Inverse Problems and Imaging
Volume 9, No. 1, 2015, 27–53

doi:10.3934/ipi.2015.9.27

A SCALABLE ALGORITHM FOR MAP ESTIMATORS IN BAYESIAN INVERSE PROBLEMS WITH BESOV PRIORS

Tan Bui-Thanh

Department of Aerospace Engineering and Engineering Mechanics
Institute for Computational Engineering & Sciences
The University of Texas at Austin
Austin, TX 78712, USA

Omar Ghattas

Institute for Computational Engineering & Sciences
Jackson School of Geosciences, and Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX 78712, USA

(Communicated by Jari Kaipio)

Abstract. We present a scalable solver for approximating the *maximum a posteriori* (MAP) point of Bayesian inverse problems with Besov priors based on wavelet expansions with random coefficients. It is a subspace trust region interior reflective Newton conjugate gradient method for bound constrained optimization problems. The method combines the rapid locally-quadratic convergence rate properties of Newton's method, the effectiveness of trust region globalization for treating ill-conditioned problems, and the Eisenstat–Walker idea of preventing oversolving. We demonstrate the scalability of the proposed method on two inverse problems: a deconvolution problem and a coefficient inverse problem governed by elliptic partial differential equations. The numerical results show that the number of Newton iterations is independent of the number of wavelet coefficients n and the computation time scales linearly in n. It will be numerically shown, under our implementations, that the proposed solver is two times faster than the split Bregman approach, and it is an order of magnitude less expensive than the interior path following primal-dual method. Our results also confirm the fact that the Besov $\mathbb{B}^1_{11}$ prior is sparsity promoting, discretization-invariant, and edge-preserving for both imaging and inverse problems governed by partial differential equations.

1. Introduction.

Besov spaces have been introduced as generalizations of more common function spaces such as Lipschitz spaces, Hölder spaces, Hölder-Zygmund spaces, and Sobolev spaces, to name a few. An extensive introduction to Besov spaces and their relation to other function spaces can be found, for example, in [36]. They have been recently used as alternative regularization in deterministic

2010 *Mathematics Subject Classification.* Primary: 62G99, 49N45; Secondary: 49K20, 49K21.

Key words and phrases. Bayesian inversion, Besov space priors, MAP, wavelet, bound-constrained optimization, trust region, partial differential equations, Newton method, split Bregman method, interior point method, deconvolution, inverse problem, sparsity, discretization-invariant, edge-preserving.

This research was supported by AFOSR grant FA9550-09-1-0608; DOE grants DE-SC0002710, DE-FG02-08ER25860, DE-FC52-08NA28615, and DEFC02-06ER25782; and NSF grants CMS-1028889, OPP-0941678, DMS-0724746, and CMS-0619078. We are indebted to the anonymous referees for their critical and useful comments that improved the paper substantially.

inversions and prior in Bayesian inversions that can preserve sharp or discontinuous interfaces; see, e.g., [31, 37] for imaging, [14, 30] for deterministic inverse problems, and [27, 25, 12] for Bayesian inversions. An additional attractive feature of Besov priors is that, when combined with a wavelet-based approach for example, they promote sparsity in the *maximum a posterior* (MAP) estimates that have very small number of nonzero wavelet coefficients, since Besov norms are sparse [20, 25, 19, 14].

This paper proposes a scalable method for approximating the MAP point of Bayesian inverse problems with Besov priors, based on wavelet expansions with random coefficients, introduced in [27] for linear inverse problems, and later extended to nonlinear inversions in [12]. To begin, let us consider the following additive noise-corrupted pointwise observation model

$$\hat{y}_j := u\left(x_j, m\right) + \eta_j, \quad j = 1, \dots, K, \tag{1}$$

where $\{x_j\}_{j=1}^K$ is the set of points at which u, the field variable (specified later), is observed, η_j the additive noise, $\hat{y}_j$ the actual noisy observations, and m the unknown parameter we wish to invert for. Concatenating all the observations, one can rewrite (1) as

$$\hat{\mathbf{y}} := \mathcal{G}\left(m\right) + \boldsymbol{\eta}, \tag{2}$$

with $\mathcal{G}\left(m\right) := \left[u\left(x_1, m\right), \dots, u\left(x_K, m\right)\right]^T$ denoting the mapping from the distributed parameter m to the noise-free observables, $\boldsymbol{\eta}$ being normally distributed as $\mathcal{N}\left(0, \mathbf{L}\right)$ with bounded covariance matrix $\mathbf{L}$, and $\hat{\mathbf{y}} = \left[\hat{y}_1, \dots, \hat{y}_K\right]^T$.

The inverse problem can be now formulated as

$$\inf_m \mathcal{J} = \frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}\left(m\right)\right|_{\mathbf{L}}^2 \tag{3}$$

subject to

$$\mathcal{A}\left(m, u\right) = f, \tag{4}$$

where $\left|\cdot\right|_{\mathbf{L}}^2 := \left(\cdot, \mathbf{L}^{-1}\cdot\right)$ denotes the weighted Euclidean norm, and $\mathcal{A}$ an operator, e.g. (discretized) differential/integral operator, that maps (m, u) to a function equal to the forcing function f (right hand side of differential/integral equation). Equation (4) is known as the forward equation/model. This optimization problem is however ill-posed. An intuitive reason is that observations $\hat{\mathbf{y}}$ are sparse, and hence they provide limited information about the distributed parameter m. As a result, the Jacobian of the parameter-to-observation map $\mathcal{G}$ contains limited spectral information. Indeed, we have shown that the Gauss-Newton approximation of the Hessian (which is the square of this Jacobian, and is also equal to the full Hessian of the data misfit $\mathcal{J}$ evaluated at the optimal parameter) is a compact operator [5, 6, 7], and hence its range space is effectively finite-dimensional.

One way to overcome the ill-posedness is to use *Tikhonov regularization*, which proposes to solve the nearby problem

$$\inf_m \frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}\left(m\right)\right|_{\mathbf{L}}^2 + \frac{\kappa}{2}\left\|R^{1/2}m\right\|^2, \tag{5}$$

where κ is a regularization parameter, R some regularization operator, and $\left\|\cdot\right\|$ some appropriate norm.

The above method is representative of deterministic inverse solution techniques that typically do not take into account the randomness due to measurements and other sources, though one can equip the deterministic solution with a confidence region by post-processing (see, e.g., [38] and references therein). To incorporate randomness, including that coming from the measurement model (2), into the solution of the inverse problem, one can use a *Bayesian* framework developed, for example, in [17, 28, 26, 35, 33]. In this framework, instead of seeking a distributed parameter m that minimizes the Tikhonov-regularized functional (5), we seek a statistical description of all possible parameters that conform to some prior knowledge and at the same time are consistent with the observations. These parameters are distributed according to the so-called posterior measure. The Bayesian approach does this by reformulating the inverse problem as a problem in *statistical inference*, incorporating uncertainties in the observations, the parameter-to-observable map, and prior information on the parameter. This approach is appealing since it can incorporate most, if not all, kinds of randomness in a systematic manner.

Let us denote μ and ν as the prior and posterior measures on m, respectively. Then, the Bayesian solution [35, 27, 12] to the above inverse problem is given by the Radon-Nikodym derivative

$$\frac{d\nu}{d\mu}\left(m|\hat{\mathbf{y}}\right) \sim \exp\left(-\frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}\left(m\right)\right|^2_{\mathbf{L}}\right). \tag{6}$$

In this paper, we are interested in exploring the posterior (6) using a variational approach. That is, we would like to determine the MAP point (see, e.g., [35] for the definition of the MAP point in finite and infinite dimensional settings). To that end, we first need to specify the prior measure μ. For the sake of clarity of the exposition, we limit ourselves to the case in which the unknown parameter is distributed on one dimensional space. We further assume that it is periodic; particularly, m is a function on one dimensional torus $\mathbb{T} = [0, 1]$ with two end points identified with each other. In order to represent functions on $\mathbb{T}$, we use a wavelet approach [13] so that m can be written as

$$m\left(x\right) := w_0 + \sum_{j=0}^{\infty}\sum_{k=0}^{2^j-1} w_{j,k}\psi_{j,k}\left(x\right), \tag{7}$$

where $\{\psi_{j,k}\}_{k=0}^{2^j-1}$ are orthonormal wavelet basis on the jth level and $w_{j,k}$ are the corresponding wavelet coefficients. Following [25], we define m as a function in the Besov space $B^s_{qq}\left(\mathbb{T}\right)$ if

$$\left\|m\right\|_{B^s_{qq}(\mathbb{T})} := \left(\left|w_0\right|^q + \sum_{j=0}^{\infty} 2^{j\left(qs+\frac{q}{2}-1\right)}\left(\sum_{k=0}^{2^j-1}\left|w_{j,k}\right|^q\right)\right)^{\frac{1}{q}} < \infty.$$

In this paper we restrict ourselves to the interesting case of $s = q = 1$. Furthermore, m is said to be distributed by the Besov prior $B^1_{11}\left(\mathbb{T}\right)$ with parameter κ, denoted as $\left(\kappa, B^1_{11}\left(\mathbb{T}\right)\right)$, if m is given by (7) and if κw_0 and $\kappa 2^{j/2} w_{j,k}$ are independent random variables with probability density

$$\pi\left(w\right) = \frac{1}{2}\exp\left(-\left|w\right|\right).$$

This completes the definition of the Besov prior $\mu := \left(\kappa, B^1_{11}\left(\mathbb{T}\right)\right)$.

We define the MAP point of the posterior ν as

$$m^{MAP} := \arg \inf_{m \in B^1_{11}(\mathbb{T})} J\left(m\right) := \frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}\left(m\right)\right|^2_{\mathbf{L}} + \kappa\left\|m\right\|_{B^1_{11}(\mathbb{T})}.$$

The existence of such a point has been proved (see, eg., [19]).

Next, we approximate the MAP point by discretizing both the prior and the parameter-to-observation map (also known as the likelihood). Owing to the series representation (7), it is natural to approximate the prior by truncating the wavelet expansion. Let us make the truncation at the Nth level so that we have the total of $n := 2^N$ unknown wavelet coefficients, i.e.,

$$\mathbf{w} := \left[w_0, w_{0,0}, w_{1,0}, w_{1,1}, \dots, w_{N-1,2^{N-1}-1}\right] \in \mathbb{R}^n,$$

and define the truncated approximation m_n as

$$m_n\left(x\right) := w_0 + \sum_{j=0}^{N-1} \sum_{k=0}^{2^j-1} w_{j,k} \psi_{j,k}\left(x\right).$$

Clearly, by definition, we have

$$\left\|m_n\right\|_{B^1_{11}(\mathbb{T})} := \left|w_0\right| + \sum_{j=0}^{N-1} \sum_{k=0}^{2^j-1} 2^{j/2} \left|w_{j,k}\right| = \left\|\mathbf{W}\mathbf{w}\right\|_{\ell^1} := \sum_{l=1}^{n} \left|\left(\mathbf{W}\mathbf{w}\right)_l\right|,$$

where $\mathbf{W}$ is the diagonal matrix containing all the weights $2^{j/2}$. An approximation to the MAP point can be defined as

$$m^{MAP}_{n,h} := \arg\min_{\mathbf{w}\in\mathbb{R}^n} J_{n,h}\left(\mathbf{w}\right) := \frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}_h\left(m_n\right)\right|^2_{\mathbf{L}} + \kappa \sum_{l=1}^{n} \left|\left(\mathbf{W}\mathbf{w}\right)_l\right|, \tag{8}$$

where $\mathcal{G}_h$ denotes a discretization of the parameter-to-observation map $\mathcal{G}$ with some parameter h.

