can accurately estimate the Fisher information matrix and can therefore calculate both power spectrum estimates from data and the attendant error bars and window functions, we turn to the first end-to-end results of the algorithm. In this section, we demonstrate the power our method and the improvements that it offers over that of LT. First, in Sec. IVA we show that our technique reproduces the results of LT in the regions of Fourier space where they overlap. Then in Sec. IVB, we highlight the improvements stemming from novel aspects of our method, especially the inclusion of the pixelization factor $|\Phi(\mathbf{k})|^2$ in \mathbf{Q}^α and \mathbf{N} and the separation of point sources into resolved point sources (\mathbf{R}) and unresolved point sources (\mathbf{U}), by showing how different parts of our algorithm affect \mathbf{F} . In Sec. IV C

we examine just how much faster our algorithm is than that of LT, and lastly in Sec. IV D we forecast the cosmological constraining power of the 128-tile deployment of the MWA.

A. Comparison to Liu and Tegmark

First we want to verify that our method reproduces that of LT in the regions of Fourier space accessible to both methods. Figure 12 provides an explicit comparison to LT's Figs. 2 and 8. These plots show the shaded regions representing their method and overplotted, white-outlined contours representing ours. Both are on the same color scale. These plots show error bars in temperature units in k_\perp - k_\parallel space and a selection of window functions in both the case where $\mathbf{C} = \mathbf{N}$ and $\mathbf{C} = \mathbf{U} + \mathbf{G} + \mathbf{N}$. They are generated from the same survey geometry with identical foreground and noise parameters. In the regions where the methods overlap, we see very good agreement between the two methods.

In addition to the modes shown in the shaded regions in Fig. 12, the LT method can access Fourier modes longer than the box size, which we cannot. This is no great loss—these modes are poorly constrained by a single data cube. Moreover, they are generally those most contaminated by foregrounds; the low- k_\perp modes will see heavy Galactic synchrotron contamination while the low- k_\parallel modes will be contaminated by types of the foregrounds. We imagine that very low- k_\perp Fourier modes, those that depend on correlations between data cubes that cannot be joined without violating the flat-sky approximation, will still be analyzed by the LT method. Because our method can handle many more voxels, it excels in measuring both medium and high- k modes that require high spectral and spatial resolution.

B. Novel effects on the Fisher matrix

A simple way to understand the different effects that our forms of \mathbf{C} and \mathbf{Q}^α have on the Fisher information matrix, especially the novel inclusions of \mathbf{R} and $|\Phi(\mathbf{k})|^2$, is to build up the Fisher matrix component by component. In Fig. 13 we do precisely that by plotting the diagonal elements of \mathbf{F} . These diagonal elements are related to the vertical error bars on our band powers. Large values of $F^{\alpha\alpha}$ correspond to band powers about which we have more information.

In the top two panels of Fig. 13, we show the first novel effect that our method takes into account. In them, we can see how modeling the finite size of our voxels affects the information available in the case where $\mathbf{C} = \mathbf{I}$ (the color scale for these two panels only is arbitrary). In the top left panel, we have set $|\Phi(\mathbf{k})|^2 = 1$, which corresponds to the delta function pixelization of LT. We see that the amount of information depends only on k_\perp . This is purely a binning effect: our bins in k_\perp are concentric circles with constant increments in radius; higher values of k_\perp incorporate more

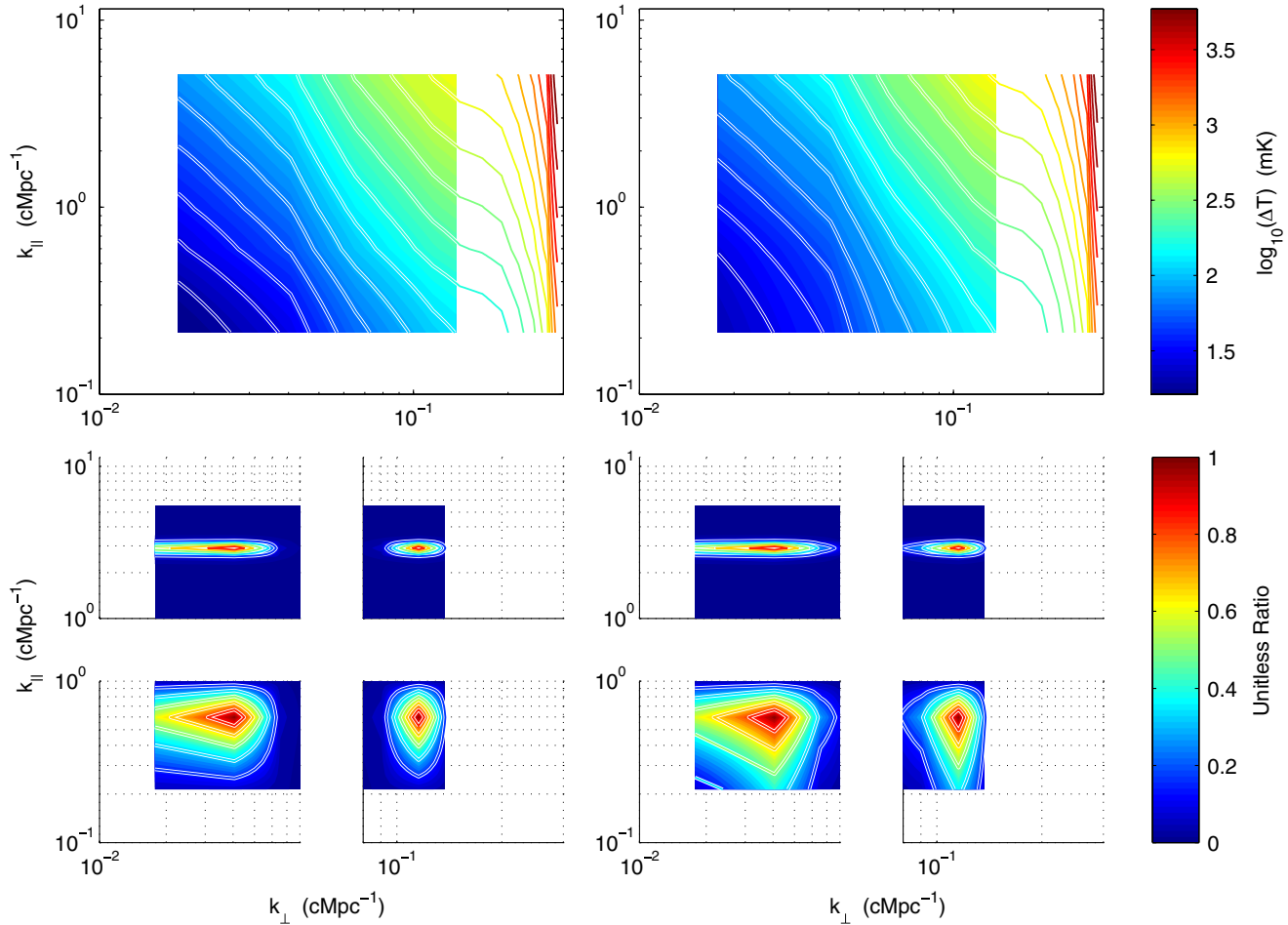


FIG. 12 (color online). Our method faithfully reproduces the results of LT in the regions of Fourier space accessible to both methods. Here we recreate both the vertical error bar contours from LT’s Fig. 8 (top two panels) and a few selected window functions from LT’s Fig. 2 (bottom two groups of panels). The shaded regions represent the LT results; the white-outlined, colored contours are overplotted to show our results. Both are on the same color scale. Following LT, we have plotted both the case without foregrounds ($\mathbf{C} = \mathbf{N}$, left two panels) and the case with foregrounds ($\mathbf{C} = \mathbf{N} + \mathbf{U} + \mathbf{G}$, right two panels), which allows us to get a sense for the effects of foregrounds on our power spectrum estimates, error bars, and window functions.

volume in Fourier space, except at the high- k_{\perp} edge where the circles are large enough to only include the corners of the data cube. In the top right panel, we see that including $|\Phi(\mathbf{k})|^2 \neq 1$ affects our ability to measure high- k modes, which depends increasingly on our real space resolution and is limited by the finite size of our voxels.

In the middle left panel, we now set $\mathbf{C} = \mathbf{N}$. In comparison to $\mathbf{C} = \mathbf{I}$, the new covariance matrix (and thus new vector \mathbf{x} for the Monte Carlo calculation of \mathbf{F}), shifts the region of highest information to a much lower value of k_{\perp} . Though there are fewer Fourier modes that sample this region, there are far more baselines in the array configuration at the corresponding baseline length. Our noise covariance is calculated according to our derivation in Sec. III D 3 for 1000 hours of observation with the 128-tile deployment of the MWA [27].

We next expand to $\mathbf{C} = \mathbf{U} + \mathbf{N}$ for the middle right panel, where we have classified all point sources as “unresolved.” In other words, we take S_{cut} in Eq. (31) to be large

(we choose 200 Jy, which is representative of some of the brightest sources at our frequencies). As we expect, smooth spectrum contamination reduces our ability to measure power spectrum modes with low values of k_{\parallel} . This is because of the exponentially decaying eigenvalue spectrum of \mathbf{U}_{\parallel} , most of which is smaller than the eigenvalues of \mathbf{N} . The effect is seen across k_{\perp} because the characteristic clustering scale of unresolved point sources is smaller than the pixel size; localized structure in real space corresponds to unlocalized power in Fourier space.

