

Multi-resolution Bayesian CMB component separation through Wiener filtering with a pseudo-inverse preconditioner[★]

D. S. Seljebotn¹, T. Bærland², H. K. Eriksen¹, K.-A. Mardal², and I. K. Wehus¹

¹ Institute of Theoretical Astrophysics, University of Oslo, PO Box 1029 Blindern, 0315 Oslo, Norway
e-mail: h.k.k.eriksen@astro.uio.no

² Department of Mathematics, University of Oslo, PO Box 1080 Blindern, 0316 Oslo, Norway

Received 4 October 2017 / Accepted 7 May 2019

ABSTRACT

We present a Bayesian model for multi-resolution component separation for cosmic microwave background (CMB) applications based on Wiener filtering and/or computation of constrained realizations, extending a previously developed framework. We also develop an efficient solver for the corresponding linear system for the associated signal amplitudes. The core of this new solver is an efficient preconditioner based on the pseudo-inverse of the coefficient matrix of the linear system. In the full sky coverage case, the method gives an increased speed of the preconditioner, and it is easier to implement in terms of practical computer code. In the case where a mask is applied and prior-driven constrained realization is sought within the mask, this is the first time full convergence has been achieved at the full resolution of the *Planck* data set.

Key words. methods: statistical – cosmic background radiation

1. Introduction

The cosmic microwave background (CMB) has proved invaluable for our understanding of early universe physics. Providing a direct image of the cosmos as it appeared only 380 000 years after the Big Bang, these fluctuations represent the initial conditions for all subsequent structure formation, eventually leading to the universe we observe today, consisting of galaxies, solar systems, and planets. Furthermore, through detailed statistical analysis of these temperature variations, cosmologists are today able to constrain many important cosmological parameters to percent-level precision (e.g., [Planck Collaboration XIII 2016](#)).

Steadily improving microwave detector technology has enabled this progress, resulting from a host of ground-based, sub-orbital, and satellite experiments. Amongst the latter are the most recent NASA's *Wilkinson* Microwave Anisotropy Probe (WMAP) mission ([Bennett et al. 2013](#)) and ESA's *Planck* mission ([Planck Collaboration I 2016](#)). The primary goal of all these experiments is to produce the cleanest possible image of the true cosmological CMB sky. However, when observing the sky at millimeter wavelengths, many other physical emission mechanisms contribute to the total sky brightness besides the CMB. The most important ones on large angular scales are synchrotron, free-free, thermal, and spinning dust, and CO emission, all of which arise from particles that are present within our own galaxy, the Milky Way (e.g., [Planck Collaboration X 2015](#)). On small angular scales, radio and far-infrared emission from distant galaxies are important, as is the Sunyaev–Zeldovich effect arising from hot gas in massive clusters. Collectively, all these signal are called “foregrounds” with respect to the CMB.

The process of extracting the cosmological signal from foreground-contaminated observations is often called CMB

component separation, which informally refers to combining observations taken at different frequencies into a single estimate of the true cosmological sky. A wide range of numerical and statistical methods have been developed for this purpose, including parametric methods ([Eriksen et al. 2008](#); [Errard et al. 2011](#)), various internal linear combination methods ([Bennett et al. 2003](#); [Eriksen et al. 2004a](#); [Remazeilles et al. 2011](#); [Basak & Delabrouille 2012](#)), spectral matching methods ([Cardoso et al. 2008](#)), template fitting methods ([Fernández-Cobos et al. 2012](#)), and many others.

Component separation becomes increasingly important as the sensitivity of CMB detectors continues to improve, and the foreground-induced uncertainties make up an increasingly large fraction of the total CMB uncertainty budget. Indeed, for the latest generation CMB experiments, which target the predicted B-mode polarization signal arising from primordial gravitational waves produced during the Big Bang, component separation is one of the single most important challenges (e.g., [BICEP2 Collaboration 2014](#)).

In this paper we consider the Bayesian component separation framework pioneered by [Jewell et al. \(2004\)](#), [Wandelt et al. \(2004\)](#), and [Eriksen et al. \(2008\)](#), and in particular efficient evaluation of the so-called Wiener filter solution employed by codes such as Commander ([Eriksen et al. 2004b](#)). This framework has been established by the *Planck* collaboration for astrophysical component separation and CMB extraction ([Planck Collaboration X 2015](#)). However, the Commander implementation that was used in the *Planck* 2015 (and earlier) analysis suffers from one important limitation; it requires all frequency channels to have the same effective instrumental beam, or “point spread function”. In general, the angular resolution of a given sky map is typically inversely proportional to the wavelength, and the solution to this problem adopted by the *Planck* 2015 Commander analysis was simply to smooth all frequency maps to a common (lowest) angular resolution. This, however,

[★] The prototype benchmark code is available at <https://github.com/dagss/cmbcr>

implies an effective loss of both sensitivity and angular resolution, and is as such highly undesirable. To establish an effective algorithm that avoids this problem is the main goal of the current paper, and the methods presented in this paper have already been used extensively in the latest *Planck* 2018 data release, as described by [Planck Collaboration IV \(2019\)](#).

First, we start by presenting a native multi-resolution version of the Bayesian CMB component separation model. This is a straightforward extension of the single-resolution model presented in [Eriksen et al. \(2008\)](#). Second, we construct a novel solver for the resulting linear system, based on the pseudo-inverse of the corresponding coefficient matrix. We find that this solver outperforms existing solvers in all situations we have applied it to. Additionally, the full sky version of the preconditioner is easier to implement in terms of computer code than the simple diagonal preconditioner that is most commonly used in the literature ([Eriksen et al. 2004b](#)). The big advantage, however, is most clearly seen in the presence of partial sky coverage, where the speed-ups reach factors of hundreds. For full sky coverage, where a diagonal preconditioner performs reasonably, the improvement is a more modest, a factor of 2–3.

The previous Commander implementation ([Eriksen et al. 2004b](#)) employs the conjugate gradient (CG) method to solve the filter equation. (also often referred to as the “constrained realization” (CR) system when adding a stochastic term on the right-hand side of the equation), using a preconditioner based on computing matrix elements in spherical harmonic domain. While this approach worked well enough for WMAP, the convergence rate scales with the signal-to-noise ratio of the experiment, and for the sensitivity of *Planck*, it becomes impractical in the case of partial sky coverage, requiring several thousand iterations for convergence, if it converges at all.

[Elsner & Wandelt \(2013\)](#) develop a messenger field algorithm that provides an approximate solution to the same problem, which should be sufficient for many purposes. This approximate algorithm was recently improved by [Kodi Ramanah et al. \(2017\)](#). An example of the quality of the approximation can be seen in Fig. 4 of [Kodi Ramanah et al. \(2017\)](#); the difference from the exact Wiener-filter is on the level 1%–10%, with the differences being concentrated around the edges of the mask. The algorithm in this paper, in contrast, computes the exact Wiener filter at a lower computational cost.

To our knowledge the multi-level, pixel-domain solver described by [Seljebotn et al. \(2014\)](#) represents the state of the art for exact solvers prior to this paper. However, the multi-level solver has some weaknesses that became apparent in our attempt to generalize it for the purpose of component separation. First, due to only working locally in pixel domain, it is ineffective in deconvolving the signal on high ℓ , where the spherical harmonic transfer function of the instrumental beam falls below $b_\ell = 0.2$ or so. In the case of [Seljebotn et al. \(2014\)](#) this was not a problem because the solution is dominated by a Λ CDM-type prior on the CMB before reaching these scales. However, for the purposes of component separation one wants to apply no prior or rather weak priors, and so the solver in [Seljebotn et al. \(2014\)](#) fails to converge. The second problem is that it is challenging to work in pixel domain with multi-resolution data for which the beam sizes vary with a factor of 10, as is the case in *Planck*. Thirdly, extensive tuning was required on each level to avoid ringing problems. Finally, the algorithm in [Seljebotn et al. \(2014\)](#) requires extensive memory-consuming pre-computations.

In contrast, the solver developed in this paper (1) offers very cheap pre-computations; (2) does not depend on applying a prior; (3) is robust with respect to the choice of statistical priors

on the component amplitudes; and (4) has much less need for tuning. The solver combines a number of techniques, and fundamentally consists of two parts:

1. A pseudo-inverse preconditioner, developed in Sect. 3 and Appendix A, which improves on the simple diagonal preconditioner by better incorporating inhomogeneities in the RMS maps of the data. This performs very well on its own in cases with full sky coverage.
2. A mask-restricted solver, which should be applied in addition in cases of partial sky coverage. This solver is developed in Sect. 4. By solving the system under the mask as a separate sub-step, we converge in about a hundred CG iterations rather than thousands of iterations.

These two techniques are somewhat independent, and it would be possible to use the mask-restricted solver together with the older diagonal preconditioner. Still we present them together in this paper in order to establish what we believe is the new state of the art for solving the CMB component separation and/or constrained realization problem.

2. Bayesian multi-resolution CMB component separation

2.1. Spherical harmonic transforms in linear algebra

We assume that the reader is familiar with spherical harmonic transforms (SHTs), and simply note that they are the spherical analogue of Fourier transforms (for further details see, e.g., [Reinecke & Seljebotn 2013](#)). The present work requires us to be very specific about the properties of these transforms as part of linear systems. We will write Y for the transform from harmonic domain to pixel domain (spherical harmonic synthesis), and $Y^T W$ for the opposite transform (spherical harmonic analysis). The W matrix is diagonal, containing the per-pixel quadrature weights employed in the analysis integral for a particular grid.

Of particular importance for the present work is that no perfect grid exists on the sphere; all spherical grids have slightly varying distances between grid points. This means that some parts of the sphere will see smaller scales than other parts, and that, ultimately, there is no discrete version of the spherical harmonic transform analogous to the discrete Fourier transforms that maps \mathbb{R}^n to \mathbb{R}^n . Specifically, Y is rectangular and thus not invertible. Usually Y is configured such that the number of rows (pixels) is greater than the number of columns (spherical harmonic coefficients), in which case an harmonic signal is unaltered by a round-trip to pixel domain so that $Y^T W Y = I$. In that case, the converse operator, $Y Y^T W$, is singular, but in a very specific way: it takes a pixel map and removes any scales from it that are above the band-limit L .

2.2. Component separation model

[Eriksen et al. \(2008\)](#) describe a Bayesian model for CMB component separation under the assumption that all observed sky maps have the same instrumental beam and pixel resolution. For full resolution analysis of *Planck* data this is an unrealistic requirement, as the full-width half-maximum (FWHM) of the beam (point spread function) span a large range, from 4.4 to 32 arc-min, and so one loses much information by downgrading data to a common resolution. In this paper we generalize the model to handle sky maps observed with different beams and at different resolutions.

We will restrict our attention to the CMB component and diffuse foregrounds. [Eriksen et al. \(2008\)](#) additionally incorporate

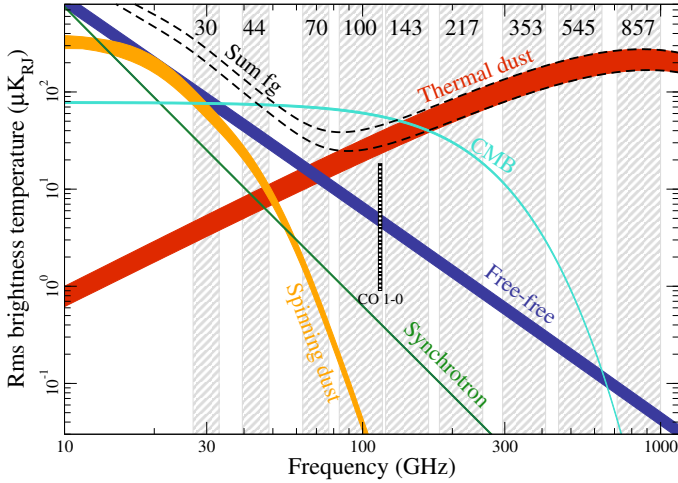


Fig. 1. Illustration of the spectral energy density response of each component in the microwave emission. The shaded bands indicate the nine different observation frequencies of the *Planck* space observatory. Our goal is to create a map of each component in the sky; “CMB” emission, “Thermal dust” emission, and so on, after making assumptions about the spectral behaviour of each component such as we do in this figure. We model the spectral behaviour as slightly different in each pixel; hence it is indicated using bands rather than lines. Figure 2 adds further detail. This figure is reproduced directly from [Planck Collaboration X \(2015\)](#).

