Perturbative Approach Solve Map-Making Equation

Bai-Qiang Qiang

March 15, 2021

1 Introduction

Cosmic microwave background is an electromagnetic radiation coming from early stage of our universe. Based on hot Big Bang model, before the recombination epoch, the photons were tightly coupled with free electrons and protons via Thomson scattering. As universe expanding and cooling down, free electrons and protons combined into neutral hydrogen atom, this process is called recombination in cosmology. Shortly after this epoch, the photon could propagate freely would not be scattered by charged particle. Now we received those photon produced at that time and called it cosmic microwave background radiation. Studying these photons coming from early universe could help us constrain theoretical model and cosmological constants [2]. The next generation CMB observations will have much higher resolution and generates more data. So we need an efficient way to processing data. One of the these processing is map making, which gives an estimated map based on observation data.

Recently Elsner and Wandelt[1] introduced a new method called messenger field to solve Wiener filer, and then this technique was being applied to map making equation[3]. It's been shown that this messenger field method is equivalent to applying a preconditioner to original problem and introduced an extra cooling parameter λ, but whether this cooling parameter will boost performance compare to conjugate gradient method is still controversial[4]. Here I'm gonna give a detailed analysis regarding 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 $\mathbf{n}$. Pointing matrix P acting on the map gives the signal of 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's not zero, we can always subtract its mean value to make it zero. And noise covariance matrix could be written as $N = \langle \mathbf{n}\mathbf{n}^\dagger \rangle$.

2 Map Making Setup

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*, its variance N are different for various frequencies, usually detectors have $1/f$ noise pattern [6]. 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

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

and we could define

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

therefore the estimated map $\hat{\mathbf{m}}$ is the one minimize $\chi^2(\mathbf{m})$. To find out expression for $\hat{\mathbf{m}}$, we first take derivative with respect to vector $\mathbf{m}$

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{m}} \chi^2(\mathbf{m}) &= \frac{\partial}{\partial \mathbf{m}} (\mathbf{d} - P\mathbf{m})^\dagger N^{-1} (\mathbf{d} - P\mathbf{m}) \\
&= \frac{\partial}{\partial \mathbf{m}} \Big(\mathbf{d}^\dagger N^{-1} \mathbf{d} - \mathbf{d}^\dagger N^{-1} P\mathbf{m} - \mathbf{m}^\dagger P^\dagger N^{-1} \mathbf{d} + \mathbf{m}^\dagger P^\dagger N^{-1} P\mathbf{m} \Big) \\
&= -2P^\dagger N^{-1} \mathbf{d} + 2P^\dagger N^{-1} P\mathbf{m}
\end{aligned} \tag{4}$$

then set it equals to zero $\frac{\partial}{\partial \mathbf{m}} \chi^2(\hat{\mathbf{m}}) = 0$, we get the *map making equation*

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

This is also called COBE method for map making.

3 Some Nice Properties

3.1 Unbiased linear estimator

Linear estimator means $\hat{\mathbf{m}}$ could be written as $\hat{\mathbf{m}} = W\mathbf{d}$, it's linear combination of $\mathbf{d}$. We say the estimator is unbiased if

$$\begin{aligned}
&\langle \hat{\mathbf{m}} \rangle = m \\
\Rightarrow\ &\langle W\mathbf{d} \rangle = m \\
\Rightarrow\ &\langle W(Pm + \mathbf{n}) \rangle = m \\
\Rightarrow\ &WP = I
\end{aligned} \tag{6}$$

At last step we used the property $\langle \mathbf{n} \rangle = 0$. And for the generalized least square estimator matrix $W = \left(P^\dagger N P\right)^{-1} P^\dagger N^{-1}$, which satisfy the condition $WP = I$. Therefore $\hat{\mathbf{m}}$ is unbiased estimated map.

3.2 Minimize variance $\mathbf{Var}[\hat{\mathbf{m}}_i]$ under constrain of unbiased linear estimators

The covariance of the estimator $\hat{\mathbf{m}} = W\mathbf{d}$ is

$$\begin{aligned} \mathrm{Cov}[\hat{\mathbf{m}}] &= \mathrm{Cov}[W\mathbf{d}] \\ &= \mathrm{Cov}[WP\mathbf{m} + W\mathbf{n}] \\ &= \mathrm{Cov}[W\mathbf{n}] \\ &= WNW^\dagger \end{aligned} \tag{7}$$

Here we use a trick [5], if consider the matrix $W = W_{GLS} + W'$ where $W_{GLS} = \left(P^\dagger N P\right)^{-1} P^\dagger N^{-1}$ is the matrix for GLS estimation, and in order to satisfy the condition $WP = I$, we should have $W'P = 0$. Then the covariance matrix

$$\begin{aligned} \mathrm{Cov}[\hat{\mathbf{m}}] &= W_{GLS} N W_{GLS}^\dagger + W' N W'^\dagger + W_{GLS} N W'^\dagger + W' N W_{GLS}^\dagger \\ &= \Big(P^\dagger N P\Big)^{-1} + W' N W'^\dagger + \Big(P^\dagger N P\Big)^{-1} P^\dagger W'^\dagger + W' P \Big(P^\dagger N P\Big)^{-1} \\ &= \Big(P^\dagger N P\Big)^{-1} + W' N W'^\dagger \end{aligned} \tag{8}$$

where the last line used condition $W'P = 0$.

The variance $\mathrm{Var}[\hat{\mathbf{m}}_i]$ is diagonal elements of covariance matrix $\mathrm{Cov}[\hat{\mathbf{m}}]$

$$\begin{aligned} \mathrm{Var}[\hat{\mathbf{m}}_i] &= \left\{ \Big(P^\dagger N P\Big)^{-1} \right\}_{ii} + \left\{ W' N W'^\dagger \right\}_{ii} \\ &= \left\{ \Big(P^\dagger N P\Big)^{-1} \right\}_{ii} + W'_{i,:} N W'^\dagger_{i,:} \end{aligned} \tag{9}$$

where $W'_{i,:}$ is the i^{th} row vector of W'. Since the noise covariance matrix N is positive semi-definite matrix, therefore $W'_{i,:} N W'^\dagger_{i,:} \geq 0$. If $W' = 0$ Then we have $W = W_{GLS}$, variance $\mathrm{Var}[\hat{\mathbf{m}}_i]$ would have its minimum variance.

