Cooling Improves Cosmic Microwave Background Map-making when Low-frequency Noise is Large

Bai-Chiang Chiang (強百強) and Kevin M. Huffenberger ©
Department of Physics, Florida State University, Tallahassee, Fl 32306, USA
Received 2021 September 29; accepted 2021 October 19; published 2021 November 24

Abstract

In the context of cosmic microwave background data analysis, we study the solution to the equation that transforms scanning data into a map. As originally suggested in “messenger” methods for solving linear systems, we split the noise covariance into uniform and nonuniform parts and adjust their relative weights during the iterative solution. With simulations, we study mock instrumental data with different noise properties, and find that this “cooling” or perturbative approach is particularly effective when there is significant low-frequency noise in the timstream. In such cases, a conjugate gradient algorithm applied to this modified system converges faster and to a higher fidelity solution than the standard conjugate gradient approach. We give an analytic estimate for the parameter that controls how gradually the linear system should change during the course of the solution.

Unified Astronomy Thesaurus concepts: Cosmic microwave background radiation (322); Astronomy data analysis (1858); Computational methods (1965)

1. Introduction

In observations of the cosmic microwave background (CMB), map-making is an intermediate step between the collection of raw scanning data and the scientific analyses, such as the estimation of power spectra and cosmological parameters. Next generation CMB observations will generate much more data than those today, and so it is worth exploring efficient ways to process the data even though, on paper, the map-making problem has long been solved.

The time-ordered scanning data is summarized by

\[ d = Pm + n, \]

where \( d, m, \) and \( n \) are the vectors of time-ordered data (TOD), the CMB sky-map signal, and measurement noise. \( P \) is a sparse matrix in the time-by-pixel domain that encodes the telescope’s pointing. Of several map-making methods (Tegmark 1997), one of the most common is the method introduced for the Cosmic Background Explorer (COBE; Janssen & Gulkis 1992). This optimal, linear solution is

\[ (P^TN^{-1}P)\hat{m} = P^TN^{-1}d, \]

where \( \hat{m} \) provides the standard generalized least squares minimization of the \( \chi^2 \) statistic,

\[ \chi^2(m) = (d - Pm)^T N^{-1}(d - Pm). \]

Here we assume that the noise has zero mean \( \langle n \rangle = 0 \), and the noise covariance matrix \( N = \langle nn^T \rangle \) is diagonal in frequency space. In the case where the noise is Gaussian, the COBE solution is also the maximum likelihood solution.

With current computational power, we cannot solve for \( \hat{m} \) by calculating \( (P^TN^{-1}P)^{-1}P^TN^{-1}d \) directly. The noise covariance matrix \( N \) is often sparse in the frequency domain and the pointing matrix \( P \) is sparse in the time-by-pixel domain. In experiments currently under design, there may be \( \sim 10^{16} \) time samples and \( \sim 10^9 \) pixels, so these matrix inversions are intractable unless the covariance is uniform (proportional to the identity matrix \( I \)). We can use iterative methods, such as conjugate gradient descent, to avoid the matrix inversions, and execute each matrix multiplication in a basis where the matrix is sparse, using a fast Fourier transform to go between the frequency and time domain.

As an alternative to conjugate gradient descent, Huffenberger & Næss (2018) showed that the “messenger” iterative method could be adapted to solve the linear map-making system, based on the approach from Elsner & Wandelt (2013) to solve the linear Wiener filter. This technique splits the noise covariance into a uniform part and the remainder, and introduces an additional vector that represents the signal plus uniform noise. This messenger field acts as an intermediary between the signal and the data and has a covariance that is conveniently sparse in every basis. Elsner & Wandelt (2013) also introduced a cooling scheme that takes advantage of the split covariance: over the course of the iterative solution, we adjust the relative weight of those two parts. Starting with the uniform covariance, the modified linear system gradually transforms to the final system, under the control of a cooling parameter. In numerical experiments, Huffenberger & Næss (2018) found that a map produced by the cooled messenger method converged significantly faster than for standard conjugate gradient methods, and to higher fidelity, especially on large scales.

