Draft version June 20, 2021
Typeset using LaTeX **twocolumn** style in AASTeX631

Cosmic Microwave Background map-making solutions improve with cooling

Bai-Qiang Qiang (KMH: want chinese characters?)[1] and Kevin M. Huffenberger [1]

[1] *Department of Physics, Florida State University, Tallahassee, Florida 32306*

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 non-uniform parts and adjusting their relative weight during the iterative solution. This "cooling" or perturbative approach is particularly effective when there is significant low-frequency noise in the timestream. A conjugate gradient algorithm applied to this modified system converges faster and to a higher fidelity solution than the standard conjugate gradient approach, for the same computational cost per iteration. We conclude that cooling is helpful separate from its appearance in the messenger methods. We give an analytical expression for the parameter that controls how gradually should change during the course of the solution.

Keywords: Computational methods — Cosmic microwave background radiation — Astronomy data reduction

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

$$\mathbf{d} = P\mathbf{m} + \mathbf{n} \tag{1}$$

where $\mathbf{d}$, $\mathbf{m}$, and $\mathbf{n}$ are the vectors of time-ordered data (TOD), the CMB sky-map signal, and measurement noise, and P is the sparse matrix that encodes the telescope's pointing. Of several mapmaking 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

$$\left(P^\dagger N^{-1} P\right)\hat{\mathbf{m}} = P^\dagger N^{-1}\mathbf{d} \tag{2}$$

where $\hat{\mathbf{m}}$ provides the generalized least squares minimization of the χ^2 statistic

$$\chi^2(\mathbf{m}) \equiv (\mathbf{d} - P\mathbf{m})^\dagger N^{-1}(\mathbf{d} - P\mathbf{m}). \tag{3}$$

Here we assume that the noise has zero mean $\langle \mathbf{n} \rangle = \mathbf{0}$, and noise covariance matrix is $N = \left\langle \mathbf{n}\mathbf{n}^\dagger \right\rangle$. Thus mapmaking is a standard linear regression problem. In case the noise is Gaussian, the COBE solution is also the maximum likelihood solution.

With current computation power, we cannot solve for $\hat{\mathbf{m}}$ by calculating $\left(P^\dagger N^{-1} P\right)^{-1} P^\dagger N^{-1}\mathbf{d}$ directly, since the $(P^\dagger N^{-1} P)$ matrix is too large to invert. The noise covariance matrix N is often sparse in frequency domain and the pointing matrix P is sparse in the time-by-pixel domain, and their product is dense. In experiments currently under design, there may be $\sim 10^{16}$ time samples and $\sim 10^9$ pixels, so these matrix inversions are intractable. 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 mapmaking 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 represent 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 the 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 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. However in numerical tests, Papež et al. (2018) did not find benefits to the cooling modification of the mapmaking system, in contrast to 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 mapmaking 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 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 mapmaking equation and our numerical experiments. In Section 3 we present our results. In Section 4 we interpret the mapmaking approach and its computational cost. In Section 5 we conclude. In appendices 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 λ to the map-making equation, and solved the linear system with the alternative covariance $N(\lambda) = \lambda\tau I + \bar{N}$. The parameter τ represents the uniform level of (white) noise in the covariance, $\bar{N}$ is the balance of the noise covariance, and the parameterized covariance equals the original covariance when the cooling parameter $\lambda = 1$. In this work we find it more convenient to work with the inverse cooling parameter $\eta = \lambda^{-1}$ and define the covariance as

$$N(\eta) = \tau I + \eta\bar{N} \tag{4}$$

which leads to the same system of mapmaking equations. (This is because $N(\eta) = \lambda^{-1}N(\lambda)$ and the mapmaking equation (Eq. 5) is insensitive to to scalar multiple of the covariance since is appears on both sides.) Since the non-white part $\bar{N}$ is the troublesome portion of the covariance, and we can think of the η parameter as turning it on slowly, adding a perturbation to the solution achieved at a particular stage, building ultimately upon the initial uniform covariance model.

Papež et al. (2018) showed that the conjugate gradient method can be easily applied to the parameterized mapmaking equation by iterating on

$$P^\dagger N(\eta)^{-1}P\ \hat{\mathbf{m}} = P^\dagger N(\eta)^{-1}\mathbf{d} \tag{5}$$

as we adjust the parameter. In our numerical experiments, we confirm that the conjugate gradient approach is converging faster than the fixed point iterations suggested by the messenger mapmaking method in Huffenberger & Næss (2018). For simplicity we fix the preconditioner to $M = P^\dagger P$ for all of calculations.

When $\eta = 0$, the noise covariance matrix $N(0)$ is proportional to identity matrix I, and solution is given by simple binned map $\mathbf{m}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$, which can be solved directly. The non-uniform part $\bar{N}$ is the troublesome portion of the covariance, and we can think of the η parameter as turning it on slowly, adding a perturbation to the solution achieved at a particular stage, building ultimately upon the initial uniform covariance model.

