Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Block Paige-Saunders Bidiagonalization Framework for Large-Scale Nuclear Norm Regularized Least Squares Problems

Bo Feng

bofeng@yzu.edu.cn

Yangzhou University

Research Article

Keywords: Nuclear norm regularization, Block PSB process, Krylov subspace, Restarted algorithm

Posted Date: July 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10284990/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Block Paige-Saunders Bidiagonalization Framework for Large-Scale Nuclear Norm Regularized Least Squares Problems

Bo Feng*

July 7, 2026

Abstract

This paper proposes a framework for solving large-scale nuclear norm regularized least squares (NNRLS) problems based on the block Paige-Saunders bidiagonalization (PSB) process. First, the original high-dimensional NNRLS problem is projected onto the block Krylov subspace generated by the k-step block PSB process, which reduces the original large-scale problem to a small-scale NNRLS problem that is then efficiently solved via the primal accelerated proximal gradient (PAPG) method. Second, a convergence analysis is conducted for the proposed block PSB method: it is proven that the distance between the global optimal solution and the block Krylov search subspace converges linearly; when the optimal solution is of full column rank, the convergence rate of the approximate optimal value is comparable to that of the conjugate gradient method and generally outperforms that of proximal gradient method. To alleviate the memory consumption and computational burden arising from the increasing dimension of the search subspace, a restarted block PSB algorithm is developed in this paper. Numerical experiments verify that the proposed algorithm exhibits superior computational efficiency and robustness in the vast majority of test cases.

Key words: Nuclear norm regularization; Block PSB process; Krylov subspace; Restarted algorithm

AMS: 65F50, 65F99, 90C06, 90C25

*School of Mathematics, Yangzhou University, Yangzhou, 225002, Jiangsu, China. E-mail: bofeng@yzu.edu.cn. **Funding:** The work of this author was supported by the Youth Fund of the National Natural Science Foundation of China (Grant No. 12501529), the Basic Research Program of Jiangsu Province (Youth Program of Natural Science Foundation of Jiangsu Province, Grant No. BK20250892), and the China Postdoctoral Science Foundation (Grant No. 2025M783141).

1 Introduction

In many practical scenarios, we aim to find a low-rank matrix $\boldsymbol{X} \in \mathbb{R}^{n\times\ell}$ that minimizes the error between $\boldsymbol{CX}$ and $\boldsymbol{D}$, with $\boldsymbol{C} \in \mathbb{R}^{m\times n}$, $\boldsymbol{D} \in \mathbb{R}^{m\times\ell}$. A widely used effective approach is to introduce the nuclear norm $\|\boldsymbol{X}\|_*$ as a regularization term in the objective function [17], where nuclear norm is defined as the sum of the singular values of $\boldsymbol{X}$. In this paper, we are interested in solving the following large-scale nuclear norm regularized least squares (NNRLS) problem [12, 17, 28]:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{n\times\ell}} \left\{ f(\boldsymbol{X}) := \frac{1}{2}\|\boldsymbol{CX}-\boldsymbol{D}\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{X}\|_* \right\}, \tag{1}$$

where $\mu > 0$ and $\|\cdot\|_{\mathsf{F}}$ denotes Frobenius norm.

The NNRLS (1) arises from many practical problems. The convex multi-task learning [3, 17] model for linear predictors $g_s(\boldsymbol{w}) = \boldsymbol{x}_s^{\mathsf{T}}\boldsymbol{w}$ with nuclear norm regularization can be written as the following optimization problem

$$\min_{\{\boldsymbol{x}_s\}} \sum_{s=1}^{\ell} \left(\sum_{i=1}^{m} L(\boldsymbol{w}_i^{\mathsf{T}}\boldsymbol{x}_s, y_i^{(s)}) \right) + \mu\|\boldsymbol{X}\|_*.$$

