PNKH-B: A PROJECTED NEWTON-KRYLOV METHOD FOR LARGE-SCALE BOUND-CONSTRAINED OPTIMIZATION[*]

KELVIN KAN[†], SAMY WU FUNG[‡], AND LARS RUTHOTTO[§]

Abstract. We present PNKH-B, a projected Newton-Krylov method with a low-rank approximated Hessian metric for approximately solving large-scale optimization problems with bound constraints. PNKH-B is geared toward situations in which function and gradient evaluations are expensive, and the (approximate) Hessian is only available through matrix-vector products. This is commonly the case in large-scale parameter estimation, machine learning, and image processing. In each iteration, PNKH-B generates a low-rank approximation of the (approximate) Hessian using Lanczos tridiagonalization and then solves a quadratic projection problem to update the iterate. The key idea is to compute the projection with respect to the norm defined by the low-rank approximation. Hence, PNKH-B can be viewed as a projected variable metric method. We present an interior point method to solve the quadratic projection problem efficiently. Since the interior point method effectively exploits the low-rank structure, its computational cost only scales linearly with respect to the number of variables, and it only adds negligible computational time. We also experiment with variants of PNKH-B that incorporate estimates of the active set into the Hessian approximation. We prove the global convergence to a stationary point under standard assumptions. Using three numerical experiments motivated by parameter estimation, machine learning, and image reconstruction, we show that the consistent use of the Hessian metric in PNKH-B leads to fast convergence, particularly in the first few iterations. We provide our MATLAB implementation at https://github.com/EmoryMLIP/PNKH-B.

Key words. projected Newton-Krylov method, bound-constrained optimization, large-scale optimization

AMS subject classifications. 49M15, 90C06, 90C25, 90C26, 90C51

1. Introduction.

In this paper, we introduce PNKH-B, a projected Newton-Krylov method with a low-rank approximated Hessian metric, for approximately solving large-scale optimization problems with bound constraints such as

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{l} \le \mathbf{x} \le \mathbf{u}. \tag{1.1}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, the inequalities are applied component-wise, and the vectors $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n \cup \{\pm\infty\}$ with $\mathbf{l} \le \mathbf{u}$ define the box $C = [\mathbf{l}, \mathbf{u}]$. While our method also applies to small and medium scale problems, we focus in this paper on situations in which evaluating f and its gradient is computationally expensive and the (approximate) Hessian is only available through matrix-vector products. This is common in PDE-constrained optimization [9, 12, 14, 32, 36, 41, 44], image processing [6, 20, 21, 22, 39, 40, 42], neural networks [7, 15, 16, 17, 35], etc. PNKH-B aims to approximately solve the problem using only a few function and gradient evaluations and Hessian-vector products.

[*] Submitted to the editors DATE.

Funding: KK and LR are supported by the U.S. National Science Foundation (NSF) Grant No. DMS 1751636.
SWF is supported by AFOSR MURI FA9550-18-1-0502, AFOSR Grant No. FA9550-18-1-0167, and ONR Grant No. N00014-18-1-2527.

[†] Department of Mathematics, Emory University, USA (kelvin.kan@emory.edu).

[‡] Department of Mathematics, University of California, Los Angeles, CA, USA (swufung@math.ucla.edu).

[§] Departments of Mathematics and Computer Science, Emory University, USA (lruthotto@emory.edu).

As in other projected variable metric methods, the kth PNKH-B iteration reads

$$\mathbf{x}_{k+1} = \Pi_{\|\cdot\|_{\mathbf{H}_k}}\left(\mathbf{y}_{k+1}\right), \quad \text{with} \quad \mathbf{y}_{k+1} = \mathbf{x}_k - \mu_k \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k), \tag{1.2}$$

where $\mathbf{H}_k$ is a positive definite matrix, μ_k is a suitable step size, $\Pi_{\|\cdot\|_{\mathbf{H}_k}}$ is the projection operator onto the box C with the variable metric induced by $\mathbf{H}_k$, that is,

$$\Pi_{\|\cdot\|_{\mathbf{H}_k}}(\mathbf{y}) = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{z}-\mathbf{y}\|_{\mathbf{H}_k}^2 \quad \text{subject to} \quad \mathbf{l} \leq \mathbf{z} \leq \mathbf{u}. \tag{1.3}$$

Here, for a vector $\mathbf{v} \in \mathbb{R}^n$ the norm induced by the approximate Hessian is $\|\mathbf{v}\|_{\mathbf{H}_k} := \sqrt{\mathbf{v}^\top \mathbf{H}_k \mathbf{v}}$.

The projection problem (1.3) is a convex quadratic program, which has no closed-form solution in general. Exceptions are projected gradient methods that are obtained by using a diagonal Hessian approximation, where the projection simplifies to

$$\Pi_{\|\cdot\|_2}(\mathbf{y}) = \max\{\min\{\mathbf{y}, \mathbf{u}\}, \mathbf{l}\}. \tag{1.4}$$

While being simple and robust, the projected gradient method is not a method of choice in our setting due to the large number of iterations needed to obtain a reasonable solution [11]. Similarly, the convergence of other first-order methods such as AdaGrad [10], which uses a diagonal Hessian approximation, is also not fast enough for large-scale problems. On the other extreme using a projected Newton step (obtained using $\mathbf{H}_k = \nabla^2 f(\mathbf{x}_k)$) is intractable due to the computational cost of solving the large-scale quadratic program involved in the projection (1.3).

In this paper, we propose PNKH-B, an iterative method for large-scale bound-constrained optimization. PNKH-B uses the metric obtained from a low-rank approximation of the (approximate) Hessian consistently. Since the same metric is used in the computation of the search direction and projection, we prove the global convergence of our method under mild assumptions. A main contribution is an interior-point method for computing the projection step (1.3) that, by exploiting the low-rank structure of $\mathbf{H}_k$, only adds negligible computational costs compared to (1.4). Our method can be seen as a variable metric projected Newton-Krylov scheme since the Hessian changes at each iteration. Moreover, we use Lanczos tridiagonalization [26] to compute a basis of the Krylov subspace defined by the (approximate) Hessian and gradient at the kth step. Although not shown here, our method can be straightforwardly extended to other low-rank representations, e.g., arising in L-BFGS [5]. We demonstrate the benefits of using the low-rank Hessian metric using three numerical experiments that are motivated by PDE parameter estimation, machine learning, and image reconstruction.

The remainder of this paper is organized as follows. In Section 2, we give an overview of the related work. In Section 3, we describe our PNKH-B. In Section 4, we provide theoretical guarantees. In Section 5, we present three experimental results to illustrate the effectiveness of our method. We conclude with a discussion and future outlooks in Section 6.

2. Related Work.

Projected inexact Newton and quasi-Newton schemes are among the most effective and commonly used solvers for constrained large-scale optimization problems such as (1.1). They have been studied and applied extensively in the past four decades, e.g., [4, 5, 37, 38]. In this section, we give a brief overview of the schemes that bear similarities with our approach.

One property that can be used to group existing schemes is the metric used to determine the search direction and projection. We refer to schemes that use the same

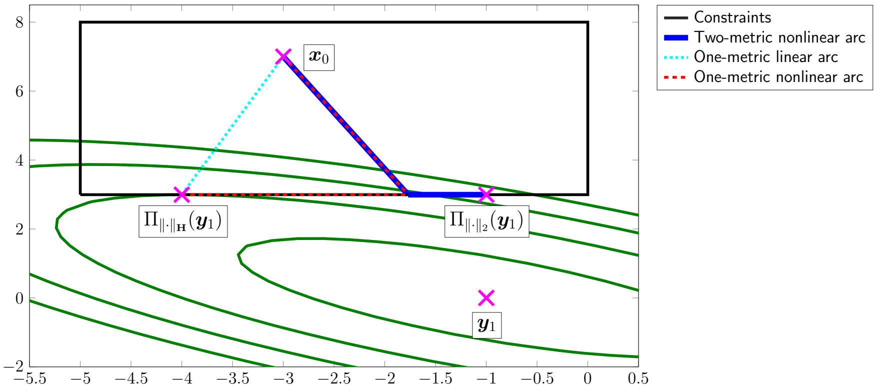

Fig. 1: An illustration of the first iterations of different methods on the quadratic optimization problem (2.1), where $\mathbf{H} = [1, 1; 1, 2]$, $\mathbf{b} = [1; 1]$, $\mathbf{l} = [-5; 3]$, $\mathbf{u} = [0; 8]$, $\mathbf{x}_0 = [-3; 7]$, $\mathbf{y}_1 = \mathbf{x}_0 - \mathbf{H}^{-1}\nabla f(\mathbf{x}_0) = -\mathbf{H}^{-1}\mathbf{b} = [-1; 0]$ is the updated variable before projection, $\Pi_{\|\cdot\|_{\mathbf{H}}}(\mathbf{y}_1) = [-4; 3]$ is the projection with the Hessian metric and is the optimal solution, and $\Pi_{\|\cdot\|_2}(\mathbf{y}_1) = [-1; 3]$ is the projection with the Euclidean metric. The linear/nonlinear line search arcs for one/two-metric methods are shown. The one-metric nonlinear arc, which is used in our proposed PNKH-B, is the best one as it searches along the boundary and gives the optimal solution. The one-metric linear arc is less natural, does not search along the boundary and gives suboptimal iterate whenever step size 1 is not used. Finally, the two-metric nonlinear arc searches for the opposite direction of the one-metric nonlinear arc. It gives a suboptimal iterate $\Pi_{\|\cdot\|_2}(\mathbf{y}_1)$ and it will be stuck at $\Pi_{\|\cdot\|_2}(\mathbf{y}_1)$ even when the exact Hessian is used, i.e., it generates $\Pi_{\|\cdot\|_2}(\mathbf{y}_k) = \Pi_{\|\cdot\|_2}(\mathbf{y}_1)$ for all $k \geq 2$.

metric for both steps (e.g., (1.2)) as one-metric schemes and schemes that use different metrics for the two steps as two-metric schemes. Another distinguishing feature is the order in which projections and line searches are performed. Schemes that project each line search iterate (e.g., (1.2)) in general lead to a nonlinear arc, while applying the projection only once before the line search yields a linear arc. In the following, we will review existing methods according to these choices and provide an example to highlight their differences.