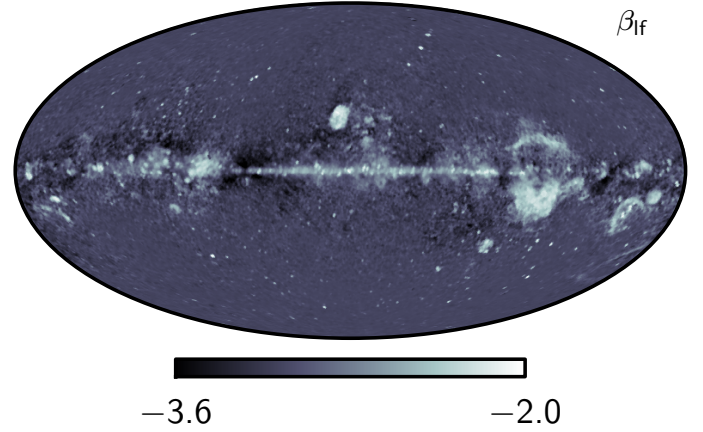


Fig. 2. Example spectral index map $\beta(\hat{n})$. Each pixel value corresponds to the slope in Fig. 1 in that pixel for a single combined low-frequency emission component (using a single component to represent the “Free-free”, “Synchrontron”, and “Spinning dust” emission types indicated in Fig. 1. In reality, each pixel is the sum of multiple slightly different spectral behaviours from different clouds of particles behind one another, but we instead work with a single compromise value for the dominating emission in the given direction. The mixing maps of this paper, $q_{v,k}(\hat{n}_i)$, are taken to be proportional to $\nu^{\beta(\hat{n}_i)}$, where each component k has a different β -map. This figure is reproduced directly from [Planck Collaboration X \(2015\)](#).

template components in the model for linear component separation. These are particularly useful for dealing with point sources, where beam asymmetry is much more noted than for the diffuse foregrounds. Recent versions of Commander sample template amplitudes as an additional Gibbs step, rather than as part of the linear system for component separation, so as to more easily include a positivity constraint on such amplitudes. We will therefore ignore templates in this paper.

The microwave sky is observed as N_{obs} different sky maps \mathbf{d}_v with different instrumental characteristics, and we wish to separate these into N_{comp} distinct diffuse foreground components s_k . The key to achieving this is to specify the spectral energy density (SED) of each component (see Figs. 1 and 2). [Eriksen et al. \(2008\)](#) describe the Gibbs sampling steps employed to fit the SED to data. For the purposes of this paper we will simply assume that the SED information is provided as input, in the form of mixing maps. Our basic assumption, ignoring any instrumental effects, is that the true microwave emission in a direction \hat{n} on the sky, integrated over a given frequency bandpass labelled ν , is given by

$$f_\nu(\hat{n}) = \sum_k q_{\nu,k}(\hat{n}) s_k(\hat{n}), \quad (1)$$

where $s_k(\hat{n})$ represents the underlying component amplitude, and $q_{\nu,k}(\hat{n})$ represents an assumed mixing map. We will model the CMB component simply as another diffuse component, but note that the mixing maps are in that case spatially constant.

We do not observe $f_\nu(\hat{n})$ directly, but rather take as input a pixelized sky map \mathbf{d}_v , where f_ν has been convolved with an instrumental beam and then further contaminated by instrumental noise. To simplify notation we employ a notation with stacked vectors, and write

$$\mathbf{d} \equiv \begin{bmatrix} \mathbf{d}_{30\text{GHz}} \\ \mathbf{d}_{70\text{GHz}} \\ \vdots \end{bmatrix}, \quad \mathbf{s} \equiv \begin{bmatrix} \mathbf{s}_{\text{cmb}} \\ \mathbf{s}_{\text{dust}} \\ \vdots \end{bmatrix}.$$

Our data observation model can then be written

$$\mathbf{d} = \mathbf{P}\mathbf{s} + \mathbf{n}, \quad (2)$$

where \mathbf{P} is an $N_{\text{obs}} \times N_{\text{comp}}$ block-matrix where each block (v, k) projects component s_k to the observed sky map \mathbf{d}_v , and \mathbf{n} represents instrumental noise, and is partitioned in the same way as \mathbf{d} . The noise \mathbf{n}_v is a pixelized map with the same resolution as \mathbf{d}_v , and assumed to be Gaussian distributed with zero mean. For our experiments we also assume that the noise is independent between pixels, so that $\text{Var}(\mathbf{n}_v) \equiv \mathbf{N}_v$ is a diagonal matrix, although this is not a fundamental requirement of the method. We do, however, require that the matrix \mathbf{N}_v^{-1} can somehow be efficiently applied to a vector.

Each component s_k represents the underlying, unconvolved field. In our implementation we work with s_k being defined by the spherical harmonic expansion of $s_k(\hat{n})$, truncated at some band-limit L_k , that is, we assume $s_{\ell,m} = 0$ for $\ell > L_k$. The choice of L_k is essentially a part of the model, and typically chosen to match a resolution that the observed sky maps will support. Additionally, each component s_k may have an associated Gaussian prior $p(s)$, specified through its covariance matrix \mathbf{S} . The role of the prior is to introduce an assumption on the smoothness of s . The prior typically does not come into play where the signal for a component is strong, but in regions that lack the component it serves to stabilize the solution, such that less noise is interpreted as signal. Computationally it is easier to assume the same smoothness prior everywhere on the sky, in which case the covariance matrix $\text{Var}(s) = \mathbf{S}$ is diagonal in spherical harmonic domain with elements given by the spherical harmonic power spectrum, $C_{k,\ell}$. However, this is not a necessary assumption, and we comment on a different type of prior in Sect. 6.

The CMB power spectrum prior, given by $C_{\text{cmb},\ell}$, is particularly crucial. For the purposes of full sky component separation one would typically not specify any prior so as not to bias the CMB. For the purposes of estimating foregrounds, however, or

filling in a CMB realization within a mask, one may insert a fiducial power spectrum predicted by some cosmological models.

We now return to the projection operator \mathbf{P} , which projects each component to the sky. This may be written in the form of a block matrix with each column representing a component k and each row representing a sky observation ν , for example,

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{30\text{ GHz,cmb}} & \mathbf{P}_{30\text{ GHz,dust}} & \dots \\ \mathbf{P}_{70\text{ GHz,cmb}} & \mathbf{P}_{70\text{ GHz,dust}} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad (3)$$

with each block taking the form

$$\mathbf{P}_{\nu,k} = \mathbf{Y}_\nu \mathbf{B}_\nu \widetilde{\mathbf{Q}}_{\nu,k}. \quad (4)$$

The operator \mathbf{Y}_ν denotes spherical harmonic synthesis to the pixel grid employed by \mathbf{d}_ν . We assume in this paper an azimuthally symmetric instrumental beam for each sky map, in which case the beam convolution operator \mathbf{B}_ν is diagonal in spherical harmonic domain with elements $b_{\nu,\ell}$. This transfer function decays to zero as ℓ grows at a rate that fits the band-limit of the grid. Finally $\widetilde{\mathbf{Q}}_{\nu,k}$ is an operator that denotes point-wise multiplication of the input with the mixing map $q_{\nu,k}(\hat{n})$; computationally this should be done in pixel domain, so that

$$\widetilde{\mathbf{Q}}_{\nu,k} = \mathbf{Y}_{\nu,k}^T \mathbf{W}_{\nu,k} \mathbf{Q}_{\nu,k} \mathbf{Y}_{\nu,k}, \quad (5)$$

where $\mathbf{Q}_{\nu,k}$ contains $q_{\nu,k}(\hat{n})$ on its diagonal. The subscripts on the SHT operators indicate that these are defined on a grid specific to this mixing operation. We note that $\widetilde{\mathbf{Q}}_{\nu,k}$ will cause the creation of new small-scale modes, implying that, technically speaking, the band-limit of $\widetilde{\mathbf{Q}}_{\nu,k} \mathbf{s}_k$ is $2L_k$ rather than L_k for a full-resolution mixing map $q_{\nu,k}$. For typical practical system solving, however, the mixing matrices are usually smoother than the corresponding amplitude maps (due to lower effective signal-to-noise ratios), and the model may incorporate an approximation in that $\widetilde{\mathbf{Q}}_{\nu,k}$ truncates the operator output at some lower band-limit. At any rate, the grid used for $\mathbf{Q}_{\nu,k}$ should accurately represent $q_{\nu,k}$ up to this band-limit. For this purpose it is numerically more accurate to use a Gauss-Legendre grid, rather than the HEALPix¹ grid (Górski et al. 2005) that is usually used for \mathbf{d}_ν . The solver of the present paper simply treats this as a modelling detail, and any reasonable implementation works fine as long as $\widetilde{\mathbf{Q}}_{\nu,k}$ is not singular. However, to achieve reasonable efficiency, we do require $q_{\nu,k}(\hat{n})$ to be relatively flat, such that approximating $\widetilde{\mathbf{Q}}_{\nu,k}$ with a simple scalar $q_{\nu,k}$ is a meaningful zero-order representation. In practice we typically find ratios between the maximum and minimum values of $q_{\nu,k}(\hat{n})$ of 1.5–3. In general, the higher the contrast, the slower is the convergence of the solver.

2.3. Constrained realization linear system

We have specified a model for the components \mathbf{s} where the likelihood $p(\mathbf{d})$ and component priors $p(\mathbf{s})$ are Gaussian, and so the Bayesian posterior is also Gaussian (Jewell et al. 2004; Wandelt et al. 2004; Eriksen et al. 2004b, 2008), and given by

$$p(\mathbf{s}|\mathbf{d}, \mathbf{S}) \propto e^{-\frac{1}{2} \mathbf{s}^T (\mathbf{S}^{-1} + \mathbf{P}^T \mathbf{N}^{-1} \mathbf{P})^{-1} \mathbf{s}}. \quad (6)$$

To explore this density we are typically interested in either i) the mean vector, or ii) drawing samples from the density. Both can be computed by solving a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with

$$\mathbf{A} \equiv \mathbf{S}^{-1} + \mathbf{P}^T \mathbf{N}^{-1} \mathbf{P} \quad (7)$$

¹ <http://healpix.jpl.nasa.gov>

and

$$\mathbf{b} \equiv \mathbf{P}^T \mathbf{N}^{-1} \mathbf{d} + \mathbf{P}^T \mathbf{N}^{-1/2} \boldsymbol{\omega}_1 + \mathbf{S}^{-1/2} \boldsymbol{\omega}_2. \quad (8)$$

The vectors $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ should either be

- (i) zero, in which case the solution \mathbf{x} will be the mean $\mathbf{E}(\mathbf{s}|\mathbf{d}, \mathbf{S})$, also known as the Wiener-filtered map; or
- (ii) vectors of variates from the standard Gaussian distribution, in which case the solution \mathbf{x} will be samples drawn $p(\mathbf{s}|\mathbf{d}, \mathbf{S})$. We refer to such samples as the constrained realizations.

For convenience we refer to the system as the constrained realization (CR) system in both cases. The computation of the right-hand side \mathbf{b} is straightforward and not discussed further in this paper; our concern is the efficient solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$.

For current and future CMB experiments, the matrix \mathbf{A} is far too large for the application of dense linear algebra. However, we are able to efficiently apply the matrix to a vector, by applying the operators one after the other, and so we can use iterative solvers for linear systems. Since \mathbf{A} is symmetric and positive definite, the recommended iterative solver is the conjugate gradients (CG) method (see Shewchuk 1994, for a tutorial). One starts with an arbitrary guess, say, $\mathbf{x}_1 = \mathbf{0}$. Then, for each iteration, a residual $\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i$ is computed, and then this residual is used to produce an updated iterate \mathbf{x}_{i+1} that lies closer to the true solution \mathbf{x}_{true} .