In the bottom left panel, we have included information about the positions of roughly 200 resolved point sources above 100 mJy, with random fluxes drawn from our source count distribution [Eq. (29)] and random spectral indices drawn from a Gaussian centered on $\kappa_n = 0.5$ with a width of 0.15. By doing this, we reduce S_{cut} in our model for \mathbf{U} down to 100 mJy. Including all this extra information—positions, fluxes, flux uncertainties, spectral indices, and spectral index uncertainties—provides us with significantly

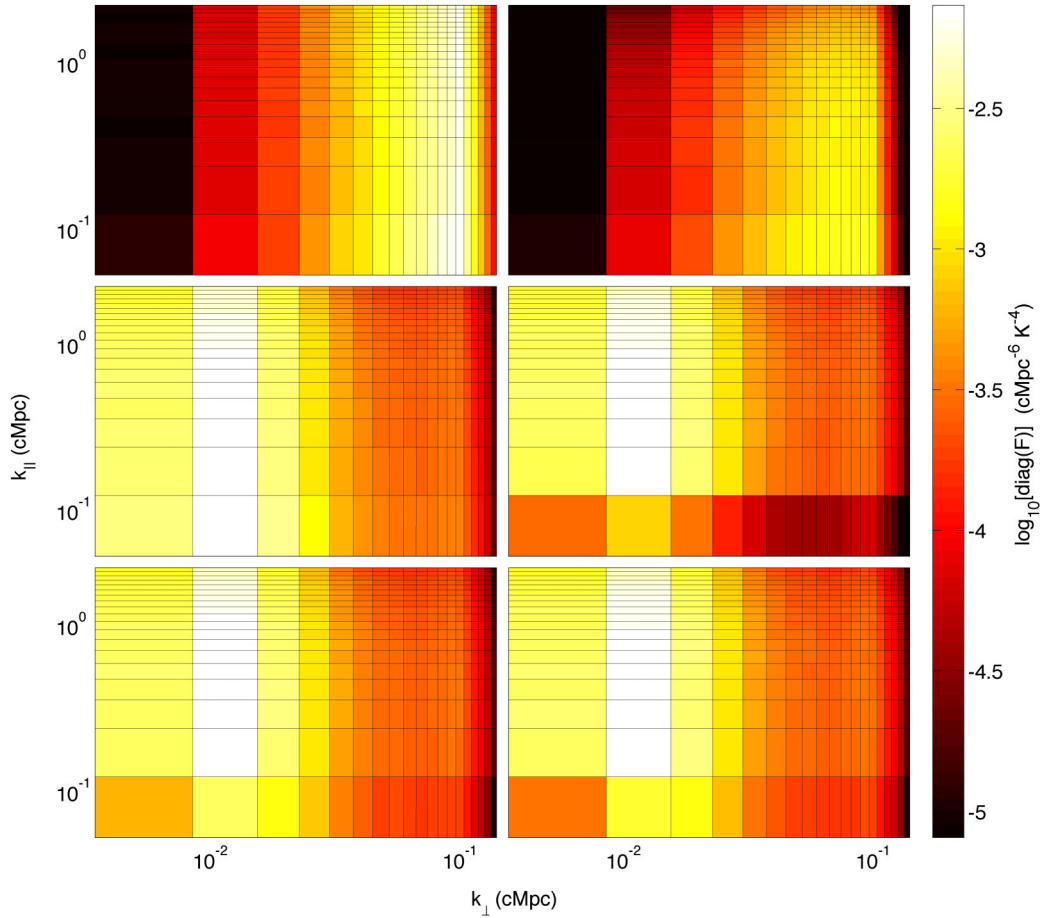


FIG. 13 (color online). The Fisher information matrix provides a useful window into understanding the challenges presented to measuring the 21 cm power spectrum by the various contaminants. In this figure, we add these effects one by one, reading from left to right and top to bottom, to see how the diagonal of the Fisher matrix (expanded along the k_{\perp} and k_{\parallel} directions), is affected. Brighter regions represent, roughly speaking, more information and thus smaller error bars (in power spectrum units). We comment in more detail on each panel individually in Sec. IV B, including upon the advantages of the novel aspects of our technique. *Top left*: the covariance matrix is taken to be the identity and pixelization effects on \mathbf{Q}^{α} are ignored. *Top right*: the pixelization factor $|\Phi(\mathbf{k})|^2$ is included and not set to 1. *Middle left*: the noise expected from 1000 hours of observation with the MWA 128-tile configuration is included. *Middle right*: all point sources (up to 200 Jy) are modeled as unresolved; all information about their positions is ignored. *Bottom left*: resolved point sources are included in the model, with all point sources dimmer than 100 mJy considered unresolved. *Bottom right*: in addition to bright point sources, galactic synchrotron is also included.

more Fisher information at low- k_{\parallel} , where foregrounds dominate, and thus smaller errors on those modes. Additionally, by incorporating resolved point sources as part of our inverse covariance weighting, we no longer have to worry about forward propagating errors from any external point-source subtraction scheme. In the left panel of Fig. 14 we see the ratio of this panel to the middle right panel.

Finally, in the bottom right panel of Fig. 13 we show the effect of including Galactic synchrotron radiation. Adding \mathbf{G} has the expected effect; we already know that \mathbf{G} has only a few important eigenmodes which correspond roughly to the lowest Fourier modes both parallel and perpendicular to the line of sight. As a result, we only see a noticeable effect in the bottom left corner of the k_{\perp} - k_{\parallel} plane; we include the ratio of the two figures in the right panel of

Fig. 14 for clarity. Otherwise, our Galaxy has very little effect on the regions of interest. In fact, the similarity between this panel and the middle left panel tells us something very striking: in the regions of Fourier space that our data most readily probes, foregrounds (once properly downweighted) should not prove an insurmountable obstacle to power spectrum estimation.

The set of plots in Fig. 13 is useful for developing a heuristic understanding of how noise and foregrounds affect the regions in which we can most accurately estimate the 21 cm power spectrum. With it, we can more easily identify the precise regions of k -space that we expect to be minimally contaminated by foregrounds and can thus tailor our instruments and our observations to the task of measuring the 21 cm power spectrum.

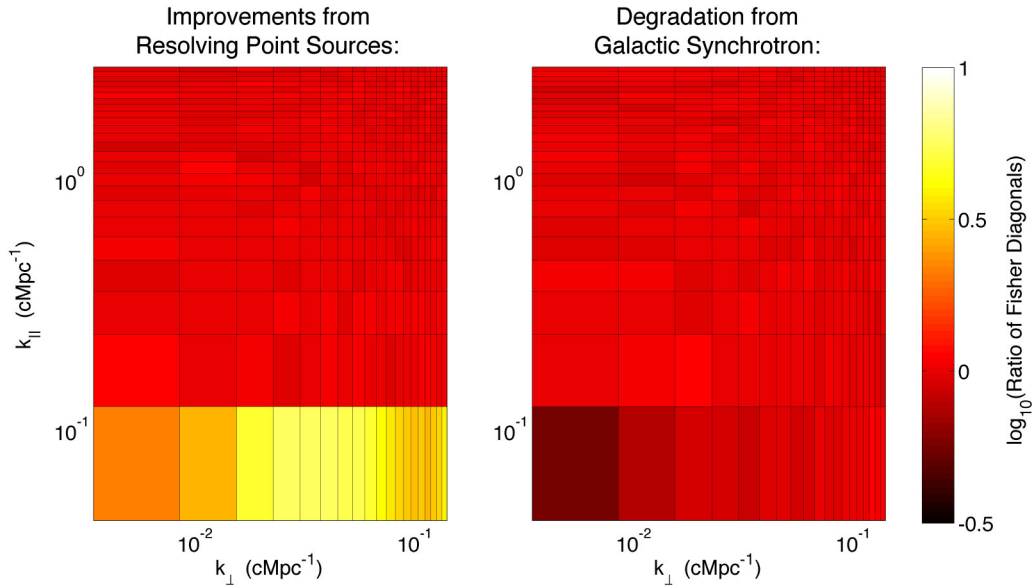


FIG. 14 (color online). By comparing the Fisher matrices arising from the various covariance models we explore in Sec. IV B and Fig. 13, the precise improvements are brought into sharper focus. In the left panel, we can see the information we gain by explicitly including resolved point sources. Shown here is the ratio of the bottom left panel of Fig. 13 to the middle right panel. By taking into account precise position, flux, and spectral information and uncertainties, we improve our ability to measure the power spectrum at the longest scales parallel to the line of sight, effectively ameliorating the effects of foregrounds. In the right panel, we see the remarkably small effect that the galactic synchrotron radiation has on our ability to measure the 21 cm power spectrum. Shown here is the ratio of the bottom right panel of Fig. 13 to the bottom left. Because we take spatial information into account, the strong spatial and spectral coherence of the signal from our Galaxy is confined to the bottom left corner of the k_{\parallel} - k_{\perp} plane.

C. Computational scaling of the method

Now that we understand how our technique works, we want to also see that it works as quickly as promised by and achieves the desired computational speed up over the LT method. Specifically, we want to show that we can achieve the theoretical $\mathcal{O}(N \log N)$ performance in reproducing the results of LT. We also want to better understand the computational cost of including resolved point sources so as to compare that cost to benefits outlined in Section IV B. We have therefore tested the algorithm's speed for a wide range of data cube sizes; we present the results of that study in Fig. 15.

In this figure, we show the combined setup and runtime for power spectrum estimates including 1000 Monte Carlo simulations of $\hat{\mathbf{q}}$ for estimating the Fisher matrix on a single modern CPU. For each successive trial, we scale the box by the same ratio in all three dimensions. Because we maintain a fixed flux cut, increasing the linear size of the box by a factor of 2 increases the number of resolved point sources in the box by a factor of 4 and the number of voxels by a factor of 8. With any more than a few point sources, the computational cost becomes dominated by point sources, leading to an overall complexity of $\mathcal{O}(NN_R)$. In this case, the largest data cubes include about 400 point sources over a field of about 50 square degrees, accounting for about 15% of the lines of sight in the data cube.

For any given analysis project, there exists a trade-off between including additional astrophysical information

into the analysis and the computational complexity of that analysis; at some point the marginal cost of a slower algorithm exceeds marginal benefit of including more bright point sources. It is beyond the scope of this paper to prescribe a precise rubric for where to draw the line between resolved and unresolved point sources. However, we can confidently say that the algorithm runs no slower than $\mathcal{O}(N^{5/3})$ and can often run at or near $\mathcal{O}(N \log N)$ if only the brightest few point sources are treated individually.

D. Implications for future surveys

Though the primary purpose of this paper is to describe an efficient method for 21 cm power spectrum analysis, our technique enables us immediately to make predictions about the potential performance of upcoming instruments. In this section we put all our new machinery to work in order to see just how well the upcoming 128-tile deployment of the MWA can perform.