At this point, the cost function $J_{n,h}\left(\mathbf{w}\right)$ is clearly not differentiable. To address this issue, we can transform the Besov prior term into an equivalent but differentiable form, and hence the formulation is now amenable to conventional optimization techniques that require gradient and the Hessian of the cost function. Alternatively, we can use/devise special optimization techniques that can cope with the non-differentiability. For example, the work in [25, 20] cast (8) into a convex quadratic programming problem for which an interior path following primal-dual method can be used. An alternative technique can be developed based on the gradient projection approached proposed in [16]. Moreover, by realizing that the cost function (8) is analogous to a ℓ^1-regularization problem, there are several existing optimization schemes that can be used (see, e.g., [24] for a survey). It should be also mentioned that a split Bregman method for ℓ^1-regularized problems proposed in [18] is fast and easy to implement.

In this paper, we propose to use a scalable optimization solver for the MAP estimation problem (8). The method is a subspace trust region interior reflective Newton conjugate gradient method for bound constrained optimization problems. It combines the rapid locally-quadratic convergence rate properties of Newton's method, the effectiveness of trust region globalization for treating ill-conditioned problems, and the Eisenstat–Walker idea of preventing oversolving (i.e., outside the basin of attraction of the minimizer, the Newton step is computed inexactly, but automatically and adaptively becomes more accurate closer to the minimizer). It uses the Coleman-Li approach [10] so that the convergence is quadratic even if the constraints are active. The method is a modification of the subspace, interior, and conjugate gradient approach in [3]. It has been used as the backbone for large-scale parametric model reduction techniques developed in [9, 4].

2. **Subspace trust region interior reflective Newton conjugate gradient method.** In this section, we present in details a subspace trust region interior reflective Newton conjugate gradient (STIRNCG) method. We begin by casting the cost function (8) into a differentiable equivalence. Let us split the positive and negative parts of $\mathbf{W}\mathbf{w}$ by introducing two positive vectors $\mathbf{w}^+$ and $\mathbf{w}^-$ such that

$$\left|(\mathbf{W}\mathbf{w})_l\right| = \mathbf{w}_l^+ + \mathbf{w}_l^-, \quad \text{and } (\mathbf{W}\mathbf{w})_l = \mathbf{w}_l^+ - \mathbf{w}_l^-.$$

The MAP estimate problem (8) can be then rewritten as

$$m_{n,h}^{MAP} = \arg\min_{\mathbf{w}^\pm \in \mathbb{R}^n} J_{n,h}\left(\mathbf{w}^\pm\right) = \frac{1}{2}\left|\hat{\mathbf{y}} - \mathcal{G}_h\left(m_n\right)\right|_{\mathbf{L}}^2 + \kappa \mathbf{1}^T\left(\mathbf{w}^+ + \mathbf{w}^-\right), \tag{9a}$$

subject to

$$\mathbf{W}\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-, \quad \mathbf{w}^\pm \geq 0. \tag{9b}$$

Here, $\mathbf{1} = \{1, \ldots, 1\}^T$ is the vector with all components equal to 1.

One can observe that the minimization problem (9) is differentiable with respect to $\mathbf{w}^\pm$, assuming $\mathcal{G}$ is. Furthermore, it can be easily viewed as an instance of the following generic bound-constrained optimization problem

$$\min_{\mathbf{z}} \mathcal{F}\left(\mathbf{z}\right) \tag{10a}$$

subject to

$$\mathbf{z}^{min} \leq \mathbf{z} \leq \mathbf{z}^{max}, \tag{10b}$$

where $\mathbf{z} := \left(\mathbf{w}^+, \mathbf{w}^-\right)$, $\mathbf{z}_i^{min} = 0$, $\mathbf{z}_i^{max} = \infty$, $i = 1, \ldots, 2n$, and $\mathcal{F} := J_{n,h}$.

In the literature, there are several methods to solve the above bound-constrained optimization problem (see, e.g., [29, 40, 15, 22] and references therein). In particular, the method of Coleman-Li [10] is adopted here. Since our main goal is to make the cost of solving the optimization problem as small as possible, we shall combine the modified Coleman-Li scaling developed in [21] and the subspace trust region interior reflective method in [3]. The following is a detailed exposition of the STIRNCG method proposed in [4, 9].

The first order necessary optimality system for the bound constrained problem (10) can be written in the following form

$$\mathbf{D}\left(\mathbf{z}\right)\nabla\mathcal{F}\left(\mathbf{z}\right) = 0 \tag{11}$$

where the diagonal elements of the Coleman-Li diagonal scaling matrix $\mathbf{D}$ are given by

$$\mathbf{D}_{ii}\left(\mathbf{z}\right) = \mathbf{D}_{ii}^{CL}\left(\mathbf{z}\right) = \begin{cases} \mathbf{z}_i - \mathbf{z}_i^{min} & \text{if } \nabla\mathcal{F}_i > 0 \\ \mathbf{z}_i^{max} - \mathbf{z}_i & \text{if } \nabla\mathcal{F}_i < 0 \\ \min\left\{\mathbf{z}_i - \mathbf{z}_i^{min}, \mathbf{z}_i^{max} - \mathbf{z}_i\right\} & \text{otherwise} \end{cases},$$

where $\nabla\mathcal{F}_i$ denotes the gradient of $\mathcal{F}$ with respect to $\mathbf{z}_i$. We define $\mathbf{D}_{ii} = 1$ if the right hand side is infinity, that is, no scaling is needed in this case. Next, the Newton step $\mathbf{s}$ for (11) at a current optimization point, $\mathbf{z}^k$, satisfies

$$\mathbf{M}\left(\mathbf{z}^k\right)\mathbf{s} = -\mathbf{D}\left(\mathbf{z}^k\right)\nabla\mathcal{F}\left(\mathbf{z}^k\right),$$

where $\mathbf{M} = \mathbf{D}\nabla^2\mathcal{F} + diag\left(|\nabla\mathcal{F}|\right)$. Since the Coleman-Li scaling yields linear convergence for degenerate cases when some bound constraints are active at the optimal solution $\mathbf{z}^*$, i.e., $\nabla \mathcal{F}_i = 0$ if $\mathbf{z}_i^* \in \left\{\mathbf{z}_i^{min}, \mathbf{z}_i^{max}\right\}$, we follow [21] to use the following modified scaling diagonal matrix

$$\mathbf{D}_{ii}\left(\mathbf{z}\right) = \begin{cases} \mathbf{D}_{ii}^{CL}\left(\mathbf{z}\right) & \text{if } \left|\nabla \mathcal{F}_i\right| < \min\left\{\mathbf{z}_i - \mathbf{z}_i^{min}, \mathbf{z}_i^{max} - \mathbf{z}_i\right\}^2 \\ \mathbf{D}_{ii}^{CL}\left(\mathbf{z}\right) & \text{if } \min\left\{\mathbf{z}_i - \mathbf{z}_i^{min}, \mathbf{z}_i^{max} - \mathbf{z}_i\right\} < \left|\nabla \mathcal{F}_i\right|^2 \\ 1 & \text{otherwise} \end{cases} .$$

Now, following [3] we use a subspace trust region interior reflective Newton framework to solve the bound-constrained optimization problem (10). The subspace trust region subproblem we need to solve is given as

$$\min_{\mathbf{s}\in\mathbb{R}^n} \left\{\varphi^k\left(\mathbf{s}\right) : \left\|\mathbf{D}\mathbf{s}\right\|_2 \le \Delta^k \text{ and } \mathbf{s} \in \mathcal{S}^k\right\}, \tag{12}$$

where Δ^k is the current trust region radius, whose updating rule is given in Algorithm 3, $\mathcal{S}^k$ a (at most) two dimensional subspace determined by Algorithm 2, and $\varphi^k\left(\mathbf{s}\right)$ the merit function given by

$$\varphi^k\left(\mathbf{s}\right) = \mathbf{s}^T\nabla\mathcal{F}\left(\mathbf{z}^k\right) + \frac{1}{2}\mathbf{s}^T\mathbf{D}^{-1}\left(\mathbf{z}^k\right)\mathbf{M}\left(\mathbf{z}^k\right)\mathbf{s}.$$

We are in the position to present the bound-constrained optimization solver in Algorithm 1. The algorithm is defined to converge if either

$$\left|\mathcal{F}\left(\mathbf{z}^k + \mathbf{s}\right) - \mathcal{F}\left(\mathbf{z}^k\right)\right| \le \varepsilon_F\left(1 + \mathcal{F}\left(\mathbf{z}^k\right)\right), \text{ or } \left\|\mathbf{s}\right\|_2 \le \varepsilon_X, \text{ or } \left\|\nabla\mathcal{F}\left(\mathbf{z}^k\right)\right\| \le \varepsilon_G,$$

is true, where $\varepsilon_F, \varepsilon_X$ and ε_G are prescribed tolerances for the cost functional value, optimization variables, and the gradient, respectively. The detail on how to carry out a simple reflection strategy can be referred to [3].

Algorithm 1 Bound-constrained optimization solver

```
while not converged do
    At the current Newton step $\mathbf{z}^k$, compute the gradient $\nabla\mathcal{F}\left(\mathbf{z}^k\right)$.
    Compute the subspace $\mathcal{S}^k$ according to Algorithm 2.
    Solve the subspace trust region problem (12).
    Compute the ratio between the actual and the predicted reductions
        $ared := \mathcal{F}\left(\mathbf{z}^k + \mathbf{s}\right) - \mathcal{F}\left(\mathbf{z}^k\right), \quad pred := \varphi^k\left(\mathbf{s}\right), \quad \rho := \frac{ared}{pred}.$
    Update $\mathbf{z}^{k+1} \leftarrow \mathbf{z}^k$ and the trust region radius $\Delta^{k+1} \leftarrow \Delta^k$ via Algorithm 3.
end while
```

3. **Numerical examples.** In this section, we consider two inverse problems; a deconvolution problem and an inverse problem governed by an elliptic partial differential equation (PDE). In both problems, the Besov prior is employed. All the codes are written in `Matlab`, and CPU time are computed using `tic-toc` functions. We begin with the former and compare three different optimization techniques, namely a split Bregman method, an interior path following primal dual approach, and the STIRNCG technique, in estimating the MAP point. We then choose the most efficient method to solve the latter. In particular, Section 3.1 presents briefly the deconvolution problem of interest and the discretization technique. This is then followed by an application of the STIRNCG method in Section 3.1.1, the interior path following primal dual approach in Section 3.1.2, and the split Bregman

Algorithm 2 Subspace computation with Eisenstat–Walker inexactness

```
STEP 1: Solve the Newton equation $\mathbf{D}^{-1}\mathbf{M}\mathbf{s} = -\nabla\mathcal{F}$ using the conjugate gradient
    (CG) method.
if negative curvature direction is detected then
    Set $posdef = 0$ and stop the CG solver
    return  $\mathbf{s}$, $\mathbf{d}$, $posdef$, and $kcg$
else if the residual of the Newton equation is less than or equal to
    $\min\left(1.e-1, \|\nabla\mathcal{F}\|_2\right)\|\nabla\mathcal{F}\|_2$ then
    Set $posdef = 1$ and stop the CG solver
    return  $\mathbf{s}$, $\mathbf{d}$, $posdef$, and $kcg$
else if the number of CG iterations is larger than $n/2$ then
    Set $posdef = 1$ and stop the CG solver
    return  $\mathbf{s}$, $\mathbf{d}$, $posdef$, and $kcg$
end if
STEP 2: Compute the subspace $\mathcal{S}^k$ as follows
if $posdef = 1$ then
    if $kcg = 1$ then
        $\mathcal{S}^k = span\left\{\mathbf{D}sgn\left(\nabla\mathcal{F}\right)\right\}$
    else
        $\mathcal{S}^k = span\left\{\mathbf{D}\nabla\mathcal{F}, \mathbf{s}\right\}$
    end if
else if $posdef = 0$ then
    if $kcg = 1$ then
        $\mathcal{S}^k = span\left\{\mathbf{D}sgn\left(\nabla\mathcal{F}\right)\right\}$
    else
        $\mathcal{S}^k = span\left\{\mathbf{D}sgn\left(\nabla\mathcal{F}\right), \mathbf{d}\right\}$
    end if
end if
```

method in Section 3.1.3, to solve the resulting discretized problem. A detailed comparison among these methods in terms of computation time, scalability, etc will be discussed in each of these sections. As shall be shown, the STIRNCG method seems to be superior to the others, and thus it will be employed to solve the elliptic PDE-constrained problem in Section 3.2. To further confirm the scalability of the STIRNCG approach, we also compare it with the split Bregman method.

