Draft version April 12, 2021
Typeset using LaTeX **twocolumn** style in AASTeX631

Perturbative Approach to Solve the Map-Making Equation

Bai-Qiang Qiang[1] and Kevin M. Huffenberger[1]

[1] *Florida State University*

ABSTRACT

Some abstract

Keywords: Some Keywords

1. INTRODUCTION

The cosmic microwave background is an electromagnetic radiation coming from early stage of our Universe. Based on the hot Big Bang model, before the recombination epoch, the photons were tightly coupled with free electrons and protons via Thomson scattering. As the universe expanded and cooled down, free electrons and protons combined into neutral hydrogen atom. This process is called recombination in cosmology. Shortly after this epoch, the photons could propagate freely and would not be scattered by charged particles. Today, we receive the photons produced at that time and called them the cosmic microwave background radiation. Studying these photons coming from early universe can help us to constrain theoretical model and cosmological parameters **?**. The next generation CMB observations will have much higher resolution and generate more data. So we need an efficient way to process the data. One step of the the processing is map making, which gives an estimated map based on the raw observation data.

Recently Elsner and Wandelt **?** introduced a new method called messenger field to solve Wiener filter, and then this technique was being applied to map making equation by Huffenberger and Naess **?**. It has been shown that this messenger field method is equivalent to applying a preconditioner to the original problem and introducing an extra cooling parameter λ, but whether this cooling parameter will boost performance compare to the (traditional) conjugate gradient method is still controversial **?**. Here I give a detailed analysis of this parameter and show that it may improve performance under some circumstances, if we properly choose its values.

The map making procedure could be summarized in equation

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

where $\mathbf{d}$, P, $\mathbf{m}$, $\mathbf{n}$ are time-ordered data (TOD), pointing matrix, CMB map, and noise. The time-ordered data we collected is given by map signal $P\mathbf{m}$ plus noise $\mathbf{n}$. The pointing matrix P acting on the map gives the signal of the map at some specific position of sky where telescope is pointing at. Here we could assume that the noise has zero mean $\langle \mathbf{n} \rangle = \mathbf{0}$, since if it is not zero, we can always subtract its mean value to make it zero. And noise covariance matrix could be written as $N = \langle \mathbf{n}\mathbf{n}^\dagger \rangle$.

As we can see the map making model Eq.(1) mathematically is a standard linear regression problem, with *design matrix* being pointing matrix P, and *regression coefficients* are $\mathbf{m}$. Naturally, we want to estimate linear regression coefficients $\mathbf{m}$, with *generalized least square* (GLS) technique. The noise $\mathbf{n}$ is *heteroscedastic*: the variances N are different for various frequencies. Usually detectors have a $1/f$ noise pattern **?**. The *generalized least square* (GLS) will provide better estimation than *ordinary least square* (OLS) method, because the data is heteroscedastic so we would like to focusing on fitting the data with lower noise.

The GLS estimated map $\hat{\mathbf{m}}$ is given by

$$\hat{\mathbf{m}} = \arg\min_{\mathbf{m}} \chi^2(\mathbf{m}) \tag{2}$$

where

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

The estimated map $\hat{\mathbf{m}}$ is the one that minimizes $\chi^2(\mathbf{m})$.

$$\hat{\mathbf{m}} = \left(P^\dagger N P\right)^{-1} P^\dagger N^{-1} \mathbf{d} \tag{4}$$

This is also called the *map making equation*.

2. SOLVING THE MAP MAKING EQUATION

Rewrite the map making equation Eq.(4) in the form

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

If we define $A = P^\dagger N^{-1} P$ and $\mathbf{b} = P^\dagger N^{-1}\mathbf{d}$, then it could be written as $A\hat{\mathbf{m}} = \mathbf{b}$.

Based on current computation power, it is impossible to solve $\hat{\mathbf{m}}$ by calculating $\hat{\mathbf{m}} = \left(P^\dagger N^{-1} P\right)^{-1} P^\dagger N^{-1}\mathbf{d}$ directly, since the noise covariance matrix N is sparse in frequency domain, and pointing matrix P is sparse in (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. It is impossible to do these matrix multiplication directly and then take inverse. However, for a vector with size of map $\hat{\mathbf{m}}$, we could calculate $P^\dagger N^{-1} P\hat{\mathbf{m}} = A\hat{\mathbf{m}}$ by first taking Fourier transform $P\hat{\mathbf{m}}$ then inverse Fourier transform $N^{-1}P\hat{\mathbf{m}}$. This means it can be solved by the conjugate gradient method.

2.1. *Preconditioner*

To improve the performance of the conjugate gradient method, we could apply a preconditioner M to original problem $A\hat{\mathbf{m}} = \mathbf{b}$, which then becomes $M^{-1}A\hat{\mathbf{m}} = M^{-1}\mathbf{b}$. The preconditioner should reduce the condition number of original problem, so that the conjugate gradient method will converge faster. We want the preconditioner to capture as much information as possible from matrix A, but still keep it relative easy to calculate M^{-1}. For example, if $M = A$, $M^{-1}A\hat{\mathbf{m}} = M^{-1}\mathbf{b}$ would be solved immediately, but M^{-1} will be extremely difficult to calculate. We could simply choose $M = P^\dagger P$, and the operation $M^{-1}\mathbf{m} = \left(P^\dagger P\right)^{-1}\mathbf{m}$ is the average over each pixel of map $\mathbf{m}$.

For the conjugate gradient method, we need an initial guess map $\hat{\mathbf{m}}_0$. We can use zero vector $\hat{\mathbf{m}}_0 = \mathbf{0}$ as initial guess, but the simple binned map $\hat{\mathbf{m}}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$ is a better choice (it is a the solution for white noise case $N \propto I$). (Papež et al. 2018**?**) showed that using $\hat{\mathbf{m}}_0$ as initial guess could improve performance significantly compare to zero vector $\mathbf{0}$ in come cases. As stated before, we can calculate $\left(P^\dagger P\right)^{-1}$ acting on any map-sized object, and $P^\dagger\mathbf{d}$ is indeed a map size object, so we could obtain simple binned map by calculating $\hat{\mathbf{m}}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$ directly.