We envision 1000 hours of integration on a field that is 9° on each side, centered on $z = 8$ with $\Delta z = 0.5$. With a frequency resolution of 40 kHz and an angular resolution of 8 arcminutes, our data cube contains over 10^6 voxels. We completed over 1000 Monte Carlo iterations on our 12-core server in about one week. We use the foreground parameters outlined above in Secs. II D and III D. In Figs. 16 through 18, we show the diagonal elements of the Fisher matrix we have calculated, the temperature power spectrum error bars, and a sampling of window functions.

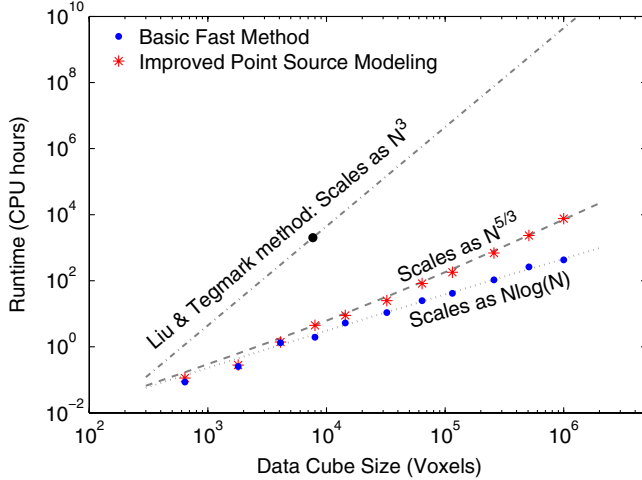


FIG. 15 (color online). Our algorithm scales with the number of voxels, N , as $\mathcal{O}(N \log N)$ in the best case and as $\mathcal{O}(N^{5/3})$ in the worst, depending on the treatment of bright point sources. If we choose to ignore all information about the location, brightness, and spectra of bright point sources, we can estimate the power spectrum in $\mathcal{O}(N \log N)$. If we choose to take into account this extra information, the algorithmic complexity increases to $\mathcal{O}(NN_R)$, where N_R is the number of bright, resolved point sources. For a fixed minimum flux for “bright” sources, this leads to $\mathcal{O}(N^{5/3})$ complexity for uniform scaling in all three dimensions. Both scenarios represent a major improvement over the LT method, which scales as $\mathcal{O}(N^3)$.

In Fig. 16 we plot the diagonal elements of the Fisher matrix, which are related directly to the power spectrum errors. Drawing from the discussion in Sec. IV B, we can see clearly the effects of the array layout (and thus noise), of foregrounds (included resolved point sources), and of pixelization. Interestingly, until pixelization effects set in at the highest values of k_{\parallel} , the least contaminated region spans a large range of values of k_{\parallel} . One way of probing more cosmological modes is to increase the frequency resolution of the instrument. The number of modes accessible to the observation scales with $(\Delta\nu)^{-1}$, though the amplitude of the noise scales with $(\Delta\nu)^{-1/2}$. As long as the noise level is manageable and the cosmological signal is not dropping off too quickly with k , increasing the frequency resolution seems like a good deal.

In Fig. 17 we show the vertical error bars that we expect on power spectrum estimates in temperature units. The most important fact about this plot is that there is a large region where we expect that vertical error bars will be sufficiently small that we should be able to detect a 10 mK signal with signal to noise greater than 1. This is especially the case at fairly small values of k , which is surprising since these k modes were supposed to be the most contaminated by foregrounds. There are two reasons why this happens.

First, the conversion to temperature units [Eq. (25)] introduces a factor of $(k_{\perp}^2 k_{\parallel})^{1/2}$ that raises the error bars

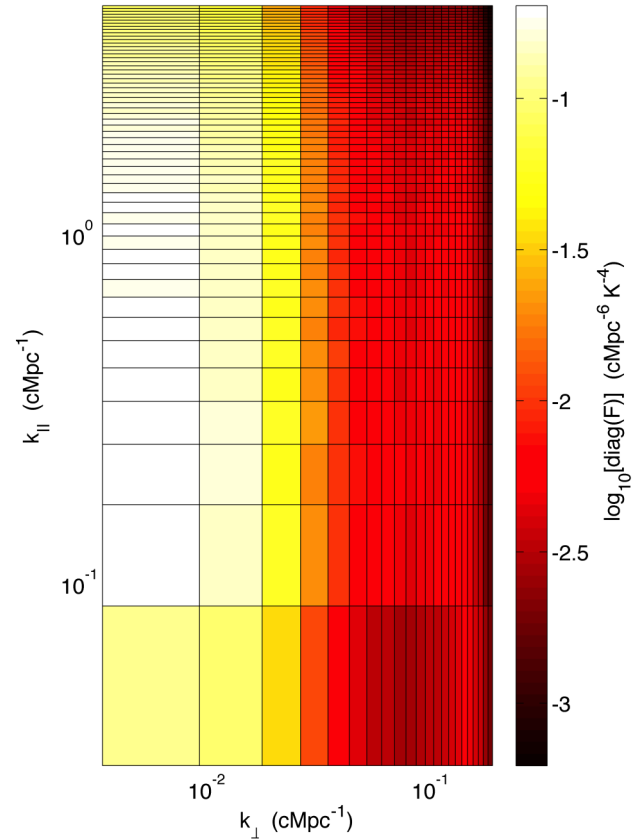


FIG. 16 (color online). The diagonal of the Fisher matrix predicted for 1000 hours of observation with the MWA with 128 tiles shows the region of power spectrum space least contaminated by noise and foregrounds. Noise, and thus array layout, dominates the shape of the region of maximum information, creating a large, vertical region at a value of k_{\perp} corresponding to the typical separation between antennas in the compact core of the array. The contaminating effects of the foregrounds are clearly visible at low- k_{\parallel} .

for larger values of k . Second, the strongest foreground modes overlap significantly with one of the $k = 0$ modes of the discrete Fourier transform, which we exclude for our power spectrum estimate (this is just the average value of the cube in all three directions, which is irrelevant to an interferometer that is insensitive to the average).

Another way to think about it is this: because the coherence length of the foregrounds along the line of sight is much longer than the size of any box small enough to comfortably ignore cosmological evolution, we expect that the most contaminated Fourier mode will be precisely the one we ignore. Unlike the LT method, our method cannot easily measure modes much longer than the size of the data cube. Along the line of sight, these modes have very wide window functions and are the most contaminated by foregrounds. Perpendicular to the line of sight, these modes are better measured by considering much larger maps where the flat sky approximation no longer holds. For the purposes of measuring these low- k modes,

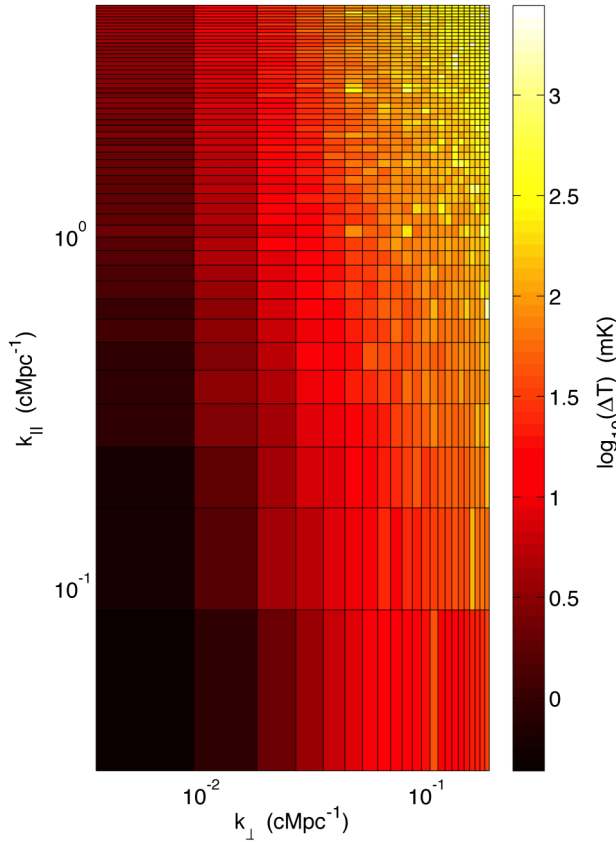


FIG. 17 (color online). The expected error bars in temperature units on decorrelated estimates of the power spectrum highlight a sizable region of k -space where we expect to be able to use the MWA with 128 tiles to detect a fiducial 10 mK signal with a signal to noise ratio greater than 1. Perhaps surprisingly, the smallest error bars are still on the smallest k modes accessible by our method, though some of them are contaminated by large foregrounds. This is because our conversion to temperature units includes a factor of $(k_{\perp}^2 k_{\parallel})^{1/2}$, which accounts for the difference between this figure and Fig. 16. From the shape of the region of smallest error, we can better appreciate the extent to which noise and our array layout determines where in k -space we might expect to be able to detect the EoR. The noisiness at high- k is due to Monte Carlo noise and can be improved with more CPU hours.

the LT method can provide a useful complement to ours. Large-scale modes from down-sampled maps can be measured by LT; smaller-scale modes from full-resolution maps can be measured by our method. Then both can be combined to estimate the power spectrum across a large range of scales.

And finally, in Fig. 18 we show many different window functions for a selection of values of k_{\perp} and k_{\parallel} that spans the whole space. In general, these window functions are quite narrow, meaning that each band power measurement probes only a narrow range of scales. The widest windows we see look wide for two reasons. First, linearly separated bins appear wider at low- k when plotted logarithmically. Second, foregrounds cause contaminated bins to leak into

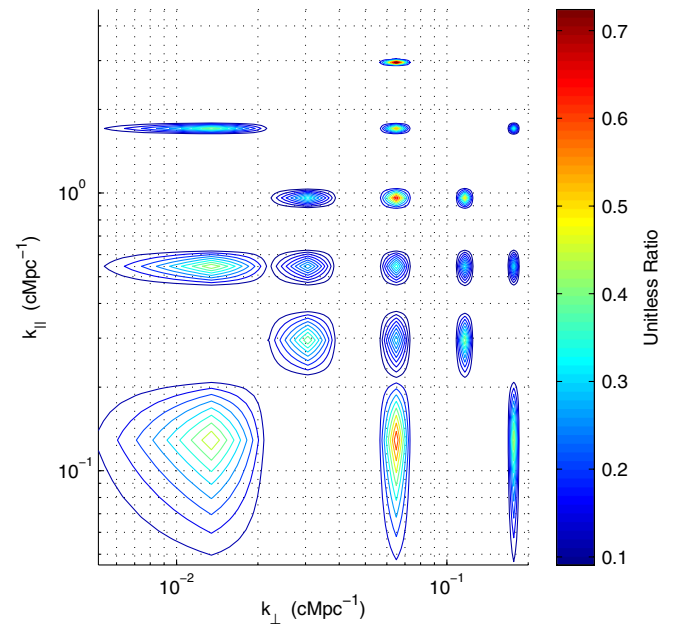


FIG. 18 (color online). We can see from a sampling of window functions that our band power spectrum estimates \hat{p}^{α} represent the weighted averages of p^{α} over a narrow range of scales, especially at higher values of k_{\perp} and k_{\parallel} . The widest window functions can be attributed to binning (with linearly binned data, low- k bins look larger on logarithmic axes) and to foregrounds. This is good news, because it will enable us to accurately make many independent measurements of the power spectrum and therefore better constrain astrophysical and cosmological parameters.

nearby bins, especially at low- k_{\parallel} and moderate k_{\perp} . We saw hints of this effect in Fig. 12 when comparing noise-only simulations to simulations with both noise and foregrounds.

