SIAM J. OPTIM.
Vol. 35, No. 3, pp. 1439–1468

PROJECTED NEWTON METHOD FOR LARGE-SCALE BAYESIAN LINEAR INVERSE PROBLEMS[*]

HAIBO LI[†]

Abstract. Computing the regularized solution of Bayesian linear inverse problems as well as the corresponding regularization parameter is highly desirable in many applications. This paper proposes a novel iterative method, termed the *Projected Newton method* (PNT), that can simultaneously update the regularization parameter and solution step by step without requiring any expensive matrix inversions or decompositions. By reformulating the Tikhonov regularization as a constrained minimization problem and leveraging its Lagrangian function, a Newton-type method coupled with a Krylov subspace method is designed for the unconstrained Lagrangian function. The resulting PNT algorithm only needs solving a small-scale linear system to get a descent direction of a merit function at each iteration, thus significantly reducing computational overhead. Rigorous convergence results are proved, showing that PNT always converges to the unique regularized solution and the corresponding Lagrangian multiplier. Experimental results on both small-scale and large-scale Bayesian inverse problems demonstrate its excellent convergence property, robustness, and efficiency. Given that the most demanding computational tasks in PNT are primarily matrix-vector products, it is particularly well-suited for large-scale problems.

Key words. Bayesian inverse problem, Tikhonov regularization, constrained optimization, Newton method, generalized Golub–Kahan bidiagonalization, projected Newton direction

MSC codes. 65J22, 65J20, 65K10, 90C06

DOI. 10.1137/24M1645838

1. Introduction.

Inverse problems arise in various scientific and engineering fields, where the aim is to recover unknown parameters or functions from noisy observed data. Applications include image reconstruction, computed tomography, medical imaging, geoscience, data assimilation and so on [6, 25, 28, 30, 49]. A linear inverse problem of the discrete form can be written as

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the underlying quantity to reconstruct, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is the discretized forward model matrix, and $\boldsymbol{b} \in \mathbb{R}^m$ is the vector of observation with noise $\boldsymbol{\epsilon}$. We assume that the distribution of $\boldsymbol{\epsilon}$ is known, which follows a zero mean Gaussian distribution with positive definite covariance matrix $\boldsymbol{M}$, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$. A big challenge for reconstructing a good solution is the ill-posedness of inverse problems, which means that there may be multiple solutions that fit the observation equally well, or the solution is very sensitive with respect to observation perturbation.

To overcome the ill-posedness, regularization is a commonly used technique. From a Bayesian perspective [28, 53], this corresponds to adding a prior distribution of the desired solution to constrain the set of possible solutions to improve stability and uniqueness. By treating $\boldsymbol{x}$ and $\boldsymbol{b}$ as random variables, the observation vector $\boldsymbol{b}$ has a conditional probability density function (pdf) of the form $p(\boldsymbol{b}|\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2_{\boldsymbol{M}^{-1}}\right)$. To get a regularized solution, this paper considers

[*] Received by the editors March 12, 2024; accepted for publication (in revised form) April 4, 2025; published electronically July 3, 2025.

https://doi.org/10.1137/24M1645838

[†] School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC, 3010 Australia (haibo.li@unimelb.edu.au).

a Gaussian prior about the desired solution with the form $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \mu^{-1}\boldsymbol{N})$, where $\boldsymbol{N}$ is a positive definite covariance matrix. Then the Bayes' formula leads to

$$p(\boldsymbol{x}|\boldsymbol{b}, \lambda) \propto p(\boldsymbol{x}|\lambda)p(\boldsymbol{b}|\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \frac{\mu}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2\right),$$

where $\|\boldsymbol{x}\|_{\boldsymbol{B}} := (\boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x})^{1/2}$ is the $\boldsymbol{B}$-norm of $\boldsymbol{x}$ for a positive definite matrix $\boldsymbol{B}$. Maximizing the posterior pdf $p(\boldsymbol{x}|\boldsymbol{b}, \lambda)$ leads to the Tikhonov regularization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 + \mu\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2\}, \tag{1.2}$$

where the regularization term $\mu\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2$ enforces extra structure on the solution that comes from the prior distribution of $\boldsymbol{x}$.

The parameter μ in the Gaussian prior $\mathcal{N}(\boldsymbol{0}, \mu^{-1}\boldsymbol{N})$ is crucial for obtaining a good regularized solution, which controls the trade-off between the data-fit term and regularization term. There is tremendous effort in determining a proper value of μ. For the standard 2-norm problem, i.e., $\boldsymbol{M} = \boldsymbol{I}$ and $\boldsymbol{N} = \boldsymbol{I}$, the classical parameter-selection methods include the L-curve criterion [23], generalized cross-validation [22], unbiased predictive risk estimation [47], and discrepancy principle [40]. There are also some iterative methods based on solving a nonlinear equation of μ; see, e.g., [2, 20, 37, 46]. However, the aforementioned methods cannot be directly applied to (1.2). A common procedure needs to first transform (1.2) into standard 2-norm form

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\|\boldsymbol{L}_M(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})\|_2^2 + \mu\|\boldsymbol{L}_N\boldsymbol{x}\|_2^2\}, \tag{1.3}$$

where $\boldsymbol{M}^{-1} = \boldsymbol{L}_M^\top \boldsymbol{L}_M$ and $\boldsymbol{N}^{-1} = \boldsymbol{L}_N^\top \boldsymbol{L}_N$ are the Cholesky factorizations, and then apply the parameter-selection methods. This procedure needs the matrix inversions of $\boldsymbol{M}$ and $\boldsymbol{N}$ as well as the Cholesky factorizations of $\boldsymbol{M}^{-1}$ and $\boldsymbol{N}^{-1}$. For large-scale matrices, these two types of computations are almost impossible or extremely expensive.

For large-scale problems, there exist some iterative regularization methods that can avoid choosing μ in advance. A class of commonly used iterative methods is based on Krylov subspace [35], where the original linear system is projected onto lower-dimensional subspaces to become a series of small-scale problems [19, 27, 33, 42]. For dealing with the general-form Tikhonov regularization term $\|\boldsymbol{L}_N\boldsymbol{x}\|_2^2$, some recent Krylov iterative methods include [26, 29, 34, 39, 45] and so on. When the Cholesky factor $\boldsymbol{L}_N$ is not accessible, a key difficulty is dealing with the prior covariance $\boldsymbol{N}$, which means that the subspaces should be constructed elaborately such that the prior information of $\boldsymbol{x}$ can be effectively incorporated into these subspaces [8, 32]. Such methods have been proposed in [7, 8, 9], where a statistically inspired priorconditioning technique is used to whiten the noise and the desired solution. However, these methods still require large-scale matrix inversions and Cholesky factorizations, which prohibits their applications to large-scale problems.

Recently, there are several Krylov methods for directly solving (1.1) without choosing μ in advance and can avoid the matrix inversions and Cholesky factorizations [12, 32]. These methods use the generalized Golub–Kahan bidiagonalization (gen-GKB), which can iteratively reduce the original large-scale problem to small-scale ones and generate Krylov subspaces that effectively incorporate the prior information of $\boldsymbol{x}$ encoded by $\boldsymbol{N}$. In [32], the regularization effect of the proposed method comes from early stopping the iteration, where the iteration number plays the role of the regularization parameter, while in [12], the authors proposed a hybrid regularization method that simultaneously computes the regularized parameter and solution step by step. Although these two methods are very efficient for large-scale problems, there may be some issues in certain situations. The method in [32] only computes a good regularized solution but not a good μ. However, in some applications, we need an accurate estimate of μ to get the posterior distribution of $\boldsymbol{x}$ for sampling and uncertainty quantification [17, 52, 53]. For the hybrid method in [12], the convergence property does not have a solid theoretical foundation, and it has been numerically found that the method sometimes does not converge to a good solution, which is a common potential flaw for hybrid methods [11, 48].

Many optimization methods have been proposed for inverse problems, particularly those stemming from image processing that leads to total variation regularization or ℓ_p regularization. These methods include the Bregman iteration [21, 43, 56], iterative shrinkage thresholding [3, 15], and many others [1, 36, 54]. However, these methods either need a good parameter μ in advance or cannot well deal with $\boldsymbol{M}^{-1}$ and $\boldsymbol{N}^{-1}$. In [31] the author proposed a modification of the Newton method that can iteratively compute a good μ and regularized solution simultaneously. However, this method needs to solve a large-scale linear system at each iteration, which is very costly for large-scale problems. This method was improved in [13, 14], where the Newton method is successfully combined with a Krylov subspace method to get a so-called projected Newton method. Compared with the original method, the projected Newton method only needs to solve a small-scale linear system at each iteration, thereby very efficient for large-scale Tikhonov regularization (1.3). However, for solving (1.2), this method needs to compute $\nabla(\frac{1}{2}\|\boldsymbol{x}\|^2_{\boldsymbol{N}^{-1}}) = \boldsymbol{N}^{-1}\boldsymbol{x}$ to construct subspaces, which is also very costly. Besides, their methods lack rigorous proof of convergence.

In this paper, we develop a new efficient iterative method for (1.2) that simultaneously updates the regularization parameter and solution step by step, and it does not require any expensive matrix inversions or Cholesky factorizations. This method follows the Newton-type approach for noise constrained Tikhonov regularization proposed in [13], where the gen-GKB process is integrated to compute a projected Newton direction by solving a small-scale linear system at each iteration, thereby it is also named the *projected Newton method* (PNT). The main contributions of this paper are listed as follows:

- We reformulate the regularization of the original Bayesian linear inverse problem as a noise constrained minimization problem and prove the existence, uniqueness, and positivity of its Lagrangian multiplier λ under a very reasonable assumption. The correspondence between the constrained minimization problem and Tikhonov regularization (1.2) is connected by $\mu = 1/\lambda$.
- We propose a gen-GKB based Newton-type method to compute the regularized solution by optimizing its Lagrangian function and obtaining the corresponding Lagrangian multiplier. A series of Krylov subspaces is generated by gen-GKB, avoiding the need for costly matrix inversions or Cholesky factorizations. Using the subspace projection technique, we need only solve a small-scale linear system to compute the descent direction at each iteration.
- A rigorous proof of convergence for the proposed method is provided. With a very practical initialization $(\boldsymbol{x}_0, \lambda_0)$, we prove that PNT always converges to the unique solution of the constrained minimization problem and the corresponding Lagrangian multiplier.

We use both small-scale and large-scale inverse problems to test the proposed method and compare it with other state-of-the-art methods. The experimental results demonstrate excellent convergence properties of PNT, and it is very robust and efficient for regularizing Bayesian linear inverse problems. Since the most computationally intensive operations in PNT primarily involve matrix-vector products, it is especially appropriate for large-scale problems.

This paper is organized as follows. In section 2, we formulate the noise constrained minimization problem for regularizing (1.1) and study its properties. In section 3, we propose the PNT method. In section 4, we prove the convergence of PNT. Numerical results are presented in section 5 and conclusions are provided in section 6.

2. Noise constrained minimization for Bayesian inverse problems.

In order to get a good estimate of μ in (1.2), the discrepancy principle (DP) criterion is commonly used, which depends on the variance of the noise. Based on DP, we can rewrite (1.2) as an equivalent form of noise constrained minimization problem.

2.1. Noise constrained minimization.

If $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is a white Gaussian noise, the DP criterion states that the 2-norm discrepancy between the data and predicted output $\|\boldsymbol{A}\boldsymbol{x}(\mu) - \boldsymbol{b}\|_2$ should be of the order of $\|\boldsymbol{\epsilon}\|_2 \approx \sqrt{m}\sigma$, where $\boldsymbol{x}(\mu)$ is the solution to (1.2); see [28, sect. 5.6]. If $\boldsymbol{\epsilon}$ is a general Gaussian noise, notice that (1.1) leads to $\boldsymbol{L}_M\boldsymbol{b} = \boldsymbol{L}_M\boldsymbol{A}\boldsymbol{x} + \boldsymbol{L}_M\boldsymbol{\epsilon}$, and $\boldsymbol{L}_M\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, thereby this transformation whitens the noise. Since $\bar{\boldsymbol{\epsilon}} := \boldsymbol{L}_M\boldsymbol{\epsilon}$ is a white Gaussian noise with zero mean and covariance $\boldsymbol{I}$, it follows that

$$\mathbb{E}\left[\|\bar{\boldsymbol{\epsilon}}\|_2^2\right] = \mathbb{E}\left[\operatorname{trace}\left(\bar{\boldsymbol{\epsilon}}^\top \bar{\boldsymbol{\epsilon}}\right)\right] = \mathbb{E}\left[\operatorname{trace}\left(\bar{\boldsymbol{\epsilon}}\bar{\boldsymbol{\epsilon}}^\top\right)\right] = \operatorname{trace}\left(\mathbb{E}\left[\bar{\boldsymbol{\epsilon}}\bar{\boldsymbol{\epsilon}}^\top\right]\right) = \operatorname{trace}(\boldsymbol{I}) = m.$$

Therefore, the DP for (1.1) can be written as

$$\|\boldsymbol{A}\boldsymbol{x}(\mu) - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \|\boldsymbol{L}_M\boldsymbol{A}\boldsymbol{x}(\mu) - \boldsymbol{L}_M\boldsymbol{b}\|_2^2 = \tau m, \tag{2.1}$$

where τ is chosen to be marginally greater than 1, such as $\tau = 1.01$.

Using this expression of DP, we rewrite the regularization of (1.1) as the noise constrained minimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2 \quad \text{s.t.} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 \le \frac{\tau m}{2}, \tag{2.2}$$

where its Lagrangian is

$$\mathcal{L}(\boldsymbol{x}, \lambda) = \frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2 + \frac{\lambda}{2}\left(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \tau m\right) \tag{2.3}$$

with $\lambda \ge 0$ the Lagrangian multiplier. To further investigate (2.2) and (2.3), we first state the following basic assumption, which is used throughout this paper.

Assumption 1. For all $\boldsymbol{x} \in \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \min\}$, it holds that

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 < \tau m < \|\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2. \tag{2.4}$$

The first inequality means that the naive solutions to (1.1) fit the observation very well, and it ensures the feasible set of (2.2) is nonempty. The second inequality comes from the condition $\|L_{\boldsymbol{M}}\boldsymbol{\epsilon}\|_2 < \|L_{\boldsymbol{M}}\boldsymbol{b}\|_2$, meaning that the noise does not dominate the observation, which ensures the effectiveness of the regularization. Under this assumption, the following result describes the solution to (2.2).

THEOREM 2.1. *The noise constrained minimization* (2.2) *has a unique solution* $\boldsymbol{x}^*$ *satisfying* $\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \tau m$. *Furthermore, there is a unique* $\lambda^* > 0$*, which is the Lagrangian multiplier corresponding to* $\boldsymbol{x}^*$ *in* (2.3).

Proof. Let $\varphi(\boldsymbol{x}) := \frac{1}{2}(\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \tau m)$, which is a convex function. In (2.2) we seek solutions to $\min \frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2$ in the feasible set $S := \{\boldsymbol{x} \in \mathbb{R}^n : \varphi(\boldsymbol{x}) \leq 0\}$, which is the 0-lower level set of $\varphi(\boldsymbol{x})$. Note that S is a compact and convex set and $\frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2$ is continuous and strictly convex. Thus, there is a unique solution $\boldsymbol{x}^*$ to (2.2). Suppose λ^* is a Lagrangian multiplier corresponding to $\boldsymbol{x}^*$. By the Karush–Kuhn–Tucker (KKT) condition [41, sect. 12.3], the solution $(\boldsymbol{x}^*, \lambda^*)$ satisfies

$$\begin{cases} \boldsymbol{N}^{-1}\boldsymbol{x}^* + \lambda^*\nabla\varphi(\boldsymbol{x}^*) = \boldsymbol{0}, \\ \lambda^*\varphi(\boldsymbol{x}^*) = 0, \\ \lambda^* \geq 0. \end{cases}$$

If $\lambda^* = 0$, then $\boldsymbol{N}^{-1}\boldsymbol{x}^*$, leading to $\boldsymbol{x}^* = \boldsymbol{0}$. This means $\boldsymbol{0} \in S$, i.e. $\|\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 \leq \tau m$, a contradiction. Consequently, it must hold $\lambda^* > 0$. From the relation $\lambda^*\varphi(\boldsymbol{x}^*) = 0$ we have $\varphi(\boldsymbol{x}^*) = 0$, i.e. $\|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \tau m$.

For the uniqueness of λ^*, here we give two proofs. In the first proof, we note that $\nabla\varphi(\boldsymbol{x}^*) = \boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}) \neq \boldsymbol{0}$ since $\boldsymbol{x}^* \notin \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \min\}$ by Assumption 1. Therefore, the linear independence constraint qualification (LICQ) holds at $\boldsymbol{x}^*$, which leads to the uniqueness of λ^*; see [41, sect. 12.3].