Papež et al. (2018) showed that the messenger field approach is equivalent to a fixed-point iteration scheme, and studied its convergence properties in detail. Furthermore, they showed that the split covariance and the modified system that incorporates the cooling can be solved by other means, including a conjugate gradient technique, which should generally show better convergence properties than the fixed-point scheme. With simulations, we have confirmed this conclusion. In their numerical tests, Papež et al. (2018) did not find benefits to the cooling modification of the map-making
system, in contrast to the findings of Huffenberger & Næss (2018).

In this paper, we show that the difference arose because the numerical tests in Papež et al. (2018) used much less low-frequency (or 1/f) noise than Huffenberger & Næss (2018), and show that the cooling technique improves map-making performance especially when the low-frequency noise is large. This performance boost depends on a proper choice for the pace of cooling. Kodi Ramanah et al. (2017) showed that for the Wiener filter the cooling parameter should be chosen as a geometric series. In this work, we give an alternative interpretation of the parameterizing process and show that for map-making the optimal choice (unsurprisingly) is also a geometric series.

In Section 2 we describe our methods for treating the map-making equation and our numerical experiments. In Section 3 we present our results. In Section 4, we list our conclusions. In the Appendix we derive the prescription for our cooling schedule.

2. Methods

2.1. Parameterized Conjugate Gradient Method

The messenger field approach introduced an extra cooling parameter A to the map-making equation, and solved the linear system with an alternative parameterized covariance \( N(\lambda) = \lambda I + \bar{N} \). The parameter \( \tau = \min(\text{diag}(N)) \) represents the uniform level of (white) noise in the original covariance. The remainder \( \bar{N} \equiv N - \tau I \) is the nonuniform part of the original noise covariance. (Here \( N \) without any arguments denotes the original noise covariance matrix \( N = (m \eta^{-1}) \).) In this work we find it more convenient to work with the reciprocal of cooling parameter \( \eta = \lambda^{-1} \), which represents the degree of heteroscedasticity (nonuniformity) in the parameterized covariance

\[
N(\eta) = \tau I + \eta \bar{N}.
\]

When \( \eta = 1 \) this parameterized covariance \( N(\eta) \) equals \( N \).

Papež et al. (2018) showed that the conjugate gradient method can be easily applied to the cooled map-making problem. In our notation, this is equivalent to iterating on the parameterized map-making equation

\[
(P^\dagger N(\eta)^{-1} P) \hat{m}(\eta) = P^\dagger N(\eta)^{-1} d,
\]

as we adjust the parameter through a set of levels \{\eta\}. (We use \( \hat{m} \) with no \( \eta \) argument to mean the estimated \( \hat{m} \) in Equation (2), independent of \( \eta \).) This equation leads to the same system as the parameterized equation in the messenger field method, because \( N(\eta) = \lambda^{-1} N(\lambda) \) and the condition number do not change upon scalar multiplication to both sides of the equation. For concreteness we fix the preconditioner to \( M = P^\dagger P \) for all calculations.

When \( \eta = 0 \), the noise covariance matrix \( N(\eta) \) is homoscedastic (uniform), and the solution is given by the simple binned map \( \hat{m}(0) = (P^\dagger P)^{-1} P^\dagger d \), which can be solved directly.

Since the non-white part of \( \bar{N} \) is the troublesome portion of the covariance, we can think of the \( \eta \) parameter as increasing the heteroscedasticity of the system, adding a perturbation to the solution achieved at a particular stage, building ultimately upon the initial uniform covariance model. Therefore, this quasi-static process requires \( \eta \) increase as

\[
0 = \eta_0 \leq \eta_1 \leq \cdots \leq \eta_{\text{final}} = 1,
\]

at which point we arrive at the desired map-making equation, and the solution \( \hat{m}(1) = \hat{m} \).

We may iterate more than once at each intermediate \( \eta_i \); we solve Equation (5) with conjugate gradient iterations using the result from the previous calculation \( \hat{m}(\eta_{i-1}) \) as the initial value. We move to the next parameter \( \eta_{i+1} \) when the norm of the residual vector

\[
||r(m, \eta_i)|| = ||P^\dagger N(\eta_i)^{-1} P m - P^\dagger N(\eta_i)^{-1} d||
\]

is an order of magnitude smaller than the norm of the right-hand side of Equation (5).

\[
||r(m, \eta_i)|| < 0.1 \ ||P^\dagger N(\eta_i)^{-1} d||.
\]

This is not stringent enough to completely converge at this \( \eta \)-level, but we find that it causes the system to converge sufficiently to allow us to move on to the next \( \eta \).