An extensively studied two-metric scheme [4, 13] uses a search direction induced by an approximated Hessian norm and a projection with respect to the Euclidean metric. That is, its formulation in each iteration is given by (1.2) and (1.3) except that the projection uses the Euclidean metric as this provides a closed-form to the projection problem using (1.4). Its line search induces a nonlinear arc, see Figure 1. However, its convergence is not guaranteed in general, see Example 1, [4] and [24, Chapter 5.5.1]. The global convergence of this approach for convex problems with linear constraints was proven by [4, 13] when a variable partitioning scheme is used. It partitions the components of the kth iterate into an active set in which the components are at or close to the boundary of the feasible set and an inactive set in which the components are in the interior of the feasible set. A search direction induced by the Euclidean norm is used for the active components and a search direction induced by $\|\cdot\|_{\mathbf{H}_k}$ is used for the inactive components. They also prove local superlinear convergence under certain conditions. Since then, variable partitioning has become a recurring theme for two-metric schemes [14, 19, 25, 27, 28, 29, 38]. Although the projection of the convergent two-metric scheme (1.4) can be computed immediately, it requires appropriate scaling for the Euclidean norm induced search direction before combining the two search directions. Moreover, when many constraints are active, two-metric schemes essentially become projected gradient methods. A specific drawback in Newton-Krylov schemes for large-scale problems is that the partitioning of the variables complicates the design of effective preconditioners. Given a preconditioner $\mathbf{M}_k$ for the approximate Hessian $\mathbf{H}_k$ and $\mathbf{P}_k$ the projection operator onto the inactive set at the kth step, the most natural choice is to precondition $\mathbf{P}_k^\top \mathbf{H}_k \mathbf{P}_k$ by $\mathbf{P}_k^\top \mathbf{M}_k^{-1} \mathbf{P}_k$ since it is intractable to compute $(\mathbf{P}_k^\top \mathbf{M}_k^{-1} \mathbf{P}_k)^{-1}$. However, $(\mathbf{P}_k^\top \mathbf{M}_k^{-1} \mathbf{P}_k)^{-1} \neq \mathbf{P}_k^\top \mathbf{M}_k \mathbf{P}_k$ in general.

Another well-studied approach is the one-metric scheme with linear arc. It is generally performed as follows. At the kth iteration, it (approximately) solves (1.2) with $\mu_k = 1$ to obtain a projection. Then it performs a line search along the straight line connecting the current iterate $\mathbf{x}_k$ and the projection; this linear line search is done in order to limit the number of solving costly projections. This scheme is studied with different approximations of the Hessian, solvers for the projection (1.2) with $\mu_k = 1$ or backtracking schemes to determine the next iterate $\mathbf{x}_{k+1}$. For instance, the widely-applied L-BFGS-B [5, 33] uses the limited-memory BFGS matrix for $\mathbf{H}_k$ and approximately solves the projection (1.2) with $\mu_k = 1$ without any constraints, then it truncates the path toward the solution in order to satisfy the constraints. Finally it backtracks along the straight line to obtain $\mathbf{x}_{k+1}$. Other variants of this one-metric method with linear arc include [2, 3, 18, 31, 37]. Although the consistent choice of metric could generate a better update direction than the two-metric scheme, it results in a suboptimal iterate which does not lie in the boundary whenever a step size of 1 is not used. Also, because the line search is just simply along the straight line connecting the previous iterate $\mathbf{x}_k$ and the projection, it can result in an inferior iterate $\mathbf{x}_{k+1}$ when compared to a more natural line search along a nonlinear arc used in our method, see Figure 1.

Example 1. *We illustrate the differences between one-metric and two-metric schemes with linear and nonlinear arcs, respectively, using a two-dimensional quadratic program*

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{b}^\top \mathbf{x} \quad \textit{subject to} \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \tag{2.1}$$

The first iteration before projection of the one-metric method with nonlinear arc reads

$$\mathbf{y}_1(\mu) = \mathbf{x}_0 - \mu \mathbf{H}^{-1}\nabla f(\mathbf{x}_0) = (1-\mu)\mathbf{x}_0 - \mu \mathbf{H}^{-1}\mathbf{b},$$

where μ is a step size determined by a backtracking line search scheme. The projection with the Hessian metric is given by

$$\Pi_{\|\cdot\|_{\mathbf{H}}}(\mathbf{y}_1(\mu)) = \operatorname*{arg\,min}_{\mathbf{l}\leq\mathbf{z}\leq\mathbf{u}} \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} - (1-\mu)\mathbf{z}^\top \mathbf{H}\mathbf{x}_0 + \mu \mathbf{b}^\top \mathbf{z}. \tag{2.2}$$

When $\mu = 1$, i.e., the first step of the backtracking line search, the projection problem is equivalent to the original optimization problem. So the backtracking line search stops at the first step and the one-metric method with the nonlinear line search converges in one iteration. This is because the Hessian metric projection is consistent with the steepest descent direction $\mathbf{H}^{-1}\nabla f$ induced by the Hessian metric. If for a non-quadratic objective function, the initial step size is not accepted, then the nonlinear and *linear arc lead to different iterates; see Figure 1. Solving the projection problem with the Euclidean metric leads to a suboptimal projection at which the scheme stagnates in the absence of any of the remedies outlined above.*

3. PNKH-B.

In this section, we introduce our PNKH-B. In Subsection 3.1, we present an outline of the algorithm. At each iteration, each backtracking line search requires computing a projection, which is a quadratic program. In Subsection 3.2, we present the derivation and implementation of an interior point method to solve the quadratic program effectively. In Subsection 3.3, we present two variants of our PNKH-B which incorporate the current estimates of the active set.

3.1. Outline of PNKH-B.

Our projected Newton-Krylov method with a low-rank approximated Hessian metric (PNKH-B) is a one-metric method that approximately solves the bound-constrained optimization problem (1.1). At each iteration PNKH-B is given by (1.2) and (1.3).

The global convergence of PNKH-B is guaranteed under standard assumptions; see Section 4. We set $\mathbf{H}_k$ as a low-rank approximation of the (approximate) Hessian at $\mathbf{x}_k$ generated by Lanczos tridiagonalization [26]. Specifically, the Krylov subspace is defined by the (approximate) Hessian and gradient at $\mathbf{x}_k$. The low-rank approximation is given by $\mathbf{H}_k = \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top$, where $\mathbf{V}_k \in \mathbb{R}^{n\times l}$ has orthonormal columns, $\mathbf{T}_k \in \mathbb{R}^{l\times l}$ is tridiagonal and l is the rank of the low-rank approximation. We slightly abuse notation and denote the pseudoinverse $\mathbf{V}_k\mathbf{T}_k^{-1}\mathbf{V}_k^\top$ by $\mathbf{H}_k^{-1}$ in order to be consistent with the conventional notation used in Newton's method. The matrix in the norm $\|\cdot\|_{\mathbf{H}_k}$ is shifted by $c\mathbf{I}$ with c being a small scalar to render it positive definite, and hence the norm is well-defined. Lanczos tridiagonalization is suitable for large-scale problems because is does not require the explicit (approximate) Hessian $\mathbf{G}_k$, but only the function $g_k : \mathbf{y} \mapsto \mathbf{G}_k\mathbf{y}$. Using the low-rank approximation, we effectively compute the pseudoinverse $\mathbf{H}_k^{-1}$ and the projection $\Pi_{\|\cdot\|_{\mathbf{H}_k}}(\cdot)$, which has to be done once for each line search. The projection problem is solved using an interior point method, which exploits the low-rank approximation effectively and scales only linearly with the number of variables. The interior-point method will be discussed in Subsection 3.2. The outline of our PNKH-B is summarized in Algorithm 3.2.

3.2. Interior Point Method.

In this section, we present the derivation and effective implementation of the interior point method tailored to exploit the low-rank structure in (1.3).

Derivation. We use a standard primal-dual interior point method to solve the projection problem (1.3), which we derive following the outline in [34, Chapter 16.6]. To obtain $\mathbf{x}_{k+1}$, we re-formulate (1.3) as

$$\min_{\mathbf{z},\mathbf{w}} \frac{1}{2}\mathbf{z}^\top\mathbf{H}_k\mathbf{z} - \mathbf{z}^\top\mathbf{H}_k\mathbf{y}_{k+1} \quad \text{subject to} \quad \mathbf{K}\mathbf{z} - \mathbf{b} = \mathbf{w} \text{ and } \mathbf{w} \geq \mathbf{0},$$

where $\mathbf{w} \in \mathbb{R}^{2n}$ is a slack vector and

$$\mathbf{K} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{l} \\ -\mathbf{u} \end{bmatrix}.$$

From this, we obtain the KKT conditions

$$\mathbf{H}_k\mathbf{z} - \mathbf{H}_k\mathbf{y}_{k+1} - \mathbf{K}^\top\boldsymbol{\lambda} = \mathbf{0}, \tag{3.1a}$$

$$\mathbf{K}\mathbf{z} - \mathbf{b} - \mathbf{w} = \mathbf{0}, \tag{3.1b}$$

$$w_i\lambda_i = 0, \;\text{ for } i = 1, ..., 2n, \tag{3.1c}$$

$$\mathbf{w} \geq \mathbf{0}\;,\; \boldsymbol{\lambda} \geq \mathbf{0}, \tag{3.1d}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{2n}$ is a vector of Lagrange multipliers. Since the problem (1.3) is convex and the interior of the feasible set is non-empty, Slater's condition is satisfied and hence the KKT conditions are necessary and sufficient. As usual in interior point methods, we consider the perturbed KKT conditions

$$F(\mathbf{z}, \mathbf{w}, \boldsymbol{\lambda}; \sigma, \xi) = \begin{bmatrix} \mathbf{H}_k\mathbf{z} - \mathbf{H}_k\mathbf{y}_{k+1} - \mathbf{K}^\top\boldsymbol{\lambda} \\ \mathbf{K}\mathbf{z} - \mathbf{b} - \mathbf{w} \\ \mathbf{W}\boldsymbol{\Lambda}\mathbf{e} - \sigma\xi\mathbf{e} \end{bmatrix} = \mathbf{0}, \tag{3.2}$$

where $\mathbf{W} = \operatorname{diag}(\mathbf{w})$, $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})$, $\mathbf{e} \in \mathbb{R}^{2n}$ is a vector of ones, $\sigma \in [0, 1]$ and $\xi = \mathbf{w}^\top\boldsymbol{\lambda}/(2n)$ is the duality measure. The solutions of (3.2) define the central path and tend to the solution of (3.1) [34, Section 16.6].