In the second proof, we note that for any $\lambda \geq 0$, there is a unique $\boldsymbol{x}_\lambda$ that solves the first equality of the KKT condition:

$$\boldsymbol{N}^{-1}\boldsymbol{x} + \lambda\nabla\varphi(\boldsymbol{x}) = \boldsymbol{0} \quad \Leftrightarrow \quad (\boldsymbol{N}^{-1} + \lambda\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A})x = \lambda\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{b}, \tag{2.5}$$

since $\boldsymbol{N}^{-1} + \lambda\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A}$ is positive definite. Here we prove a stronger property: *there exist a unique* $\lambda \geq 0$ *such that* $\|\boldsymbol{A}\boldsymbol{x}_\lambda - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \tau m$. The existence of such a λ has been proved, since $\boldsymbol{x}^* = \boldsymbol{x}_{\lambda^*}$. For the uniqueness, define two functions

$$K(\lambda) := \frac{1}{2}\|\boldsymbol{x}_\lambda\|_{\boldsymbol{N}^{-1}}^2, \quad H(\lambda) := \frac{1}{2}\left(\|\boldsymbol{A}\boldsymbol{x}_\lambda - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \tau m\right).$$

Note that $\mathcal{L}(\boldsymbol{x}, \lambda)$ is strictly convex for a fixed $\lambda > 0$, which has the unique minimizer $\boldsymbol{x}_\lambda$. Thus, for any two positive $\lambda_1 \neq \lambda_2$, we have $\mathcal{L}(\boldsymbol{x}_{\lambda_1}, \lambda_1) < \mathcal{L}(\boldsymbol{x}_{\lambda_2}, \lambda_1) \Leftrightarrow K(\lambda_1) + \lambda_1 H(\lambda_1) < K(\lambda_2) + \lambda_1 H(\lambda_2)$, since $\boldsymbol{x}_{\lambda_1} \neq \boldsymbol{x}_{\lambda_2}$; see Lemma 2.2. Similarly, we have $K(\lambda_2) + \lambda_2 H(\lambda_2) < K(\lambda_1) + \lambda_2 H(\lambda_1)$. Adding the above two inequalities leads to $(\lambda_1 - \lambda_2)(H(\lambda_1) - H(\lambda_2)) < 0$, meaning that $H(\lambda)$ is a strictly monotonic decreasing function. Therefore, there is a unique λ such that $H(\lambda) = 0$. □

We emphasize that Assumption 1 is essential for ensuring the validity of Theorem 2.1 and plays a key role in the regularization of (1.1). If the left inequality of Assumption 1 is violated, then either the feasible set of (2.2) is empty when $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 > \tau m$ or it becomes the equivalent least squares problem

$$\min \frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2 \quad \text{s.t.} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \min$$

when $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \tau m$. The latter case means that no regularization is required, which makes the problem much easier to handle. The right inequality of Assumption 1 ensures the positivity of the Lagrangian multiplier λ^*, which is a necessary condition. To see it, let us assume there exist a $\lambda^* > 0$ and follow the second proof for the uniqueness of λ^*. From the KKT condition it must hold that $\varphi(\boldsymbol{x}_{\lambda^*}) = H(\lambda^*) = 0$. Now $H(0) = \frac{1}{2}(\|\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \tau m) \leq 0$ and $H(+\infty) = \frac{1}{2}(\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{M^{-1}}^2 - \tau m) < 0$ for any $\boldsymbol{x} \in \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}$. Since $H(\lambda)$ is strictly monotonically decreasing, the only possible zero root of $H(\lambda)$ is $\lambda = 0$, which leads to a contradiction. In this case, it implies that the noise in b is too large, resulting in $\lambda^* = 0$ and a very poor regularized solution $\boldsymbol{x}^* = \boldsymbol{0}$.

Lemma 2.2. *For each $\lambda \geq 0$, the regularization problem*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\lambda\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 + \|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}}^2\} \tag{2.6}$$

has the unique solution $\boldsymbol{x}_\lambda$. If $\lambda_1 \neq \lambda_2$, then $\boldsymbol{x}_{\lambda_1} \neq \boldsymbol{x}_{\lambda_2}$.

Proof. Note that the normal equation of (2.6) is equivalent to (2.5). Thus, $\boldsymbol{x}_\lambda$ is the unique solution to (2.6). Using the Cholesky factors of $\boldsymbol{M}^{-1}$ and $\boldsymbol{N}^{-1}$, and noticing that $\boldsymbol{L}_N$ is invertible, we can write the generalized singular value decomposition (GSVD) [55] of $\{\boldsymbol{L}_M\boldsymbol{A}, \boldsymbol{L}_N\}$ as $\boldsymbol{L}_M\boldsymbol{A} = \boldsymbol{U}_A\boldsymbol{\Sigma}_A\boldsymbol{Z}^{-1}$, $\boldsymbol{L}_N = \boldsymbol{U}_N\boldsymbol{\Sigma}_N\boldsymbol{Z}^{-1}$ with

$$\boldsymbol{\Sigma}_A = \begin{array}{c} \left(\begin{array}{cc} \boldsymbol{D}_A & \\ & \boldsymbol{0} \end{array}\right) \begin{array}{c} r \\ m-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} \end{array}, \quad \boldsymbol{\Sigma}_N = \begin{array}{c} \left(\begin{array}{cc} \boldsymbol{D}_N & \\ & \boldsymbol{I} \end{array}\right) \begin{array}{c} r \\ n-r \end{array} \\ \begin{array}{cc} r & n-r \end{array} \end{array},$$

where $\boldsymbol{U}_A \in \mathbb{R}^{m\times m}$ and $\boldsymbol{U}_N \in \mathbb{R}^{n\times n}$ are orthogonal, $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$ in nonsingular, $r = \operatorname{rank}(\boldsymbol{A})$, and $\boldsymbol{D}_A = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ with $1 > \sigma_1 \geq \cdots \geq \sigma_r > 0$ and $\boldsymbol{D}_N = \operatorname{diag}(\rho_1, \ldots, \rho_r)$ with $0 < \rho_1 \leq \cdots \leq \rho_r < 1$, such that $\sigma_i^2 + \rho_i^2 = 1$. Then $\boldsymbol{x}_\lambda$ can be expressed as $\boldsymbol{x}_\lambda = \sum_{i=1}^r \frac{\lambda\sigma_i}{\lambda\sigma_i^2+\rho_i^2}(\boldsymbol{u}_{A,i}^\top \boldsymbol{L}_M\boldsymbol{b})\boldsymbol{z}_i$ where $\boldsymbol{u}_{A,i}$ is the ith column of $\boldsymbol{U}_A$. Since $\{\boldsymbol{z}_i\}_{i=1}^r$ are linear independent, if $\boldsymbol{x}_{\lambda_1} = \boldsymbol{x}_{\lambda_2}$, then it must hold that

$$\frac{\lambda_1\sigma_i}{\lambda_1\sigma_i^2+\rho_i^2} = \frac{\lambda_2\sigma_i}{\lambda_2\sigma_i^2+\rho_i^2} \quad \Leftrightarrow \quad (\lambda_1-\lambda_2)\sigma_i\rho_i^2 = 0, \quad i = 1, \ldots, r.$$

Since $\sigma_i\rho_i > 0$ for $i = 1, \ldots, r$, we obtain $\lambda_1 = \lambda_2$. □

Remark 1. From the proof of Theorem 2.1, we find that λ plays the role of μ^{-1} in (1.2), meaning that $\boldsymbol{x}(\mu) = \boldsymbol{x}_\lambda$ if $\lambda = \mu^{-1}$. In fact, there is a one-to-one correspondence between (1.2) and (2.2). Note that $\boldsymbol{x}^* = \boldsymbol{x}_{\lambda^*}$. Comparing (2.6) with (1.2), we can use $(\lambda^*)^{-1}$ as a good estimate of the optimal regularization parameter.

Corollary 2.3. *Let $\mathbb{R}^+ = [0, \infty)$. Write the gradient of $\mathcal{L}(\boldsymbol{x}, \lambda)$ as*

$$F(\boldsymbol{x}, \lambda) = \begin{pmatrix} \lambda\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x} \\ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \frac{\tau m}{2} \end{pmatrix}. \tag{2.7}$$

Then $F(\boldsymbol{x}, \lambda) = \boldsymbol{0}$ has a unique solution $(\boldsymbol{x}^*, \lambda^*)$ in $\mathbb{R}^n \times \mathbb{R}^+$, which is the unique minimizer and corresponding Lagrangian multiplier of (2.2).

2.2. Newton method.

A modification of the Newton method was proposed in [31] to solve the nonlinear equation $F(\boldsymbol{x}, \lambda) = \boldsymbol{0}$, which is referred to as the Lagrange method since it is based on the Lagrangian of (2.2). In this method, the Jacobian matrix of $F(\boldsymbol{x}, \lambda)$ is first computed as

$$J(\boldsymbol{x}, \lambda) = \begin{pmatrix} \lambda\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A} + \boldsymbol{N}^{-1} & \boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) \\ (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^\top\boldsymbol{M}^{-1}\boldsymbol{A} & 0 \end{pmatrix} \tag{2.8}$$

at the current iterate $(\boldsymbol{x}, \lambda)$, and then it computes the Newton direction $(\Delta\boldsymbol{x}^\top, \Delta\lambda)^\top$ by solving inexactly the linear system

$$J(\boldsymbol{x}, \lambda)\begin{pmatrix} \Delta\boldsymbol{x} \\ \Delta\lambda \end{pmatrix} = -F(\boldsymbol{x}, \lambda) \tag{2.9}$$

using the MINRES solver [44]. We remark that this method is essentially a Newton–Krylov method [5] for optimizing the nonlinear and nonconvex Lagrangian function

(2.3). It was shown that the computed $(\Delta \boldsymbol{x}, \Delta\lambda)$ is a descent direction for the merit function

$$h_w(\boldsymbol{x}, \lambda) = \frac{1}{2}\left(\|\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, \lambda)\|_2^2 + w|\nabla_\lambda \mathcal{L}(\boldsymbol{x}, \lambda)|^2\right)$$

with a $w > 0$, which means that $(\Delta \boldsymbol{x}^\top, \Delta\lambda)\nabla h_w(\boldsymbol{x}, \lambda) \leq 0$. By a backtracking line search strategy to determine a step length $\gamma > 0$, the iterate is updated as $(\boldsymbol{x}, \lambda) \leftarrow (\boldsymbol{x}, \lambda) + \gamma(\Delta \boldsymbol{x}, \Delta\lambda)$.

An advantage of this method is that it can compute a good regularized solution and its regularization parameter simultaneously. However, for large-scale problems, we need to compute $\boldsymbol{M}^{-1}$ and $\boldsymbol{N}^{-1}$ to form $F(\boldsymbol{x}, \lambda)$ and $J(\boldsymbol{x}, \lambda)$, which is almost impossible. Moreover, at each iteration, an $(n+1) \times (n+1)$ linear system (2.9) needs to be solved, which is very computationally expensive even if we only compute a less accurate solution by an iterative algorithm.

In [13], the authors proposed a projected Newton method, where at each iteration, the large-scale linear system (2.9) is projected to be a small-scale linear system that can be solved cheaply. However, this method can only deal with the standard $\ell_2 - \ell_2$ regularization, which means we can only apply this method to (1.3) by the substitution $\bar{\boldsymbol{x}} = \boldsymbol{L}_N \boldsymbol{x}$, requiring the expensive Cholesky factorization of $\boldsymbol{N}^{-1}$. A generalization of this method [14] can deal with a general-form regularization term. However, for (2.2), it needs to compute $\nabla(\frac{1}{2}\|x\|_{\boldsymbol{N}^{-1}}^2) = \boldsymbol{N}^{-1}x$ to construct subspace for projecting (2.9), also very costly.

3. Projected Newton method based on generalized Golub–Kahan bidiagonalization.

To reduce expensive computations of the Newton method for large-scale problems, we design a new projected Newton method to solve (2.2). This method uses the gen-GKB to construct Krylov subspaces to compute projected Newton directions by only solving small-scale problems, and it does not need any expensive matrix inversions or decompositions. This method is composed of the following three main steps:

Step 1: *Construct Krylov subspaces.* We adopt gen-GKB to iteratively construct a series of low-dimensional Krylov subspaces; see Algorithm 3.1.

Step 2: *Compute the projected Newton direction.* At each iteration, we compute the projected Newton direction by solving a small-scale problem; see (3.12).

Step 3: *Determine the step-length to update solution.* We use the Armijo backtracking line search to determine a step-length and update the solution; see Routine 1.

Algorithm 3.1. Generalized Golub–Kahan bidiagonalization (gen-GKB).

```
Input: $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, $\boldsymbol{N} \in \mathbb{R}^{n\times n}$
$\bar{\boldsymbol{s}} = \boldsymbol{M}^{-1}\boldsymbol{b}$, $\beta_1 = \bar{\boldsymbol{s}}^\top \boldsymbol{b}$, $\boldsymbol{u}_1 = \boldsymbol{b}/\beta_1$, $\bar{\boldsymbol{u}}_1 = \bar{\boldsymbol{s}}/\beta_1$
$\bar{\boldsymbol{r}} = \boldsymbol{A}^\top \bar{\boldsymbol{u}}_1$, $\boldsymbol{r} = \boldsymbol{N}\bar{\boldsymbol{r}}$
$\alpha_1 = (\boldsymbol{r}^\top \bar{\boldsymbol{r}})^{1/2}$, $\bar{\boldsymbol{v}}_1 = \bar{\boldsymbol{r}}/\alpha_1$, $\boldsymbol{v}_1 = \boldsymbol{r}/\alpha_1$
for $i = 1, 2, \dots, k$ do
    $\boldsymbol{s} = \boldsymbol{A}\boldsymbol{v}_i - \alpha_i \boldsymbol{u}_i$, $\bar{\boldsymbol{s}} = \boldsymbol{M}^{-1}\boldsymbol{s}$
    $\beta_{i+1} = (\boldsymbol{s}^\top \bar{\boldsymbol{s}})^{1/2}$, $\boldsymbol{u}_{i+1} = \boldsymbol{s}/\beta_{i+1}$, $\bar{\boldsymbol{u}}_{i+1} = \bar{\boldsymbol{s}}/\beta_{i+1}$
    $\bar{\boldsymbol{r}} = \boldsymbol{A}^\top \bar{\boldsymbol{u}}_{i+1} - \beta_{i+1}\bar{\boldsymbol{v}}_i$, $\boldsymbol{r} = \boldsymbol{N}\bar{\boldsymbol{r}}$
    $\alpha_{i+1} = (\boldsymbol{r}^\top \bar{\boldsymbol{r}})^{1/2}$, $\bar{\boldsymbol{v}}_{i+1} = \bar{\boldsymbol{r}}/\alpha_{i+1}$, $\boldsymbol{v}_{i+1} = \boldsymbol{r}/\alpha_{i+1}$
end for
Output: $\{\alpha_i, \beta_i\}_{i=1}^{k+1}$, $\{\boldsymbol{u}_i, \boldsymbol{v}_i\}_{i=1}^{k+1}$
```

In the next subsection, we present detailed derivations of the whole algorithm. All the proofs can be found in subsection 3.2.

3.1. Derivation of projected Newton method.

This subsection presents detailed derivations for the above three steps.

Step 1. *Construct Krylov subspaces by gen-GKB.* The gen-GKB process has been proposed for solving Bayesian linear inverse problems in [12, 32]. The basic idea is to treat $\boldsymbol{A}$ as the compact linear operator

$$\boldsymbol{A}:(\mathbb{R}^n,\langle\cdot,\cdot\rangle_{\boldsymbol{N}^{-1}})\rightarrow(\mathbb{R}^m,\langle\cdot,\cdot\rangle_{\boldsymbol{M}^{-1}}),\ \boldsymbol{x}\mapsto\boldsymbol{A}\boldsymbol{x},$$

where $\boldsymbol{x}$ and $\boldsymbol{A}\boldsymbol{x}$ are vectors under the canonical bases of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Here the two inner products are defined as $\langle\boldsymbol{x},\boldsymbol{x}'\rangle_{\boldsymbol{N}^{-1}}:=\boldsymbol{x}^\top\boldsymbol{N}^{-1}\boldsymbol{x}'$ and $\langle\boldsymbol{y},\boldsymbol{y}'\rangle_{\boldsymbol{M}^{-1}}:=\boldsymbol{y}^\top\boldsymbol{M}^{-1}\boldsymbol{y}'$. Therefore, we can define

$$\boldsymbol{A}^*:(\mathbb{R}^m,\langle\cdot,\cdot\rangle_{\boldsymbol{M}^{-1}})\rightarrow(\mathbb{R}^n,\langle\cdot,\cdot\rangle_{\boldsymbol{N}^{-1}}),\ \boldsymbol{y}\mapsto\boldsymbol{A}^*\boldsymbol{y},$$

which is the adjoint operator of $\boldsymbol{A}$, by the relation $\langle\boldsymbol{A}\boldsymbol{x},\boldsymbol{y}\rangle_{\boldsymbol{M}^{-1}}=\langle\boldsymbol{x},\boldsymbol{A}^*\boldsymbol{y}\rangle_{\boldsymbol{N}^{-1}}$. Note that $(\boldsymbol{A}\boldsymbol{x})^\top\boldsymbol{M}^{-1}\boldsymbol{y}=\boldsymbol{x}^\top\boldsymbol{N}^{-1}\boldsymbol{A}^*\boldsymbol{y}$ for any $\boldsymbol{x}\in\mathbb{R}^n$ and $\boldsymbol{y}\in\mathbb{R}^m$. Thus, the matrix-form expression of $\boldsymbol{A}^*$ is $\boldsymbol{A}^*=\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}$.

Applying the standard Golub–Kahan bidiagonalization (GKB) to the compact operator $\boldsymbol{A}$ with starting vector $\boldsymbol{b}$ between the two Hilbert spaces $(\mathbb{R}^n,\langle\cdot,\cdot\rangle_{\boldsymbol{N}^{-1}})$ and $(\mathbb{R}^m,\langle\cdot,\cdot\rangle_{\boldsymbol{M}^{-1}})$, we can obtain the gen-GKB process; see [10] for GKB for compact operators. The basic recursive relations of gen-GKB are as follows:

$$\beta_1\boldsymbol{u}_1=\boldsymbol{b}, \tag{3.1a}$$

$$\alpha_i\boldsymbol{v}_i=\boldsymbol{A}^*\boldsymbol{u}_i-\beta_i\boldsymbol{v}_{i-1}, \tag{3.1b}$$

$$\beta_{i+1}\boldsymbol{u}_{i+1}=\boldsymbol{A}\boldsymbol{v}_i-\alpha_i\boldsymbol{u}_i, \tag{3.1c}$$

where α_i and β_i are computed such that $\|\boldsymbol{u}_i\|_{\boldsymbol{M}^{-1}}=\|\boldsymbol{v}_i\|_{\boldsymbol{N}^{-1}}=1$, and $\boldsymbol{v}_0:=\boldsymbol{0}$. The whole iterative process is summarized in Algorithm 3.1. For more details of the derivation, please see [32].

We remark that computing with $\boldsymbol{M}^{-1}$ cannot be avoided, but for the most commonly encountered cases that $\boldsymbol{\epsilon}$ is a Gaussian noise with uncorrelated components, $\boldsymbol{M}$ is diagonal and thereby $\boldsymbol{M}^{-1}$ can be directly obtained. For applications that $\boldsymbol{\epsilon}$ is a colored Gaussian noise such that $\boldsymbol{M}$ is not diagonal, computing $\boldsymbol{M}$ is the most expensive operation. In these cases, the proposed PNT method based on gen-GKB may not be the optimal choice.

The following result gives the basic property of gen-GKB; see [32] for the proof.