2.2. Analytical Expression for \{\eta\} Series

The next question is how to appropriately choose these monotonically increasing parameters \( \eta \). We also want to determine \( \eta_1, \dotsc, \eta_{n-1} \) before starting conjugate gradient iterations, because the time-ordered data \( d \) is very large, and we do not want to keep it in the system memory during calculation or repeatedly read it in from the disk. If we determine \( \eta_1, \dotsc, \eta_{n-1} \) before the iterations, then we can precompute the right-hand side of Equation (5) for each \( \eta_i \) and keep these map-sized objects, instead of the entire time-ordered data.

In the Appendix, we show that a generic good choice for the \( \eta \) parameters is given by this geometric series

\[
\eta_i = \min\left\{ (2^i - 1) \frac{\tau}{\max(\bar{N}_i)}, 1 \right\},
\]

where \( \bar{N}_i \) are the eigenvalues of \( \bar{N} \) under frequency representation. This is one of our main results. It not only tells us how to choose parameters \( \eta_i \), but also when we should stop the perturbation, and set \( \eta = 1 \). For example, if the noise covariance matrix \( N \) is almost uniform, then \( \bar{N} = N - \tau I \approx 0 \), and we would have \( \tau / \max(\bar{N}_i) \approx 1 \). This tells us that we do not need to use the parameterized method at all, because \( \eta_0 = 0 \) and \( \eta_1 = \eta_2 = \cdots = 1 \). This corresponds to the standard conjugate gradient method with a simple binned map as the initial guess (as recommended by Papež et al. 2018).

2.3. Intuitive Interpretation of \( \eta \)

This section includes a way to interpret the role of \( \eta \) that is less technical than in the Appendix. Our ultimate goal is to find \( \hat{m}(1) \), which minimizes \( \chi^2(m) \) in Equation (3). Since \( N \) is diagonal in frequency space, \( \chi^2 \) could be written as a sum of all frequency modes \( ||(d - Pm)||^2 \) with weight \( N_i^{-1} \), such as

\[
\chi^2(m) = \sum_f ||(d - Pm)||^2 N_f^{-1}.
\]

The weight is large for low-noise frequency modes (small \( N_f \)) and small for high-noise modes, which means \( \chi^2(m) \) would favor the low-noise modes, and therefore the conjugate gradient map-making focuses on minimizing the error

\[
e^2 \equiv ||d - Pm|| \text{ in the low-noise part.}
\]

After introducing \( \eta \), we minimize \( \chi^2(m, \eta) \) in Equation (A1) instead. For \( \eta = 0 \), \( N^{-1}(0) \propto \lambda \) the system is homoscedastic and the estimated map \( \hat{m}(0) \) does not prioritize any frequency modes. As we slowly increase \( \eta \), we decrease the weight for the
high-noise modes, and focus on minimizing error for the low-noise part. If we start with $\eta_1 = 1$ directly, which corresponds to the vanilla conjugate gradient method, then the algorithm will focus mostly on minimizing the low-noise part, such that $\chi^2$ would converge very quickly on the low-noise modes (typically high temporal frequencies and small spatial scales), but slowly on the high-noise modes (low frequencies and large scales). However, by introducing the $\eta$ parameter, we let the solver first treat every frequency equally. Then as $\eta$ slowly increases, it gradually gives more focus on the lowest noise part.

2.4. Computational Cost

To properly compare the performance cost of this method with respect to the vanilla conjugate gradient method with the simple preconditioner, we need to compute their computational cost at each iteration. We could define $A(\eta) \equiv P^T N(\eta)^{-1} P$ and $b(\eta) \equiv P^T N(\eta)^{-1} d$, and Equation (5) could be written as $A(\eta) m(\eta) = b(\eta)$. The right-hand side $b(\eta)$ could be computed before iterating, since we have determined $\eta$ in advance, so it will not introduce extra computational cost. The most demanding part of the conjugate gradient method is calculating its left-hand side $A(\eta) m$, because it contains a Fourier transform of $P m$ from the time domain to frequency domain and an inverse Fourier transform of $N(\eta)^{-1} P m$ from the frequency domain back to time domain, which is order $O(n \log n)$ with $n$ being the length of time-ordered data. Compared to the traditional conjugate gradient method, we swap $N^{-1}$ with $N(\eta)^{-1}$, and the cost is the same for one step, since both methods need a fast Fourier transform and inverse fast Fourier transform at one iteration.