3.3 Minimize mean square error under constrain of unbiased linear estimator

The error is defined as the difference between estimated map and real one

$$\begin{aligned}\varepsilon &\equiv \hat{\mathbf{m}} - \mathbf{m} \\ &= W\mathbf{d} - \mathbf{m} \\ &= (WP - I)\mathbf{m} + W\mathbf{n} \\ &= W\mathbf{n}\end{aligned} \tag{10}$$

where the last line used relation $WP = I$ for unbiased estimator W.

Now we need to minimize mean square error

$$\begin{aligned}\left\langle \varepsilon^\dagger \varepsilon \right\rangle &= \left\langle \mathrm{Tr}\Big(\varepsilon\varepsilon^\dagger\Big) \right\rangle \\ &= \left\langle \mathrm{Tr}\Big(W\mathbf{n}\mathbf{n}^\dagger W^\dagger\Big) \right\rangle \\ &= \mathrm{Tr}\Big(WNW^\dagger\Big) \\ &= \mathrm{Tr}(\mathrm{Cov}[\hat{\mathbf{m}}]) \\ &= \sum_i \mathrm{Var}[\hat{\mathbf{m}}_i]\end{aligned} \tag{11}$$

the second line we used property $\varepsilon^\dagger\varepsilon$ is a scalar, so $\varepsilon^\dagger\varepsilon = \mathrm{Tr}\big(\varepsilon^\dagger\varepsilon\big) = \mathrm{Tr}\big(\varepsilon\varepsilon^\dagger\big)$, and the fourth line is because trace is a linear operation and $\left\langle \mathbf{n}\mathbf{n}^\dagger \right\rangle = N$, the fifth line comes from Eq.(7). In Section 3.2 we have shown that the generalized least square matrix W_{GLS} minimize $\mathrm{Var}[\hat{\mathbf{m}}_i]$ for each i, therefore it also minimize the mean square error $\left\langle \varepsilon^\dagger\varepsilon \right\rangle = \sum_i \mathrm{Var}[\hat{\mathbf{m}}_i]$.

3.4 Maximum likelihood estimator

Previous properties does not depends on the noise distribution, if we assume that the noise is multivariate normal distributed $\mathbf{n} \sim \mathcal{N}(0, N)$ with mean 0 covariance N. Its likelihood function will be

$$L(\mathbf{d}; \mathbf{m}) = \frac{1}{\sqrt{(2\pi)^n |N|}} \exp\left(-\frac{1}{2}(\mathbf{d} - P\mathbf{m})^\dagger N (\mathbf{d} - P\mathbf{m})\right) \tag{12}$$

and Log-likelihood

$$\log\left(L(\mathbf{d}; \mathbf{m})\right) = -\frac{1}{2}(\mathbf{d} - P\mathbf{m})^\dagger N (\mathbf{d} - P\mathbf{m}) + \mathrm{cont.} \tag{13}$$

maximizing this log-likelihood function with respect to $\mathbf{m}$, is equivalent to minimize $\chi^2(\mathbf{m}) = (\mathbf{d} - P\mathbf{m})^\dagger N (\mathbf{d} - P\mathbf{m})$, which is $\hat{\mathbf{m}}$.

4 Solve Map Making Equation

The map making equation Eq.(5) derived from Generalized Least Square estimation,

$$\left(P^\dagger N^{-1} P\right)\hat{\mathbf{m}} = P^\dagger N^{-1}\mathbf{d} \tag{14}$$

If we define $A = P^\dagger N^{-1} P$ and $b = P^\dagger N^{-1} d$, 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. It's 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 $N^{-1}P\hat{\mathbf{m}}$. Which means it could be solved by conjugate gradient method.

4.1 Preconditioner

To improve the performance of conjugate gradient method, we could apply 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 condition number of original problem, such that conjugate gradient method would 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 average over each pixel of map $\mathbf{m}$.

For conjugate gradient method, we need a initial guess map $\hat{\mathbf{m}}_0$. Sure we can use zero vector $\hat{\mathbf{m}}_0 = \mathbf{0}$ as initial guess, but simple binned map $\hat{\mathbf{m}}_0 = \left(P^\dagger P\right)^{-1} P^\dagger\mathbf{d}$ would be a better choice, which is a the solution for white noise case $N \propto I$. (Papež et al. 2018[4]) 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 size 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 conjugate gradient method with simple preconditioner $M = P^\dagger P$, we've got all we need. Next we only need to use conjugate gradient algorithm solve the problem.

4.2 Parameterized Conjugate Gradient Method

We could also parameterize map making equation Eq.(14), and it may improve performance in some cases. The idea is that map making equation Eq.(14) 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. So what if we parameterize noise covariance matrix N with a parameter η, such that initially $\eta = \eta_i$, $N(\eta_i) \propto I$ and final $\eta = \eta_f$ and $N(\eta_f) \propto N$, such that the final solution is what we want. We expect 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.(14), we are solving

$$\left(P^\dagger N(\eta)^{-1} P\right) \hat{\mathbf{m}}(\eta) = P^\dagger N(\eta)^{-1} \mathbf{d} \tag{15}$$

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 a trouble maker, we could think of it as a perturbation term, which add 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.(14).

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.(15) then becomes

$$\left(P^\dagger (\tau I + \eta \bar{N})^{-1} P\right) \hat{\mathbf{m}}(\eta) = P^\dagger \left(\tau I + \eta \bar{N}\right)^{-1} \mathbf{d} \tag{16}$$

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 $\hat{\mathbf{m}}(\eta_0) = \mathbf{m}_0 = \left(P^\dagger P\right)^{-1} P^\dagger \mathbf{d}$.