We then apply Newton's method to find the root of the system (3.2). At the jth iteration of Newton's method, the step is obtained by solving

$$\begin{bmatrix} \mathbf{H}_k & 0 & -\mathbf{K}^\top \\ \mathbf{K} & -\mathbf{I} & 0 \\ 0 & \boldsymbol{\Lambda}_j & \mathbf{W}_j \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z}_j \\ \Delta\mathbf{w}_j \\ \Delta\boldsymbol{\lambda}_j \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_j \\ -\mathbf{v}_j \\ -\mathbf{W}_j\boldsymbol{\Lambda}_j\mathbf{e} + \sigma\xi_j\mathbf{e} \end{bmatrix}, \tag{3.3}$$

which is obtained by differentiating F in (3.2) and setting

$$\begin{aligned} \mathbf{r}_j &= \mathbf{H}_k\mathbf{z}_j - \mathbf{H}_k\mathbf{y}_{k+1} - \mathbf{K}^\top\boldsymbol{\lambda}_j, \\ \mathbf{v}_j &= \mathbf{K}\mathbf{z}_j - \mathbf{b} - \mathbf{w}_j. \end{aligned}$$

Here, $\mathbf{r}_j$ and $\mathbf{v}_j$ are the dual and primal residuals, respectively. After computing $\Delta\mathbf{z}_j$, $\Delta\mathbf{w}_j$ and $\Delta\boldsymbol{\lambda}_j$, the update of the interior point method is

$$(\mathbf{z}_{j+1}, \mathbf{w}_{j+1}, \boldsymbol{\lambda}_{j+1}) = (\mathbf{z}_j, \mathbf{w}_j, \boldsymbol{\lambda}_j) + \beta_j(\Delta\mathbf{z}_j, \Delta\mathbf{w}_j, \Delta\boldsymbol{\lambda}_j),$$

where $\beta_j = \min(\beta_j^{\text{pri}}, \beta_j^{\text{dual}})$ and

$$\begin{aligned} \beta_j^{\text{pri}} &= \max\{\beta \in (0, 1] : \mathbf{w}_j + \beta\Delta\mathbf{w}_j \geq (1 - \tau)\mathbf{w}_j\}, \\ \beta_j^{\text{dual}} &= \max\{\beta \in (0, 1] : \boldsymbol{\lambda}_j + \beta\Delta\boldsymbol{\lambda}_j \geq (1 - \tau)\boldsymbol{\lambda}_j\}. \end{aligned}$$

The parameter $\tau \in (0, 1]$ controls the distance to the boundary of the feasible set. While there are other schemes to determine the step size (see, e.g., [8]), this simple choice has been effective in our experiments.

Efficient Implementation. The most crucial step of the interior point method is the computation of the solution of the step in (3.3). Our implementation exploits the low-rank structure of $\mathbf{H}_k$ to directly solve the linear system with $\mathcal{O}(nl^2)$ floating point operations, where l is the rank of the low-rank approximation; see Algorithm 3.1 for an overview.

To compute the update, we first multiply the third equation of (3.3) by $\mathbf{\Lambda}_j^{-1}$ and add it to the second equation of (3.3) and obtain

$$\begin{bmatrix} \mathbf{H}_k & -\mathbf{K}^\top \\ \mathbf{K} & \mathbf{\Lambda}_j^{-1}\mathbf{W}_j \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z}_j \\ \Delta \boldsymbol{\lambda}_j \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_j \\ -\mathbf{v}_j - \mathbf{w}_j + \sigma \xi_j \mathbf{\Lambda}_j^{-1}\mathbf{e} \end{bmatrix}. \tag{3.4}$$

Multiplying the second equation of (3.4) by $\mathbf{K}^\top \mathbf{W}_j^{-1}\mathbf{\Lambda}_j$ and adding it to the first equation, we obtain

$$(\mathbf{H}_k + \mathbf{K}^\top \mathbf{W}_j^{-1}\mathbf{\Lambda}_j \mathbf{K})\Delta \mathbf{z}_j = -\mathbf{r}_j + \mathbf{K}^\top \mathbf{p}_j, \tag{3.5}$$

where $\mathbf{p}_j = \mathbf{W}_j^{-1}\mathbf{\Lambda}_j(-\mathbf{v}_j - \mathbf{w}_j + \mathbf{\Lambda}_j^{-1}\sigma\xi_j\mathbf{e})$. In our implementation, we shift $\mathbf{H}_k$ in (3.5) by adding $c\mathbf{I}$ to make the projection norm well-defined. Here, $c > 0$ is a small scalar. We note that the shift is only applied to $\mathbf{H}_k$ in (3.5), i.e. the matrix of the norm $\|\cdot\|_{\mathbf{H}_k}$, but not the matrix of the search direction $\mathbf{H}_k^{-1}\nabla f(\mathbf{x}_k)$. This is because the inverse of the shift c^{-1} is large and will dominate the search direction. The right hand side of (3.5) can be computed explicitly in $\mathcal{O}(n)$ operations. Defining $\mathbf{E}_j = c\mathbf{I} + \mathbf{K}^\top \mathbf{W}_j^{-1}\mathbf{\Lambda}_j\mathbf{K}$ and noticing that $\mathbf{E}_j$ is diagonal and invertible, we can use the Woodbury matrix identity to invert the left hand side of (3.5). Specifically

$$\begin{aligned} (\mathbf{H}_k + \mathbf{E}_j)^{-1} &= (\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top + \mathbf{E}_j)^{-1} \\ &= \mathbf{E}_j^{-1} - \mathbf{E}_j^{-1}\mathbf{V}_k(\mathbf{T}_k^{-1} + \mathbf{V}_k^\top \mathbf{E}_j^{-1}\mathbf{V}_k)^{-1}\mathbf{V}_k^\top \mathbf{E}_j^{-1} \\ &= \mathbf{E}_j^{-1} - \underbrace{\mathbf{E}_j^{-1}\mathbf{B}_k}_{\in\mathbb{R}^{n\times l}}\underbrace{(\mathbf{I} + \mathbf{B}_k^\top \mathbf{E}_j^{-1}\mathbf{B}_k)^{-1}}_{\in\mathbb{R}^{l\times l}}\underbrace{\mathbf{B}_k^\top \mathbf{E}_j^{-1}}_{\in\mathbb{R}^{l\times n}}, \end{aligned} \tag{3.6}$$

where $\mathbf{T}_k = \mathbf{R}_k^\top \mathbf{R}_k$ is the Cholesky factorization of $\mathbf{T}_k$, $\mathbf{B}_k = \mathbf{V}_k\mathbf{R}_k^\top$, and l is the rank of the low-rank approximation. From (3.6), we see that it requires $\mathcal{O}(nl^2)$ flops to compute the solution $\Delta\mathbf{z}_j$ of (3.5).

After obtaining $\Delta\mathbf{z}_j$, we substitute $\Delta\mathbf{z}_j$ into (3.3) and (3.4) and obtain

$$\Delta\boldsymbol{\lambda}_j = \mathbf{p}_j - \mathbf{W}_j^{-1}\mathbf{\Lambda}_j\mathbf{K}\Delta\mathbf{z}_j, \quad \text{and} \quad \Delta\mathbf{w}_j = \mathbf{K}\Delta\mathbf{z}_j + \mathbf{v}_j,$$

whose computation require $\mathcal{O}(n)$ flops.

Overall, exploiting the fact that $\mathbf{H}_k$ is a rank-l approximation of the Hessian, the interior point method requires $\mathcal{O}(nl^2)$ flops per iteration, where n is the number of variables.

3.3. Incorporating Estimates of the Active Set.

We introduce two variants of PNKH-B that seek to accelerate the convergence by using estimates of the active set. The intuitive idea is to ignore coordinate dimensions associated with constraints that are currently active during the construction of the low-rank approximation $\mathbf{H}_k$. To this end, we partition $\mathbf{x}_k$ into active and inactive components. To update the inactive coordinates, we exploit curvature information, and to update those active coordinates, we use a scaled projected gradient descent step. Our procedure and estimation of the active coordinates are essentially the same as in the two-metric schemes [4, 14, 38], which crucially rely on this step to ensure convergence. Being a one-metric scheme, the convergence theory of PNKH-B applies both with and without partitioning. However, in practice it can be advantageous to use estimates of the active set.

At the kth iteration, let $\mathcal{A}_k \subset \{1, 2, \ldots, n\}$ contain the indices of the components that are estimated to be active and let $m = |\mathcal{A}_k|$. We denote with $\mathbf{R}_k \in \mathbb{R}^{m\times n}$ and