At each $\eta$ level, we use the residual to determine whether to switch to the next level ($\eta_{i+1}$), as in Equation (7). Calculation of the residual vector $r(m, \eta_i)$ is part of the conjugate gradient algorithm, so this will not add extra cost either. Therefore, overall introducing the $\eta$ will not have extra computational cost within the conjugate gradient iterations.

However, we start a new conjugate gradient algorithm whenever $\eta_i$ updates to $\eta_{i+1}$. Thus we must reinitialize the conjugate gradient algorithm, recalculating the residual $r(m, \eta_{i+1})$ based on new $\eta_{i+1}$. This residual calculation contains an extra $A(\eta_{i+1}) m$ operation. Therefore, if we have a series $\eta_i, \eta_2, \eta_3, \ldots, \eta_n$, there will have $n - 1$ extra $A(\eta) m$ operations compare to the traditional conjugate gradient method. If the total number of iterations is much larger than $n$, then this extra cost is negligible. For our simulation, this extra step would have a rather significant impact on the final result. To have a fair comparison between the parameterized and traditional conjugate gradient method, we will present our results with the number of $P^T N(\eta)^{-1} P m$ operations as the horizontal axis.

2.5. Numerical Simulations

To compare these algorithms, we need to do some simple simulations of scanning processes, and generate the time-ordered data from a random sky signal.¹ Our sky is a small rectangular area, with two orthogonal directions $x$ and $y$, both with ranges from $-1^\circ$ to $+1^\circ$. The signal has Stokes parameters $(I, Q, U)$ for intensity and linear polarization.

1 The source code and other information are available at https://github.com/Bai-Chiang/CMB_map_making_with_cooling.

Figure 1. Noise power spectra that we use in our map-making simulations. These show a variety of low-frequency behavior, parameterized by Equation (9), with white noise at high frequency and a low-frequency power-law slope $\alpha = 3$. Here we show two knee frequencies, $f_{knee} = 10$ Hz (solid lines) and $f_{knee} = 0.1$ Hz (dashed lines). For each knee frequency, we have shown an unflattened spectrum ($f_{apo} = 0$ Hz), and two flattened ones ($f_{apo} = 0.1 f_{knee}$ and 0.01$f_{knee}$). The vertical line shows our scanning frequency.

For the scanning process, our mock telescope contains nine detectors, each with different sensitivity to polarization $Q$ and $U$. It scans the sky with a raster scanning pattern. Its scanning frequency is $f_{scan} = 0.1$ Hz and sampling frequency is $f_{sample} = 100$ Hz. The telescope scans the sky horizontally then vertically. This gives the noiseless signal $s$. The sky signal in the timesream has an rms of 56 $\mu$K. The signal is continuous, so that it has structure on subpixel scales, but we find that our main conclusions remain the same when the input signal is pixelized. In map-making, we digitize the position $(x, y)$ into $512 \times 512$ pixels.

We model the noise power spectrum with

$$P(f) = \sigma^2 \left(1 + \frac{f_{knee}^\alpha + f_{apo}^\alpha}{f^\alpha + f_{apo}^\alpha}\right)$$

(9)

which is white at high frequencies, a power law below the knee frequency, and gives us the option to flatten the low-frequency noise below an apodization frequency (like in Papež et al. 2018). Note that as $f_{apo} \to 0$, $P(f) \to \sigma^2(1 + (f/f_{knee})^{-\alpha})$, and it becomes a 1/$f$-type noise model.

Dünner et al. (2013) measured the slopes of the atmospheric noise in the Atacama under different water vapor conditions, finding $\alpha = 2.7$ to 2.9. Here we use $\sigma^2 = 10 \mu$K$^2$. $\alpha = 3$, and compare the performance under different noise models. In our calculations, we choose different combinations of $f_{knee}$ and $f_{apo}$ as in Figure 1. The noise spectra with the most low-frequency noise have high $f_{knee}$ or low cutoff $f_{apo}$.