From the starting point, the cooling scheme requires the inverse cooling parameter η increase as $0 = \eta_0 \leq \eta_1 \leq \cdots \leq \eta_{\text{final}} = 1$, at which point we arrive at the desired mapmaking equation. We may iterate more than once at each intermediate η_i: we solve with conjugate gradient iterations

$$\left(P^\dagger N(\eta_i)^{-1}P\right)\hat{\mathbf{m}}(\eta_i) = P^\dagger N(\eta_i)^{-1}\mathbf{d}, \tag{6}$$

using the result from previous calculation $\hat{\mathbf{m}}(\eta_{i-1})$ as the initial value, and move to next parameter η_{i+1} when the residual $\left(P^\dagger N(\eta_i)^{-1}P\right)\hat{\mathbf{m}}(\eta_i) - P^\dagger N(\eta_i)^{-1}\mathbf{d} \simeq \mathbf{0}$. During our calculation we move to next η parameter when the norm of residual $\|\mathbf{r}\|$ is smaller than one tenth of standard deviation of each pixel, under the assumption that the noise is purely white with power spectrum $P(f) = \sigma^2$, which is the minimum value of our model Eq. (8). KMH: I think you have to be specific about the residual threshold.

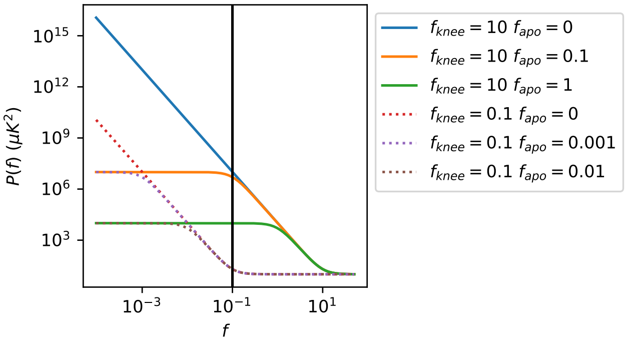

Figure 1. The noise power spectrum based on Eq. (8) with $\sigma^2 = 10$ μK^2 and $\alpha = 3$. Two knee frequencies $f_{\rm knee} = 10$ (solid lines) and $f_{\rm knee} = 0.1$ (dashed lines). For each knee frequency, we have $f_{\rm apo} = 0$, $0.1 f_{\rm knee}$ and $0.01 f_{\rm knee}$. The vertical line shows our scanning frequency.

2.2. *Choice of inverse cooling parameters η*

The next question is how to choose these monotonically increasing parameters η. If we choose them inappropriately, the solution converge slowly, because we waste effort converging on the wrong system. We also want to determine $\eta_1, \cdots, \eta_{n-1}$ before starting conjugate gradient iterations. The time ordered data $\mathbf{d}$ is very large, and we do not want to keep it in the system memory during calculation. If we determine $\eta_1, \cdots, \eta_{n-1}$ before the iterations, then we can precompute the right-hand side $P^\dagger N(\eta)^{-1}\mathbf{d}$ for each η_i and keep these map-sized objects in memory, instead of the entire time-ordered data.

In the appendix, we show that a generic good choice for the η parameters are the geometric series

$$\eta_i = \min\left\{ \left(2^i - 1\right) \frac{\tau}{\max(\bar{N}_f)},\ 1 \right\}, \tag{7}$$

where $\bar{N}_f$ is the frequency representation of the non-uniform part of the covariance. This is the main result. It tells us not only how to choose parameters η_i, but also when we should stop the perturbation, and set $\eta = 1$. For example, if noise covariance matrix N is almost white noise, then $\bar{N} = N - \tau I \approx 0$, and we would have $\tau/\max(\bar{N}_f) \gg 1$. This tell us that we don't need to use parameterized method at all, because $\eta_0 = 0$ and $\eta_1 = \eta_2 = \cdots = 1$. This corresponds to the standard conjugate gradient method with simple binned map as the initial guess (as recommended by Papež et al. 2018).

2.3. *Numerical Simulations*

To compare these algorithms, we need to do some simple simulation of scanning processes, and generate 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 range from $-1°$ to $+1°$. The signal has stokes parameters (I, Q, U) for intensity and linear polarization.

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 and scanning frequency $f_{\rm scan} = 0.1$ Hz and sampling frequency $f_{\rm sample} = 100$ Hz. The telescope scans the sky horizontally and then vertically, and then digitizes the position (x, y) into 512×512 pixels. This gives noiseless signal $P\mathbf{s}$.