Proposition 3.1. *The group of vectors* $\{\boldsymbol{u}_i\}_{i=1}^k$ *is an* $\boldsymbol{M}^{-1}$*-orthonormal basis of the Krylov subspace*

$$\mathcal{K}_k(\boldsymbol{A}\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1},\boldsymbol{b})=\operatorname{span}\{(\boldsymbol{A}\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1})^i\boldsymbol{b}\}_{i=0}^{k-1}, \tag{3.2}$$

and $\{\boldsymbol{v}_i\}_{i=1}^k$ *is an* $\boldsymbol{N}^{-1}$*-orthonormal basis of the Krylov subspace*

$$\mathcal{K}_k(\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A},\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{b})=\operatorname{span}\{(\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A})^i\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{b}\}_{i=0}^{k-1}. \tag{3.3}$$

Define $\boldsymbol{U}_{k+1}=(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{k+1})$ and $\boldsymbol{V}_{k+1}=(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{k+1})$. Then Proposition 3.1 indicates that $\boldsymbol{U}_{k+1}^\top\boldsymbol{M}^{-1}\boldsymbol{U}_{k+1}=\boldsymbol{I}$ and $\boldsymbol{V}_{k+1}^\top\boldsymbol{N}^{-1}\boldsymbol{V}_{k+1}=\boldsymbol{I}$. We remark that gen-GKB will eventually terminate in at most $\min\{m,n\}$ steps, since the column rank of $\boldsymbol{U}_k$ or $\boldsymbol{V}_k$ cannot exceed $\min\{m,n\}$. If we define the *termination step* as

$$k_t:=\max\{k:\alpha_k\beta_k>0\}, \tag{3.4}$$

then $\boldsymbol{V}_k$ will eventually expand to be $\boldsymbol{V}_{k_t}$ with $k_t\leq\min\{m,n\}$.

By (3.1a)–(3.1c), we can write the k-step ($k \le k_t$) gen-GKB in the matrix-form

$$\beta_1 \boldsymbol{U}_{k+1} e_1 = \boldsymbol{b}, \tag{3.5a}$$

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{U}_{k+1}\boldsymbol{B}_k, \tag{3.5b}$$

$$\boldsymbol{N}\boldsymbol{A}^\top \boldsymbol{M}^{-1}\boldsymbol{U}_{k+1} = \boldsymbol{V}_k \boldsymbol{B}_k^\top + \alpha_{k+1}\boldsymbol{v}_{k+1}e_{k+1}^\top, \tag{3.5c}$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_{k+1}$ are the first and $(k+1)$th columns of the identity matrix of order $k+1$, respectively, and

$$\boldsymbol{B}_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}. \tag{3.6}$$

Note that $\boldsymbol{B}_k$ has full column rank if $k \le k_t$. At the k_tth iteration, it is possible that either $\beta_{k_t+1} = 0$ occurs first or $\alpha_{k_t+1} = 0$ occurs first. For the former case, the relations (3.5) are replaced by

$$\beta_1 \boldsymbol{U}_{k_t} e_1 = \boldsymbol{b}, \tag{3.7a}$$

$$\boldsymbol{A}\boldsymbol{V}_{k_t} = \boldsymbol{U}_{k_t}\underline{\boldsymbol{B}}_{k_t}, \tag{3.7b}$$

$$\boldsymbol{N}\boldsymbol{A}^\top \boldsymbol{M}^{-1}\boldsymbol{U}_{k_t} = \boldsymbol{V}_{k_t}\underline{\boldsymbol{B}}_{k_t}^\top, \tag{3.7c}$$

where $\underline{\boldsymbol{B}}_{k_t}$ is the first $k \times k$ part of $\boldsymbol{B}_{k_t}$ by discarding β_{k_t+1}.

Step 2. *Compute the projected Newton direction.* At the kth iteration, we update $\boldsymbol{x}_k \in \operatorname{span}\{\boldsymbol{V}_k\}$ and λ_k from the previous ones. For any $\boldsymbol{x} \in \operatorname{span}\{\boldsymbol{V}_k\}$ of the form $\boldsymbol{x} = \boldsymbol{V}_k\boldsymbol{y}$ with $\boldsymbol{y} \in \mathbb{R}^k$, define the projected gradient of $\mathcal{L}(\boldsymbol{x}, \lambda)$ as

$$F^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \boldsymbol{V}_k^\top & \\ & 1 \end{pmatrix} F(\boldsymbol{x}, \lambda) \tag{3.8}$$

and the projected Jacobian of $F(\boldsymbol{x}, \lambda)$ as

$$J^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \boldsymbol{V}_k^\top & \\ & 1 \end{pmatrix} J(\boldsymbol{x}, \lambda) \begin{pmatrix} \boldsymbol{V}_k & \\ & 1 \end{pmatrix}. \tag{3.9}$$

Remark 2. Since gen-GKB must terminate at the k_tth iteration and $\boldsymbol{V}_k$ eventually expands to be $\boldsymbol{V}_{k_t}$, we need to discuss the two different cases that $k \le k_t$ and $k > k_t$. For notational simplicity, in the rest part of the paper, we use $\boldsymbol{V}_k$ and $\boldsymbol{B}_k$ by default unless stated otherwise to denote

$$\boldsymbol{V}_k = \begin{cases} \boldsymbol{V}_k, & k \le k_t, \\ \boldsymbol{V}_{k_t}, & k > k_t, \end{cases} \qquad \boldsymbol{B}_k = \begin{cases} \boldsymbol{B}_k, & k \le k_t, \\ \boldsymbol{B}_{k_t}, & k > k_t, \end{cases} \qquad \boldsymbol{x}_k = \begin{cases} \boldsymbol{V}_k\boldsymbol{y}_k, & k \le k_t, \\ \boldsymbol{V}_{k_t}\boldsymbol{y}_k, & k > k_t, \end{cases}$$

where $\boldsymbol{y} \in \mathbb{R}^k$ for $k \le k_t$ and $\boldsymbol{y} \in \mathbb{R}^{k_t}$ for $k > k_t$. Moreover, for the case $\beta_{k_t+1} = 0$, the relations (3.5) are replaced by (3.7) and $\boldsymbol{B}_{k_t}$ is replaced by $\underline{\boldsymbol{B}}_{k_t}$. In the subsequent discussions, we employ the unified notations as presented in (3.5), but the readers can readily differentiate between the two cases.

Notice that $\boldsymbol{y}$ is uniquely determined by $\boldsymbol{x} = \boldsymbol{V}_k\boldsymbol{y}$ since $\boldsymbol{V}_k$ has full-column rank. Thus, $F^{(k)}(\boldsymbol{y}, \lambda)$ and $J^{(k)}(\boldsymbol{y}, \lambda)$ are well-defined. The next result shows how we can obtain $F^{(k)}(\boldsymbol{y}, \lambda)$ and $J^{(k)}(\boldsymbol{y}, \lambda)$ from $\boldsymbol{B}_k$ without any additional computations.

LEMMA 3.2. *For any $\boldsymbol{x} \in \operatorname{span}\{\boldsymbol{V}_k\}$ with the form $\boldsymbol{x} = \boldsymbol{V}_k\boldsymbol{y}$, the projected gradient of $\mathcal{L}(\boldsymbol{x}, \lambda)$ has the expression*

$$F^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \lambda \boldsymbol{B}_k^\top(\boldsymbol{B}_k\boldsymbol{y} - \beta_1 e_1) + \boldsymbol{y} \\ \frac{1}{2}\|\boldsymbol{B}_k\boldsymbol{y} - \beta_1 e_1\|_2^2 - \frac{\tau m}{2} \end{pmatrix}, \tag{3.10}$$

and the projected Jacobian of $F(\boldsymbol{x}, \lambda)$ has the expression

$$J^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \lambda \boldsymbol{B}_k^\top \boldsymbol{B}_k + \boldsymbol{I} & \boldsymbol{B}_k^\top(\boldsymbol{B}_k\boldsymbol{y} - \beta_1 e_1) \\ (\boldsymbol{B}_k\boldsymbol{y} - \beta_1 e_1)^\top \boldsymbol{B}_k & 0 \end{pmatrix}. \tag{3.11}$$

Now we can compute the projected Newton direction for updating the solution. Starting from an initial solution $(\boldsymbol{x}_0, \lambda_0)$, consider the following two cases.

Case 1. *Update $(\boldsymbol{x}_k, \lambda_k)$ from $(\boldsymbol{x}_{k-1}, \lambda_{k-1})$ for $k \le k_t$.* Suppose at the $(k-1)$th iteration, we have $\boldsymbol{x}_{k-1} = \boldsymbol{V}_{k-1}\boldsymbol{y}_{k-1}$, where $\boldsymbol{x}_0 := \boldsymbol{0}$ and $\boldsymbol{y}_0 := ()$ is an empty vector. Let $\bar{\boldsymbol{y}}_{k-1} = (\boldsymbol{y}_{k-1}^\top, 0)^\top \in \mathbb{R}^k$. If $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ is nonsingular, we compute the Newton direction for the projected function $F^{(k)}(\boldsymbol{y}, \lambda)$ at $(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$:

$$\begin{pmatrix} \Delta \boldsymbol{y}_k \\ \Delta \lambda_k \end{pmatrix} = -J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})^{-1} F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1}). \tag{3.12a}$$

Then we update $(\bar{\boldsymbol{y}}_k, \lambda_k)$ by

$$\boldsymbol{y}_k = \bar{\boldsymbol{y}}_{k-1} + \gamma_k \Delta \boldsymbol{y}_k, \quad \lambda_k = \lambda_{k-1} + \gamma_k \Delta \lambda_k \tag{3.12b}$$

with a suitably chosen step-length $\gamma_k > 0$, and let $\boldsymbol{x}_k = \boldsymbol{V}_k \boldsymbol{y}_k$.

Case 2. *Update $(\boldsymbol{x}_k, \lambda_k)$ from $(\boldsymbol{x}_{k-1}, \lambda_{k-1})$ for $k > k_t$.* At each iteration, we seek a solution of the form $\boldsymbol{x}_k = \boldsymbol{V}_{k_t}\boldsymbol{y}_k$ with $\boldsymbol{y}_k \in \mathbb{R}^{k_t}$. We compute the Newton direction

$$\begin{pmatrix} \Delta \boldsymbol{y}_k \\ \Delta \lambda_k \end{pmatrix} = -J^{(k_t)}(\boldsymbol{y}_{k-1}, \lambda_{k-1})^{-1} F^{(k_t)}(\boldsymbol{y}_k, \lambda_{k-1}), \tag{3.12c}$$

and then compute

$$\boldsymbol{y}_k = \boldsymbol{y}_{k-1} + \gamma_k \Delta \boldsymbol{y}_k, \quad \lambda_k = \lambda_{k-1} + \gamma_k \Delta \lambda_k \tag{3.12d}$$

to get $\boldsymbol{x}_k = \boldsymbol{V}_{k_t}\boldsymbol{y}_k$.

For both of the two cases, we call $(\Delta \boldsymbol{y}_k, \Delta \lambda_k)$ the *projected Newton direction,* since it is the Newton direction of a projected problem. The corresponding update formula for $\boldsymbol{x}_k$ is

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \gamma_k \Delta \boldsymbol{x}_k, \quad \Delta \boldsymbol{x}_k := \boldsymbol{V}_k \Delta \boldsymbol{y}_k,$$

which is easy to be verified. For notational simplicity, in the subsequent part we always use the unified notation

$$\bar{\boldsymbol{y}}_{k-1} = \begin{cases} (\boldsymbol{y}_{k-1}^\top,\ 0)^\top, & k \le k_t, \\ \boldsymbol{y}_{k-1}, & k > k_t \end{cases} \tag{3.13}$$

for $\bar{\boldsymbol{y}}_{k-1}$. Following the notations stated in Remark 2 and (3.13), we can use (3.12a) and (3.12b) to describe the update procedure for both of the two cases.

It is vital to make sure that the projected Jacobian matrix $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ is always nonsingular. This desired property is given in the following result. The proof appears as a part of the proof of Lemma 4.6.

PROPOSITION 3.3. *If we choose $\boldsymbol{x}_0 = \boldsymbol{0}$ and $\bar{\boldsymbol{y}}_0 = 0$, then at each iteration $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ is nonsingular as long as $\lambda_{k-1} \geq 0$.*

In order to investigate the convergence behavior of the method, define the following merit function:

$$h(\boldsymbol{x}, \lambda) = \frac{1}{2}\left[\|\lambda \boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}\|_{\boldsymbol{N}}^2 + \left(\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \frac{\tau m}{2}\right)^2\right]. \tag{3.14}$$

Notice from Corollary 2.3 that $(\boldsymbol{x}^*, \lambda^*)$ is the unique minimizer of $h(\boldsymbol{x}, \lambda)$ and that $h(\boldsymbol{x}^*, \lambda^*) = 0$. The following result shows that $(\Delta\boldsymbol{x}_k^\top, \Delta\lambda_k)^\top$ is indeed a descent direction for $h(\boldsymbol{x}, \lambda)$.

THEOREM 3.4. *Let $\Delta\boldsymbol{x}_k = \boldsymbol{V}_k \Delta\boldsymbol{y}_k$. Then it holds that*

$$\nabla h(\boldsymbol{x}_{k-1}, \lambda_{k-1})^\top \begin{pmatrix} \Delta\boldsymbol{x}_k \\ \Delta\lambda_k \end{pmatrix} = -2h(\boldsymbol{x}_{k-1}, \lambda_{k-1}) \leq 0. \tag{3.15}$$

Theorem 3.4 is a desired property for a gradient descent type algorithm. At the $(k-1)$-th iteration, if $h(\boldsymbol{x}_{k-1}, \lambda_{k-1}) = 0$, then we have $(\boldsymbol{x}_{k-1}, \lambda_{k-1}) = (\boldsymbol{x}^*, \lambda^*)$, meaning we have obtained the unique solution to (2.2). Otherwise, $(\Delta\boldsymbol{x}_k^\top, \Delta\lambda_k)^\top$ is a descent direction of $h(\boldsymbol{x}, \lambda)$ at $(\boldsymbol{x}_{k-1}, \lambda_{k-1})$, thereby we can continue updating the solution by a backtracking line search strategy.

Step 3. *Determine step-length by backtracking line search.* For the case that $h(\boldsymbol{x}_{k-1}, \lambda_{k-1}) \neq 0$, we need to determine a step-length γ_k such that $h(\boldsymbol{x}_k, \lambda_k)$ decreases strictly. To this end, we use the backtracking line search procedure to ensure that the *Armijo condition* [41, sect. 3.1] is satisfied:

$$h(\boldsymbol{x}_k, \lambda_k) \leq h(\boldsymbol{x}_{k-1}, \lambda_{k-1}) + c\gamma_k(\Delta\boldsymbol{x}_{k-1}^\top, \Delta\lambda_k)\nabla h(\boldsymbol{x}_{k-1}, \lambda_{k-1}), \tag{3.16}$$

where $(\boldsymbol{x}_k, \lambda_k) = (\boldsymbol{x}_{k-1}, \lambda_{k-1}) + \gamma_k(\Delta\boldsymbol{x}_k, \Delta\lambda_k)$, and $c \in (0, 1)$ is a fixed constant. At each iteration, we can quickly compute $h(\boldsymbol{x}_k, \lambda_k)$ based on the following result.

LEMMA 3.5. *Let*

$$\bar{F}^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \lambda \bar{\boldsymbol{B}}_k^\top(\bar{\boldsymbol{B}}_k\boldsymbol{y} - \beta_1 e_1) + y \\ \frac{1}{2}\|\bar{\boldsymbol{B}}_k\boldsymbol{y} - \beta_1 e_1\|_2^2 - \frac{\tau m}{2} \end{pmatrix}, \quad \bar{\boldsymbol{B}}_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_{k+1} & \alpha_{k+1} \end{pmatrix}.$$

Then we have

$$h(\boldsymbol{x}_{k-1}, \lambda_{k-1}) = \frac{1}{2}\|F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})\|_2^2, \quad h(\boldsymbol{x}_k, \lambda_k) = \frac{1}{2}\|\bar{F}^{(k)}(\bar{\boldsymbol{y}}_k, \lambda_k)\|_2^2. \tag{3.17}$$

We remark that in the above expression we have $\bar{\boldsymbol{B}}_k = \boldsymbol{B}_{k_t}$ for $k \geq k_t$, and specifically, we have $\bar{\boldsymbol{B}}_k = \underline{\boldsymbol{B}}_{k_t}^\top$ if $\beta_{k_t+1} = 0$. The following theorem shows the existence of a suitable step-length; see, e.g. [4, p. 121, Theorem 2.1] for details.

THEOREM 3.6. *For any continuously differentiable function $f(\boldsymbol{s}) : \mathbb{R}^l \to \mathbb{R}$, suppose ∇f is Lipschitz continuous with constant $\zeta(\boldsymbol{s})$ at $\boldsymbol{s}$. If $\boldsymbol{p}$ is a descent direction at $\boldsymbol{s}$, i.e., $\nabla f(\boldsymbol{s})^\top\boldsymbol{p} < 0$, then for a fixed $c \in (0, 1)$ the Armijo condition $f(\boldsymbol{s} + \gamma\boldsymbol{p}) \leq f(\boldsymbol{s}) + c\gamma\nabla f(\boldsymbol{s})^\top\boldsymbol{p}$ is satisfied for all $\gamma \in [0, \gamma_{\max}]$ with $\gamma_{\max} = \frac{2(c-1)\nabla f(\boldsymbol{s})^\top\boldsymbol{p}}{\zeta(\boldsymbol{s})\|\boldsymbol{p}\|^2}$.*

With the aid of Lemma 3.5, a suitable step-length γ_k can be determined using the following backtracking line search strategy.

ROUTINE 1. *Armijo backtracking line search:*

1. *Given* $\gamma_{init} > 0$*, let* $\gamma^{(0)} = \gamma_{init}$ *and* $l = 0$*.*
2. *Until* $\frac{1}{2}\|\bar{F}^{(k)}(\bar{\boldsymbol{y}}_k, \lambda_k)\|_2^2 \leq \left(\frac{1}{2} - c\gamma^{(l)}\right) \|F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})\|_2^2$,
   (i) *set* $\gamma^{(l+1)} = \eta\gamma^{(l)}$*, where* $\eta \in (0,1)$ *is a fixed constant;*
   (ii) $l \leftarrow l+1$.
3. *Set* $\gamma_k = \gamma^{(l)}$*.*

We set $c = 10^{-4}$, $\gamma_{\text{init}} = 1.0$, and $\eta = 0.9$ by default. Note that at each iteration we need to ensure $\lambda_k > 0$. Suppose at the $(k-1)$th iteration we already have $\lambda_{k-1} > 0$. Then at the kth iteration, if $\Delta\lambda_k < 0$, we need only enforce $\gamma_{\text{init}} < -\lambda_{k-1}/\Delta\lambda_k$.