For the least squares loss $L(s,t) = \frac{1}{2}(s-t)^2$, this optimization problem reduces to NNRLS (1) with $\boldsymbol{C} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_m)^{\mathsf{T}}$, and the (i,j)-th entry of $\boldsymbol{D}$ is $y_i^{(j)} \in \{1,-1\}$. Here, $\{(\boldsymbol{w}_1, y_1^{(s)}),\ldots,(\boldsymbol{w}_m, y_m^{(s)})\} \subset \mathbb{R}^n \times \{-1,1\}, s = 1,\ldots,\ell$, is training data. Canonical Correlation Analysis (CCA) is a widely used technique in machine learning. However, CCA suffers from the small sample size problem when dealing with high-dimensional data. Li, Lu, Shu and Liu proposed a trace norm regularized CCA model [11], which not only addresses the small sample size problem but also reveals the underlying structures among target classes. Specifically, owing to the appealing properties of the trace norm, the model can characterize the intrinsic dimensionality of the transformation matrix. The mathematical formulation of this model is exactly NNRLS (1) [11]. In multivariate linear regression, given t observations on q responses $\boldsymbol{b} = (b_1,\ldots,b_q)^{\mathsf{T}}$ and p explanatory variables $\boldsymbol{a} = (a_1,\ldots,a_p)^{\mathsf{T}}$, and

$$\boldsymbol{B} = \boldsymbol{AX} + \boldsymbol{E},$$

where $\boldsymbol{B} = (\boldsymbol{b}^1,\ldots,\boldsymbol{b}^t)^{\mathsf{T}} \in \mathbb{R}^{t\times q}$ and $\boldsymbol{A} = (\boldsymbol{a}^1,\ldots,\boldsymbol{a}^t)' \in \mathbb{R}^{t\times p}$ consist of the data of responses and explanatory variables, respectively, $\boldsymbol{X} \in \mathbb{R}^{p\times q}$ is the coefficient matrix, $\boldsymbol{E}$ is the regression noise. M. Yuan, A. Ekici, Z. Lu, and R. Monteiro proposed a model that can simultaneously choose the number of factors, determine the factors, and estimate the factor loading matrix [28]. This model can be formulated as

$$\min_{\|\boldsymbol{X}\|_*\le\omega} \|\boldsymbol{AX}-\boldsymbol{B}\|_{\mathsf{F}}^2, \tag{2}$$

which is equivalent to NNRLS (1). In the color image denoising, a quaternion weighted nuclear norm minimization (QWNNM) method was proposed [7, 8, 27],

which requires solving the following problem:

$$\min_{\dot{\boldsymbol{X}}} \left\{ \frac{1}{2}\|\dot{\boldsymbol{C}}\dot{\boldsymbol{X}} - \dot{\boldsymbol{D}}\|_{\mathsf{F}}^2 + \mu\|\dot{\boldsymbol{X}}\|_{w,*} \right\},$$

where $\dot{\boldsymbol{C}}$, $\dot{\boldsymbol{D}}$, and $\dot{\boldsymbol{X}}$ are quaternion matrices, and $\|\cdot\|_{w,*}$ denotes the weighted nuclear norm [7, 8, 27]. The QWNNM problem is very similar to NNRLS (1), and studying NNRLS helps solve the QWNNM. In addition, nuclear norm regularization problems have also been extensively studied [5, 10, 16, 18, 20].