In the vast majority of the k_{\perp} - k_{\parallel} plane, the window functions seem to be dominated by the central bin and neighbors. Except for edge cases, no window function has contributions exceeding 10% from bins outside the central bin and its nearest neighbors. This means that we should be happy with our choice of Fourier space binning, which was designed to have bin widths equal to those of our data cube before zero padding. We also know that significantly finer binning would be inappropriate, so we do not have to worry about the tradeoff between fine binning of the power spectrum and the inversion of the Fisher matrix. Therefore, with the 128-tile deployment of the MWA, we can be confident that our estimates of the power spectrum correspond to distinct modes of the true underlying power spectrum.

V. CONCLUSIONS

With this paper, we have presented an optimal algorithm for 21 cm power spectrum estimation which is dramatically faster than the Liu and Tegmark method [24], scaling as $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^3)$, where N is the number of

voxels in the three-dimensional sky map. By using the inverse variance weighted quadratic estimator formalism adapted to 21 cm tomography by the LT method, we preserve all accessible cosmological information in our measurement to produce the smallest possible error bars and narrow window functions. Moreover, our method can incorporate additional information about the brightest point sources and thus further reduce our error bars at the cost of some—but by no means all—of that computational advantage. Our method is highly parallelizable and has only modest memory requirements; it never needs to store an entire $N \times N$ matrix.

Our method achieves this computational speed-up for measuring power spectra, error bars, and window functions by eliminating the time-consuming matrix operations of the LT method. We accomplish this using a combination of Fourier, spectral, and Monte Carlo techniques which exploit symmetries and other physical properties of our models for the noise and foregrounds.

We have demonstrated the successful simulation of error bars and window functions for the sort of massive data set we expect from the upcoming 128-tile deployment of the MWA—a data set that cannot be fully utilized using only the LT method. Our forecast predicts that 1000 hours of MWA observation should be enough to detect the fiducial 10 mK signal across much of the $k_{\parallel}-k_{\perp}$ plane accessible to the instrument. Moreover, we predict that the horizontal error bars on each band power estimate will be narrow, allowing each estimate to probe only a small range of scales.

Our results suggest several avenues for further research. Of course, the most immediate application is to begin analyzing the data already being produced by interferometers like LOFAR, GMRT, MWA, and PAPER as they start accumulating the sensitivity necessary to zero in on a detection of the EoR. The large volume of data these instruments promise to produce might make it useful to explore ways of further speeding up the Monte Carlo estimation of the Fisher matrix. There is significant redundancy in our calculated Fisher matrix because the window function shapes vary only relatively slowly with k -scale. We believe that one can reduce the number of Monte Carlo simulations needed to attain the same accuracy by adding a postprocessing step that fits the Fisher matrix to a parametrized form. This should work best in the regions of the $k_{\parallel}-k_{\perp}$ plane that are fairly uncontaminated by foregrounds, where Fisher matrix elements are expected to vary most smoothly. It may also be possible to speed up the Monte Carlo estimation of the Fisher matrix using the trace evaluation technology of Ref. [33].

The forecasting power of our method to see whether a particular observing campaign might reveal a particular aspect the power spectrum need not be limited to measurements of the EoR. Our method provides an opportunity to precisely predict what kind of measurement, and what kind

of instrument, might be necessary for observing 21 cm brightness temperature fluctuations during the cosmic dark ages. Our method should prove useful for weighing a number of important design considerations: What is the optimal array configuration? What is the optimal survey volume? What about angular resolution? Spectral resolution? And in what sense are these choices optimal for doing astrophysics and cosmology?

To help answer such questions, our technique could be used to compare the myriad of ideas for and possible implementations of future projects like HERA and the SKA and even to help find an optimal proposal. For example, one plan for achieving large collecting area is building a hierarchically regular array (a so-called “Omniscopes”) that takes advantage of FFT correlation [31] and redundant baseline calibration [66]. There exist many array configurations that fit into this category and it is not obvious what the optimal Omniscopes might look like.

The quest to detect a statistical signal from the Epoch of Reionization is as daunting as it is exciting. It is no easy task to find that needle in a haystack of noise and foregrounds. However, now that we are for the first time armed with a method that can extract all the cosmological information from a massive data set without a prohibitive computational cost, we can feel confident that a sufficiently sensitive experiment can make that first detection—not just in theory, but also in practice.

ACKNOWLEDGMENTS

The authors wish to thank Gianni Bernardi, Michael Betancourt, Katherine Deck, Jacqueline Hewitt, Andrew Lutmirski, Michael Matejek, Miguel Morales, Ue-Li Pen, John Rutherford, Leo Stein, Christopher Williams, and Phillip Zuckin for many useful discussions and advice. This work is partially supported by the Bruno Rossi Graduate Fellowship and NSF Grant No. AST-1105835. A. L. acknowledges support from the Berkeley Center for Cosmological Physics.

APPENDIX A: TOEPLITZ MATRICES

In this appendix, we briefly review how to rapidly multiply by Toeplitz matrices. We need to employ the advantages of Toeplitz matrices because the assumption that our covariance matrices are diagonal in real space or in Fourier space, as was the case in Ref. [33], break down for covariance matrices with coherence lengths much larger than the box size.

A “Toeplitz” matrix is any matrix with the same number for every entry along its main diagonal and with every other diagonal similarly constant [67]. In general, a Toeplitz matrix is uniquely defined by the entries in its first row and its first column: if $i \geq j$ then $T_{ij} = T_{1+i-j,1}$ and if $i \leq j$ then $T_{ij} = T_{1,1-i+j}$. If the first row of a matrix is repeated with a cyclic permutation by one to the right in

each successive row, then it is a special kind of Toeplitz matrix called a ‘‘circulant’’ matrix. Circulant matrices are diagonalized by the discrete Fourier transform [67]. Given a circulant matrix \mathbf{C} with first column \mathbf{c} , the product of \mathbf{C} and some arbitrary vector \mathbf{v} can be computed in $\mathcal{O}(N \log N)$ time because

$$\mathbf{C} \mathbf{v} = \mathbf{F}^\dagger \text{diag}(\mathbf{F}\mathbf{c})\mathbf{F}\mathbf{v}, \quad (\text{A1})$$

where \mathbf{F} is the unitary, discrete Fourier transform matrix [67]. Reading Eq. (A1) from right to left, we see that every matrix operation necessary for this multiplication can be performed in $\mathcal{O}(N \log N)$ time or better.

Conveniently, any symmetric Toeplitz matrix can be embedded in a circulant matrix twice its size. Given a symmetric Toeplitz matrix \mathbf{T} , we can define another symmetric Toeplitz matrix \mathbf{S} with an arbitrary constant along its main diagonal. If we specify that the rest of the first row (besides the first entry) is the reverse of the rest of the first row of \mathbf{T} (again ignoring the first entry), the fact that the matrix is Toeplitz and symmetric completely determines the other entries. For example,

$$\text{if } \mathbf{T} = \begin{pmatrix} 5 & 3 & 2 \\ 3 & 5 & 3 \\ 2 & 3 & 5 \end{pmatrix}, \quad \text{then } \mathbf{S} = \begin{pmatrix} 0 & 2 & 3 \\ 2 & 0 & 2 \\ 3 & 2 & 0 \end{pmatrix}. \quad (\text{A2})$$

It is straightforward to verify that the matrix \mathbf{C} , defined as

$$\mathbf{C} \equiv \begin{pmatrix} \mathbf{T} & \mathbf{S} \\ \mathbf{S} & \mathbf{T} \end{pmatrix}, \quad (\text{A3})$$

is a circulant matrix. We can now multiply \mathbf{C} by a zero-padded vector so as to yield the product of the Toeplitz matrix and the original vector, $\mathbf{T}\mathbf{x}$, that we are looking for

$$\mathbf{C} \begin{pmatrix} \mathbf{v} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{T}\mathbf{v} \\ \mathbf{S}\mathbf{v} \end{pmatrix}. \quad (\text{A4})$$

Therefore, we can multiply any Toeplitz matrix by a vector in $\mathcal{O}(N \log N)$.

APPENDIX B: NOISE COVARIANCE MATRIX DERIVATION

In this appendix, we derive the the form of \mathbf{N} , the noise covariance matrix, in Eq. (57) by combining the form of $P^N(\mathbf{k}, \lambda)$, the noise power spectrum, in Eq. (32) with Eq. (33), which relates \mathbf{N} , to $P^N(\mathbf{k}, \lambda)$. To accomplish this, we simplify $P^N(\mathbf{k}, \lambda)$ into a form that is more directly connected to our data cube. We then approximate the integrals in Eq. (33) by assuming that the uv coverage is piecewise constant in cells corresponding to our Fourier space grid.¹³

¹³We also assume that measurements in nearby uv cells are uncorrelated, which may not be true if the baselines are not coplanar; instead \mathbf{N} , would have to be modeled as sparse rather than diagonal in angular Fourier space.