4.2.1 Choosing perturbation parameters η

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 at 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 object in RAM, instead of entire time ordered data $\mathbf{d}$.

First let's 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 denote $N(\eta)$. The estimated map $\hat{\mathbf{m}}(\eta) = \left(P^\dagger N_\eta^{-1} P\right)^{-1} P^\dagger N_\eta^{-1} \mathbf{d}$ minimize

$$\chi^2(\mathbf{m}, \eta) = (\mathbf{d} - P\mathbf{m})^\dagger N_\eta^{-1} (\mathbf{d} - P\mathbf{m}) \tag{17}$$

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

$$\begin{aligned}\chi^2(\hat{\mathbf{m}}(\eta),\eta) &= \big(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\big)^\dagger N_\eta^{-1}\big(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\big)\\ &= \mathbf{d}^\dagger\left[N_\eta^{-1}-N_\eta^{-1}P\Big[P^\dagger N_\eta^{-1}P\Big]^{-1}P^\dagger N_\eta^{-1}\right]\mathbf{d}\end{aligned} \tag{18}$$

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

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) &= \frac{\mathrm{d}}{\mathrm{d}\eta}\Big(\mathbf{d}^\dagger N_\eta^{-1}\mathbf{d}\Big)-\frac{\mathrm{d}}{\mathrm{d}\eta}\left(\mathbf{d}^\dagger N_\eta^{-1}P\Big(P^\dagger N_\eta P\Big)^{-1}P^\dagger N_\eta^{-1}\mathbf{d}\right)\\ &= \mathbf{d}^\dagger N_\eta^{-1}\big[-\bar{N}+\bar{N}N_\eta^{-1}P\Big(P^\dagger N_\eta P\Big)^{-1}P^\dagger\\ &\qquad -P\Big(P^\dagger N_\eta^{-1}P\Big)^{-1}P^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}P\Big(P^\dagger N_\eta^{-1}P\Big)^{-1}P^\dagger\\ &\qquad +P\Big(P^\dagger N_\eta^{-1}P\Big)^{-1}P^\dagger N_\eta^{-1}\bar{N}\big]N_\eta^{-1}\mathbf{d}\end{aligned} \tag{19}$$

Simplify this expression with identity $\hat{\mathbf{m}}=\big(P^\dagger N_\eta^{-1}P\big)^{-1}P^\dagger N_\eta^{-1}\mathbf{d}$, and yields

$$\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) = -(\mathbf{d}-P\hat{\mathbf{m}}(\eta))^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta)) \tag{20}$$

also notice that $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) = \frac{\partial}{\partial\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)$, 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 analysis, let's assume that the noise covariance matrix $N=\big\langle\mathbf{n}\mathbf{n}^\dagger\big\rangle$ is diagonal under frequency domain. Therefore $\bar{N}$ and N_η are also diagonal in 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.(20), 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 decrease as η changes from 0 to 1.

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

$$\begin{aligned}-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)} &= -\delta\eta\frac{1}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)}\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)\\ &= \delta\eta\frac{(\mathbf{d}-P\hat{\mathbf{m}}(\eta))^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta))}{\big(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\big)^\dagger N_\eta^{-1}\big(\mathbf{d}-P\hat{\mathbf{m}}(\eta)\big)}\end{aligned} \tag{21}$$

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

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(0),0)}{\chi^2(\hat{\mathbf{m}}(0),0)} = \delta\eta\frac{1}{\tau}\frac{(\mathbf{d}-P\hat{\mathbf{m}}(0))^\dagger\bar{N}(\mathbf{d}-P\hat{\mathbf{m}}(0))}{\big(\mathbf{d}-P\hat{\mathbf{m}}(0)\big)^\dagger\big(\mathbf{d}-P\hat{\mathbf{m}}(0)\big)}$$
$$\leq \frac{\delta\eta}{\tau}\max(\bar{N}_f) \tag{22}$$

where at first line we used the property $N_{\eta=0} = \tau I$, and second line because positive semi-definite $\bar{N}$ is diagonal in frequency domain its maximum eigenvalue is $\max(\bar{N}_f)$. To prove this, notice that matrix $\bar{N}$ is diagonalized in frequency space with eigenvalues $\bar{N}_f \geq 0$ and corresponding eigenvector $\mathbf{e}_f$ (these eigenvectors form a complete orthogonal basis), any vector could be decomposed into these frequency basis $\mathbf{v} = \sum_f \alpha_f \mathbf{e}_f$, therefore we have $\frac{\mathbf{v}^\dagger\bar{N}\mathbf{v}}{\mathbf{v}^\dagger\mathbf{v}} = \frac{\sum_f \alpha_f^2\bar{N}_f}{\sum_f \alpha_f^2} \leq \max(\bar{N}_f)$

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 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 won't going too fast, we don't want $\left|\delta\chi^2(\hat{\mathbf{m}}(0),0)/\chi^2(\hat{\mathbf{m}}(0),0)\right|$ exceeds 1. So we could set upper bound $\delta\eta\max(\bar{N}_f)/\tau = 1$ and set

$$\eta_1 = \frac{\tau}{\max(\bar{N}_f)} = \frac{\min(N_f)}{\max(N_f)-\min(N_f)} \tag{23}$$

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 other parameters η_m with $m > 1$? We could use similar analysis, suppose $\eta_{m+1} = \eta_m + \delta\eta_m$ with small $\delta\eta_m$, and the relative decrease

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)}{\chi^2(\hat{\mathbf{m}}(\eta_m),\eta_m)} = \delta\eta_m\frac{(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m))^\dagger N_{\eta_m}^{-1}\bar{N}N_{\eta_m}^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m))}{\big(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m)\big)^\dagger N_{\eta_m}^{-1}\big(\mathbf{d}-P\hat{\mathbf{m}}(\eta_m)\big)}$$
$$\leq \delta\eta_m\ \max\left(\frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\right) \tag{24}$$