Algorithm 3.1 Interior Point Method for the Projection Problem (1.3)

```
Inputs: low-rank approximation $\mathbf{H}_k = \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top \approx \nabla^2 f(\mathbf{x}_k)$, point to be projected $\mathbf{y}_{k+1}$, initial guess $\mathbf{z}_0 \in \mathbb{R}^n$, $\boldsymbol{\lambda}_0, \mathbf{w}_0 \in \mathbb{R}^{2n}$, $\tau \in (0,1]$ and tol $> 0$
compute the Cholesky factorization of $\mathbf{T}_k = \mathbf{R}_k^\top\mathbf{R}_k$
compute $\mathbf{B}_k = \mathbf{V}_k\mathbf{R}_k^\top$
for $j = 0, 1, 2, \ldots$ do
  compute $\mathbf{r}_j = \mathbf{H}_k\mathbf{z}_j - \mathbf{H}_k\mathbf{y}_{k+1} - \mathbf{K}^\top\boldsymbol{\lambda}_j$
  compute $\mathbf{v}_j = \mathbf{K}\mathbf{z}_j - \mathbf{b} - \mathbf{w}_j$
  compute $\mathbf{p}_j = \mathbf{W}_j^{-1}\boldsymbol{\Lambda}_j(-\mathbf{v}_j - \mathbf{w}_j + \boldsymbol{\Lambda}_j^{-1}\sigma\xi_j\mathbf{e})$
  compute $\mathbf{E}_j = c\mathbf{I} + \mathbf{K}^\top\mathbf{W}_j^{-1}\boldsymbol{\Lambda}_j\mathbf{K}$
  compute $\Delta\mathbf{z}_j = \left(\mathbf{E}_j^{-1} - \mathbf{E}_j^{-1}\mathbf{B}_k(\mathbf{I} + \mathbf{B}_k^\top\mathbf{E}_j^{-1}\mathbf{B}_k)^{-1}\mathbf{B}_k^\top\mathbf{E}_j^{-1}\right)(-\mathbf{r}_j + \mathbf{K}^\top\mathbf{p}_j)$
  compute $\Delta\boldsymbol{\lambda}_j = \mathbf{p}_j - \mathbf{W}_j^{-1}\boldsymbol{\Lambda}_j\mathbf{K}\Delta\mathbf{z}_j$
  compute $\Delta\mathbf{w}_j = \mathbf{K}\Delta\mathbf{z}_j + \mathbf{v}_j$
  compute $\beta_j = \min(\beta_\tau^{\text{pri}}, \beta_\tau^{\text{dual}})$, where $\beta_\tau^{\text{pri}} = \max\{\beta \in (0,1] : \mathbf{w}_j + \beta\Delta\mathbf{w}_j \geq (1-\tau)\mathbf{w}_j\}$ and $\beta_\tau^{\text{dual}} = \max\{\beta \in (0,1] : \boldsymbol{\lambda}_j + \beta\Delta\boldsymbol{\lambda}_j \geq (1-\tau)\boldsymbol{\lambda}_j\}$
  update the variables $(\mathbf{z}_{j+1}, \mathbf{w}_{j+1}, \boldsymbol{\lambda}_{j+1}) = (\mathbf{z}_j, \mathbf{w}_j, \boldsymbol{\lambda}_j) + \beta_j(\Delta\mathbf{z}_j, \Delta\mathbf{w}_j, \Delta\boldsymbol{\lambda}_j)$
  if $\|\mathbf{r}_j\|_2 <$ tol and $\|\mathbf{v}_j\|_2 <$ tol then
    break
  end if
end for
Output: $\mathbf{z}_{j+1}$ approximate projection of $\mathbf{y}_{k+1}$ onto $C$
```

$\mathbf{P}_k \in \mathbb{R}^{(n-m)\times n}$ the projection operators onto the active and inactive set, respectively. For example, $\mathbf{R}_k$ can be constructed by selecting the rows of an identity matrix associated with $\mathcal{A}_k$. We shall discuss two common choices for constructing $\mathcal{A}_k$ below. Given $\mathbf{P}_k$ and $\mathbf{R}_k$, the intermediate step in (1.2) is

$$\mathbf{y}_{k+1} = \mathbf{x}_k - \mu_k\left(\mathbf{P}_k^\top\tilde{\mathbf{H}}_k^{-1}\mathbf{P}_k\nabla f(\mathbf{x}_k) + \nu_k^{-1}\mathbf{R}_k^\top\mathbf{R}_k\nabla f(\mathbf{x}_k)\right). \tag{3.7}$$

Here, $\tilde{\mathbf{H}}_k$ is a rank-l approximation of the projected (approximate) Hessian $\mathbf{P}_k\mathbf{G}_k\mathbf{P}_k^\top$ and the constant $\nu_k > 0$ is used to balance the sizes of both steps. In practice, this number is often chosen based on the norm of the step for the inactive components. One can verify that this leads to the PNKH-B scheme with the Hessian approximation

$$\mathbf{H}_k = \begin{pmatrix}\mathbf{P}_k^\top & \mathbf{R}_k^\top\end{pmatrix}\begin{pmatrix}\tilde{\mathbf{H}}_k & 0 \\ 0 & \nu_k\mathbf{I}_m\end{pmatrix}\begin{pmatrix}\mathbf{P}_k \\ \mathbf{R}_k\end{pmatrix}. \tag{3.8}$$

We use the separability introduced by this construction in the projection, which decouples into using (1.4) on the active components and using the interior point method on the inactive components.

We obtain two variants of PNKH-B that differ only by the strategy to estimate active and inactive variables.

PNKH-B (boundary index). Perhaps the most straightforward estimate of the active set is to choose the components at which the bound-constraints are active, i.e.,

$$\mathcal{A}_k^{\text{bound}} = \{i : (\mathbf{x}_k)_i = l_i \;\text{ or }\; (\mathbf{x}_k)_i = u_i\}. \tag{3.9}$$

This choice has been used successfully in [14].

Algorithm 3.2 Outline of PNKH-B for solving (1.1)

```
Inputs: Initial guess $\mathbf{x}_0 \in C$, tolerance xtol and gtol, line search parameter $\alpha \in (0,1)$, and the rank of the low-rank approximation $l$
for $k = 0, 1, 2, \dots$ do
  select estimate of active set, $\mathcal{A}_k \in \{\emptyset, \mathcal{A}_k^{\text{bound}}, \mathcal{A}_k^{\epsilon}\}$, and build projection matrices $\mathbf{P}_k$ and $\mathbf{R}_k$.
  compute $f(\mathbf{x}_k)$, $\nabla f(\mathbf{x}_k)$ and (approximate) Hessian $\mathbf{G}_k \approx \nabla^2 f(\mathbf{x}_k)$
  compute the Lanczos tridiagonalization $\tilde{\mathbf{H}}_k = \mathbf{V}_k \mathbf{T}_k \mathbf{V}_k^\top \approx \mathbf{P}_k \mathbf{G}_k \mathbf{P}_k^\top$ with initial vector $-\mathbf{P}_k \nabla f(\mathbf{x}_k)$ (use matrix-free implementation)
  compute the Hessian approximation $\mathbf{H}_k$ in (3.8) and the search direction $-\mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k)$
  set $\mu = 1$
  for $i = 0, 1, 2, \dots$ do
    solve the projection $\mathbf{x}_t = \Pi_{\|\cdot\|_{\mathbf{H}_k}}(\mathbf{x}_k - \mu \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k))$ (see Subsection 3.2)
    if $f(\mathbf{x}_t) < f(\mathbf{x}_k) + \alpha \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_t - \mathbf{x}_k)$ then
      set $\mathbf{x}_{k+1} = \mathbf{x}_t$ and break
    else
      set $\mu = \mu/2$
    end if
  end for
  if $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 / \|\mathbf{x}_k\|_2 <$ xtol or norm of projected gradient $<$ gtol then
    break
  end if
end for
Output: approximate solution $\mathbf{x}_{k+1} \in C$.
```

PNKH-B (ϵ index). As an alternative active set estimation scheme we use the one proposed in [4, 38]. Here, for some $\epsilon > 0$, the idea is to use an ϵ margin around the boundary and also consider the sign of the partial derivative so that curvature information is used for those constraints predicted to become inactive, i.e.,

$$\mathcal{A}_k^{\epsilon} = \{ i \,:\, [(\mathbf{x}_k)_i \le l_i + \epsilon \,\wedge\, \partial_i f(\mathbf{x}_k) > 0] \text{ or } [(\mathbf{x}_k)_i \ge u_i - \epsilon \,\wedge\, \partial_i f(\mathbf{x}_k) < 0] \}\,. \tag{3.10}$$

4. Proof of Global Convergence.

In this section, we introduce and prove the theorem, which guarantees the global convergence of PNKH-B under mild assumptions. We first state the main theorem.

Theorem 4.1 (Global Convergence). *Suppose*

1. *f is twice differentiable, and ∇f is Lipschitz continuous.*
2. *$\inf_{\mathbf{x}}\{f(\mathbf{x}) | \mathbf{x} \in C\}$ is attained, and C is a box.*
3. *The approximated Hessian $\mathbf{H}_k$'s in our method are symmetric and uniformly positive definite, i.e. $\mathbf{H}_k \succeq s\mathbf{I}$ for some $s > 0$ and for all $k \in \mathbb{N}$.*

Then the sequence $\{\mathbf{x}_k\}_k$ generated by PNKH-B converges to a stationary point of (1.1) regardless of the choice of the starting point $\mathbf{x}_0 \in C$.

While the low-rank Hessian approximation $\mathbf{H}_k$ is not uniformly positive definite on $\mathbb{R}^n$, it is on the subspace spanned by the columns of $\mathbf{V}_k$. Two ways to obtain a uniformly positive definite approximation from $\mathbf{H}_k$ are adding a sufficiently large shift to all vectors in the orthogonal complement of this subspace or adding a multiple of the identity. Since such a matrix is symmetric positive definite, the assumptions hold PNKH-B with and without variable partitioning. Unlike two-metric methods, the convergence of PNKH-B does not hinge upon active/inactive variable partitioning. Moreover by the theorem, our methods globally converge to the optimal solution for convex problems. We now begin to prove the theorem. The proof follows the approach in [31], which studies proximal Newton-type methods. We first state and prove some lemmas, which will be used to prove the global convergence.

Lemma 4.2 (Descent Direction). *If f is twice differentiable, $\mathbf{H}_k$ is symmetric positive definite, and C is a box, then for any $\mu_k > 0$, the update step $\mathbf{d}_k := \mathbf{x}_{k+1} - \mathbf{x}_k$ generated by (1.2) and (1.3) satisfies*

$$\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k \leq -\frac{1}{\mu_k}\mathbf{d}_k^\top \mathbf{H}_k \mathbf{d}_k. \tag{4.1}$$

Hence the update step $\mathbf{d}_k$ is a descent direction.