The residual \mathbf{r}_i , which is readily available, is used as a proxy for the error, $\mathbf{e}_i = \mathbf{x}_{\text{true}} - \mathbf{x}_i$, which is unavailable as we do not know \mathbf{x}_{true} . The key is that since $\mathbf{A}\mathbf{x}_{\text{true}} = \mathbf{b}$,

$$\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{x}_i = \mathbf{A}(\mathbf{x}_{\text{true}} - \mathbf{x}_i) = \mathbf{A}\mathbf{e}_i,$$

and since \mathbf{A} is linear, reducing the magnitude of \mathbf{r}_i will also lead to a reduction in the error \mathbf{e}_i . In a production run the error \mathbf{e}_i is naturally unavailable, but during development and debugging it is highly recommended to track it. This can be done by generating an \mathbf{x}_{true} , then generating $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}}$, and tracking the error \mathbf{e}_i while running the solver on this input. The benchmark results presented in this paper are generated in this manner.

The number of iterations required by CG depends on how uniform (or clustered) the eigenspectrum of the matrix is. Therefore the main ingredient in a linear solver is a good preconditioner that improves the eigenspectrum. A good preconditioner is a symmetric, positive definite matrix \mathbf{M} such that $\mathbf{M}\mathbf{x}$ can be quickly computed, and where the eigenspectrum of $\mathbf{M}\mathbf{A}$ is as flat as possible, or at least clustered around a few values. Intuitively, \mathbf{M} should in some sense approximate \mathbf{A}^{-1} .

3. Full sky preconditioner based on pseudo-inverses

3.1. Pseudo-inverses

The inspiration for the solver presented below derives from the literature on the numerical solution of saddle-point systems. Our system \mathbf{A} , in the form given in Eq. (7), is an instance of a so-called Schur complement, and approximating the inverse of such Schur complements plays a part in most solvers for saddle-point systems. An excellent review on the solution of saddle-point systems is given in Benzi et al. (2005). The technique we will use was originally employed by Elman (1999) and Elman et al. (2006), who use appropriately scaled pseudo-inverses to develop solvers for Navier-Stokes partial differential equations.

The pseudo-inverse is a generalization of matrix inverses to rectangular and/or singular matrices. In our case, we will deal

with an $m \times n$ -matrix U with linearly independent columns and $m > n$, in which case the pseudo-inverse is given by

$$U^+ \equiv (U^T U)^{-1} U^T.$$

We note that in this case, U^+ is simply the matrix that finds the least-squares solution to the linear system $Ux = b$, i.e., that minimizes $\min_x \|Ux - b\|^2$. This matrix has the property that

$$U^+ U = (U^T U)^{-1} U^T U = I,$$

and so it is a left inverse. Unlike the real inverse however, $U U^+ \neq I$. This follows from U^+ having the same shape as U^T , so that $U U^+$ is necessarily a singular matrix. For small matrices, U^+ can be computed by using the QR-decomposition. In the case that U does not have full rank, the pseudo-inverse has a different definition and should be computed using the singular value decomposition (SVD); the definition above is, however, sufficient for our purposes.

3.2. Full-sky, single component, without prior

For the first building block of our preconditioner we start with a much simpler linear problem. Assuming a model with a single sky map d , a single component, no mixing maps, and no prior, the linear system of Eq. (7) simply becomes

$$B Y^T N^{-1} Y B x = b. \quad (9)$$

We note that Y^T is not the same as spherical harmonic analysis, which in our notation reads $Y^T W$. Without the weight matrix W , Y^T represents adjoint spherical harmonic synthesis, and simply falls out algebraically from the transpose sky projection P^T .

Since the basis of x is in spherical harmonic domain, the beam convolution operator B is diagonal and trivially inverted. The remaining matrix $Y^T N^{-1} Y$ is diagonal in pixel domain, so if only Y had been invertible we could have solved the system directly. Inspired by Elman (1999), we simply pretend that Y is square, and let $Y^T W$ play the role of the inverse, so that

$$(B Y^T N^{-1} Y B)^{-1} \approx B^{-1} Y^T W N W Y B^{-1}. \quad (10)$$

We stress that because Y is not exactly invertible and $Y^T W$ is a pseudo-inverse, this is only an approximation that should be used as a preconditioner inside an iterative solver. For the HEALPix grid in particular, $Y^T W$ is rather inaccurate and we only have $Y^T W Y \approx I$; it is, however, a very good approximation, as can be seen in Fig. 3.

3.3. Full sky, multiple components, with priors

Next, we want to repeat the above trick for a case with multiple sky maps d_v , multiple components x_k , and a prior. We start by assuming that the mixing operators $\tilde{Q}_{v,k}$ can be reasonably approximated by a constant,

$$\tilde{Q}_{v,k} \approx q_{v,k} I,$$

although, as noted above, this is only a matter of computational efficiency of the preconditioner. The optimal value for $q_{v,k}$ is given in Appendix A.2. In the following derivation we will simply assume equality in this statement, keeping in mind that all the manipulations only apply to the preconditioner part of the CG search, and so do not affect the computed solution.

First we note that the matrix of Eq. (7) can be written

$$A = \begin{bmatrix} P^T & I \end{bmatrix} \begin{bmatrix} N^{-1} & \\ & S^{-1} \end{bmatrix} \begin{bmatrix} P \\ I \end{bmatrix}.$$

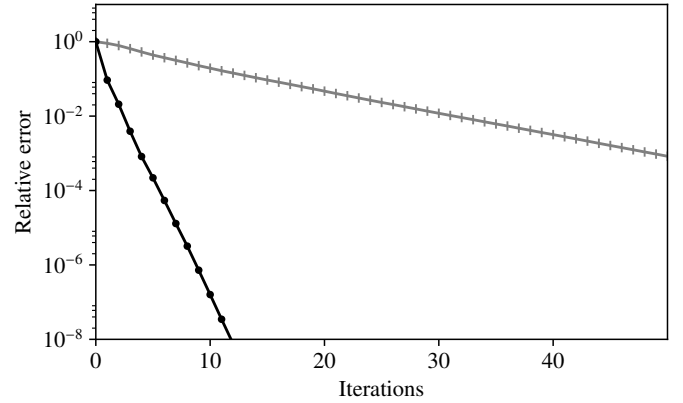


Fig. 3. Convergence of preconditioner of Eq. (10) (black circles) compared to a simple diagonal preconditioner (grey ticks). In this case, we fit a single CMB component to a single 143 GHz *Planck* band without specifying a prior. Section 5 gives further details on the benchmark setup and the diagonal preconditioner.

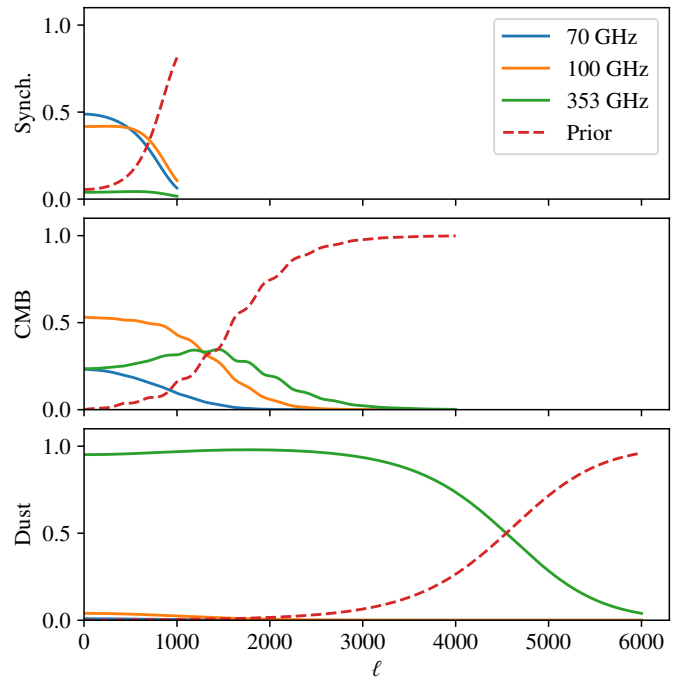


Fig. 4. Visualization of the matrix U in an example setup. For each component k we plot the coefficients along the corresponding column of U , normalized so that the sum is 1 for each ℓ . We note how the CMB has most support from the 100 GHz band for low ℓ , then gradually switches to the 353 GHz band and finally the prior as ℓ increases.

Writing the system in this way makes it evident that in the Bayesian framework, the priors are treated just like another “observation” of the components. These “prior observations” play a bigger role for parts of the solution where the instrumental noise is high, and a smaller role where the instrumental noise is low (see Fig. 4).

The idea is now to further rearrange and re-scale the system in such a way that all information that is not spatially variant is expressed in the projection matrix on the side, leaving a unit-free matrix containing only spatial variations in the middle. Since S^{-1} is assumed to be diagonal, it is trivial to factor it and leave only the identity in the centre matrix. Turning to the inverse-noise term N^{-1} , we find experimentally that re-scaling with a scalar

performed well. Specifically, we define

$$\tilde{N}_v^{-1} \equiv \alpha_v^{-2} Y_v^T N_v^{-1} Y_v,$$

where α takes the value that minimizes $\|\tilde{N}_v^{-1}(\alpha_v) - \mathbb{I}\|_2$. Computing α_v is cheap, and its optimal value is derived in Appendix A.1. A similar idea is used by [Elsner & Wandelt \(2013\)](#), as they split N^{-1} into a spherical harmonic term and a pixel term, but their split is additive rather than multiplicative. Our system can now be written

$$\mathbf{A} = \mathbf{U}^T \mathbf{T} \mathbf{U}, \quad (11)$$

where \mathbf{T} contains the unit-free spatial structure of the system, and \mathbf{U} the spatially invariant, but ℓ -dependent, structure of the system:

$$\mathbf{T} \equiv \begin{bmatrix} \tilde{N}^{-1} & & & \\ & \mathbb{I} & & \\ & & & \\ & & & \end{bmatrix}, \quad \mathbf{U} \equiv \begin{bmatrix} \alpha \mathbf{B} \tilde{\mathbf{Q}} \\ \mathbf{S}^{-1/2} \end{bmatrix}.$$

In order to elucidate the block structure of these matrices, we write out an example with three bands and two components:

$$\mathbf{T} = \begin{bmatrix} \tilde{N}_1^{-1} & & & & & \\ & \tilde{N}_2^{-1} & & & & \\ & & \tilde{N}_3^{-1} & & & \\ & & & \mathbb{I} & & \\ & & & & & \mathbb{I} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \alpha_1 \mathbf{B}_1 \tilde{\mathbf{Q}}_{1,1} & \alpha_1 \mathbf{B}_1 \tilde{\mathbf{Q}}_{1,2} \\ \alpha_2 \mathbf{B}_2 \tilde{\mathbf{Q}}_{2,1} & \alpha_2 \mathbf{B}_2 \tilde{\mathbf{Q}}_{2,2} \\ \alpha_3 \mathbf{B}_3 \tilde{\mathbf{Q}}_{3,1} & \alpha_3 \mathbf{B}_3 \tilde{\mathbf{Q}}_{3,2} \\ \mathbf{S}_1^{-1/2} & \\ & \mathbf{S}_2^{-1/2} \end{bmatrix}.$$

Under the assumption that $\tilde{\mathbf{Q}}_{v,k} = q_{v,k} \mathbb{I}$, the matrix \mathbf{U} is block-diagonal with small blocks of variable size when seen in ℓ - and m -major ordering. Specifically, the entries in the ‘‘data-blocks’’ (v, k') in the top part of \mathbf{U} are given by

$$U_{(\ell,m,v),(\ell',m',k')} = \alpha_v b_{v,\ell} q_{v,k'} \delta_{\ell,\ell'} \delta_{m,m'}, \quad (12)$$

and the entries in the ‘‘prior-blocks’’ (k, k') in the bottom part of \mathbf{U} are given by

$$U_{(\ell,m,k),(\ell',m',k')} = C_{k,\ell}^{-1/2} \delta_{k,k'} \delta_{\ell,\ell'} \delta_{m,m'}. \quad (13)$$

In the event that no prior is present for a component, the corresponding block can simply be removed from the system, or one may equivalently set $C_{k,\ell}^{-1/2} = 0$.

At this point we must consider the multi-resolution nature of our setup. Our model assumes a band-limit L_k for each component, so that each component has $(L_k + 1)^2$ corresponding columns in \mathbf{U} , and, if a prior is used, an additional corresponding $(L_k + 1)^2$ rows. Each sky observation has a natural band-limit L_v where the beam transfer function $b_{v,\ell}$ has decayed so much that including further modes is numerically irrelevant, and so each band has $(L_v + 1)^2$ corresponding rows in \mathbf{U} . When we view \mathbf{U} in the block-diagonal ℓ - and m -major ordering, the block sizes are thus variable. Each band-row only participates for $\ell \leq L_v$, and each component-column and component-row only participates for $\ell \leq L_k$. A way to see this is to consider that the first blocks for $\ell = 0$ have size $(N_{\text{obs}} + N_{\text{comp}}) \times N_{\text{comp}}$ for all m ; then, as ℓ is increased past some L_v , or L_k , corresponding rows or columns disappear from the blocks.