The upper bound at 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 $N_{\eta_m}^{-1}\bar{N}N_{\eta_m}^{-1}$ is $\lambda_f = \bar{N}_f(\tau+\eta_m\bar{N}_f)^{-2}$, and the eigenvalue for matrix $N_{\eta_m}^{-1}$ is $\gamma_f = (\tau+\eta_m\bar{N}_f)^{-1}$. Their eigenvalues are related by $\lambda_f = \frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\gamma_f$. For any vector $\mathbf{v} = \sum_f \alpha_f\mathbf{e}_f$, we have $\frac{\mathbf{v}^\dagger N_{\eta_m}^{-1}\bar{N}N_{\eta_m}^{-1}\mathbf{v}}{\mathbf{v}^\dagger N_{\eta_m}^{-1}\mathbf{v}} = \frac{\sum_f \alpha_f^2\lambda_f}{\sum_f \alpha_f^2\gamma_f} = \frac{\sum_f \alpha_f^2\gamma_f\bar{N}_f/(\tau+\eta_m\bar{N}_f)}{\sum_f \alpha_f^2\gamma_f} \leq \max\left(\frac{\bar{N}_f}{\tau+\eta_m\bar{N}_f}\right)$

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

$$\delta\eta_m = \min\left(\frac{\tau+\eta_m\bar{N}_f}{\bar{N}_f}\right) = \eta_m + \frac{\tau}{\max(\bar{N}_f)} \tag{25}$$

Therefore

$$\eta_{m+1} = \eta_m + \delta\eta_m = 2\eta_m + \frac{\tau}{\max(\bar{N}_f)} \tag{26}$$

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

$$\eta_m + \frac{\tau}{\max(\bar{N}_f)} = \left(\eta_1 + \frac{\tau}{\max(\bar{N}_f)}\right)2^{m-1} = \frac{\tau}{\max(\bar{N}_f)}2^m \tag{27}$$

Note that $m = 0$ and $\eta_0 = 0$ also satisfy this expression and we've got final expression for all η_i

$$\eta_i = \min\left\{1,\ \frac{\tau}{\max(\bar{N}_f)}\left(2^i - 1\right)\right\} \tag{28}$$

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

Ta-DA! Eq.(**??**) not only tell us how to choose parameters η_i, it also tell us 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 $\eta_1 = \eta_2 = \cdots = 1$.

4.2.2 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 minimize $\chi^2(\mathbf{m}) = (\mathbf{d}-P\mathbf{m})^\dagger N^{-1}(\mathbf{d}-P\mathbf{m})$, 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 $\mathbf{r} \equiv \mathbf{d} - P\mathbf{m}$ at low noise part.

*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.(28) is geometric series, only a few η_m values won't satisfy this condition.

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 $\eta = 0$, $N_{\eta=0}^{-1} \propto I$ 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 vanilla conjugate gradient method, then 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 increasing, it gradually shifting 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 using Eq.(20)

$$\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta) = -(\mathbf{d}-P\hat{\mathbf{m}}(\eta))^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}(\mathbf{d}-P\hat{\mathbf{m}}(\eta)) \tag{20}$$

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 vanilla conjugate gradient method only focusing on low noise part and it may have difficulty at these regions.

4.3 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.(15)

$$\left(P^\dagger N(\eta)^{-1}P\right)\hat{\mathbf{m}}(\eta) = P^\dagger N(\eta)^{-1}\mathbf{d} \tag{15}$$

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.

4.4 Messenger Field Method

Messenger field method is a fixed point iterative solver introduced by Elsner and Wandelt (2013) [1] to solve Wiener filter. Later on Huffenberger and Næss (2018) [3] applied this method to map-making problem, and showed that in some cases messenger field is better than conjugate gradient with simple preconditioner. And Papež el al.(2018) [4] proved that messenger field is equivalent to apply a preconditioner to map making equation Eq.(5), 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 $\lambda = 1/\eta$.

Before introducing messenger field method, let's first prove one identity

$$
\begin{aligned}
&\left(P^\dagger T^{-1} P\right)^{-1} P^\dagger T^{-1} \left(T^{-1} + \bar{N}^{-1}\right)^{-1} T^{-1} P \\
&= \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger T^{-1} \left(I + T\bar{N}^{-1}\right)^{-1} P \\
&= \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger T^{-1} \left(I - T\bar{N}^{-1} + T\bar{N}^{-1} T\bar{N}^{-1} - \cdots\right) P \\
&= I - \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger \cancel{T^{-1}} T\bar{N}^{-1} \left(I - T\bar{N}^{-1} + T\bar{N}^{-1} T\bar{N}^{-1} - \cdots\right) P \\
&= I - \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger \bar{N}^{-1} \left(I + T\bar{N}^{-1}\right)^{-1} P \\
&= I - \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger \left(\bar{N} + T\right)^{-1} P \\
&= I - \left(P^\dagger T^{-1} P\right)^{-1} P^\dagger N^{-1} P
\end{aligned}
\tag{29}
$$

where at third and fifth line we used expansion $(I + A)^{-1} = I - A + A^2 - \cdots$