Proof of Lemma 4.2. By the second projection theorem [1, Chapter 9.3], the iterate $\mathbf{x}_{k+1} = \Pi_{\|\cdot\|_{\mathbf{H}_k}}(\mathbf{y}_{k+1})$ if and only if

$$(\mathbf{y}_{k+1} - \mathbf{x}_{k+1})^\top \mathbf{H}_k (\mathbf{z} - \mathbf{x}_{k+1}) \leq 0 \quad \text{for all } \mathbf{z} \in C. \tag{4.2}$$

Substituting $\mathbf{y}_{k+1} = \mathbf{x}_k - \mu_k \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k)$ and $\mathbf{z} = \mathbf{x}_k$ in (4.2), we obtain

$$(\mathbf{x}_k - \mu_k \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k) - \mathbf{x}_{k+1})^\top \mathbf{H}_k (\mathbf{x}_k - \mathbf{x}_{k+1}) \leq 0,$$

which is equivalent to (4.1). □

Lemma 4.3 (Armijo Line Search Condition). *Suppose C is a box, ∇f is Lipschitz continuous with constant $L > 0$, $\mathbf{H}_k$'s are symmetric, and $\mathbf{H}_k \succeq s\mathbf{I}$ for all $k \in \mathbb{N}$ and for some $s > 0$, i.e.*

$$||\mathbf{z}||_{\mathbf{H}_k}^2 \geq s||\mathbf{z}||_2^2, \quad \textit{for all } \mathbf{z} \textit{ and for all } k \in \mathbb{N}.$$

For line search parameter $\alpha \in (0,1)$, if step size μ_k satisfies

$$\mu_k \leq \min\left(1, \frac{2s}{L}(1-\alpha)\right),$$

then the following sufficient descent condition is satisfied

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + \alpha \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k). \tag{4.3}$$

Proof of Lemma 4.3. Since $\mathbf{x}_k, \mathbf{x}_{k+1} \in C$, by the Lipschitz continuity of f, we have

$$\begin{aligned} f(\mathbf{x}_{k+1}) &\leq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k) + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2^2 \\ &\leq f(\mathbf{x}_k) + \alpha \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k). \end{aligned}$$

Here, the first step uses $\mu_k \leq \frac{2s}{L}(1-\alpha)$, $\mathbf{H}_k \succeq sI$ and (4.1) and in the second step we use $\nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k) \leq 0$; see Lemma 4.2. □

Lemma 4.4. *Suppose the assumptions on f, C and $\mathbf{H}_k$ are the same as those in Lemma 4.3. Also the backtracking Armijo line search scheme is used. Then $\mathbf{x}_*$ is a stationary point of (1.1) if and only if $\mathbf{x}_*$ is a fixed point of our method.*

Proof of Lemma 4.4. The iterate $\mathbf{x}_*$ is a fixed point of our method if and only if

$$\mathbf{x}_* = \Pi_{\|\cdot\|_{\mathbf{H}_*}} \left(\mathbf{x}_* - \mu_* \mathbf{H}_*^{-1} \nabla f(\mathbf{x}_*)\right), \tag{4.4}$$

where $\mathbf{H}_*$ is the Hessian approximation, and $\mu_* > 0$ is the step size. By the second projection theorem again, it is equivalent to

$$(\mathbf{x}_* - \mu_* \mathbf{H}_*^{-1} \nabla f(\mathbf{x}_*) - \mathbf{x}_*)^\top \mathbf{H}_* (\mathbf{z} - \mathbf{x}_*) \leq 0 \quad \text{for all } \mathbf{z} \in C.$$

This is simplified to $\nabla f(\mathbf{x}_*)^\top (\mathbf{z} - \mathbf{x}_*) \geq 0$ for all $\mathbf{z} \in C$, which is true if and only if $\mathbf{x}_*$ is a stationary point of the problem. ∎

Now, we are ready to prove Theorem 4.1, the global convergence of our method.

Proof of Theorem 4.1. The sequence $\{f(\mathbf{x}_k)\}_k$ is decreasing because the update directions are descent directions (Lemma 4.2) and the backtracking Armijo line search scheme guarantees sufficient descent at each step (Lemma 4.3). Since f is closed and its infimum in C is attained, the decreasing sequence $\{f(\mathbf{x}_k)\}_k$ converges to a limit.

By the sufficient descent condition (4.3), the convergence of $\{f(\mathbf{x}_k)\}_k$ and $\alpha > 0$,

$$\nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k)$$

converges to zero. By Lemma 4.2, one has

$$(\mathbf{x}_{k+1} - \mathbf{x}_k)^\top \mathbf{H}_k (\mathbf{x}_{k+1} - \mathbf{x}_k) \leq -\mu_k \nabla f(\mathbf{x}_k)^\top (\mathbf{x}_{k+1} - \mathbf{x}_k).$$

Hence $(\mathbf{x}_{k+1} - \mathbf{x}_k)^\top \mathbf{H}_k (\mathbf{x}_{k+1} - \mathbf{x}_k)$ converges to zero. Since $\mathbf{H}_k$'s are uniformly positive definite, $\mathbf{x}_{k+1} - \mathbf{x}_k$ converges to the zero vector.

This implies that the sequence $\{\mathbf{x}_k\}_k$ converges to a fixed point of our method. By Lemma 4.4, the sequence converges to a stationary point of the problem. ∎

5. Experimental Results.

We apply PNKH-B to three large-scale numerical experiments from applications. We compare its performance with two state-of-the-art projected Newton-CG (PNCG) methods, which are two-metric schemes. In Subsection 5.1, we discuss the comparing schemes. In Subsection 5.2, we consider a PDE parameter estimation problem. In Subsection 5.3, we apply our method to an image classification problem. In Subsection 5.4, we experiment with an image reconstruction problem. All these applications require fitting a computational model to data, which is typically noisy. Therefore, and since the computational models can be expensive, we seek to use the optimization scheme to obtain a high-quality reconstruction within only a few iterations. In all three experiments, using the low-rank approximated Hessian metric during the projection renders PNKH-B competitive with respect to the optimization performance and reconstruction quality to similar state-of-the-art two-metric methods.

5.1. Benchmark Methods.

We compare PNKH-B to an implementation of the two-metric scheme described in [14] and a variant that includes the ϵ-indexing scheme from [4, 38]. We refer to the scheme obtained using $\mathcal{A}_k^{\text{bound}}$ as PNCG (boundary index) and the scheme obtained using $\mathcal{A}_k^{\epsilon}$ as PNCG (ϵ index); see Subsection 3.3. The main difference between these schemes and our proposed method is the projection. The PNCG schemes use (1.4) to project all the components and are therefore considered two-metric schemes. In contrast, our PNKH-B scheme uses the metric induced by the low-rank approximated Hessian metric during the projection and is therefore a one-metric scheme.

5.2. Experiment 1: Direct Current Resistivity. We use PNKH-B to solve a PDE parameter estimation problem motivated by the Direct Current Resistivity (DCR) described in [14, 36]; see also [9, 32, 41, 44] for background and different instances of this problem.

Model Description. The goal of DCR in geophysical imaging is to estimate the conductivity of the subsurface by means of indirect measurement obtained on the earth's surface. Specifically, it first uses electrical sources on the surface to generate direct currents to create electric potential fields in the subsurface. Measurements of these potential fields are then collected on the surface. The parameter estimation aims at reconstructing a three-dimensional image of the conductivity in the subsurface that is consistent with the measurements; for more details and illustrations of the DCR experiment see, e.g., [9, 14, 32, 36].

To set up the problem instance, we follow the same discretize-then-optimize approach described in [14] that is also used in [36, 43]. Using a uniform mesh with N_m cells and N_n nodes, we obtain the discrete forward problem

$$\mathbf{D} = \mathbf{P}^\top \mathbf{A}(\mathbf{m})^{-1}\mathbf{Q} + \boldsymbol{\epsilon} = \mathbf{P}^\top \mathbf{U} + \boldsymbol{\epsilon}, \tag{5.1}$$

where $\mathbf{A}(\mathbf{m}) \in \mathbb{R}^{N_n \times N_n}$ is a finite-volume discretization of the Poisson operator for the conductivity model $\mathbf{m} \in \mathbb{R}^{N_m}$, $\mathbf{P} \in \mathbb{R}^{N_n \times N_r}$ is the receiver matrix that maps the fields to data, the columns of $\mathbf{Q} \in \mathbb{R}^{N_n \times N_s}$ are discretized sources, the columns of $\mathbf{U} \in \mathbb{R}^{N_n \times N_s}$ are the potential fields, and $\boldsymbol{\epsilon} \in \mathbb{R}^{N_r \times N_s}$ is Gaussian noise. Here N_r and N_s are the number of receivers and sources, respectively. Note that with suitable discretization and boundary conditions, $\mathbf{A}$ is non-singular, which means that $\mathbf{m} \mapsto \mathbf{U}(\mathbf{m})$ is well-defined and differentiable.

Given the measurement data $\mathbf{D}$, sources $\mathbf{Q}$, and receivers $\mathbf{P}$, we estimate the corresponding model parameter $\mathbf{m}$ by solving the optimization problem

$$\min_{\mathbf{m}} \frac{1}{2}\|\mathbf{P}^\top \mathbf{A}(\mathbf{m})^{-1}\mathbf{Q} - \mathbf{D}\|_F^2 + \frac{\gamma}{2}\|\mathbf{L}(\mathbf{m} - \mathbf{m}_{\text{ref}})\|_2^2 \quad \text{subject to} \quad \mathbf{m}_l \le \mathbf{m} \le \mathbf{m}_u.$$

Here, $\gamma > 0$ is a regularization parameter, $\mathbf{m}_{\text{ref}}$ is a given reference model, $\mathbf{L}$ is a regularization operator, $\mathbf{m}_l$ and $\mathbf{m}_u$ are the upper and lower bounds respectively, which are used to enforce the physical constraints for the model parameters.

As common, we use the Gauss-Newton approximate Hessian $\mathbf{G}$ given by

$$\mathbf{G} = \mathbf{J}(\mathbf{m})^\top \mathbf{J}(\mathbf{m}) + \gamma \mathbf{L}^\top \mathbf{L},$$

where the Jacobian of the residual of (5.1) is

$$\mathbf{J}(\mathbf{m}) = -\mathbf{P}^\top \mathbf{A}(\mathbf{m})^{-1}(\nabla_{\mathbf{m}}(\mathbf{A}(\mathbf{m})\mathbf{U}))^\top. \tag{5.2}$$

Note that the dimensions of $\mathbf{m}$ are typically very large. Moreover each evaluation of the objective function or product with the Jacobian $\mathbf{J}$ or its transpose or computing the approximate Hessian-vector multiplication $\mathbf{y} \mapsto \mathbf{G}\mathbf{y}$ require inverting the PDE operator $\mathbf{A}$ (i.e., solving the PDE) $\min(N_r, N_s)$ times per source. Hence the computations in each outer (Newton-Krylov method) or inner (line search) iterations when solving the DCR model problem are very expensive, especially when there are a lot of sources.

