




Master in High Performance Computing

# Accelerating Production of Cosmic Microwave Background Maps with Deflation Preconditioners

*Supervisor*:
Luca Heltai

*Candidate*:
Giuseppe Puglisi

2nd edition
2015–2016

*We have seen that computer programming is an art,*
*because it applies accumulated knowledge to the world,*
*because it requires skill and ingenuity, and especially*
*because it produces objects of beauty.*

—Donald E. Knuth

## ACKNOWLEDGEMENTS

I would like to thank Luca Heltai for the useful suggestions and discussions we had during this research work. I think this was a remarkable starting collaboration and an example of good interdisciplinary among the Cosmology and the Mathlab groups at SISSA. I am very sure we keep collaborating in the future due to the large quantity of ideas we are going to test in order to cope the map-making problem.
I thank Giulio Fabbian and Carlo Baccigalupi, my PhD supervisors, who strongly supported me in attending the Master in High Performance Computing.
I thank Stefano Cozzini, Giuseppe Brandino and Nicola Cavallini because I realized how much is true what Donald Knuth said by chatting with them.
I am very grateful to the APC team in Paris, i.e. Radek Stompor and Davide Poletti for the very useful chats we had during the developing phase of COSMOMAP2.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACRONYMS

**CMB** Cosmic Microwave Background

**CES** Constant Elevation Scan

**TOD** Time Ordered Data

**PCG** Preconditioned Conjugate Gradient

**SPD** Symmetric Positive Definite

**RHS** Right Hand Side

# 1 INTRODUCTION

## 1.1 THE COSMIC MICROWAVE BACKGROUND RADIATION

Since last decades, we have been able to estimate the age of the Universe, investigate its evolution in time depending on energy and matter contents and observe its first emitted light: the Cosmic Microwave Background (CMB). Since the discovery of the CMB (Penzias and Wilson, 1965), the efforts of several generations of space, ground and balloon telescopes allowed us to directly measure the tiny temperature CMB anisotropies arose from a tight interplay of gravitational and quantum physics. For the latest measurements see Planck Collaboration et al. (2014).

General relativity equations suggest that Universe started to expand and cooling from a hot and high density state, the *Big Bang*. Three observational evidences support the Big Bang theory: (i) the galaxy recessional velocity measurements, (ii) the primordial abundances of the light elements (D, $^4$He, $^3$He, $^7$Li), computed from mathematical models as ratios to the amount of hydrogen, in agreement with the abundances measured in galactic surveys, (iii) the observation of a nearly isotropic radiation the CMB.

According to the theory of the Big Bang, in the hot, dense conditions of the early Universe, photons were tightly coupled to matter via Compton scattering. When the Universe was about 380.000 years old, the temperature dropped below 3000 K allowing atomic hydrogen to recombine and releasing the photons. These photons, then, travelled freely through the Universe as it expanded and cooled. The cosmological relic of them is in the CMB, this is why the CMB is the light emitted at the time of *recombination* from the *Last Scattering Surface* (LSS).

The CMB is a radiation from every direction in the sky. The form of its spectrum is perfectly described by a *blackbody* spectrum with temperature of 2.725 K. In (Figure 1.1) one sees the consistency of the Cosmic Background Explorer (COBE,Fixsen et al. (1996)) data with a blackbody spectrum, error bars are emphasized by a factor of 500.

When it was discovered in the 1965, the CMB was found to be remarkably uniform across the sky. However, in 1992 that the COBE satellite discovered temperature variations at the level of 1 part in 100.000 (Smoot et al., 1992). Temperature maps of the CMB form a snapshot image of the tiny density fluctuations in the primeval Universe. These density fluctuations are thought to grow by gravitational attraction into the familiar structures we see today (stars, galaxies, and clusters of galaxies) according to the gravitational instability model of structure formation.

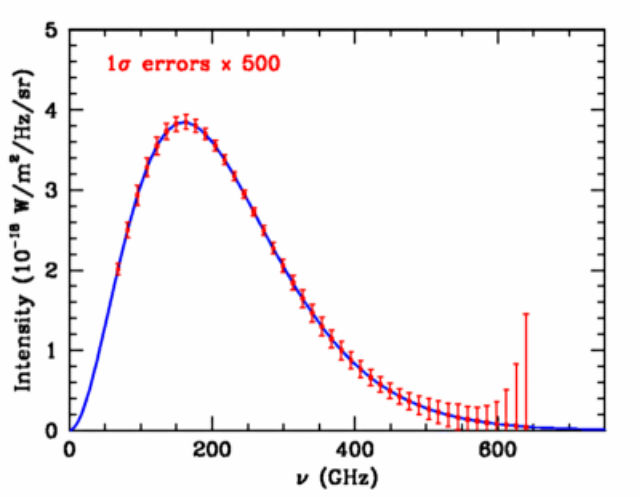


**Figure 1:** Comparison of the intensity of radiation observed with the FIRAS radiometer carried by COBE with a blackbody spectrum with temperature 2.725K. The 1σ experimental uncertainty in intensity is indicated by the tiny vertical bars; the uncertainty in wavelength is negligible (Fixsen et al., 1996).

The CMB is linearly polarized and presents anisotropies 10 times fainter than the temperature ones. In the polarization anisotropies, cosmologists are looking for the imprints of a stochastic background of Gravitational Waves produced when the Universe experienced an exponential expansion, the so-called *inflation*. These primordial gravitational waves are hidden in the *B-modes* CMB polarization maps and they have not detected yet because of the diffused polarized emission from our Galaxy at the microwave frequencies, known as *Galactic foregrounds*, and several technological challenges to get higher sensitivity in polarization detectors at the sub-millimeter wavelength regime. However, lots of efforts have been made by several cosmology groups in order to measure the B-mode: this discovery represents an ultimate way to probe the Universe at the ultra-high energy regimes ($\sim 10^{16}$ GeV).

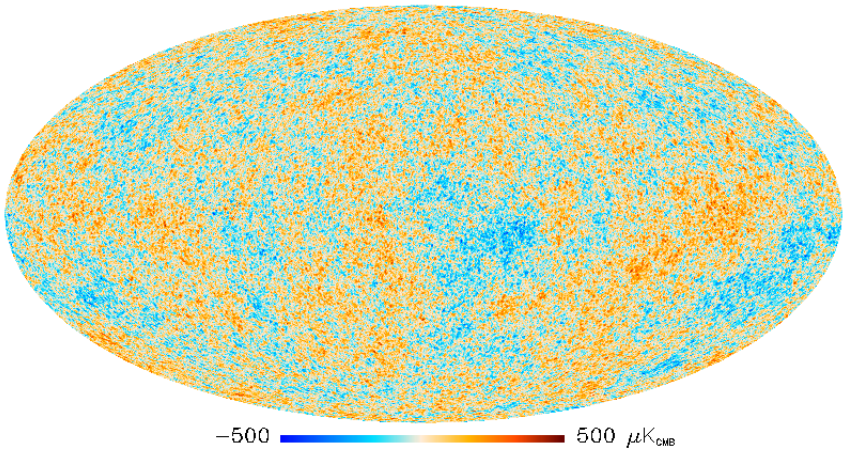


**Figure 2:** The CMB map observed by Planck (Planck Collaboration et al., 2014).

In this work I mainly focused on implementing a fast pipeline to produce the CMB maps from data sets obtained from a typical CMB ground based telescope: Polarbear (The Polarbear Collaboration: P. A. R. Ade et al., 2014). I have been part of this collaboration since 2014 and the procedure described in this work has been proposed as one of the two pipelines for the next Polarbear observational seasons.

## 1.2 CMB MAP-MAKING

The *map-making* procedure of CMB radiation starts from a sequence of time ordered data (aka TOD) composed by $\sim 10^7 \div 10^9$ noisy samples, representing the observation measurements in the time domain by hundreds (or thousands) detectors in the focal plane.

Let us consider a collection of measurements of the TOD stream, $d_t$ (encoding $N_t$ total measurements) performed during a certain lapse of time by one detector of a CMB telescope. As one may expect the measurement can be easily modelled as the sum of an astrophysical signal $s_t$ coming from the line of sight plus the instrumental noise of the detector itself $n_t$. Furthermore, a certain pixel of the sky will have been observed as many times as the telescope scanning direction intercepts that pixel. This is fully encoded into the *pointing matrix* $P_{tp}$, ($N_t \times N_p$, which is a sparse and tall matrix). Thus, since our goal is to get a map, $x_p$ ( with $N_p \ll N_t$ in the pixel domain, $N_p \approx 10^4 \div 10^5$ [1]), the data model can be written as:

$$d_t = s_t + n_t = P_{tp} x_p + n_t. \tag{1}$$

For a given telescope the structure of the pointing matrix can be quite complex since it involves many characteristics of the instrument. In our case

1 Depending on the chosen pixelization grid.

it can be sparse or block-sparse depending whether the measurements are sensitive or not to the polarization. In fact, for total intensity measurement obtained from a single dish experiment, $P$ has only one non-zero element per row corresponding to the p-th pixel observed at time $t$ and $s_t = I_{p_t}$. In the case of polarization measurements, Equation 1 becomes a combination of the three *Stokes parameters*, $I$, $Q$, $U$:

$$s_t = I_{p_t} + Q_{p_t}\cos(2\phi_t) + U_{p_t}\sin(2\phi_t) \tag{2}$$

with $\phi_t$ being the orientation of the polariser with respect to the sky coordinates at time $t$. Therefore, the pointing matrix has three non zero entries per row and $x_p$ becomes an array containing three maps related to the three Stokes parameters commonly related to linear polarization.

A General Least Squares (GLS) solution to the data model in Equation 1 is given by (in a more compact form):

$$\hat{x} = \left(P^t M P\right)^{-1} P^t M d, \tag{3}$$

with $M$ being a positive definite weight matrix (Tegmark, 1997a). If $M \equiv C_n^{-1}$ than $\hat{x}$ is a minimum variance estimator and a maximum likelihood solution.