After apply preconditioner $P^\dagger T^{-1}P$ to map making equation Eq.(5), we get:

$$\begin{aligned}
&\Big(P^\dagger T^{-1}P\Big)^{-1}\Big(P^\dagger N^{-1}P\Big)\hat{\mathbf{m}} = \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger N^{-1}\mathbf{d} \\
\Rightarrow &\hat{\mathbf{m}} - \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger T^{-1}\big(T^{-1}+\bar{N}^{-1}\big)^{-1}T^{-1}P\hat{\mathbf{m}} = \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger N^{-1}\mathbf{d} \\
\Rightarrow &\hat{\mathbf{m}} = \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger T^{-1}\big(T^{-1}+\bar{N}^{-1}\big)^{-1}\big[T^{-1}P\hat{\mathbf{m}} + \big(T^{-1}+\bar{N}^{-1}\big)TN^{-1}\mathbf{d}\big] \\
\Rightarrow &\hat{\mathbf{m}} = \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger T^{-1}\big(T^{-1}+\bar{N}^{-1}\big)^{-1}\big[T^{-1}P\hat{\mathbf{m}} + \big(I+\bar{N}^{-1}T\big)(\bar{N}+T)^{-1}\mathbf{d}\big] \\
\Rightarrow &\hat{\mathbf{m}} = \Big(P^\dagger T^{-1}P\Big)^{-1}P^\dagger T^{-1}\big(T^{-1}+\bar{N}^{-1}\big)^{-1}\big[T^{-1}P\hat{\mathbf{m}} + \bar{N}^{-1}\mathbf{d}\big] \qquad (30)
\end{aligned}$$

where the second line we used identity Eq.(29).

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

$$\begin{cases} \mathbf{t}_i = \big((\lambda T)^{-1}+\bar{N}^{-1}\big)^{-1}\big[(\lambda T)^{-1}P\hat{\mathbf{m}}_i + \bar{N}^{-1}\mathbf{d}\big] \\ \hat{\mathbf{m}}_{i+1} = \Big(P^\dagger(\lambda T)^{-1}P\Big)^{-1}P^\dagger(\lambda T)^{-1}\mathbf{t}_i \end{cases} \qquad (31)$$

This is fixed point iteration form of messenger field method. It's solving map making equation Eq.(5) with preconditioner $P^\dagger(\lambda T)^{-1}P$

$$\Big(P^\dagger(\lambda T)^{-1}P\Big)^{-1}P^\dagger(\bar{N}+\lambda T)^{-1}P\hat{\mathbf{m}} = \Big(P^\dagger(\lambda T)^{-1}P\Big)^{-1}P^\dagger(\bar{N}+\lambda T)^{-1}\mathbf{d} \qquad (32)$$

substitute $T = \tau I$

$$\tau\Big(P^\dagger P\Big)^{-1}P^\dagger\left(\tau I + \frac{1}{\lambda}\bar{N}\right)^{-1}P\hat{\mathbf{m}} = \tau\Big(P^\dagger P\Big)^{-1}P^\dagger\left(\tau I + \frac{1}{\lambda}\bar{N}\right)^{-1}\mathbf{d} \qquad (33)$$

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.

5 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 sky with raster scanning pattern and scanning frequency $f_{\text{scan}} = 0.1$ sampling frequency $f_{\text{sample}} = 100$. The telescope scan the sky horizontally and then vertically, and then digitize position (x, y) into 512×512 pixel. This gives noiseless signal $\mathbf{s}$.

Noise power spectrum is given by

$$P(f) = \sigma^2 \left(1 + \frac{f_{\text{knee}}^\alpha + f_{\text{apo}}^\alpha}{f^\alpha + f_{\text{apo}}^\alpha}\right) \tag{34}$$

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

$$N_{ff'} = P(f)\frac{\delta_{ff'}}{\Delta_f} \tag{35}$$

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.

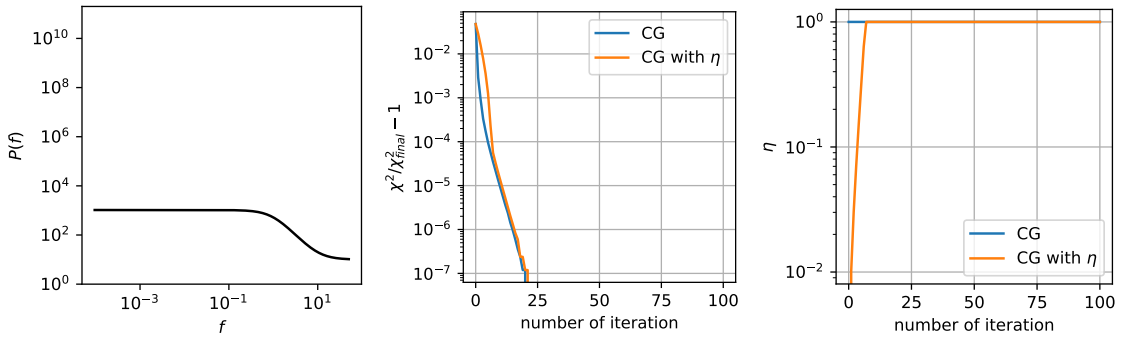
6 Results

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 spectrum. Here note that χ^2 in Figure(1b) 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.(17). The χ^2_{min} is calculated from perturbative conjugate gradient method with more intermediate η values, and more iterations after $\eta = 1$.

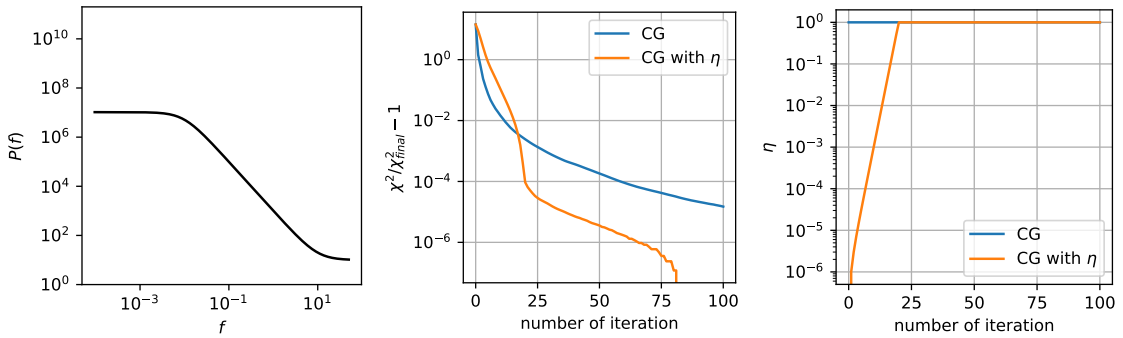
As we can see in Figure(1b), if the condition number of noise covariance matrix $\kappa(N)$ is small, and the noise is almost white noise, the performance between different these two methods is small. Vanilla conjugate gradient method converge faster, because its perturbation parameter $\eta_i = \{0,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



(a) (b) (c)

Figure 1: The left graph shows the noise power spectrum Eq.(34) with $f_{\text{apo}} \approx 0.99$ and $\kappa(N) = 10^2$. The center one shows the $\chi^2(\mathbf{m})/\chi^2_{\text{final}} - 1$, 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 $\eta = 1$.