To simplify $P^N(\mathbf{k}, \lambda)$, we first note that because the term $B(\mathbf{k}, \lambda)$ in Eq. (32) represents the synthesized beam and is normalized to peak at unity, we can reinterpret the factor of $(f^{\text{cover}})^{-1} B^{-2}(\mathbf{k}, \lambda)$ as an inverted and normalized uv coverage. When $f^{\text{cover}} = 1$, the array has uniform coverage. We want to replace the factor $(f^{\text{cover}})^{-1} B^{-2}(\mathbf{k}, \lambda)$ with a quantity directly tied our choice of pixelization of the uv plane and written in terms of the simplest observational specification: the total time that baselines spend observing a particular uv cell, $t(\mathbf{k}, \lambda)$. We already know that the noise power is inversely proportional to that time because more time yields more independent samples.

To relate $t^{-1}(\mathbf{k}, \lambda)$ to $(f^{\text{cover}})^{-1} B^{-2}(\mathbf{k}, \lambda)$, we want to make sure that the formula yields the same answer for peak density in the case of a complete coverage. In other words, we want to find the constant t_{max} such that

$$\frac{t_{\text{max}}}{t(\mathbf{k}, \lambda)} = (f^{\text{cover}})^{-1} B^{-2}(\mathbf{k}, \lambda). \quad (\text{B1})$$

The time spent in the most observed cell is related to the size of the cell in the uv plane, the density of baselines in that cell, and the total integration time of the observation, τ . The cell size is determined by the pixelization of our data cube. We have divided each slice of our data cube of size $L_x \times L_y$ into $n_x \times n_y$ pixels. In Fourier space, this corresponds to a pixel size of

$$\Delta k_x = \frac{2\pi}{L_x} = \frac{2\pi}{d_M \Delta \theta_x n_x}, \quad (\text{B2})$$

where $\Delta \theta_x$ and is the angular pixelization in the x direction. An equivalent relation is true for the y direction. Since $\Delta u = \Delta k_x d_M / (2\pi)$, we have that the area in uv space of each of our grid points is

$$\Delta v \Delta u = \frac{1}{\Delta \theta_x n_x} \frac{1}{\Delta \theta_y n_y} = \frac{1}{\Omega_{\text{pix}} n_x n_y}. \quad (\text{B3})$$

The maximum density of baselines is the density of the autocorrelations,¹⁴ which is

$$n_{\text{max}} = N_{\text{ant}} \left(\frac{A_{\text{ant}}}{\lambda^2} \right)^{-1}, \quad (\text{B4})$$

where the quantity $(A_{\text{ant}}/\lambda^2)$ is the area in the uv plane associated with a single baseline [5]. We thus have that

$$t_{\text{max}} \equiv n_{\text{max}} \Delta u \Delta v \tau = \frac{N_{\text{ant}} \lambda^2 \tau}{A_{\text{ant}} \Omega_{\text{pix}} n_x n_y}. \quad (\text{B5})$$

Now we can substitute Eq. (B1) into Eq. (32) to get a more useful form of $P^N(\mathbf{k}, \lambda)$,

¹⁴If the use of autocorrelations (which most observations throw out, due to their unfavorable noise properties) is troubling, then it is helpful to recall that for a large and fully filled array, the uv density of the shortest baselines is approximately the same as the uv density of the autocorrelations.

$$P^N(\mathbf{k}, \lambda) = \frac{\lambda^4 T_{\text{sys}}^2 y d_M^2}{A_{\text{ant}}^2 \Omega_{\text{pix}} n_x n_y} \frac{1}{t(\mathbf{k}_\perp, \lambda)}. \quad (\text{B6})$$

In general, $t(\mathbf{k}_\perp, \lambda)$ depends in a nontrivial way on the array layout. As such, the integral expression for \mathbf{N} , in Eq. (33) with this form of $P^N(\mathbf{k}, \lambda)$ is only analytically tractable along the line of sight. Integrating k_z , we get that

$$N_{ij} = \frac{\delta_{z_i z_j}}{\Delta z} \frac{\lambda^4 T_{\text{sys}}^2 y d_M^2}{A_{\text{ant}}^2 \Omega_{\text{pix}} n_x n_y} \int j_0^2(k_x \Delta x / 2) j_0^2(k_y \Delta y / 2) \times e^{ik_x(x_i - x_j) + ik_y(y_i - y_j)} \frac{1}{t(\mathbf{k}_\perp, \lambda_i)} \frac{dk_x dk_y}{(2\pi)^2}. \quad (\text{B7})$$

We note that \mathbf{N} , is uncorrelated between frequency channels, as we would expect.

Along the other two dimensions, we will approach the problem by approximating the integrand as piecewise constant in Fourier cells, turning the integral into a sum and the dk into a Δk . We will use the index l to run over all Fourier modes perpendicular to the line of sight. Using the fact that the line of sight voxel length $\Delta z = y \Delta \nu$ and that $L_x L_y = \Omega_{\text{pix}} d_M^2 n_x n_y$, we have that

$$N_{ij} = \frac{\lambda^4 T_{\text{sys}}^2 \delta_{z_i z_j}}{A_{\text{ant}}^2 (\Omega_{\text{pix}} n_x n_y)^2 \Delta \nu} \times \sum_l^{\text{all } x \& y} \left[j_0^2(k_{x,l} \Delta x / 2) j_0^2(k_{y,l} \Delta y / 2) e^{ik_{x,l}(x_i - x_j)} \times e^{ik_{y,l}(y_i - y_j)} \frac{1}{t_l(\lambda_i)} \right]. \quad (\text{B8})$$

Next, we can turn this form into one that is more clearly computationally easy to multiply by a vector by introducing another Kronecker delta,

$$N_{ij} = \frac{\lambda^4 T_{\text{sys}}^2 \delta_{z_i z_j}}{A_{\text{ant}}^2 (\Omega_{\text{pix}} n_x n_y)^2 \Delta \nu} \sum_l^{\text{all } x \& y} e^{ik_{x,l} x_i} e^{ik_{y,l} y_i} \times \sum_m^{\text{all } x \& y} \left[j_0^2(k_{x,l} \Delta x / 2) j_0^2(k_{y,m} \Delta y / 2) e^{-ik_{x,m} x_j} \times e^{-ik_{y,m} y_j} \frac{\delta_{lm}}{t_l(\lambda_i)} \right]. \quad (\text{B9})$$

Finally, if we extend l and m to index over all frequency channels and all Fourier modes perpendicular to the line of sight, we can write down the noise covariance matrix as $\mathbf{N} = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}} \mathbf{F}_\perp$ where \mathbf{F}_\perp and \mathbf{F}_\perp^\dagger are the discrete, unitary two-dimensional Fourier and inverse Fourier transforms and where $\tilde{\mathbf{N}}$ can be written as

$$\tilde{N}_{lm} = \frac{\lambda^4 T_{\text{sys}}^2 j_0^2(k_{x,l} \Delta x / 2) j_0^2(k_{y,m} \Delta y / 2)}{A_{\text{ant}}^2 (\Omega_{\text{pix}})^2 n_x n_y \Delta \nu} \frac{\delta_{lm}}{t_l}. \quad (\text{B10})$$

The result, therefore, is a matrix that can be multiplied by a vector in $\mathcal{O}(N \log N)$.

APPENDIX C: CONSTRUCTION OF THE PRECONDITIONER

In this final appendix, we show how to construct the preconditioner that we use to speed up the conjugate gradient method for multiplying \mathbf{C}^{-1} by a vector. We devise our preconditioner by looking at \mathbf{C} piece by piece, building up pairs of matrices that make our covariances look more like the identity matrix. We start with $\mathbf{C} = \mathbf{N}$, generalize to $\mathbf{C} = \mathbf{U} + \mathbf{N}$, and then finally incorporate \mathbf{R} and \mathbf{G} to build the full preconditioner.

1. Constructing a preconditioner for \mathbf{N}

Our first task is to find a pair of preconditioning matrices that turn \mathbf{N} into the identity,

$$\mathbf{P}_N \mathbf{N} \mathbf{P}_N^\dagger = \mathbf{I}. \quad (\text{C1})$$

Because $\mathbf{N} = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}} \mathbf{F}_\perp$, and because $\tilde{\mathbf{N}}$ is a diagonal matrix, we define \mathbf{P}_N and \mathbf{P}_N^\dagger as follows:

$$\mathbf{P}_N = \tilde{\mathbf{N}}^{-1/2} \mathbf{F}_\perp, \quad \mathbf{P}_N^\dagger = \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}}^{-1/2}. \quad (\text{C2})$$

Since applying \mathbf{P}_N only requires multiplying by the inverse square root of a diagonal matrix and Fourier transforming in two dimensions, the complexity of applying \mathbf{P}_N to a vector is less than $\mathcal{O}(N \log N)$.

2. Constructing a preconditioner for \mathbf{U}

The matrix \mathbf{U} [Eq. (55)] can be written as the tensor product of three Toeplitz matrices, one for each dimension, bookended by two diagonal matrices, \mathbf{D}_U . Furthermore, since \mathbf{D}_U depends only on frequency (as we saw in Sec. III D 2), its effect can be folded into \mathbf{U}_z such that

$$\mathbf{D}_U [\mathbf{U}_x \otimes \mathbf{U}_y \otimes \mathbf{U}_z] \mathbf{D}_U \equiv \mathbf{U}_x \otimes \mathbf{U}_y \otimes \mathbf{U}_z'. \quad (\text{C3})$$

It is generally the case that \mathbf{U}_x and \mathbf{U}_y are both well approximated by the identity matrix. This reflects the fact that the spatial clustering of unresolved point sources is comparable with the angular resolution of the instrument. This assumption turns out to be quite good for fairly compact arrays, since for an array with 1 km as its longest baseline—the sort of compact array thought to be optimal for 21 cm cosmology—we expect an angular resolution on the order of 10 arcminutes, which is comparable to the fiducial value of 7 arcminutes that LT took to describe the clustering length scale for unresolved point sources. That value appears to be fairly reasonable given the results of Refs. [68,69]. For the purposes of devising a preconditioner only, we can therefore adopt the simplification that

$$\mathbf{U} \approx \mathbf{I}_x \otimes \mathbf{I}_y \otimes \mathbf{U}_z, \quad (\text{C4})$$

where we have dropped the prime for notational simplicity. Looking back at Fig. 4, this form of \mathbf{U} neatly explains the stair-stepping behavior of the eigenvalues: for every eigenvalue of \mathbf{U}_z , \mathbf{U} has $n_x \times n_y$ similar eigenvalues.