In code it is easier to introduce appropriate zero-padding for $\ell > L_k$ and $\ell > L_v$ so that one can work with a single constant block size. Either way, the numerical results are equivalent. Since \mathbf{U} consists of many small blocks on the diagonal, computing the pseudo-inverse is quick, and the memory use and compute time for constructing \mathbf{U}^+ scales as $O(N_{\text{comp}}^2 N_{\text{obs}} L^2)$.

With this efficient representation of \mathbf{U}^+ in our toolbox, we again use the trick of [Elman \(1999\)](#) to construct a preconditioner in the form

$$\mathbf{A}^{-1} = (\mathbf{U}^T \mathbf{T} \mathbf{U})^{-1} \approx \mathbf{U}^+ \mathbf{T}^{-1} (\mathbf{U}^+)^T. \quad (14)$$

The lower right identity blocks of \mathbf{T} require no inversion. The \tilde{N}_v^{-1} -blocks in the upper left section of \mathbf{T} can be approximately inverted using the technique presented in Sect. 3.2, so that

$$(\tilde{N}_v^{-1})^{-1} \approx \alpha_v^2 Y_v^T W_v N_v W_v Y_v.$$

Let \mathbf{T}^+ denote \mathbf{T}^{-1} approximated in this manner, and the resulting preconditioner is

$$\mathbf{M}_{\text{PI}} = \mathbf{U}^+ \mathbf{T}^+ (\mathbf{U}^+)^T. \quad (15)$$

Computationally speaking, this has the optimal operational form: the pseudo-inverse \mathbf{U}^+ and its transpose can be applied simply by a series of small matrix-vector products in parallel using the pre-computed blocks, while application of \mathbf{T}^+ as an operator requires $2N_{\text{obs}}$ SHTs.

So far we have only considered the pseudo-inverse as an algebraic trick, but some analytic insights are also available. First we look at two special cases. The preconditioned system can be written

$$\mathbf{M}_{\text{PI}} \mathbf{A} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{T}^+ \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{T} \mathbf{U}.$$

First, we note that if the real model indeed has a flat inverse-noise variance map, then $\mathbf{T}^+ = \mathbf{T} = \mathbb{I}$ and the preconditioner is perfect, $\mathbf{M}_{\text{PI}} \mathbf{A} = \mathbf{A}$. Second, we consider the case in which the inverse-noise variance maps are not flat, but that rather (1) $N_{\text{comp}} = N_{\text{obs}}$; (2) there are no priors; and (3) the band-limit of each sky observation matches that of its dominating component. In this case, \mathbf{U} is square and invertible, and $(\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T = \mathbf{U}^{-1}$, and thus $\mathbf{M}_{\text{PI}} \mathbf{A} = \mathbf{U}^{-1} \mathbf{T}^+ \mathbf{T} \mathbf{U} \approx \mathbf{A}$, using the very good approximation developed in Sect. 3.2.

In the generic case, $\mathbf{T} \neq \mathbb{I}$, and \mathbf{U} has more rows than columns. First we note that for an other equivalent model with constant inverse-noise variance maps, the corresponding system matrix is $\mathbf{A}_{\text{F}} \equiv \mathbf{U}^T \mathbf{U}$. Further, we can define a system that acts just like \mathbf{A} in terms of effects that are spatially invariant, but which has the inverse effect when it comes to the scale-free spatial variations: $\mathbf{A}_{\text{SPI}} \equiv \mathbf{U}^T \mathbf{T}^+ \mathbf{U}$. With these definitions we have

$$\mathbf{M}_{\text{PI}} \mathbf{A} = \mathbf{A}_{\text{F}}^{-1} \mathbf{A}_{\text{SPI}} \mathbf{A}_{\text{F}}^{-1} \mathbf{A}.$$

The operator above applies the spatially variant effects once forward and once inversely, and the spatially invariant effects twice forward and twice inversely. Everything that is done is also undone, and in this sense $\mathbf{M}_{\text{PI}} \mathbf{A}$ can be said to approximate \mathbf{A} .

To see where the approximation breaks down, one must consider what \mathbf{A}_{SPI} actually represents, which is a linear combination of the spatial inverses of the inverse-noise maps and the priors. The \mathbf{U} matrix gives more weight to a band with more information for each component. For instance, in a *Planck*-type setup that includes a thermal dust component, the spatial inverse of the 857 GHz band will be given strong weight, while the spatial inverse of 30 GHz will be entirely neglected, as desired. Likewise, the prior terms will be given little weight in the data-dominated regime at low multipoles, and then gradually be introduced as the data becomes noise dominated.

However, the weights between the spatial inverses ignore spatial variations, and only account for the average within each multipole ℓ . Thus the preconditioner only works well when faced

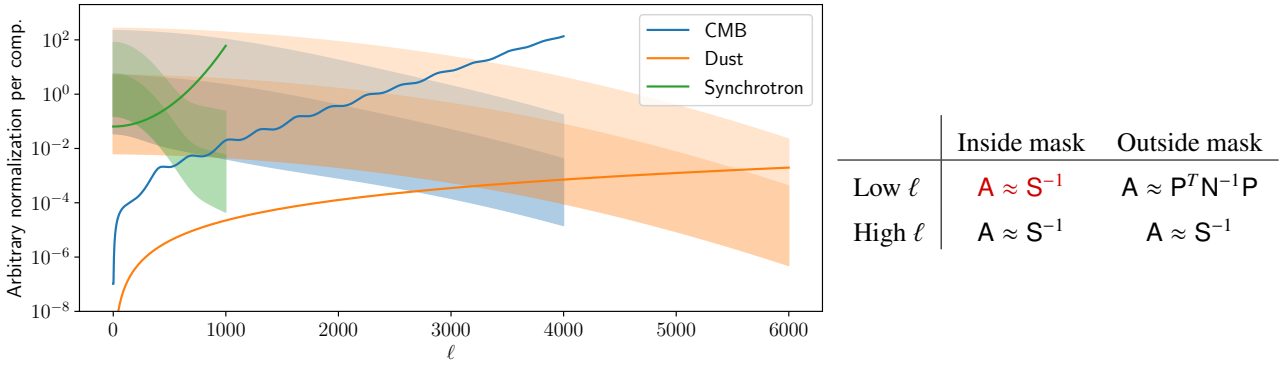


Fig. 5. *Left panel:* relative amplitudes of the prior term (solid lines; S_k^{-1}) and inverse-noise term (diffuse bands; $\sum_v P_{v,k}^T N_v^{-1} P_{v,k}$) on the diagonal of the coefficient matrix A . Each full inverse-noise band has been evaluated by summing over the nine *Planck* frequency bands. The darker part of each band displays the extents of the inverse-noise terms if one reduces contrast by adding synthetic regularization noise to the 1% least noisy pixels. This in turn reduces the condition number of the full matrix. Where the solid lines lie above the diffuse bands, the full matrix is prior dominated; where they lie below, the full matrix is data dominated. *Right panel:* schematic setup of which term dominates A in different regimes. The pseudo-inverse preconditioner of Sect. 3 automatically resolves the low- ℓ versus high- ℓ split, taking into account cross-component couplings. To handle the crucial low- ℓ regime inside the mask (red), the extension described in Sect. 4 is required.

with modest spatial variations in the inverse-noise maps. This is illustrated in Fig. 5. For the spatially invariant part of the preconditioner, the crossover between the inverse-noise dominated regime and the prior-dominated regime must happen at a single point in ℓ -space, whereas in reality this point varies based on spatial position for ℓ in the mid-range where the prior lines are crossing the inverse-noise bands.

Figure 6 summarizes the performance of the above preconditioner in a realistic full sky component separation setup in terms of iteration count. The analysis set-up corresponds to a standard nine-band *Planck* data set in terms of instrumental noise levels and beam characteristics, as most recently reported by [Planck Collaboration IV \(2019\)](#).

1. Synchrotron with a band-limit of $L_{\text{synch}} = 1000$, a Gaussian prior $C_{\text{synch},\ell}$ with FWHM of 30 arc-minutes, and with signal-to-noise ratio of unity at $\ell = 350$.
2. CMB with a band-limit of $L_{\text{cmb}} = 4000$, a Λ CDM power spectrum prior, and with signal-to-noise ratio of unity around $\ell = 1600$.
3. Thermal dust with a band-limit $L_{\text{dust}} = 6000$ and a signal-to-noise ratio of unity around $\ell = 4500$.

All mixing matrices are adopted from the *Commander* temperature analysis described by [Planck Collaboration IV \(2019\)](#), which accounts for full spatial variations in both the synchrotron² and thermal dust components.

In the top panel we show a model for which the full-resolution high-contrast mixing maps are retained for both the dust and synchrotron components. The spatial variations in the mixing maps $q_{v,k}(\hat{n})$ are not modelled in the preconditioner, and so both preconditioners struggle in this case, but the pseudo-inverse preconditioner is still twice as fast as the block-diagonal preconditioner. In the middle panel, we apply a Gaussian low-pass filter to the mixing maps, reducing both local spatial variation and contrast. We applied Gaussian filters with a FWHM of 20 arc-minutes for the dust component and 1° for the synchrotron component. This improves the convergence rate by

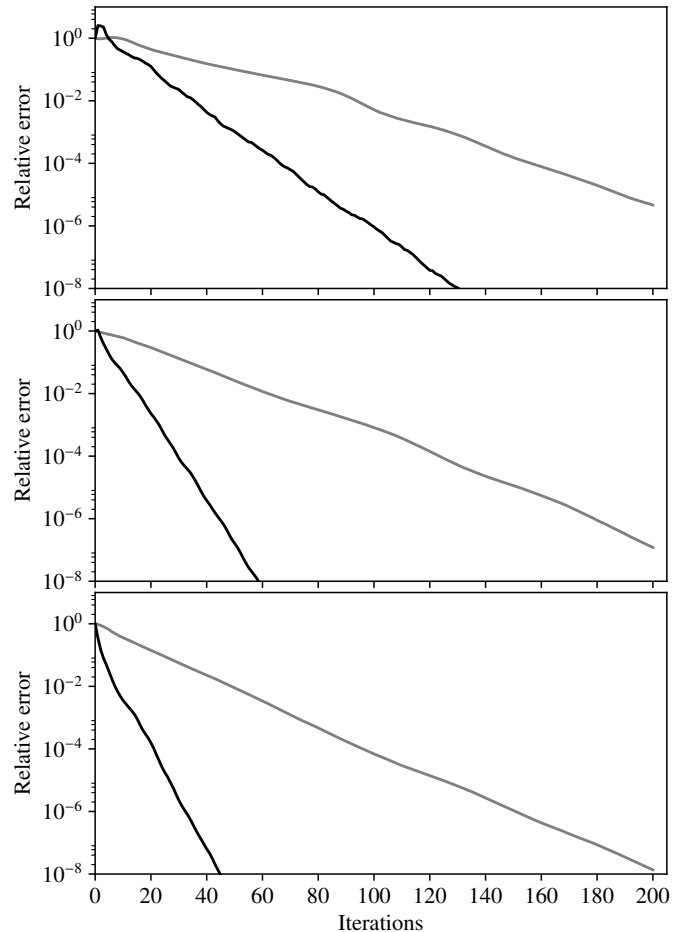


Fig. 6. Convergence of the pseudo-inverse preconditioner (shown in black lines) for a full-sky component separation model for three different sets of mixing matrices with varying spatial structure. For comparison, grey lines show the convergence of a block diagonal preconditioner. The top panel shows a case with full-resolution mixing matrices; the middle panel shows a case with mixing matrices smoothed to 20' and 1° FWHM for thermal dust and synchrotron emission, respectively; and the bottom panel shows a case with spatially uniform mixing matrices. Further details are given in Sect. 3.3.