3.1. **A deconvolution problem.** This section will describe a one dimensional deconvolution and its discretization using trapezoidal rule and truncated wavelet expansion. Similar to [25], we consider a deconvolution problem in which the observations $u\left(x_j\right)$ is given by

$$u\left(x_j, m\right) := \int_{\mathbb{T}} G\left(x_j - y\right) m\left(y\right)\, dy$$

with the kernel G specified as

$$G\left(x\right) := \begin{cases} c\left(x+a\right)^2\left(x-a\right)^2 & \text{if } |x| < a \\ 0 & \text{otherwise} \end{cases}.$$

Here, c is a normalized constant such that $\int_{\mathbb{T}} G\left(x\right) dx = 1$, and $a = 0.04$. We approximate the integral using the trapezoidal rule on a uniform mesh of size $h =$

Algorithm 3 Trust region radius computation

```
Input: $\mu = 1.e-1, \beta = 0.25, \nu = 0.75, \gamma_0 = 0.0625, \gamma_1 = 0.5, \gamma_2 = 2, \Delta^{max}$ and
    $BT^{max}$
if ($\rho > \mu$ and $ared < 0$) or $ared < 0$ then
    if $\rho \le \beta$ and $\|\mathbf{s}\|_2 < \gamma_1 \Delta^k$ then
        $\Delta^{k+1} = \gamma_1 \Delta^k$
    else if $\rho > \nu$ then
        if $\|\mathbf{s}\|_2 \ge 0.8\Delta^k$ then
            $\Delta^{k+1} = \min\left\{\gamma_2 \Delta^k, \Delta^{max}\right\}$
        else
            $\Delta^{k+1} = \min\left\{\max\left\{\Delta^k, \gamma_2 \|\mathbf{s}\|_2\right\}, \Delta^{max}\right\}$
        end if
    end if
else
    $k_{BT} = 0$
    while $ared \ge 0$ and $k_{BT} \le BT^{max}$ do
        if $\rho \le 0$ then
            $\Delta^{k+1} = \gamma_0 \min\left\{\Delta^k, \|\mathbf{s}\|_2\right\}$
        else
            $\Delta^{k+1} = \gamma_1 \min\left\{\Delta^k, \|\mathbf{s}\|_2\right\}$
        end if
        Compute the subspace $\mathcal{S}^k$ according to Algorithm 2
        Solve the subspace trust region problem (12).
        Compute $ared, pred$ and $\rho$
        $k_{BT} = k_{BT} + 1$
    end while
end if
Update $\mathbf{z}^{k+1} = \mathbf{z}^k + \mathbf{s}$
```

$1/n$. Recall that $n = 2^N$. Let us denote by $\mathbf{m} = \left\{m_n\left(x_{n,k}\right)\right\}_{k=0}^{2^N-1}$ the collection of values of $m_n\left(x\right)$ evaluated at mesh points

$$x_{n,k} = \frac{k}{2^N}, \quad k = 0, \dots, 2^N - 1.$$

In this case, the discrete parameter-to-observation map simply reads

$$\mathcal{G}_{1/n}\left(\mathbf{m}\right) := \mathbf{A}_n \mathbf{m},$$

where $\mathbf{A}_n$ is the discretization of G using the trapezoidal rule on the mesh $\left\{x_{n,k}\right\}_{k=0}^{n-1}$. We have $K = 64$ observations with $x_j = (j-1)/2^6$, $j = 1, \dots, K$. In order to avoid the inverse crime, we first generate u on a finer and different mesh, then interpolate it at the computational mesh points. To generate $\hat{\mathbf{y}}$, we add a Gaussian noise with covariance matrix $\mathbf{L} = \sigma^2 \mathbf{I}$ where σ is a fraction of the maximum value of $u\left(x_j\right)$ and $\mathbf{I}$ is the $n \times n$ identity matrix.

Since we are using a wavelet approach, the relation between $\mathbf{m}$ and $\mathbf{w}$ can be expressed as

$$\mathbf{B}\mathbf{m} = \mathbf{w}, \text{ and } \mathbf{m} = \mathbf{B}^{-1}\mathbf{w},$$

where $\mathbf{B}$ is the wavelet transform and $\mathbf{B}^{-1}$ the inverse wavelet transform. The MAP estimation problem (8) can be rewritten as

$$m_{n,h}^{MAP} = \arg \min_{\mathbf{w}^{\pm}\in\mathbb{R}^n} J_{n,h}\left(\mathbf{w}^{\pm}\right) = \frac{1}{2}\left|\hat{\mathbf{y}} - \mathbf{A}_n\mathbf{m}\right|^2_{\mathbf{L}} + \kappa\mathbf{1}^T\left(\mathbf{w}^{+} + \mathbf{w}^{-}\right), \tag{13a}$$

subject to

$$\mathbf{m} = \mathbf{B}^{-1}\mathbf{W}^{-1}\left(\mathbf{w}^{+} - \mathbf{w}^{-}\right), \quad \mathbf{w}^{\pm} \geq 0. \tag{13b}$$

This is a convex quadratic optimization problem with respect to $\mathbf{w}^{\pm}$ and we are going to use three different solution methods.

3.1.1. *STIRNCG.* In this section we will first provide the expressions for the gradient and Hessian-vector product that are required by the STIRNCG method. Next, we will study the discretization-invariant property and the sparsity of the MAP estimation using the Haar wavelet as the number of wavelet levels increases. We also record the CPU time and the number of Newton iterations to demonstrate the scalability of the STIRNCG method. The robustness with respect to noise of the MAP estimation using Besov prior is also studied, and the behavior of the MAP as the "regularization parameter" κ varies is investigated. Finally, we numerically study the MAP estimations for different type of wavelets, and compare the STIRNCG approach with a popular $\varepsilon-$trick.

The STIRNCG method has been already described in Section 2. This method requires the gradient and Hessian-vector products. It is straightforward to see that the gradient is given by

$$\frac{\partial J_{n,h}}{\partial \mathbf{w}^{+}} = \frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\mathbf{A}_n^T\left(\mathbf{A}_n\mathbf{m} - \hat{\mathbf{y}}\right) + \kappa\mathbf{1},$$

$$\frac{\partial J_{n,h}}{\partial \mathbf{w}^{-}} = -\frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\mathbf{A}_n^T\left(\mathbf{A}_n\mathbf{m} - \hat{\mathbf{y}}\right) + \kappa\mathbf{1},$$

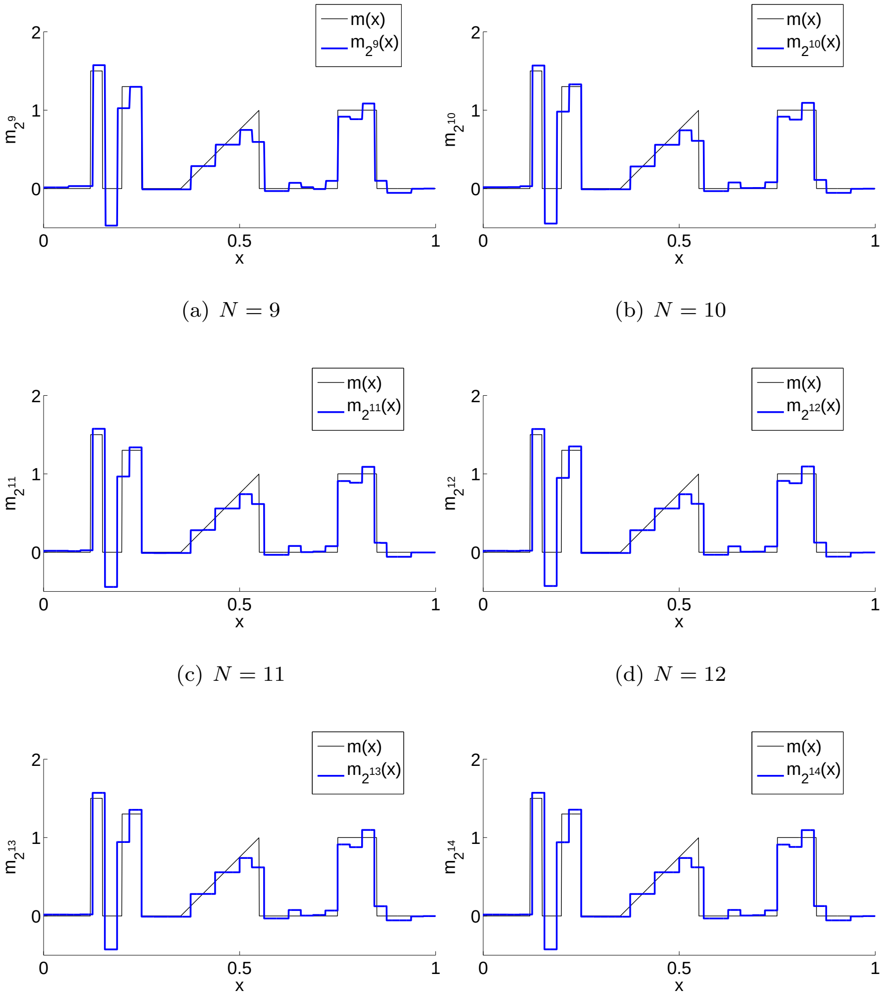
while the product of the Hessian with $(\tilde{\mathbf{w}}^{+}, \tilde{\mathbf{w}}^{-})$ reads

$$\frac{\partial^2 J_{n,h}}{\partial \mathbf{w}^{+}\partial \mathbf{w}} \cdot \left(\tilde{\mathbf{w}}^{+}, \tilde{\mathbf{w}}^{-}\right) = \frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\mathbf{A}_n^T\mathbf{A}_n\mathbf{B}^{-1}\mathbf{W}^{-1}\left(\tilde{\mathbf{w}}^{+} - \tilde{\mathbf{w}}^{-}\right),$$

$$\frac{\partial^2 J_{n,h}}{\partial \mathbf{w}^{-}\partial \mathbf{w}} \cdot \left(\tilde{\mathbf{w}}^{+}, \tilde{\mathbf{w}}^{-}\right) = -\frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\mathbf{A}_n^T\mathbf{A}_n\mathbf{B}^{-1}\mathbf{W}^{-1}\left(\tilde{\mathbf{w}}^{+} - \tilde{\mathbf{w}}^{-}\right).$$

We arbitrarily take zeros as the initial guess for the STIRNCG solver. The tolerances for the cost functional value, optimization variables, and the gradient are chosen as $\varepsilon_F = 5.e-3$, $\varepsilon_X = 1.e-3$, and $\varepsilon_G = 1.e-3$, respectively. For the observation noise, we take $\sigma = 0.05\max_j\left\{u\left(x_j\right)\right\}$ which accounts for 5% percent noise. We arbitrarily choose $\kappa = 20$ for the Besov regularization parameter.