For the conjugate gradient method with simple preconditioner $M = P^\dagger P$, we have all we need. Next we only need to use conjugate gradient algorithm solve the problem.

2.2. *Parameterized Conjugate Gradient Method*

The above results have appeared in the literature. Here we now show that we can improve performance in some cases by using a parameterized version of the map making equation Eq.(5). The idea is that map making equation Eq.(5) is hard to solve due to noise covariance matrix is sandwiched between $P^\dagger P$. But if noise covariance matrix N is proportional to identity matrix I, then its solution is given by simple binned map $\mathbf{m}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$, which could be solved directly. We can parameterize the noise covariance matrix N with a parameter η, such that initially $\eta = \eta_i$ we have $N(\eta_i) \propto I$ and in the end $\eta = \eta_f$ and $N(\eta_f) \propto N$, such that the final solution is what we want. We expect that the parameterized noise covariance matrix $N(\eta)$ would connect our initial guess $\hat{\mathbf{m}}_0$ and final solution $\hat{\mathbf{m}}$ as we change η from η_i to η_f.

Now instead of Eq.(5), we are solving

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

Now question is how to find $N(\eta)$ such that $N(\eta_i) \propto I$ and $N(\eta_f) \propto N$? Since the non-white noise part of N is the difficult portion, we could think of it as a perturbation term, which adds upon the white noise. Initially there is only white noise and solution is given by $\hat{\mathbf{m}}_0$, then we gradually add extra noise into this equation by changing η from 0 to 1. At the end when $\eta = 1$ we are solving equation Eq.(5).

Therefore we separate noise covariance matrix into two parts $N = \tau I + \bar{N}$ where τ is the minimum eigenvalue of N. Then we define $N(\eta) = \tau I + \eta\bar{N}$, with perturbation parameter η which satisfies $\eta_i = 0$ and $\eta_f = 1$.

Eq.(6) then becomes

$$\left(P^\dagger(\tau I + \eta\bar{N})^{-1} P\right)\hat{\mathbf{m}}(\eta) = P^\dagger\left(\tau I + \eta\bar{N}\right)^{-1}\mathbf{d} \tag{7}$$

We require the perturbation parameter η being monotonically increase series $0 = \eta_0 < \eta_1 < \cdots < \eta_n = 1$. For some specific η_m, we use conjugate gradient method to solve equation $\left(P^\dagger N(\eta_m)^{-1} P\right)\hat{\mathbf{m}}(\eta_m) = P^\dagger N(\eta_m)^{-1}\mathbf{d}$ with simple preconditioner $P^\dagger P$, and using $\hat{\mathbf{m}}(\eta_{m-1})$ as the initial value. The initial guess is $\hat{\mathbf{m}}(\eta_0) = \mathbf{m}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$.

2.2.1. *Choosing perturbation parameters η*

The next question is how we choose these monotonically increasing parameters η. If we choose these parameters inappropriately, it would only makes it converge slower. Also we want to determine $\eta_1, \cdots, \eta_{n-1}$ before starting conjugate gradient iteration. That's because time ordered data $\mathbf{d}$ is very large, and we don't want to keep it in the system RAM during calculation. If $\eta_1, \cdots, \eta_{n-1}$ could be determined before the iterations, then we can first calculate $P^\dagger N(\eta)^{-1}\mathbf{d}$ for each η_m and store these map-sized objects in RAM, instead of the entire time-ordered data $\mathbf{d}$.

First let us try to find out our starting point η_1. What would be good value for η_1?

Here to simplify notation, I will use N_η to denote $N(\eta)$. The estimated map $\hat{\mathbf{m}}(\eta) = \left(P^\dagger N_\eta^{-1} P\right)^{-1} P^\dagger N_\eta^{-1}\mathbf{d}$ which minimizes

$$\chi^2(\mathbf{m},\eta) = (\mathbf{d}-P\mathbf{m})^\dagger N_\eta^{-1}(\mathbf{d}-P\mathbf{m}). \tag{8}$$

For some specific η value, the minimum χ^2 value is given by

$$\chi^2(\hat{\mathbf{m}}(\eta),\eta) = \left(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\right)^\dagger N_\eta^{-1}\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\right) \tag{9}$$

Now let us see how $\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ changes as we change η.

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)\\ =&-\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\right)^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta)).\end{aligned} \tag{10}$$

Notice that $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) = \frac{\partial}{\partial\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ (the total derivative is the partial derivative), because by the definition of $\hat{\mathbf{m}}(\eta)$ it minimize $\chi^2(\mathbf{m},\eta)$ for some fixed η value, implies $\frac{\partial}{\partial\mathbf{m}}\chi^2(\hat{\mathbf{m}}(\eta),\eta) = 0$.

To further simplify the analysis, let's assume that the noise covariance matrix $N = \left\langle \mathbf{n}\mathbf{n}^\dagger\right\rangle$ is diagonal in the frequency domain. Therefore $\bar{N}$ and N_η are also diagonal in the frequency domain by definition, and all the diagonal elements are greater than or equal to zero, because covariance matrix is positive semi-definite. Also, we can conclude that matrix $N_\eta^{-1}\bar{N}N_\eta^{-1}$ is positive semi-definite matrix. Based on Eq. (10), we know that $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) \le 0$, so $\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ is always decreasing as η changes from 0 to 1.