Overall, the whole procedure of PNT is presented in Algorithm 3.2. In the PNT algorithm, at each kth iteration, computing the projected Newton direction requires solving only the $(k+1)$-order linear system (3.12a), which can be done very quickly when $k \ll n$. Starting from the termination step k_t, at each subsequent iteration, a (k_t+1)-order linear system (3.12c) needs to be solved. We numerically find that the algorithm almost always obtains a satisfied solution before gen-GKB terminates. The PNT method is a natural generalization of the projected Newton method in [13]. Specifically, when $\boldsymbol{M} = \boldsymbol{I}$ and $\boldsymbol{N} = \boldsymbol{I}$, it can be confirmed that both methods are identical.

We remark that for very large-scale problems, it may take too many iterations for PNT to converge. In this case, we can update the solution starting from the k_0th step of gen-GKB to save some computation for solving (3.12a). This means that we first run $k_0 - 1$ steps gen-GKB to construct a (k_0-1)-dimensional subspace and then start to update the solution from the k_0th iteration. From the derivation of PNT, it can be easily verified that if we set $\bar{\boldsymbol{y}}_{k_0-1} = \boldsymbol{0} \in \mathbb{R}^{k_0}$, then $(\Delta\boldsymbol{x}_k^\top, \Delta\lambda_k)^\top$ is a descent direction of $h(\boldsymbol{x}, \lambda)$ at each iteration $k \geq k_0$. We refer to this modified PNT method as PNT-md.

Algorithm 3.2. Projected Newton method (PNT) for (2.2) and (2.3).

```
Input: $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{M} \in \mathbb{R}^{m\times m}$, $\boldsymbol{N} \in \mathbb{R}^{n\times n}$, $\tau \gtrsim 1$
Initialization: $\lambda_0 > 0$, $\bar{\boldsymbol{y}}_0 = \boldsymbol{0}$; $c = 10^{-4}$, $\eta = 0.9$; tol $> 0$
Compute $\beta_1$, $\alpha_1$, $\boldsymbol{u}_1$, $\boldsymbol{v}_1$ by Algorithm 3.1
for $k = 1, 2, \dots$ do
    Compute $\beta_{k+1}$, $\alpha_{k+1}$, $\boldsymbol{u}_{k+1}$, $\boldsymbol{v}_{k+1}$ by Algorithm 3.1; Form $\boldsymbol{B}_{k+1}$ and $\boldsymbol{V}_k$
    (Terminate gen-GKB if $\beta_{k+1}$ or $\alpha_{k+1}$ is extremely small)
    Compute $F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ and $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ by (3.10) and (3.11)
    Compute $(\Delta\boldsymbol{y}_k, \Delta\lambda_k)$ by (3.12a)
    if $\Delta\lambda_k > 0$ then
        $\gamma_{\text{init}} = 1$
    else
        $\gamma_{\text{init}} = \min\{1, -\eta\lambda_{k-1}/\Delta\lambda_k\}$        ▷ Ensure the positivity of $\lambda_k$
    end if
    Determine the step-length $\gamma_k$ by Routine 1
    Update $(\boldsymbol{y}_k, \lambda_k)$ by (3.12b)
    if $\frac{1}{2}\|\bar{F}^{(k)}(\bar{\boldsymbol{y}}_k, \lambda_k)\|_2 \leq$ tol then
        Compute $\boldsymbol{x}_k = \boldsymbol{V}_k\boldsymbol{y}_k$; Stop iteration
    end if
end for
Output: Final solution $(\boldsymbol{x}_k, \lambda_k)$
```

From the derivations, we find that the success of PNT is attributed to the fact that $\boldsymbol{M}$ and $\boldsymbol{N}$ can induce inner products. Therefore, the PNT method cannot be directly used to handle the total variation (TV) or ℓ_1 regularization terms. One possible approach for handling TV or ℓ_1 norms is to approximate them with weighted ℓ_2 norms at each iterated point [50, 51]. Furthermore, for the nonlinear inverse problem $\boldsymbol{b} = G(\boldsymbol{x}) + \boldsymbol{\epsilon}$ with differentiable G, at each iterated point we can approximate $\|G(\boldsymbol{x}) - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2$ by a quadratic convex function using the first-order Taylor expansion of G. The above approaches follow a similar idea to the sequential quadratically constrained quadratic programming (SQCQP) method [16, 38]. This allows us to obtain a sequence of optimization problems similar to (2.2), which can be solved efficiently by PNT. Theoretical and computational aspects of this approach will be further studied in the future.

3.2. Proofs. We gave the proofs of all the results in subsection 3.1. Remember again that we always follow the notations as stated in Remark 2 and (3.13).

Proof of Lemma 3.2. By (3.8) and (3.9) we have

$$F^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \lambda(\boldsymbol{A}\boldsymbol{V}_k)^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{V}_k^\top \boldsymbol{N}^{-1}\boldsymbol{V}_k\boldsymbol{y} \\ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \frac{\tau m}{2} \end{pmatrix},$$

and

$$J^{(k)}(\boldsymbol{y}, \lambda) = \begin{pmatrix} \lambda(\boldsymbol{A}\boldsymbol{V}_k)^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{V}_k) + \boldsymbol{V}_k^\top \boldsymbol{N}^{-1}\boldsymbol{V}_k & (\boldsymbol{A}\boldsymbol{V}_k)^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b}) \\ (\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b})^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{V}_k) & 0 \end{pmatrix}.$$

Using relations (3.5) and Proposition 3.1, we have $\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b} = \boldsymbol{U}_{k+1}(\boldsymbol{B}_k - \beta_1 e_1)$, leading to

$$(\boldsymbol{A}\boldsymbol{V}_k)^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b}) = (\boldsymbol{U}_{k+1}\boldsymbol{B}_k)^\top \boldsymbol{M}^{-1}\boldsymbol{U}_{k+1}(\boldsymbol{B}_k - \beta_1 e_1) = \boldsymbol{B}_k^\top(\boldsymbol{B}_k - \beta_1 e_1),$$

and

$$\|\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \|\boldsymbol{U}_{k+1}(\boldsymbol{B}_k - \beta_1 e_1)\|_{\boldsymbol{M}^{-1}}^2 = \|\boldsymbol{B}_k - \beta_1 e_1\|_2^2.$$

If $\beta_{k_t+1} = 0$, then for $k \geq k_t$, the relation $\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y} - \boldsymbol{b} = \boldsymbol{U}_{k+1}(\boldsymbol{B}_k - \beta_1 e_1)$ is replaced by $\boldsymbol{A}\boldsymbol{V}_{k_t}\boldsymbol{y} - \boldsymbol{b} = \boldsymbol{U}_{k_t}(\underline{\boldsymbol{B}}_{k_t} - \beta_1 e_1)$. Therefore, the above identity is also applied to the case $k \geq k_t$. Now we have proved (3.10). The expression (3.11) can be proved similarly. □

In order to prove Lemma 3.5, we first give the following result.

LEMMA 3.7. *Let* $\widehat{\boldsymbol{N}} = \begin{pmatrix} \boldsymbol{N} & \\ & 1 \end{pmatrix}$. *Then we have the following identity:*

$$\|F(\boldsymbol{x}_{k-1}, \lambda_{k-1})\|_{\widehat{\boldsymbol{N}}} = \|F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})\|_2. \tag{3.18}$$

Proof. First, notice that

$$\|F(\boldsymbol{x}_{k-1}, \lambda_{k-1})\|_{\widehat{\boldsymbol{N}}}^2 = \|\lambda_{k-1}\boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}\|_{\boldsymbol{N}}^2 + \left(\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 - \frac{\tau m}{2}\right)^2.$$

For the first term of the above summation, we have

$$\begin{aligned} &\|\lambda_{k-1}\boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}_{k-1}\|_{\boldsymbol{N}}^2 \\ &\quad = (\lambda_{k-1}\boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}_{k-1})^\top \boldsymbol{N}(\lambda_{k-1}\boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}) \\ &\qquad + \boldsymbol{N}^{-1}\boldsymbol{x}_{k-1}), \end{aligned}$$

and

$$\begin{aligned}
&\boldsymbol{N}(\lambda_{k-1}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})+\boldsymbol{N}^{-1}\boldsymbol{x}_{k-1})\\
&\quad=\lambda_{k-1}\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{U}_{k+1}(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\boldsymbol{V}_k\bar{\boldsymbol{y}}_{k-1}\\
&\quad=\lambda_{k-1}(\boldsymbol{V}_k\boldsymbol{B}_k^\top+\alpha_{k+1}\boldsymbol{v}_{k+1}e_{k+1}^\top)(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\boldsymbol{V}_k\bar{\boldsymbol{y}}_{k-1}\\
&\quad=\boldsymbol{V}_k(\lambda_{k-1}\boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\bar{\boldsymbol{y}}_{k-1}),
\end{aligned}$$

where we have used

$$\alpha_{k+1}\boldsymbol{v}_{k+1}e_{k+1}^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)=\alpha_{k+1}\beta_{k+1}\boldsymbol{v}_{k+1}e_k^\top\bar{\boldsymbol{y}}_{k-1}=0,$$

because $e_k^\top\bar{\boldsymbol{y}}_{k-1}=0$ for $k\le k_t$ and $\alpha_{k+1}\beta_{k+1}=0$ for $k>k_t$. Similarly, we have

$$\lambda_{k-1}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})+\boldsymbol{N}^{-1}\boldsymbol{x}_{k-1}=\boldsymbol{N}^{-1}\boldsymbol{V}_k(\lambda_{k-1}\boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\bar{\boldsymbol{y}}_{k-1}).$$

We also have

$$\|\lambda_{k-1}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})+\boldsymbol{N}^{-1}\boldsymbol{x}_{k-1}\|_{\boldsymbol{N}}=\|\lambda_{k-1}\boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\bar{\boldsymbol{y}}_{k-1}\|_2$$

and

$$\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2-\frac{\tau m}{2}=\frac{1}{2}\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1\|_2^2-\frac{\tau m}{2}.$$

The desired result immediately follows by using (3.10). □

Proof of Lemma 3.5. First, notice that $h(\boldsymbol{x},\lambda)=\frac{1}{2}\|F(\boldsymbol{x},\lambda)\|_{\widehat{N}}^2$. Combining the above relation with Lemma 3.7 we obtain the first identity of (3.17). Also, for $k<k_t$ we have $h(\boldsymbol{x}_k,\lambda_k)=\frac{1}{2}\|F^{(k+1)}(\bar{\boldsymbol{y}}_k,\lambda_k)\|_2^2$ with

$$F^{(k+1)}(\bar{\boldsymbol{y}}_k,\lambda_k)=\begin{pmatrix}\lambda\boldsymbol{B}_{k+1}^\top(\boldsymbol{B}_{k+1}\bar{\boldsymbol{y}}_k-\beta_1e_1)+\bar{\boldsymbol{y}}_k\\ \dfrac{1}{2}\|\boldsymbol{B}_{k+1}\bar{\boldsymbol{y}}_k-\beta_1e_1\|_2^2-\dfrac{\tau m}{2}\end{pmatrix}.$$

Since the last element of β_1e_1 and $\bar{\boldsymbol{y}}_k$ is zero, it is easy to verify that

$$\boldsymbol{B}_{k+1}\bar{\boldsymbol{y}}_k-\beta_1e_1=\begin{pmatrix}\bar{\boldsymbol{B}}_k\bar{\boldsymbol{y}}_k-\beta_1e_1\\0\end{pmatrix},\quad \boldsymbol{B}_{k+1}^\top(\boldsymbol{B}_{k+1}\bar{\boldsymbol{y}}_k-\beta_1e_1)=\bar{\boldsymbol{B}}_k^\top(\bar{\boldsymbol{B}}_k\bar{\boldsymbol{y}}_k-\beta_1e_1).$$

For $k\ge k_t$, we have $F^{(k)}(\boldsymbol{y},\lambda)=F^{(k+1)}(\boldsymbol{y},\lambda)=\bar{F}^{(k)}(\boldsymbol{y},\lambda)$. Therefore, we prove the second identity of (3.17). □

In order to prove Theorem 3.4, we need Lemma 3.7 and the following result.

LEMMA 3.8. *For any $k\ge1$, we have the following identity:*

$$\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}J(\boldsymbol{x}_{k-1},\lambda_{k-1})\widehat{\boldsymbol{N}}F(\boldsymbol{x}_{k-1},\lambda_{k-1})=J^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1})F^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1}).$$

Proof. First, notice from (2.8) that

$$\begin{aligned}
&\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}J(\boldsymbol{x}_{k-1},\lambda_{k-1})\widehat{\boldsymbol{N}}\\
&\quad=\begin{pmatrix}\lambda_{k-1}(\boldsymbol{A}\boldsymbol{V}_k)^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N}+\boldsymbol{V}_k^\top & (\boldsymbol{A}\boldsymbol{V}_k)^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})\\ (\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N} & 0\end{pmatrix}.
\end{aligned}$$

Using (3.5c) and similar derivations to the proof of Lemma 3.7, we get

$$\begin{aligned}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N} &= (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)^\top\boldsymbol{U}_{k+1}^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N}\\ &= (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)^\top(\boldsymbol{B}_k\boldsymbol{V}_k^\top+\alpha_{k+1}e_{k+1}\boldsymbol{v}_{k+1}^\top)\\ &= (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)^\top\boldsymbol{B}_k\boldsymbol{V}_k^\top. \end{aligned}\tag{3.19}$$

Also, we can get

$$\begin{aligned}(\boldsymbol{A}\boldsymbol{V}_k)^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N} = (\boldsymbol{U}_{k+1}\boldsymbol{B}_k)^\top\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N} &= \boldsymbol{B}_k^\top(\boldsymbol{B}_k\boldsymbol{V}_k^\top+\alpha_{k+1}e_{k+1}\boldsymbol{v}_{k+1}^\top)\\ &= \boldsymbol{B}_k^\top\boldsymbol{B}_k\boldsymbol{V}_k^\top+\alpha_{k+1}\beta_{k+1}e_k\boldsymbol{v}_{k+1}^\top,\end{aligned}$$

and

$$\begin{aligned}(\boldsymbol{A}\boldsymbol{V}_k)^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b}) &= (\boldsymbol{B}_k\boldsymbol{U}_{k+1})^\top\boldsymbol{M}^{-1}\boldsymbol{U}_{k+1}(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)\\ &= \boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1).\end{aligned}$$

Using (3.11), we get

$$\begin{aligned}&\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}J(\boldsymbol{x}_{k-1},\lambda_{k-1})\widehat{\boldsymbol{N}}\\ &= \begin{pmatrix}(\lambda_{k-1}\boldsymbol{B}_k^\top\boldsymbol{B}_k+\boldsymbol{I})\boldsymbol{V}_k^\top & \boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)\\ (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)^\top\boldsymbol{B}_k\boldsymbol{V}_k^\top & 0\end{pmatrix}+\begin{pmatrix}\alpha_{k+1}\beta_{k+1}e_{k+1}\boldsymbol{v}_{k+1}^\top & \\ & 0\end{pmatrix}\\ &= J^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1})\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}+\begin{pmatrix}\alpha_{k+1}\beta_{k+1}e_k\boldsymbol{v}_{k+1}^\top & \\ & 0\end{pmatrix}.\end{aligned}$$

Using similar derivations to the proof of Lemma 3.7, we get

$$\begin{aligned}F(\boldsymbol{x}_{k-1},\lambda_{k-1}) &= \begin{pmatrix}\boldsymbol{N}^{-1} & \\ & 1\end{pmatrix}\begin{pmatrix}\lambda_{k-1}\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b})+\boldsymbol{x}_{k-1}\\ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}_{k-1}-\boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2-\frac{\tau m}{2}\end{pmatrix}\\ &= \begin{pmatrix}\boldsymbol{N}^{-1} & \\ & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_k & \\ & 1\end{pmatrix}\begin{pmatrix}\lambda_{k-1}\boldsymbol{B}_k^\top(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1)+\bar{\boldsymbol{y}}_{k-1}\\ \frac{1}{2}\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1}-\beta_1e_1\|_2^2-\frac{\tau m}{2}\end{pmatrix}\\ &= \begin{pmatrix}\boldsymbol{N}^{-1} & \\ & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_k & \\ & 1\end{pmatrix}F^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1}).\end{aligned}$$

Using the relations

$$\begin{aligned}&\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{N}^{-1} & \\ & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_k & \\ & 1\end{pmatrix}\\ &= \boldsymbol{I},\quad \begin{pmatrix}\alpha_{k+1}\beta_{k+1}e_k\boldsymbol{v}_{k+1}^\top & \\ & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{N}^{-1} & \\ & 1\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_k & \\ & 1\end{pmatrix}=\boldsymbol{0},\end{aligned}$$

we finally obtain the desired result. ∎

Proof of Theorem 3.4. Notice $J(\boldsymbol{x},\lambda)$ is the Jacobian of $F(\boldsymbol{x},\lambda)$. Using $h(\boldsymbol{x},\lambda)=\frac{1}{2}\|F(\boldsymbol{x},\lambda)\|_{\widehat{N}}^2$ we get $\nabla h(\boldsymbol{x},\lambda)=J(\boldsymbol{x},\lambda)\widehat{\boldsymbol{N}}F(\boldsymbol{x},\lambda)$, leading to

$$\begin{aligned}\nabla h(\boldsymbol{x}_{k-1},\lambda_{k-1})^\top\begin{pmatrix}\Delta\boldsymbol{x}_k\\ \Delta\lambda_k\end{pmatrix} &= \begin{pmatrix}\Delta\boldsymbol{y}_k\\ \Delta\lambda_k\end{pmatrix}^\top\begin{pmatrix}\boldsymbol{V}_k^\top & \\ & 1\end{pmatrix}J(\boldsymbol{x}_{k-1},\lambda_{k-1})\widehat{\boldsymbol{N}}F(\boldsymbol{x}_{k-1},\lambda_{k-1})\\ &= \begin{pmatrix}\Delta\boldsymbol{y}_k\\ \Delta\lambda_k\end{pmatrix}^\top J^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1})F^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1})\\ &= -\|F^{(k)}(\bar{\boldsymbol{y}}_{k-1},\lambda_{k-1})\|_2^2 = -\|F(\boldsymbol{x}_{k-1},\lambda_{k-1})\|_{\widehat{\boldsymbol{N}}}^2\\ &= -2h(\boldsymbol{x}_{k-1},\lambda_{k-1})\leq 0,\end{aligned}$$

where we have used Lemmas 3.7 and 3.8. ∎

4. Convergence analysis. The objective of this section is to prove the convergence of PNT, which is stated in the following result and Corollary 4.8.