The noise covariance matrix

$$N_{yy} = P(f) \frac{\delta y}{\Delta f}$$

(10)

is a diagonal matrix in frequency space, where $\Delta f$ is equal to the reciprocal of total scanning time $T \approx 1.05 \times 10^4$ s.
Finally, we get the simulated time-ordered data $d = s + n$ by adding up the signal and noise.

## 3. Results

We compare the standard conjugate gradient method versus the conjugate gradient with our perturbed linear system. Both methods use the simple preconditioner $P^{\top}P$. Figure 2 shows the $\chi^2$ results for $1/f$ noise models ($f_{apo} = 0$) with different knee frequencies. Note that the $\chi^2$ values in all figures are calculated based on the standard $\chi^2(m)$ in Equation (3), not the $\eta$-dependent $\chi^2(m, \eta)$ of the modified system (Equation (A1)). The minimum $\chi^2_{\text{min}}$ that we use for comparison is calculated from a deliberately slowed and well-converged parameterized conjugate gradient method: one with 100 $\eta$ values and that halts when the final norm of the residual $\|r(m, 1)\|$ is smaller than $10^{-5} \times \|P^NI^{-1}d\|$, or 100 iterations after $\eta = 1$. From Figure 2, we can see for the $1/f$ noise model, when $f_{\text{knee}} \lesssim 10 f_{\text{scan}}$ the parameterized method starts showing an advantage over the vanilla conjugate gradient method.

In Figure 3, we fixed $f_{\text{knee}} = 10$ Hz, and change $f_{\text{apo}}$ as we decrease $f_{\text{apo}}$ relative to $f_{\text{knee}}$, increasing the amount of low-frequency noise, the parameterized conjugate gradient method performs better.

Looking at the power spectrum in Figure 1, when $f_{\text{knee}}$ is small, or $f_{\text{apo}}$ is large, there is not much low-frequency noise. These situations correspond to the left-side plots in Figure 2 and Figure 3. The right-side graphs have a significant amount of low-frequency noise. We conclude that the introduction of the slowly varying $\eta$ parameter improves performance most when there are large low-frequency noise contributions.

We also tried different $1/f$ noise slopes $\alpha$. For $\alpha = 2$, the conclusion is the same as $\alpha = 3$. When $\alpha = 1$, the low-frequency noise is reduced compared to the cases with steeper slopes, and the vanilla conjugate gradient method is preferred, except some cases with very large knee frequency like...
The Astrophysical Journal, 922:97 (6pp), 2021 December 1

To simplify notation, we use $N_\eta$ to denote the parameterized covariance matrix $N(\eta) = \tau I + \eta N$. For some specific $\eta$ value, the estimated map $\hat{\mu}(\eta) = (P^\dagger N^{-1} P)^{-1} P^\dagger N^{-1} d$ minimizes

$$
\chi^2(m, \eta) = (d - P\hat{\mu}(\eta))^\dagger N^{-1}_\eta (d - P\hat{\mu}(\eta)).
$$

(A1)

with $\eta$ being fixed. We restrict to the case that the noise covariance matrix $N$ is diagonal in the frequency domain, and represent the frequency-domain eigenvalues as $N_f$.

The perturbative scheme works like this. We start with $\chi^2(\hat{\mu}(\eta_0), \eta_0)$ with $\hat{\mu}(\eta_0) = (P^\dagger P)^{-1} P^\dagger d$, which could be solved directly. Then we use the conjugate gradient method to find $\hat{\mu}(\eta_1)$ and its corresponding $\chi^2(\hat{\mu}(\eta_1), \eta_1)$. So let us consider $\eta_1 = \eta_0 + \delta\eta = \delta\eta$ such that $\eta_1 = \delta\eta$ is a very small quantity, $\delta\eta \ll 1$. (Remember $\eta_0 = 0$.) Since $\hat{\mu}(\eta)$ minimizes $\chi^2(m, \eta)$ with $\eta$ being fixed, we have $\frac{\partial}{\partial \eta} \chi^2(\hat{\mu}(\eta), \eta) = 0$, and using the chain rule

$$
\frac{d}{d\eta} \chi^2(\hat{\mu}(\eta), \eta) = \frac{\partial}{\partial \eta} \chi^2(\hat{\mu}(\eta), \eta) = - (d - P\hat{\mu}(\eta))^\dagger N^{-1}_\eta (d - P\hat{\mu}(\eta)).
$$