The relative decrease of $\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ at η is defined as

$$\begin{aligned}&-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)}\\ =&\delta\eta\frac{(\mathbf{d}-P\hat{\mathbf{m}}(\eta))^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta))}{\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\right)^\dagger N_\eta^{-1}\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\right)}\end{aligned} \tag{11}$$

Here we put a minus sign in front of $\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$, such that it's non-negative. If we choose $\eta_1 = \eta_0 + \delta\eta = \delta\eta$ such that $\eta_1 = \delta\eta$ is very small quantity. Then the relative decrease from $\eta_0 = 0$ to $\eta_1 = \delta\eta$ is

$$\begin{aligned}-\frac{\delta\chi^2(\hat{\mathbf{m}}(0),0)}{\chi^2(\hat{\mathbf{m}}(0),0)} &= \delta\eta\frac{1}{\tau}\frac{(\mathbf{d}-P\hat{\mathbf{m}}(0))^\dagger\bar{N}(\mathbf{d}-P\hat{\mathbf{m}}(0))}{\left(\mathbf{d}-P\hat{\mathbf{m}}(0)\right)^\dagger\left(\mathbf{d}-P\hat{\mathbf{m}}(0)\right)}\\ &\le \frac{\delta\eta}{\tau}\max(\bar{N}_f)\end{aligned} \tag{12}$$

where in first line we used the property $N_{\eta=0} = \tau I$, and in second line we used the property that the positive semi-definite $\bar{N}$ is diagonal in frequency domain and its maximum eigenvalue is $\max(\bar{N}_f)$.

Ideally, we want $\delta\chi^2(\hat{\mathbf{m}}(0),0) = \chi^2(\hat{\mathbf{m}}(1),1) - \chi^2(\hat{\mathbf{m}}(0),0)$, such that it would get close to the final χ^2 at next iteration. Here if we assume that initial χ^2 value $\chi^2(\hat{\mathbf{m}}(0),0)$ is much larger than final value $\chi^2(\hat{\mathbf{m}}(1),1)$, then we would expect $\left|\delta\chi^2(\hat{\mathbf{m}}(0),0)/\chi^2(\hat{\mathbf{m}}(0),0)\right| \approx 1^-$. To make sure it will not start too fast, we do not want $\left|\delta\chi^2(\hat{\mathbf{m}}(0),0)/\chi^2(\hat{\mathbf{m}}(0),0)\right|$ to exceed 1. So we could set an upper bound $\delta\eta\max(\bar{N}_f)/\tau = 1$ and set

$$\eta_1 = \frac{\tau}{\max(\bar{N}_f)} = \frac{\min(N_f)}{\max(N_f)-\min(N_f)} \tag{13}$$

Here N_f is the eigenvalues of noise covariance matrix N under 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 could use a similar analysis, and suppose $\eta_{m+1} = \eta_m + \delta\eta_m$ with a small $\delta\eta_m$, and find the relative decrease

$$\begin{aligned}&-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)}{\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)}\\ =&\delta\eta_m\frac{(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m))^\dagger N_{\eta_m}^{-1}\bar{N}N_{\eta_m}^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m))}{\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m)\right)^\dagger N_{\eta_m}^{-1}\left(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m)\right)}\\ \le&\delta\eta_m\max\left(\frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\right)\end{aligned} \tag{14}$$

Similarly, we could set the upper bound $\delta\eta_m\max\left(\frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\right) = 1$, [1] and then we get

$$\delta\eta_m = \min\left(\frac{\tau+\eta_m\bar{N}_f}{\bar{N}_f}\right) = \eta_m + \frac{\tau}{\max(\bar{N}_f)}. \tag{15}$$

Therefore

$$\eta_{m+1} = \eta_m + \delta\eta_m = 2\eta_m + \frac{\tau}{\max(\bar{N}_f)} \tag{16}$$

As we can see, $\eta_1, \cdots, \eta_n$ should increase like a geometric series. And written in the form $\eta_{m+1} + \frac{\tau}{\max(\bar{N}_f)} = 2\left(\eta_m + \frac{\tau}{\max(N_f)}\right)$ it's easy to see that for $m \ge 1$, $\eta_m + \frac{\tau}{\max(\bar{N}_f)}$ forms a geometric series

$$\eta_m + \frac{\tau}{\max(\bar{N}_f)} = \frac{\tau}{\max(\bar{N}_f)}2^m \tag{17}$$

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

[1] Here we also 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 $0 \simeq \eta_m \ll 1$. Since final result Eq.(18) is geometric series, only a few η_m values won't satisfy this condition.

for all η_i

$$\eta_i = \min\left\{1,\ \frac{\tau}{\max(\bar{N}_f)}\left(2^i - 1\right)\right\} \tag{18}$$

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

This is the main result. Eq. (18) 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 $\frac{\tau}{\max(\bar{N}_f)} \gg 1$. This tell us that we don't need to use parameterized method at all, because $\eta_1 = 1$. Note that the vanilla conjugate gradient method with simple binned map as initial guess corresponds to choosing $\eta_0 = 0$ and $\eta_1 = \eta_2 = \cdots = 1$.

2.2.2. *Intuitive Interpretation of η*

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 $\chi^2(\mathbf{m}) = (\mathbf{d} - P\mathbf{m})^\dagger N^{-1}(\mathbf{d} - P\mathbf{m})$. Here we also assumed that N is diagonal in frequency space. With this condition χ^2 could be written as a sum of all frequency mode $|(\mathbf{d} - P\mathbf{m})_f|^2$ with weight N_f^{-1}, such as $\chi^2(\mathbf{m}) = \sum_f |(\mathbf{d} - P\mathbf{m})_f|^2 N_f^{-1}$. 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, because low noise part has higher weight. 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 each η value as it increase from 0 to 1. For $\eta = 0$, $N^{-1}_{\eta=0} \propto I$ and the estimated map $\hat{\mathbf{m}}(\eta = 0)$ does not prioritize any frequency mode when minimizing the error. 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 only focusing on minimizing low noise part, such that χ^2 would converge very fast at low noise region, but relative slow on high noise part. However by introducing η parameter, we let the solver first treat every frequency equally. Then as η slowly increases, it gradually shifts focus to low noise part. 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.(10). 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.