THEOREM 4.1. *Suppose the PNT algorithm is initialized with* $\bar{\boldsymbol{y}}_0 = \boldsymbol{0}$*,* $\boldsymbol{x}_0 = \boldsymbol{0}$*, and* $\lambda_0 > 0$*. Then we either have*

$$h(\boldsymbol{x}_k, \lambda_k) = 0 \tag{4.1}$$

for some $k < \infty$*, or have*

$$\lim_{k\to\infty} h(\boldsymbol{x}_k, \lambda_k) = 0. \tag{4.2}$$

Notice that $(\boldsymbol{x}^*, \lambda^*)$ is the unique minimizer of $h(\boldsymbol{x}, \lambda)$ and $h(\boldsymbol{x}^*, \lambda^*) = 0$. Therefore, (4.1) implies that the algorithm finds the exact solution to (2.2) and (2.3) at the kth iteration. In the following part, we prove (4.2) under the assumption that $h(\boldsymbol{x}_k, \lambda_k) > 0$ for any $k \geq 1$. We need a series of lemmas, which are Lemmas 4.2–4.7. All of these lemmas follow the same assumption of Theorem 4.1.

LEMMA 4.2. *For any matrix* $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ *with full column rank and* $\boldsymbol{d} \in \mathbb{R}^m$*, if the vector sequence* $\{\boldsymbol{w}_k\} \in \mathbb{R}^n$ *satisfies* $\lim_{k\to\infty} \|\boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{w}_k - \boldsymbol{d})\|_2 = 0$*, then* $\lim_{k\to\infty} \boldsymbol{w}_k = \boldsymbol{w}^* := \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \|\boldsymbol{C}\boldsymbol{w} - \boldsymbol{d}\|_2$.

Proof. First, notice that $\boldsymbol{w}^*$ is well-defined, since $\operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^n} \|\boldsymbol{C}\boldsymbol{w} - \boldsymbol{d}\|_2$ has a unique solution for the full column rank matrix $\boldsymbol{C}$. For any $\boldsymbol{w}_k$, let $\boldsymbol{w}_k = \boldsymbol{w}^* + \bar{\boldsymbol{w}}_k$. Then we have

$$\lim_{k\to\infty} \|\boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{w}_k - \boldsymbol{d})\|_2 = \lim_{k\to\infty} \|\boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{w}^* - \boldsymbol{d}) + \boldsymbol{C}^\top\boldsymbol{C}\bar{\boldsymbol{w}}_k\|_2 = \lim_{k\to\infty} \|\boldsymbol{C}^\top\boldsymbol{C}\bar{\boldsymbol{w}}_k\|_2 = 0,$$

since $\boldsymbol{C}^\top(\boldsymbol{C}\boldsymbol{w}^* - \boldsymbol{d}) = \boldsymbol{0}$. Now we have $\|\bar{\boldsymbol{w}}_k\|_2 \to 0$ since $\boldsymbol{C}^\top\boldsymbol{C}$ is positive definite and all norms of $\mathbb{R}^n$ are equivalent. Therefore, we have $\|\boldsymbol{w}_k - \boldsymbol{w}^*\|_2 \to 0$ or the equivalent form $\lim_{k\to\infty} \boldsymbol{w}_k = \boldsymbol{w}^*$. □

LEMMA 4.3. *If the unique solution to* $\min_{\boldsymbol{y}\in\mathbb{R}^{k_t}} \|\boldsymbol{B}_{k_t} y - \beta_1 e_1\|_2$ *is* $\boldsymbol{y}_{\min}$*, then* $\boldsymbol{x}_{\min} := \boldsymbol{V}_{k_t}\boldsymbol{y}_{\min}$ *is the unique solution to*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{x}\|_{\boldsymbol{N}^{-1}} \quad \text{s.t.} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \min. \tag{4.3}$$

Proof. It is easy to verify that both $\min_{\boldsymbol{y}\in\mathbb{R}^{k_t}} \|\boldsymbol{B}_{k_t} y - \beta_1 e_1\|_2$ and (4.3) have a unique solution. A vector $\boldsymbol{x}$ is the unique solution to (4.3) if and only if

$$\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}, \quad \boldsymbol{x} \perp_{\boldsymbol{N}^{-1}} \mathcal{N}(\boldsymbol{A}),$$

where $\perp_{\boldsymbol{N}^{-1}}$ means the orthogonality relation under the $\boldsymbol{N}^{-1}$-inner product. Now we verify the above two conditions for $\boldsymbol{x}_{\min}$. For the first condition, using the relations $\boldsymbol{A}\boldsymbol{x}_{\min} = \boldsymbol{A}\boldsymbol{V}_{k_t}\boldsymbol{y}_{\min} = \boldsymbol{U}_{k_t+1}\boldsymbol{B}_{k_t}\boldsymbol{y}_{\min}$ and (3.5c), we get

$$\begin{aligned}
\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{\min} - \boldsymbol{b}) &= \boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{U}_{k_t+1}(\boldsymbol{B}_{k_t}\boldsymbol{y}_{\min} - \beta_1 e_1) \\
&= \boldsymbol{N}^{-1}(\boldsymbol{V}_{k_t}\boldsymbol{B}_{k_t}^\top + \alpha_{k_t+1}\boldsymbol{v}_{k_t+1}e_{k_t+1}^\top)(\boldsymbol{B}_{k_t}\boldsymbol{y}_{\min} - \beta_1 e_1) \\
&= \boldsymbol{N}^{-1}[\boldsymbol{V}_{k_t}\boldsymbol{B}_{k_t}^\top(\boldsymbol{B}_{k_t}\boldsymbol{y}_{\min} - \beta_1 e_1) + \alpha_{k_t+1}\beta_{k_t+1}\boldsymbol{v}_{k_t+1}e_{k_t}^\top\boldsymbol{y}_{\min}] \\
&= \boldsymbol{0},
\end{aligned}$$

since $\boldsymbol{B}_{k_t}^\top(\boldsymbol{B}_{k_t}\boldsymbol{y}_{\min} - \beta_1 e_1) = \boldsymbol{0}$ and $\alpha_{k_t+1}\beta_{k_t+1} = 0$. For the second condition, by Proposition 3.1 we have

$$\begin{aligned}
\boldsymbol{x}_{\min} \in \operatorname{span}\{\boldsymbol{V}_{k_t}\} &= \operatorname{span}\{(\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{A})^i\boldsymbol{N}\boldsymbol{A}^\top\boldsymbol{M}^{-1}\boldsymbol{b}\}_{i=0}^{k_t-1} \\
&\subseteq \mathcal{R}(\boldsymbol{N}\boldsymbol{A}^\top) = \boldsymbol{N}\mathcal{N}(\boldsymbol{A})^\perp.
\end{aligned}$$

Write $\boldsymbol{x}_{\min} = \boldsymbol{N}\bar{\boldsymbol{x}}_{\min}$ with $\bar{\boldsymbol{x}}_{\min} \in \mathcal{N}(\boldsymbol{A})^{\perp}$. For any $\boldsymbol{w} \in \mathcal{N}(\boldsymbol{A})$, we have

$$\langle \boldsymbol{x}_{\min}, \boldsymbol{w}\rangle_{\boldsymbol{N}^{-1}} = \langle \boldsymbol{N}\bar{\boldsymbol{x}}_{\min}, \boldsymbol{w}\rangle_{\boldsymbol{N}^{-1}} = \langle \bar{\boldsymbol{x}}_{\min}, \boldsymbol{w}\rangle_2 = 0.$$

Therefore, it holds that $\boldsymbol{x}_{\min} \perp_{\boldsymbol{N}^{-1}} \mathcal{N}(\boldsymbol{A})$. □

LEMMA 4.4. *There exist a positive constant C_1 such that for any $k \geq 1$,*

$$\|\boldsymbol{A}^{\top}\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b})\|_{\boldsymbol{N}} = \|\boldsymbol{B}_k^{\top}(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)\|_2 \geq C_1 > 0. \tag{4.4}$$

Proof. First, we get from (3.19) the first identity:

$$\begin{aligned}
&\|\boldsymbol{A}^{\top}\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b})\|_{\boldsymbol{N}}^2 \\
&\quad = (\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b})^{\top}\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N}\boldsymbol{N}^{-1}\left((\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b})^{\top}\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{N}\right)^{\top} \\
&\quad = \left(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\right)^{\top}\boldsymbol{B}_k\boldsymbol{V}_k^{\top}\boldsymbol{N}^{-1}\boldsymbol{V}_k^{\top}\boldsymbol{B}_k^{\top}\left(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\right) \\
&\quad = \|\boldsymbol{B}_k^{\top}\left(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\right)\|_2^2.
\end{aligned}$$

Then, we prove

$$\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2 \geq \sqrt{\tau m} \tag{4.5}$$

by mathematical induction. For $k = 1$, we have $\|\boldsymbol{A}\boldsymbol{x}_0 - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \|\boldsymbol{b}\|_{\boldsymbol{M}^{-1}} > \sqrt{\tau m}$ since $\boldsymbol{x}_0 = \boldsymbol{0}$. Suppose $\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} \geq \sqrt{\tau m}$ for $k \geq 1$. We have

$$\begin{aligned}
&\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 = \|\boldsymbol{A}\boldsymbol{V}_k(\bar{\boldsymbol{y}}_{k-1} + \gamma_k\Delta\boldsymbol{y}_k) - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 \\
&\quad = \|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 + \gamma_k^2\|\boldsymbol{A}\boldsymbol{V}_k\Delta\boldsymbol{y}_k\|_{\boldsymbol{M}^{-1}}^2 + 2\gamma_k(\boldsymbol{A}\boldsymbol{x}_{k-1} - b)^{\top}\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{V}_k\Delta\boldsymbol{y}_k \\
&\quad = \|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2^2 + \gamma_k^2\|\boldsymbol{A}\boldsymbol{V}_k\Delta\boldsymbol{y}_k\|_{\boldsymbol{M}^{-1}}^2 + 2\gamma_k(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)^{\top}\boldsymbol{B}_k\Delta\boldsymbol{y},
\end{aligned}$$

since

$$\begin{aligned}
(\boldsymbol{A}\boldsymbol{x}_{k-1} - b)^{\top}\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{V}_k &= (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)^{\top}\boldsymbol{U}_{k+1}^{\top}\boldsymbol{M}^{-1}\boldsymbol{U}_{k+1}\boldsymbol{B}_k \\
&= (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)^{\top}\boldsymbol{B}_k.
\end{aligned}$$

Writing $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})\binom{\Delta\boldsymbol{y}_k}{\Delta\lambda_k} = -F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ in the matrix form and using $\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2 \geq \sqrt{\tau m}$, we get from the second equality of the above equation that $(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)^{\top}\boldsymbol{B}_k\Delta\boldsymbol{y} = -\frac{1}{2}(\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2^2 - \tau m) \leq 0$. Since $\gamma_k \leq 1$, we get

$$\begin{aligned}
&\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}}^2 \\
&\quad \geq \|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2^2 + \gamma_k^2\|\boldsymbol{A}\boldsymbol{V}_k\Delta\boldsymbol{y}_k\|_{\boldsymbol{M}^{-1}}^2 - \left(\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2^2 - \tau m\right) \\
&\quad = \tau m + \gamma_k^2\|\boldsymbol{A}\boldsymbol{V}_k\Delta\boldsymbol{y}_k\|_{\boldsymbol{M}^{-1}}^2 \geq \tau m.
\end{aligned}$$

Therefore, we prove (4.5).

To obtain the lower bound in (4.4), we investigate two cases: $k < k_t$ and $k \geq k_t$.

Case 1. $k < k_t$. For this case, we have

$$\|\boldsymbol{B}_k^{\top}\left(\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\right)\|_2 \geq \sigma_{\min}(\boldsymbol{B}_k)\|\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1\|_2 \geq \sigma_{\min}(\underline{\boldsymbol{B}}_{k_t})\sqrt{\tau m} > 0,$$

where $\sigma_{\min}(\cdot)$ is the smallest singular value of a matrix, and $\underline{\boldsymbol{B}}_{k_t}$ is the first $k_t \times k_t$ part of $\boldsymbol{B}_{k_t}$.

Case 2. $k \geq k_t$. For this case, we can write $\boldsymbol{x}_{k-1}$ as $\boldsymbol{x}_{k-1} = \boldsymbol{V}_{k_t}\bar{\boldsymbol{y}}_{k-1}$. Remember that $\bar{\boldsymbol{y}}_{k-1} = \boldsymbol{y}_{k-1}$ if $k > k_t$. We first prove $\|\boldsymbol{B}_{k_t}^\top(\boldsymbol{B}_{k_t}\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)\|_2 \neq 0$. If it is not true, then $\bar{\boldsymbol{y}}_{k-1} = \operatorname{argmin}_{\boldsymbol{y}} \|\boldsymbol{B}_{k_t}\boldsymbol{y} - \beta_1 e_1\|_2$. By Lemma 4.3, $\boldsymbol{x}_{k-1}$ is the solution to (4.3). Thus, it must hold that $\|\boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} < \sqrt{\tau m}$ by Assumption 1, which contradicts (4.5). Now suppose the lower bound in (4.4) is not true. Then there exists a subsequence $\{\bar{\boldsymbol{y}}_{k_j-1}\}$ with $k_j \geq k_t$ such that $\lim_{j\to\infty}\|\boldsymbol{B}_{k_t}^\top(\boldsymbol{B}_{k_t}\bar{\boldsymbol{y}}_{k_j-1} - \beta_1 e_1)\|_2 = 0$. By Lemma 4.2, we have $\lim_{j\to\infty}\bar{\boldsymbol{y}}_{k_j-1} = \boldsymbol{y}_{\min} := \operatorname{argmin}_{\boldsymbol{y}}\|\boldsymbol{B}_{k_t}\boldsymbol{y} - \beta_1 e_1\|_2$, leading to $\lim_{j\to\infty}\boldsymbol{x}_{k_j-1} = \lim_{k_j\to\infty}\boldsymbol{V}_{k_t}\bar{\boldsymbol{y}}_{k_j-1} = \boldsymbol{V}_{k_t}\boldsymbol{y}_{\min}$. It follows from Lemma 4.3 that $\boldsymbol{V}_{k_t}\boldsymbol{y}_{\min} = \boldsymbol{x}_{\min}$, which is the solution to (4.3). Therefore, it must hold that$\lim_{j\to\infty}\|\boldsymbol{A}\boldsymbol{x}_{k_j-1} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} = \|\boldsymbol{A}\boldsymbol{x}_{\min} - \boldsymbol{b}\|_{\boldsymbol{M}^{-1}} < \sqrt{\tau m}$ by Assumption 1, which contradicts (4.5). Summarizing both of the two cases, the desired result is proved. □

LEMMA 4.5. *The points* $\{(\boldsymbol{x}_k, \lambda_k)\}_{i=0}^\infty$ *generated by the PNT algorithm lie in a bounded set of* $\mathbb{R}^n \times \mathbb{R}^+$.

Proof. First, notice that $h(\boldsymbol{x}_0, \lambda_0) \geq h(\boldsymbol{x}_1, \lambda_1) \geq \cdots$. We need only prove $\{(\boldsymbol{x}_k, \lambda_k)\}_{k\geq k_t}$ is bounded above. In this case, notice that $\boldsymbol{x}_k = \boldsymbol{V}_{k_t}\boldsymbol{y}_k$ and

$$h(\boldsymbol{x}_k, \lambda_k) = \frac{1}{2}\left[\|\lambda_k \boldsymbol{A}^\top \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}_k\|_{\boldsymbol{N}}^2 + \left(\frac{1}{2}\|\boldsymbol{B}_{k_t}\boldsymbol{y}_k - \beta_1 e_1\|_2^2 - \frac{\tau m}{2}\right)^2\right].$$

If the points do not lie in a bounded set, there exists a subsequence $\{(\boldsymbol{x}_{k_j}, \lambda_{k_j})\}$ with $k_j \geq k_t$ such that $(\boldsymbol{x}_{k_j}, \lambda_{k_j}) \to \infty$. If $\|\boldsymbol{x}_{k_j}\|_2 \to \infty$, then $\|\boldsymbol{y}_{k_j}\|_2 \to \infty$, since $\boldsymbol{x}_{k_j} = \boldsymbol{V}_{k_t}\boldsymbol{y}_{k_j}$ and $\boldsymbol{V}_{k_t}$ has full column rank. This leads to $\|\boldsymbol{B}_{k_t}\boldsymbol{y}_{k_j}\|_2 \to \infty$ since $\boldsymbol{B}_{k_t}$ has full column rank. It follows that the second term of $h(\boldsymbol{x}_{k_j}, \lambda_{k_j})$ tends to infinity and $h(\boldsymbol{x}_{k_j}, \lambda_{k_j}) \to \infty$, a contradiction. Therefore, it must hold that $\|\boldsymbol{x}_{k_j}\|_2$ is bounded above and $\lambda_{k_j} \to \infty$. By Lemma 4.4, we have $\|\lambda_{k_j}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k_j} - \boldsymbol{b})\|_{\boldsymbol{N}} \geq \lambda_{k_j}C_1$. Notice that $\{\boldsymbol{N}^{-1}\boldsymbol{x}_{k_j}\}$ lie in a bounded set. It follows that $\|\lambda_{k_j}\boldsymbol{A}^\top\boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}_{k_j} - \boldsymbol{b}) + \boldsymbol{N}^{-1}\boldsymbol{x}_{k_j}\|_{\boldsymbol{N}} \to \infty$ and $h(\boldsymbol{x}_{k_j}, \lambda_{k_j}) \to \infty$, also a contradiction. □

LEMMA 4.6. *There exist a positive constant* $C_2 < +\infty$ *such that for any* $k \geq 1$,

$$\|J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})^{-1}\|_2 \leq C_2. \tag{4.6}$$

Proof. First, we prove that $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ is always nonsingular. Write it as

$$J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1}) = \begin{pmatrix} \lambda_{k-1}\boldsymbol{B}_{k_t}^\top\boldsymbol{B}_k + \boldsymbol{I} & \boldsymbol{B}_k^\top(\boldsymbol{B}_{k_t}\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1) \\ (\boldsymbol{B}_k\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)^\top\boldsymbol{B}_k & 0 \end{pmatrix} =: \begin{pmatrix} \boldsymbol{C}_k & \boldsymbol{d}_k \\ \boldsymbol{d}_k^\top & 0 \end{pmatrix}$$

and notice that

$$\begin{pmatrix} \boldsymbol{C}_k & \boldsymbol{d}_k \\ \boldsymbol{d}_k^\top & 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{C}_k & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1}\boldsymbol{d}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k \\ \boldsymbol{0} & 1 \end{pmatrix}^{-1}.$$

It follows that

$$\begin{pmatrix} \boldsymbol{C}_k & \boldsymbol{d}_k \\ \boldsymbol{d}_k^\top & 0 \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{I} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k \\ \boldsymbol{0} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{C}_k^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & -(\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1}\boldsymbol{d}_k)^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1} & 1 \end{pmatrix}, \tag{4.7}$$

Since $\boldsymbol{C}_k$ is positive definite and $\|\boldsymbol{d}_k\|_2 \geq C_1 > 0$.