Since only a few eigenvalues of \mathbf{U}_z are large, it is pedagogically useful to first address a simplified version of the preconditioning problem where \mathbf{U}_z is approximated as a rank 1 matrix by cutting off its spectral decomposition after the first eigenvalue. We will later return to include the other relevant eigenvalues. We therefore write \mathbf{U} as follows:

$$\mathbf{U} \approx \mathbf{I}_x \otimes \mathbf{I}_y \otimes \lambda \mathbf{v}_z \mathbf{v}_z^\dagger, \quad (\text{C5})$$

where \mathbf{v}_z is the normalized eigenvector of \mathbf{U} .

Let us now take a look at the action of \mathbf{P}_N and \mathbf{P}_N^\dagger on $\mathbf{U} + \mathbf{N}$:

$$\begin{aligned} \mathbf{P}_N(\mathbf{U} + \mathbf{N})\mathbf{P}_N^\dagger &= \mathbf{I} + \tilde{\mathbf{N}}^{-1/2} \mathbf{F}_\perp (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \lambda \mathbf{v}_z \mathbf{v}_z^\dagger) \mathbf{F}_\perp^\dagger \tilde{\mathbf{N}}^{-1/2} \\ &= \mathbf{I} + \tilde{\mathbf{N}}^{-1/2} (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \lambda \mathbf{v}_z \mathbf{v}_z^\dagger) \tilde{\mathbf{N}}^{-1/2} \\ &\equiv \mathbf{I} + \tilde{\mathbf{U}}. \end{aligned} \quad (\text{C6})$$

Our next goal, therefore, is to come up with a new matrix \mathbf{P}_U that, when applied to $\mathbf{I} + \tilde{\mathbf{U}}$ gives us something close to \mathbf{I} .

We now take a closer look at $\tilde{\mathbf{U}}$. Since it is a good approximation to say that $\tilde{\mathbf{N}}$ only changes perpendicular to the line of sight,¹⁵ we can rewrite $\tilde{\mathbf{U}}$:

$$\begin{aligned} \tilde{\mathbf{U}} &\approx (\tilde{\mathbf{N}}_\perp^{-1/2} \otimes \mathbf{I}_z) (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \lambda \mathbf{v}_z \mathbf{v}_z^\dagger) (\tilde{\mathbf{N}}_\perp^{-1/2} \otimes \mathbf{I}_z) \\ &= (\tilde{\mathbf{N}}_\perp^{-1}) \otimes (\lambda \mathbf{v}_z \mathbf{v}_z^\dagger), \end{aligned} \quad (\text{C7})$$

where $\tilde{\mathbf{N}}_\perp$ is still a diagonal matrix, though only in two dimensions, generated from a baseline distribution averaged over frequency slices. We now form a pair of preconditioning matrices, \mathbf{P}_U and \mathbf{P}_U^\dagger of the form $\mathbf{P}_U \equiv \mathbf{I} - \beta \mathbf{\Pi}$ where $\mathbf{\Pi}$ has the property that $\mathbf{\Pi} \tilde{\mathbf{U}} = \tilde{\mathbf{U}}$ and that $\tilde{\mathbf{U}} \mathbf{\Pi}^\dagger = \tilde{\mathbf{U}}$. The matrix that fits this description is

$$\begin{aligned} \mathbf{\Pi} &= \tilde{\mathbf{N}}^{-1/2} (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \mathbf{v}_z \mathbf{v}_z^\dagger) \tilde{\mathbf{N}}^{1/2} \\ &\approx (\mathbf{I}_x \otimes \mathbf{I}_y \otimes \mathbf{v}_z \mathbf{v}_z^\dagger) = \frac{1}{\lambda} \mathbf{U}, \end{aligned} \quad (\text{C8})$$

since $\tilde{\mathbf{N}}$ only affects the x and y components and thus passes through the inner matrix. This also means that $\mathbf{\Pi} = \mathbf{\Pi}^\dagger$ and that $\mathbf{\Pi} = \mathbf{\Pi}^2$. The result for $\mathbf{P}_U(\mathbf{I} + \tilde{\mathbf{U}})\mathbf{P}_U^\dagger$ is

$$\begin{aligned} &(\mathbf{I} - \beta \mathbf{\Pi})(\mathbf{I} + \tilde{\mathbf{U}})(\mathbf{I} - \beta \mathbf{\Pi}^\dagger) \\ &= \mathbf{I} + \tilde{\mathbf{U}} - 2\beta \tilde{\mathbf{U}} - 2\beta \mathbf{\Pi} + \beta^2 \tilde{\mathbf{U}} + \beta^2 \mathbf{\Pi}. \end{aligned} \quad (\text{C9})$$

The trick now is that for each uv cell, $\tilde{\mathbf{U}}$ has only one eigenvalue, which we call $\bar{\lambda}_l$ (again using l as an index over both directions perpendicular to the line of sight),

$$\bar{\lambda}_l = \frac{\lambda}{(\tilde{\mathbf{N}}_\perp)_{ll}}. \quad (\text{C10})$$

Likewise, $\mathbf{\Pi}$ only has one eigenvalue, 1. By design, these eigenvalues correspond to the same eigenvector. Since our

¹⁵Were it not for the breathing of the synthesized beam with frequency, $\tilde{\mathbf{N}}$, would only change perpendicularly to the line of sight. Since it is a small effect when considered over a modest redshift range, we can ignore it in the construction of our preconditioner. After all, we only need to make \mathbf{PCP}^\dagger close to \mathbf{I} .

goal is to have the matrix in equation (C9) be the identity, we need only pick β such that

$$1 = 1 + \bar{\lambda}_l - 2\beta \bar{\lambda}_l - 2\beta + \beta^2 \bar{\lambda}_l + \beta^2. \quad (\text{C11})$$

Solving the quadratic equation, we get

$$\mathbf{P}_U \equiv \mathbf{I} - \left[\sum_l \left(1 - \sqrt{\frac{1}{1 + \bar{\lambda}_l}} \right) \mathbf{v}_z \mathbf{v}_z^\dagger \otimes \overleftrightarrow{\delta}_{x,x_l} \otimes \overleftrightarrow{\delta}_{y,y_l} \right], \quad (\text{C12})$$

where the pair of $\overleftrightarrow{\delta}$ matrices pick out a particular uv cell. If we want to generalize to more eigenvectors of \mathbf{U}_z , we simply need to keep subtracting off sums of matrices on the right-hand side of Eq. (C12),

$$\mathbf{P}_U \equiv \mathbf{I} - \sum_k \left[\sum_l \left(1 - \sqrt{\frac{1}{1 + \bar{\lambda}_{l,k}}} \right) \mathbf{v}_{z,k} \mathbf{v}_{z,k}^\dagger \otimes \overleftrightarrow{\delta}_{x,x_l} \otimes \overleftrightarrow{\delta}_{y,y_l} \right]. \quad (\text{C13})$$

This works because every set of vectors corresponding to a value of k is orthogonal to every other set. Each term in the above sum acts on a different subspace of \mathbf{C} independent of all the other terms in the sum.

If the relevant vectors $\mathbf{v}_{z,k}$ are precomputed, applying \mathbf{P}_U can be done in $\mathcal{O}(Nm(\mathbf{U}_z))$ where $m(\mathbf{U}_z)$ is defined as the number of relevant eigenvalues of \mathbf{U}_z that need preconditioning or, equivalently, the number of ‘‘steps’’ in the eigenvalues of \mathbf{U} in Fig. 4 above the noise floor. We examine how $m(\mathbf{U}_z)$ scales with the size of the data cube in Sec. III E 2. Because the fall off of the eigenvalues is exponential [70], we expect the scaling of m to be logarithmic.

In general, we can pick some threshold $\theta \geq 1$ to compare to the largest value of $\bar{\lambda}_{l,k}$ for a given k and then do not precondition modes with eigenvalues smaller than θ . One might expect there to be diminishing marginal utility to preconditioning the eigenvalues nearest 1. We explore how to optimally cut off the spectral decomposition in Sec. III E 3 by searching for a value of θ where the costs and benefits of preconditioning equalize.

3. Constructing a preconditioner for \mathbf{R} and \mathbf{G}

We now turn our attention to the full matrix \mathbf{C} . The fundamental challenge to preconditioning all of the matrices in \mathbf{C} simultaneously is that the components of \mathbf{R} and \mathbf{G} perpendicular to the line of sight are diagonalized in completely different bases. However, \mathbf{U} , \mathbf{G} , and \mathbf{R} have very similar components parallel to the line of sight, due to the fact they all represent spectrally smooth radiation of astrophysical origin.

We can write down \mathbf{R} as follows:

$$\mathbf{R} = \sum_n \left[\overleftrightarrow{\delta}_{x,x_n} \otimes \overleftrightarrow{\delta}_{y,y_n} \otimes \left(\sum_k \lambda_{n,k} \mathbf{v}_{z,n,k} \mathbf{v}_{z,n,k}^\dagger \right) \right], \quad (\text{C14})$$

which can be interpreted as a set of matrices describing spectral coherence, each localized to one point source, and all of which are spatially uncorrelated. And likewise, we can write down \mathbf{G} as

$$\mathbf{G} = \sum_{i,j,k} [\lambda_{x_i} \lambda_{y_j} \lambda_{z_k} \mathbf{v}_{x_i} \mathbf{v}_{x_i}^\dagger \otimes \mathbf{v}_{y_j} \mathbf{v}_{y_j}^\dagger \otimes \mathbf{v}_{z_k} \mathbf{v}_{z_k}^\dagger]. \quad (\text{C15})$$

We now make two key approximations for the purposes of preconditioning. First, we assume that all the z_k eigenvectors are the same, so $\mathbf{v}_{z_k} \approx \mathbf{v}_{z_{n,k}}$ for all n , all of which are also taken to be the same as the eigenvectors that appear in the preconditioner for \mathbf{U} in Eq. (C13). Second, as in Sec. C 2, we are only interested in acting upon the largest eigenvalues of \mathbf{R} and \mathbf{G} . To this end, we will ultimately only consider the largest values of $\lambda_{n,k}$ and $\lambda_{i,j,k} \equiv \lambda_{x_i} \lambda_{y_j} \lambda_{z_k}$, which will vastly reduce the computational complexity of the preconditioner.