2.3. *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.(6) could be computed before iterations, so it won't introduce extra computational cost during iterations. 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 $N^{-1}(\eta)$, it won't add extra cost. Therefore the computational cost it the same for one step.

However in previous analysis, our choice of parameters η_i is based on $\delta\chi^2(\hat{\mathbf{m}}(\eta_i), \eta_i)$ which is evaluated at $\mathbf{m} = \hat{\mathbf{m}}(\eta_i)$ the estimated map at η_i. We update η_i to η_{i+1} when $\mathbf{m} \approx \hat{\mathbf{m}}(\eta_i)$. How do we know current map $\mathbf{m}$ is close to $\hat{\mathbf{m}}(\eta_i)$? Because 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 specific value. Since when doing conjugate gradient algorithm we calculate $\mathbf{r}$ and stop the iteration when $||\mathbf{r}||$ is small enough, now after introducing parameter η, we move to next parameter η_{i+1} when $||\mathbf{r}(\eta_i)||$ is small enough. Again, this won't add extra cost compare to vanilla conjugate gradient method.

Therefore we find that the only significant cost after adding perturbation parameter η, is to to find out $\mathbf{b}(\eta_i)$ for each $\eta_i \neq 1$ before starting the iterations. And this is one time calculation, it's negligible compare to remaining calculations.

2.4. *Messenger Field Method*

The messenger field method is a fixed point iterative solver introduced by Elsner and Wandelt (2013) **?** to solve Wiener filter. Later on Huffenberger and Næss (2018) **?** applied this method to map-making problem, and showed that in some cases messenger field is better than conjugate gradient with a simple preconditioner. Papež el al.(2018) **?** proved that messenger field is equivalent to apply a preconditioner to map making equation Eq.(4), and it can be solved using both fixed

point iteration and preconditioned conjugate gradient methods. They showed that in some conjugate gradient with simple preconditioner outperforms messenger field method with both fixed point iteration and preconditioned conjugate gradient methods.

Messenger field method similarly separate noise covariance matrix $N = \bar{N} + T$, with $T = \tau I$ and τ being the minimum eigenvalue of N. Then there is a cooling parameter λ such that $N(\lambda) = \bar{N} + \lambda T$, with initial λ being a very large number and final λ being 1. As you might guess λ is related to η by $\lambda = 1/\eta$.

After apply preconditioner $P^\dagger T^{-1} P$ to the map making equation Eq.(4), we get:

$$\hat{\mathbf{m}} = \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger T^{-1} \left(T^{-1} + \bar{N}^{-1}\right)^{-1} \times \left[T^{-1} P \hat{\mathbf{m}} + \bar{N}^{-1} \mathbf{d}\right] \tag{19}$$

To add cooling parameter λ, we only need to change T to λT and N to $N(\lambda)$. Then we could write it as fixed point iteration form

$$\begin{cases} \mathbf{t}_i = \left((\lambda T)^{-1} + \bar{N}^{-1}\right)^{-1} \left[(\lambda T)^{-1} P \hat{\mathbf{m}}_i + \bar{N}^{-1} \mathbf{d}\right] \\ \hat{\mathbf{m}}_{i+1} = \left(P^\dagger (\lambda T)^{-1} P\right)^{-1} P^\dagger (\lambda T)^{-1} \mathbf{t}_i \end{cases} \tag{20}$$

This is fixed point iteration form of messenger field method. It's solving map making equation Eq.(4) with preconditioner $P^\dagger (\lambda T)^{-1} P$. If we substitute $T = \tau I$, then messenger field method is equivalent solving

$$\tau \left(P^\dagger P\right)^{-1} P^\dagger \left(\tau I + \frac{1}{\lambda} \bar{N}\right)^{-1} P \hat{\mathbf{m}} = \tau \left(P^\dagger P\right)^{-1} P^\dagger \left(\tau I + \frac{1}{\lambda} \bar{N}\right)^{-1} \mathbf{d} \tag{21}$$

since multiplying a constant won't change the condition number, it's equivalent to solve map making equation with perturbation parameter $\eta = 1/\lambda$ and simple preconditioner.

3. NUMERICAL SIMULATIONS

To compare these algorithms, we need to do some simple simulation of scanning process, and generate time ordered data from random sky signal. Our sky is a small rectangular area, with two orthogonal directions x and y, both with range from -1° to $+1^\circ$. The electromagnetic signal is described as four stokes parameters $(S_0, S_1, S_2, S_3) = (I, Q, U, V)$. We model the overall electromagnetic signal is created by some normal distributed sources in the sky, with intensity $I_i(x, y) = A_i \exp\left(-\frac{1}{2} \frac{(x-x_i)^2 + (y-y_i)^2}{\sigma_i^2}\right)$, for each source centered at (x_i, y_i). In our simulation, $A_i \sim \text{Unif}(-100, 100)$, $\sigma_i \sim \text{Unif}(0.05^\circ, 0.2^\circ)$ and the center of each source $x_i, y_i \sim \text{Unif}(-1^\circ, +1^\circ)$. Every source has its degree of polarization $p_i \sim \text{Unif}(0, 1)$ and polarization angle $\psi_i \sim \text{Unif}(0, \pi)$. Here we ignored angle χ_i, because our detectors won't be sensitive to circular polarization. Finally, the stokes parameters over sky is given by $S_0(x, y) = \sum_i I_i(x, y)$, $S_1(x, y) = \sum_i I_i(x, y) p_i \cos(2\psi_i)$, $S_2(x, y) = \sum_i I_i(x, y) p_i \sin(2\psi_i)$. Again, we ignored S_3, because it describes circular polarization.