To give an upper bound on $\|J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})^{-1}\|_2$, we need only consider $k \geq k_t$, where $\boldsymbol{B}_k = \boldsymbol{B}_{k_t}$ in $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$. Since $\sigma_{\min}(\boldsymbol{C}_k) = \sigma_{\min}(\lambda_{k-1}\boldsymbol{B}_{k_t}^\top\boldsymbol{B}_{k_t} + \boldsymbol{I}) \geq 1$, we have $\|\boldsymbol{C}_k^{-1}\|_2 \leq 1$. By Lemma 4.5, there exist a positive constant $C_3 < +\infty$ such that $\lambda_k \leq C_3$, thereby

$$\sigma_{\max}(\boldsymbol{C}_k) \leq \sigma_{\max}(\lambda_{k-1}\boldsymbol{B}_{k_t}^\top\boldsymbol{B}_{k_t}) + \sigma_{\max}(\boldsymbol{I}) \leq C_3\sigma_{\max}(\boldsymbol{B}_{k_t}^\top\boldsymbol{B}_{k_t}) + 1 =: \bar{C}_3.$$

By Lemma 4.4 we have $\|\boldsymbol{d}_k\|_2 = \|\boldsymbol{B}_{k_t}^\top(\boldsymbol{B}_{k_t}\bar{\boldsymbol{y}}_{k-1} - \beta_1 e_1)\|_2 \geq C_1$. On the other hand, by Lemma 4.5 we know that $\|\boldsymbol{x}_{k-1}\|_2 = \|\boldsymbol{V}_{k_t}\bar{\boldsymbol{y}}_{k-1}\|_2$ is bounded above, thereby $\|\bar{\boldsymbol{y}}_{k-1}\|_2$ is bounded above since $\boldsymbol{V}_{k_t}$ has full column rank. Thus, there exists a positive constant $\bar{C}_1$ such that $\|\boldsymbol{d}_k\|_2 \leq \bar{C}_1$, leading to

$$\|\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\|_2 \leq \|\boldsymbol{C}_k^{-1}\|_2\|\boldsymbol{d}_k\|_2 \leq \bar{C}_1\sigma_{\min}(\boldsymbol{C}_k)^{-1} \leq \bar{C}_1,$$

and

$$\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1}\boldsymbol{d}_k \geq \sigma_{\min}(\boldsymbol{C}_k^{-1})\|\boldsymbol{d}_k\|_2^2 = \sigma_{\max}(\boldsymbol{C}_k)^{-1}\|\boldsymbol{d}_k\|_2^2 \geq C_1^2/\bar{C}_3 > 0.$$

Therefore, we have

$$\begin{aligned}\left\|\begin{pmatrix}\boldsymbol{0} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\\ \boldsymbol{0} & 0\end{pmatrix}\right\|_2^2 &= \left\|\begin{pmatrix}\boldsymbol{0} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\\ \boldsymbol{0} & 0\end{pmatrix}^\top\begin{pmatrix}\boldsymbol{0} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\\ \boldsymbol{0} & 0\end{pmatrix}\right\|_2\\ &= \left\|\begin{pmatrix}\boldsymbol{0} & \\ & \|\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\|_2^2\end{pmatrix}\right\|_2 \leq \bar{C}_1^2\end{aligned}$$

and

$$\left\|\begin{pmatrix}\boldsymbol{I} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\\ \boldsymbol{0} & 1\end{pmatrix}\right\|_2 \leq \left\|\begin{pmatrix}\boldsymbol{I} & \\ & 1\end{pmatrix}\right\|_2 + \left\|\begin{pmatrix}\boldsymbol{0} & -\boldsymbol{C}_k^{-1}\boldsymbol{d}_k\\ \boldsymbol{0} & 0\end{pmatrix}\right\|_2 \leq 1 + \bar{C}_1.$$

Similarly, we have

$$\left\|\begin{pmatrix}\boldsymbol{C}_k^{-1} & \boldsymbol{0}\\ \boldsymbol{0} & -(\boldsymbol{d}_k^\top\boldsymbol{C}_k^{-1}\boldsymbol{d}_k)^{-1}\end{pmatrix}\right\|_2 \leq \max\{1, \bar{C}_3/C_1^2\}.$$

Using the expression of $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ in (4.7), we finally obtain the desired result. ∎

Lemma 4.7. *There exists a positive constant C_4 such that for any $k \geq 1$,*

$$\gamma_k \geq C_4 > 0. \tag{4.8}$$

Proof. By Theorems 3.6 and 3.4, at each iteration the Armijo backtracking line search must terminate in finite steps with a γ_k satisfying

$$\gamma_k \geq \min\left\{1, \frac{4(1-c)\eta h(\boldsymbol{x}_{k-1}, \lambda_{k-1})}{\zeta(\boldsymbol{x}_{k-1}, \lambda_{k-1})\|(\Delta\boldsymbol{x}_k^\top, \Delta\lambda_k)\|_2^2}\right\},$$

where $\zeta(\boldsymbol{x}_{k-1}, \lambda_{k-1})$ is the Lipschitz constant of ∇h at $(\boldsymbol{x}_{k-1}, \lambda_{k-1})$; see also [4, p. 122, Corollary 2.1]. Now we prove $\zeta(\boldsymbol{x}_{k-1}, \lambda_{k-1})$ are bounded above. Notice that $\nabla h(\boldsymbol{x}, \lambda) = J(\boldsymbol{x}, \lambda)\widehat{N}F(\boldsymbol{x}, \lambda)$. Thus, all the elements in the Jacobian of $\nabla h(\boldsymbol{x}, \lambda)$ are polynomials of $(\boldsymbol{x}, \lambda)$ with degrees not bigger than 4. Since $\{(\boldsymbol{x}_{k-1}, \lambda_{k-1})\}$ lie in a bounded set, the norms of the Jacobians of $\nabla h(\boldsymbol{x}, \lambda)$ at the points $\{(\boldsymbol{x}_{k-1}, \lambda_{k-1})\}$ are bounded above. Therefore, the Lipschitz constants $\zeta(\boldsymbol{x}_{k-1}, \lambda_{k-1})$ are bounded above.

Let $\zeta(\boldsymbol{x}_{k-1}, \lambda_{k-1}) \leq \zeta_0$ with $0 < \zeta_0 < +\infty$ for any $k \geq 1$. Then by Lemmas 4.6 and 3.7, we have

$$\begin{aligned}\left\|\begin{pmatrix}\Delta\boldsymbol{x}_k\\ \Delta\lambda_k\end{pmatrix}\right\|_2 &\leq \left\|\begin{pmatrix}\boldsymbol{V}_k & \\ & 1\end{pmatrix}\right\|_2\left\|\begin{pmatrix}\Delta\boldsymbol{y}_k\\ \Delta\lambda_k\end{pmatrix}\right\|_2\\ &\leq (\|\boldsymbol{V}_{k_t}\|_2 + 1)\|J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})^{-1}\|_2\|F^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})\|_2\\ &\leq C_2(\|\boldsymbol{V}_{k_t}\|_2 + 1)(2h(\boldsymbol{x}_{k-1}, \lambda_{k-1}))^{1/2}.\end{aligned}$$

Then we obtain

$$\gamma_k \geq \min\left\{1, \frac{4(1-c)\eta h(\boldsymbol{x}_{k-1},\lambda_{k-1})}{\zeta_0\|(\Delta\boldsymbol{x}_k^\top,\Delta\lambda_k)^\top\|_2^2}\right\} \geq \min\left\{1, \frac{2(1-c)\eta}{\zeta_0 C_2^2(\|\boldsymbol{V}_{k_t}\|_2+1)^2}\right\} =: C_4.$$

The desired result is obtained. □

Proof of Theorem 4.1. By Lemma 4.5, the sequence $\{(\boldsymbol{x}_k,\lambda_k)\}_{k=1}^\infty$ is contained in a bounded set, thereby there exists a convergent subsequence $\{(\boldsymbol{x}_{k_j},\lambda_{k_j})\}_{j=1}^\infty$. Suppose $(\boldsymbol{x}_{k_j},\lambda_{k_j}) \to (\widehat{\boldsymbol{x}},\widehat{\lambda})$. It follows that $h(\boldsymbol{x}_{k_j},\lambda_{k_j}) \to h(\widehat{\boldsymbol{x}},\widehat{\lambda})$ since $h(\boldsymbol{x},\lambda)$ is continuous. Note that $h(\boldsymbol{x}_{k_j},\lambda_{k_j})$ is nonincreasing, thereby $h(\widehat{\boldsymbol{x}},\widehat{\lambda}) \leq h(\boldsymbol{x}_{k_j},\lambda_{k_j})$ for any k_j. Thus, for any $\varepsilon>0$, there exist a $k_\star \in \mathbb{N}$ such that $h(\boldsymbol{x}_{k_j},\lambda_{k_j}) < h(\widehat{\boldsymbol{x}},\widehat{\lambda})+\varepsilon,\ k_j>k_\star$. Select one k_j that satisfies $k_j>k_\star$. For any $k\geq k_j$, we have $h(\boldsymbol{x}_k,\lambda_k)\leq h(\boldsymbol{x}_{k_j},\lambda_{k_j}) < h(\widehat{\boldsymbol{x}},\widehat{\lambda})+\varepsilon$, which means that $\lim_{k\to\infty} h(\boldsymbol{x}_k,\lambda_k) = h(\widehat{\boldsymbol{x}},\widehat{\lambda})$. The Armijo condition and Theorem 3.4 lead to $h(\boldsymbol{x}_{k+1},\lambda_{k+1}) - h(\boldsymbol{x}_k,\lambda_k) \leq c\gamma_k\left(\Delta\boldsymbol{x}_k^\top,\Delta\lambda_k\right)\nabla h(\boldsymbol{x}_k,\lambda_k) \leq 0$. Taking the limit on both sides leads to $\lim_{k\to\infty} c\gamma_k\left(\Delta\boldsymbol{x}_k^\top,\Delta\lambda_k\right)\nabla h(\boldsymbol{x}_k,\lambda_k) = 0$. By Lemma 4.7 we get $\lim_{k\to\infty}\left(\Delta\boldsymbol{x}_k^\top,\Delta\lambda_k\right)\nabla h(\boldsymbol{x}_k,\lambda_k) = 0$. Noticing by Theorem 3.4 that $-2h(\boldsymbol{x}_k,\lambda_k) = \left(\Delta\boldsymbol{x}_k^\top,\Delta\lambda_k\right)\nabla h(\boldsymbol{x}_k,\lambda_k)$, we get $h(\widehat{\boldsymbol{x}},\widehat{\lambda}) = \lim_{k\to\infty} h(\boldsymbol{x}_k,\lambda_k) = 0$. This proves the desired result. □

Now we can give the convergence result of $(\boldsymbol{x}_k,\lambda_k)$.

COROLLARY 4.8. *The sequence* $\{(\boldsymbol{x}_k,\lambda_k)\}_{k=0}^\infty$ *generated by the PNT algorithm eventually converges to* $(\boldsymbol{x}^*,\lambda^*)$, *i.e., the solution of* (2.2) *and the corresponding Lagrange multiplier.*

Proof. Using the same notations as the proof of Theorem 4.1, we obtain that $(\widehat{\boldsymbol{x}},\widehat{\lambda}) = (\boldsymbol{x}^*,\lambda^*)$, since $h(\boldsymbol{x},\lambda)$ has the unique zero point $(\boldsymbol{x}^*,\lambda^*)$. Therefore, the subsequence $\{(\boldsymbol{x}_{k_j},\lambda_{k_j})\}_{j=1}^\infty$ defined in the proof of Theorem 4.1 converges to $(\boldsymbol{x}^*,\lambda^*)$. Now we need to prove the whole sequence $\{(\boldsymbol{x}_k,\lambda_k)\}_{k=1}^\infty$ converges to $(\boldsymbol{x}^*,\lambda^*)$. Assume that there is a subsequence $\{(\boldsymbol{x}_{l_j},\lambda_{l_j})\}_{j=1}^\infty$ that does not converge to $(\boldsymbol{x}^*,\lambda^*)$. We can select a subsequence from $\{(\boldsymbol{x}_{l_j},\lambda_{l_j})\}_{j=1}^\infty$ that converges to a point $(\bar{\boldsymbol{x}},\bar{\lambda}) \neq (\boldsymbol{x}^*,\lambda^*)$. Since $h(\boldsymbol{x}_{l_j},\lambda_{l_j})$ is nonincreasing with respect to j, using the same procedure as the proof of Theorem 4.1, we can obtain again that $h(\bar{\boldsymbol{x}},\bar{\lambda})=0$, leading to $(\bar{\boldsymbol{x}},\bar{\lambda}) = (\boldsymbol{x}^*,\lambda^*)$, a contradiction. Therefore, any subsequence of $\{(\boldsymbol{x}_k,\lambda_k)\}_{k=0}^\infty$ converges to $(\boldsymbol{x}^*,\lambda^*)$, thereby $\{(\boldsymbol{x}_k,\lambda_k)\}_{k=0}^\infty$ converges to $(\boldsymbol{x}^*,\lambda^*)$. □

5. Experimental results.

We test the PNT method and compare it with the standard Newton method, which refers to the method in [31] but (2.9) is solved using direct matrix inversions. These two methods use the same initialization and backtracking line search strategy. The setting of hyperparameters follows Algorithm 3.2, and we set $\tau=1.001$ and $\lambda_0=0.1$ in all the experiments. We also implement the generalized hybrid iterative method proposed in [12] (denoted by genHyb), which is also based on gen-GKB. The genHyb iteratively computes approximations to μ_{opt} and $\boldsymbol{x}_{opt} = \boldsymbol{x}(\mu_{opt})$, where μ_{opt} is the optimal Tikhonov regularization parameter, that is $\mu_{opt} = \min_{\mu>0}\|\boldsymbol{x}(\mu)-\boldsymbol{x}_{\text{true}}\|_2$; the kth approximate Lagrangian multiplier is $\lambda_k = 1/\mu_k$. All the experiments are performed on MATLAB R2023b. The codes are available at https://github.com/Machealb/InverProb_IterSolver.

All the inverse problems in the experiments are ill-posed and satisfy Assumption 1. We use three types of ill-posed inverse problems to test the proposed method. The characteristics of these problems are summarized in Table 1.

5.1. Small-scale problems.

We choose two small-scale one-dimensional (1D) inverse problems from [24]. The first problem is heat, an inverse heat equation

Table 1
Properties of the inverse problems in the experiments.

Problem	$m \times n$	Ill-posedness	Description
heat	2000×2000	moderate	inverse heat equation
shaw	3000×3000	severe	1D image restoration
PRblurshake	$128^2 \times 128^2$	mild	2D image deblurring
PRblurspeckle	$128^2 \times 128^2$	mild	2D image deblurring
PRspherical	23168×128^2	mild	computed tomography

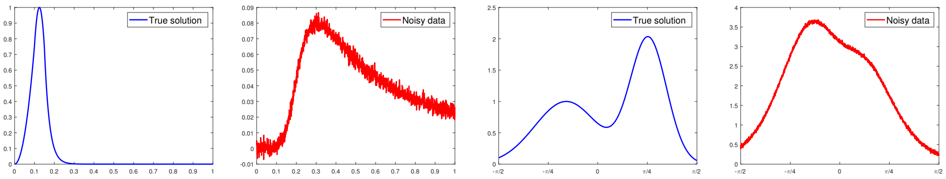

Fig. 1. *True solution and noisy observed data. The first two:* heat. *The last two:* shaw.

described by the Volterra integral equation of the first kind on $[0, 1]$. The second problem is shaw, a 1D image restoration model described by the Fredholm integral equation of the first kind on $[-\pi/2, \pi/2]$. We use the code in [24] to discretize the two problems to generate $\boldsymbol{A}$, $\boldsymbol{x}_{\text{true}}$ and $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$, where $m = n = 2000$ and $m = n = 3000$ for heat and shaw, respectively. We set the noisy observation vector $\boldsymbol{b}$ as $\boldsymbol{b} = \boldsymbol{b}_{\text{true}} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a Gaussian noise. For heat, we set $\boldsymbol{\epsilon}$ as a white noise (i.e., $\boldsymbol{M}$ is a scalar matrix) with noise level $\varepsilon := \|\boldsymbol{\epsilon}\|_2/\|\boldsymbol{b}_{\text{true}}\|_2 = 5 \times 10^{-2}$; for shaw, we set $\boldsymbol{\epsilon}$ as a uncorrelated nonwhite noise (i.e., $\boldsymbol{M}$ is a diagonal matrix) with noise level $\varepsilon = 10^{-2}$. The true solutions and noisy observed data for these two problems are shown in Figure 1.