(A2)

Then the fractional decrease of $\chi^2(\hat{\mu}(\eta_0), \eta_0)$ from $\eta_0$ to $\eta_1 = \delta\eta$ is

$$
- \frac{\delta \chi^2(\hat{\mu}(\eta_0), \eta_0)}{\chi^2(\hat{\mu}(\eta_0), \eta_0)} = - \frac{\delta\eta}{\chi^2(\hat{\mu}(\eta_0), \eta_0)} \chi^2(\hat{\mu}(\eta_1), \eta_1) = \delta\eta \frac{1}{\tau} \frac{(d - P\hat{\mu}(\eta_0))^\dagger N (d - P\hat{\mu}(\eta_0))}{(d - P\hat{\mu}(\eta_0))^\dagger (d - P\hat{\mu}(\eta_0))}.
$$

(A3)

Here we put a minus sign in front of this expression such that it is nonnegative, and use $N_{\eta=0} = \tau I$ at the second equality. We want $|\delta \chi^2(\hat{\mu}(\eta_0), \eta_0)| = \chi^2(\hat{\mu}(\eta_0), \eta_0) - \chi^2(\hat{\mu}(\eta_1), \eta_1)$ to be large to encourage fast convergence. Therefore, $\chi^2(\hat{\mu}(\eta_1), \eta_1)$ is much smaller than $\chi^2(\hat{\mu}(\eta_0), \eta_0)$, or $\chi^2(\hat{\mu}(\eta_1), \eta_1) \ll \chi^2(\hat{\mu}(\eta_0), \eta_0)$. Then we would expect

$$
- \frac{\delta \chi^2(\hat{\mu}(0), 0)}{\chi^2(\hat{\mu}(0), 0)} = 1 - \frac{\chi^2(\hat{\mu}(\eta), \eta)}{\chi^2(\hat{\mu}(0), 0)} \approx 1.
$$

(A4)

Here we use the notation $1^-$, which means the upper bound is close to but strictly smaller than 1. Now we can use Equation (A3) and let it equal 1; thus $\delta\eta = - \chi^2(\hat{\mu}(\eta_0), \eta_0) \frac{d}{d\eta} \chi^2(\hat{\mu}(\eta_0), \eta_0)$.

Applying this same idea to $\eta_{m+1} = \eta_m + \delta\eta_m$ with $m \geq 1$, we would get

$$
\delta\eta_m = - \chi^2(\hat{\mu}(\eta_m), \eta_m) \frac{d}{d\eta} \chi^2(\hat{\mu}(\eta_m), \eta_m). 
$$

(A5)

As mentioned in the main text, we need to determine the entire series $\eta$ before conjugate gradient iterations. We do not have the $\hat{\mu}(\eta_m)$ with which to calculate them and need to find another approach.

Let us go back to Equation (A3). Since we cannot calculate $d - P\hat{\mu}(\eta_m)$ before making the map, we treat it as an arbitrary

4. Conclusions

We analyzed the parameterized conjugate gradient map-making method that is inspired by the messenger field idea of separating the white noise out of the noise covariance matrix. Then we gave an analytical expression for the series of $\eta$ parameters that govern how quickly the modified covariance adjusts to the correct covariance, and showed that this method adds only the extra computational cost of reinitializing the conjugate gradient process based on a new $\eta$ parameter.

We tested this method for different noise power spectra, both flattened and nonflattened at low frequency. The results showed that the parameterized method is faster than the traditional conjugate gradient method when there is a significant amount of low-frequency noise. It could be further improved if we could get a more accurate estimation for the change in $\chi^2$ as a function of the $\eta$ parameter, either before iteration or without using time-ordered data during iteration.

Also note that we fixed the preconditioner as $M = P^\dagger P$ during our calculation, this parameterizing process could be applied to any preconditioner and possibly improve performance when there is a significant amount of low-frequency noise.