² We note that although we refer to this component as “synchrotron”, it does in fact include contributions from both free-free and spinning dust emission. However, for the purposes of the present paper, the specific astrophysics is irrelevant, and we adopt for convenience the simpler naming convention.

about a factor of two. Finally, in the bottom panel the mixing maps have no spatial variation at all, and the entire foreground model is described by the preconditioner. This represents the limiting optimal case for this particular setup.

The pseudo-inverse preconditioner is at a greater advantage in situations with moderate structure and contrasts in the mixing maps. Unlike the diagonal preconditioner, the pseudo-inverse preconditioner solves for the structure in the inverse-noise matrix N^{-1} , but neither solves for the spatial structure in mixing maps. On a harder problem in general, it seems that diagonal preconditioner is put at a (relative) advantage by having more iterations at which to also work on the inverse-noise term.

It should be noted that the pseudo-inverse preconditioner requires extra SHTs compared to a standard diagonal preconditioner, and so requires more computing time per iteration. This penalty is highly model specific as it depends on N_{comp} and N_{obs} . In general, each multiplication with A requires $(2N_{\text{comp}} + 4N_{\text{obs}})$, which translates into 42 SHTs for our case, whereas application of the pseudo-inverse preconditioner requires $2N_{\text{obs}} = 18$ SHTs. Additional optimizations can be made if one or more mixing matrices are spatially uniform, as for instance is the case for the CMB component. If all components have the same resolution, this translates to each iteration of the pseudo-inverse solver taking about 33% longer than the block-diagonal preconditioner, and with a third of the iterations needed, we end up with a total run-time reduction of 60%. For this model, the synchrotron with a flat mixing map has much lower angular resolution than the dust component with a variable mixing map, and so the speed-up is somewhat larger. In the future it could be possible to use approximate SHTs to significantly reduce this overhead (see Sect. 6).

4. Constrained realization under a mask

So far we have only considered the full sky case. In many practical applications, we additionally want to mask out parts of the sky, either because of missing data, or because we do not trust our model in a given region of the sky. Within such scenarios, it is useful to distinguish between two very different cases:

- (i) Partial sky coverage, where only a small patch of the sky has been observed, and we wish to perform component separation only within this patch. The typical use case for this setup is ground-based or sub-orbital CMB experiments.
- (ii) Natively full sky coverage, but too high foregrounds in a given part of the sky to trust our model. In this case, one often masks out part of the sky, but still seeks a solution to the system under the mask, constrained by the observed sky at the edges of the mask and determined by the prior inside the mask. By ignoring data from this region we at least avoid the CMB component being contaminated by foreground emission. Of course, the solution will not be the true CMB sky either, but it will have statistically correct properties for use inside of a Gibbs sampler (Jewell et al. 2004; Wandelt et al. 2004; Eriksen et al. 2004b, 2008).

We expect the solver developed in the previous section to work well in case (i) given appropriate modifications, but leave such modifications for future work, and focus solely on case (ii) in what follows.

4.1. Incorporating a mask in the model while avoiding ringing effects

We recall from Sect. 2.2 that our data model reads

$$\mathbf{d} = \mathbf{Y}_v \mathbf{B}_v \widetilde{\mathbf{Q}}_{v,k} \mathbf{s} + \mathbf{n},$$

where \mathbf{d} are the pixels of the observed sky maps. We follow Eriksen et al. (2004b) and Seljebotn et al. (2014) and introduce a mask in the model by declaring that the masked-out pixels are missing from \mathbf{d} , or, equivalently, that N^{-1} is zero in these pixels. This is straightforward from a modelling perspective, but has an inconvenient numerical problem: ideally, we want to specify the beam operator \mathbf{B} using a spherical harmonic transfer function b_ℓ , but, because b_ℓ must necessarily be band-limited, \mathbf{B} exhibits ringing in its tails in pixel basis. Specifically, in pixel space the beam operator first exhibits an exponential decay, as desired, but then suddenly stops decaying before it hits zero. At this point, it starts to observe the entire sky through the ringing “floor” (see figures in Seljebotn et al. 2014), and it becomes non-local. When the signal-to-noise ratio of the data in question is high enough compared to such numerical effects, the model will try to predict the signal component s within the mask through deconvolution of the pixels at the edge of the mask, regardless of their distance. This causes a major complication for all solvers of this type, and in Seljebotn et al. (2014) we had to carefully tune the solver to avoid this ringing effect.

In the present solver we introduce the mask in the mixing maps $\mathbf{Q}_{v,k}$, rather than in the noise model, but only in the preconditioner. This corresponds to making the approximation that beam convolution and masking commute, even though these operations do in fact not commute. The sky is then split cleanly into one set of pixels outside the mask that hits the full matrix $\mathbf{S}^{-1} + \mathbf{P}^T \mathbf{N}^{-1} \mathbf{P}$, and another set of pixels (those under the mask) that only hits the prior term \mathbf{S}^{-1} . We then proceed to apply separate preconditioners in the two regions. This simple approximation, although far from exact, is good enough to support good convergence for the final solver.

4.2. Independently solving for signals under a mask

Let us consider the schematic description of each regime in the right panel of Fig. 5. The pseudo-inverse preconditioner, which we will denote \mathbf{M}_{PI} , automatically finds a good split per component for the low- and high- ℓ regimes, respectively, but is, in the same way as a block-diagonal preconditioner, blind to the different regimes inside and outside of the mask. As a result, it performs poorly for large scales inside the mask, that is, for multipoles lower than the point of unity signal-to-noise ratio shown in Fig. 7.

To solve this, we supplement the pseudo-inverse preconditioner with a second preconditioner that is designed to work well only inside the mask, where it is possible to simplify the system. Let \mathbf{Z} denote spherical harmonic synthesis to the pixels within the mask only; that is, we first apply \mathbf{Y} , and then select only the masked pixels. Then, building on standard domain decomposition techniques, a preconditioner that provides a solution only within the masked region is given by

$$\mathbf{M}_{\text{mask}} = \mathbf{Z}^T (\mathbf{Z} \mathbf{A} \mathbf{Z}^T)^{-1} \mathbf{Z} = \mathbf{Z}^T (\mathbf{Z} \mathbf{S}^{-1} \mathbf{Z}^T)^{-1} \mathbf{Z}. \quad (16)$$

We will develop a solver for the inner mask-restricted system in Sect. 4.4. The idea is then to use both the full-sky solver developed in Sect. 3.2 \mathbf{M}_{PI} , and the mask-restricted solver \mathbf{M}_{mask} . The simplest possible way of doing this is simply to add them together, $\mathbf{M}_{\text{add}} \equiv \mathbf{M}_{\text{PI}} + \mathbf{M}_{\text{mask}}$, and we find experimentally that this simple combination performs well for our purposes. In our experiments we have chosen \mathbf{M}_{PI} to simply be the unaltered full sky pseudo-inverse. As the full system matrix \mathbf{A} does not in fact include values from the RMS map inside the mask, there is some freedom here to choose other values, although we found other choices to have little effect on overall convergence.

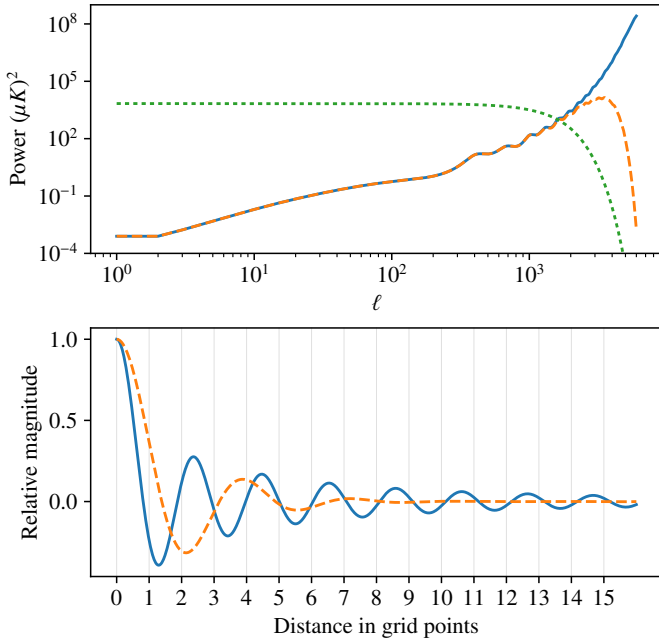


Fig. 7. Harmonic filtering of the mask-restricted system $\mathbf{ZS}^{-1}\mathbf{Z}^T$. *Top panel:* inverse CMB power spectrum $1/C_\ell$ (solid blue), on the diagonal of the \mathbf{S}^{-1} matrix behaves as ℓ^2 near the beginning, increasing in steepness to ℓ^8 at $\ell = 6000$. It crosses the diagonal of the inverse-noise term \mathbf{N}^{-1} (dotted green) at around $\ell = 1600$. Above this point the system becomes prior dominated and the pseudo-inverse preconditioner works well both inside and outside the mask. Since we do not need the dedicated mask solver to solve for high ℓ , we apply a filter r_ℓ as described in Sect. 4.4, resulting in a filtered prior r_ℓ^2/C_ℓ (dashed orange). *Bottom panel:* \mathbf{S}^{-1} operator in pixel domain. The plot displays $(\mathbf{Y}\mathbf{S}^{-1}\mathbf{Y}^T)_{ij}$ as a function of the distance between grid points i and j . Both the unfiltered operator $1/C_\ell$ (solid blue) and the low-pass filtered r_ℓ^2/C_ℓ (dashed orange) have a “Mexican hat” or Wavelet-like azimuthally symmetric shape. The low-pass filter ensures that the oscillations decay quickly, making the system easier to solve in pixel domain than the unfiltered version.

The preconditioner described above, however, may break down for more demanding models. We found that the most important feature in how well \mathbf{M}_{add} works is how far the inverse-noise term decays before it is overtaken by the prior term (see Fig. 7). In the case of analysing *Planck* data, we find that the inverse-noise term decays by a factor of roughly $\lambda = 0.15$ at this point, which is unproblematic. In simulations with lower resolution or higher noise, such that $\lambda = 0.01$, convergence is hurt substantially, and at $\lambda = 0.001$ the preconditioner breaks down entirely.

Since our own use cases (which are targeted towards *Planck*) are not affected by this restriction, we have not investigated this issue very closely. We have, however, diagnosed the effect at low resolution with a dense system solver for $(\mathbf{ZS}^{-1}\mathbf{Z})^{-1}$. In these studies, we find that the problem is intimately connected with how the two preconditioners are combined, and it may well be that more sophisticated methods for combining preconditioners will do a better job. For the interested reader, we recommend [Tang et al. \(2009\)](#) for an introduction to the problem, as they cover many related methods arising from different fields using a common terminology and notation. In particular it would be interesting to use deflation methods to “deflate” the mask subspace out of the solver. It will still be limited by the fundamental approximation in which we moved the mask from the inverse-noise term \mathbf{N}^{-1} to the mixing term \mathbf{Q} , and this sets a limit to how well the preconditioner can work along the edges of the mask.

Finally, we give a word of warning: in the full sky case, we have been able to re-scale the CR system arbitrarily without affecting the essentials of the system. For instance, [Eriksen et al. \(2008\)](#) scale \mathbf{A} with $\mathbf{S}^{1/2}$ so that the system matrix becomes $\mathbf{I} + \mathbf{S}^{1/2}\mathbf{P}^T\mathbf{N}^{-1}\mathbf{P}\mathbf{S}^{1/2}$. This has no real effect on the spherical harmonic preconditioners, but in pixel domain it changes the shape of each term. The natural unscaled form of \mathbf{A} is localized in pixel domain. The inverse-noise term essentially looks like a sum of the instrumental beams, while the prior defines smoothness couplings between a pixel and its immediate neighbourhood. However, $\mathbf{S}^{1/2}$ is a highly non-local operator, and multiplying with this factor decreases locality and causes a break-down of our method. There may of course be other filters that would increase locality in pixel domain instead of decrease it, in which case it could be beneficial to apply them.