If one than collects all temporal templates, which have to be filtered out from the data into a single template matrix, $T$, (whose each column correspond to one template ) we can define a filtering operator. It de-projects the temporal components of several templates from the data:

$$d' := (\mathbb{1} - T(T^tT)^{-1}T^t)d \tag{4}$$

Since we want not only to filter all modes (belonging to the subspace spanned by $T$) but also weight all the modes orthogonal to this subspace by a symmetric weight matrix, $M$, we can generalize Equation 4[2] as follows:

$$d' = (M - MT(T^tMT)^{-1}T^tM)d \equiv F_T d. \tag{5}$$

With the help of the filtering operator we can now easily generalise the map-making equation Equation 3 as :

$$\hat{x} = \left(P^t F_T P\right)^{-1} P^t F_T d, \tag{6}$$

By looking at the Equation 6 one may notice that in order to get the map $\hat{x}$ one should compute and invert a $N_p \times N_p$ matrix resulting from the first product $P^tF_TP$ (usually dense and without some general symmetry). The computation usually can be performed by explicitly factorising a huge matrix (with a number of operations (flops) $\sim \mathcal{O}(N_p^3)$) (Tegmark, 1997b; Borrill, 1999; Stompor et al., 2002). Recently, Poletti et al. (2016) applied an explicit inversion of the system matrix in a very similar context: small regions of the sky (0.1%) observed by the Polarbear Telescope by means of distributed data among several processes and exploiting `Scalapack` routines.

2 This generalization is unique if we require the operator being symmetric.

This approach to the map-making problem is becoming more and more unfeasible because of the increasing need by the several CMB collaborations of observing larger regions of the sky.

An alternative to the explicit inversion approach is represented by iterative methods (Wright, 1996; Oh et al., 1999; Doré et al., 2001). They involves algorithms as the one we are considering in this work, namely the *Conjugate Gradient* within the class of *Krylov* methods, (Golub and Van Loan, 1996). They are aimed at iteratively solving the map-making equation by means of large matrix-vector products. Usually, they do not even need to store the whole system matrix in the memory and they circumvent the matrix inversion issue.

So far, many CMB iterative solvers are based on a preconditioned conjugate gradient (PCG) method assuming white noise in the time domain data (Wandelt and Hansen, 2003; Naess et al., 2014). The map-making problem started to be a numerical challenge and it is getting more and more challenging as the sizes of the maps of current and forthcoming CMB experiments will increase.

## 1.3 THE PRECONDITIONED CONJUGATE GRADIENT

By looking at Equation 6, it is worth noticing that we can rewritten it as a linear system:

$$\begin{aligned} \left(P^t F_T P\right) x &= P^t F_T d, \\ &\Downarrow \\ Ax &= b, \\ \text{where } A &= P^t F_T P, \qquad (7) \\ \text{and } b &= P^t F_T d, \qquad (8) \end{aligned}$$

with $A$ symmetric and positive definite (SPD).

The key-point of the Conjugate Gradient (CG) method is that after a certain number of iteration steps the vector $x^{(k)}$ gets the true solution $x$ of the linear system without inverting the matrix $A$ but by minimizing the residual at k-th iteration step: $r^{(k)} = \| b - Ax^{(k)} \|$. Therefore, $x^{(k)}$ satisfies:

$$\lim_{k\to\infty} x^{(k)} = x,$$

The iterative process is stopped when it is reached a fixed tolerance: $\| x^{(k+1)} - x^{(k)} \| < \epsilon_{tol}$. In Appendix A we describe the whole algorithm, here we would like to emphasize the fact that the most frequent computational operation in the CG is the matrix-vector multiplication. It is thus not necessary to store the matrix $A$ in the memory, but given the very known structure of the pointing and filtering matrix we implemented directly the matrices $P$, $F_T$ as linear operators.

Moreover, it is well known that the convergence rate of the CG depends on the condition number $\kappa$ of the matrix $A$. It can be shown that (Golub and Van Loan, 1996) after $k$ iterations of CG,

$$\| x - x^{(k)} \|_A \leqslant 2 \| x - x^{(0)} \|_A \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k, \tag{9}$$

where $x^{(0)}$ is an initial guess for the solution $x$ and $\| x \|_A$ is the norm of $x$ defined as $\| x \|_A = \sqrt{x^T A x}$. The condition number, $\kappa = \kappa(A)$, is given by the ratio of the largest to the smallest eigenvalue of $A$.

The system matrices related to the map-making problem are commonly bad conditioned, i.e. $\kappa \gg 1$ because of the degeneracies introduced by the time-domain filtering, $F_T$. It is then very useful to *precondition* the linear system by a matrix $M_P$ such that $M_P A$ has a smaller condition number (or a more clustered eigenspectrum) and thus achieves the CG convergence within a smaller number of iterations. This algorithm is usually referred as *Preconditioned Conjugate Gradient* (PCG) and is aimed at solving the following linear system:

$$M_P A x = M_P b \tag{10}$$

where $M_P$ is called *preconditioner*. For further details or a full description of the algorithm please refer to Appendix A.

## 1.4 THE JACOBI PRECONDITIONER

One of the easiest and most intuitive preconditioners is the *Jacobi Preconditioner* which in the fashion of the Equation 3 is defined as:

$$M_P \equiv M_{BD} = (P^t \, diag(C_n^{-1}) P)^{-1}. \tag{11}$$

In particular, when no time domain filtering is introduced and the instrumental noise does not have extra-correlations (among detectors) but it is fully described by its covariance matrix $C_n$, $M_{BD}$ is exactly $A^{-1}$. Given the sparsity structure of the pointing matrix $P$, described above, one could easily realize that $M_{BD}$ is block diagonal and the sizes of each block is equal to the number of the Stokes parameters we are taking into account. This is why, we will refer to it in the following sections as the standard or block-diagonal preconditioner. This class of preconditioners has a further desirable property that it could be parallelized pretty well, especially on distributed memory computers.

The action of this preconditioner onto the eigenspectrum of the matrix $A$ is to shift the large eigenvalues nearly to the unity. However, this reflects on a reduction of the condition number as far as smaller (close to zero) eigenvalues are not concerned. Unfortunately, in our case matrix $A$ has very small eigenvalues with a very *low-energy* content (being related to the smallest eigenvalues). On the other hand, they have a very high content in noise which hinders the convergence of PCG.

## 1.5 DEFLATION PRECONDITIONERS

An alternative preconditioner may be found among the class of so called *Deflation* preconditioners that have proven to be successful in presence of few isolated extremal eigenvalues. They act as de-projectors from the so called *deflation subspace*, $\mathcal{Z}$. This subspace is generated by $r$ linearly independent eigenvectors that are related to the smallest eigenvalues and constitute the columns of the *deflation* matrix $Z$. The projection onto the deflation subspace may be defined as:

$$R = \mathbb{1} - AZ(Z^tAZ)^{-1}Z^t \tag{12}$$

The deflation subspace, $\mathcal{Z} = span\{Z\}$, has a dimension given by $r = rank(Z)$. Since $r \ll N_p$ it is very easy to invert the *coarse* operator $E = (Z^tAZ)$, either via an LU factorization or an eigen-decomposition. As $A$ is SPD, so is $E$. The projector $R$ is orthogonal to any vector $w \in \mathcal{Z}$ since $RAZ = 0$.

On the other hand, being $A$ SPD, one can build the orthogonal complement $\mathcal{Y}$ of $\mathcal{Z}$, i.e. such that $\mathcal{Y} \oplus \mathcal{Z}$ and $y^tz = 0$ for any $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$. The easiest choice for a basis of $\mathcal{Y}$ could be all the remaining larger eigenvectors of $A$, $Y$, such that $\mathcal{Y} = span\{Y\}$. For all of these $Y$, orthogonal to the deflation subspace it holds that $RAY = \lambda RY = \lambda Y$.

One can prove that if $\mathcal{Z}$ is invariant for $A$, then $\mathcal{Y}$ is so for $RA$. The demonstration comes by stating that for any invertible matrix $B$, $AY = YB$, hence $RAY = RYB = YB$. Finally, when $\mathcal{Z}$ is invariant :

$$\kappa_{eff}(RA) = \frac{\lambda_{N_p}(A)}{\lambda_r(A)}. \tag{13}$$

In other words, the deflation of an invariant subspace cancels the corresponding small eigenvalues, and leaves untouched the remaining eigenvalues.

In the exact precision algebra, $R$ would be a very efficient preconditioner, as for each steps of an iterative CG-like solver would be orthogonal to the null space of the $RA$. However, we deal in a finite precision arithmetic and the zero eigenvalues are often as bothersome as the small ones due to the numerical precision of the machine.

The solution to this issue comes by combining this deflation preconditioner with another one. Due to such combination we refer it as the *two-level* preconditioner. Following Szydlarski et al. (2014) we combined it to the standard Jacobi preconditioner:

$$\begin{aligned} M_{2l} =& M_{BD}R + ZE^{-1}Z^t \\ =& M_{BD}(\mathbb{1} - AZ(Z^tAZ)^{-1}Z^t) + ZE^{-1}Z^t. \end{aligned} \tag{14}$$

We note that this new preconditioner defined above fixes the issue by rescaling all the null eigenvalues of $RA$ to one:

$$M_{2l}AZ = Z$$

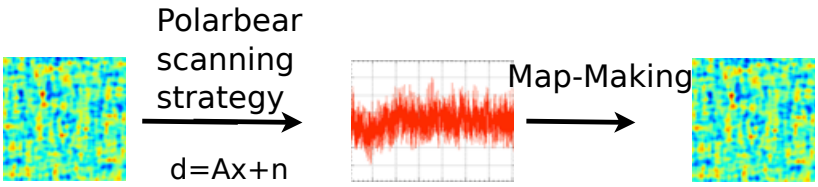


**Figure 3:** The pipeline followed in this work.

and since what we stated above, $M_{2l}$ acts onto a vector $y \in \mathcal{Y}$ ( the orthogonal complement of $\mathcal{Z}$) as the $M_{BD}$ does:

$$M_{2l}Ay = M_{BD}Ay.$$

In conclusion, if we define $\mathcal{Z}$ as the deflation subspace that includes all small eigenvalues of $A$, the two-level preconditioner will shift both the small and large eigenvalues of $A$ close to one. Therefore, what is very challenging in this context is to construct the basis $Z$ ensuring that the smallest eigenvalues (being the most troublesome eigenvalues in terms of the PCG convergence) are included and building the deflation subspace numerically efficient in order to have competitive and higher performances compared to the standard preconditioner.

## 1.6 DESCRIPTION OF COSMOMAP2

We have implemented a `Python` package `Cosmic Microwave linear Operators for MAp making Preconditioners 2 levels` (COSMOMAP2[3]). This package is aimed at solving the map-making problem by means of the PCG and its fundamental bricks are the matrices involved in Equation 6 and 10 all implemented as Linear Operators. We capitalize in developing a fast implementation of the matrix-vector product especially for the time-domain operators (such as $P$, $F_T$).