Since the nuclear norm is nonsmooth, it complicates solving NNRLS (1). The nuclear norm minimization problem can be reformulated as a semidefinite program (SDP) [4, 21]. However, for large-scale NNRLS problems, SDP solvers incur a high computational cost. To overcome the nonsmoothness, existing works have reformulated NNRLS (1) into a nonconvex smooth form, which is then solved by the conjugate gradient method or the alternating minimization method [19, 25, 26]. Nevertheless, such a nonconvex reformulation introduces spurious stationary points and cannot guarantee convergence to a global minimum. Proximal gradient method are also widely used to solve NNRLS (1). The convergence rate of proximal gradient method is typically $\mathcal{O}(k^{-1})$ or $\mathcal{O}(k^{-2})$ [2], where k denotes the iteration count. Pong, Tseng, Ji, and Ye proposed new equivalent primal and dual forms of NNRLS (1) [17], including a reduced dual formulation that minimizes a convex quadratic function over an operator-norm ball. The reduced dual problem can be solved by gradient-projection methods, where an SVD of an $n \times \ell$ matrix is required at each projection step. In [12], Lu, Monteiro, and Yuan explore a variant of Nesterov's smooth method [13, 14] due to Tseng [23], as well as interior-point methods for solving (2).

In this paper, we propose an efficient subspace method based on the block Paige-Saunders bidiagonalization (PSB) process to solve the large-scale NNRLS problem. The core idea of our method is to project the original high-dimensional NNRLS problem onto a low-dimensional block Krylov subspace generated by the block PSB process, thereby reducing the original problem to a small-scale NNRLS problem that can be solved efficiently by the primal accelerated proximal gradient (PAPG) method [17]. We conduct a comprehensive convergence analysis for the proposed block PSB method, proving that the distance between the global optimal solution $\boldsymbol{X}_{\text{opt}}$ and the block Krylov subspace converges linearly with respect to the iteration number k. Notably, when the optimal solution $\boldsymbol{X}_{\text{opt}}$ is of full column rank, the convergence speed of $f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\text{opt}})$ (where $\boldsymbol{X}_k$ denotes the approximate solution) is comparable to that of the conjugate gradient (CG) method, which is faster than that of the proximal gradient method. To address the issues of increasing memory consumption and computational complexity caused by the growing dimension of the block Krylov subspace as k increases, we further develop a restarted block PSB algorithm that constrains the subspace dimension to a fixed size in each restart phase. Numerical experiments on synthetic sparse data and real-world high-dimensional matrices verify that the proposed restarted block PSB algorithm is effective. Compared with gradient-based methods, Krylov subspace methods generally exhibit faster convergence rates.

This paper is organized as follows. In Section 2, we present some preliminary results on NNRLS and introduce the block PSB process. In Section 3, we propose a block PSB method for solving the large-scale NNRLS problem, and the convergence analysis is presented in subsection 3.1. A restarted block PSB method is introduced in Section 4. We present numerical experiments in Section 5 to demonstrate the effectiveness of the proposed algorithm. Some concluding remarks are given in Section 6. Throughout this paper, we denote by $(\cdot)^\top$ the transpose of a matrix or vector, by $\|\cdot\|_2$ the 2-norm of a matrix or vector, and by $\mathcal{R}(\boldsymbol{A})$ the range space of a matrix $\boldsymbol{A}$. In this paper, $\boldsymbol{A} \succcurlyeq \boldsymbol{0}$ ($\boldsymbol{A} \succ \boldsymbol{0}$) indicates that $\boldsymbol{A}$ is a symmetric positive semidefinite (positive definite) matrix. Let $\boldsymbol{0}$ and $\boldsymbol{I}$ denote the zero vector/matrix and identity matrix, respectively, whose dimensions are clear from the context.

2 Preliminaries

2.1 Optimality conditions

In order to describe the optimality conditions for the NNRLS (1), we first characterize the subgradients of the nuclear norm. Recall that for a convex function $\varphi : \mathbb{E} \to \mathbb{R}$, the subdifferential of φ at $x \in \mathbb{E}$ is the compact convex set [2, 24]

$$\partial\varphi(x) := \{d \in \mathbb{E} : \varphi(y) \geq \varphi(x) + \langle d, y - x\rangle,\ \forall y \in \mathbb{E}\}\,, \tag{3}$$