Our strategy for overcoming the difficulty of the different bases is to simply add the two perpendicular parts of the matrices and then decompose the sum into its eigenvalues and eigenvectors. We therefore define

$$\mathbf{\Gamma} \equiv \mathbf{R} + \mathbf{G} \quad (\text{C16})$$

(choosing the symbol $\mathbf{\Gamma}$ because it looks like \mathbf{R} and sounds like \mathbf{G}). Given the above approximations, we can reexpress $\mathbf{\Gamma}$ as follows:

$$\mathbf{\Gamma} \approx \sum_k (\mathbf{\Gamma}_{\perp,k} \otimes \mathbf{v}_{z_k} \mathbf{v}_{z_k}^\dagger), \quad (\text{C17})$$

where we have defined each $\mathbf{\Gamma}_{\perp,k}$ as

$$\mathbf{\Gamma}_{\perp,k} \equiv \left(\sum_n \lambda_{n,k} \vec{\delta}_{x,x_n} \otimes \vec{\delta}_{y,y_n} \right) + \left(\sum_{i,j} \lambda_{i,j,k} \mathbf{v}_{x_i} \mathbf{v}_{x_i}^\dagger \otimes \mathbf{v}_{y_j} \mathbf{v}_{y_j}^\dagger \right). \quad (\text{C18})$$

Due to the high spectral coherence of the foregrounds, only a few values of k need to be included to precondition for $\mathbf{\Gamma}$. Considering the limit on angular box size imposed by the flat sky approximation and the limit on angular resolution imposed by the array size, this should require at most a few eigenvalue determinations of matrices no bigger than about 10^4 entries on a side. Moreover, those eigenvalue decompositions need only be computed once and then only partially stored for future use. In practice, this is not a rate-limiting step, as we see in Sec. III E 2.

We now write down the eigenvalue decomposition of $\mathbf{\Gamma}$:

$$\mathbf{\Gamma} = \sum_k \left(\sum_l \lambda_{l,k} \mathbf{v}_{\perp,l} \mathbf{v}_{\perp,l}^\dagger \right) \mathbf{v}_{z_k} \mathbf{v}_{z_k}^\dagger. \quad (\text{C19})$$

Before we attack the general case, we assume that only one value of $\lambda_{l,k}$ is worth preconditioning—we generalize to the full $\mathbf{P}_{\mathbf{\Gamma}}$ later. We now know that if we have a matrix that looks like $\mathbf{I} + \bar{\mathbf{U}}$ we can make it look like \mathbf{I} . So can we take $\mathbf{I} + \bar{\mathbf{U}} + \bar{\mathbf{\Gamma}}$, where $\bar{\mathbf{\Gamma}} \equiv \mathbf{P}_{\mathbf{N}} \mathbf{\Gamma} \mathbf{P}_{\mathbf{N}}^\dagger$, and turn it into $\mathbf{I} + \bar{\mathbf{U}}$? Looking at $\bar{\mathbf{\Gamma}}$,

$$\begin{aligned} \bar{\mathbf{\Gamma}} &= \tilde{\mathbf{N}}^{-1/2} \mathbf{F}_{\perp} \lambda_{\mathbf{\Gamma}} \mathbf{v} \mathbf{v}^\dagger \mathbf{F}_{\perp}^\dagger \tilde{\mathbf{N}}^{-1/2} \\ &= \lambda_{\mathbf{\Gamma}} (\tilde{\mathbf{N}}_{\perp}^{-1/2} \tilde{\mathbf{v}}_{\perp} \tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1/2}) \otimes \mathbf{v}_z \mathbf{v}_z^\dagger, \end{aligned} \quad (\text{C20})$$

where $\lambda_{\mathbf{\Gamma}}$ is the sole eigenvalue we are considering and where $\tilde{\mathbf{v}}_{\perp} \equiv \mathbf{F}_{\perp} \mathbf{v}_{\perp}$.

Again, we will look at a preconditioner of the $\mathbf{P}_{\mathbf{\Gamma}} = \mathbf{I} - \beta \mathbf{\Pi}$ where

$$\mathbf{\Pi} \equiv (\tilde{\mathbf{N}}_{\perp}^{-1/2} \tilde{\mathbf{v}}_{\perp} \tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1/2}) \otimes \mathbf{v}_z \mathbf{v}_z^\dagger. \quad (\text{C21})$$

This time, the $\tilde{\mathbf{N}}_{\perp}^{\pm 1/2}$ matrices do not pass through the eigenvectors to cancel one another out. We now exploit the spectral similarity of foregrounds and the fact that $\tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{v}}_{\perp} = \mathbf{v}_z^\dagger \mathbf{v}_z = 1$ to obtain

$$\mathbf{P}_{\mathbf{\Gamma}} \bar{\mathbf{U}} \mathbf{P}_{\mathbf{\Gamma}}^\dagger = \bar{\mathbf{U}} + \frac{\lambda_{\mathbf{U}}}{\lambda_{\mathbf{\Gamma}}} (\beta^2 - 2\beta) \bar{\mathbf{\Gamma}}. \quad (\text{C22})$$

This is very useful because it means that if we pick β properly, we can get the second term to cancel the $\bar{\mathbf{\Gamma}}$ terms we expect when we calculate the full effect of $\mathbf{P}_{\mathbf{\Gamma}}$ and $\mathbf{P}_{\mathbf{N}}$ on $\mathbf{N} + \mathbf{U} + \mathbf{\Gamma}$. Noting that the sole eigenvalue of $\bar{\mathbf{\Gamma}}$ is $\bar{\lambda}_{\mathbf{\Gamma}} \equiv \lambda_{\mathbf{\Gamma}} \tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1} \tilde{\mathbf{v}}_{\perp}$, we also define $\bar{\lambda}_{\mathbf{U}} \equiv \lambda_{\mathbf{U}} \tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1} \tilde{\mathbf{v}}_{\perp}$. Multiplying our preconditioner by our matrices, we see that the equality of the single eigenvalues yields another quadratic equation for β :

$$\begin{aligned} 1 + \bar{\lambda}_{\mathbf{U}} &= 1 - 2\beta + \beta^2 + (\beta^2 - 2\beta + 1) \bar{\lambda}_{\mathbf{\Gamma}} \\ &+ \bar{\lambda}_{\mathbf{\Gamma}} \frac{\lambda_{\mathbf{U}}}{\lambda_{\mathbf{\Gamma}}} (\beta^2 - 2\beta). \end{aligned} \quad (\text{C23})$$

Solving, we finally have our $\mathbf{P}_{\mathbf{\Gamma}}$ that acts on $\mathbf{I} + \bar{\mathbf{U}} + \bar{\mathbf{\Gamma}}$ and yields $\mathbf{I} + \bar{\mathbf{U}}$:

$$\mathbf{P}_{\mathbf{\Gamma}} = \mathbf{I} - \left(1 - \sqrt{\frac{1 + \bar{\lambda}_{\mathbf{U}}}{1 + \bar{\lambda}_{\mathbf{U}} + \bar{\lambda}_{\mathbf{\Gamma}}}} \right) [(\tilde{\mathbf{N}}_{\perp}^{-1/2} \tilde{\mathbf{v}}_{\perp} \tilde{\mathbf{v}}_{\perp}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1/2}) \otimes \mathbf{v}_z \mathbf{v}_z^\dagger]. \quad (\text{C24})$$

Finally, generalizing to multiple eigenvalues and taking advantage of the orthonormality of the eigenvectors, we have

$$\begin{aligned} \mathbf{P}_{\mathbf{\Gamma}} &= \mathbf{I} - \sum_{k,m} \left[\left(1 - \sqrt{\frac{1 + \bar{\lambda}_{\mathbf{U}_k}}{1 + \bar{\lambda}_{\mathbf{U}_k} + \bar{\lambda}_{\mathbf{\Gamma}_{k,m}}}} \right) \right. \\ &\left. \times ((\tilde{\mathbf{N}}_{\perp}^{-1/2} \tilde{\mathbf{v}}_{\perp_m} \tilde{\mathbf{v}}_{\perp_m}^\dagger \tilde{\mathbf{N}}_{\perp}^{-1/2}) \otimes \mathbf{v}_{z_k} \mathbf{v}_{z_k}^\dagger) \right]. \end{aligned} \quad (\text{C25})$$

The result of this somewhat complicated preconditioner is a reduction of the condition number of the matrix to be inverted by many orders of magnitude (see Fig. 5).

Lastly, we include Fourier transforms at the front and the back of the preconditioner so that the result, when multiplied by a real vector, returns a real vector. Therefore, the total preconditioner we use for \mathbf{C} is

$$\mathbf{F}_{\perp}^\dagger \mathbf{P}_{\mathbf{U}} \mathbf{P}_{\mathbf{\Gamma}} \mathbf{P}_{\mathbf{N}} (\mathbf{R} + \mathbf{U} + \mathbf{N} + \mathbf{G}) \mathbf{P}_{\mathbf{N}}^\dagger \mathbf{P}_{\mathbf{\Gamma}}^\dagger \mathbf{P}_{\mathbf{U}} \mathbf{F}_{\perp}. \quad (\text{C26})$$