The other remarkable section of the code refers to the simulation pipeline of input data processing. We simulated observations by exploiting the scanning strategy and instrumental noise expectation from the Polarbear telescope pipeline, as described in the cartoon in Figure 3. In this way we have full control of the outputs of our code by comparing them with the input map, which is a signal only CMB map. The documentation of COSMOMAP2 is available online at `http://giuspugl.github.io/`.

3 https://github.com/giuspugl/COSMOMAP2

This work can be divided in two main parts: (i) in Chapter 2 we focused in a comparison of performances involving two different preconditioners introduced in Section 1.4 and Section 1.5 and (ii) in Chapter 3 we proposed an optimal solution which further improves performances of the two-level preconditioner.

# 2

# THE COMPARISON BETWEEN JACOBI AND 2-LEVEL PRECONDITIONERS

In this chapter we compare the performances of two different preconditioners: the Jacobi and the two-level ones applied on the same simulated dataset. To be very close to the observation of the Polarbear telescope we simulated signal observations by scanning the sky with the Polarbear strategy.

In this work we are more interested to apply, benchmark and test this new class of preconditioner to the map-making problem of a CMB telescope as Polarbear in presence of time-domain filtering. Indeed, although a realistic dataset contains signal and noise into the data stream, the properties of the whole sample are statistically equivalent with or without simulated noise in the dataset.

As described in the end of previous chapter, the two-level preconditioner relies on a deflation preconditioner, i.e. a de-projection of the deflation subspace $\mathbb{Z}$ generated by the eigenvectors related to the smallest eigenvalues of our system matrix.

However, we do not have to compute the whole eigenspectrum of the matrix (see Poletti et al. (2016)). We thus exploited the *Krylov* subspace projection (Golub and Van Loan, 1996) algorithms to compute approximations of the smallest eigenvalues: the so-called *Ritz* eigenvalues.

## 2.1 THE ARNOLDI ALGORITHM

The Krylov subspace algorithms are based upon the structure of a sequence of vectors naturally produced by the power method, one of those is the

*Arnoldi* algorithm, the key-method of the Generalized Minimal Residual method (GMRES) (Golub and Van Loan, 1996).

An examination of the behaviour of the sequence of vectors produced by the power method sequence suggests that the successive vectors may contain considerable information along eigenvector directions corresponding to eigenvalues other than the one with largest magnitude. The expansion coefficients of the vectors in the sequence evolves in a very structured way. Therefore, linear combinations of the these vectors might well be devised to expose additional eigenvectors.

Our aim is to find an approximation to the eigenvalues of a matrix $B$ whose linear system is :

$$Bx = b \tag{15}$$

and to obtain additional information through various linear combinations of the power sequence. It is, then natural to formally consider the Krylov subspace defined as

$$\mathcal{K}_m(B, b) = span\{b, Bb, B^2b, \ldots, B^{m-1}b\} \tag{16}$$

where $b$ is the *starting guess vector*. Such a Krylov subspace therefore approximates the solution as:

$$x_m = x_0 + \mathcal{P}_{m-1}(B)r_0,$$

with $\mathcal{P}_{m-1}$ polynomial of degree $m-1$. The Arnoldi algorithm is a fast algorithm which deliver an orthogonal basis of the Krylov subspace and is summarized in Algorithm 1.

**Algorithm 1** Basic Arnoldi Algorithm

**Require:** : $r_0$, $w_1 = r_0 / \parallel r_0 \parallel$

**for** $j = 1 \rightarrow m$ **do**
  **for** $i = 1 \rightarrow j$ **do**
    $h_{i,j} = (Bw_j, w_i)$
  **end for**
  $v_j = Bw_j - \sum_{i=1}^{j} h_{i,j} w_i$
  $h_{j+1,j} = \parallel v_j \parallel$
  $w_{j+1} = v_j / h_{j+1,j}$
**end for**

The output from the Arnoldi algorithm is an orthonormal basis $W^{(m)} = (w_1|w_2|\ldots|w_m)$, together with a set of scalars $h_{i,j}$ (with $i, j = 1, \ldots, m$ and $i \leqslant j+1$) plus an extra-coefficient $h_{m+1,m}$. The former set of coefficients are the elements of an upper Hessenberg matrix $H_m$with non-negative subdiagonal elements and is usually called as a *m-step Arnoldi Factorization* of $B$. If $B$ is hermitian then $H_m$ is symmetric,real and tridiagonal and the columns

of $W^{(m)}$ are referred as *Lanczos vectors* (otherwise they are called *Arnoldi vectors* ). B and $H_m$ are related via the following relation:

$$BW^{(m)} = W^{(m)}H_m + h_{m+1,m}w_{m+1}e_m^t, \tag{17}$$

where $e_m$ is a $1 \times m$ unit vector with 1 on the $m$-th component. In other words, $H_m$ is the projection of B in the subspace generated by the columns of $W^{(m)}$ within a certain error threshold given by the matrix $\tilde{W}_m = h_{m+1,m}w_{m+1}e_m^t$. However, the purpose here is to investigate the use of this factorization to obtain approximate eigenvalues and eigenvectors of B. Let us consider an eigenpair of $H_m$, $(\lambda_i, y_i)$:

$$H_m y_i = \lambda_i y_i,$$

then the vector $v_i = W^{(m)}y_i$ satisfies:

$$\| Bv_i - \lambda_i v_i \| = \| (BW^{(m)} - W^{(m)}H_m)v_i \| = \| \tilde{W}_m v_i \| . \tag{18}$$

The eigenpairs of $H_m$ are called *Ritz eigenpairs* and they are immediately available since the size of $H_m$ is $\sim \mathcal{O}(100)$ for a typical CMB case (the error threshold is set about $10^{-6}$). Hence the relation in Equation 18 may be used to provide computable rigorous bounds on the accuracy of the eigenvalues of $H_m$ as approximations to eigenvalues of B.

## 2.2 BUILDING THE DEFLATION SUBSPACE

Given the definitions of the 2-level preconditioner made in Section 1.5, the matrix to which apply the Arnoldi algorithm described above is $M_{BD}A$, hence we set $B = M_{BD}A$.[1].

The procedure calculates the set of Ritz eigenpairs of $M_{BD}A$, $(\lambda_i, v_i)$. However, we select from all the eigenpairs the first $r$ smallest eigenvalues and the eigenvectors related to them. This allows us to build an orthonormal basis of the deflation subspace $\mathcal{Z}$ with the selected $r$ eigenvectors $Z_D$ and construct the two-level preconditioner with them.

We select all the eigenvalues $\lambda_i$ satisfying:

$$\frac{\lambda_i}{\lambda_{max}} < 10^{-2}.$$

Usually, the size of the deflation subspace is $rank(Z_D) \approx \mathcal{O}(10)$.

The Ritz eigenvectors $Z_D$can be shown as maps in the sky patch and they have the peculiar feature to appear as long modes across the observational patch along the direction of the degeneracy introduced by the filtering. In Figure 4 are shown some of those Ritz eigenvectors. The first row of Figure 4 shows maps with horizontal or circular structures. They are introduced by a filtering of the ground template. The other panels (second and third rows)

1 We defined the matrix A in Equation 7 and $M_{BD}$ in (11)

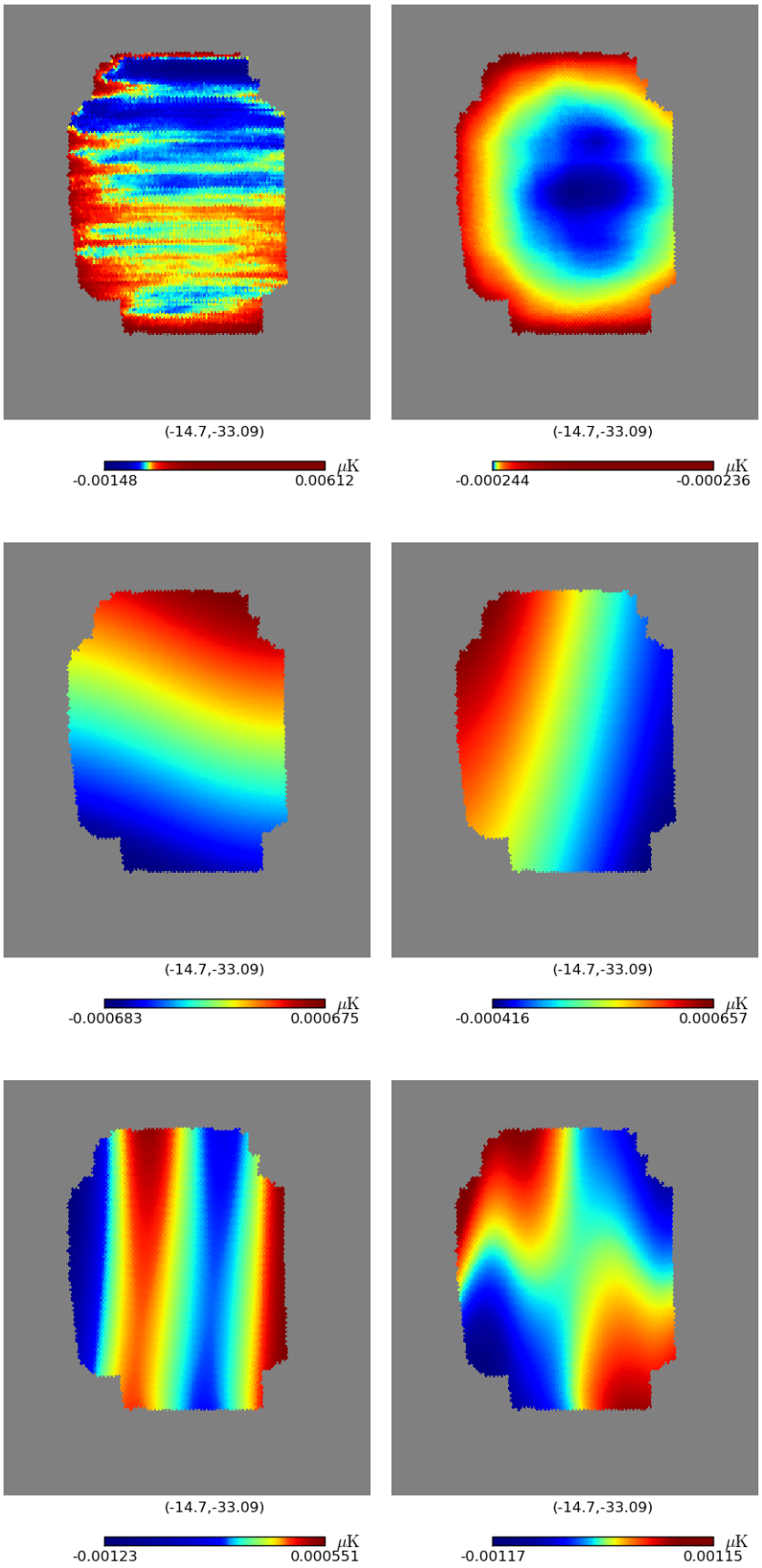


**Figure 4:** Some of the deflation eigenvectors related to the selected small eigenvalues. They change depending on which kind of filtering is applied to the data. First row are modes with a filtering of the ground template, whereas the others are modes with polynomial filters.

shown in Figure 4 are Ritz eigenvectors computed by applying to the whole dataset different polynomial filters. We filtered from the data samples the component parallel to Legendre polynomials (up to the $3^{rd}$ order). This is very useful to remove the atmospheric contamination especially for observation from the ground.