For the scanning process, our single telescope contains nine detectors, each has different sensitivity to polarization S_1 and S_2. It scans the sky with a raster scanning pattern and scanning frequency $f_{\text{scan}} = 0.1$ Hz sampling frequency $f_{\text{sample}} = 100$ Hz. The telescope scans the sky horizontally and then vertically, and then digitizes the position (x, y) into 512×512 pixel. This gives noiseless signal $\mathbf{s}$.

The noise power spectrum is given by

$$P(f) = \sigma^2 \left(1 + \frac{f_{\text{knee}}^\alpha + f_{\text{apo}}^\alpha}{f^\alpha + f_{\text{apo}}^\alpha}\right) \tag{22}$$

Here we fixed $\sigma^2 = 10\ \mu\text{K}^2$, $\alpha = 2$ and $f_{\text{knee}} = 10$ Hz, and change f_{apo} to compare the performance under different noise models. Note that as $f_{\text{apo}} \to 0$, $P(f) \to \sigma^2 \left(1 + (f/f_{\text{knee}})^{-1}\right)$, it becomes a $1/f$ noise model. The noise covariance matrix

$$N_{ff'} = P(f) \frac{\delta_{ff'}}{\Delta_f} \tag{23}$$

is a diagonal matrix in frequency space, where Δ_f is equal to reciprocal of total scanning time T.

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

4. RESULTS

First let's compare the results with vanilla conjugate gradient method with simple preconditioner $P^\dagger P$. The results are showed in Figure (1) for different kinds of noise power spectra. Here note that χ^2 in Figure (**??**) is calculated based on Eq.(3)

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

not $\chi^2(\mathbf{m}, \eta)$ in Eq.(8). The χ^2_{min} is calculated from perturbative conjugate gradient method with more intermediate η values, and more iterations after $\eta = 1$.

As we can see in Figure(**??**), if the condition number of noise covariance matrix $\kappa(N)$ is small, and the noise is almost white noise, the performance between different these two methods is small. The vanilla conjugate gradient method converge faster, because its perturbation parameter $\eta_i = \{0, 1, 1, \cdots\}$, however for the perturbation method its η value will slowly reach 1 in first few iterations as we can see in Figure(**??**).

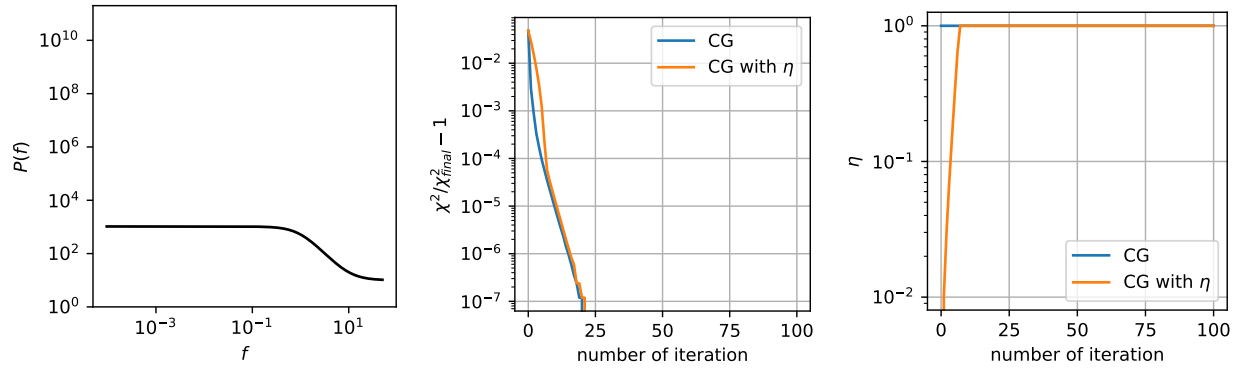

Figure 1. The left graph shows the noise power spectrum Eq.(22) with $f_{\rm apo} \approx 0.99$ and $\kappa(N) = 10^2$. The center one shows the $\chi^2(\mathbf{m})/\chi^2_{\rm final} - 1$, with $\chi^2(\mathbf{m})$ calculated based on Eq.(3). The right one shows the η value for each iteration. For vanilla conjugate gradient method η always equal to 1, so it's a horizontal line at $\eta = 1$.

Notice that as we increase $\kappa(N)$, or equivalently decrease $f_{\rm apo}$, the perturbation parameter η starts showing its benefits, as showed in Figure(2) and Figure(3). It outperforms the vanilla conjugate gradient method when $f_{\rm apo} \approx 0$ and the noise power spectrum becomes the $1/f$ noise model, which usually is the intrinsic noise of instruments**?**.

In the conjugate gradient method with messenger cooling parameter λ, the number of cooling parameters we need is an extra free parameter. After the number of λ is determined, we construct a geometric series with fixed initial and final value, which uses `logspace` function in `numpy`. Since I show in Eq.(21) that the messenger field cooling parameter λ is equivalent to $1/\eta$, I use η for further analysis.

Now let us compare the performance difference between choosing η parameters based on Eq.(18) and fixing number of η parameters n_η manually. Here 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), (5), and (6).

When $\kappa(N)$ is small, and Eq.(18) tells us that only a few η parameters are good enough, see Figure(**??**). If unfortunately we choose n_η being large value, like 15 or 30, then it will ends up converge slowly, because it needs at least 15 or 30 iterations to reach $\eta = 1$, at least one iteration per η level.

On the other hand if $\kappa(N)$ is very large and the power spectrum is $1/f$ noise, we need more η parameters. If n_η is too small, for example $n_\eta = 5$ in Figure(**??**), it may be better than the vanilla conjugate gradient method, but it is still far from optimal.