Experimental Results. In this experiment, we solve a 3-dimensional DCR problem on a mesh containing $36 \times 36 \times 12$ cells discretizing the domain $\Omega = (0, 1)^3$. The test problem features 25 sources and 1,369 receivers located on the top surface. Following

Table 1: Comparison of runtime of the five methods and the interior point method (IPM) on the three experiments. The sizes of variables of experiment 1, 2 and 3 are 9900, 40010 and 8192, respectively. The total number of iterations is 20. The tests are run on a laptop computer with an Intel Core i5-7200U CPU, 8 GB RAM, and the software platform is MATLAB R2018b.

Runtime (s)	PNCG (b. index)	PNCG (ϵ index)	PNKH-B	PNKH-B (b. index)	PNKH-B (ϵ index)
Experiment 1	173.4	170.2	176.9	168.7	167.4
IPM (mean)			8.1 (0.31)	2.1 (0.10)	2.3 (0.11)
Experiment 2	795.1	767.7	777.4	783.5	753.0
IPM (mean)			16.1 (0.52)	9.0 (0.31)	10.4 (0.35)
Experiment 3	29.8	28.5	39.1	33.7	34.0
IPM (mean)			10.8 (0.36)	5.1 (0.19)	5.1 (0.20)

the finite volume discretization presented in [14], we use a cell-centered discretization of the model $\mathbf{m}$ and nodal discretizations of the sources, receivers, and fields. We add 1% noise to the data and enforce smoothness by using a diffusion regularizer with regularization parameter $\alpha = 10^{-3}$. We also use symmetric successive over-relaxation (SSOR) as a preconditioner.

In our setup, we exclude voxels close to the boundary, sources, and receivers from the inversion. As a result, our model $\mathbf{m}$ is discretized over $30 \times 30 \times 10$ cells instead; in particular, $\mathbf{m}$ has size $n = 9900$. The bounds $\mathbf{m}_l$ and $\mathbf{m}_u$ are set as vectors of all -4.6's and -1's, respectively. The upper bound is purposely set as smaller than some pixel values of the ground truth to test the ability of the methods to identify the active variables. The main cost of the parameter estimation is the large number of discrete PDE solves to evaluate the objective function, its gradient, and matrix-vector products with $\mathbf{J}$ and $\mathbf{J}^\top$. Therefore, we limit the number of CG/Lanczos to five in all instances.

The experimental results for the DCR problem are shown in Figures 2 to 4. In Figure 2(a)-(b), the proposed methods have a significant boost in the initial convergence on the objective function value and the norm of the projected gradient. This is particularly evident in the early iterations as can be seen, e.g., by a one-order reduction of the objective function and projected gradient in the second iteration and the visual quality of the parameter estimate at the third iteration; see Figure 3. At this iteration, we see that the proposed PNKH-B and PNKH-B (ϵ index)'s results are closer to the ground truth and appear smoother. While the results obtained using all methods are similar at the final iteration, we note that the PNCG scheme with boundary indices leads to a non-smooth reconstruction; see Figure 4. Since PNCG is a two-metric scheme, the loss of the smoothness might be due to suboptimal scaling of the gradient step in (3.7) or the inconsistency of the preconditioner caused by the indexing. The proposed methods also have slightly smaller objective values after 20 iterations. Table 1 shows that all five methods require a comparable runtime. We highlight that the added costs of the interior point method used to compute the projection is only between 1.2% and 5% and took on average between 0.1 and 0.3 seconds. While the Lanczos tridiagonalization in PNKH-B takes longer on average than the conjugate gradient method in PNCG, PNKH-B required fewer backtracking line search iterations and hence PDE solves.

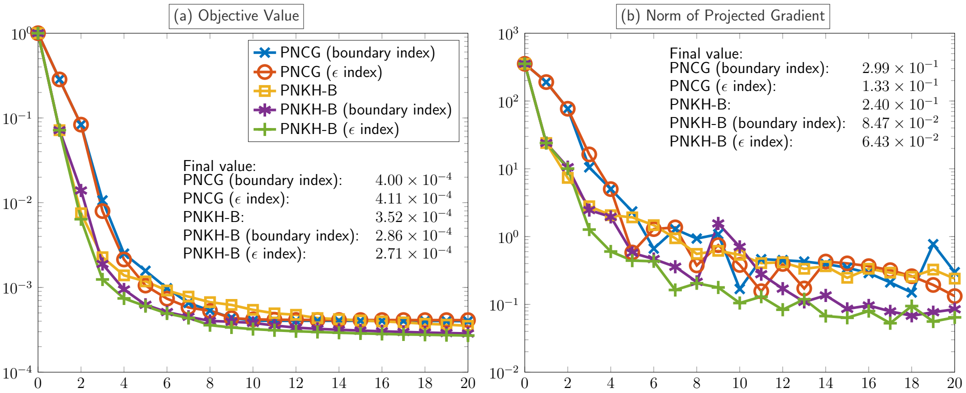

Fig. 2: Comparison of the convergence of two PNCG methods and three variants of PNKH-B for the direct current resistivity experiment in Subsection 5.2. (a): Relative reduction of objective function. (b): Norm of the projected gradient.

5.3. Experiment 2: Image Classification. We compare the performance of PNKH-B and PNCG for a multinomial logistic regression (MLR) arising in the supervised classification of hand-written digits in the MNIST dataset [30].

Model Description. Let n_f denote the number of features, n_c the number of classes, and Δ_{n_c} be the unit simplex in $\mathbb{R}^{n_c}$. Given training data $\{(\mathbf{b}_j, \mathbf{c}_j)\}_{j=1}^{N} \subset \mathbb{R}^{n_f} \times \Delta_{n_c}$, the supervised classification problem aims at training a hypothesis function $h_{\mathbf{X}} : \mathbb{R}^{n_f} \to \Delta_{n_c}$ that accurately approximates the input-output relationship for new examples, i.e.,

$$h_{\mathbf{X}}(\mathbf{d}_i^{\text{test}}) \approx \mathbf{c}_i^{\text{test}}, \qquad \text{for} \qquad i = 1, \ldots, M. \tag{5.3}$$

Here, $\mathbf{X}$ are parameters of the hypothesis function and $\{\mathbf{d}_i^{\text{test}}, \mathbf{c}_i^{\text{test}}\}_{i=1}^{M}$ is a test dataset, which is not used during training.

A common strategy for finding the hypothesis function is by solving the MLR problem

$$\min_{\mathbf{X}_l \leq \mathbf{X} \leq \mathbf{X}_u} \frac{1}{N} \sum_{j=1}^{N} -\mathbf{c}_j^{\top} \log\left(h_{\mathbf{X}}(\mathbf{d}_j)\right) \quad \text{where} \quad h_{\mathbf{X}}(\mathbf{d}_j) = \frac{\exp\left(\mathbf{X}\mathbf{d}_j\right)}{\mathbf{e}_{n_c}^{\top} \exp\left(\mathbf{X}\mathbf{d}_j\right)}. \tag{5.4}$$

Here, the hypothesis function is a linear model followed by a softmax transformation, which ensures that $h_{\mathbf{X}}(\mathbf{d}) \in \Delta_{n_c}$ and the objective is to minimize the cross-entropy between the predicted probability distribution and the label. In the formulation above, we use $\mathbf{X}_l, \mathbf{X}_u \in \mathbb{R}^{n_c \times n_f}$ to model lower and upper bounds on the entries of $\mathbf{X}$, respectively, with the goal to regularize the problem and improve generalization, which means improving the performance on the test data set. Since the MLR problem is a smooth convex optimization problem, we use $\mathbf{G} = \nabla^2 f(\mathbf{X})$.

In our experiment, we use the MNIST dataset [30], which consists of 60,000 28 × 28 grey-scale hand-written images of digits ranging from 0 to 9 that are split into $N = 50,000$ training images and $M = 10,000$ validation images. Applying the hypothesis function to the (vectorized) images directly provides suboptimal performance. Therefore, we follow the approach in [23] and apply a single layer neural

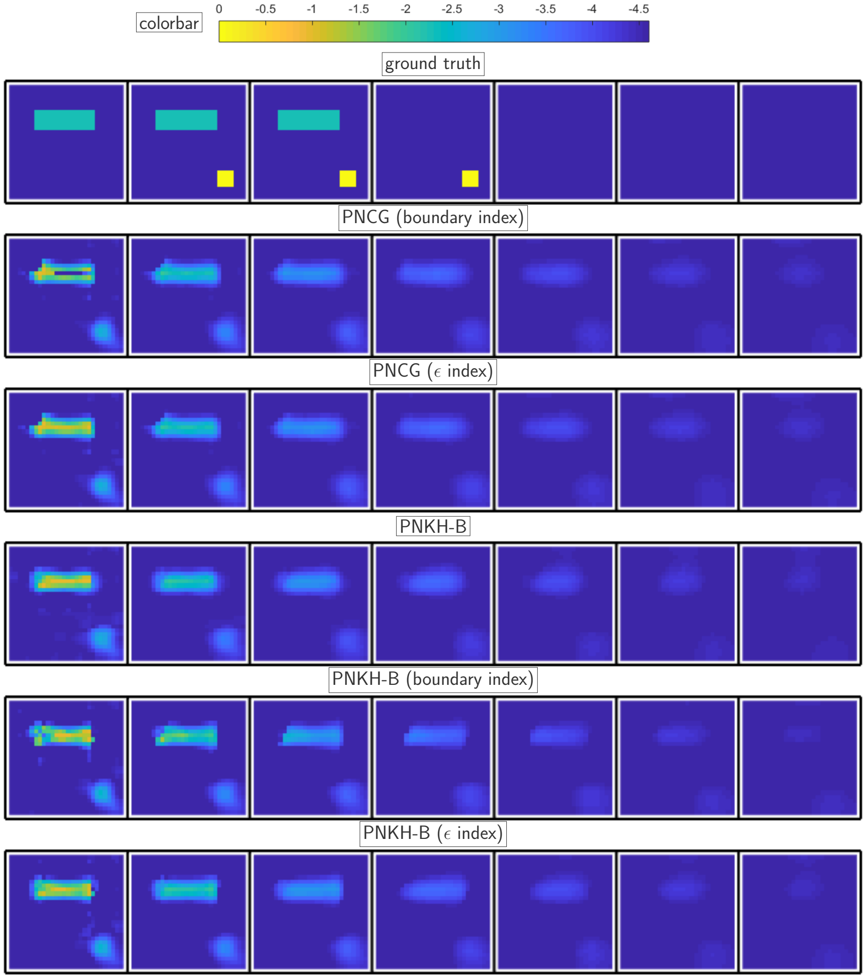

Fig. 3: Results after the third iteration on DCR generated by the five methods. The upper bound is purposely set to be $\mathbf{m}_u = -1$, which is smaller than some pixel values in the ground truth to test the ability of the methods to identify active variables.