However, exact eigenvectors of such linear system have been studied in a paper which I coauthored with Poletti et al. (2016). The ones related to the smallest eigenvalues are shown in Figure 9 of Poletti et al. (2016) as *nearly degenerate modes* and they are consistent with the approximated eigenvectors computed via the Arnoldi algorithm.

## 2.3 COMPARING JACOBI AND 2-LEVEL PRECONDITIONERS

To summarize what has been discussed in the previous sections Section 2.1 and 2.2, one could state that the Jacobi preconditioner does not help the convergence of the CG algorithm since the condition number of the preconditioned matrix remains very large ($\mathcal{O}(500)$), because the eigenspectrum of $A$ is not so clustered. We remind that the number of CG iterations is essentially proportional to it. We thus followed the solution proposed by Szydlarski et al. (2014) and applied the two-level preconditioner to the Polarbear dataset, once we have run the Arnoldi algorithm to build a basis to the Deflation subspace with the Ritz eigenvectors.

When $M_{2l}$ is applied to the system matrix $A$, we are allowed us to represent $M_{2l}A$ as a two block matrix. The first block is the one related to the deflation subspace for which it holds :

$$M_{2l}Az = z, \ \forall z \in \mathcal{Z}$$

and the second one is identified by the vectors $y \in \mathcal{Y}$ orthogonal to $\mathcal{Z}$ where it is possible to run the CG. In this space the matrix $M_{2l}A$ has a smaller condition number $\mathcal{O}(10)$, see the considerations made at the end Section 1.5, therefore, the CG gets the tolerance threshold with a small number of iterations.

In order to compare the performances of both the Jacobi and the two-level preconditioners we consider a common right hand side (RHS) vector $b$ as it has been defined in Equation 8 collecting 4 constant elevation scans (CES) of Polarbear telescope (15 minutes each, $N_t \sim 10^8$) and we filter our data set with a 0-th order Legendre polynomial (the action of such a filter is to remove the average value of the TOD within a sub-scan chunk of data). The Arnoldi algorithm reached the error threshold $10^{-7}$ after 211 iterations. We then selected 8 Ritz eigenvectors whose eigenvalues were smaller than 0.01:

$$\tilde{\lambda}_i = [-2 \cdot 10^{-15}, -1 \cdot 10^{-15}, -1 \cdot 10^{-16},$$

$$5 \cdot 10^{-16}, 1 \cdot 10^{-15}, 2 \cdot 10^{-15}, 5 \cdot 10^{-4}, 6 \cdot 10^{-4}]$$

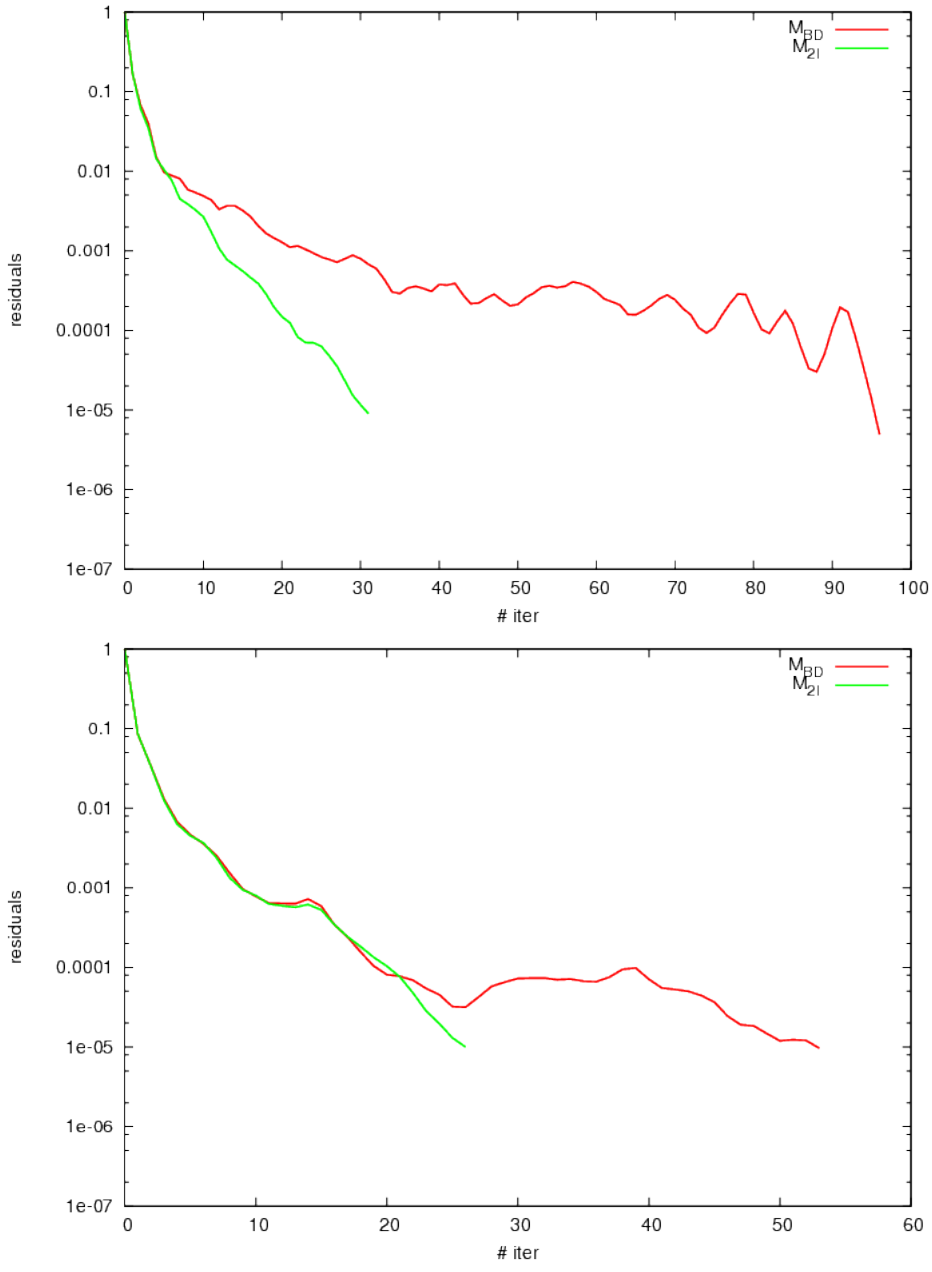


**Figure 5:** Residuals $\| r^{(k)} \| / \| b \|$ vs the CG iteration steps for the (solid red) Jacobi and (solid green) two-level preconditioners. The top panel refers to the run with only one Stokes Parameter (the intensity $I$ one), the bottom with both the $Q, U$ linear polarization parameters.

Therefore, we run *serially* the CG with the two preconditioners and we stop it as soon as we reach the convergence threshold:

$$\frac{\| r^{(k)} \|}{\| b \|} = \frac{\| M_P A x^{(k)} - b \|}{\| b \|} = 10^{-5}.$$

Moreover, since we developed the COSMOMAP2 code to solve the map-making problem for all the Stokes parameters, i.e. intensity + polarization, in Figure 5 are shown results for both intensity (top) and polarization (bottom) cases. The difference among the two cases is that the number of pixels per map, i.e. $N_p \approx 24,000$, doubles if one solves only for polarization maps, i.e. the two Stokes parameters[2]. Figure 5 shows the residuals at each CG iteration step for the Jacobi and two-level preconditioners respectively in red and green.

It is remarkable to state that the performances for the two-level preconditioner do not change if a larger number of pixels are involved. On the contrary, looking at the performances of $M_{BD}$ the iteration steps are even larger for intensity only maps than for the polarization case (91 vs 52). This is due to the fact that the number of iteration steps is strictly related to the condition number of the preconditioned matrix, whereas it does not depend on the number of pixels. The plateau observed for the Jacobi convergence performance is very well known by the Polarbear collaboration and already observed by Szydlarski et al. (2014). It is due to the presence of the small degenerate eigenvalues of the system matrix A for which the CG is not able to converge. They constrain the convergence to hang up within a higher threshold ($10^{-4}$) and makes the residuals to fluctuate around it. Vice versa, the two-level preconditioner gets the CG tolerance with a speed-up of $2 \div 3$ iteration steps less compared to the Jacobi one as expected.

However, one may argue that the gain in execution time is negligible since the total time took to run the Arnoldi algorithm and the PCG is larger than the PCG using simply the Jacobi preconditioner. In fact, this is the case: by timing these two runs, the PCG took $\sim 1 \div 2$ minutes for both the methodologies, whereas the Arnoldi algorithm took $\sim 30$ minutes. This is quite expected since the algorithm needs $\sim \mathcal{O}(100)$ iterations and encodes several matrix vector operations onto time domain vectors($N_t \sim 10^8$).

This would not be an issue if it could be possible to compute the deflation subspace by considering only a single case containing enough structure informations which may be shared among several RHSs. If we were able to build such a deflation subspace, we could overcome this issue by running only once the Arnoldi algorithm with the most representative basis of the Deflation subspace.

Due to our strong interests in seeking the most representative deflation subspace among several observations, we devote the next Chapter 3 to it.