For heat, we assume a Gaussian prior $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \mu^{-1}\boldsymbol{N})$ with $\boldsymbol{N}$ coming from the Gaussian kernel κ_G, i.e., the ij element of $\boldsymbol{N}$ is $[\boldsymbol{N}]_{ij} = K_G(r_{ij})$, $K_G(r) := \exp\left(-r^2/(2l^2)\right)$, where $r_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2$ and $\{\boldsymbol{p}_i\}_{i=1}^n$ are discretized points of the domain of the true solution; the parameter l is set as $l = 0.1$. For shaw, we construct $\boldsymbol{N}$ using the exponential kernel $K_{exp}(r) := \exp\left(-(r/l)^\nu\right)$, where the parameters l and ν are set as $l = 0.1$ and $\nu = 1$. We set $\tau = 1.001$ for both the two problems. We factorize $\boldsymbol{M}^{-1}$ and $\boldsymbol{N}^{-1}$ to form (1.3) and solve it directly to find μ_{opt} and $\boldsymbol{x}_{opt}$; the corresponding Lagrangian multiplier is $\lambda_{opt} = 1/\mu_{opt}$. We also compute the μ of (2.1) and the corresponding regularized solution, which is denoted by μ_{DP} and $\boldsymbol{x}_{DP}$, respectively. Therefore, the solution to (2.2) and (2.3) is $(\boldsymbol{x}^*, \lambda^*) = (\boldsymbol{x}_{DP}, 1/\mu_{DP})$. We use the optimal Tikhonov solution and the DP solution as the baseline for the subsequent tests.

For these two small-scale problems, we also implement the projected Newton method in [13] based on the transformation (1.3) as a comparison. This means that we solve

$$\min_{\bar{\boldsymbol{x}} \in \mathbb{R}^n} \{\|(\boldsymbol{L}_M \boldsymbol{A} \boldsymbol{L}_N)\bar{\boldsymbol{x}} - \boldsymbol{L}_M \boldsymbol{b}\|_2^2 + \mu\|\bar{\boldsymbol{x}}\|_2^2\},$$

using the method in [13] and then compute the regularized solution $\boldsymbol{x}_k = \boldsymbol{L}_N^{-1}\bar{\boldsymbol{x}}_k$. This Cholesky factorization based method is abbreviated as Ch-PNT.

We compare the convergence behaviors of PNT, Ch-PNT, Newton, and genHyb methods by plotting the relative error curve of $\boldsymbol{x}_k$ with respect to $\boldsymbol{x}_{\text{true}}$ and the convergence curves of λ_k and merit functions. The solutions $(\boldsymbol{x}_{DP}, 1/\mu_{DP})$ and $(\boldsymbol{x}_{opt}, 1/\mu_{opt})$

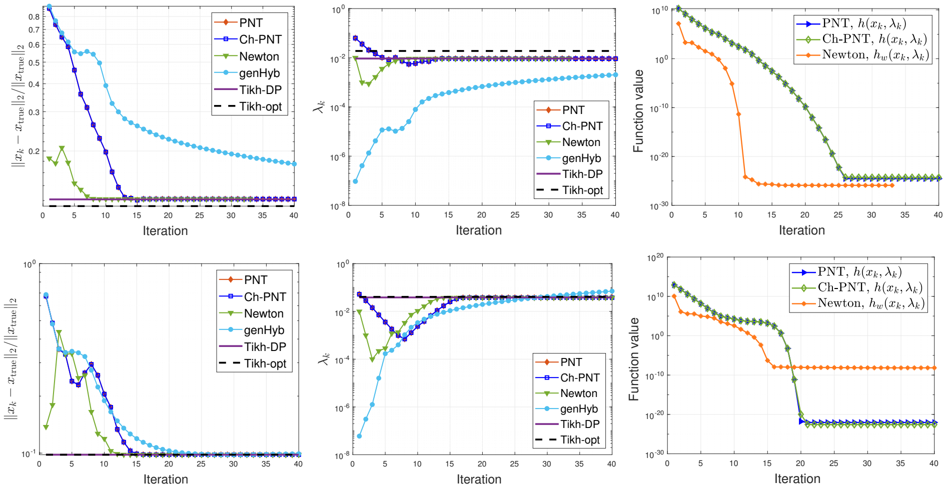

FIG. 2. *Relative errors of iterative solutions, convergence of λ_k, and convergence of merit functions. Top:* heat. *Bottom:* shaw.

are used as baselines. From Figure 2 we find that both PNT and Newton methods converge very fast to $\boldsymbol{x}_{DP}$ and $\lambda_{DP} := 1/\mu_{DP}$ with very few iterations, and PNT converges only slightly slower than Newton. We also find that the convergence behaviors of PNT and Ch-PNT are almost identical. This is not surprising, as both methods utilize the same subspaces for projecting the large-scale system and employ the same hyperparameters and update procedures. For heat, the error of the DP solution is slightly higher than the optimal Tikhonov solution, because DP slightly underestimates λ. The merit functions of both PNT and Newton decrease monotonically, and $h(\boldsymbol{x}_k, \lambda_k)$ of PNT eventually decreases to an extremely small value for the two problems. We remark that we set $w = 1$ for $h_w(\boldsymbol{x}, \lambda)$ in all the tests. For Newton method for heat, we stop the iterate at $k = 34$ because the step-length γ_k is too small. In comparison, genHyb converges much slower than the previous two methods, especially for heat.

Figure 3 plots the recovered solutions computed by PNT and genHyb methods at the final iterations; the solution by Newton is almost the same as that by PNT, thereby we omit it. We also plot the optimal Tikhonov regularized solution as a comparison, where the DP solution is very similar and omitted. Both PNT and genHyb can recover good regularized solutions, and PNT is slightly better for heat.

To further demonstrate the performance of PNT, we present the variation of the condition number of $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ during the iteration of PNT in Figure 4. This condition number is denoted by $\kappa(J^{(k)})$ in the two pictures. We observe that the condition number does not increase significantly during the iteration. This ensures that the small-scale linear system (3.12a) can be solved directly via matrix inversions without any issues.

To show the advantage of the computational efficiency of PNT over Ch-PNT and Newton, we gradually increase the scale of the test problems and measure the running time of the three methods, where all of them stop at the first iteration such that $\left| \|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|^2_{\boldsymbol{M}^{-1}} - \tau m \right| \leq 10^{-8}$. The time data are listed in Table 2. We also compute the ratio of the running time, i.e., the value of Ch-PNT-time/PNT-time and Newton-time/PNT-time. For shaw, we find that all three methods stop with similar iteration

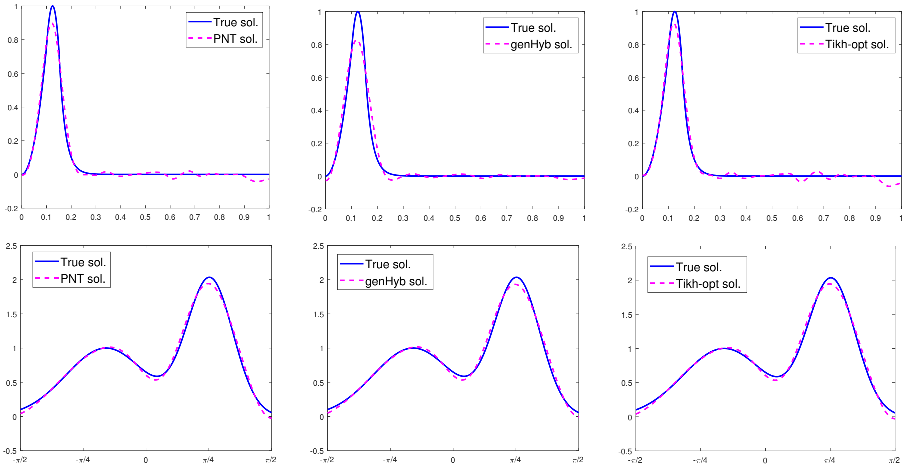

Fig. 3. *Comparison of reconstructed solutions at the final iterations with the optimal Tikhonov regularized solution. Top:* heat. *Bottom:* shaw.

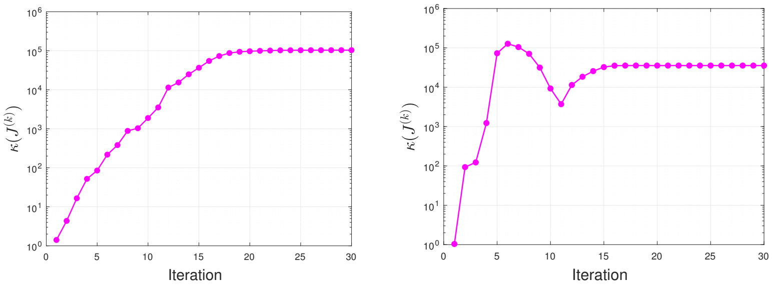

Fig. 4. *Variation of the condition number of* $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ *during the iteration of* PNT. *Left:* heat. *Right:* shaw.

Table 2

Running time (measured in seconds) of PNT, Ch-PNT, *and* Newton *methods as the scale of the problems increasing from* $n = 1000$ *to* $n = 5000$. *Both of the two methods stop at the first* k *(in parentheses) such that* $|\|\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}\|^2_{\boldsymbol{M}^{-1}} - \tau m| \leq 10^{-8}$. *The ratio of the running time between* PNT *and* Ch-PNT *is denoted as* ***ratio-1***, *while the ratio of the running time between* PNT *and* Newton *is denoted as* ***ratio-2***.

n	1000	2000	3000	4000	5000
			heat		
PNT	0.021(18)	0.114(21)	0.164 (19)	0.347 (19)	0.492 (19)
Ch-PNT	0.032 (18)	0.280 (21)	0.375 (19)	0.764 (19)	1.314 (19)
Newton	0.249 (10)	2.568 (11)	6.062 (10)	13.914 (10)	26.127 (11)
ratio-1	**1.5**	**2.5**	**2.3**	**2.2**	**2.7**
ratio-2	**11.9**	**22.5**	**37.0**	**40.1**	**53.1**
			shaw		
PNT	0.014 (17)	0.051 (16)	0.158 (17)	0.293 (18)	0.479 (19)
Ch-PNT	0.051 (17)	0.170 (16)	0.438 (19)	0.873 (18)	1.819 (19)
Newton	0.455 (16)	3.675 (15)	8.904 (14)	26.835 (16)	52.768 (16)
ratio-1	**3.6**	**3.3**	**2.8**	**3.0**	**3.8**
ratio-2	**32.5**	**72.1**	**56.4**	**91.6**	**110.2**

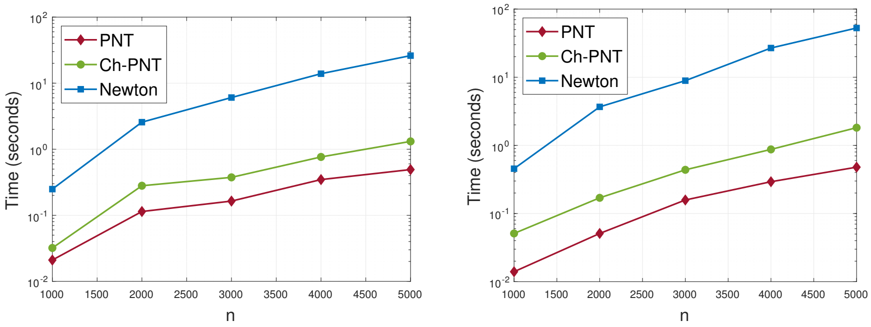

FIG. 5. *Comparison of scalability of* PNT*,* Ch-PNT*, and* Newton *methods as the scale of the problems increasing from* $n = 1000$ *to* $n = 5000$*. Left:* heat*. Right:* shaw*.*

numbers, and the computational speed of PNT is much faster than Newton, with the speedup ratio varying from 41 to 157. For heat, we find that Newton stops with only about half iteration numbers of PNT's. However, the total running time of PNT is still much smaller than Newton's, with the speedup ratio varying from 8 to 43. To compare the scalability of PNT, Ch-PNT and Newton more clearly, we use the data in Table 2 to plot the curve of time growth with respect to n in Figure 5. Clearly, PNT saves much more time compared to Newton while obtaining solutions with the same accuracy. Although the advantage of PNT over Ch-PNT is not significant for small-scale problems, Ch-PNT is not feasible for large-scale problems due to the prohibitive cost of Cholesky factorization.

5.2. Large-scale problems. We choose three two-dimensional (2D) image deblurring and computed tomography inverse problems from [18]. The first problem is PRblurshake, which simulates a spatially invariant motion blur caused by the shaking of a camera. The second problem is PRblurspeckle, which simulates a spatially invariant blur caused by atmospheric turbulence. The third problem is PRspherical that models spherical means tomography. The true images and noisy observed data are shown in Figure 6, where all the images have 128×128 pixels, and $\boldsymbol{\epsilon}$ are uncorrelated nonwhite Gaussian noises with $\varepsilon = 10^{-3}$, 5×10^{-3}, and 10^{-2}, respectively. We have $m = n = 128^2$ for the first two problems, and $m = 23168, n = 128^2$ for the third problem.

For PRblurshake and PRblurspeckle, we construct $\boldsymbol{N}$ using the Gaussian kernel with $l = 10$ and $l = 1$, respectively. For PRspherical, we construct $\boldsymbol{N}$ using the Matérn kernel

$$K_M(r) := \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} B_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right),$$

where $\Gamma(\cdot)$ is the gamma function, $B_{\nu}(\cdot)$ is the modified Bessel function of the second kind, and l and ν are two positive parameters of the covariance; we set $l = 100$ and $\nu = 1.5$. For the three large-scale problems, it is almost impossible to get $(\mu_{opt}, \boldsymbol{x}(\mu_{opt}))$ and $(\mu_{DP}, \boldsymbol{x}(\mu_{DP}))$ by solving (1.3). The standard Newton method and the methods in [13, 14] cannot be applied because these methods have to deal with $\boldsymbol{N}^{-1}$. To test the performance of PNT, here we only compare it with genHyb. Additionally, we also implement PNT-md to demonstrate that it can save some computation compared to PNT.

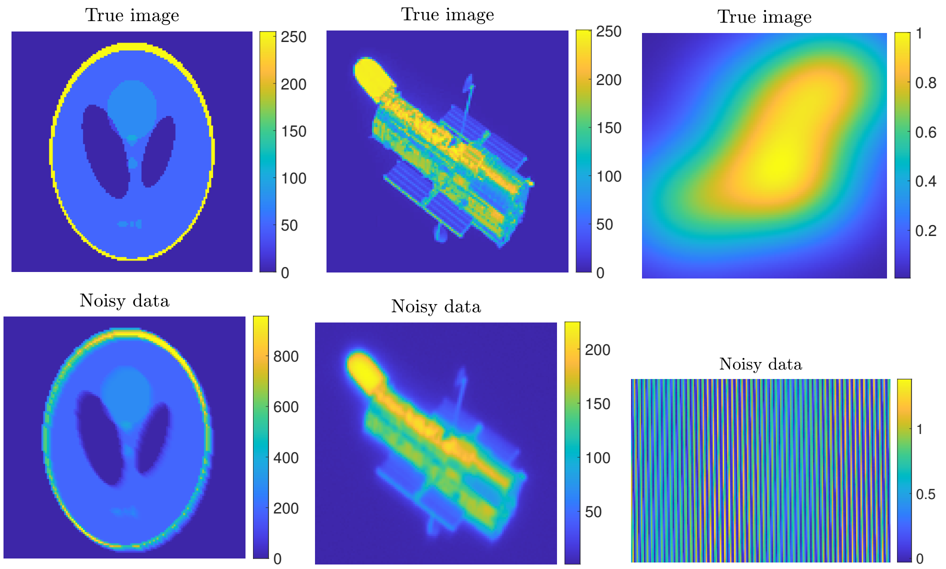

Fig. 6. *True solution and noisy observed data for deblurring and tomography problems. From the leftmost column to the rightmost column are* PRblurshake, PRblurspeckle, *and* PRspherical.

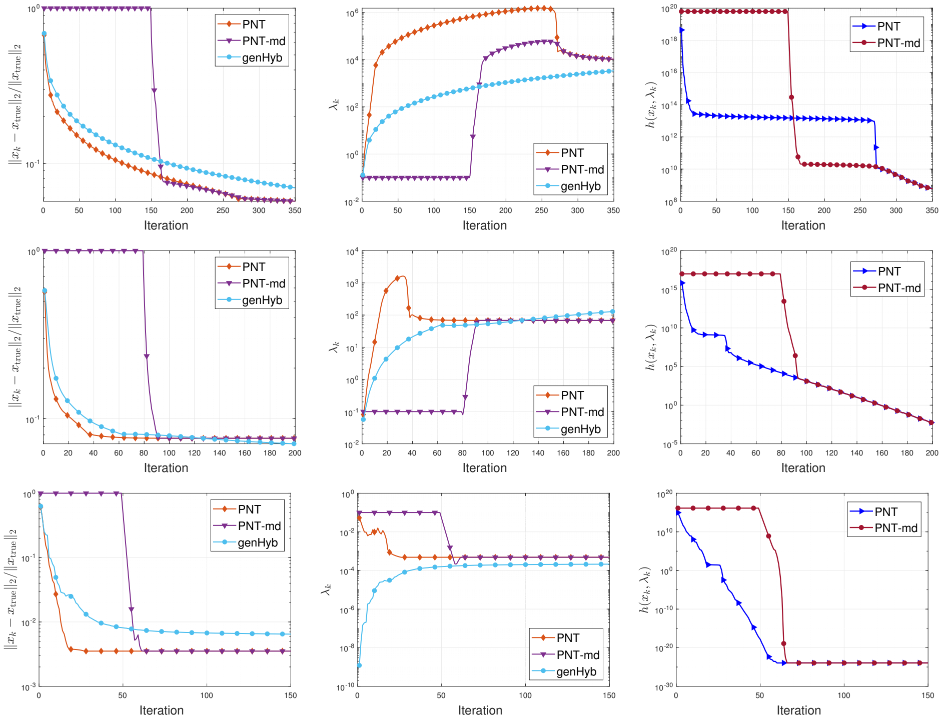

Fig. 7. *Relative errors of iterative solutions, convergence of λ_k, and convergence of merit functions. Top:* PRblurshake. *Middle:* PRblurspeckle. *Bottom:* PRspherical.