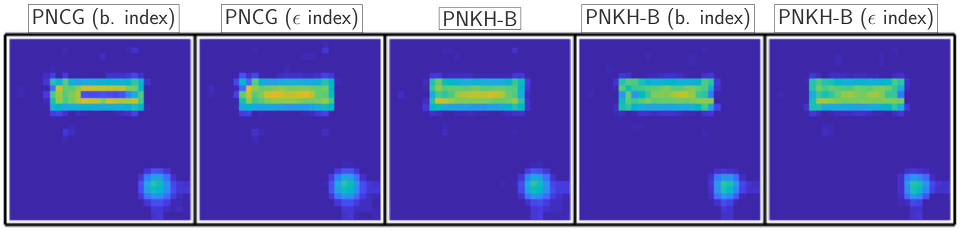

Fig. 4: First slice of the final results on DCR generated by the five methods. There are noticeable artifacts in the final results of PNCG.

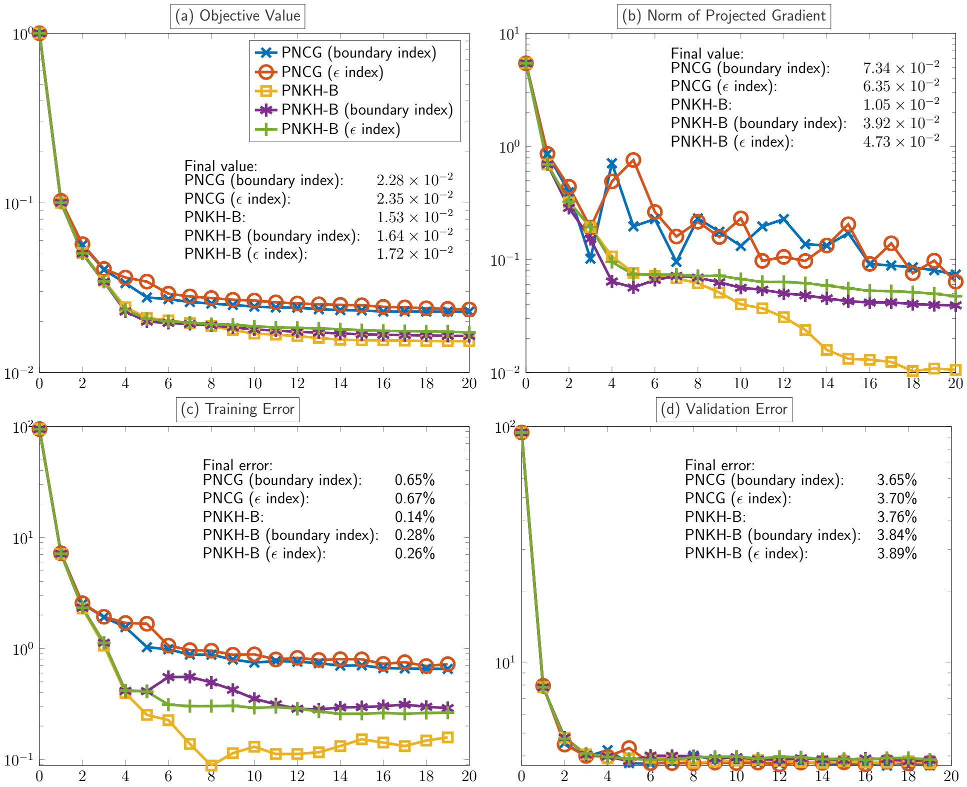

Fig. 5: Comparison of the convergence of two PNCG methods and three variants of PNKH-B for the image classification problem in Subsection 5.3. (a): Relative reduction of objective function. (b): Norm of the projected gradient. (c): Training errors (d): Validation errors

network to obtain feature vectors $n_f = 4001$-dimensional space. Here, the last component is equal to 1 for all images to model a bias term and the other components are obtained using a random affine transformation and a tanh activation function.

Experimental Results. We use a fixed number of 20 inexact Newton steps with 20 CG/Lanczos iterations per step for all five methods and manually tune the bounds on $\mathbf{X}$ so that the trained hypothesis function performs well on the validation data. In our case, we choose the entries of $\mathbf{X}_l$ and $\mathbf{X}_u$ to be -0.05 and 0.05, respectively. The performance of the optimization schemes and the accuracy of the hypothesis function can be seen in Figure 5. In particular, in Figure 5(a)-(c), the three PNKH-B methods boost the initial convergence and outperform the PNCG methods with respect to the objective function value, norm of the projected gradient, and training error by some margin. The comparison for the validation data is overall comparable, but the PNCG schemes achieve slightly lower error rates; see Figure 5(d). Despite the more expensive projection step, the PNKH-B variants require a similar runtime in this experiment; see Table 1.

5.4. Experiment 3: Spectral Computed Tomography. We consider an image reconstruction problem arising in energy-windowed spectral computed tomography (CT). The goal is to identify the material composition of an object from measurements taken with x-rays at different energy levels and from different projection angles. Our experimental setup follows [20, 21] which also provide an excellent description and derivation of the problem.

Model Description. As a forward model, we consider the discretized energy-windowed spectral CT model

$$\mathbf{y} = (\mathbf{S}^\top \otimes \mathbf{I})\exp\{-(\mathbf{C} \otimes \mathbf{A})\mathbf{w}\} + \boldsymbol{\epsilon}, \tag{5.5}$$

where $\mathbf{I} \in \mathbb{R}^{(N_d \cdot N_p)\times(N_d \cdot N_p)}$ is the identity matrix, $\mathbf{S} \in \mathbb{R}^{N_e \times N_b}$ contains the spectrum energy of each energy window, $\mathbf{C} \in \mathbb{R}^{N_e \times N_m}$ contains the attenuation coefficients of each material at each energy level, $\mathbf{A} \in \mathbb{R}^{(N_d \cdot N_p)\times N_v}$ contains the lengths of the x-ray beams, $\mathbf{y} \in \mathbb{R}^{N_d \cdot N_p \cdot N_b}$ is the observed data containing the x-ray photons of each energy window, $\mathbf{w} \in \mathbb{R}^{N_v \cdot N_m}$ represents the weights of the materials of each pixel (and is the unknown variable), and $\boldsymbol{\epsilon} \in \mathbb{R}^{N_d \cdot N_p \cdot N_b}$ is the measurement noise. Here, N_p is the number of angles of the x-ray beams, N_b is the number of detectors and each of them detects a specific energy window, N_m is the number of materials, N_e is the number of energy levels of the emitted x-ray beams, N_d and N_v are related to the number of pixels of the image. In particular, for an image of size $n \times n$, $N_d = n$ and $N_v = n^2$.

The goal of the energy-windowed spectral CT model is to estimate the weights of materials $\mathbf{w}$ given the other variables except the noise in (5.5). Hence we formulate the following optimization problem

$$\min_{0 \leq \mathbf{w} \leq \mathbf{w}_u} \frac{1}{2}\|\mathbf{y} - (\mathbf{S}^\top \otimes \mathbf{I})\exp\{-(\mathbf{C} \otimes \mathbf{A})\mathbf{w}\}\|_2^2 + \frac{\gamma_1}{2}\|\mathbf{D}\mathbf{w}_{1:N_v}\|_2^2 + \gamma_2 \sum_{i=N_v+1}^{2N_v} \mathbf{w}_i.$$

Here, the bound constraints are used to enforce physical bounds, where the weights cannot be negative and cannot exceed the upper bound $\mathbf{w}_h$. The second and third terms are the regularization terms also used in [21]. The second term involves the discrete gradient operator $\mathbf{D}$ and enforces smoothness of the first material and the last term promotes sparsity of the second material. As common in nonlinear least-squares problems, we use the Gauss-Newton approximation of the Hessian, i.e.,

$$\mathbf{G} = \mathbf{J}(\mathbf{w})^\top \mathbf{J}(\mathbf{w}) + \gamma_1 \mathbf{D}^\top \mathbf{D},$$

where $\mathbf{D}$ is a discrete differential operator acting on the first N_v entries.

Experimental Results. The size of the variables of this problem is $n = 8192$. Since the Kronecker products are implemented effectively, the CT model problem is the least intense among the three testing problems in terms of computational cost. Therefore, we set the number of CG/Lanczos iterations to 60 for all five methods. Moreover, we purposely choose a tight bound $\mathbf{w}_u = [1.5, 1.5, ..., 1.5]^\top$ to test the ability of the methods to compute a solution with many active entries, specifically some entries in the ground truth are outside of this bound. The experimental results of the CT model problem are shown in Figures 6 and 7. The proposed methods converge faster initially and all schemes achieve comparable results. In the second iteration of Figure 6(a), the iterate of the three proposed methods achieve 25 times smaller objective function values than the comparing methods. This also leads to a considerable improvement in the reconstruction quality; see Figure 7. In Figure 6(b), PNKH-B with boundary index and ϵ index give competitive performance in terms of the norm of the projected

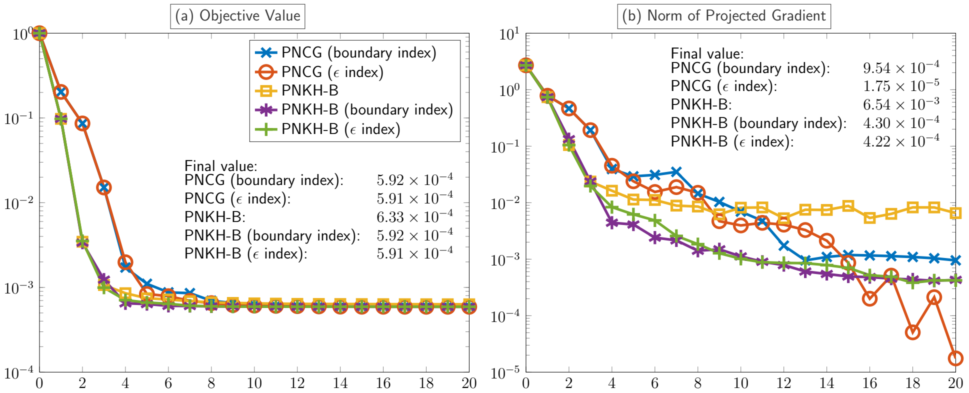

Fig. 6: Comparison of the convergence of two PNCG methods and three variants of PNKH-B for the energy-windowed spectral CT problem in Subsection 5.4. (a): Relative reduction of objective function. (b): Norm of the projected gradient.