4.3. Including a low-pass filter in the mask-restriction

The feature that most strongly defines the mask-restricted system $\mathbf{ZS}^{-1}\mathbf{Z}^T$ is the shape of the mask, and thus pixel basis is the natural domain in which to approach this system. The operator $\mathbf{ZS}^{-1}\mathbf{Z}^T$ acts as a convolution with an azimuthally symmetric kernel on the pixels within the mask. In Fig. 7 we plot a cut through this convolution kernel (blue in bottom panel). Mainly due to the sharp truncation at the band-limit L , oscillations extend far away from the centre of the convolution kernel. To make the system easier to solve, we follow [Seljebotn et al. \(2014\)](#) and insert a low-pass filter as part of the restriction operator \mathbf{Z} , so that the projection from spherical harmonics to the pixels within the mask is preceded by multiplication with the transfer function

$$r_\ell = \exp(-\beta\ell^2(\ell + 1)^2), \quad (17)$$

where we choose β so that $r_{L/2}^2 = 0.05$. The resulting system now has a transfer function of r_ℓ^2/C_ℓ , whose associated convolution kernel is much more localized (dashed orange), making it easier to develop a good solver for $\mathbf{ZS}^{-1}\mathbf{Z}^T$.

After introducing this low-pass filter we no longer have equality in Eq. (16), but only approximately that $\mathbf{ZAZ}^T \approx \mathbf{ZS}^{-1}\mathbf{Z}^T$. This appears to not hurt the overall method, as r_ℓ is rather narrow when seen as a pixel-domain convolution (unlike $\sqrt{C_\ell}$, as noted above). Also we note that we have now over-pixelized the system $\mathbf{ZS}^{-1}\mathbf{Z}$, as higher frequency information has been suppressed and the core of the convolution kernel is supported by two pixels. Our attempts at representing $\mathbf{ZS}^{-1}\mathbf{Z}$ on a coarser grid failed however, because the solution will not converge along the edge of the mask unless the grid of \mathbf{Z} exactly matches the grid of the mixing map $\mathbf{Q}_{v,k}$. While the resulting system $\mathbf{ZS}^{-1}\mathbf{Z}^T$ on the full-resolution grid is poorly conditioned for the smallest scales, this does not prevent us from applying iterative methods to solve for the larger scales.

4.4. Multi-grid solver for the mask-restricted system

Finally we turn our attention to constructing an approximate inverse for $\mathbf{ZS}^{-1}\mathbf{Z}^T$. We now write the same system matrix as

$$\mathbf{G} = \mathbf{ZS}^{-1}\mathbf{Z}^T = \mathbf{YDY}^T, \quad (18)$$

where \mathbf{D} is a diagonal matrix with $d_\ell = r_\ell^2/C_\ell$ on the diagonal, and it should be understood that the spherical harmonic synthesis \mathbf{Y} only projects to grid points within the mask.

As noted in [Seljebotn et al. \(2014\)](#), in the case where $d_\ell \propto \ell^2$ this is simply the Laplacian partial differential equation on the

sphere, and the multi-grid techniques commonly used for solving this system are also effective in our case. We will focus on the case where C_ℓ is the CMB power spectrum; in this case $1/C_\ell$ starts out proportional to ℓ^2 , increasing to ℓ^6 around $\ell \sim 1600$, eventually reaching ℓ^8 at $\ell \sim 6000$. In theory this should make the system harder to solve than the Laplacian, but it seems that in our solver the application of the low-pass filter described in the previous section is able to work around this problem.

To solve the system $\mathbf{G}\mathbf{x} = \mathbf{b}$ using iterative methods, we might start with a simple diagonal approximate inverse,

$$\mathbf{M} \equiv \text{diag}(\mathbf{G})^{-1},$$

which is in fact a constant scaling since \mathbf{G} is spatially invariant. This turns out to work well as a preconditioner for the intermediate scales of the solution. For smaller scales (higher ℓ) the quickly decaying restriction r_ℓ starts to dominate over $1/C_\ell$ such that the combined effect is that of a low-pass filter; such filters cannot to our knowledge be efficiently deconvolved in pixel domain and an harmonic-domain preconditioner would be required. Luckily, we do not need to solve for these smaller scales, as the pseudo-inverse preconditioner will find the correct solution in this regime, and the restriction operator \mathbf{Z} will at any rate filter out whatever contribution comes from the solution of \mathbf{G} .

The problem at larger scales (lower ℓ) is that the approximate inverse would have to embed inversion of the coupling of two distant pixels through a series of intermediate pixels in-between; this is beyond the reach of our simple diagonal preconditioner. For this reason we introduce a multi-grid (MG) V-cycle, where we recursively solve the system on coarser resolutions. For each coarsening, the preconditioner is able to see further on the sphere, as indirect couplings in the full-resolution system are turned into direct couplings in the coarser versions of the system. A basic introduction to multi-grid methods can be found for example in Hackbush (1985) or the overview given in Seljebotn et al. (2014).

The first ingredient in MG is an hierarchy of grids, which are denoted relatively, with h denoting an arbitrary level and H denoting the grid on the next coarser level. We have opted for a HEALPix grid for $\mathbf{Q}_{v,k}$ and \mathbf{G} , and use its hierarchical structure to define the coarser grid, simply letting $N_{\text{side},H} = N_{\text{side},h}/2$. We also need to consider which subset of grid points to include to represent the region within the mask. We got the best results by only including those pixels of H that are covered 100% by the mask in the fine grid h , so that no pixel on any level ever represents a region outside the full-resolution mask.

The second ingredient in MG is the restriction operator l_h^H , which transfers a vector from grid h to grid H . We tried restriction operators both in pixel domain and spherical harmonic domain, and spherical harmonic restriction performed better by far. Thus we define

$$l_h^H = \mathbf{Y}_H \mathbf{R}_H \mathbf{Y}_h^T \mathbf{W}_h, \quad (19)$$

where we use subscripts to indicate the grid of each operator, and where \mathbf{R}_H has some harmonic low-pass filter $r_{H,\ell}$ on its diagonal. In Seljebotn et al. (2014) the corresponding filter had to be carefully tuned to avoid problems with ringing, because the N^{-1} -term created high contrasts in the system matrix. In the present method we no longer have to deal with the N^{-1} term, and the requirements on the low-pass filter are much less severe, as long as they correspond to a convolution kernel with a FWHM of roughly one pixel on the coarse grid. A Gaussian band-limited at $L_H = 3N_{\text{side},H} = L_h/2$ performed slightly better than the filter of Eq. (17) in our tests, even if it has somewhat more ringing at this band-limit.

The third ingredient in MG is the coarsened linear system

$$\mathbf{G}_H = l_h^H \mathbf{G}_h l_h^H = \mathbf{Y}_H \mathbf{D}_H \mathbf{Y}_H^T, \quad (20)$$

where $d_{H,\ell} = r_{H,\ell}^2 d_{h,\ell}$ is band-limited at $L_H = L_h/2^3$. We stress again that the grid H embeds the structure of the mask, so that \mathbf{Y}_H in this context denotes spherical harmonic synthesis only to grid points within the mask. In computer code, zero padding is used outside of the mask before applying \mathbf{Y}_H^T , and entries outside the mask are discarded after applying \mathbf{Y}_H .

The fourth ingredient in MG is an approximate inverse, in this context named the smoother. The name refers to removing small scales in the error $\mathbf{x}_{\text{true}} - \mathbf{x}$. Removing these scales happens through approximately solving the system, and should not be confused with applying a low-pass filter. In our case we will use the simple constant smoother \mathbf{M} discussed above, although in combination with a damping factor $\omega = 0.2$, so that the eigenvalues of $\omega \mathbf{M} \mathbf{G}$ are bounded above by 2 as required by the MG method. We write $\mathbf{M}_h = \omega \text{diag}(\mathbf{G}_h)^{-1}$ for the smoother on level h .

Finally, the ingredients are combined in the simplest possible MG V-cycle algorithm (see Fig. 8). It turns out that the restriction and interpolation operations can share one SHT each with the associated system matrix multiplication, so that six SHTs are required per level. Furthermore, the SHTs can be performed at different resolutions for additional savings. Like the pseudo-inverse preconditioner, the mask-restricted solver appears to work well with approximate SHTs, enabling further savings in the future (see Sect. 6).

Figure 9 shows the results of solving the full system when inserting this algorithm as an approximation of $(\mathbf{Z}\mathbf{S}^{-1}\mathbf{Z}^T)^{-1}$ in Eq. (16). While the diagonal preconditioner degrades, \mathbf{M}_{add} converges very quickly. The pseudo-inverse preconditioner \mathbf{M}_{PI} by itself shows much the same behaviour as the diagonal preconditioner in this situation (not plotted).

5. Benchmark notes

The implementation used for the convergence plots in this paper are produced using a prototype implementation written in a mixture of Fortran, Cython, and Python, and is available under an open source license⁴. As a prototype, it does not support polarization or distributed computing with the Message Passing Interface (MPI). A full implementation in the production quality Commander code is in progress.

Simulations are performed with a known \mathbf{x}_{true} drawn randomly from a Gaussian distribution, and a right-hand side given by $\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}}$. Then the convergence statistic denoted “relative error” in these figures simply reads

$$e_i \equiv \|\mathbf{x}_i - \mathbf{x}_{\text{true}}\|.$$

Finally, unless otherwise noted, we add regularization noise to the 1% of the highest signal-to-noise pixels in the RMS maps. As noted in Fig. 3, this is more of an advantage for the diagonal preconditioner than the pseudo-inverse preconditioner, but this typically mimics what one would do in real analysis cases.

In the present paper we have focused strictly on algorithm development, and as such the prototype code is not optimized; we have not invested the effort to benchmark the preconditioners

³ The matrix coarsening must be done in another way if using a pixel-domain restriction operator. In that case \mathbf{D}_H is in principle a dense matrix due to pixelization irregularities, but can still be very well approximated by a diagonal matrix. Details are given in Appendix B.

⁴ <https://github.com/dagss/cmbcr>

Basic-V-cycle(h, \mathbf{b}):

Inputs:

h – The current level
 \mathbf{b} – Right-hand side

H denotes the coarser level relative to h .

Output:

Approximation of $\mathbf{G}_h^{-1}\mathbf{b}$

if h is bottom level:

$\mathbf{x} \leftarrow \mathbf{G}_h^+\mathbf{b}$

By SVD pseudo-inverse

else:

$\mathbf{x} \leftarrow \mathbf{M}_h\mathbf{b}$

Pre-smoothing

$\mathbf{r}_h \leftarrow \mathbf{b} - \mathbf{G}_h\mathbf{x}$

Compute residual

$\mathbf{r}_H \leftarrow \mathbf{I}_h^H\mathbf{r}_h$

Restrict residual

$\mathbf{c}_H \leftarrow \text{Basic-V-Cycle}(H, \mathbf{r}_H)$

Recurse for coarse correction

$\mathbf{c}_h \leftarrow (\mathbf{I}_h^H)^T\mathbf{c}_H$

Interpolate correction

$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{c}_h$

Add correction

$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{M}_h(\mathbf{b} - \mathbf{G}_h\mathbf{x})$

Post-smoothing

return \mathbf{x}

Optimized-V-cycle(h, \mathbf{b}):

Inputs:

h – The current level
 \mathbf{b} – Right-hand side

H denotes the coarser level relative to h .