We model the noise power spectrum with

$$P(f) = \sigma^2 \left(1 + \frac{f_{\rm knee}^\alpha + f_{\rm apo}^\alpha}{f^\alpha + f_{\rm apo}^\alpha} \right) \tag{8}$$

which is white at high frequencies, a power law below the knee frequency, and gives us the option to apodize the low frequency noise below an apodization frequency (like Papež et al. 2018). Note that as $f_{\rm apo} \to 0$, $P(f) \to \sigma^2(1 + (f/f_{\rm knee})^{-\alpha})$, and it becomes a $1/f$ 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 fixed $\sigma^2 = 10$ μK^2, $\alpha = 3$, and $f_{\rm knee} = 10$ Hz, and change $f_{\rm apo}$ to compare the performance under different noise models.

The noise covariance matrix

$$N_{ff'} = P(f) \frac{\delta_{ff'}}{\Delta_f} \tag{9}$$

is a diagonal matrix in frequency space, where Δ_f is equal to reciprocal of total scanning time $T \approx 1.05 \times 10^4$ seconds. In our calculations we choose different combination of $f_{\rm knee}$ and $f_{\rm apo}$, some of the power spectrum are shown in Figure 1.

Finally, we get the simulated time ordered data $\mathbf{d} = \mathbf{s} + \mathbf{n}$ by adding up signal and noise.

3. RESULTS

First let's compare the results with vanilla conjugate gradient method with simple preconditioner $P^\dagger P$. Figure (2) shows the results for 1/f noise model ($f_{\rm apo} = 0$) with different knee frequency. In Figure (3) we fixed $f_{\rm knee} = 10$ Hz, and change $f_{\rm apo}$. Here note that χ^2 in all figures are calculated based on Eq. (3) not $\chi^2(\mathbf{m}, \eta)$ in Eq. (A1). The $\chi^2_{\rm min}$ is calculated from perturbative conjugate gradient method with 100 η values, and it stops when the norm of residual $\|\mathbf{r}\| =$

[1] The source code and other information are available at https://github.com/Bai-Qiang/map_making_perturbative_approach

$\left\|P^{\dagger}N^{-1}\mathbf{d}-(P^{\dagger}N^{-1}P)\mathbf{m}\right\|$ per pixel is smaller than 10^{-10}, or after 1000 iterations.

As we can see in Figure (2) and the first graph in Figure (3), for $1/f$ noise model, when $f_{\text{knee}} \gtrsim 10 f_{\text{scan}}$ the parameterized method starts showing advantage over vanilla conjugate gradient method. From Figure (3) we can see that as we increase f_{apo} while fix f_{knee}, these two methods performs similar.

If we look at the power spectrum in Figure (1), when f_{knee} is small or f_{apo} is large there are not many large scale low frequency noise. So introducing η parameter could improve perform when there are large low noise contribution.

We also tried different α values. For $\alpha = 2$, the conclusion is the same as $\alpha = 3$. When $\alpha = 1$, there are not many low frequency noise, the vanilla conjugate gradient is preferred, except some cases with very large knee frequency like $f_{\text{knee}} = 100$ Hz and $f_{\text{apo}} = 0$ would favor parameterized method. In Papež et al. 2018, the $\alpha = 1$ and the noise power spectrum is apodized at $0.1 f_{\text{knee}}$, which corresponds to $f_{\text{apo}} \approx 0.1 f_{\text{knee}}$, and their knee frequency is the same as scanning frequency, so $f_{\text{knee}} = f_{\text{scan}} = 0.1$ in our cases. In their case there are not many low frequency noise, and we confirm that vanilla conjugate gradient method would converge faster.

4. DISCUSSION

4.1. *Intuitive Interpretation of η*

KMH: most of this is pretty similar to discussion in Huffenberger and Naess. The last paragraph is new.

In this section, let me introduce another way to understand the role of η. Our ultimate goal is to find $\hat{\mathbf{m}}(\eta = 1)$ which minimizes $\chi^2(\mathbf{m}) = (\mathbf{d} - P\mathbf{m})^{\dagger}N^{-1}(\mathbf{d} - P\mathbf{m})$. Since N is diagonal in frequency space, χ^2 could be written as a sum of all frequency mode $\left|(\mathbf{d} - P\mathbf{m})_f\right|^2$ with weight N_f^{-1}, such as $\chi^2(\mathbf{m}) = \sum_f \left|(\mathbf{d} - P\mathbf{m})_f\right|^2 N_f^{-1}$. N_f^{-1} is large when there is little noise at that frequency, and vice versa. Which means $\chi^2(\mathbf{m})$ would favor the low noise frequency mode over high noise ones. In other words the optimal map $\hat{\mathbf{m}}$ focusing on minimize the error $\mathbf{r} \equiv \mathbf{d} - P\mathbf{m}$ in the low-noise part.