5. POSSIBLE IMPROVEMENTS

As you may have noticed in Figure(5) and Figure(6), the perturbation parameter based on Eq.(18) is more than needed, especially for $1/f$ noise case. From Figure(**??**) we know that Eq.(18) gives us $n_\eta \approx 40$, however based on χ^2 result in Figure(**??**), we notice that $n_\eta \approx 30$ or even $n_\eta \approx 15$ is good enough. Also, for the nearly-white-noise case, we could certainly choose $n_\eta = 1$ such that $\eta_1 = 1$ which corresponds to vanilla conjugate gradient method, based on χ^2 result in Figure(**??**). However Eq.(18) gives us $n_\eta \approx 6$, see Figure(**??**), even though it does not make the final χ^2 result much different at the end.

Is it possible to further improve the analysis, such that it produces smaller n_η? Let's examine how we get η_i series. Remember that we determine $\delta\eta$ value based on the upper bound of $-\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$, in Eq.(14). Here I rewrite it in a simplified form

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)} = \delta\eta\frac{\hat{\mathbf{r}}_\eta^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}\hat{\mathbf{r}}_\eta}{\hat{\mathbf{r}}_\eta^\dagger N_\eta^{-1}\hat{\mathbf{r}}_\eta} \leq \frac{\delta\eta}{\eta + \frac{\tau}{\max(N_f)-\tau}} \tag{24}$$

with $\mathbf{r}_\eta = \mathbf{d} - P\hat{\mathbf{m}}(\eta) = \left[1 - P\left(P^\dagger N_\eta^{-1}P\right)^{-1}P^\dagger N_\eta^{-1}\right]\mathbf{d} \equiv \mathcal{P}_\eta\mathbf{d}$. We treated $\mathbf{r}_\eta$ as an arbitrary vector in frequency domain, since we don't know how to calculate $\mathcal{P}_\eta$ for $\eta \neq 0$, and it's hard to analyze the projection matrix $\mathcal{P}_\eta$ in frequency space, as it contains $\left(P^\dagger N_\eta^{-1}P\right)^{-1}$. Note that we have to determine all of η value before calculation, because we don't want to keep the time ordered data in system RAM, so we need to somehow analytically analyze $\mathcal{P}_\eta$, and its behavior in frequency space.

Unless $\mathbf{r}_\eta$ almost only has large noise modes, $\frac{\rm d}{{\rm d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ won't get close to the upper bound $1/\left(\eta + \frac{\tau}{\max(N_f)-\tau}\right)$. Based on the analysis in Section(2.2.2), for small η the estimated map $\hat{\mathbf{m}}(\eta)$ does not only focusing on minimizing error $\mathbf{r}_\eta$ at low

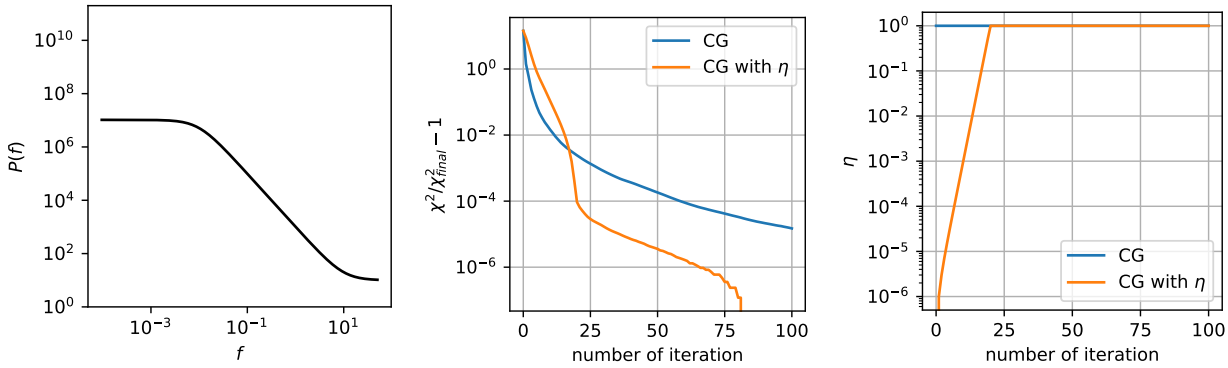

Figure 2. The figure shows results for $f_{\rm apo} \approx 9.8 \times 10^{-3}$ and $\kappa(N) = 10^6$.

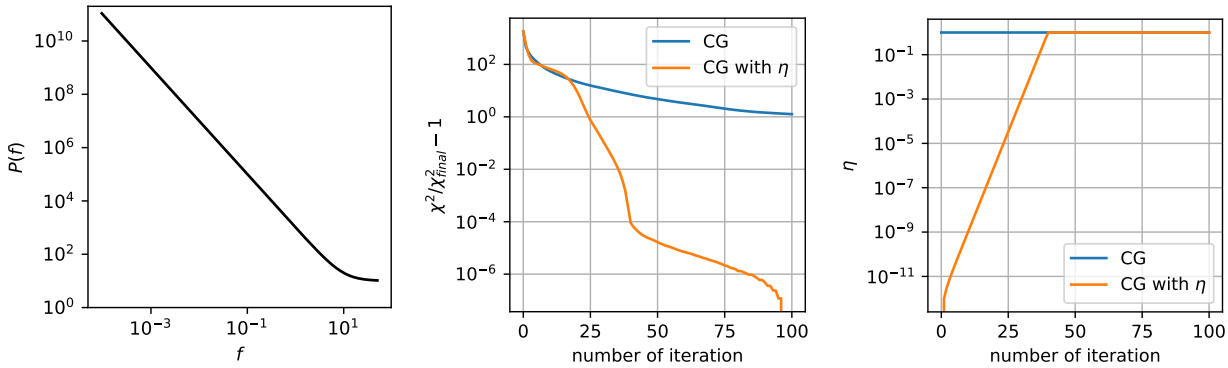

Figure 3. The figure shows results for $f_{\rm apo} \approx 9.8 \times 10^{-6}$ and $\kappa(N) = 10^{12}$.