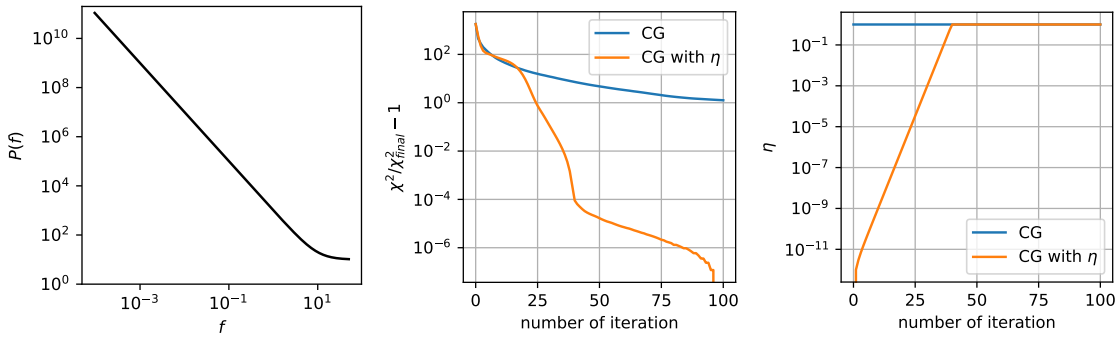
Notice that as we increase $\kappa(N)$, or equivalently decreasing f_{apo}, the perturbation parameter η starts showing its benefits, as showed in Figure(2) and Figure(3). It would outperforms vanilla conjugate gradient method, when $f_{\text{apo}} \approx 0$ and noise power spectrum becomes $1/f$ noise model, which usually is the intrinsic noise of instruments[6].



(a) (b) (c)

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

In 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've showed in Eq.(33) that the messenger field cooling parameter λ is equivalent to $1/\eta$. I would use η for further analysis.



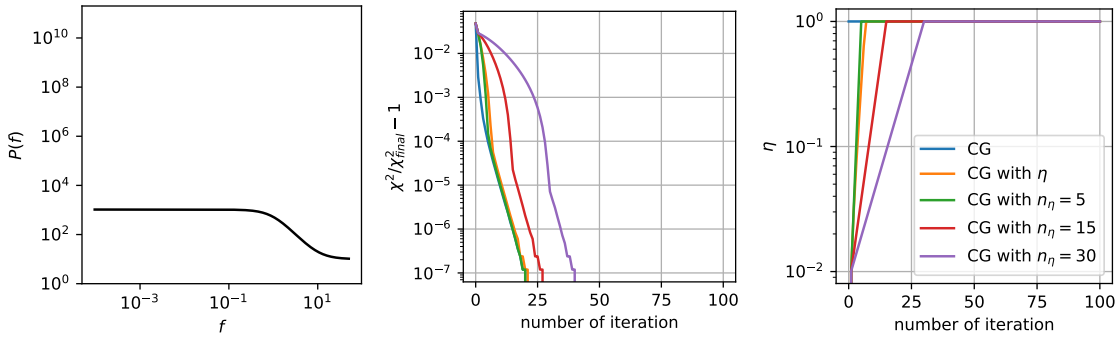
(a) (b) (c)

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

Now let's compare the performance difference between choosing η parameters based on Eq.(28) 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).

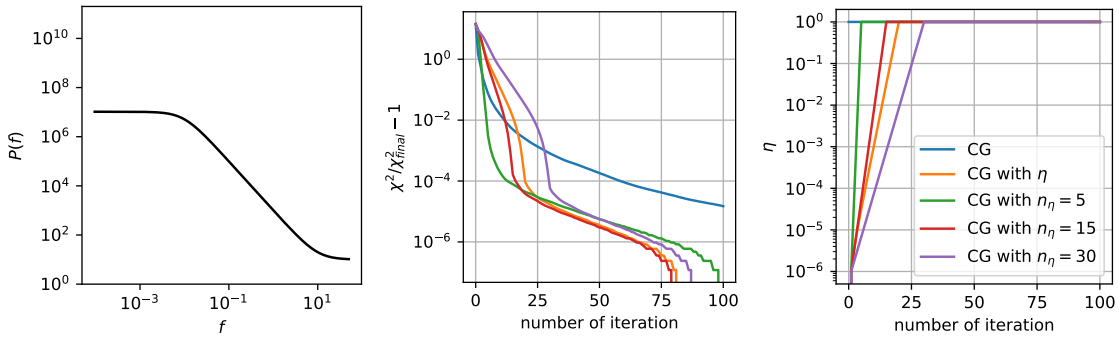
When $\kappa(N)$ is small, and Eq.(28) tells us that only a few η parameters are good enough, see Figure(4b). If unfortunately we choose n_η being large value, like 15 or 30, then it will ends up converge slowly, because it need at least 15 or 30 iterations to reach $\eta = 1$.

On the other hand if $\kappa(N)$ is very large and power spectrum is $1/f$ noise, we need more η parameters. If n_η is too small, for example $n_\eta = 5$ in Figure(6b), which is better than vanilla conjugate gradient method, but still far from optimal.