After introducing η, we minimize $\chi^2(\mathbf{m}, \eta) = (\mathbf{d} - P\mathbf{m})^{\dagger}N_{\eta}^{-1}(\mathbf{d} - P\mathbf{m})$. For $\eta = 0$, $N_{\eta=0}^{-1} \propto I$ and the estimated map $\hat{\mathbf{m}}(\eta = 0)$ does not prioritize any frequency mode. As we slowly increase η, we decrease the weight for the frequency modes which have large noise, and focusing minimizing error for low noise part. If we start with $\eta_1 = 1$ directly, which corresponds to the vanilla conjugate gradient method, then the entire conjugate gradient solver will focus most on minimizing the low noise part, such that χ^2 would converge very fast at low noise region, but slowly on high noise part. Since it focus on low noise part only, it may be stuck at some local minimum point. To get to the global minimum, it need to adjust the low noise part, that would be difficult if it's stuck at an local minimum. However by introducing η parameter, we let the solver first treat every frequency equally. Then as η slowly increases, it gradually shifts focus from the highest noise to the lowest noise part. KMH: I feel what this is missing is why the high-noise modes get stuck though.

If we write the difference between final and initial χ^2 value as $\chi^2(\hat{\mathbf{m}}(1), 1) - \chi^2(\hat{\mathbf{m}}(0), 0) = \int_0^1 \mathrm{d}\eta \, \frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta), \eta)$, and use Eq. (A2). We note that when η is very small, the $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta), \eta)$ would have relatively large contribution from medium to large noise region, comparing to large η. So introducing η might improve the convergence of χ^2 at these regions, because the vanilla conjugate gradient method only focuses on the low noise part and it may have difficulty at these regions.

4.2. *Computational Cost*

To properly compare the performance cost of this method with respect to vanilla conjugate gradient method with simple preconditioner, we need to compare their computational cost at each iteration. The right hand side of parameterized map-making equation Eq. (5) could be computed before iterations, so it won't introduce extra computational cost. The most demanding part of conjugate gradient method is calculating $P^{\dagger}N^{-1}P\hat{\mathbf{m}}$, because it contains a Fourier transform of $P\hat{\mathbf{m}}$ from time domain to frequency domain and an inverse Fourier transform of $N^{-1}P\hat{\mathbf{m}}$ from frequency domain back to time domain, which is order $\mathcal{O}(n \log n)$ with n being the length of time ordered data. If we change N^{-1} to $N(\eta)^{-1}$, it won't add extra cost, since both matrices are diagonal in frequency domain. Therefore the computational cost it the same for one step.

However our previous analysis is based on $\chi^2(\hat{\mathbf{m}}(\eta_i), \eta_i)$ which is evaluated at $\hat{\mathbf{m}}(\eta_i)$ the estimated map at η_i. So We should update η_i to η_{i+1} when $\mathbf{m} \approx \hat{\mathbf{m}}(\eta_i)$. How do we know this condition is satisfied? Since for each new η_i value, we are solving a new set of linear equations $A(\eta_i)\hat{\mathbf{m}} = \mathbf{b}(\eta_i)$ with $A(\eta_i) = P^{\dagger}N(\eta_i)^{-1}P$ and $\mathbf{b}(\eta_i) = P^{\dagger}N(\eta_i)^{-1}\mathbf{d}$, and we could stop calculation and moving to next value η_{i+1} when the norm of residual $||\mathbf{r}(\eta_i)|| = ||\mathbf{b}(\eta_i) - A(\eta_i)\mathbf{m}||$ smaller than some small value. Calculate $||\mathbf{r}(\eta_i)||$ is part of conjugate gradient algorithm, so this won't add extra cost compare to vanilla conjugate gradient method. Therefore, overall introducing η won't have extra computational cost.

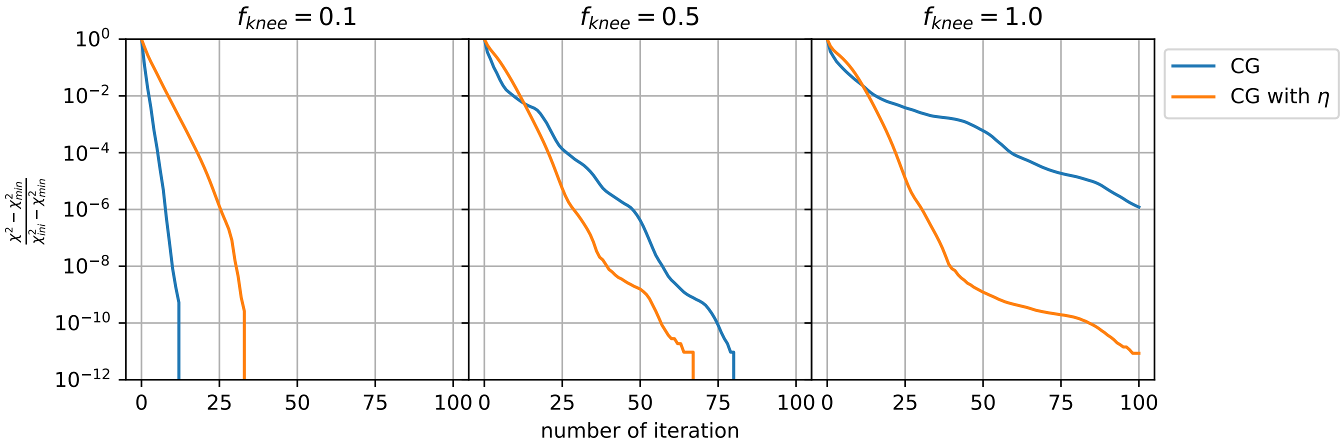

Figure 2. These three figures show the $\frac{\chi^2(\mathbf{m})-\chi^2_{\rm min}}{\chi^2_{\rm ini}-\chi^2_{\rm min}}$ changes for each iteration under different noise covariance matrix with fixed $f_{\rm apo} = 0$ and $f_{\rm knee}$ being 0.1, 0.5, and 1.0.