Figure 1 shows the MAP estimations on top of the target function m for various values of $N = \{9, 10, 11, 12, 13, 14\}$ (equivalently $n = \{512, 1024, 2048, 4096, 8192, 16384\}$) using the Haar wavelets. As can be seen, the structure of the MAP is the same for all n. Clearly, $B^1_{11}\left(\mathbb{T}\right)$ is discretization-invariant since the MAP converges as n increases; indeed, the MAP does not seem to change for $n \geq 12$. More quantitatively, we compute the relative L^2-error between the MAP for each n with that of the finest resolution, and record the result in the third row of Table 1. Clearly, the error approaches zero as the resolution increases. One of the appealing features of Besov priors is the edge-preserving property [27, 25], and this is clearly confirmed in Figure 1. Table 1 also presents the CPU time taken (in seconds), the number of Newton iterations (the number of sequential quadratic programings), and the number of nonzero wavelet coefficients (nnz). One can observe that the number of Newton iterations is independent of the number of wavelet coefficients. Since either the wavelet transform or inverse wavelet transform takes $\mathcal{O}(n)$ operations, we expect that the time taken to solve the deconvolution problem scales linearly in n and this is clearly demonstrated on the second row of Table 1. The sparsity promoting ability of the Besov prior for MAP construction is also shown on the fourth row, in which we can see that the number of nonzero number of wavelet coefficients is very small and essentially independent of the total number of wavelet coefficients. For example, with $N = 14$ the sparsity density is $25/2^{14} = 0.15\%$.



(a) $N = 9$

(b) $N = 10$

(c) $N = 11$

(d) $N = 12$

(e) $N = 13$

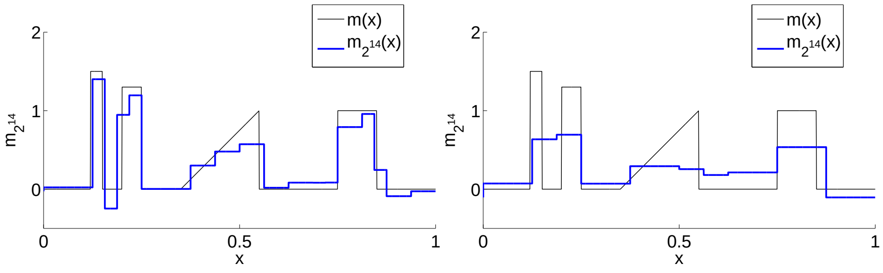
(f) $N = 14$

Figure 1. The target function $m(x)$ and its finite dimensional reconstructions with Besov prior $\left(20, B_{11}^1(\mathbb{T})\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. Noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u(x_j)\}$.

Table 1. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B_{11}^{1}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. Noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u\left(x_j\right)\}$.

N	9	10	11	12	13	14
time (s)	2.35	2.35	3.10	6.42	15.46	49.83
relative L^2-error (%)	16.78	11.50	7.78	4.90	2.61	0.00
nnz	30	30	28	26	25	25
# Newtons	7	7	7	9	9	9

Next, we study how the MAP estimation changes as the noise model varies, particularly with 10% and 20% noise. Figure 2 shows the estimations for $N = 14$. As can be observed, though the MAP structure is generally similar to that of 5% noise, the magnitude is reduced as noise increases. The reason is that the first term (the misfit) in the cost functional (13a) decreases while penalty due to the second term (the prior) stays the same. Consequently, wavelet coefficients (and hence the magnitude of MAP estimations) must reduce for the cost to be minimized. Similar to Table 1, Tables 2 and 3 show, for 10% and 20% noise, respectively, the linear time taken by our STIRNCG solver, discretization-invariance (converging to zero of the relative L^2-error), the fixed number of Newton iterations regardless of the number of wavelet coefficients, and the constant number of nonzero coefficients in the MAP estimation. This highlights the robustness of the Besov prior, and hence the STIRNCG solver, with respect to noise. These tables also present a trend that, as the noise increases, the number of Newton iterations, the number of nonzero wavelet coefficients, and the CPU time are less. This is expected since the Besov regularization term increasingly dominates the misfit term, and hence the problem of finding the MAP point is more well-posed. It should be pointed out that as the noise increases the MAP has simpler structure and hence the convergence of the relative L^2-error is faster as shown in Tables 1, 2, and 3.



(a) Noise $\sigma = 0.1 \max_j \{u\left(x_j\right)\}$

(b) Noise $\sigma = 0.2 \max_j \{u\left(x_j\right)\}$

Figure 2. The target function $m\left(x\right)$ and its finite dimensional reconstructions with Besov prior $\left(20, B_{11}^{1}\left(\mathbb{T}\right)\right)$ and Haar wavelets for $N = 14$. Noise level is 10%, i.e., $\sigma = 0.1 \max_j \{u\left(x_j\right)\}$ or 20%, i.e., $\sigma = 0.2 \max_j \{u\left(x_j\right)\}$.

TABLE 2. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. Noise level is 10%, i.e., $\sigma = 0.1 \max_j \{u(x_j)\}$.

N	9	10	11	12	13	14
time (s)	1.06	1.12	1.52	2.82	7.26	23.88
relative L^2-error (%)	15.43	10.43	7.01	4.47	2.37	0.00
nnz	20	21	18	18	18	18
# Newtons	4	4	4	4	4	4

TABLE 3. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. Noise level is 20%, i.e., $\sigma = 0.2 \max_j \{u(x_j)\}$.

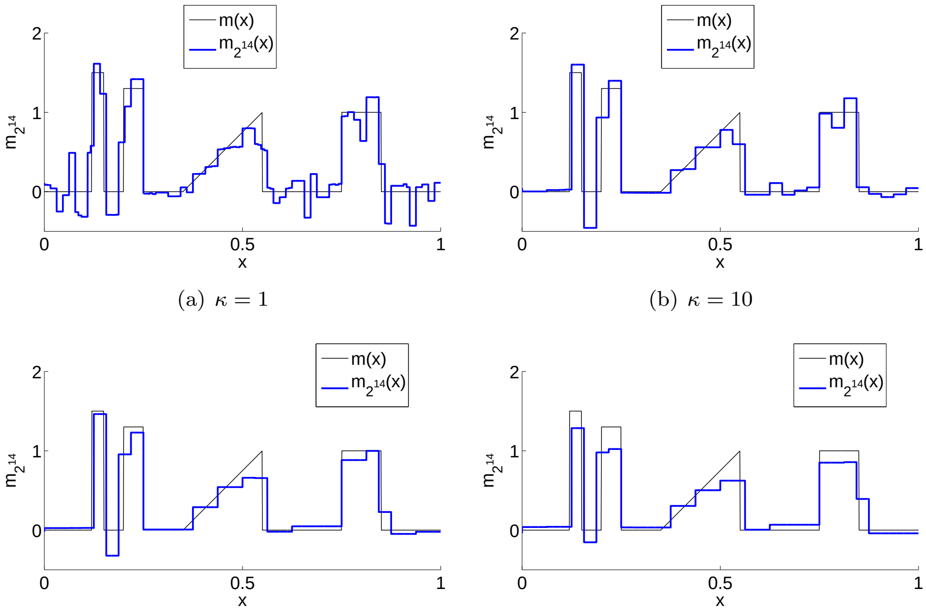
N	9	10	11	12	13	14
time (s)	0.54	0.58	0.60	1.19	3.47	11.77
relative L^2-error (%)	8.47	5.85	3.94	2.50	1.33	0.00
nnz	8	9	10	9	9	9
# Newtons	3	3	2	2	2	2

A question arises on the behavior of the MAP estimation and the STIRNCG solver as the regularization parameter κ varies. (Note that the optimization formulation also varies, and hence the MAP estimators.) To address it, we choose $\kappa = \{1, 10, 50, 100\}$, 5% noise, and $N = 14$. Figure 3 shows the MAP estimations together with the target. As expected, the MAP is noisier as κ decreases since the misfit contribution increases (and hence estimations match the noisy data more closely). Consequently, we also expect that the MAP is less sparse and that the number of Newton iterations generally grows (and hence the CPU time). These are confirmed in Table 4.

TABLE 4. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(\kappa, B^1_{11}\left(\mathbb{T}\right)\right)$, where $\kappa = \{1, 10, 50, 100\}$ and $N = 14$. Noise level is 5%, i.e., $\sigma = 0.05 \max\{m\}$.

κ	1	10	50	100
time (s)	67.83	68.14	36.20	29.20
nnz	110	34	25	20
# Newtons	8	10	6	6

Let us now consider other wavelet bases, namely, Daubechies 7, Beylkin, Coiflet 2, and Vaidyanathan, and see whether the MAP is still sparse and the scalability of the STIRNCG solver still holds. We also expect the structure of the MAP will change since these bases are smooth as opposed to the Haar basis. The MAP



(a) $\kappa = 1$

(b) $\kappa = 10$

(c) $\kappa = 50$

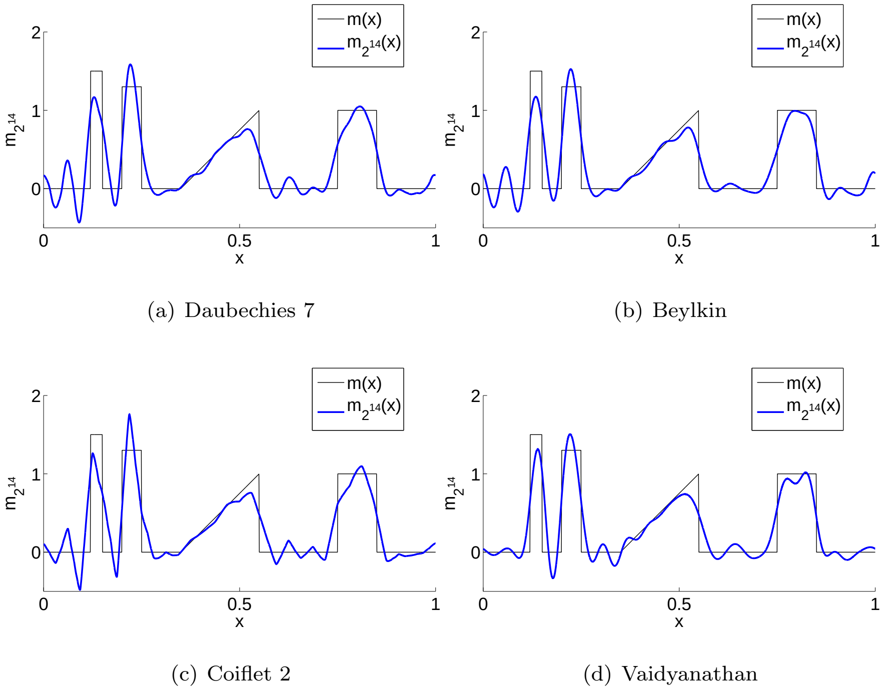
(d) $\kappa = 100$

Figure 3. The target function $m(x)$ and its finite dimensional reconstructions with Besov prior $\left(\kappa, B^1_{11}(\mathbb{T})\right)$ and Haar wavelets, where $\kappa = \{1, 10, 50, 100\}$ and $N = 14$. Noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u(x_j)\}$.

estimations are shown in Figure 4. Clearly, the edge-preserving property is lost while the linear part of the target is reconstructed quite well. This is the direct consequence of smoothness of these wavelet bases and wavelet expansion truncation. Table 5 compares the CPU time, the number of nonzero wavelet coefficients, and the number of Newton iterations. The results indicate that, using either of these orthonormal wavelet bases, the sparsity of the MAP and the scalability of the STIRNCG solver remains unchanged. To the end of the paper, we will exclusively use the Haar wavelets.