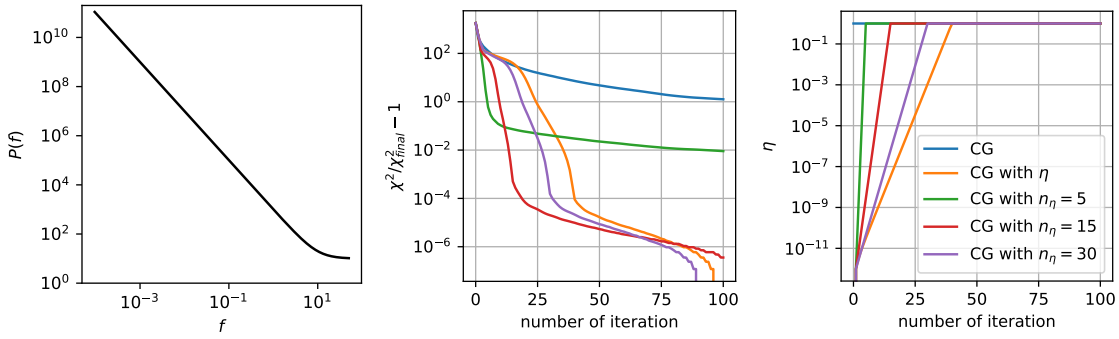
(a) (b) (c)

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



(a) (b) (c)

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



(a) (b) (c)

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

7 Possible improvements

As you may have noticed in Figure(5) and Figure(6), the perturbation parameter based on Eq.(28) is more than needed, especially for $1/f$ noise case. From Figure(6c) we know that Eq.(28) gives us $n_\eta \approx 40$, however based on χ^2 result in Figure(6b), we notice that $n_\eta \approx 30$ or even $n_\eta \approx 15$ is good enough. Also, for the almost 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(4b). However Eq.(28) gives us $n_\eta \approx 6$, see Figure(4c), even though it doesn't make 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.(24). Here I rewrite it in a simplified form

$$-\frac{\delta\chi^2(\hat{\mathbf{m}}(\eta),\eta)}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)} = -\delta\eta\frac{\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)}{\chi^2(\hat{\mathbf{m}}(\eta),\eta)} = \delta\eta\frac{\hat{\mathbf{r}}_\eta^\dagger N_\eta^{-1}\bar{N}N_\eta^{-1}\hat{\mathbf{r}}_\eta}{\hat{\mathbf{r}}_\eta^\dagger N_\eta^{-1}\hat{\mathbf{r}}_\eta} \leq \frac{\delta\eta}{\eta+\frac{\tau}{\max(N_f)-\tau}} \tag{36}$$

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 time ordered data in system RAM, so we need somehow analytically analysis $\mathcal{P}_\eta$, and its behavior in frequency space.

Unless $\mathbf{r}_\eta$ almost only has large noise modes, $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$ won't get close to the upper bound $1/\left(\eta+\frac{\tau}{\max(N_f)-\tau}\right)$. Based on the analysis in Section(4.2.2), for small η the estimated map $\hat{\mathbf{m}}(\eta)$ does not only focusing on minimizing error $\mathbf{r}_\eta$ at low 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 $\frac{\mathrm{d}}{\mathrm{d}\eta}\chi^2(\hat{\mathbf{m}}(\eta),\eta)/\chi^2(\hat{\mathbf{m}}(\eta),\eta)$, and get larger $\delta\eta$ value. This should make η goes to 1 faster, and yields less η parameters we need.

Also noting that the η values determined from Eq.(28)

$$\eta_i = \min\left\{1,\ \frac{\tau}{\max(\bar{N}_f)}\left(2^i-1\right)\right\} \tag{28}$$

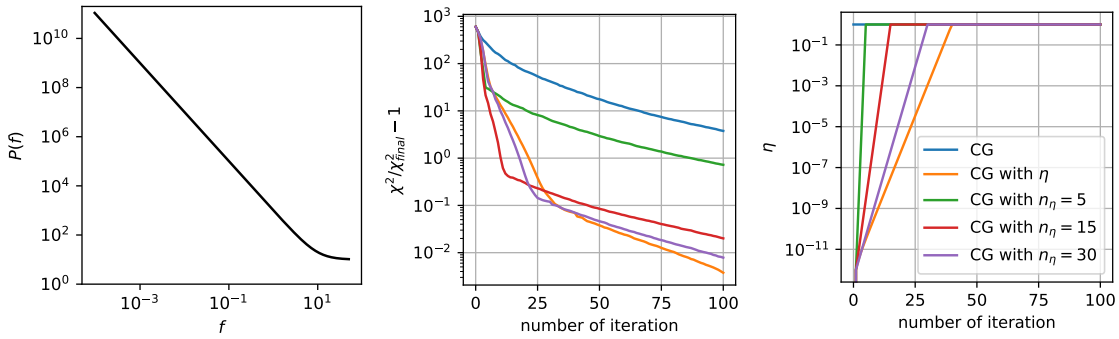
are not dependent on any scanning information, it only depends on noise power spectrum $P(f)$, or noise covariance matrix N. Figure(7) and Figure(8) show two examples with same parameters as in Figure(6) except scanning frequency f_{scan}, in Figure(7) it scans very slow and in Figure(8) it's very fast. In these two cases our η values based on Eq.(28) 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.