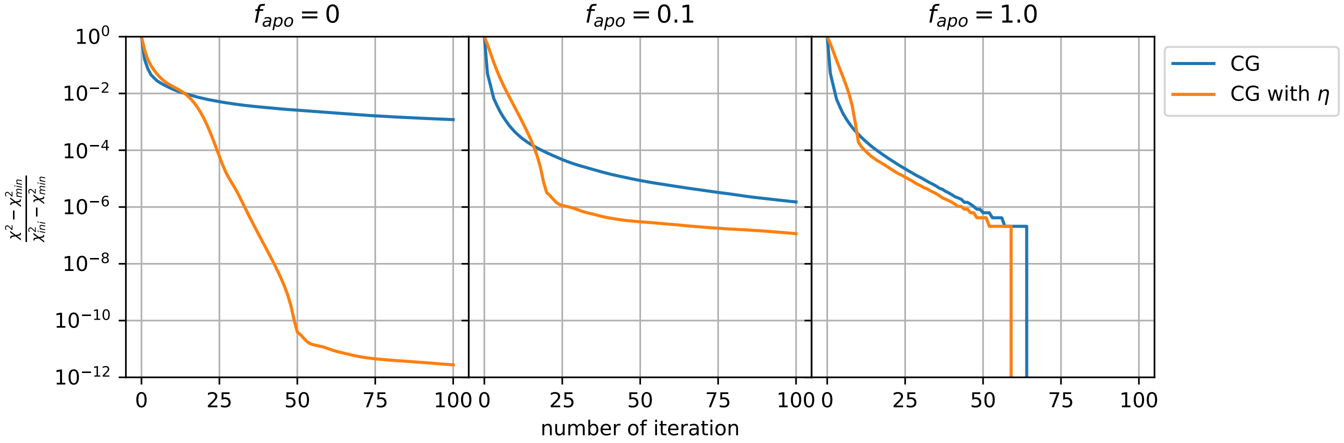

Figure 3. These three figures show the $\frac{\chi^2(\mathbf{m})-\chi^2_{\rm min}}{\chi^2_{\rm ini}-\chi^2_{\rm min}}$ changes for each iteration under different noise covariance matrix with fixed $f_{\rm knee} = 10$ and $f_{\rm apo}$ being 0, 0.1, and 1.0.

4.3. *Other η Choices*

Now let us compare the performance difference between choosing η parameters based on Eq. (7) and fixing number of η parameters n_η manually. We choose the η_i values using function `numpy.logspace(start=`$\ln(\eta_1)$`, stop=0, num=`n_η`, base=`e`)`. The results are showed in Figure (4).

In some cases the η series determined by Eq. (7) is ideal (the first graph in Figure (4)), in other cases Eq. (7) gives too many η values such that it is not optimal (the second and third graph in Figure (4)).

4.4. *Future Prospects*

In Appendix A, we determine $\delta\eta_m$ value based on the upper bound of $-\delta\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)/\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)$, and choose $\delta\eta_m$ such that the upper bound is equal to 1. The reason we use this upper bound instead of using

$$\delta\eta_m = -\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)/\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m) \tag{10}$$

directly, is that we don't want to keep the time ordered data $\mathbf{d}$ in system memory. In Figure (5) we can see if we use Eq. (10) for each $\delta\eta_m$, indeed it can improve performance. Especially for the third graph where the power spectrum does not have lots of low frequency noise by using Eq. (5) the result is close to vanilla conjugate gradient method. To further improve this method, we need to find more accurate expression for Eq. (A7).

5. CONCLUSIONS

KMH: We need some discussion of the things that haven't yet been demonstrated with the PCG, like multiple messenger fields. Has the Kodi-Ramanah dual mes-