Finally, by comparing Figure 6 and 7 one can notice that the residual maps shown in the right panels are different. The difference maps in Figure 6, de-

2 Namely, the number of pixels in the output maps is equal to the number of Stokes parameters involved times the number of pixel per map

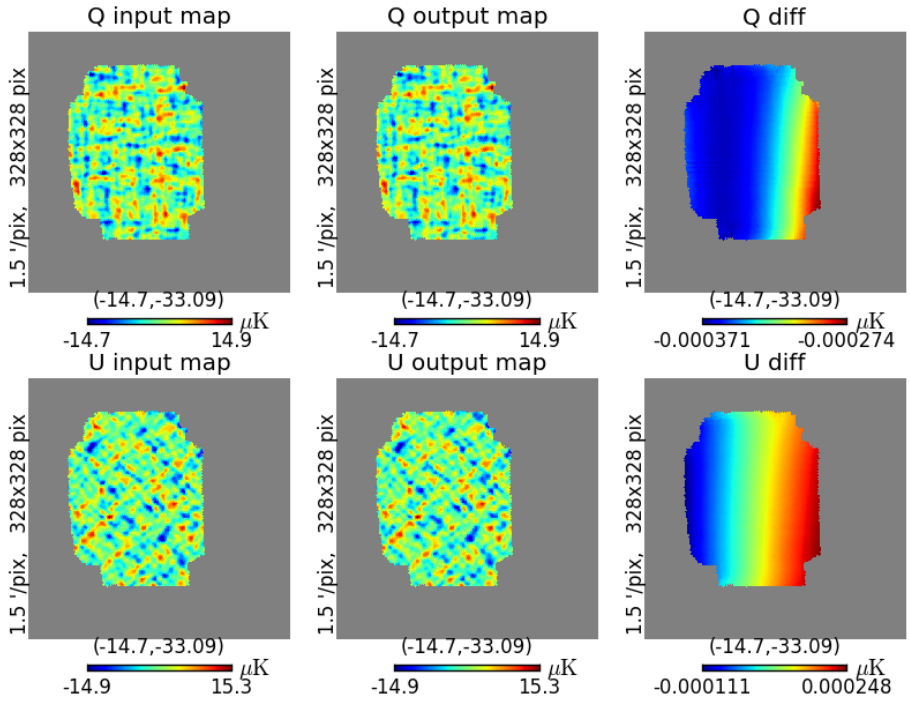


**Figure 6:** From top to bottom Q, U maps computed with Jacobi preconditioner (middle panel). They are compared to the input ones (left panel ) which are used to simulate the Polarbear data, differences are shown in the right panel

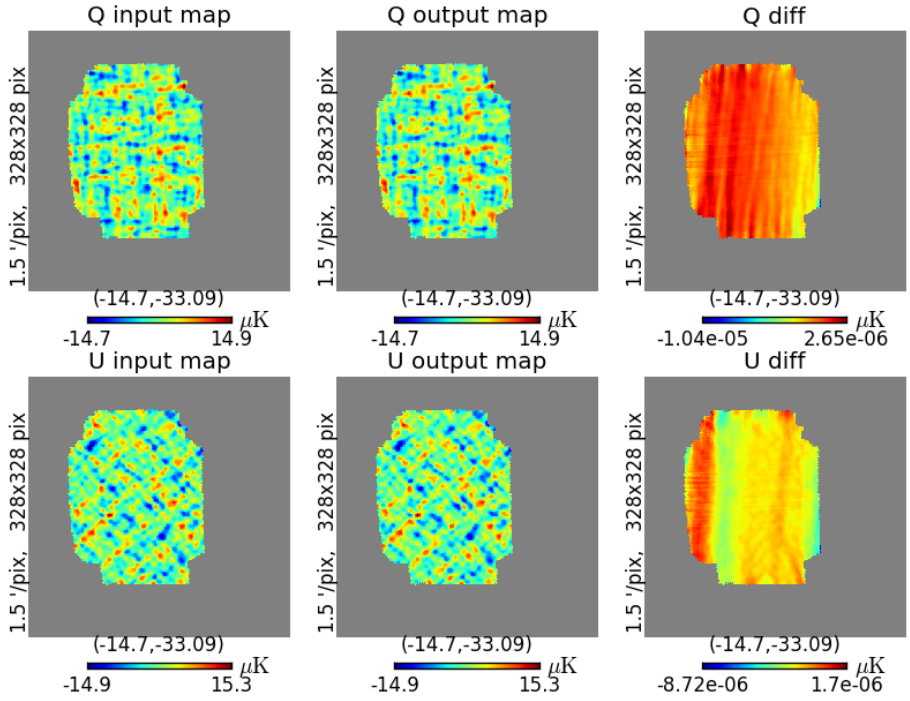


**Figure 7:** As in Figure 6, Q, U maps computed with the two-level preconditioner.

fined as the difference between the input and the output maps, look very similar to the Ritz eigenvector maps shown in Figure 4. The interpretation to that is quite straightforward: the PCG with the block-diagonal preconditioner converges more hardly because is not able to solve the component of the solution parallel to the low modes, which as we have already commented, is strictly related to the smallest eigenvalues. On the other hand, the difference maps in Figure 7 have narrower ranges (around $\pm 10^{-6}$), though we can still infer some low mode trend in them, it represents a clear indication of the better quality of the run performed with two-level preconditioner.

## 2.4 COMPARISON ON THE FULL SEASON DATASET

The first season of Polarbear observations began in 2012 June and ended in 2013 June. Since the full dataset is divided in CES, there is not actual need to communicate among the processor, the problem can be already solved *in parallel* since the characteristic of the full set of data. Thus, each processor solves the linear system in Equation 6 for one daily maps (in total there are $\sim$ 300 days of observations, usually encoding $\sim 20 \div 30$ CES files per day, each one containing the scanning strategy, the RHS, the sub-scans etc...). The total number of samples is about $N_t \approx 10^9$ and we solved the map-making problem for maps of $N_p \approx 24,000$ pixels each (as the one shown in Figure 7). Moreover, given the cosmological importance of the CMB polarization signal, we focused mainly on the $Q, U$ maps which essentially doubles the number of pixels $N_p \approx 48,000$.

The results shown hereafter refer to polarization only maps.

The deflation subspace has been computed by running once the Arnoldi algorithm on one day of data. As described in the previous section, it required about 30 minutes to converge in serial execution ($\sim$ 0.5 **cpuh**). We found 6 Ritz eigenvectors with $10^{-6}$ as the Arnoldi tolerance, we therefore applied the two-level preconditioner (built with this deflation basis) to the whole dataset.

We set the PCG tolerance to $10^{-7}$ and ran the PCG with both the preconditioners. Moreover, if the PCG does not get the tolerance within 150 steps we stop it and flag the run as `unconverged`.

Since the whole dataset of Polarbear collaboration have been located in the storage systems at NERSC (Berkeley, CAL), we performed our runs on the Edison computational system[3] across 100 processors.

All the executions for both preconditioners converged and each one required $\sim$ **600 cpuh**. The statistics of the runs are summarized in Table 2 and through the histograms in Figure 8. However, since the histograms do not show a preferred symmetry, we computed also the median values (see Table 1) for the execution times, iteration steps and residual norms. They are consistent with the values listed in Table 2.

3 NERSC website.

One can easily realize that there is an improvement in terms of performances: the average of PCG execution time and of the iteration steps for the two-level preconditioner is a factor of 1.5 less than the Jacobi one.

This is an interesting result since it confirmed our expectations that the degeneracies introduced by the smallest eigenvalues observed in one month of data are roughly the same for the remaining data set (see Poletti et al. (2016)). However, we ask ourselves whether it is possible to improve these performances and compute *a priori* a better deflation subspace since there are specifics in common among all the RHSs and similar degeneracies when the same filters are applied.

| | PCG execution time [sec] | Iteration steps | $\| r^k \| / \| b \|$ $(\times 10^{-8})$ |
|---|---|---|---|
| $M_{BD}$ | 392.8 | 32 | 5.8 |
| $M_{2l}$ | 260.4 | 20 | 6.2 |

**Table 1:** Median values of PCG runs with the Jacobi and the two level preconditioner.

| | PCG execution time [sec] | Iteration steps | $\| r^k \| / \| b \|$ $(\times 10^{-8})$ |
|---|---|---|---|
| $M_{BD}$ | $402.3 \pm 113.2$ | $32.3 \pm 6.4$ | $6.1 \pm 2.2$ |
| $M_{2l}$ | $283.7 \pm 134.5$ | $22.8 \pm 11.1$ | $6.4 \pm 1.9$ |

**Table 2:** Mean and standard deviation of PCG runs with the Jacobi and the two level preconditioner.

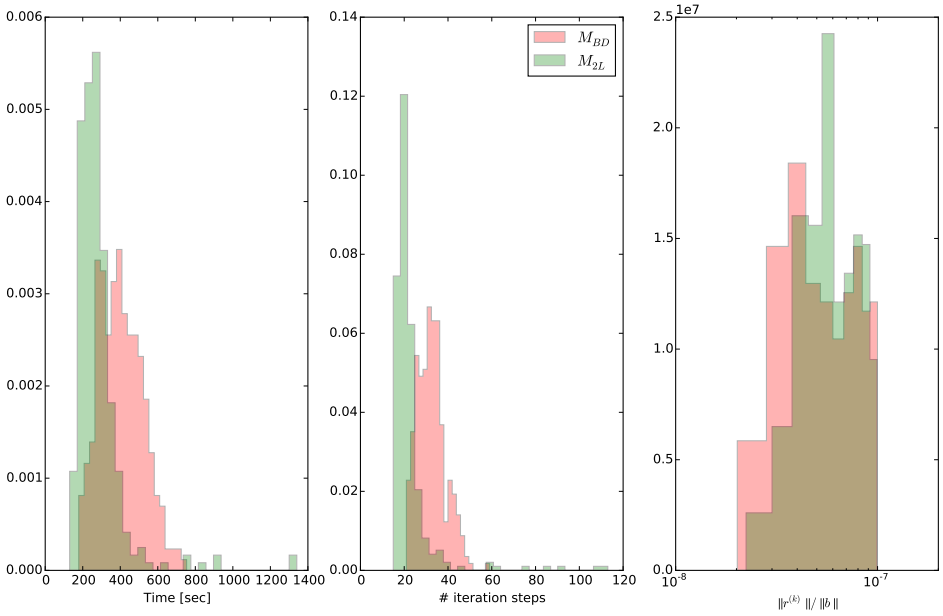


**Figure 8:** Normed histograms of (left) PCG execution time, (middle) total iteration steps, (right) norm of the residuals for the $M_{BD}$ (red) and the $M_{2l}$ (green) preconditioners.