- [1] R. Barkana and A. Loeb, *Phys. Rep.* **349**, 125 (2001).
- [2] S. R. Furlanetto, S. P. Oh, and F. H. Briggs, *Phys. Rep.* **433**, 181 (2006).
- [3] M. F. Morales and J. S. B. Wyithe, *Annu. Rev. Astron. Astrophys.* **48**, 127 (2010).
- [4] J. R. Pritchard and A. Loeb, *Rep. Prog. Phys.* **75**, 086901 (2012).
- [5] M. McQuinn, O. Zahn, M. Zaldarriaga, L. Hernquist, and S. R. Furlanetto, *Astrophys. J.* **653**, 815 (2006).
- [6] M. G. Santos and A. Cooray, *Phys. Rev. D* **74**, 083517 (2006).
- [7] J. D. Bowman, M. F. Morales, and J. N. Hewitt, *Astrophys. J.* **661**, 1 (2007).
- [8] S. R. Furlanetto, A. Lidz, A. Loeb, M. McQuinn, J. R. Pritchard, P. R. Shapiro, M. A. Alvarez, D. C. Backer, J. D. Bowman, J. O. Burns, C. L. Carilli, R. Cen, A. Cooray, N. Gnedin, L. J. Greenhill, Z. Haiman, J. N. Hewitt *et al.*, [arXiv:0902.3259](https://arxiv.org/abs/0902.3259).
- [9] J. S. B. Wyithe, A. Loeb, and P. M. Geil, *Mon. Not. R. Astron. Soc.* **383**, 1195 (2008).
- [10] T. Chang, U. Pen, J. B. Peterson, and P. McDonald, *Phys. Rev. Lett.* **100**, 091303 (2008).
- [11] P. Madau, A. Meiksin, and M. J. Rees, *Astrophys. J.* **475**, 429 (1997).
- [12] P. Tozzi, P. Madau, A. Meiksin, and M. J. Rees, *Astrophys. J.* **528**, 597 (2000).
- [13] P. Tozzi, P. Madau, A. Meiksin, and M. J. Rees, *Nucl. Phys. B, Proc. Suppl.* **80**, C509+ (2000).
- [14] I. T. Iliev, P. R. Shapiro, A. Ferrara, and H. Martel, *Astrophys. J. Lett.* **572**, L123 (2002).
- [15] S. R. Furlanetto, A. Sokasian, and L. Hernquist, *Mon. Not. R. Astron. Soc.* **347**, 187 (2004).
- [16] A. Loeb and M. Zaldarriaga, *Phys. Rev. Lett.* **92**, 211301 (2004).
- [17] S. R. Furlanetto, M. Zaldarriaga, and L. Hernquist, *Astrophys. J.* **613**, 16 (2004).
- [18] R. Barkana and A. Loeb, *Astrophys. J.* **626**, 1 (2005).
- [19] K. J. Mack and D. H. Wesley, [arXiv:0805.1531](https://arxiv.org/abs/0805.1531).
- [20] Y. Mao, M. Tegmark, M. McQuinn, M. Zaldarriaga, and O. Zahn, *Phys. Rev. D* **78**, 023529 (2008).
- [21] S. Clesse, L. Lopez-Honorez, C. Ringeval, H. Tashiro, and M. H. G. Tytgat, *Phys. Rev. D* **86**, 123506 (2012).
- [22] A. de Oliveira-Costa, M. Tegmark, B. M. Gaensler, J. Jonas, T. L. Landecker, and P. Reich, *Mon. Not. R. Astron. Soc.* **388**, 247 (2008).
- [23] G. Bernardi, D. A. Mitchell, S. M. Ord, L. J. Greenhill, B. Pindor, R. B. Wayth, and J. S. B. Wyithe, *Mon. Not. R. Astron. Soc.* **413**, 411 (2011).
- [24] A. Liu and M. Tegmark, *Phys. Rev. D* **83**, 103006 (2011).
- [25] M. A. Garrett, [arXiv:0909.3147](https://arxiv.org/abs/0909.3147).
- [26] G. Paciga, T.-C. Chang, Y. Gupta, R. Nityanada, J. Odegova, U.-L. Pen, J. B. Peterson, J. Roy, and K. Sigurdson, *Mon. Not. R. Astron. Soc.* **413**, 1174 (2011).
- [27] S. J. Tingay *et al.*, [arXiv:1206.6945](https://arxiv.org/abs/1206.6945).
- [28] A. R. Parsons, D. C. Backer, G. S. Foster, M. C. H. Wright, R. F. Bradley, N. E. Gugliucci, C. R. Parashare, E. E. Benoit, J. E. Aguirre, D. C. Jacobs, C. L. Carilli, D. Herne, M. J. Lynch, J. R. Manley, and D. J. Werthimer, *Astron. J.* **139**, 1468 (2010).
- [29] C. L. Williams *et al.*, *Astrophys. J.* **755**, 47 (2012).
- [30] L. J. Greenhill and G. Bernardi, [arXiv:1201.1700](https://arxiv.org/abs/1201.1700).
- [31] M. Tegmark and M. Zaldarriaga, *Phys. Rev. D* **82**, 103501 (2010).
- [32] C. L. Carilli and S. Rawlings, *New Astron. Rev.* **48**, 979 (2004).
- [33] U.-L. Pen, *Mon. Not. R. Astron. Soc.* **346**, 619 (2003).
- [34] P. M. Sutter, B. D. Wandelt, and S. S. Malu, *Astrophys. J. Suppl. Ser.* **202**, 9 (2012).
- [35] X. Wang, M. Tegmark, M. G. Santos, and L. Knox, *Astrophys. J.* **650**, 529 (2006).
- [36] L. Gleser, A. Nusser, and A. J. Benson, *Mon. Not. R. Astron. Soc.* **391**, 383 (2008).
- [37] J. D. Bowman, M. F. Morales, and J. N. Hewitt, *Astrophys. J.* **695**, 183 (2009).
- [38] A. Liu, M. Tegmark, and M. Zaldarriaga, *Mon. Not. R. Astron. Soc.* **394**, 1575 (2009).
- [39] V. Jelić, S. Zaroubi, P. Labropoulos, R. M. Thomas, G. Bernardi, M. A. Brentjens, A. G. de Bruyn, B. Ciardi, G. Harker, L. V. E. Koopmans, V. N. Pandey, J. Schaye, and S. Yatawatta, *Mon. Not. R. Astron. Soc.* **389**, 1319 (2008).
- [40] G. Harker, S. Zaroubi, G. Bernardi, M. A. Brentjens, A. G. de Bruyn, B. Ciardi, V. Jelić, L. V. E. Koopmans, P. Labropoulos, G. Mellema, A. Offringa, V. N. Pandey, J. Schaye, R. M. Thomas, and S. Yatawatta, *Mon. Not. R. Astron. Soc.* **397**, 1138 (2009).
- [41] A. Liu, M. Tegmark, J. Bowman, J. Hewitt, and M. Zaldarriaga, *Mon. Not. R. Astron. Soc.* **398**, 401 (2009).
- [42] G. Harker, S. Zaroubi, G. Bernardi, M. A. Brentjens, A. G. de Bruyn, B. Ciardi, V. Jelić, L. V. E. Koopmans, P. Labropoulos, G. Mellema, A. Offringa, V. N. Pandey, A. H. Pawlik, J. Schaye, R. M. Thomas, and S. Yatawatta, *Mon. Not. R. Astron. Soc.* **405**, 2492 (2010).
- [43] J. Cho, A. Lazarian, and P. T. Timbie, *Astrophys. J.* **749**, 164 (2012).
- [44] M. McQuinn, L. Hernquist, M. Zaldarriaga, and S. Dutta, *Mon. Not. R. Astron. Soc.* **381**, 75 (2007).
- [45] M. Tegmark, A. J. S. Hamilton, M. A. Strauss, M. S. Vogeley, and A. S. Szalay, *Astrophys. J.* **499**, 555 (1998).
- [46] R. Barkana and A. Loeb, *Astrophys. J. Lett.* **624**, L65 (2005).
- [47] Sk. S. Ali, S. Bharadwaj, and B. Pandey, *Mon. Not. R. Astron. Soc.* **363**, 251 (2005).
- [48] A. Nusser, *Mon. Not. R. Astron. Soc.* **364**, 743 (2005).
- [49] R. Barkana, *Mon. Not. R. Astron. Soc.* **372**, 259 (2006).
- [50] M. Tegmark, *Phys. Rev. D* **55**, 5895 (1997).
- [51] J. R. Bond, A. H. Jaffe, and L. Knox, *Phys. Rev. D* **57**, 2117 (1998).
- [52] R. A. Fisher, *J. R. Stat. Soc.* **98**, 39 (1935).
- [53] M. Tegmark, A. J. S. Hamilton, and Y. Xu, *Mon. Not. R. Astron. Soc.* **335**, 887 (2002).
- [54] T. Di Matteo, R. Perna, T. Abel, and M. J. Rees, *Astrophys. J.* **564**, 576 (2002).
- [55] M. Tegmark and M. Zaldarriaga, *Phys. Rev. D* **79**, 083530 (2009).
- [56] D. W. Hogg, [arXiv:astro-ph/9905116](https://arxiv.org/abs/astro-ph/9905116).
- [57] S. P. Oh, D. N. Spergel, and G. Hinshaw, *Astrophys. J.* **510**, 551 (1999).
- [58] N. Jarosik *et al.*, *Astrophys. J. Suppl. Ser.* **170**, 263 (2007).
- [59] M. R. Hestenes and E. Stiefel, *J. Res. Natl. Bur. Stand.* **49**, 409 (1952).
- [60] J. Shewchuk (unpublished).

- [61] A. Datta, J.D. Bowman, and C.L. Carilli, *Astrophys. J.* **724**, 526 (2010).
- [62] H. Vedantham, N.U. Shankar, and R. Subrahmanyam, *Astrophys. J.* **745**, 176 (2012).
- [63] M.F. Morales, B. Hazelton, I. Sullivan, and A. Beardsley, *Astrophys. J.* **752**, 137 (2012).
- [64] C.M. Trott, R.B. Wayth, and S.J. Tingay, *Astrophys. J.* **757**, 101 (2012).
- [65] T.A. Wood and G. Chan, *J. Comput. Graph. Stat.* **3**, 409 (1994).
- [66] A. Liu, M. Tegmark, S. Morrison, A. Lutomirski, and M. Zaldarriaga, *Mon. Not. R. Astron. Soc.* **408**, 1029 (2010).
- [67] R.M. Gray, *Foundations and Trends in Communications and Information Theory* (Princeton University, Princeton, NJ, 2006).
- [68] G. Bernardi, A.G. de Bruyn, M.A. Brentjens, B. Ciardi, G. Harker, V. Jelić, L.V.E. Koopmans, P. Labropoulos, A. Offringa, V.N. Pandey, J. Schaye, R.M. Thomas, S. Yatawatta, and S. Zaroubi, *Astron. Astrophys.* **500**, 965 (2009).
- [69] A. Ghosh, J. Prasad, S. Bharadwaj, S.S. Ali, and J.N. Chengalur, *Mon. Not. R. Astron. Soc.* **426**, 3295 (2012).
- [70] A. Liu and M. Tegmark, *Mon. Not. R. Astron. Soc.* **419**, 3491 (2012).