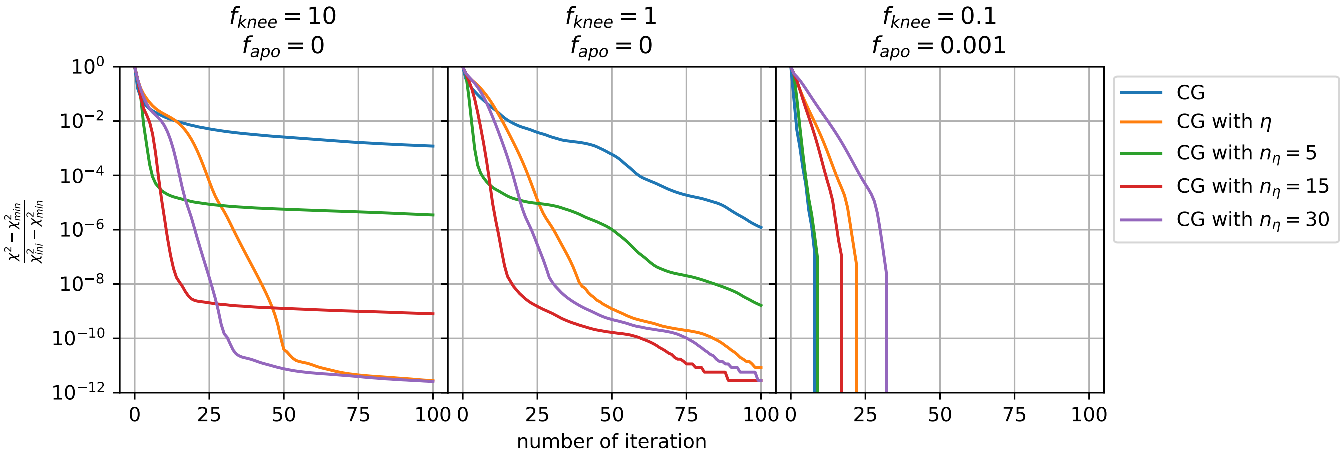

Figure 4. The blue line and the orange line are vanilla conjugate gradient method and parameterized conjugate gradient method. For three extra lines, we fix the number of η parameter n_η manually. Instead of using Eq. (7), we use `numpy.logspace(start=`$\ln(\eta_1)$`, stop=0, num=`n_η`, base=`e`)` to get all η parameters.

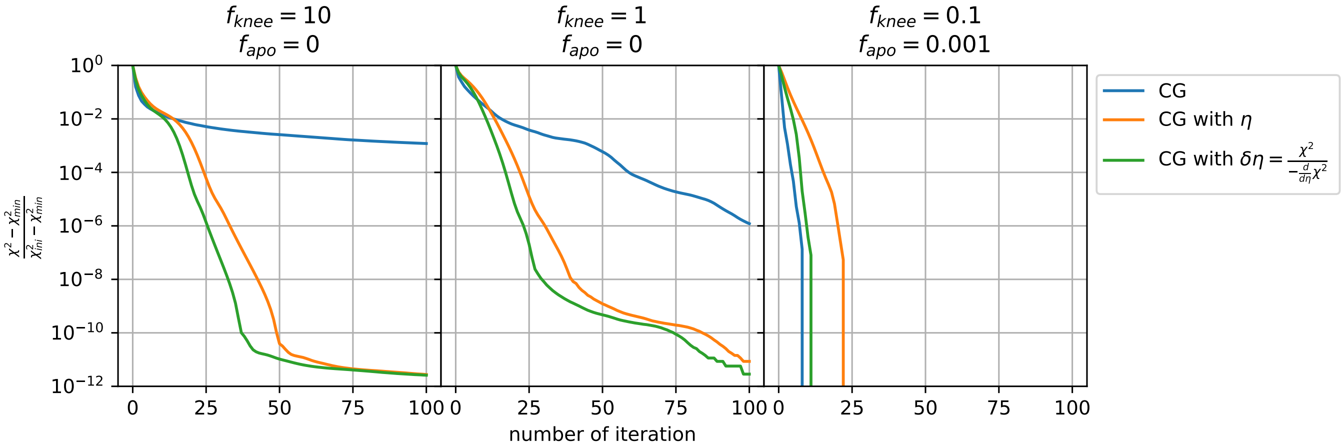

Figure 5. The blue line and orange line is the same as those in Figure (4) for reference. The extra green line shows the result when $\delta\eta_m$ is determined from Eq. (10) not from expression Eq. (7).

senger field scheme been demonstrated in a PCG scheme by Papez?

We presented a parameterized conjugate gradient method with parameter η based on the idea of messenger field separating the white noise out of noise covariance matrix. Then we gave an analytical expression for η series, and showed that this method would not introduce extra computational cost than traditional conjugate method.

We tested this method under different power spectrum both apodized and non-apodized. The results showed that this method is faster than traditional conjugate gradient method when there are significant amount of low frequency noise. But it could be further improved if we could get more accurate estimation for Eq. (10), either before iteration or without using time ordered data during iteration.

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

Papež et al. (2018) showed that the messenger field method solving Wiener filter problem introduced by Elsner & Wandelt (2013) could also be written as parameterized conjugate gradient algorithm. Then Kodi Ramanah et al. (2017) introduced dual messenger field method to Wiener filter. If applying our idea to Wiener filter problem, hopefully, it may also bring improvements.

BQ and KH are supported by NSF award 1815887.