8 Conclusion

Here we discussed a method to solve map making equation Eq.(5)

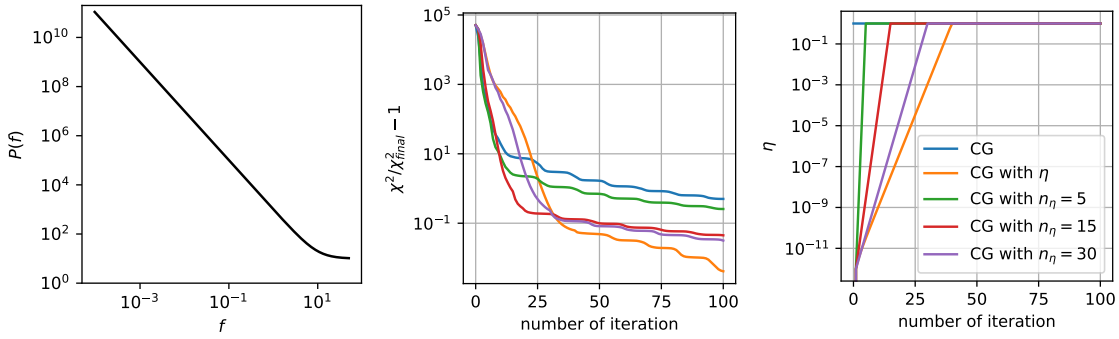
$$\hat{\mathbf{m}} = \left(P^\dagger N P\right)^{-1}P^\dagger N^{-1}\mathbf{d} \tag{5}$$

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



(a) (b) (c)

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



(a) (b) (c)

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

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 timer 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.(28) are not prefect. 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 analysis the pattern 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` [3].

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

References

[1] F. Elsner and B. D. Wandelt. Efficient Wiener filtering without preconditioning. A&A, 549:A111, January 2013.

[2] Particle Data Group, P A Zyla, R M Barnett, J Beringer, O Dahl, D A Dwyer, D E Groom, C J Lin, K S Lugovsky, E Pianori, D J Robinson, C G Wohl, W M Yao, K Agashe, G Aielli, B C Allanach, C Amsler, M Antonelli, E C Aschenauer, D M Asner, H Baer, Sw Banerjee, L Baudis, C W Bauer, J J Beatty, V I Belousov, S Bethke, A Bettini, O Biebel, K M Black, E Blucher, O Buchmuller, V Burkert, M A Bychkov, R N Cahn, M Carena, A Ceccucci, A Cerri, D Chakraborty, R Sekhar Chivukula, G Cowan, G D'Ambrosio, T Damour, D de Florian, A de Gouvêa, T DeGrand, P de Jong, G Dissertori, B A Dobrescu, M D'Onofrio, M Doser, M Drees, H K Dreiner, P Eerola, U Egede, S Eidelman, J Ellis, J Erler, V V Ezhela, W Fetscher, B D Fields, B Foster, A Freitas, H Gallagher, L Garren, H J Gerber, G Gerbier, T Gershon, Y Gershtein, T Gherghetta, A A Godizov, M C Gonzalez-Garcia, M Goodman, C Grab, A V Gritsan, C Grojean, M Grünewald, A Gurtu, T Gutsche, H E Haber, C Hanhart, S Hashimoto, Y Hayato, A Hebecker, S Heinemeyer, B Heltsley, J J Hernández-Rey, K Hikasa, J Hisano, A Höcker, J Holder, A Holtkamp, J Huston, T Hyodo, K F Johnson, M Kado, M Karliner, U F Katz, M Kenzie, V A Khoze, S R Klein, E Klempt, R V Kowalewski, F Krauss, M Kreps, B Krusche, Y Kwon, O Lahav, J Laiho, L P Lellouch, J Lesgourgues, A R Liddle, Z Ligeti, C Lippmann, T M Liss, L Littenberg, C Lourengo, S B Lugovsky, A Lusiani, Y Makida, F Maltoni, T Mannel, A V Manohar, W J Marciano, A Masoni, J Matthews, U G Meißner, M Mikhasenko, D J Miller, D Milstead, R E Mitchell, K Mönig, P Molaro, F Moortgat, M Moskovic, K Nakamura, M Narain, P Nason, S Navas, M Neubert, P Nevski, Y Nir, K A Olive, C Patrignani, J A Peacock, S T Petcov, V A Petrov, A Pich, A Piepke, A Pomarol, S Profumo, A Quadt, K Rabbertz, J Rademacker, G Raffelt, H Ramani, M Ramsey-Musolf, B N Ratcliff, P Richardson, A Ringwald, S Roesler, S Rolli, A Romaniouk, L J Rosenberg, J L Rosner, G Rybka, M Ryskin, R A Ryutin, Y Sakai, G P Salam, S Sarkar, F Sauli, O Schneider, K Scholberg, A J Schwartz, J Schwiening, D Scott, V Sharma, S R Sharpe, T Shutt, M Silari, T Sjöstrand, P Skands, T Skwarnicki, G F Smoot, A Soffer, M S Sozzi, S Spanier, C Spiering, A Stahl, S L Stone, Y Sumino, T Sumiyoshi, M J Syphers,

F Takahashi, M Tanabashi, J Tanaka, M Taševský, K Terashi, J Terning, U Thoma, R S Thorne, L Tiator, M Titov, N P Tkachenko, D R Tovey, K Trabelsi, P Urquijo, G Valencia, R Van de Water, N Varelas, G Venanzoni, L Verde, M G Vincter, P Vogel, W Vogelsang, A Vogt, V Vorobyev, S P Wakely, W Walkowiak, C W Walter, D Wands, M O Wascko, D H Weinberg, E J Weinberg, M White, L R Wiencke, S Willocq, C L Woody, R L Workman, M Yokoyama, R Yoshida, G Zanderighi, G P Zeller, O V Zenin, R Y Zhu, S L Zhu, F Zimmermann, J Anderson, T Basaglia, V S Lugovsky, P Schaffner, and W Zheng. Review of Particle Physics. *Progress of Theoretical and Experimental Physics*, 2020(8), 08 2020. 083C01.

[3] Kevin M. Huffenberger and Sigurd K. Næss. Cosmic microwave background mapmaking with a messenger field. *The Astrophysical Journal*, 852(2):92, jan 2018.

[4] J. Papež, L. Grigori, and R. Stompor. Solving linear equations with messenger-field and conjugate gradient techniques: An application to CMB data analysis. A&A, 620:A59, November 2018.

[5] Cosma Shalizi. Weighted and generalized least squares.

[6] Max Tegmark. CMB mapping experiments: A designer's guide. Phys. Rev. D, 56(8):4514–4529, October 1997.