where $\mathbb{E}$ is a Euclidean space, and $\langle\cdot,\cdot\rangle$ denotes the inner product on $\mathbb{E}$. Let $\boldsymbol{X}_{\rm opt}$ denotes the global optimal solution of NNRLS (1). Thus, the subdifferential of the nuclear norm at $\boldsymbol{X}_{\rm opt}$ is [2, 24]

$$\partial\|\boldsymbol{X}_{\rm opt}\|_* = \Big\{\boldsymbol{S} \in \mathbb{R}^{n\times\ell} : \|\boldsymbol{Y}\|_* \geq \|\boldsymbol{X}_{\rm opt}\|_* + \mathsf{trace}\left[(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{S}\right],\ \forall \boldsymbol{Y} \in \mathbb{R}^{n\times\ell}\Big\}. \tag{4}$$

Consider the SVD decomposition of $\boldsymbol{X}_{\rm opt}$:

$$\begin{aligned}
&\boldsymbol{X}_{\rm opt} = \begin{pmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{pmatrix}\begin{pmatrix}\boldsymbol{\Sigma}_1 & \\ & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{V}_1^{\mathsf{T}} \\ \boldsymbol{V}_2^{\mathsf{T}}\end{pmatrix} = \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^{\mathsf{T}} \in \mathbb{R}^{n\times\ell},\\
&\boldsymbol{\Sigma}_1 = diag(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r},\ \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,
\end{aligned}$$

where $r = \mathrm{rank}(\boldsymbol{X}_{\rm opt})$. Therefore, $\partial\|\boldsymbol{X}_{\rm opt}\|_*$ can be characterized via its SVD decomposition [24, 18]:

$$\partial\|\boldsymbol{X}_{\rm opt}\|_* = \Big\{\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}} \in \mathbb{R}^{n\times\ell} : \|\boldsymbol{T}\|_2 \leq 1\Big\}.$$

Specifically, if $\mathrm{rank}(\boldsymbol{X}_{\rm opt}) = \ell$, then

$$\partial\|\boldsymbol{X}_{\rm opt}\|_* = \boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} = \boldsymbol{X}_{\rm opt}(\boldsymbol{X}_{\rm opt}{}^{\mathsf{T}}\boldsymbol{X}_{\rm opt})^{-\frac{1}{2}}.$$

Notice that $\widetilde{f}(\boldsymbol{X}) = \|\boldsymbol{C}\boldsymbol{X} - \boldsymbol{D}\|_{\mathsf{F}}^2$ is convex. Thus, $\boldsymbol{X}_{\rm opt}$ is a global optimal solution of (1) if and only if (cf. [2, Theorem 3.72] and [2, Theorem 3.35])

$$-(\boldsymbol{C}^{\mathcal{T}}\boldsymbol{C}\boldsymbol{X}_{\rm opt} - \boldsymbol{C}^{\mathcal{T}}\boldsymbol{D}) = -\nabla\widetilde{f}(\boldsymbol{X}_{\rm opt}) \in \partial(\mu\|\boldsymbol{X}_{\rm opt}\|_*) = \mu\cdot\partial\|\boldsymbol{X}_{\rm opt}\|_*.$$

From the above analysis, the following theorem can be readily derived.

Theorem 1 *The $\boldsymbol{X}_{\rm opt}$ is a global optimal solution of (1) if and only if*

$$\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\boldsymbol{X}_{\rm opt} - \boldsymbol{C}^{\mathsf{T}}\boldsymbol{D} = -\mu(\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}}) \quad \textit{with} \quad \|\boldsymbol{T}\|_2 \le 1. \tag{5}$$