APPENDIX

A. THE SEQUENCE OF INVERSE COOLING PARAMETERS

We know that the initial inverse cooling parameter $\eta_0 = 0$. What would be good value for the next parameter η_1? To simplify notation, we use N_η to denote $N(\eta) = \tau I + \eta \bar{N}$. For some specific η value, the minimum χ^2 value is given by the optimized map $\hat{\mathbf{m}}(\eta) = \left(P^\dagger N_\eta^{-1} P\right)^{-1} P^\dagger N_\eta^{-1} \mathbf{d}$, which minimizes

$$\chi^2(\hat{\mathbf{m}}(\eta), \eta) = \left(\mathbf{d} - P\hat{\mathbf{m}}(\eta)\right)^\dagger N_\eta^{-1} \left(\mathbf{d} - P\hat{\mathbf{m}}(\eta)\right). \tag{A1}$$

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.

Let us first consider $\eta_1 = \eta_0 + \delta\eta = \delta\eta$ such that $\eta_1 = \delta\eta$ is very small quantity, $\delta\eta \ll 1$. Since $\hat{\mathbf{m}}(\eta)$ minimizes $\chi^2(\hat{\mathbf{m}}(\eta), \eta)$, we have $\frac{\partial}{\partial \hat{\mathbf{m}}} \chi^2(\hat{\mathbf{m}}(\eta), \eta) = 0$, and using chain rule

$$\frac{\mathrm{d}}{\mathrm{d}\eta} \chi^2(\hat{\mathbf{m}}(\eta), \eta) = \frac{\partial}{\partial \eta} \chi^2(\hat{\mathbf{m}}(\eta), \eta) = -(\mathbf{d} - P\hat{\mathbf{m}}(\eta))^\dagger N_\eta^{-1} \bar{N} N_\eta^{-1} (\mathbf{d} - P\hat{\mathbf{m}}(\eta)) \tag{A2}$$

Then the fractional decrease of $\chi^2(\hat{\mathbf{m}}(0), 0)$ from $\eta_0 = 0$ to $\eta_1 = \delta\eta$ is

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(0), 0)}{\chi^2(\hat{\mathbf{m}}(0), 0)} = -\delta\eta \frac{\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(0), 0)}{\chi^2(\hat{\mathbf{m}}(0), 0)} = \delta\eta \frac{1}{\tau} \frac{(\mathbf{d} - P\hat{\mathbf{m}}(0))^\dagger \bar{N} (\mathbf{d} - P\hat{\mathbf{m}}(0))}{\left(\mathbf{d} - P\hat{\mathbf{m}}(0)\right)^\dagger \left(\mathbf{d} - P\hat{\mathbf{m}}(0)\right)} \tag{A3}$$

Here we put a minus sign in front of this expression such that it's non-negative, and use $N_{\eta=0} = \tau I$ at the second equality. Since it is hard to analyze $\mathbf{d} - P\hat{\mathbf{m}}$ under frequency domain, we treat it as an arbitrary vector, then the least upper bound is given by

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(0), 0)}{\chi^2(\hat{\mathbf{m}}(0), 0)} \leq \frac{\delta\eta}{\tau} \max(\bar{N}_f) \tag{A4}$$

where $\max(\bar{N}_f)$ is the maximum eigenvalue of $\bar{N}$. Here if we assume that initial χ^2 value $\chi^2(\hat{\mathbf{m}}(0), 0)$ is much larger than final value $\chi^2(\hat{\mathbf{m}}(1), 1)$, $\chi^2(\hat{\mathbf{m}}(0), 0) \gg \chi^2(\hat{\mathbf{m}}(1), 1)$, then we would expect

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(0), 0)}{\chi^2(\hat{\mathbf{m}}(0), 0)} = 1 - \frac{\chi^2(\hat{\mathbf{m}}(1), 1)}{\chi^2(\hat{\mathbf{m}}(0), 0)} \approx 1^- \tag{A5}$$

The upper bound is strictly smaller than 1. Ideally, if $\delta\chi^2(\hat{\mathbf{m}}(0), 0) = \chi^2(\hat{\mathbf{m}}(1), 1) - \chi^2(\hat{\mathbf{m}}(0), 0)$, then it would get close to the final χ^2 at next iteration, but we do not know the final $\chi^2(\hat{\mathbf{m}}(1), 1)$. So we want $\left|\frac{\delta\chi^2(\hat{\mathbf{m}}(0),0)}{\chi^2(\hat{\mathbf{m}}(0),0)}\right|$ to be as large as possible, so it could converge fast, but subject to another constraint that the least upper bound cannot exceed 1. Therefore we can choose $\delta\eta$ such that the least upper bound is equal to 1. Thus we choose

$$\eta_1 \equiv \frac{\tau}{\max(\bar{N}_f)} = \frac{\min(N_f)}{\max(N_f) - \min(N_f)}. \tag{A6}$$

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 $\eta_1 \approx \kappa^{-1}(N)$.