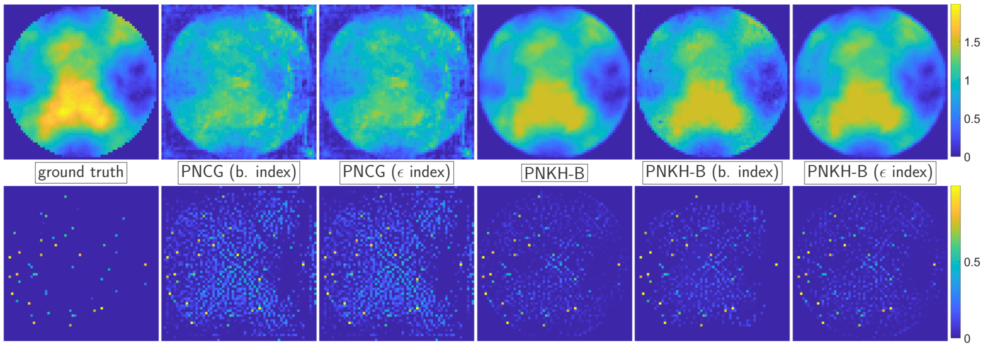

Fig. 7: Reconstructed images after the second iteration generated by the five methods on CT. The top and bottom images are the estimated composition of the two materials. The upper bound is purposely set to be $\mathbf{w}_u = 1.5$, which is smaller than some pixel values in the ground truth, to test the ability of the methods to identify active variables.

gradient. PNCG with ϵ index generates the best final norm of projection gradient. In this example, the overhead of the PNKH-B is around 15% due to the higher ratio between the costs of the projection and the forward model, which is less expensive compared to the other experiments; see Table 1.

6. Conclusion.

We present PNKH-B, a Projected Newton-Krylov method for bound-constrained minimization whose search direction and projection rely on a low-rank approximation of the (approximate) Hessian. Our method can be seen as a generalization of Newton-CG methods to bound-constrained problems since we compute the low-rank approximation of the Hessian using a few steps of Lanczos tridiagonalization. The novelty of our method is the use of the metric induced by this approximation in the projection step. We contribute an interior point method that effectively exploits the low-rank approximation to achieve a complexity that is linear with respect to the number of variables. The consistent use of the metric leads to a simpler algorithm compared to two-metric schemes that require partitioning into active and inactive variables to ensure convergence. We also propose two variants of the framework, which incorporate the current knowledge of the active/inactive variables; this improved the convergence in some cases. The experimental results on PDE parameter estimation, machine learning and image reconstruction show that the proposed methods lead to faster initial convergence with moderate runtime overhead compared to the existing state-of-the-art projected Newton-CG methods. Our methods are also competitive in the final objective value, norm of the projected gradient and reconstruction quality. We provide our MATLAB code at https://github.com/EmoryMLIP/PNKH-B.

Acknowledgments. The authors would like to thank Yunyi Larry Hu and James Nagy for sharing the data and code for the computer tomography experiment.

REFERENCES

[1] A. Beck, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*, vol. 19, SIAM, 2014.

[2] S. Becker and J. Fadili, *A quasi-Newton proximal splitting method*, in Advances in Neural Information Processing Systems, 2012, pp. 2618–2626.

[3] S. Becker, J. Fadili, and P. Ochs, *On quasi-Newton forward–backward splitting: Proximal calculus and convergence*, arXiv preprint arXiv:1801.08691, (2018).

[4] D. P. Bertsekas, *Projected Newton Methods for Optimization Problems with Simple Constraints*, SIAM Journal on Control and Optimization, 20 (1982), pp. 221–246.

[5] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Control and Optimization, 16 (1995), pp. 1190–1208.

[6] R. H. Chan, K. K. Kan, M. Nikolova, and R. J. Plemmons, *A two-stage method for spectral–spatial classification of hyperspectral images*, Journal of Mathematical Imaging and Vision, (2020), pp. 1–18.

[7] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, *Reversible architectures for arbitrarily deep residual neural networks*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[8] F. Curtis and J. Nocedal, *Steplength selection in interior-point methods for quadratic programming*, Applied Mathematics Letters, 20 (2007), pp. 516–523.

[9] A. Dey and H. F. Morrison, *Resistivity modeling for arbitrarily shaped three-dimensional structures*, Geophysics, 44 (1979), pp. 753–780.

[10] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159.

[11] J. C. Dunn, *Newton's method and the Goldstein step-length rule for constrained minimization problems*, SIAM Journal on Control and Optimization, 18 (1980), pp. 659–674.

[12] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, *Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations*, SIAM Journal on Scientific Computing, 33 (2011), pp. 407–432.

[13] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM Journal on Control and Optimization, 22 (1984), pp. 936–964.

[14] E. Haber, *Computational methods in geophysical electromagnetics*, Mathematics in Industry (Philadelphia), Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.

[15] E. Haber, K. Lensink, E. Triester, and L. Ruthotto, *IMEXnet: A forward stable deep neural network*, arXiv preprint arXiv:1903.02639, (2019).

[16] E. Haber and L. Ruthotto, *Stable architectures for deep neural networks*, Inverse Problems, 34 (2017), p. 014004.

[17] E. Haber, L. Ruthotto, E. Holtham, and S.-H. Jun, *Learning across scales—multiscale methods for convolution neural networks*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[18] W. W. Hager and H. Zhang, *A new active set algorithm for box constrained optimization*, SIAM Journal on Optimization, 17 (2006), pp. 526–557.

[19] J. L. Herring, J. Nagy, and L. Ruthotto, *Gauss–Newton optimization for phase recovery from the bispectrum*, IEEE Transactions on Computational Imaging, (2019).

[20] Y. Hu, *Matrix Computations and Optimization for Spectral Computed Tomography*, PhD thesis, Emory University, 2019.

[21] Y. Hu, M. S. Andersen, and J. G. Nagy, *Spectral computed tomography with linearization and preconditioning*, SIAM Journal on Scientific Computing, 41 (2019), pp. S370–S389.

[22] Y. Hu, J. G. Nagy, J. Zhang, and M. S. Andersen, *Nonlinear optimization for mixed attenuation polyenergetic image reconstruction*, Inverse Problems, 35 (2019), p. 064004.

[23] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, *Extreme learning machine: theory and applications*, Neurocomputing, 70 (2006), pp. 489–501.

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

[25] D. Kim, S. Sra, and I. S. Dhillon, *Tackling box-constrained optimization via a new projected quasi-Newton approach*, SIAM Journal on Control and Optimization, 32 (2010), pp. 3548–3563.

[26] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, (1950), pp. 255–282.

[27] G. Landi and E. L. Piccolomini, *A projected Newton-CG method for nonnegative astronomical image deblurring*, Numerical Algorithms, 48 (2008), pp. 279–300.

[28] G. Landi and E. L. Piccolomini, *An improved Newton projection method for nonnegative deblurring of Poisson-corrupted images with Tikhonov regularization*, Numerical Algorithms, 60 (2012), pp. 169–188.

[29] G. Landi and E. L. Piccolomini, *Nptool: a MATLAB software for nonnegative image restoration with Newton projection methods*, Numerical Algorithms, 62 (2013), pp. 487–504.

[30] Y. LeCun, *The MNIST database of handwritten digits*, http://yann. lecun. com/exdb/mnist/, (1998).

[31] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM Journal on Optimization, 24 (2014), pp. 1420–1443.

[32] P. R. McGillivray, *Forward modeling and inversion of DC resistivity and MMR data*, PhD thesis, University of British Columbia, 1992.

[33] J. L. Morales and J. Nocedal, *Remark on "algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization"*, ACM Transactions on Mathematical Software (TOMS), 38 (2011), pp. 1–4.

[34] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[35] L. Ruthotto and E. Haber, *Deep neural networks motivated by partial differential equations*, Journal of Mathematical Imaging and Vision, (2019), pp. 1–13.

[36] L. Ruthotto, E. Treister, and E. Haber, *jinv–a flexible julia package for PDE parameter estimation*, SIAM Journal on Scientific Computing, 39 (2017), pp. S702–S722.

[37] M. Schmidt, E. Berg, M. Friedlander, and K. Murphy, *Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm*, in Artificial Intelligence and Statistics, 2009, pp. 456–463.

[38] M. Schmidt, D. Kim, and S. Sra, *Projected Newton-type methods in machine learning*, Optimization for Machine Learning, (2012).

[39] C. Wang, G. Ballard, R. Plemmons, and S. Prasad, *Joint 3D localization and classification of space debris using a multispectral rotating point spread function*, Applied Optics, 58 (2019), pp. 8598–8611.

[40] C. Wang, R. Chan, M. Nikolova, R. Plemmons, and S. Prasad, *Nonconvex optimization for 3-dimensional point source localization using a rotating point spread function*, SIAM Journal on Imaging Sciences, 12 (2019), pp. 259–286.

[41] S. Wu Fung, *Large-Scale Parameter Estimation in Geophysics and Machine Learning*, PhD thesis, Emory University, 2019, https://search.proquest.com/docview/2374443204?accountid=14512.

[42] S. Wu Fung and Z. W. Di, *Multigrid optimization for large-scale ptychographic phase retrieval*, SIAM Journal on Imaging Sciences, 13 (2020), pp. 214–233.

[43] S. Wu Fung and L. Ruthotto, *A multiscale method for model order reduction in PDE parameter estimation*, Journal of Computational and Applied Mathematics, 350 (2019), pp. 19–34.

[44] S. Wu Fung and L. Ruthotto, *An uncertainty-weighted asynchronous ADMM method for parallel PDE parameter estimation*, SIAM Journal on Scientific Computing, 41 (2019), pp. S129–S148.