Output:

Approximation of $\mathbf{G}_h^{-1}\mathbf{b}$

if h is bottom level:

$\mathbf{x} \leftarrow \mathbf{G}_h^+\mathbf{b}$

else:

$\mathbf{x} \leftarrow \mathbf{M}_h\mathbf{b}$

$\mathbf{u} \leftarrow \mathbf{Y}_h^T\mathbf{x}$

SHT at L_h, N_h

$\tilde{\mathbf{r}} \leftarrow \mathbf{Y}_h^T\mathbf{W}_h\mathbf{b} - \mathbf{J}\mathbf{D}_h\mathbf{u}$

SHT at L_H, N_h

$\mathbf{r}_H \leftarrow \mathbf{Y}_H\mathbf{R}_H\tilde{\mathbf{r}}$

SHT at L_H, N_H

$\mathbf{c}_H \leftarrow \text{Optimized-V-Cycle}(H, \mathbf{r}_H)$

$\tilde{\mathbf{c}} \leftarrow \mathbf{R}_H\mathbf{Y}_H^T\mathbf{c}_H$

SHT at L_H, N_H

$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{W}_h\mathbf{Y}_h\tilde{\mathbf{c}}$

SHT at L_H, N_h

$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{M}_h(\mathbf{b} - \mathbf{Y}_h\mathbf{D}(\mathbf{u} + \mathbf{J}\tilde{\mathbf{c}}))$

SHT at L_h, N_h

return \mathbf{x}

Fig. 8. Pseudo-code for the MG V-cycle. The matrices involved are defined in the main text. *Basic-V-Cycle*: the clean textbook version, exposing the basic structure of the algorithm. The important feature of the algorithm is that the solution vector \mathbf{x} is never transferred directly between levels. Instead, a residual \mathbf{r}_H is computed, which takes the role as the right-hand side \mathbf{b} on the coarser level. The coarse solution is a correction \mathbf{c}_H that is then added to the solution vector \mathbf{x} . The resulting full V-cycle is a symmetric linear operator that can be used as a preconditioner for CG. We keep recursing until there are less than 1000 coefficients left, and then solve using a pseudo-inverse based on the SVD, $\mathbf{G}_h^+ \approx \mathbf{G}_h^{-1}$, as \mathbf{G}_h may, depending on the size of the mask, be singular due to \mathbf{D}_h being truncated at L_h . *Optimized-V-Cycle*: in this code we have inserted $\mathbf{I}_h^H = \mathbf{Y}_H\mathbf{R}_H\mathbf{Y}_h^T\mathbf{W}_h$ and $\mathbf{G}_h = \mathbf{Y}_h\mathbf{D}_h\mathbf{Y}_h^T$, and then reorganized the expressions so that the restriction and interpolation steps each share one SHT with the corresponding application of \mathbf{G}_h . The \mathbf{J} operator denotes $\mathbf{Y}_h^T\mathbf{W}_h\mathbf{Y}_h$. The effect of this operator is to zero out any contribution that falls outside of the mask in the (full sky) spherical harmonic vectors; but numerical experiments indicate that the term can in practice be neglected also when using a small mask. In our numerical experiments we approximate $\mathbf{J} \approx \mathbf{I}$, reducing the total number of SHTs to six per level. The comments indicate the required resolution for each SHT, with L_h (L_H) referring to fine (coarse) harmonic band-limit, and N_h (N_H) referring to fine (coarse) grid.

in terms of CPU time spent. As detailed in Sects. 3.3 and 4.4, the additional cost for a particular use case can be calculated from the number of extra SHTs.

The block-diagonal preconditioner we use as a comparison point is described in further detail by Eriksen et al. (2004b). In the notation of this paper, it can be written

$$\mathbf{M}_{\text{diag}} \equiv (\mathbf{U}^T \text{diag}(\mathbf{T})\mathbf{U})^{-1},$$

where each element of $\text{diag}(\mathbf{T})$ can be computed in $O(L)$ time by a combination of Fourier transforms and computing the associated Legendre polynomials, which is available for example in the latest version of Libsharp⁵. Since the matrix $\mathbf{U}^T \text{diag}(\mathbf{T})\mathbf{U}$ consists of $(\ell_{\text{max}} + 1)^2$ blocks of size $N_{\text{comp}} \times N_{\text{comp}}$, the inversion is cheap.

For the component separation case we provide three benchmarks in Fig. 7, with varying degrees of difficulty introduced through the mixing maps. For the benchmark of Fig. 9, we only included the CMB component, and so only a flat mixing map was in use.

6. Discussion and outlook

In this paper we have presented a versatile Bayesian model for the multi-resolution CMB component separation and constrained realization problem, as well as an efficient solver for the associated linear system. This model is currently in active use for component separation for the *Planck* 2017 collaboration. The final result is the ability to perform exact, full-resolution, multi-resolution component separation of full-sky microwave data within a reasonable number of conjugate gradient iterations.

⁵ <https://github.com/dagss/libsharp>.

To achieve such good convergence several novel techniques were employed. First, we developed a novel pseudo-inverse based preconditioner. For the full sky case this provides a speed-up of parameters. Second, we extended the model with a mask through the mixing maps, rather than through the noise covariance matrix, to avoid ringing problems associated with going between spherical harmonic domain and pixel domain. Third, we solved for the solution under the mask using a dedicated multi-grid solver in pixel domain restricted to the area under mask, where the linear system can be simplified.

We note that the pseudo-inverse preconditioner not only performs very well for the full sky case with reasonably uniform mixing maps, but it is also very simple to implement, significantly simpler than the previously standard diagonal preconditioner. As mentioned earlier, this technique is of course not novel for or restricted to CMB component separation; it has been in use for solving Navier-Stokes equations for some time. The fundamental idea is to approximate the total inverse of a sum with the best linear combination of inverses of each term. A problem related to this paper, which in particular fits this description, is the basic CMB map-making equation (e.g., Tegmark 1997). This equation is a sum over individual time segments of observations, which can in isolation be inverted in Fourier domain. If the pseudo-inverse preconditioner works well in this case, as we believe it will, it may speed up exact maximum likelihood map makers substantially.

Regarding more direct extensions of the work in the present paper, a natural next step is the use of a pixel domain basis instead of spherical harmonic coefficients to represent the microwave components. We have so far assumed an isotropic prior for all components that can be specified in the form of a power spectrum C_ℓ , with a sharp band-limit L . This model

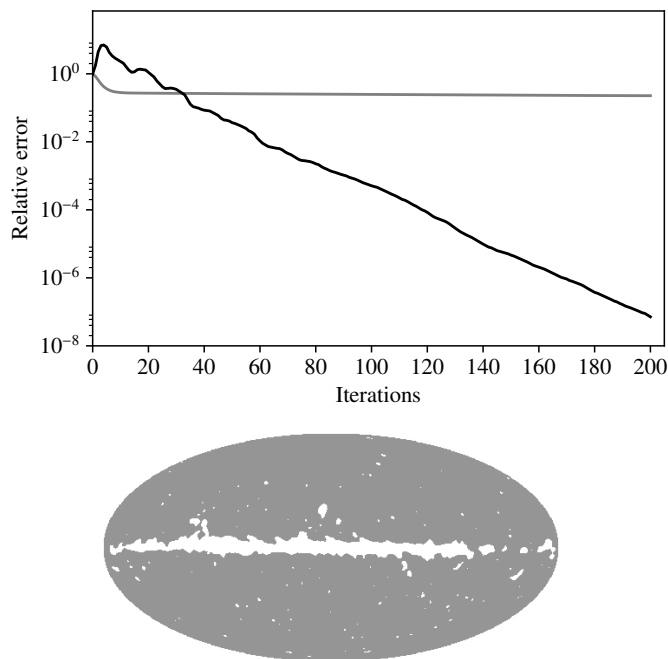


Fig. 9. Convergence of the M_{add} preconditioner (*top panel*) when including sky mask (Planck Collaboration XI 2016) in the model (*bottom panel*). We fit a single CMB component to hypothetical foreground-cleaned maps on all nine *Planck* bands, specifying a fiducial C_ℓ prior for the CMB power spectrum. We plot the convergence when using a diagonal preconditioner (grey) and the M_{add} preconditioner developed in Sect. 4 (black). Section 5 gives details on the benchmark setup.

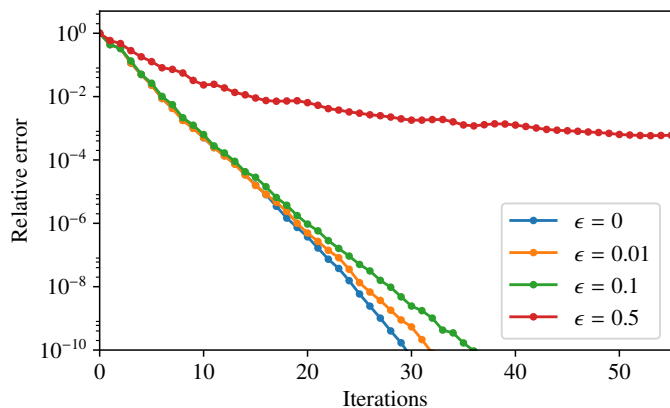


Fig. 10. Effect of employing approximate SHTs in the preconditioner. To simulate approximate SHTs, for every SHT computer, we have multiplied each coefficient of the result with a random variate drawn uniformly from the interval $[1 - \epsilon, 1 + \epsilon]$, simulating numerical noise at the ϵ -level. Note that we carried out these simulations on a resolution of $N_{\text{side}} = 128$, using the downgrade procedure outlined in Appendix C. The experiment is otherwise identical to the one in Fig. 9. The approximate SHTs are employed both in the pseudo-inverse preconditioner and the mask-restricted multi-grid solver, while the (non-preconditioner) matrix-vector multiplication uses the full accuracy provided by Libsharp.

has a tendency to excite ringing in the resulting maps around sharp objects, unless much time is spent tuning the priors, or one adopts a very high band-limit L for all components. Working with pixel-domain vectors and, with the exception of the CMB, pixel-domain prior specifications, one could easily define the models that are more robust against this problem. Also, we know

that the diffuse foregrounds have much variation where their signal is strong, but should be more heavily stabilized where their signal is weak. Such a non-isotropic prior is easier to model using a sparse matrix in pixel domain. Of particular interest are the so-called conditional auto-regressive (CAR) models, which have a natural interpretation and which directly produce sparse inverses S^{-1} .

Several SHT algorithms have been developed with a computational scaling of $O(L^2 \log L)$ or $O(L^2 (\log L)^2)$. These are generally not used within the astrophysical community because of the large pre-factors involved and lack of optimized implementations. Some of these algorithms (e.g., Tygert 2008, 2010; Seljebotn 2012) can produce low-accuracy results at a much lower cost, since their pre-factor includes a term $O(d^2)$ with d representing the number of digits in the computation. As shown in Fig. 10, the preconditioner developed in this paper works well with approximate SHTs, and optimized implementations of these algorithms may speed up the preconditioner to the point where its computational cost is negligible compared to the matrix-vector multiplication.

Acknowledgements. We thank Jeff Jewell, Sigurd Næss, Martin Reinecke, and Mikolaj Szydlarski for useful input. This work was supported by European Research Council grants StG2010-257080 and CoG2017-772253, and the H2020 COMPET-4 grant 776282. Some of the results in this paper have been derived using the HEALPix (Górski et al. 2005), Libsharp (Reinecke & Seljebotn 2013), Healpy, NumPy, and SciPy software packages.

References

- Basak, S., & Delabrouille, J. 2012, *MNRAS*, 419, 1163
 Bennett, C. L., Hill, R. S., Hinshaw, G., et al. 2003, *ApJS*, 148, 97
 Bennett, C. L., Larson, D., Weiland, J. L., et al. 2013, *ApJS*, 208, 20
 Benzi, M., Golub, G. H., & Liesen, J. 2005, *Acta Numer.*, 14, 1
 BICEP2 Collaboration 2014, *Phys. Rev. Lett.*, 112, 241101
 Cardoso, J., Martin, M., Delabrouille, J., Betoule, M., & Patanchon, G. 2008, *IEEE J. Sel. Top. Sig. Process.*, 2, 735
 Elman, H. C. 1999, *SIAM J. Sci. Comput.*, 20, 1299
 Elman, H. C., Howle, V. E., Shadid, J., Shuttleworth, R., & Tuminaro, R. 2006, *SIAM J (Comput. Sci)*, 27
 Elsner, F., & Wandelt, B. D. 2013, *A&A*, 549, A111
 Eriksen, H. K., Banday, A. J., Górski, K. M., & Lilje, P. B. 2004a, *ApJ*, 612, 633
 Eriksen, H. K., O'Dwyer, I. J., Jewell, J. B., et al. 2004b, *ApJS*, 155, 227
 Eriksen, H. K., Jewell, J. B., Dickinson, C., et al. 2008, *ApJ*, 676, 10
 Errard, J., Stivoli, F., & Stompor, R. 2011, *Phys. Rev. D*, 84, 069907
 Fernández-Cobos, R., Vielva, P., Barreiro, R. B., & Martínez-González, E. 2012, *MNRAS*, 420, 2162
 Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, *ApJ*, 622, 759
 Hackbush, W. 1985, *Multi-grid methods and applications* (Berlin: Springer Verlag)
 Jewell, J., Levin, S., & Anderson, C. H. 2004, *ApJ*, 609, 1
 Kodi Ramanah, D., Lavaux, G., & Wandelt, B. D. 2017, *MNRAS*, 468, 1782
 Planck Collaboration X. 2015, *A&A*, 594, A10
 Planck Collaboration I. 2016, *A&A*, 594, A1
 Planck Collaboration XI. 2016, *A&A*, 594, A11
 Planck Collaboration XIII. 2016, *A&A*, 594, A13
 Planck Collaboration IV. 2019, *A&A*, in press, <https://doi.org/10.1051/0004-6361/201833881>
 Reinecke, M., & Seljebotn, D. S. 2013, *A&A*, 512, A112
 Remazeilles, M., Delabrouille, J., & Cardoso, J.-F. 2011, *MNRAS*, 418, 467
 Seljebotn, D. S. 2012, *ApJS*, 199, 5
 Seljebotn, D. S., Mardal, K.-A., Jewell, J. B., Eriksen, H. K., & Bull, P. 2014, *ApJS*, 210, 24
 Shewchuk, J. R. 1994, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*
 Tang, J. M., Nabben, R., Vuik, C., & Erlangga, Y. A. 2009, *J. Sci. Comput.*, 39, 3
 Tegmark, M. 1997, *ApJ*, 480, L87
 Tygert, M. 2008, *J. Comput. Phys.*, 227, 4260
 Tygert, M. 2010, *J. Comput. Phys.*, 229, 6181
 Wandelt, B. D., Larson, D. L., & Lakshminarayanan, A. 2004, *Phys. Rev. D*, 70, 083511