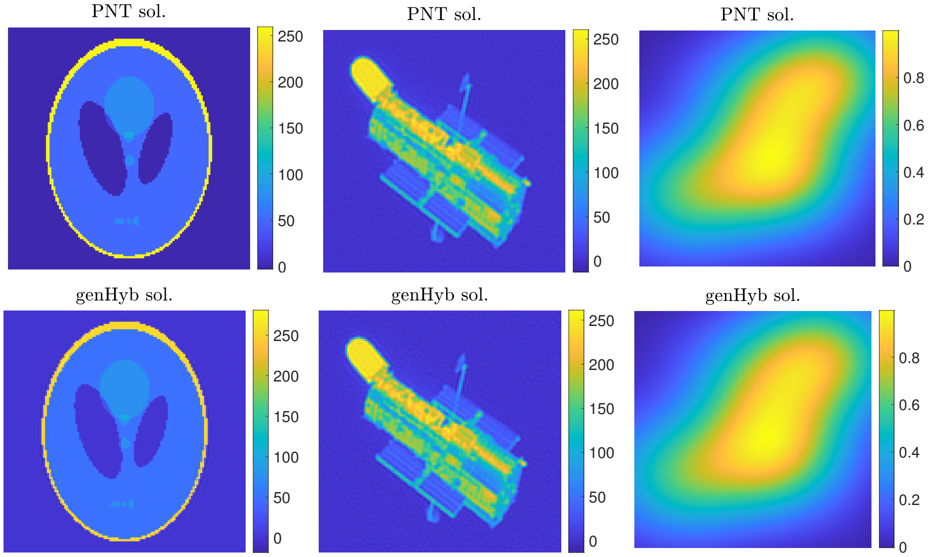

FIG. 8. *Reconstructed solutions at the final iterations by* PNT *and* genHyb*. From the leftmost column to the rightmost column are* PRblurshake*,* PRblurshake*, and* PRspherical*.*

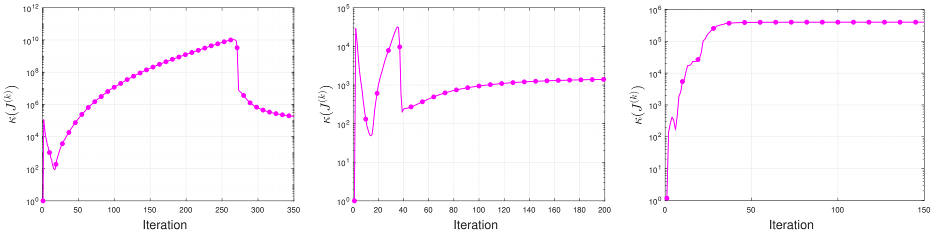

FIG. 9. *Variation of the condition number of* $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ *during the iteration of* PNT*. Left:* PRblurshake*. Middle:* PRblurspeckle*. Right:* PRspherical*.*

The relative error curves of the three methods, the convergence curves of λ_k and $h(\boldsymbol{x}_k, \lambda_k)$ are plotted in Figure 7. For PNT-md, we set $k_0 = 150, 80, 50$ for the three problems, respectively. It can be observed that PNT for the last two problems converges very fast: the variations of relative error and λ_k become quickly stabilized after 50 to 150 iterations, although for the second problem $h(\boldsymbol{x}_k, \lambda_k)$ are still decreasing significantly after 200 iterations. The genHyb method for PRblurshake and PRspherical converges slower, and it obtains two solutions with larger relative errors than that of PNT. This is because genHyb underestimates λ more than PNT. For all three problems, PNT-md converges very quickly from k_0, achieving solutions with the same accuracy as PNT while requiring nearly the same total number of iterations. The reconstructed images are shown in Figure 8, which reveals the effectiveness of PNT and genHyb. The variation of the condition number of $J^{(k)}(\bar{\boldsymbol{y}}_{k-1}, \lambda_{k-1})$ is shown in Figure 9. The condition number does not grow very large during the iteration, allowing the small-scale linear system (3.12a) to be solved directly without issues.

To further test the robustness of PNT and genHyb as the noise level gradually increases, we set the noise level of PRblurspeckle to be $\varepsilon = 5 \times 10^{-2}$, 10^{-1}, 5×10^{-1}. Figure 10 shows the corresponding relative error curves and the curves of $h(\boldsymbol{x}_k, \lambda_k)$.

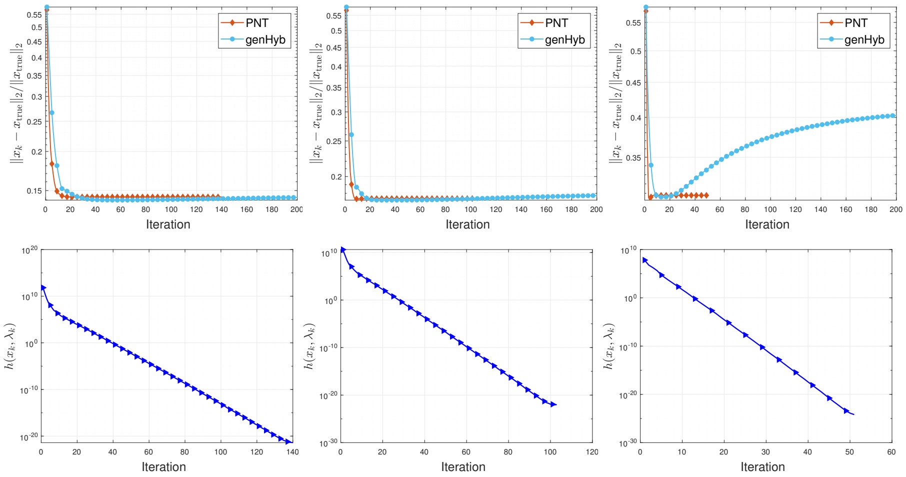

FIG. 10. *Relative errors of iterative solutions by* PNT *and* genHyb*, and the decrease of* $h(\boldsymbol{x}_k, \lambda_k)$*. The test problem is* PRblurspeckle*. From the leftmost column to the rightmost column, the noise levels are* $\varepsilon = 5 \times 10^{-2},\ 10^{-1},\ 5 \times 10^{-1}$.

We can find that, when the noise is not very big, both PNT and genHyb converge stably with almost the same accuracy. However, when the noise gradually increases, the situations are very different. First, we find that as the noise increases, PNT still converges stably, and faster. Second, $h(\boldsymbol{x}_k, \lambda_k)$ can always decrease to an extremely small value, which is promised by Theorem 4.1. The iterate of PNT stops when the step-length γ_k becomes too small (less than 10^{-16}), which happens more early if the noise is bigger. In comparison, the convergence of genHyb becomes unstable as the noise increases. For $\varepsilon = 10^{-1}$, it can be observed that the relative error for genHyb increases slightly after a certain iteration, while for $\varepsilon = 5 \times 10^{-1}$, the increase of relative error appears earlier and more clearly. This is a typical potential weakness of hybrid regularization methods, a challenge that the PNT method successfully addresses.

6. Conclusion.

For large-scale Bayesian linear inverse problems, we have proposed the projected Newton (PNT) method as a novel iterative approach for simultaneously updating both the regularization parameter and solution without any computationally expensive matrix inversions or decompositions. By reformulating the Tikhonov regularization as a corresponding constrained minimization problem and leveraging its Lagrangian function, the regularized solution and the corresponding Lagrangian multiplier can be obtained from the unconstrained Lagrangian function using a Newton-type method. To reduce the computational overhead of the Newton method, the generalized Golub–Kahan bidiagonalization is applied to project the original large-scale problem to become small-scale ones, where the projected Newton direction is obtained by solving the small-scale linear system at each iteration. We have proved that the projected Newton direction is a descent direction of a merit function, and the points generated by PNT eventually converge to the unique minimizer of this merit function, which is just the regularized solution and the corresponding Lagrangian multiplier.

Experimental tests on both small-scale and large-scale Bayesian inverse problems have demonstrated the excellent convergence property, robustness, and efficiency of PNT. The most demanding computational tasks in PNT are primarily matrix-vector products, making it particularly well-suited for large-scale problems.

An important remaining question is the convergence rate of PNT, i.e., how fast the three quantities $\|\boldsymbol{x}_k - \boldsymbol{x}_{DP}\|_2$, $|\lambda_k - \lambda_{DP}|$, and $h(\boldsymbol{x}_k, \lambda_k)$ converge to zero? The convergence rate may depend on several factors, including the ill-posedness of (1.1), the smoothness of the true solution, the noise level, and the properties of $\{\boldsymbol{A}, \boldsymbol{M}, \boldsymbol{N}\}$. We will conduct theoretical investigations into this issue in future work.

Acknowledgments. The author thanks Dr. Felipe Atenas for helpful discussions. The author also thanks the two anonymous referees and Editor Professor Defeng Sun for their detailed reading of the manuscript and many invaluable comments. In particular, the author thanks the first reviewer for suggesting the first approach for proving the uniqueness of λ^*, which is much simpler.

REFERENCES

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, *An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems*, IEEE Trans. Image Process., 20 (2010), pp. 681–695, https://doi.org/10.1109/TIP.2010.2076294.

[2] F. S. V. Bazán, *Fixed-point iterations in determining the Tikhonov regularization parameter*, Inverse Problems, 24 (2008), 035001, https://doi.org/10.1088/0266-5611/24/3/035001.

[3] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[4] J. Blowey, A. Craig, and T. Shardlow, *Frontiers in Numerical Analysis* 2002, Springer Science & Business Media, Durham, 2003.

[5] P. N. Brown and Y. Saad, *Convergence theory of nonlinear Newton–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330, https://doi.org/10.1137/0804017.

[6] T. Buzug, *Computed Tomography*, Springer, 2008, https://doi.org/10.1007/978-3-540-39408-2.

[7] D. Calvetti, F. Pitolli, J. Prezioso, E. Somersalo, and B. Vantaggi, *Priorconditioned CGLS-based quasi-MAP estimate, statistical stopping rule, and ranking of priors*, SIAM J. Sci. Comput., 39 (2017), pp. S477–S500, https://doi.org/10.1137/16M108272X.

[8] D. Calvetti, F. Pitolli, E. Somersalo, and B. Vantaggi, *Bayes meets Krylov: Statistically inspired preconditioners for CGLS*, SIAM Rev., 60 (2018), pp. 429–461, https://doi.org/10.1137/15M1055061.

[9] D. Calvetti and E. Somersalo, *Priorconditioners for linear systems*, Inverse Problems, 21 (2005), pp. 1397–1418, https://doi.org/10.1088/0266-5611/21/4/014.

[10] N. A. Caruso and P. Novati, *Convergence analysis of LSQR for compact operator equations*, Linear Algebra Appl., 583 (2019), pp. 146–164, https://doi.org/10.1016/j.laa.2019.08.024.

[11] J. Chung, J. G. Nagy, and D. P. O'Leary, *A weighted-GCV method for Lanczos-hybrid regularization.*, Electron. Trans. Numer. Anal., 28 (2008), pp. 149–167.

[12] J. Chung and A. K. Saibaba, *Generalized hybrid iterative methods for large-scale Bayesian inverse problems*, SIAM J. Sci. Comput., 39 (2017), pp. S24–S46, https://doi.org/10.1137/16M1081968.

[13] J. Cornelis, N. Schenkels, and W. Vanroose, *Projected Newton method for noise constrained Tikhonov regularization*, Inverse Problems, 36 (2020), 055002, https://doi.org/10.1088/1361-6420/ab7d2b.

[14] J. Cornelis and W. Vanroose, *Projected Newton method for noise constrained ℓ_p regularization*, Inverse Problems, 36 (2020), 125004, https://doi.org/10.1088/1361-6420/abb2fc.

[15] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457, https://doi.org/10.1002/cpa.20042.

[16] M. Fukushima, Z.-Q. Luo, and P. Tseng, *A sequential quadratically constrained quadratic programming method for differentiable convex minimization*, SIAM J. Optim., 13 (2003), pp. 1098–1119, https://doi.org/10.1137/S1052623401398120.

[17] A. Garbuno-Inigo, F. Hoffmann, W. Li, and A. M. Stuart, *Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler*, SIAM J. Appl. Dyn. Syst., 19 (2020), pp. 412–441, https://doi.org/10.1137/19M1251655.

[18] S. Gazzola, P. C. Hansen, and J. G. Nagy, *IR Tools: A MATLAB package of iterative regularization methods and large-scale test problems*, Numer. Algorithms, 81 (2019), pp. 773–811, https://doi.org/10.1007/s11075-018-0570-7.

[19] S. Gazzola, P. Novati, and M. R. Russo, *On Krylov projection methods and Tikhonov regularization*, Electron. Trans. Numer. Anal., 44 (2015), pp. 83–123.

[20] S. Gazzola and M. Sabaté Landman, *Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches*, GAMM-Mitt., 43 (2020), e202000017, https://doi.org/10.1002/gamm.202000017.

[21] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343, https://doi.org/10.1137/080725891.

[22] G. H. Golub and H. G. Wahba, *Generalized Cross-Validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223, https://doi.org/10.1080/00401706.1979.10489751.

[23] P. C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., 34 (1992), pp. 561–580, https://doi.org/10.1137/1034115.

[24] P. C. Hansen, *Regularization Tools version* 4.0 *for Matlab* 7.3, Numer. Algorithms, 46 (2007), pp. 189–194, https://doi.org/10.1007/s11075-007-9136-9.

[25] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006, https://doi.org/10.1137/1.9780898718874.

[26] G. Huang, L. Reichel, and F. Yin, *On the choice of subspace for large-scale Tikhonov regularization problems in general form*, Numer. Algorithms, 81 (2019), pp. 33–55, https://doi.org/10.1007/s11075-018-0534-y.

[27] Z. Jia and H. Li, *The joint bidiagonalization method for large GSVD computations in finite precision*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 382–407, https://doi.org/10.1137/22M1483608.

[28] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, 2006.

[29] M. E. Kilmer, P. C. Hansen, and M. I. Espanol, *A projection–based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330, https://doi.org/10.1137/050645592.

[30] K. Z. Kody Law and A. Stuart, *Data Assimilation: A Mathematical Introduction*, Springer, 2015, https://doi.org/10.1007/978-3-319-20325-6.

[31] G. Landi, *The Lagrange method for the regularization of discrete ill-posed problems*, Comput. Optim. Appl., 39 (2008), pp. 347–368, https://doi.org/10.1007/s10589-007-9059-3.

[32] H. Li, *Subspace Projection Regularization for Large-Scale Bayesian Linear Inverse Problems*, preprint, https://arxiv.org/abs/2310.18618, 2023.

[33] H. Li, *The joint bidiagonalization of a matrix pair with inaccurate inner iterations*, SIAM J. Matrix Anal. Appl., 45 (2024), pp. 232–259, https://doi.org/10.1137/22M1541083.

[34] H. Li, *A preconditioned Krylov subspace method for linear inverse problems with general-form Tikhonov regularization*, SIAM J. Sci. Comput., 46 (2024), pp. A2607–A2633, https://doi.org/10.1137/23M1593802.

[35] J. Liesen and Z. Strakos, *Krylov Subspace Methods: Principles and Analysis*, Numer. Math. Sci. Comput., Oxford University Press, Oxford, 2013, https://doi.org/10.1093/acprof:oso/9780199655410.001.0001.

[36] N. Luiken and T. Van Leeuwen, *Relaxed regularization for linear inverse problems*, SIAM J. Sci. Comput., 43 (2021), pp. S269–S292, https://doi.org/10.1137/20M1348091.

[37] J. L. Mead and R. A. Renaut, *A Newton root-finding algorithm for estimating the regularization parameter for solving ill-conditioned least squares problems*, Inverse Problems, 25 (2008), 025002, https://doi.org/10.1088/0266-5611/25/2/025002.

[38] F. Messerer, K. Baumgärtner, and M. Diehl, *Survey of sequential convex programming and generalized Gauss-Newton methods*, ESAIM: Proc. Surveys, 71 (2021), pp. 64–88, https://doi.org/10.1051/proc/202171107.

[39] S. Morigi, L. Reichel, and F. Sgallari, *Orthogonal projection regularization operators*, Numer. Algorithms, 44 (2007), pp. 99–114, https://doi.org/10.1007/s11075-007-9080-8.

[40] V. A. Morozov, *Regularization of incorrectly posed problems and the choice of regularization parameter*, USSR Comput. Math. Math. Phys., 6 (1966), pp. 242–251, https://doi.org/10.1016/0041-5553(66)90046-2.

[41] J. Nocedal and S. Wright, *Numerical Optimization, S*Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.

[42] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489, https://doi.org/10.1137/0902037.

[43] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489, https://doi.org/10.1137/040605412.

[44] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629, https://doi.org/10.1137/0712047.

[45] L. Reichel, F. Sgallari, and Q. Ye, *Tikhonov regularization based on generalized Krylov subspace methods*, Appl. Numer. Math., 62 (2012), pp. 1215–1228, https://doi.org/10.1016/j.apnum.2010.10.002.

[46] L. Reichel and A. Shyshkov, *A new zero-finder for Tikhonov regularization*, BIT, 48 (2008), pp. 627–643, https://doi.org/10.1007/s10543-008-0179-7.

[47] R. A. Renaut, A. W. Helmstetter, and S. Vatankhah, *Unbiased predictive risk estimation of the tikhonov regularization parameter: Convergence with increasing rank approximations of the singular value decomposition*, BIT, 59 (2019), pp. 1031–1061, https://doi.org/10.1007/s10543-019-00762-7.

[48] R. A. Renaut, S. Vatankhah, and V. E. Ardesta, *Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems*, SIAM J. Sci. Comput., 39 (2017), pp. B221–B243, https://doi.org/10.1137/15M1037925.

[49] M. Richter, *Inverse Problems: Basics, Theory and Applications in Geophysics*, Springer, 2016, pp. 77–155, https://doi.org/10.1007/978-3-319-48384-9_3.

[50] P. Rodriguez and B. Wohlberg, *An efficient algorithm for sparse representations with ℓ_p data fidelity term*, in Proceedings of 4th IEEE Andean Technical Conference (ANDESCON), Los Alamos National Laboratory (LANL), Los Alamos, NM, 2008, https://www.osti.gov/biblio/960812.

[51] P. Rodríguez and B. Wohlberg, *Efficient minimization method for a generalized total variation functional*, IEEE Trans. Image Process., 18 (2008), pp. 322–332, https://doi.org/10.1109/TIP.2008.2008420.

[52] A. K. Saibaba, J. Chung, and K. Petroske, *Efficient Krylov subspace methods for uncertainty quantification in large Bayesian linear inverse problems*, Numer. Linear Algebra Appl., 27 (2020), e2325, https://doi.org/10.1002/nla.2325.

[53] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559, https://doi.org/10.1017/S0962492910000061.

[54] W. Tian and X. Yuan, *Linearized primal-dual methods for linear inverse problems with total variation regularization and finite element discretization*, Inverse Problems, 32 (2016), 115011, https://doi.org/10.1088/0266-5611/32/11/115011.

[55] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83, https://doi.org/10.1137/0713009.

[56] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168, https://doi.org/10.1137/070703983.