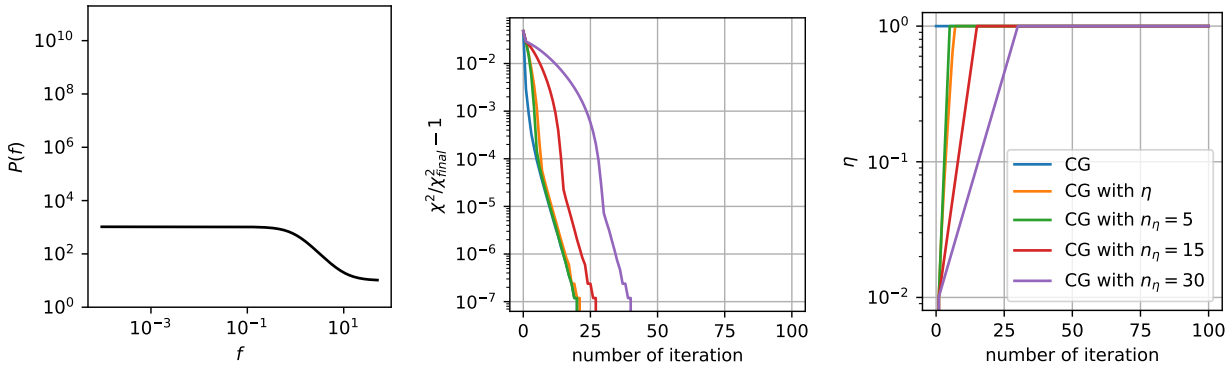

Figure 4. Same as Figure(1) with extra manually chosen n_η results.

noise region. So we would expect that there would be a fair amount of low noise modes contribution in $\mathbf{r}_\eta$ especially for the first few η values. Which means if we could somehow know the frequency distribution of $\mathbf{r}_\eta$, we could tighten the boundary of $\frac{\rm d}{{\rm d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$, and get larger $\delta\eta$ value. This should make η goes to 1 faster, and yields the fewer η parameters we need.

Also notice that the η values determined from Eq.(18)

$$\eta_i = \min\left\{1,\ \frac{\tau}{\max(\bar{N}_f)}\left(2^i - 1\right)\right\} \tag{18}$$

are not dependent on any scanning information, it only depends on noise power spectrum $P(f)$, or noise covariance matrix N. Figure(7) and Figure(8) show two examples with same parameters as in Figure(6) except scanning frequency $f_{\rm scan}$, in Figure(7) it scans very slow and in Figure(8) it's very fast. In these two cases our η values

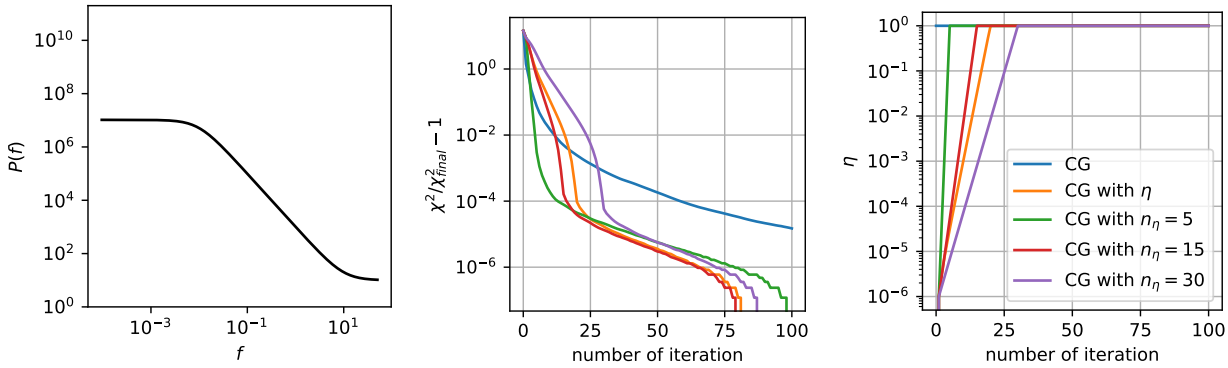

Figure 5. Same as Figure(2) with extra manually chosen n_η results.

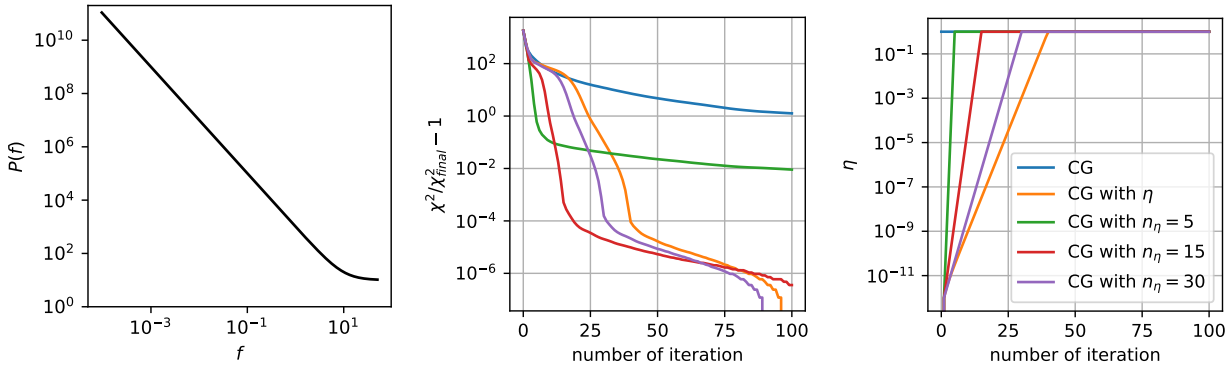

Figure 6. Same as Figure(3) with extra manually chosen n_η results.

based on Eq.(18) are better than manually selected values. Based on these two results we know, the η values should somehow depends on scanning scheme. Again that's because when we determine the upper bound of $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ we treat $\mathbf{r}_\eta = \mathbf{d} - P\hat{\mathbf{m}} = \mathcal{P}_\eta\mathbf{d}$ as an arbitrary vector, such that we lose all information related to scanning scheme in the pointing matrix P.