# 3

# ACCELERATING CONVERGENCE WITH DEFLATION PRECONDITIONERS

In the previous chapter we have seen that the PCG converges more quickly if the linear system is preconditioned by a two-level preconditioner. This is due to the fact that we reduce the condition number by means of this class of preconditioners and hence less iteration steps are needed to converge. However, it is not convenient to compute for any RHS the deflation subspace (the procedure involves the Arnoldi algorithm which takes time due to the number of iterations needed).

This is the reason why in Section 2.4 we did not compute the deflation basis for each RHS but we did only once and then we used it as the basis to all the RHS of the full Polarbear dataset.

Although we get improvements in terms of iteration steps and execution time, we wonder if it is possible to compute a more representative deflation subspace whose basis *better* approximates properties encoded in several RHSs.

This kind of issues in numerical analysis are approached via the *Low Rank Approximation* or the *Proper Orthogonal Decomposition* and usually involve the *Singular Value Decomposition* of a matrix. They are aimed at extracting a basis encoding characteristics from the system of interest. Generally speaking, these methods give a *good* approximation with substantially lower dimensions.

### 3.1 LOW RANK APPROXIMATION OF A MATRIX

For any matrix $R$ of rank $r$ it is always possible to decompose it via a Singular Value Decomposition (SVD) (Golub and Van Loan, 1996):

**Theorem 1** (SVD Theorem). *If* $R \in \mathbb{R}^{m\times n}$ *and* $\mathrm{rank}(R) = r$ $\Rightarrow \exists U, V,\ U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$, *orthogonal matrices such that:*

$$U^t R V = \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p), \tag{19}$$

with $p = \min\{m, n\}$, $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_p > 0$ and $\Sigma \in \mathbb{R}^{m\times n}$ is called the *SVD* of $R$. The $\sigma_i$ are commonly referred as the *singular values*, $u_i$ and $v_i$ respectively columns of $U$ and $V$ are known as the *left* and *right* singular vectors of $R$.

**Corollary 1.1.** *If* $U^t R V = \Sigma$ *is the SVD of* $R \in \mathbb{R}^{m\times n}$, *with* $m \geqslant n$ *then for* $i = 1, \ldots, n : Rv_i = \sigma_i u_i$ *and* $R^t u_i = \sigma_i v_i$.

The singular values $\sigma_i$ can be geometrically interpreted as the lengths of the semiaxis of an hyper-ellipsoid $E := \{Rx : \| x \| = 1\}$. The semiaxis directions are defined by $u_i$ whereas their lengths are the singular values.

From the Corollary1.1 it follows that:

$$R^t R v_i = \sigma_i R^t u_i = \sigma_i^2 v_i$$
$$R R^t u_i = \sigma_i^2 u_i$$

Therefore, there is an intimate connection between the SVD of $R$ and eigensystem of the symmetric matrices as $RR^t$ and $R^tR$.

Finding the *Low-rank approximation* of $R$ means to minimize the *Frobenius* norm[1] of $\| R - X \|_F$ such that $\mathrm{rank}(X) = k$, with $1 \leqslant k \leqslant r = \mathrm{rank}(R)$.

The SVD naturally provide a best k-rank approximation:

**Theorem 2** (Low-rank approximation). *Be* $\hat{R}_k$ *a* $k$*-rank approximation of* $R$, *given by setting to zero all the* $r - k$ *trailing singular values of* $R$, *i.e.:*

$$\hat{R}_k = U\hat{\Sigma}_k V^t, \hat{\Sigma}_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$$

*Then the minimal error is given by the Euclidean norm of the singular values that have been zeroed out in the process:*

$$\| R - \hat{R}_k \|_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2}.$$

1 The Frobenius norm is a matrix norm of an $m \times n$ matrix $A$ defined as (Golub and Van Loan, 1996):

$$\| A \|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2}.$$

## 3.2 APPLYING SVD ON POLARBEAR DATASET

In order to build the deflation subspace containing common informations of several days of observations we apply the SVD following two different methods. The first one comes straightforwardly: it computes the best approximation RHS by means of the SVD on the matrix composed by several $N_{rhs}$ RHSs, see Section 3.2.1. The second one performs $N_{rhs}$ Arnoldi runs and then builds the two-level preconditioner from the SVD applied to the matrix whose columns are the $N_{rhs}$ deflation subspace basis, see details in Section 3.2.2.

### 3.2.1 *SVD onto RHSs*

We performed a SVD onto the *RHS matrix*, $R$, a $N_p \times N_{rhs}$ matrix, whose columns are several RHSs, $R = (b_0|b_1|\dots|b_{N_{rhs}})$. We then compute the SVD onto the matrix $R$ and selected the left singular vectors, (from the columns of $U$, see Equation 19) related to the most important singular values $\tilde{\sigma}$. Since $\sigma_i^2$ are eigenvalues of $RR^t$, we selected the $\tilde{\sigma}$ that satisfy:

$$\tilde{\sigma}^2 \to \frac{\sigma_i^2}{E_{tot}} \geqslant 0.8. \tag{20}$$

i.e. the ones containing most of the energy (defined as $E_{tot} := \sum_k \sigma_k^2$ ), namely 80%.

The best estimate RHS of $R$, $b_{best}$ was then assigned to the left singular vector $\tilde{U}$ related to the selected singular values $\tilde{\sigma}$ which satisfy the Equation 20.

The SVD given the size of our problem ($N_p \sim 10^4$ and $N_{rhs} \lesssim 10$) does not require lot of time since its complexity is $\mathcal{O}(N_p^2)$ operations.[2]

We computed the SVD on $R$ given two choices of $N_{rhs} = 4, 10$ and we have found only one singular value satisfying Equation 20. Thus, chosen a value for $N_{rhs}$, we can summarize the procedure pursued as follows:

1. get the best estimate RHS as $b_{best} = \tilde{U}$ via the SVD on the RHS matrix;
2. run the Arnoldi algorithm in order to compute the deflation subspace basis. As matrix $B$, we choose $M_{BD}A_0$ (see notation as in Equation 16) where $A_0$ is the system matrix associated to one of the RHSs, columns of $R$. The Arnoldi iterations start with $b_{best}$;
3. build the deflation basis $Z_D$ and the two-level preconditioner;
4. run the PCG with $M_{2l}$, computed in the previous step, to all the data as we did in Section 2.4.

2 The computational cost of `Numpy SVD` for a dense matrix is $2(n^2m + m^2n)$. In our case $\sim 2nm^2$since $m \gg n$.

The statistics of both the cases are summarized in Table 3, Table 4 and Figure 9. As one can notice both the cases are very similar to the two-level preconditioner performances in Section 2.4.

This was quite expected since the inner core of Arnoldi algorithm is to iterate via the power method on the Krylov subspace $\mathcal{K}_k(B, b)$. The basis on this space depends more on the matrix $B$ than on the choice of the initial vector $b$.

Therefore, though it is useful to build a deflation subspace encoding informations of several daily RHS we think could more helpful to also consider the degeneracies induced by the scanning strategy, different weather conditions (such as wind speed and pressure of water vapour) during several days of observation.

### 3.2.2 *SVD onto Deflation Subspaces*

To encompass the issue spotted at the end of Section 3.2.1, we followed a procedure very similar to the one presented in the previous section: the difference is in the way we compute the final basis of the deflation subspace. In this section our aim is to build a two-level preconditioner with a deflation basis which takes into account as many approximated eigenvectors as we could with a small number of RHSs. The starting point of the procedure is the very same as the first step in Section 3.2.1. Assuming we have chosen a set of $N_{rhs}$ RHSs, as in 3.2.1:

1. get the best estimate RHS as $b_{best} = \tilde{U}$;
2. run the Arnoldi algorithm $N_{rhs}$ times, with the same Arnoldi starting vector $b_{best}$ but with $N_{rhs}$ Arnoldi matrices: $B_i = M_{BD}A_i$, with $i = 1, \ldots, N_{rhs}$; [3]
3. each Arnoldi run yields $r_i$ Ritz eigenvectors $Z_{D_i}$ and we store all of them into a matrix,$Z_{all}$;
4. perform SVD onto the matrix $Z_{all} = U_Z \Sigma_Z V_Z^t$ ;
5. set as the deflation basis $Z_D \equiv U_Z$ and build the two-level preconditioner from $Z_D$;
6. run the PCG with this two-level preconditioner to the whole set of data.

As in Section 3.2.1, we run this procedure for two different choices of $N_{rhs} = 4, 10$. We summarize the results in Table 4 and in Figure 9. Although this procedure requires to perform $N_{rhs}$ runs Arnoldi algorithm ($\sim 20 \div 30$ minutes each), they can be distributed among an equal amount of processing elements and executed in parallel. However, this is an extra time which

3 Recalling the definition of $A$ in (7), it does depend daily on the scanning strategy pursued via the pointing matrix $P_i$ as well as the noise weights and filters defined in Equation 5.

should be added to execution time of the PCG (in Table 4), which is basically the same as the one required for the cases described in Section 3.2.1 and 2.4 since the computation has been performed among $N_{rhs}$ processes. It took ~ **5 cpuh**[4].

As one can notice in Figure 9 and in Table 4, this procedure improves the convergence performances with a Speedup of 2 (with respect to the Jacobi preconditioner see Section 2.4). However, as we did in Chapter 2, it may be more helpful to look at the median values summarized in Table 3. They are consistent within the error bars to the values shown in Table 4.

Moreover, we have found that by increasing $N_{rhs}$ it may be helpful in terms of convergence iteration steps; on the other hand one has to take into account the computational resources required to perform $N_{rhs}$ Arnoldi runs, as we have noticed above. A trade off among these two key-aspects can be found by considering $N_{rhs}$ not larger than 20. In Figure 9 one can notice how the peak moves toward smaller values of execution time and iteration steps as one considers 4 (top) and 10 (bottom) RHSs.

| | $N_{rhs}$ | PCG execution time [sec] | Iteration steps | $\| r^k \| / \| b \| (\times 10^{-8})$ |
|---|---|---|---|---|
| SVD on RHS | 4 | 256.1 | 21 | 6.5 |
| SVD on RHS | 10 | 248.3 | 21 | 6.4 |
| SVD on $Z_D$ | 4 | 224.2 | 18 | 6.5 |
| SVD on $Z_D$ | 10 | 175.4 | 14 | 5.2 |

**Table 3:** Median values of PCG runs with the two-level preconditioner computed by means of SVD on several RHSs as described in Section 3.2.1 and 3.2.2.

| | $N_{rhs}$ | PCG execution time [sec] | Iteration steps | $\| r^k \| / \| b \| (\times 10^{-8})$ |
|---|---|---|---|---|
| SVD on RHS | 4 | $268.5 \pm 81.7$ | $21.7 \pm 6.2$ | $6.4 \pm 1.8$ |
| SVD on RHS | 10 | $263.4 \pm 89.5$ | $22.1 \pm 7.0$ | $6.5 \pm 1.9$ |
| SVD on $Z_D$ | 4 | $247.1 \pm 104.2$ | $19.6 \pm 7.9$ | $6.4 \pm 1.9$ |
| SVD on $Z_D$ | 10 | $186.8 \pm 74.8$ | $14.9 \pm 6.0$ | $5.6 \pm 2.1$ |

**Table 4:** Mean and standard deviation of PCG runs with the two-level preconditioner computed by means of SVD on several RHSs as described in Section 3.2.1 and 3.2.2.