Table 5. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}(\mathbb{T})\right)$. Different orthonormal wavelets bases are used: Daubechies 7, Beylkin, Coiflet 2, and Vaidyanathan wavelets for $N = 14$. Noise level is 5%, i.e., $\sigma = 0.05 \max\{m\}$.

Wavelet type	Daubechies 7	Beylkin	Coiflet 2	Vaidyanathan
time (s)	29.04	30.27	30.53	33.20
nnz	29	29	27	27
# Newtons	4	4	5	4



(a) Daubechies 7

(b) Beylkin

(c) Coiflet 2

(d) Vaidyanathan

FIGURE 4. The target function $m(x)$ and its finite dimensional reconstructions with Besov prior $\left(20, B_{11}^1(\mathbb{T})\right)$. Different orthonormal wavelets bases are used: Daubechies 7, Beylkin, Coiflet 2, and Vaidyanathan wavelets for $N = 14$. Noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u(x_j)\}$.

It should be pointed out that the trick in Section 2 to cast the $B_{11}^1(\mathbb{T})$ regularized problem into an equivalent differentiable one is not unique. For example, in the context of deconvolution problem one can replace the non-smooth cost function (8) by the following "ε-trick"

$$m_{n,h}^{MAP}(\varepsilon) := \arg\min_{\mathbf{m}\in\mathbb{R}^n} J_{n,h}(\mathbf{m}) = \frac{1}{2}\left|\hat{\mathbf{y}} - \mathbf{A}_n\mathbf{m}\right|_{\mathbf{L}}^2 + \kappa\sum_{j=1}^n \sqrt{(\mathbf{WBm})_j^2 + \varepsilon},$$

for which we can use the STIRNCG method to find $m_{n,h}^{MAP}(\varepsilon)$. In this case, there are only n optimization variables instead of $2n$, but the MAP estimate depends on ε. One can expect that $m_{n,h}^{MAP}(\varepsilon)$ converges to that of (8) as ε approaches zero. However, our numerical experiments show that the STIRNCG method requires more and more Newton iterations as ε gets smaller; this is expected since the problem is closer to the original non-smooth optimization problem. In order to obtain MAP estimates reasonably close to those in Figure 1, we take $\varepsilon = 10^{-6}$, $\varepsilon_F = 5.e-6$, $\varepsilon_X = 1.e-6$, and $\varepsilon_G = 1.e-6$. Compared to Table 1, Table 6 shows that the ε-trick does not seem to be a good method since it takes significantly more time, and many more (varying) number of Newton iterations though it has half less number of optimization variables. As can also be observed, all the wavelet coefficients in MAP estimates are nonzero and hence MAP sparsity is lost. Nevertheless, it is possible to post-process, via thresholding, the wavelet coefficients to recover the sparsity. In particular, we set wavelets coefficients of the MAP estimates to zero if their absolute value smaller than τ. We find that with $\tau = 10^{-4}$ the quality of the MAP estimates is the same (in the eyeball norm) while sufficiently small sparsity is observed; this is shown on the fifth line of Table 6. It is important to point out that no thresholding was needed for the STIRNCG results in Tables 1–5.

To the rest of this section, we take the MAP estimation on the finest resolution in Table 1 as the reference solution. As can be seen on the third line of Table 6, the relative L^2-error with respect to this reference solution converges to zero, and thus the MAP estimation of the ε-trick is also discretization-invariant.

Table 6. Time taken, the number of Newton iterations (# Newtons), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. The ε-trick method is used and the noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u\left(x_j\right)\}$.

N	9	10	11	12	13	14
time (s)	42.52	42.60	38.55	73.64	192.00	477.47
relative L^2-error (%)	16.75	11.52	7.74	4.91	2.62	1.00
nnz	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}
nnz (thresholding)	110	110	129	156	177	197
# Newtons	32	32	23	23	43	48

3.1.2. *Interior path following primal-dual approach.* In this section we present an interior path following primal-dual approach to solve the deconvolution problem and compare its performance, using both direct and iterative solvers, with that of the STIRNCG approach. The detailed description of the method can be found in, e.g., [25, 20] and the references therein. Our goal here is to give a brief introduction to the method and our actual implementation for the readers' convenience. To begin, we follow [25, 20] and cast (13) into the standard quadratic programming problem

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

subject to

$$\mathbf{A}\mathbf{x} = \mathbf{b} := M\mathbf{W}\mathbf{B}\mathbf{1},$$
$$\mathbf{x} \geq \mathbf{0},$$

where

$$\mathbf{Q} := \begin{bmatrix} \frac{1}{\sigma^2}\mathbf{A}_n^T\mathbf{A_n} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{A} := \begin{bmatrix} \mathbf{W}\mathbf{B} & -\mathbf{I} & \mathbf{I} \end{bmatrix}, \mathbf{c} := \begin{bmatrix} -\frac{1}{\sigma^2}\mathbf{A}_n^T\hat{\mathbf{y}} - \frac{M}{\sigma^2}\mathbf{A}_n^T\mathbf{A}_n\mathbf{1} \\ \kappa\mathbf{1} \\ \kappa\mathbf{1} \end{bmatrix}.$$

Here, M is an arbitrary large positive number and $\mathbf{x} = \left[\mathbf{m} + M, \mathbf{w}^+, \mathbf{w}^-\right]^T$. Using the log-barrier approach together with the Newton method [39] we end up with solving the following linear system

$$\begin{bmatrix} \mathbf{Q} & -\mathbf{A}^T & -\mathbf{I} \\ -\mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{Z} & \mathbf{0} & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{z} \end{bmatrix} = \begin{bmatrix} -\mathbf{Q}\mathbf{x} + \mathbf{A}^T\mathbf{y} + \mathbf{z} - \mathbf{c} \\ \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mu\mathbf{1} - \mathbf{X}\mathbf{z} \end{bmatrix}, \tag{14}$$

where $\mathbf{X} := diag\left(\mathbf{x}\right), \mathbf{Z} := diag\left(\mathbf{z}\right)$, and μ is the barrier parameter (also known as central path parameter). We choose to solve the Newton system by the Mehrotra predictor-corrector approach [32] that requires to invert the large spare matrix on the left hand of (14) in both predictor and corrector steps. We carry out this task by the direct method via the Schur complement reduction [1].

To start to algorithm we use the Mehrotra initial guess [32], and to update the solution we employ the following iterative process as suggested in [20]

$$
\begin{aligned}
\mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha_{primal}\Delta\mathbf{x},\\
\mathbf{y}^{k+1} &= \mathbf{y}^k + \alpha_{dual}\Delta\mathbf{y},\\
\mathbf{z}^{k+1} &= \mathbf{z}^k + \alpha_{dual}\Delta\mathbf{z},
\end{aligned} \tag{15}
$$

where $\alpha_{dual} = \alpha_d^{max}$ while a polynomial line search with Armijo rule [23] is employed for determining α_{primal} on the interval $[0, \alpha_p^{max}]$. Here,

$$
\begin{aligned}
\alpha_p^{max} &= 0.95 \min\left\{\min_{\Delta\mathbf{x}_i<0}\left(-\frac{\mathbf{x}_i}{\Delta\mathbf{x}_i}\right), 1\right\},\\
\alpha_d^{max} &= 0.95 \min\left\{\min_{\Delta\mathbf{z}_i<0}\left(-\frac{\mathbf{z}_i}{\Delta\mathbf{z}_i}\right), 1\right\}.
\end{aligned}
$$

The central path parameter μ is computed using the suggestion in [34]. Let us summarize this version of the interior point method in Algorithm 4.

Algorithm 4 Interior path following primal dual approach

```
Input: $k = 0, kmax$. Compute the initial guess $\left(\mathbf{x}^0, \mathbf{y}^0, \mathbf{z}^0\right)$ using Mehrotra approach [32].
while $\frac{\left|\left(\mathbf{x}^k\right)^T\mathbf{z}^k\right|}{1+|\mathbf{b}^T\mathbf{y}^k|} > 10^{-7}$ do
    Compute the barrier parameter $\mu^k$ as suggested in [34].
    Solve (14) using the Mehrotra predictor-corrector approach [32].
    Compute $\left(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{z}^{k+1}\right)$ as in (15).
    Set $k = k + 1$.
    if $k > kmax$ then
        return
    end if
end while
```

Since either the forward or inverse wavelet transforms requires $\mathcal{O}\left(n\right)$ operations, we form both matrices $\mathbf{B}$ and $\mathbf{B}^{-1}$ before starting the interior point solver. Finally, we stop the algorithm when either

$$
\frac{\left|\left(\mathbf{x}^k\right)^T\mathbf{z}^k\right|}{1+|\mathbf{b}^T\mathbf{y}^k|} \leq 10^{-7}
$$

is satisfied or the number of iterations exceeds a prescribed maximal value. Note that the tolerance of 10^{-7} is chosen so that MAP estimates have similar quality as those in Figure 1.

The results for various values of $N = \{9, 10, 11, 12, 13\}$ and 5% noise are shown in Table 7. Since we use the direct method, $\mathbf{B}$ needs to be constructed (we accomplish this by acting the wavelet transform on the identity matrix), and the CPU time for carrying this task is shown as B time, which is larger than the CPU time taken by the STIRNCG method, compared to Table 1. The CPU time taken by the Mehrotra algorithm (Mehrotra time) is even more, though the number of Mehrotra iterations is independent of the number of wavelet coefficients. Similar to the ε-trick method, we need to perform thresholding in order to have sparse constructions of the MAP in Table 7; in this case we use $\tau = 10^{-4}$. On the fourth line of the table, we also present the convergence of the relative L^2-error with respect to the reference solution as the number of wavelet levels increases.

Table 7. Time taken to construct $\mathbf{B}$ ($\mathbf{B}$ time), time taken by the Mehrotra algorithm (Mehrotra time), the number of Mehrotra predictor-corrector iterations (# Mehrotras), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13\}$. The interior path following primal dual method is used and the noise level is 5%, i.e., $\sigma = 0.05 \max_j \{u\left(x_j\right)\}$.