6. CONCLUSION

Here we discussed a method to solve map making equation Eq.(4)

$$\hat{\mathbf{m}} = \left(P^\dagger N P\right)^{-1} P^\dagger N^{-1}\mathbf{d} \tag{4}$$

by separating noise covariance matrix N into two parts, white noise part τI and the remaining noise $\bar{N}$. Then we could think $\bar{N}$ as a perturbation added to white noise, by introducing a parameter η, as η change from 0 to 1, we gradually add this non white noise in to system.

The η values can be predetermined analytically. This property is very important, because we don't want to keep entire time ordered data in system RAM. If these η values can be determined before calculation, then we only need to keep several map sized object, which is much smaller than timed ordered data. Also we showed that this method has same computational cost as vanilla conjugate gradient method but performs better when the condition number of noise covariance matrix $\kappa(N)$ is large, especially in $1/f$ noise case. The only extra free parameter added is to determine whether the error at current step $\mathbf{r}(\eta_i) = ||\mathbf{b}(\eta_i) - A(\eta_i)\mathbf{m}||$ is small enough such that we change advance to next value η_{i+1}.

The perturbation parameter η get from Eq.(18) are not perfect. Since it only takes in to account the noise information in N, but ignored all scanning information contained in pointing matrix P, because we are unable to analyze the structure of $\mathbf{r}_\eta = \mathbf{d} - P\hat{\mathbf{m}}(\eta) = \mathcal{P}_\eta\mathbf{d}$ in frequency space.

The analysis of η value also explains why cooling parameters $\lambda = 1/\eta$ in messenger field are chosen to be geometric series or `logspace ?`.

All of the calculation are using simple preconditioner $P^\dagger P$, but the entire analysis is independent of preconditioner. Using better preconditioners, it would also have improvements.

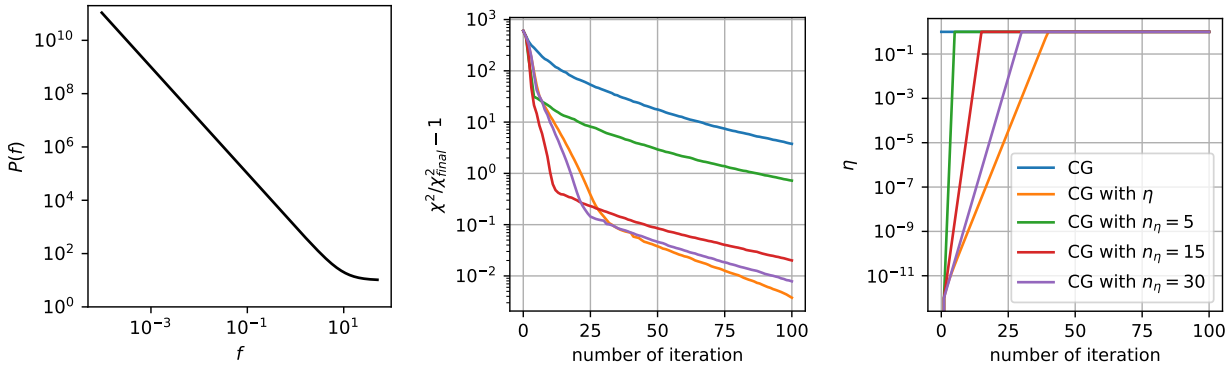

Figure 7. In this case all frequencies are the same as Figure(6) except $f_{\rm scan} = 0.001$.

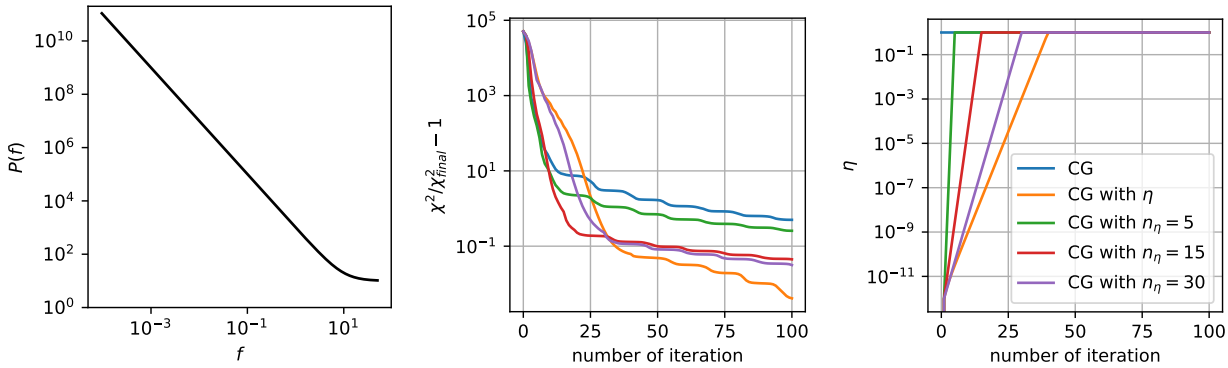

Figure 8. In this case all frequencies are the same as Figure(6) except $f_{\rm scan} = 10$.

APPENDIX

REFERENCES