2.2 The block Paige-Saunders bidiagonalization (PSB) process

The block PSB process [1] is a natural extension of the PSB process, which was originally proposed for solving least squares problems [15]. Let $\boldsymbol{D} = \boldsymbol{P}_1\boldsymbol{R}_1$ be the economy-size QR decomposition of $\boldsymbol{D}$ with $\boldsymbol{P}_1 \in \mathbb{R}^{n\times\ell}$. We can use the k-step block PSB process to generate an orthonormal basis for the block Krylov subspace

$$\mathcal{K}_k := \mathcal{K}_k(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}, \boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1) = \operatorname{span}\left\{\boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1, (\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})\boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1, \ldots, (\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})^{k-1}\boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1\right\}.$$

The following relation holds for the block PSB process

$$\boldsymbol{C}\boldsymbol{W}_k = \boldsymbol{U}_{k+1}\boldsymbol{B}_{k+1}, \quad \boldsymbol{C}^{\mathsf{T}}\boldsymbol{U}_k = \boldsymbol{W}_k\boldsymbol{B}_k^{\mathsf{T}}, \tag{6}$$

where the matrices $\boldsymbol{U}_{k+1} = (\boldsymbol{P}_1, \ldots, \boldsymbol{P}_{k+1}) \in \mathbb{R}^{m\times\ell(k+1)}$ and $\boldsymbol{W}_k = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_k) \in \mathbb{R}^{n\times k\ell}$ have orthonormal columns and

$$\boldsymbol{B}_{k+1} = \left(\begin{array}{c|c} & \\ \boldsymbol{B}_k & \\ & \\ \hline \multicolumn{2}{c}{\boldsymbol{R}_{k+1}\boldsymbol{E}_k^{\mathsf{T}}} \end{array}\right) = \left(\begin{array}{ccccc} \boldsymbol{L}_1 & & & & \\ \boldsymbol{R}_2 & \boldsymbol{L}_2 & & & \\ & \ddots & \ddots & & \\ & & & \boldsymbol{R}_k & \boldsymbol{L}_k \\ \hline & & & & \boldsymbol{R}_{k+1} \end{array}\right) \in \mathbb{R}^{\ell(k+1)\times\ell k}$$

is lower block bidiagonal with lower triangular diagonal blocks $\boldsymbol{L}_j \in \mathbb{R}^{\ell\times\ell}$ and upper triangular subdiagonal blocks $\boldsymbol{R}_j \in \mathbb{R}^{\ell\times\ell}$, where $\boldsymbol{E}_k = \begin{pmatrix}\boldsymbol{0} & \boldsymbol{I}_\ell\end{pmatrix} \in \mathbb{R}^{\ell\times\ell k}$. Moreover,

$$\mathcal{R}(\boldsymbol{W}_k) = \mathcal{K}_k, \quad \mathcal{R}(\boldsymbol{U}_k) = \mathcal{K}_k(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}, \boldsymbol{P}_1). \tag{7}$$

The block PSB process is listed in Algorithm 1.

3 A block PSB method for NNRLS

In this section, we propose a block PSB method to solve NNRLS (1). To this end, our method addresses the minimization problem at step k:

$$\begin{aligned}
\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_k} f(\boldsymbol{X}) &= \min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_k} \|\boldsymbol{C}\boldsymbol{X} - \boldsymbol{D}\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{X}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{\ell k\times\ell}} \|\boldsymbol{C}\boldsymbol{W}_k\boldsymbol{Y} - \boldsymbol{D}\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{W}_k\boldsymbol{Y}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{\ell k\times\ell}} \|\boldsymbol{U}_{k+1}\boldsymbol{B}_{k+1}\boldsymbol{Y} - \boldsymbol{P}_1\boldsymbol{R}_1\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{\ell k\times\ell}} \left\|\boldsymbol{B}_{k+1}\boldsymbol{Y} - \boldsymbol{R}_1^{(k)}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_*,
\end{aligned}$$

Algorithm 1 The block PSB process [1].