N	9	10	11	12	13
$\mathbf{B}$ time (s)	2.78	6.49	14.90	36.87	98.70
Mehrotra time (s)	2.73	18.88	96.41	608.68	4890.93
relative L^2-error (%)	16.78	11.53	7.74	4.91	2.62
nnz	22	22	22	21	22
# Mehrotras	12	12	12	11	12

We have numerically showed that the direct solver for the interior path following primal dual method is time-consuming. Alternatively, one can use iterative solver via the conjugate gradient (CG) method in both predictor and correct steps. Using this approach, we choose to stop the CG algorithm when the residual is less than 10^{-4}. This approach is matrix-free since it does not explicitly require the construction of any matrices. Instead, only matrix-vector products are needed. For example, computing the action of $\mathbf{B}$ on a vector amounts to carrying out a wavelet transform instead of constructing $\mathbf{B}$ explicitly then acting it with that vector. In order to have sparse constructions of the MAP, we use thresholding with $\tau = 10^{-4}$. In Table 8 are the total CPU time, the relative L^2-error, the number of nonzero wavelet coefficients, and the number of Mehrotra iterations. Compared to the direct solver in Table 7, the CG iterative method is much more efficient as the number of wavelet coefficients increases. For example, with $N = 13$ the former is 22 times more expensive than the latter. Nevertheless, it is the inexactness of latter that slows down the convergence of the MAP estimate towards the reference solution as indicated by the relative L^2-error. Obviously, the number of Mehrotra iterations is less for the former than for the latter since both predictor and corrector steps are now solved inexactly. Nevertheless, interior path following primal dual approach with iterative solver is still (more than) an order of magnitude more expensive than the STIRNCG approach (compare Tables 8 and 1). Faster iterative approach can be implemented using preconditioning techniques (see, e.g., [2]). However, to be fair with the STIRNCG and the split Bregman approaches, no preconditioners are employed in this paper[1] Finally, similar to the STIRNCG approach the Mehrotra

[1] Disclaimer: Our Matlab implementations of the interior point method, though we have tried to optimize it as much as we can, are by no means the best.

method with iterative solver seems to scale linearly with respect to the number of wavelet coefficients as shown on the second line of Table 8.

Table 8. Time taken by the Mehrotra algorithm (Mehrotra time), the number of Mehrotra predictor-corrector iterations (# Mehrotras), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13\}$. The interior path following primal dual method with iterative solver is used and the noise level is 5%, i.e., $\sigma = 0.05 \max_j \left\{u\left(x_j\right)\right\}$.

N	9	10	11	12	13
time (s)	43.50	56.65	78.98	132.04	217.90
relative L^2-error (%)	18.64	13.80	12.25	9.35	7.95
nnz	24	25	29	30	30
# Mehrotras	26	26	26	29	27

3.1.3. *A split Bregman approach.* This section presents a split Bregman method to solve the deconvolution problem and compare its performance, using both direct and iterative solvers, with those of the STIRNCG and the interior point methods. An excellent introduction to the split Bregman method is [18] and we do not attempt to repeat it here. Instead, we briefly present our implementation and refer the readers to [18] (and the references therein).

In order to use the split Bregman method presented in [18], we define

$$\mathbf{z} := \kappa \mathbf{W}\mathbf{w},$$

and the MAP estimation problem becomes

$$\min_{\mathbf{w},\mathbf{z}} J_{n,h}\left(\mathbf{w},\mathbf{z}\right) = \frac{1}{2\sigma^2}\left\|\hat{\mathbf{y}} - \mathbf{A}_n\mathbf{B}^{-1}\mathbf{w}\right\|_2^2 + \left\|\mathbf{z}\right\|_{\ell^1}, \tag{16a}$$

subject to

$$\kappa\mathbf{W}\mathbf{w} - \mathbf{z} = \mathbf{0}. \tag{16b}$$

A split Bregman iteration for (16) is presented in Algorithm 5. As can be seen, we need to solve an unconstrained optimization problem on the fourth line of the algorithm. For the deconvolution example, this is simple since the cost function is quadratic. Indeed, the solution is given by

$$\underbrace{\left(\frac{1}{\sigma^2}\mathbf{B}^{-T}\mathbf{A}_n^T\mathbf{A}_n\mathbf{B}^{-1} + \lambda\kappa^2\mathbf{W}^T\mathbf{W}\right)}_{\mathbf{H}}\mathbf{w}^{k+1} = \left(\frac{1}{\sigma^2}\mathbf{B}^{-T}\mathbf{A}_n^T\hat{\mathbf{y}} + \lambda\kappa\mathbf{W}^T\left(\mathbf{z}^k - \mathbf{x}^k\right)\right).$$

Clearly, one can construct $\mathbf{B}^{-1}$ (and hence $\mathbf{H}$) explicitly and this requires $\mathcal{O}\left(n^2\right)$ operations since the cost for each inverse wavelet transform scales like $\mathcal{O}\left(n\right)$. It should be noted that $\mathbf{H}$ is independent of either the loops in Algorithm 5, and hence we need to carry out the Cholesky decomposition of $\mathbf{H}$ only once at the beginning of the algorithm. The work in [18] suggests that one should set $imax = 1$ to compromise the efficiency and accuracy. However, in order to get MAP estimates that have similar quality as those in Figure 1 we find that $imax = 2$ is most efficient. We take $\varepsilon = 10^{-3}$ (line nine of Algorithm 5) and $\lambda = 10$. Note that the split Bregman method has been used for deconvolution problems with q-total variation priors [11] and the author observed that it was more robust than the interior point method.

Algorithm 5 split Bregman method

```
Input: $k = 0, \mathbf{z}^0 = \mathbf{0}, \mathbf{x}^0 = \mathbf{0}, kmax, imax$ and $\varepsilon$.
while 1 do
    Set $\mathbf{z}^k = \kappa \mathbf{W}\mathbf{w}^k$.
    for $i = 1$ to $imax$ do
        Solve $\mathbf{w}^{k+1} = \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\left\|\hat{\mathbf{y}} - \mathbf{A}_n\mathbf{B}^{-1}\mathbf{w}\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{z}^k - \kappa\mathbf{W}\mathbf{w} - \mathbf{x}^k\right\|_2^2$.
        Compute
            $$\mathbf{y} = \kappa\mathbf{W}\mathbf{w}^{k+1} + \mathbf{x}^k$$
            $$\mathbf{z}^{k+1} = diag\left(sgn\left(\mathbf{y}\right)\right)\max\left\{|\mathbf{y}| - \frac{1}{\lambda}, \mathbf{0}\right\}$$
        Set $\mathbf{z}^k = \mathbf{z}^{k+1}$.
    end for
    Compute $\mathbf{x}^{k+1} = \mathbf{x}^k + \kappa\mathbf{W}\mathbf{w}^{k+1} - \mathbf{z}^{k+1}$.
    if $\left\|\mathbf{w}^{k+1} - \mathbf{w}^k\right\|_2 \le \varepsilon\left\|\mathbf{w}^k\right\|_2$ then
        return
    end if
    Set $k = k + 1$.
    if $k > kmax$ then
        return
    end if
end while
```

Table 9 shows similar results to those of Table 7. As can be seen, the Bregman approach with direct solver is much less time-consuming than the interior path following primal dual approach with direct solver. This is partially due to the fact the former has n variables while the latter has $3n$. Note that we have used thresholding with $\tau = 10^{-4}$ to compute the number of nonzero wavelet coefficients, and hence sparse constructions of the MAP. We also observe that the time taken to construct $\mathbf{B}$ dominates the actual computation time. This suggests that, similar to the interior point method, we should replace the direct solver with an iterative one. Again, the fact that our inverse formation with Besov prior is discretization-invariant is clearly verified by the convergence of the relative L^2-error on the fourth line of Table 7. The convergence rate is comparable to that of the interior point method (compared with Table 7).

Since $\mathbf{H}$ is symmetric positive definite, a natural iterative candidate is the CG method which we can terminate early since there is no need to solve the sub-optimization exactly [18]. In particular, we stop the CG algorithm when the residual norm is less than 10^{-3}. Again, we need thresholding with $\tau = 10^{-4}$ to have sparse constructions of the MAP. Table 10 shows the results for the split Bregman method with the inexact CG solver. Clearly, iterative solver is less expensive than the direct one, and, surprisingly, they have the same number of Bregman iterations and approximately the same convergence rate. More importantly, the result shows that the Bregman time scales linearly with the number of wavelet coefficients. Compared to the iterative Mehrotra approach, iterative Bregman is more than three times

Table 9. Time taken to construct $\mathbf{B}$ ($\mathbf{B}$ time), time taken by the Bregman algorithm (Bregman time), the number of Bregman iterations (# Bregmans), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. The noise level is 5%, i.e., $\sigma = 0.05 \max_j \left\{u\left(x_j\right)\right\}$.

N	9	10	11	12	13	14
$\mathbf{B}$ time (s)	2.78	6.49	14.90	36.87	98.70	299.51
Bregman time (s)	0.13	0.29	1.23	4.41	18.44	78.28
relative L^2-error (%)	16.51	11.41	7.73	5.00	2.92	1.39
nnz	25	28	29	31	31	35
# Bregmans	60	56	54	54	53	53

faster. Nevertheless, it is still more than two times slower than the STIRNCG method; compare Table 10 and Table 1.

Table 10. Time taken by the Bregman algorithm (Bregman time), the number of Bregman iterations (# Bregmans), and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(20, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \{9, 10, 11, 12, 13, 14\}$. The noise level is 5%, i.e., $\sigma = 0.05 \max_j \left\{u\left(x_j\right)\right\}$.

N	9	10	11	12	13	14
Bregman time (s)	19.99	25.60	33.57	47.62	70.48	112.63
relative L^2-error (%)	16.51	11.40	7.72	4.90	2.90	1.34
nnz	25	26	28	31	32	35
# Bregmans	60	56	54	54	53	53

3.2. **An inverse coefficient problem governed by elliptic partial differential equation.** In this section, we consider a one dimensional heat conduction problem. The MAP estimation problem in this case reads

$$m_n^{MAP} = \arg \min_{\mathbf{w}^\pm \in \mathbb{R}^n} J_n\left(\mathbf{w}^\pm\right) = \frac{1}{2}\left|\hat{\mathbf{y}} - \hat{\mathbf{u}}\right|^2_{\mathbf{L}} + \kappa \mathbf{1}^T\left(\mathbf{w}^+ + \mathbf{w}^-\right), \tag{17a}$$

subject to

$$\mathbf{m} = \mathbf{B}^{-1}\mathbf{W}^{-1}\left(\mathbf{w}^+ - \mathbf{w}^-\right), \qquad \mathbf{w}^\pm \geq 0, \tag{17b}$$

$$-\nabla \cdot (m \nabla u) = 0 \qquad \text{in } \Omega, \tag{17c}$$

$$-m\nabla u \cdot \mathbf{n} = Bi\, u \qquad \text{on } \partial\Omega \setminus \Gamma_L, \tag{17d}$$

$$-m\nabla u \cdot \mathbf{n} = -1 \qquad \text{on } \Gamma_R, . \tag{17e}$$

Here, we denote $\Omega = (0,1)$, $\Gamma_L = \{0\}$, $\Gamma_R = \{1\}$, m the conductivity, Bi the Biot number, and $\mathbf{n}$ the outward unit normal vector; u is temperature distributed in Ω while $\hat{\mathbf{u}}$ is a vector containing the temperature evaluated at observation points.