Appendix A: Details of the pseudo-inverse preconditioner

A.1. Approximating the inverse-noise maps

We seek α_ν such that the distance between $\tilde{\mathbf{N}}_\nu^{-1}$ and the identity matrix is minimized:

$$\begin{aligned} \|\tilde{\mathbf{N}}_\nu^{-1}(\alpha_\nu) - \mathbb{I}\|_2 &= \|\alpha_\nu^{-2} \mathbf{Y}_\nu^T \mathbf{N}_\nu^{-1} \mathbf{Y}_\nu - \mathbf{Y}_\nu^T \mathbf{W}_\nu \mathbf{Y}_\nu\|_2 \\ &= \|\mathbf{Y}_\nu^T (\alpha_\nu^{-2} \mathbf{N}_\nu^{-1} - \mathbf{W}_\nu) \mathbf{Y}_\nu\|_2 \\ &= \|\mathbf{Y}_\nu^T \mathbf{W}^{1/2} (\alpha_\nu^{-2} \mathbf{W}^{-1} \mathbf{N}_\nu^{-1} - \mathbb{I}) \mathbf{W}^{1/2} \mathbf{Y}_\nu\|_2 \\ &= \|(\alpha_\nu^{-2} \mathbf{W}^{-1} \mathbf{N}_\nu^{-1} - \mathbb{I})\|_2. \end{aligned}$$

The last equality follows because all the singular values of $\mathbf{Y}^T \mathbf{W}^{1/2}$ are 1, at least for the Gauss-Legendre grid. For the HEALPix grid the statement is only approximate, within 10–20%, depending on resolution parameters, and this is close enough for our purposes. We conclude that the best choice is

$$\alpha_\nu = \sqrt{\frac{\sum_i (\tau_\nu(\hat{n}_i)/w_i)^2}{\sum_i \tau_\nu(\hat{n}_i)/w_i}}, \quad (\text{A.1})$$

where τ_i represents the pixels in the inverse-noise variance map on the diagonal of \mathbf{N}_ν^{-1} , and w_i are the quadrature weights of the associated grid. We have verified this expression experimentally by perturbing α_ν in either direction, and find that either choice leads to slower convergence. Ultimately, the method may even fail to converge if α_ν deviates too much from the optimal value.

In code, the easiest way to compute $\tau_\nu(\hat{n}_i)/w_i$ is by performing a pair of SHTs, $\mathbf{W}^{-1} \tau = \mathbf{Y} \mathbf{Y}^T \tau$. By replacing the usual analysis $\mathbf{Y}^T \mathbf{W}$ with adjoint synthesis \mathbf{Y}^T , we end up implicitly multiplying τ with \mathbf{W}^{-1} .

A.2. Approximating the mixing maps

Following a similar derivation to the previous section, the optimal scalar to approximate the mixing maps is given by minimizing

$$\|\tilde{\mathbf{Q}}_{\nu,k} - q_{\nu,k}\| = \|\mathbf{Y}^T \mathbf{W} \mathbf{Q}_{\nu,k} \mathbf{Y} - q_{\nu,k} \mathbf{Y}^T \mathbf{W} \mathbf{Y}\| = \|\mathbf{Q}_{\nu,k} - q_{\nu,k}\|,$$

so that the best choice is

$$q_{\nu,k} = \frac{\sum_i q_{\nu,k}(\hat{n}_i)^2}{\sum_i q_{\nu,k}(\hat{n}_i)}. \quad (\text{A.2})$$

In our cases, however, the difference between this quantity and the mean of the mixing map is negligible.

A.3. Possible future extension: Merging observations

Depending on the data model, it may be possible to reduce the number of SHTs required for each application of the pseudo-inverse preconditioner. When $N_{\text{obs}} > N_{\text{comp}}$, the system in some ways supplies redundant information. We assume that two rows in \mathbf{U} are (at least approximately) identical up to a constant scaling factor; this requires that the corresponding sky maps have the same beams, the same normalized spatial inverse-noise structure, and are located at the same frequency ν with the same SED for each component. That is, we require both $\tilde{\mathbf{N}}_1^{-1} = \tilde{\mathbf{N}}_2^{-1}$ and $\mathbf{U}_2 = \gamma \mathbf{U}_1$, where \mathbf{U}_ν indicates a row in \mathbf{U} and γ is an arbitrary scale factor. This situation is very typical for experiments with several

independent detectors within the same frequency channel, which is nearly always the case for modern experiments.

Under these assumptions, we have

$$\begin{aligned} &\begin{bmatrix} \mathbf{U}_1^T & \mathbf{U}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{N}_1^{-1} & \\ & \mathbf{N}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \\ &= \begin{bmatrix} (\gamma + 1)^{1/2} \mathbf{U}_1^T \end{bmatrix} \begin{bmatrix} \mathbf{N}_1^{-1} \end{bmatrix} \begin{bmatrix} (\gamma + 1)^{1/2} \mathbf{U}_1 \end{bmatrix}. \end{aligned}$$

Thus, we may combine the two rows without affecting the rest of the system, and thereby halve the number of SHTs required. Of course, two sky observations with such identical properties could have been co-added prior to solving the system, as is typically done when creating co-added frequency maps. In practice, however, there are typically many advantages in working with detector sub-sets, including an improved ability to isolate systematics effects (e.g., [Planck Collaboration X 2015](#)), and more easily allow for cross-correlation analysis. In addition, there may be experiments where some sky maps do not have identical properties and one does not wish to co-add, but they are similar enough that co-adding poses no problem if done in the preconditioner alone. One then needs to somehow produce compromise versions for \mathbf{B} , \mathbf{N}^{-1} and $\mathbf{M}_{\nu,k}$, replace the relevant matrices with the compromise versions, and finally use the row merge procedure described above to create a new \mathbf{U} solely for use in the preconditioner.

Appendix B: Alternative strategies for the mask-restricted solver

We spent some time exploring a pixel-domain restriction operator before turning to spherical harmonic restrictions. The simple restriction we attempted, averaging the four nested pixels using the standard HEALPix `udgrade` function, more than doubled the number of iterations required for a small mask when compared to a restriction in spherical harmonic domain, and had trouble converging at all for a large mask. The spherical harmonic restrictions are therefore well worth the extra time spent performing SHTs. Still, it is probably possible to pull out a little bit more performance by experimenting with averaging over a larger region with a choice of weights that approximates a Gaussian low-pass filter.

When using a pixel based restriction the system can no longer be coarsened simply by multiplying spherical harmonic transfer functions. However, since the operator is rotationally and spatially invariant it is simple to coarsen the system numerically. The idea is to image the operator in a single pixel, and then solve for the spherical harmonic transfer function that would produce this image. Let \mathbf{u} represent a unit vector located on equator on the coarse grid H . We then seek \mathbf{D}_H such that

$$\begin{aligned} \mathbf{Y}_H \mathbf{D}_H \mathbf{Y}_H^T \mathbf{u} &= \mathbf{l}_h^H \mathbf{G}_h \mathbf{l}_h^H \mathbf{u} = \mathbf{l}_h^H \mathbf{Y}_h \mathbf{D}_h \mathbf{Y}_h^T \mathbf{l}_h^H \mathbf{u} \\ \mathbf{D}_H \mathbf{Y}_H^T \mathbf{u} &= \mathbf{Y}_H^T \mathbf{W}_H \mathbf{l}_h^H \mathbf{Y}_h \mathbf{D}_h \mathbf{Y}_h^T \mathbf{l}_h^H \mathbf{u}. \end{aligned}$$

Now, assuming that \mathbf{D}_H is diagonal, we must have

$$d_{H,\ell,m} = \frac{(\mathbf{Y}_H^T \mathbf{W}_H \mathbf{l}_h^H \mathbf{Y}_h \mathbf{D}_h \mathbf{Y}_h^T \mathbf{l}_h^H \mathbf{u})_{\ell,m}}{(\mathbf{Y}_H^T \mathbf{u})_{\ell,m}}.$$

In practice, due to pixelization effects, \mathbf{D}_H cannot be fully diagonal and this equation cannot be satisfied for all ℓ, m . However, assuming that \mathbf{D}_h is isotropic it should be fully characterized by the modes $m = 0$, and, as \mathbf{u} was located on equator,

these produce a very good estimate. Using this coarsening procedure instead of the analytical coarsening procedure in Sect. 4.4 produces identical results when applied to the Gaussian restriction operator. With pixel-domain restriction operators, pixelization effects will hurt the approximation somewhat. We expect that the approximation will be hurt less if the averaging weights are a function of the physical distance between the grid points rather than the logical distance.

We have also experimented with using a Fourier basis to represent \mathbf{G}_h . When using a thin mask around equator, or a small point source, applying torus- or flat sky approximations, respectively, allows for using the much faster Fast Fourier Transforms (FFTs) instead of SHTs. The operator should then be transferred using the same principle as above. Let \mathbf{F} denote a discrete Fourier transform from harmonic space to real space, then, within a narrow equatorial band or a small patch, we require

$$\mathbf{F}\mathbf{D}_{\text{FFT}}\mathbf{F}^T\mathbf{u} \approx \mathbf{Y}\mathbf{D}_{\text{SHT}}\mathbf{Y}\mathbf{u}, \quad (\text{B.1})$$

and solve for

$$\mathbf{D}_{\text{FFT},k,k'} = \frac{(\mathbf{F}^{-1}\mathbf{Y}\mathbf{D}_{\text{SHT}}\mathbf{Y}^T\mathbf{u})_{k,k'}}{(\mathbf{F}^T\mathbf{u})_{k,k'}}.$$

Then coarser systems can be produced either analytically (restriction in harmonic domain) or by appropriate modifications to the technique above (restriction in pixel domain). We were able to produce a functional solver using this principle, but feel that the loss in flexibility was not worth the gain in performance compared to the solver presented in Sect. 4.4.

Appendix C: Realistically downgrading the system during development

For development of any solver, it is important to have a realistic low-resolution version of the system that can be solved in a few minutes. In our case, the most faithful low-resolution version of the system is not actually what would arise from a low-resolution CMB experiment, e.g., from the COsmic Background Explorer (COBE). This is because it is important to keep the same signal-to-noise ratio and the same overall growth rate of the power spectra in harmonic domain. If this is not carried out in the right manner, it is easy to get situations where the solver works very well for low resolution, but loses convergence entirely as resolution is scaled up.

We had good results with the following downgrade procedure:

- Downgrade the RMS maps to a lower N_{side} using HEALPix routines;
- Find the best fit Gaussian beam approximation to the instrumental beams, and make equivalent low-resolution beams based on scaling down the FWHM parameter;
- Downgrade each prior C_ℓ by sub-sampling coefficients. For instance, for a degrade from $N_{\text{side}} = 2048$ to $N_{\text{side}} = 256$, we take every eighth coefficient. Similarly, we divide each band-limit L_k by the relevant downgrade factor;
- Scale C_ℓ in such a way that the diagonal of \mathbf{S}^{-1} crosses the diagonal of $\mathbf{P}^T\mathbf{N}^{-1}\mathbf{P}$ at the same ℓ , relative to the full L_k , ensuring that the system has the same signal-to-noise properties as the full resolution system.