This type of analysis for the cooling parameter may also be apply to other areas, like the Wiener filter. Papež et al. (2018) showed that the messenger field method of Elsner & Wandelt (2013) for solving Wiener filter problem could also be written as a parameterized conjugate gradient algorithm. It stands to reason that such a system may also benefit from splitting and parameterizing its noise covariance, depending on the noise properties. (In the Wiener filter, Kodi Ramanah et al. 2017 additionally suggest that splitting the signal covariance and combining the uniform parts of the signal and noise.)

The benefits to map-making from a cooled messenger method seem to come from the cooling and not actually from the messenger field that inspired it. However, the messenger field approach may still have a role in the production and analysis of CMB maps. In particular, the close connection between the messenger method and Gibbs sampling may allow us to cheaply generate noise realizations of a converged map by generating samples from the map posterior distribution, something that we will continue to explore in future work.

For this work, B.C. and K.M.H. are supported by NSF award 1815887.

Appendix

The Derivation of $\eta$ Parameter Series

We know that the initial degree of heteroscedasticity $\eta_0 = 0$, which means the system is homoscedastic (uniform noise) to start. What would be a good value for the next parameter $\eta_1$?
vector, then the least upper bound of Equation (A3) is given by
\[ \frac{-\delta \chi^2(\hat{\mathbf{m}}(\eta_0), \eta_0)}{\chi^2(\hat{\mathbf{m}}(\eta_0), \eta_0)} \leq \frac{\delta \eta}{\max(N_f)}, \]  
(A6)
where \( \max(N_f) \) is the maximum eigenvalue of \( N \). We want \( \frac{-\delta \chi^2(\hat{\mathbf{m}}(\eta_0), \eta_0)}{\chi^2(\hat{\mathbf{m}}(\eta_0), \eta_0)} \) to be as large as possible, but it will not exceed 1. We combine Equations (A4) and (A6), and choose \( \delta \eta \) such that the least upper bound is equal to 1, to make sure the process is not going too fast. Thus we have
\[ \eta_i = \delta \eta = \frac{\tau}{\max(N_f)} = \frac{\min(N_f)}{\max(N_f) - \min(N_f)}. \]  
(A7)
Here \( N_f \) and \( \bar{N}_f \) are the eigenvalues of \( N \) and \( \bar{N} \) in the frequency domain. If the condition number of noise covariance matrix \( \kappa(N) = \max(N_f)/\min(N_f) \gg 1 \), then \( \eta_i \approx \kappa^{-1}(N) \).

What about the other parameters \( \eta_i \) with \( m > 1 \)? We use a similar analysis, letting \( \eta_{m+1} = \eta_m + \delta \eta_m \) with a small \( \delta \eta_m \ll 1 \). First, let us find the least upper bound
\[ \frac{-\delta \chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m)}{\chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m)} = \delta \eta_m \left( \frac{(d - \hat{P}m(\eta_m))N_{\eta_m}^{-1} \bar{N}N_{\eta_m}^{-1}(d - \hat{P}m(\eta_m))}{(d - \hat{P}m(\eta_m))N_{\eta_m}^{-1}(d - \hat{P}m(\eta_m))} \right) \]  
\[ \leq \delta \eta_m \max \left( \frac{\bar{N}_f}{\tau + \eta_m \bar{N}_f} \right). \]  
(A8)

The upper bound in the second line is a little bit tricky. Both matrix \( \bar{N} \) and \( N_{\eta_m}^{-1} \) can be simultaneously diagonalized in frequency space. For each eigenvector \( \mathbf{e}_i \), the corresponding eigenvalue of the matrix on the numerator \( N_{\eta_m}^{-1} \bar{N}N_{\eta_m}^{-1} \) is \( \lambda_f = \bar{N}_f(\tau + \eta_m \bar{N}_f)^{-2} \), and the eigenvalue for the matrix in the denominator \( N_{\eta_m}^{-1} \) is \( \gamma_f = (\tau + \eta_m \bar{N}_f)^{-1} \). Their eigenvalues are related by \( \lambda_f = [\bar{N}_f(\tau + \eta_m \bar{N}_f)]^{-1} \gamma_f \). For any vector \( \mathbf{v} = \sum \alpha_i \mathbf{e}_i \), we have
\[ \frac{\mathbf{v}^T N_{\eta_m}^{-1} \bar{N}N_{\eta_m}^{-1} \mathbf{v}}{\mathbf{v}^T N_{\eta_m}^{-1} \mathbf{v}} = \sum \alpha_i^2 \lambda_f = \sum \frac{\alpha_i^2 \gamma_f}{\tau + \eta_m \bar{N}_f} \leq \max \left( \frac{\bar{N}_f}{\tau + \eta_m \bar{N}_f} \right). \]  
(A9)