We discretize (17c)–(17e) using the standard first order conforming finite element method on a uniform mesh with $h = 2^{-N} = n^{-1}$. For simplicity, we also approximate m using the same finite element space so that

$$m_h\left(x\right) = \sum_{i=1}^{n} \mathbf{m}_i \varphi_i\left(x\right),$$

with φ_i as the finite element nodal basis functions. Consequently, the discretized version of (17c)–(17e) has the form

$$\mathcal{A}\left(\mathbf{m}\right)\mathbf{u} = F, \tag{18}$$

where $\mathbf{u} \in \mathbb{R}^{n+1}$ is the vector of nodal values of u, $F \in \mathbb{R}^{n+1}$ the right-hand side resulting from the boundary condition (17e), and $\mathcal{A}\left(\mathbf{m}\right) \in \mathbb{R}^{(n+1)\times(n+1)}$ the stiffness matrix.

Suppose that we have $K = 32$ observations with $x_j = (j-1)/2^5$, $j = 1, \ldots, K$. Let us denote by Q the observation operator such that $\hat{\mathbf{u}} = Q\mathbf{u} = \left[u\left(x_1\right), \ldots, u\left(x_K\right)\right]^T$. In order to avoid the phenomenon of inverse crimes, we generate our data using a finer computational mesh (with $h = 2^{-15}$) than the one used in the inversion, and then add a Gaussian noise similar to the deconvolution problem in Section 3.1. The fully discrete MAP problem reads

$$m_{n,h}^{MAP} = \arg \min_{\mathbf{w}^{\pm} \in \mathbb{R}^n} J_{n,h}\left(\mathbf{w}^{\pm}\right) = \frac{1}{2\sigma^2}\left|\hat{\mathbf{y}} - Q\mathbf{u}\right|^2 + \kappa \mathbf{1}^T\left(\mathbf{w}^+ + \mathbf{w}^-\right), \tag{19a}$$

subject to

$$\mathbf{m} = \mathbf{B}^{-1}\mathbf{W}^{-1}\left(\mathbf{w}^+ - \mathbf{w}^-\right), \qquad \mathbf{w}^{\pm} \geq 0, \tag{19b}$$

$$\mathcal{A}\left(\mathbf{m}\right)\mathbf{u} = F \tag{19c}$$

In order to apply the STIRNCG method to solve (19), we use the reduced space approach in which u is considered as a function of $\mathbf{m}$. By the chain rule, we obviously have the gradients as

$$\frac{\partial J_{n,h}}{\partial \mathbf{w}^+} = \mathbf{W}^{-T}\mathbf{B}^{-T}\frac{\partial J_{n,h}}{\partial \mathbf{m}} + \kappa \mathbf{1}, \quad \frac{\partial J_{n,h}}{\partial \mathbf{w}^-} = -\mathbf{W}^{-T}\mathbf{B}^{-T}\frac{\partial J_{n,h}}{\partial \mathbf{m}} + \kappa \mathbf{1}.$$

On the other hand, one can use the adjoint method to have

$$\frac{\partial J_{n,h}}{\partial \mathbf{m}_i} = \mathbf{v}^T \frac{\partial \mathcal{A}\left(\mathbf{m}\right)}{\partial \mathbf{m}_i}\mathbf{u}, \quad i = 1, \ldots, n,$$

where the adjoint vector $\mathbf{v}$ solves

$$\mathcal{A}\left(\mathbf{m}\right)^T \mathbf{v} = \frac{1}{\sigma^2} Q^T\left(\hat{\mathbf{y}} - Q\mathbf{u}\right).$$

Now, the product of the Hessian with a vector $\left(\tilde{\mathbf{w}}^+, \tilde{\mathbf{w}}^-\right)$, whereby we define $\tilde{\mathbf{m}} = \mathbf{B}^{-1}\mathbf{W}^{-1}\left(\tilde{\mathbf{w}}^+ - \tilde{\mathbf{w}}^-\right)$, can be written as

$$\frac{\partial^2 J_{n,h}}{\partial \mathbf{w}^+ \partial \mathbf{w}} \cdot \left(\tilde{\mathbf{w}}^+, \tilde{\mathbf{w}}^-\right) = \frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\left(\frac{\partial^2 J_{n,h}\mathbf{m}}{\partial \mathbf{m}^2} \cdot \tilde{\mathbf{m}}\right),$$

$$\frac{\partial^2 J_{n,h}}{\partial \mathbf{w}^- \partial \mathbf{w}} \cdot \left(\tilde{\mathbf{w}}^+, \tilde{\mathbf{w}}^-\right) = -\frac{1}{\sigma^2}\mathbf{W}^{-T}\mathbf{B}^{-T}\left(\frac{\partial^2 J_{n,h}\mathbf{m}}{\partial \mathbf{m}^2} \cdot \tilde{\mathbf{m}}\right).$$

Let us define $\mathbf{H} := \frac{\partial^2 J_{n,h}\mathbf{m}}{\partial \mathbf{m}^2} \cdot \tilde{\mathbf{m}}$. Then, by using the adjoint method the second time, we have

$$\mathbf{H}_i = \mathbf{v}^T \left(\sum_{j=1}^{n} \frac{\partial^2 \mathcal{A}(\mathbf{m})}{\partial \mathbf{m}_i \partial \mathbf{m}_j} \tilde{\mathbf{m}}_j \right) \mathbf{u} + \tilde{\mathbf{v}}^T \frac{\partial \mathcal{A}(\mathbf{m})}{\partial \mathbf{m}_i} \mathbf{u} + \mathbf{v}^T \frac{\partial \mathcal{A}(\mathbf{m})}{\partial \mathbf{m}_i} \tilde{\mathbf{u}}, \quad i = 1, \dots, n,$$

where the incremental state vector $\tilde{\mathbf{u}}$ solves

$$\mathcal{A}(\mathbf{m}) \tilde{\mathbf{u}} = -\left(\sum_{j=1}^{n} \frac{\partial \mathcal{A}(\mathbf{m})}{\partial \mathbf{m}_j} \tilde{\mathbf{m}}_j \right) \mathbf{u},$$

and the incremental adjoint vector $\tilde{\mathbf{v}}$ solves

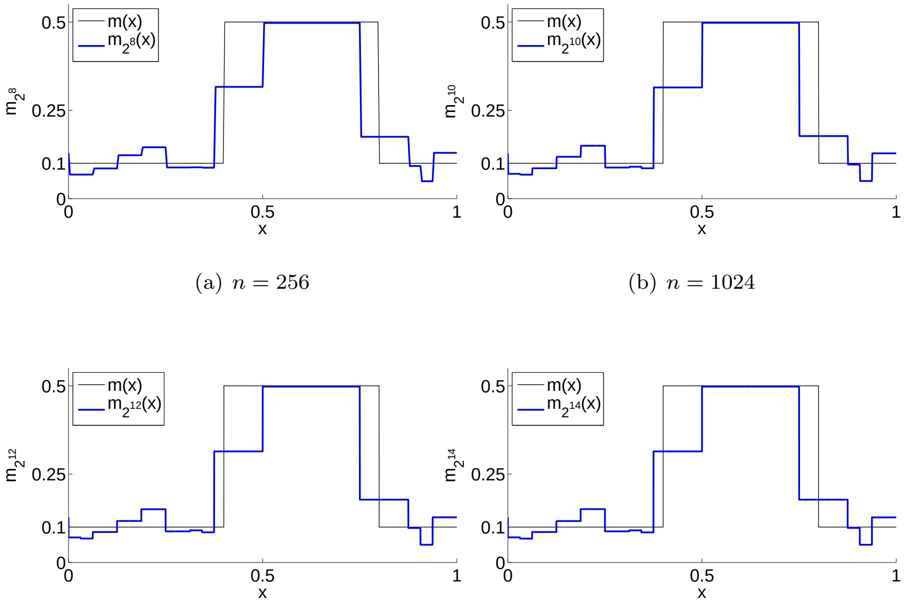
$$\mathcal{A}(\mathbf{m})^T \tilde{\mathbf{u}} = -\left(\sum_{j=1}^{n} \frac{\partial \mathcal{A}(\mathbf{m})}{\partial \mathbf{m}_j} \tilde{\mathbf{m}}_j \right) \mathbf{v} - \frac{1}{\sigma^2} Q^T Q \tilde{\mathbf{u}},$$

For the following numerical results, we take $Bi = 0.1$, $\sigma = 0.01 \max_j \{u(x_j)\}$, and $\kappa = 14$. The initial guess for $(\mathbf{w}^+, \mathbf{w}^-)$ is obtained from the wavelet transform of $m(x) = 0.1$. Note that a rigorous way to ensure the positiveness of the conductivity is to work with $m = \exp(\gamma)$, see e.g. [8]; however, for all numerical results in this section our code seems to always produce positive m, and for that reason we use the former for simplicity. For our STIRNCG solver, we use $\varepsilon_F = 5.e-8$, $\varepsilon_X = 1.e-8$, and $\varepsilon_G = 1.e-8$.

Figure 5 shows solutions $m_n(x)$ of our inverse problem for different value of $N = \{8, 10, 12, 14\}$, equivalently $n = \{256, 1024, 4096, 16384\}$, on top of the synthesized target function $m(x)$. Here, the noise variance is taken as $\sigma = 0.01 \max_j \{u(x_j)\}$. The results show again that Besov prior/regularization is discretization-invariant (since the solution already converges at $n = 4096$) and edge-preserving. Table 11 presents presents the CPU time taken (in seconds), the number of Newton iterations, and the number of nonzero wavelet coefficients. As can be observed, the solutions are very sparse since there are only 13 nonzero wavelet coefficients no matter what n is. That is, the sparsity is independent of the dimension of the problem. As expected, the CPU time is a linear function of n since the number of Newton iterations is essentially constant, independent of n. It should be pointed out that the number of Newton iterations is more than 100 for all n. We observe that most of the iterations are for the STIRNCG solver to move from the initial guess to the basin of attraction of a local minimum. Once it gets there, only a few iterations are needed since the convergence is quadratic.

Similar to the deconvolution problem, we take the MAP estimation on the finest resolution as the reference solution to compute the relative L^2-error. As can be observed in Table 11, the nonlinear inverse problem with Besov prior is discretization-invariant with finite element discretization for forward, adjoint, incremental forward, and incremental adjoint equations.

Since the Bregman method is straightforward for nonlinear inverse problems, we implement it for our heat conduction problem. To solve the nonlinear unconstrained optimization problem, we employ the STIRNCG method by setting the bounds equal to infinity. Again, we only need to solve it inexactly, and for that reason we choose $\varepsilon_F = 5.e-1$, $\varepsilon_X = 1.e-6$, and $\varepsilon_G = 5.e-1$. We take $imax = 2$, $\varepsilon = 10^{-3}$, and $\lambda = 25$ in order to obtain solutions with the same quality as those in Figure



(a) $n = 256$

(b) $n = 1024$

(c) $n = 4096$

(d) $n = 16384$

Figure 5. The target function $m(x)$ and its finite dimensional reconstructions with Besov prior $\left(14, B_{11}^1(\mathbb{T})\right)$ and Haar wavelets for various values of $N = \{8, 10, 12, 14\}$. Noise level is 1%, i.e., $\sigma = 0.01 \max_j \{u(x_j)\}$.