Input: $\boldsymbol{C}$, $\boldsymbol{D}$ and $It_{\max}$;
Output: $\boldsymbol{B}_{k+1}$, $\boldsymbol{U}_{k+1}$, and $\boldsymbol{W}_k$.
1: Compute the economy-size QR factorization $\boldsymbol{D} = \boldsymbol{P}_1\boldsymbol{R}_1$;
2: $\boldsymbol{F}_1 = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1$;
3: Compute the economy-size QR factorization $\boldsymbol{F}_1 = \boldsymbol{Z}_1\boldsymbol{L}_1^{\mathsf{T}}$;
4: **for** $k = 1, \ldots, It_{\max}$ **do**
5: $\quad\boldsymbol{G}_k = \boldsymbol{C}\boldsymbol{Z}_k - \boldsymbol{P}_k\boldsymbol{L}_k$;
6: $\quad$Compute the economy-size QR factorization $\boldsymbol{G}_k = \boldsymbol{P}_{k+1}\boldsymbol{R}_{k+1}$;
7: $\quad\boldsymbol{F}_{k+1} = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_{k+1} - \boldsymbol{Z}_k\boldsymbol{R}_{k+1}^{\mathsf{T}}$;
8: $\quad$Compute the economy-size QR factorization $\boldsymbol{F}_{k+1} = \boldsymbol{Z}_{k+1}\boldsymbol{L}_{k+1}^{\mathsf{T}}$;
9: **end for**

where $\boldsymbol{R}_1^{(k)} = (\boldsymbol{R}_1^{\mathsf{T}} \;\; \boldsymbol{0})^{\mathsf{T}} \in \mathbb{R}^{\ell(k+1)\times\ell}$. Let

$$\boldsymbol{Y}_k = \arg\min_{\boldsymbol{Y}\in\mathbb{R}^{\ell k\times\ell}} \left\|\boldsymbol{B}_{k+1}\boldsymbol{Y} - \boldsymbol{R}_1^{(k)}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_*, \tag{8}$$

we make use of

$$\boldsymbol{X}_k = \boldsymbol{W}_k\boldsymbol{Y}_k = \arg\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_k} f(\boldsymbol{X}). \tag{9}$$

as an approximation to the optimal solution $\boldsymbol{X}_{\text{opt}}$. We thus reduce the large-scale NNRLS (1) to a small-size $\ell(k+1)\times\ell k$ one, which can then be efficiently solved in practice using the PAPG method [17] applied to (8). The proposed algorithm is summarized in Algorithm 2.

3.1 Convergence analysis

The aim of this section is to prove the convergence of Algorithm 2. To this end, we require three lemmas.

Lemma 1 *Let $\boldsymbol{X}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times\ell}$, $\boldsymbol{Z} \in \mathbb{R}^{q\times n}$. Then*

(i) [22, Theorem 3.9] $\|\boldsymbol{Z}\boldsymbol{X}\|_* \le \|\boldsymbol{Z}\|_2\|\boldsymbol{X}\|_*$;

(ii) [9, Theorem 3] If $rank(\boldsymbol{X}) = rank(\boldsymbol{Y}) = \ell$. Then

$$\left\|\boldsymbol{X}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X})^{-\frac{1}{2}} - \boldsymbol{Y}(\boldsymbol{Y}^{\mathsf{T}}\boldsymbol{Y})^{-\frac{1}{2}}\right\|_{\mathsf{F}} \le \frac{\sqrt{5}\|\boldsymbol{X}-\boldsymbol{Y}\|_{\mathsf{F}}}{\max\{\sigma_\ell(\boldsymbol{X}), \sigma_\ell(\boldsymbol{Y})\}}, \tag{10}$$

where $\sigma_i(\cdot)$ denotes the i-th singular value of a matrix.