What about the other parameters η_m with $m > 1$? We use a similar analysis, letting $\eta_{m+1} = \eta_m + \delta\eta_m$ with a small $\delta\eta_m \ll 1$, and set the least upper bound of relative decrease equal to 1.

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m)}{\chi^2(\hat{\mathbf{m}}(\eta_m), \eta_m)} = \delta\eta_m \frac{(\mathbf{d} - P\hat{\mathbf{m}}(\eta_m))^\dagger N_{\eta_m}^{-1} \bar{N} N_{\eta_m}^{-1} (\mathbf{d} - P\hat{\mathbf{m}}(\eta_m))}{\left(\mathbf{d} - P\hat{\mathbf{m}}(\eta_m)\right)^\dagger N_{\eta_m}^{-1} \left(\mathbf{d} - P\hat{\mathbf{m}}(\eta_m)\right)} \tag{A7}$$

$$\leq \delta\eta_m \max\left(\frac{\bar{N}_f}{\tau + \eta_m \bar{N}_f}\right) \tag{A8}$$

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 $\mathbf{e}_f$, 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 matrix on the denominator $N_{\eta_m}^{-1}$ is $\gamma_f = (\tau+\eta_m\bar{N}_f)^{-1}$. Their eigenvalues are related by $\lambda_f = [\bar{N}_f/(\tau+\eta_m\bar{N}_f)]\gamma_f$. For any vector $\mathbf{v} = \sum_f \alpha_f \mathbf{e}_f$, we have

$$\frac{\mathbf{v}^\dagger N_{\eta_m}^{-1}\bar{N}N_{\eta_m}^{-1}\mathbf{v}}{\mathbf{v}^\dagger N_{\eta_m}^{-1}\mathbf{v}} = \frac{\sum_f \alpha_f^2\lambda_f}{\sum_f \alpha_f^2\gamma_f} = \frac{\sum_f \alpha_f^2\gamma_f\bar{N}_f/(\tau+\eta_m\bar{N}_f)}{\sum_f \alpha_f^2\gamma_f} \le \max\left(\frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\right). \tag{A9}$$

Similarly, we could set the least upper bound equal to 1. Then we get

$$\delta\eta_m = \min\left(\frac{\tau+\eta_m\bar{N}_f}{\bar{N}_f}\right) = \eta_m + \frac{\tau}{\max(\bar{N}_f)}. \tag{A10}$$

Therefore

$$\eta_{m+1} = \eta_m + \delta\eta_m = 2\eta_m + \frac{\tau}{\max(\bar{N}_f)} \tag{A11}$$

The final term $\tau/\max(\bar{N}_f) = \eta_1$ becomes subdominant after a few terms, and we see that the η_m increase like a geometric series. Here we assumed that $\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m) \gg \chi^2(\hat{\mathbf{m}}(1),1)$, which we expect it to be satisfied for our assumed $\eta_m \ll 1$. Since the final result is geometric series, only the last few η_m values fail to be much smaller than 1.

If written in the form $\eta_{m+1} + \tau/\max(\bar{N}_f) = 2\big(\eta_m + \tau/\max(\bar{N}_f)\big)$ it's easy to see that for $m \ge 1$, $\eta_m + \tau/\max(\bar{N}_f)$ forms a geometric series

$$\eta_m + \frac{\tau}{\max(\bar{N}_f)} = \left(\eta_1 + \frac{\tau}{\max(\bar{N}_f)}\right)2^{m-1} = \frac{\tau}{\max(\bar{N}_f)}2^m \tag{A12}$$

where we used $\eta_1 = \tau/\max(\bar{N}_f)$. Note that $m = 0$ and $\eta_0 = 0$ also satisfy this expression and we've got final expression for all η_m

$$\eta_m = \min\left\{1,\ \frac{\tau}{\max(\bar{N}_f)}(2^m - 1)\right\} \tag{A13}$$

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

REFERENCES

Dünner, R., Hasselfield, M., Marriage, T. A., et al. 2013, ApJ, 762, 10, doi: 10.1088/0004-637X/762/1/10

Elsner, F., & Wandelt, B. D. 2013, A&A, 549, A111, doi: 10.1051/0004-6361/201220586

Huffenberger, K. M., & Næss, S. K. 2018, The Astrophysical Journal, 852, 92, doi: 10.3847/1538-4357/aa9c7d

Janssen, M. A., & Gulkis, S. 1992, in NATO Advanced Science Institutes (ASI) Series C, ed. M. Signore & C. Dupraz, Vol. 359 (Springer), 391–408

Kodi Ramanah, D., Lavaux, G., & Wandelt, B. D. 2017, MNRAS, 468, 1782, doi: 10.1093/mnras/stx527

Papež, J., Grigori, L., & Stompor, R. 2018, A&A, 620, A59, doi: 10.1051/0004-6361/201832987

Tegmark, M. 1997, ApJL, 480, L87, doi: 10.1086/310631