4 Of course, the number of cpuh increases of a factor $N_{rhs}$, with respect to the one referred in Section 2.4.

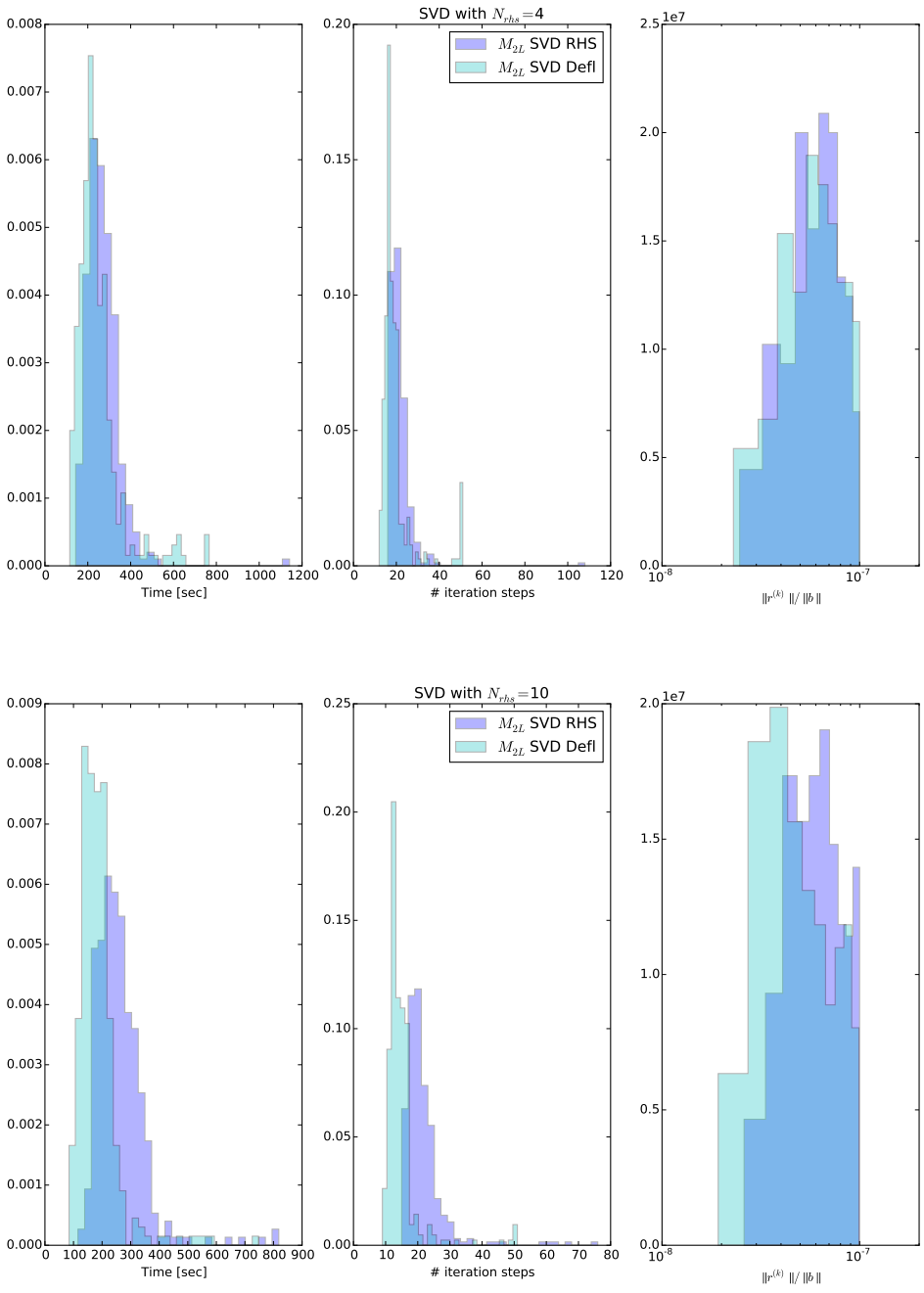


**Figure 9:** Normed histograms of the PCG runs with the two-level preconditioner computed by means of SVD on several RHSs as described in sections 3.2.1 and 3.2.2. The top (bottom) panel compare the runs with $N_{rhs} = 4(10)$.

# 4

# CONCLUSIONS

Since laste decades, the CMB map-making problem has started to be a numerical in order to quickly produce very accurate maps. This is due becasue of the cosmological implications related to the high energy universe encoded in CMB polarization. The current (as well as the forthcoming) CMB experiments are observing the cosmic signal from the ground (mostly at the Atacama Desert (Chile), and in the Antarctica).

We developed the COSMOMAP2 package whose aim is to create an interface to the Polarbear data set, process it by means of linear operators, and solve iteratively the map-making problem with the PCG. The code and its documentation are already available online in a public repository.

We implemented two preconditioners as linear operators: the Jacobi and the two-level preconditioners and we compared the performances of both on the same Polarbear data set simulated on the first seasons of data (from 2012 to 2013) in Chapter 2. They are in agreement with the results shown by Szydlarski et al. (2014).

In this work, finally, we propose a further step which improves the performances of the two-level preconditioner by constructing a common basis of the Deflation subspace by means of SVD. We tested several options and we have found a speedup of 2. Moreover, by means of this procedure, we have found a way to improve the *a priori* construction of the $M_{2l}$ preconditioner and it will be very useful in order to estimate noise maps from Monte-Carlo simulations.

However, in the next months we are planning to apply such a procedure to the Polarbear dataset. Previous results shown that the Jacobi PCG $M_{BD}$ when applied to real data does not converge under a $10^{-3}$ threshold. This

is due to the presence of small eigenvalues and degeneracies generating a sort of *iteration plateau* preventing the residuals $\| r^{(k)} \|$ to go further beyond to a lower threshold than $10^{-3}$ . We have seen this plateau-like feature in our runs on simulated data, see solid red line in Figure 5.

The two-level preconditioner is meant (by definition ) to get rid of these small eigenvalues and we do not expect such a feature. Hence, the need of having accurate maps in wide observational sky-patches can be fully achieved by the $M_{2l}$ preconditioner with optimal performances.

# APPENDIX

# A

# ITERATIVE METHODS

The general idea of iterative methods is to construct a sequence of vectors $x^{(k)}$ such that:

$$\lim_{k\to\infty} x^{(k)} = x,$$

where $x$ is the solution to eq.(3). The iterative process is stopped when it is reached a fixed tolerance: $\| x^{(k+1)} - x^{(k)} \| < \epsilon$.

The iterative methods we consider are of the form:

$$\text{given} \quad x^{(0)},\ x^{(k+1)} = Bx^{(k)} + f, \quad k \geqslant 0, \tag{21}$$

having denoted $B$ an *iteration matrix*, (depending on $A$) and $g$ a vector depending on $A$ and the right hand side $b$. An iterative method of the form (21) is said to be *consistent* if both $B$ and $g$ satisfy at the convergence $x = Bx + f$.

## A.1 PRECONDITIONED CONJUGATE GRADIENT METHOD

A general way of setting up an iterative method is based on the decomposition of the matrix $A$ of the form $A = M_P - (M_P - A)$, where $M_P$ is a non singular and suitable matrix called *Preconditioner* of $A$. Hence by applying this decomposition on eq.(3),

$$Ax = b \Leftrightarrow M_P x = (M_P - A)\, x + b$$

which is of the form (21), with

$$B = {M_P}^{-1} (M_P - A) = I - {M_P}^{-1} A \quad \text{and} \quad f = {M_P}^{-1} b.$$

Thus, eq.(21) can be written as:

$$\begin{aligned} x^{(k+1)} =& Bx^{(k)} + g \\ =& \left(I - {M_P}^{-1} A\right) x^{(k)} + {M_P}^{-1} b \\ =& x^{(k)} + {M_P}^{-1} r^{(k)} \end{aligned}$$

$$\Rightarrow x^{(k+1)} - x^{(k)} = {M_P}^{-1} r^{(k)},$$

where $r^{(k)} = b - Ax^{(k)}$ is the *residual* vector at the k-th iteration. We can generalise this method as follows:

$$x^{(k+1)} - x^{(k)} = \alpha_k {M_P}^{-1} r^{(k)} \tag{22}$$

where $\alpha_k \neq 0$ is a parameter that improves the convergence of the series $x^{(k)}$ and generally varies during the iterations. The (22) is called *dynamic preconditioned Richardson method*. To summarize this method requires at each $k+1$-th step the following operations:

- find the *preconditioned residuals*, $z^{(k)}$ by solving the linear system $M_P z^{(k)} = r^{(k)}$;
- compute the acceleration parameter $\alpha_k$;
- update the solution $x^{(k+1)} = x^{(k)} + \alpha_k z^{(k)}$;
- update the residual $r^{(k+1)} = r^{(k)} - \alpha_k A z^{(k)}$.

In the previous steps one could notice the accelerator parameter, $\alpha_k$ playing a key role through all the iterations. In the special case of symmetric and positive definite matrices one can demonstrate that the optimal choice for it is:

$$\alpha_k = \frac{\left(z^{(k)}\right)^T r^{(k)}}{\left(z^{(k)}\right)^T A z^{(k)}}, \quad k \geqslant 0,$$

this method is also called *Preconditioned gradient method*. It is called *gradient* method since to solve the system (3) means to solve the minimizer of the following quadratic form:

$$\Phi(y) = \frac{1}{2} y^T A y - y^T b,$$

which is called *energy of the system*. The gradient of $\Phi$ is given by:e

$$\nabla\Phi(y) = \frac{1}{2}\left(A^T + A\right) y - b = Ay - b.$$

Conversely, if $\nabla\Phi(x) = 0$ then $x$ is solution of our linear system.