Lemma 2 *If $\text{rank}(\boldsymbol{X}_{\text{opt}}) = \ell$, i.e., $\sigma_\ell > 0$. Then*

$$\|\boldsymbol{Y}\|_* \le \|\boldsymbol{X}_{\text{opt}}\|_* + \mathsf{trace}\left[(\boldsymbol{Y}-\boldsymbol{X}_{\text{opt}})^{\mathsf{T}}\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}}\right] + \frac{\sqrt{5}\|\boldsymbol{Y}-\boldsymbol{X}_{\text{opt}}\|_{\mathsf{F}}^2}{2\sigma_\ell}, \forall \boldsymbol{Y} \in \mathbb{R}^{n\times\ell}.$$

Algorithm 2 A block PSB method for NNRLS (1).

Input: $\boldsymbol{C}$, $\boldsymbol{D}$ and $It_{\max}$;
Output: $\boldsymbol{X}_k$;
1: Compute the economy-size QR factorization $\boldsymbol{D} = \boldsymbol{P}_1\boldsymbol{R}_1$;
2: $\boldsymbol{F}_1 = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_1$;
3: Compute the economy-size QR factorization $\boldsymbol{F}_1 = \boldsymbol{Z}_1\boldsymbol{L}_1^{\mathsf{T}}$;
4: **for** $k = 1, \ldots, It_{\max}$ **do**
5: $\quad \boldsymbol{G}_k = \boldsymbol{C}\boldsymbol{Z}_k - \boldsymbol{P}_k\boldsymbol{L}_k$;
6: $\quad$ Compute the economy-size QR factorization $\boldsymbol{G}_k = \boldsymbol{P}_{k+1}\boldsymbol{R}_{k+1}$;
7: $\quad$ Solve the reduced NNRLS

$$\boldsymbol{Y}_k = \arg\min_{\boldsymbol{Y}\in\mathbb{R}^{\ell k\times\ell}} \left\|\boldsymbol{B}_{k+1}\boldsymbol{Y} - \boldsymbol{R}_1^{(k)}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_*;$$

8: $\quad$ **if** the convergence criterion is satisfied
9: $\quad\quad$ Output $\boldsymbol{X}_k = \boldsymbol{W}_k\boldsymbol{Y}_k$;
10: $\quad$ **end if**
11: $\quad \boldsymbol{F}_{k+1} = \boldsymbol{C}^{\mathsf{T}}\boldsymbol{P}_{k+1} - \boldsymbol{Z}_k\boldsymbol{R}_{k+1}^{\mathsf{T}}$;
12: $\quad$ Compute the economy-size QR factorization $\boldsymbol{F}_{k+1} = \boldsymbol{Z}_{k+1}\boldsymbol{L}_{k+1}^{\mathsf{T}}$;
13: **end for**

Proof 1 *Let* $\psi(t) = \|\boldsymbol{X}_{\rm opt} + t(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})\|_*$ *with* $t \in [0,1]$. *Notice that*

$$\|\boldsymbol{X}_{\rm opt}\|_* \le \frac{1}{\mu}f(\boldsymbol{X}_{\rm opt}) \le \frac{1}{\mu}f(\boldsymbol{0}) = \frac{\|\boldsymbol{B}\|_F^2}{2\mu}.$$

It follows that

$$\begin{aligned}|\psi(t_1) - \psi(t_2)| &\le \left|\|\boldsymbol{X}_{\rm opt} + t_1(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})\|_* - \|\boldsymbol{X}_{\rm opt} + t_2(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})\|_*\right| \\ &\le |t_1 - t_2|\|\boldsymbol{Y} - \boldsymbol{X}_{\rm opt}\|_* \le \alpha|t_1 - t_2|,\ \forall t_1, t_2 \in [0,1],\end{aligned}$$

where $\alpha = \|\boldsymbol{Y}\|_* + \frac{\|\boldsymbol{B}\|_F^2}{2\mu} < \infty$. *Hence,* $\psi(t)$ *is Lipschitz continuous. Let* $\boldsymbol{Z}(t) = \boldsymbol{X}_{\rm opt} + t(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})$. *Consider the SVD of* $\boldsymbol{Z}(t)$*:*

$\boldsymbol{Z}(t) = \boldsymbol{U}_1(t)\boldsymbol{\Sigma}(t)(\boldsymbol{V}(t))^T$ *where* $\boldsymbol{U}_1(t) \in \mathbb{R}^{n\times\ell}$, $\boldsymbol{\Sigma}(t) \in \mathbb{R}^{\ell\times\ell}$, *and* $\boldsymbol{V}(t) \in \mathbb{R}^{\ell\times\ell}$.