Again assuming \( \chi^2(\hat{\mathbf{m}}(\eta_{m+1}), \eta_{m+1}) \ll \chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m) \), which we expect to be satisfied for \( \eta_m \ll 1 \). That is because if \( \eta \ll 1 \), \( \chi^2(\hat{\mathbf{m}}(\eta), \eta) \) would be close to the minimum \( \chi^2 \), which means \( \chi^2(\hat{\mathbf{m}}(\eta_{m+1}), \eta_{m+1}) \ll \chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m) \), which would violate our assumption. Luckily, the final result (Equation (A14)) is a geometric series, only the last few \( \eta_m \) values fail to satisfy this condition. Similarly, we could set the least upper bound equal to 1. Then we get
\[ \eta_{m+1} = \eta_m + \delta \eta_m = \eta_m + \frac{\tau}{\max(N_f)}. \]  
(A10)

Therefore
\[ \eta_{m+1} = \eta_m + \delta \eta_m = \eta_m + \frac{\tau}{\max(N_f)}. \]  
(A11)

If written in the form
\[ \eta_{m+1} + \frac{\tau}{\max(N_f)} = 2(\eta_m + \frac{\tau}{\max(N_f)}) \]  
(A12)

It is easy to see that for \( m \geq 1 \), \( \eta_m + \frac{\tau}{\max(N_f)} \) forms a geometric series
\[ \eta_m + \frac{\tau}{\max(N_f)} = \left( \eta_1 + \frac{\tau}{\max(N_f)} \right)2^{m-1} = \frac{\tau}{\max(N_f)}2^m. \]  
(A13)

where we used \( \eta_1 = \tau/\max(N_f) \). Note that \( m = 0 \) and \( \eta_0 = 0 \) also satisfy this expression and we have the final expression for all \( \eta_m \)
\[ \eta_m = \min \left( 1 - \frac{\tau}{\max(N_f)}(2^m - 1) \right). \]  
(A14)

Here we need to truncate the series when \( \eta_m > 1 \).

In numerical simulations, we find that if we update the \( \eta \) parameter based on the more precise Equation (A5), there is only marginal improvement over the \( \eta \) series given in Equation (A14) for the 1/f noise model, and slight improvement when there is not much low-frequency noise. If we use Equation (A5), it ends up with fewer \( \eta \) parameters in the series, but the interval between \( \eta_i \) and \( \eta_{i-1} \) gets larger. In our simulation this sometimes causes one more iteration at a certain \( \eta \) value, so in the end there are only slight improvements. For a large data set that needs a lot of iterations to converge from \( \eta_i \) to \( \eta_{i+1} \), where several extra iterations may not be significant, using Equation (A5) may provide a larger performance boost.

**ORCID iDs**
- Bai-Chiang Chiang
  [https://orcid.org/0000-0002-2981-4951](https://orcid.org/0000-0002-2981-4951)
- Kevin M. Huffenberger
  [https://orcid.org/0000-0001-7109-0099](https://orcid.org/0000-0001-7109-0099)

**References**
- Dünner, R., Hasselfield, M., Marriage, T. A., et al. 2013, ApJ, 762, 10
- Elsner, F., & Wandelt, B. D. 2013, A&A, 549, A111
- Huffenberger, K. M., & Nesser, S. K. 2018, ApJ, 852, 92
- Janssen, M. A., & Gulkis, S. 1992, in NATO Advanced Science Institutes (ASI) Series C, Vol. 359 ed. M. Signore & C. Dupraz (Berlin: Springer) 391–408
- Kodi Ramanah, D., Lavaux, G., & Wandelt, B. D. 2017, MNRAS, 468, 1782
- Papež, J., Grigori, L., & Stompor, R. 2018, A&A, 620, A59
- Tegmark, M. 1997, ApJL, 480, L87