The method we are dealing with is called *conjugate* gradient method and it is more efficient and effective than the preconditioned gradient method. The former is fully based on the latter but there is an extra-condition holding during each iteration: *the direction set by the residual vector at each iteration has to be perpendicular with respect to the direction of the previous ones*.

In other words, let $x^{(0)}$ be given (usually 0), from it we compute $r^{(0} = b - Ax^{(0)}$, $z^{(0)} = M_P{}^{-1} r^{(0)}$ and $p^{(0)} = z^{(0)}$, then for $k \geqslant 0$,:

$$\alpha_k = \frac{\left(p^{(k)}\right)^T r^{(k)}}{\left(p^{(k)}\right)^T A p^{(k)}},$$

Hence compute $x^{(k+1)}$, $r^{(k+1)}$, $z^{(k+1)}$ as we did in the previous section and finally to ensure orthogonality between $p^{(k)}$ and $p^{(k+1)}$ we introduce $\beta_k$ in such a way that:

$$\beta_k = \frac{\left(Ap^{(k)}\right)^T z^{(k+1)}}{\left(Ap^{(k)}\right)^T p^{(k)}}, \tag{23}$$

$$p^{(k+1)} = z^{(k+1)} - \beta_k p^{(k)}. \tag{24}$$

Finally the error estimation is given by:

$$\| x^{(k)} - x \| \leqslant \frac{2c^k}{1 + 2c^{2k}} \| x^{(0)} - x \| \quad k \geqslant 0 \tag{25}$$

where c is defined by the *condition number*[1]:

$$c = \frac{\sqrt{\kappa(P^{-1}A)} - 1}{\sqrt{\kappa(P^{-1}A)} + 1}.$$

1 $\kappa(P^{-1}A)$ is the condition number. It measures how much the output value of the function can change for a small change in the input argument. IT is defined as a product of two operator norms: $\kappa(A) = \| A^{-1} \| \cdot \| A \|$

# BIBLIOGRAPHY


J. Borrill. MADCAP - The Microwave Anisotropy Dataset Computational Analysis Package. *ArXiv Astrophysics e-prints*, November 1999.

O. Doré, R. Teyssier, F. R. Bouchet, D. Vibert, and S. Prunet. MAPCUMBA: A fast iterative multi-grid map-making algorithm for CMB experiments. *Astronomy and Astrophysics*, 374:358–370, July 2001. doi: 10.1051/0004-6361:20010692.

D. J. Fixsen, E. S. Cheng, J. M. Gales, J. C. Mather, R. A. Shafer, and E. L. Wright. The Cosmic Microwave Background Spectrum from the Full COBE FIRAS Data Set. *Astrophysical Journal*, 473:576, December 1996. doi: 10.1086/178173.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

S. Naess, M. Hasselfield, J. McMahon, M. D. Niemack, G. E. Addison, P. A. R. Ade, R. Allison, M. Amiri, N. Battaglia, J. A. Beall, F. de Bernardis, J. R. Bond, J. Britton, E. Calabrese, H.-m. Cho, K. Coughlin, D. Crichton, S. Das, R. Datta, M. J. Devlin, S. R. Dicker, J. Dunkley, R. Dünner, J. W. Fowler, A. E. Fox, P. Gallardo, E. Grace, M. Gralla, A. Hajian, M. Halpern, S. Henderson, J. C. Hill, G. C. Hilton, M. Hilton, A. D. Hincks, R. Hlozek, P. Ho, J. Hubmayr, K. M. Huffenberger, J. P. Hughes, L. Infante, K. Irwin, R. Jackson, S. Muya Kasanda, J. Klein, B. Koopman, A. Kosowsky, D. Li, T. Louis, M. Lungu, M. Madhavacheril, T. A. Marriage, L. Maurin, F. Menanteau, K. Moodley, C. Munson, L. Newburgh, J. Nibarger, M. R. Nolta, L. A. Page, C. Pappas, B. Partridge, F. Rojas, B. L. Schmitt, N. Sehgal, B. D. Sherwin, J. Sievers, S. Simon, D. N. Spergel, S. T. Staggs, E. R. Switzer, R. Thornton, H. Trac, C. Tucker, M. Uehara, A. Van Engelen, J. T. Ward, and E. J. Wollack. The Atacama Cosmology Telescope: CMB polarization at $200 < l < 9000$. *Journal of Cosmology and Astroparticle Physics*, 10:007, October 2014. doi: 10.1088/1475-7516/2014/10/007.

S. P. Oh, D. N. Spergel, and G. Hinshaw. An Efficient Technique to Determine the Power Spectrum from Cosmic Microwave Background Sky Maps. *Astrophysical Journal*, 510:551–563, January 1999. doi: 10.1086/306629.

A. A. Penzias and R. W. Wilson. A Measurement of Excess Antenna Temperature at 4080 Mc/s. *Astrophysical Journal*, 142:419–421, July 1965. doi: 10.1086/148307.

Planck Collaboration, P. A. R. Ade, N. Aghanim, M. I. R. Alves, C. Armitage-Caplan, M. Arnaud, M. Ashdown, F. Atrio-Barandela, J. Aumont, H. Aussel, and et al. Planck 2013 results. I. Overview of products and scientific results. *Astronomy and Astrophysics*, 571:A1, November 2014. doi: 10.1051/0004-6361/201321529.

Davide Poletti, Giulio Fabbian, Maude Le Jeune, Julien Peloton, Kam Arnold, Carlo Baccigalupi, Darcy Barron, Shawn Beckman, Julian Borrill, Scott Chapman, Yuji Chinone, Ari Cukierman, Anne Ducout, Tucker Elleflot, Josquin Errard, Stephen Feeney, Neil Goeckner-Wald, John Groh, Grantland Hall, Masaya Hasegawa, Masashi Hazumi, Charles Hill, Logan Howe, Yuki Inoue, Andrew H. Jaffe, Oliver Jeong, Nobuhiko Katayama, Brian Keating, Reijo Keskitalo, Theodore Kisner, Akito Kusaka, Adrian T. Lee, David Leon, Eric Linder, Lindsay Lowry, Frederick Matsuda, Martin Navaroli, Hans Paar, Giuseppe Puglisi, Christian L. Reichardt, Colin Ross, Praween Siritanasak, Nathan Stebor, Bryan Steinbach, Radek Stompor, Aritoki Suzuki, Osamu Tajima, Grant Teply, and Nathan Whitehorn. Making maps of Cosmic Microwave Background polarization for B-mode studies: the POLARBEAR example. *eprint arXiv:1608.01624*, pages 1–26, aug 2016. URL http://arxiv.org/abs/1608.01624.

G. F. Smoot, C. L. Bennett, A. Kogut, E. L. Wright, J. Aymon, N. W. Boggess, E. S. Cheng, G. de Amici, S. Gulkis, M. G. Hauser, G. Hinshaw, P. D. Jackson, M. Janssen, E. Kaita, T. Kelsall, P. Keegstra, C. Lineweaver, K. Loewenstein, P. Lubin, J. Mather, S. S. Meyer, S. H. Moseley, T. Murdock, L. Rokke, R. F. Silverberg, L. Tenorio, R. Weiss, and D. T. Wilkinson. Structure in the COBE differential microwave radiometer first-year maps. *Astrophysical Journal, Letters*, 396:L1–L5, September 1992. doi: 10.1086/186504.

R. Stompor, A. Balbi, J. D. Borrill, P. G. Ferreira, S. Hanany, A. H. Jaffe, A. T. Lee, S. Oh, B. Rabii, P. L. Richards, G. F. Smoot, C. D. Winant, and J.-H. P. Wu. Making maps of the cosmic microwave background: The MAXIMA example. *Physical Review D*, 65(2):022003, January 2002. doi: 10.1103/PhysRevD.65.022003.

M. Szydlarski, L. Grigori, and R. Stompor. Accelerating the cosmic microwave background map-making procedure through preconditioning. *Astronomy and Astrophysics*, 572:A39, December 2014. doi: 10.1051/0004-6361/201323210.

M. Tegmark. How to Make Maps from Cosmic Microwave Background Data without Losing Information. *Astrophysical Journal, Letters*, 480:L87–L90, May 1997a. doi: 10.1086/310631.

M. Tegmark. CMB mapping experiments: A designer's guide. *Physical Review D*, 56:4514–4529, October 1997b. doi: 10.1103/PhysRevD.56.4514.

The Polarbear Collaboration: P. A. R. Ade, Y. Akiba, A. E. Anthony, K. Arnold, M. Atlas, D. Barron, D. Boettger, J. Borrill, S. Chapman, Y. Chinone, M. Dobbs, T. Elleflot, J. Errard, G. Fabbian, C. Feng, D. Flanigan, A. Gilbert, W. Grainger, N. W. Halverson, M. Hasegawa, K. Hattori, M. Hazumi, W. L. Holzapfel, Y. Hori, J. Howard, P. Hyland, Y. Inoue, G. C. Jaehnig, A. H. Jaffe, B. Keating, Z. Kermish, R. Keskitalo, T. Kisner, M. Le Jeune, A. T. Lee, E. M. Leitch, E. Linder, M. Lungu, F. Matsuda, T. Matsumura, X. Meng, N. J. Miller, H. Morii, S. Moyerman, M. J. Myers, M. Navaroli, H. Nishino, A. Orlando, H. Paar, J. Peloton, D. Poletti, E. Quealy, G. Rebeiz, C. L. Reichardt, P. L. Richards, C. Ross, I. Schanning, D. E. Schenck, B. D. Sherwin, A. Shimizu, C. Shimmin, M. Shimon, P. Siritanasak, G. Smecher, H. Spieler, N. Stebor, B. Steinbach, R. Stompor, A. Suzuki, S. Takakura, T. Tomaru, B. Wilson, A. Yadav, and O. Zahn. A Measurement of the Cosmic Microwave Background B-mode Polarization Power Spectrum at Sub-degree Scales with POLARBEAR. *Astrophysical Journal*, 794:171, October 2014. doi: 10.1088/0004-637X/794/2/171.

B. D. Wandelt and F. K. Hansen. Fast, exact CMB power spectrum estimation for a certain class of observational strategies. *Physical Review D*, 67 (2):023001, January 2003. doi: 10.1103/PhysRevD.67.023001.

E. L. Wright. Scanning and Mapping Strategies for CMB Experiments. *ArXiv Astrophysics e-prints*, November 1996.