Table 11. Time taken, the number of Newton iterations (# Newtons), relative L^2-error, and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(14, B_{11}^1(\mathbb{T})\right)$ and Haar wavelets for various values of $N = \{8, 10, 12, 14\}$. Noise level is 1%, i.e., $\sigma = 0.01 \max_j \{u(x_j)\}$.

N	8	10	12	14
time (s)	119.00	390.60	1765.00	7941.42
relative L^2-error (%)	5.88	2.84	1.22	0
nnz	13	13	13	13
# Newtons	140	129	144	151

5. Comparing Tables 12 and 11 we see that the Bregman method is more time-consuming than the STIRNCG though it also essentially has dimension-independent sparsity and iterations. While STIRNCG is more than two times more efficient than Bregman for the deconvolution problem, the efficiency is less than two times for the inverse heat conduction problem. It is interesting to observe, compare the relative L^2-error in Tables 12 and 11, that both approaches have the same convergence rate for the MAP estimation.

Table 12. Time taken, the number of Newton iterations (# Newtons), relative L^2-error, and the number of nonzero wavelet coefficients (nnz) for finite dimensional reconstructions with Besov prior $\left(14, B^1_{11}\left(\mathbb{T}\right)\right)$ and Haar wavelets for various values of $N = \left\{8, 10, 12, 14\right\}$. Noise level is 1%, i.e., $\sigma = 0.01 \max_j \left\{u\left(x_j\right)\right\}$ and $\lambda = 25$.

N	8	10	12	14
time (s)	82.12	477.22	2071.72	12415.64
relative L^2-error (%)	5.88	2.86	1.26	0.34
nnz	11	13	13	13
# Newtons	42	106	41	42

4. **Conclusions.** We have considered the problem of finding finite dimensional MAP estimates for the Bayesian deconvolution problem with Besov prior $B^1_{11}\left(\mathbb{T}\right)$. The task at hand is to solve a non-differentiable optimization problem. Table 13 summarizes the performance of three numerical optimization methods employed in the paper. We have solved the problem directly using a split Bregman method which requires to solve *imax* times an unconstrained sub-optimization problem of size n, the number of wavelet coefficients, in each Bregman iteration. We have shown that the sub-optimization can be solved inexactly and *imax* needs to be at least two in order to compromise the accuracy and efficiency. The numerical results show that the number of Bregman iterations is essentially independent of n and the computation time scales linearly in n. Alternatively, we first cast the non-differentiable optimization problem into an equivalent but differentiable one. We then use the interior path following primal dual approach to solve the resulting problem of size $3n$. We have shown that the inexact CG method for the Mehrotra predictor-corrector is more efficient than the direct solver. Similar to the Bregman approach, the interior point method seems to have linear computation time with respect to n and dimension-independent number of iterations. Nevertheless, the former is about three times more efficient than the latter. Both methods yield sparse estimates of the MAP point after thresholding, and the sparsity is independent of the dimension n of the wavelet coefficients.

We have presented another alternative in solving the equivalent differentiable optimization problem, namely, the STIRNCG approach. We have shown that the STIRNCG method seems to have most economic computation time. It provides the best result that one can hope for; the number of Newton iterations is small and independent of n, and the overall computation time scales linearly in n. Numerical results show that it is more than two times faster than the iterative Bregman method (and hence more than six times faster than the iterative interior point approach). Unlike the other methods, STIRNCG does not require thresholding to obtain sparse estimations of the MAP point. Within the STIRNCG approach, we have also showed that the ε-trick is not an efficient method. Indeed, it is even more time-consuming than the Bregman and the interior point methods.

We have also applied the STIRNCG approach to elliptic PDE-constrained inverse problem, namely, the inverse heat conduction problem up to $n = 16,384$ wavelet coefficients. The numerical results again show that solving for the MAP point using STIRNCG method has dimension-independent number of Newton iterations and sparsity; hence having linear computation time with respect to the number

Table 13. Summary of the performance of the split Bregman method, the interior path following primal dual approach, and the STIRNCG method. In the table are CPU time, relative L^2-error, the number of nonzero wavelet coefficients (nnz), and the number of iterations (Bregman iterations, or Mehrotra predictor-corrector iterations, or Newton iterations). The MAP are estimated with Besov prior $\left(20, B_{11}^{1}\left(\mathbb{T}\right)\right)$ and Haar wavelets for $N = 13$ and $\sigma = 0.05 \max_j \left\{u\left(x_j\right)\right\}$.

	CPU time	error (%)	nnz	# iterations
split Bregman (direct)	117.14	2.92	31	53
split Bregman (iterative)	70.48	2.90	32	53
Interior (direct)	4989.63	2.62	22	12
Interior (iterative)	217.90	7.95	30	27
ε-trick with STIRNCG	192.00	2.62	177	43
STIRNCG	15.46	2.61	25	9

of wavelet coefficients. Similar to the deconvolution problem, we show that the Bregman method is more time-consuming than the STIRNCG approach though it may be straightforward to implement and has half number of optimization variables.

REFERENCES

[1] M. Benzi, G. H. Golub and J. Liesen, Numerical solution of saddle point problems, *Acta Numerica*, **14** (2005), 1–137.

[2] L. Bergamaschi, J. Gondzio and G. Zilli, Preconditioning indefinite systems in interior point methods for optimization, *Computational Optimization and Applications*, **28** (2004), 149–171.

[3] M. A. Branch, T. F. Coleman and Y. Li, A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems, *SIAM Journal on Scientific Computing,* **21** (1999), 1–23 (electronic).

[4] T. Bui-Thanh, *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*, PhD thesis, Department of Aeronautics and Astronautics, MIT, 2007.

[5] T. Bui-Thanh and O. Ghattas, Analysis of the Hessian for inverse scattering problems. Part I: Inverse shape scattering of acoustic waves, *Inverse Problems,* **28** (2012), 055001, 32pp.

[6] ———, Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves, *Inverse Problems,* **28** (2012), p. 055002.

[7] ———, Analysis of the Hessian for inverse scattering problems. Part III: Inverse medium scattering of electromagnetic waves. Submitted to Inverse Problems, 2012.

[8] ———, A scaled stochastic Newton algorithm for Markov chain Monte Carlo simulations, Submitted to SIAM Journal of Uncertainty Quantification, 2012.

[9] T. Bui-Thanh, K. Willcox and O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, *SIAM Journal on Scientific Computing*, **30** (2008), 3270–3288.

[10] T. F. Coleman and Y. Li, An interior trust region approach for nonlinear minimization subject to bounds, *SIAM Journal on Optimization,* **6** (1996), 418–445.

[11] S. Comelli, *A Novel Class of Priors for Edge-Preserving Methods in Bayesian Inversion*, master's thesis, Universita Degli Studi Di Milano, 2011.

[12] M. Dashti, S. Harris and A. Stuart, Besov priors for Bayesian inverse problems, *Inverse Problems and Imaging,* **6** (2012), 183–200.

[13] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, 61. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[14] I. Daubechies, M. Defrise and C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Communications on Pure and Applied Mathematics,* **57** (2004), 1413–1457.

[15] J. E. Dennis and L. N. Vicente, Trust-region interior-point algorithms for minimization methods with simple bounds, in *Applied Mathematics and Parallel Computing,* Festschrift for Klaus Ritter, H. Fischer, B. Riedmüller, and S. Schäffler, eds., Heidelberg, (1996), Physica-Verlag, pp. 97–107.

[16] M. A. T. Figueiredo, R. D. Nowak and S. J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE Journal of Selected Topics in Signal Processing,* **1** (2007), 586–597.

[17] J. N. Franklin, Well-posed stochastic extensions of ill–posed linear problems, *Journal of Mathematical Analysis and Applications,* **31** (1970), 682–716.

[18] T. Goldstein and S. Osher, The split Bregman method for L1-regularized problems, *SIAM Journal on Imaging Sciences,* **2** (2009), 323–343.

[19] A. Grasmair, M. Haltmeier and O. Scherzer, Sparse regularization with l^q penalty term, *Inverse Problems,* **24** (2008), 055020, 13pp.

[20] K. Hamalainen, A. Kallonen, V. Kolehmainen, M. Lassas, K. Niinimaki and S. Siltanen, Sparse tomography, *SIAM J. Sci. Comput.,* **35** (2013), B644–B665.

[21] M. Heinkenschloss, M. Ulbrich and S. Ulbrich, Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption, *Mathematical Programming,* **86** (1999), 615–635.

[22] C. Kanzow and A. Klug, On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints, *Computational Optimization and Applications,* **35** (2006), 177–197.

[23] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, An interior-point method for large-scale ℓ_1-regularized least squares, *IEEE Journal of Selected Topics in Signal Processing,* **1** (2007), 606–617.

[25] V. Kolehmainen, M. Lassas, K. Niinimaki and S. Siltanen, Sparsity-promoting Bayesian inversion, *Inverse Problems,* **28** (2012), 025005, 28pp.

[26] S. Lasanen, Discretizations of generalized random variables with applications to inverse problems, *Ann. Acad. Sci. Fenn. Math. Diss.,* **2002** (2002), 64 pp.

[27] M. Lassas, E. Saksman and S. Siltanen, Discretization invariant Bayesian inversion and Besov space priors, *Inverse Problems and Imaging,* **3** (2009), 87–122.

[28] M. S. Lehtinen, L. Päivärinta and E. Somersalo, Linear inverse problems for generalized random variables, *Inverse Problems,* **5** (1989), 599–612.

[29] C.-J. Lin and J. J. Moré, Newton's method for large bound–constrained optimization problems, *SIAM Journal on Optimization,* **9** (1999), 1100–1127.

[30] D. A. Lorenz and D. Trede, Optimal convergence rates for Tikhonov regularization in Besov scales, *Inverse Problems,* **24** (2008), 055010, 14pp.

[31] M. Lustig, D. Donoho and J. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, *Journal of Magnetic Resonance Imaging,* **58** (2007), 1182–1195.

[32] S. Mehrotra, On the implementation of a primal-dual interior point method, *SIAM Journal on Optimization,* **2** (1992), 575–601.

[33] P. Piiroinen, *Statistical Measurements, Experiments, and Applications*, PhD thesis, Department of Mathematics and Statistics, University of Helsinki, 2005.

[34] D. F. Shanno and R. J. Vanderbei, An interior-point method for nonconvex nonlinear programming, *Computational Optimization and Applications,* **13** (1999), 231–252.

[35] A. M. Stuart, Inverse problems: A Bayesian perspective, *Acta Numerica*, **19** (2010), 451–559.

[36] H. Triebel, *Theory of Function Spaces III*, vol. 100, Birhäuser Verlag, 2006.

[37] J. Trzasko, A. Manduca and E. Borisch, Sparse MRI reconstruction via multiscale L0-continuation, in *Proceedings of the 14th IEEE/SP Workshop o Satistical Signal Processing,* (2007), 176–180.

[38] B. Vexler, Adaptive finite element methods for parameter identification problems, *Contributions in Mathematical and Computational Sciences*, **4** (2013), 31–54.

[39] S. J. Wright, *Primal–Dual Interior–Point Methods*, SIAM, Philadelphia, PA, 1997.

[40] C. Zhu, R. H. Byrd, P. Lu and J. Nocedal, L–bfgs–b – fortran subroutines for large–scale bound constrained optimization, *ACM Transactions on Mathematical Software,* **23** (1997), 550–560.

Received November 2012; revised September 2013.

E-mail address: tanbui@ices.utexas.edu
E-mail address: omar@ices.utexas.edu