Hence,

$$\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^T \in \partial\|\boldsymbol{Z}(t)\|_*,\ \forall t \in [0,1],$$

and by (4), *we have*

$$\|\boldsymbol{Z}(\tilde{t})\|_* \ge \|\boldsymbol{Z}(t)\|_* + (\tilde{t} - t)\mathit{trace}\big[(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})^T\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^T\big],\ \forall \tilde{t} \in [0,1].$$

That is,

$$\psi(\tilde{t}) \ge \psi(t) + (\tilde{t} - t)\mathit{trace}\big[(\boldsymbol{Y} - \boldsymbol{X}_{\rm opt})^T\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^T\big],\ \forall \tilde{t} \in [0,1].$$

It follows from (3) *that*

$$\mathsf{trace}\big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}\big]\in\partial\psi(t).$$

By [8, p. 24 eq. (4.2.6)],

$$\begin{aligned}\|\boldsymbol{Y}\|_*-\|\boldsymbol{X}_{\rm opt}\|_*=\psi(1)-\psi(0)&=\int_0^1\mathsf{trace}\big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}\big]\mathrm{d}t\\&=\int_0^1\mathsf{trace}\Big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\big(\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}-\boldsymbol{U}_1(0)(\boldsymbol{V}(0))^{\mathsf{T}}\big)\Big]\mathrm{d}t\\&\quad+\mathsf{trace}\big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{U}_1(0)(\boldsymbol{V}(0))^{\mathsf{T}}\big].\end{aligned}$$

Notice that

$$\begin{aligned}&\boldsymbol{U}_1(0)(\boldsymbol{V}(0))^{\mathsf{T}}=\boldsymbol{X}_{\rm opt}(\boldsymbol{X}_{\rm opt}^{\mathsf{T}}\boldsymbol{X}_{\rm opt})^{-\frac{1}{2}}=\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}},\ \text{and}\\&\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}=\boldsymbol{Z}_\epsilon(t)\Big[(\boldsymbol{Z}_\epsilon(t))^{\mathsf{T}}\boldsymbol{Z}_\epsilon(t)\Big]^{-\frac{1}{2}},\end{aligned}$$

where

$$\boldsymbol{Z}_\epsilon(t)=\boldsymbol{U}_1(t)\left(\boldsymbol{\Sigma}(t)+\epsilon\boldsymbol{I}_\ell\right)(\boldsymbol{V}(t))^{\mathsf{T}}=\boldsymbol{Z}(t)+\epsilon\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}},\ \forall\epsilon>0.$$

It follows that

$$\begin{aligned}&\big|\|\boldsymbol{Y}\|_*-\|\boldsymbol{X}_{\rm opt}\|_*-\mathsf{trace}\big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}}\big]\big|\\=&\left|\int_0^1\mathsf{trace}\Big[(\boldsymbol{Y}-\boldsymbol{X}_{\rm opt})^{\mathsf{T}}\big(\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}-\boldsymbol{U}_1(0)(\boldsymbol{V}(0))^{\mathsf{T}}\big)\Big]\mathrm{d}t\right|\\\le&\int_0^1\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F\cdot\|\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}-\boldsymbol{U}_1(0)(\boldsymbol{V}(0))^{\mathsf{T}}\|_F\,\mathrm{d}t\\\overset{(10)}{\le}&\frac{\sqrt{5}}{\sigma_\ell}\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F\cdot\int_0^1\|\boldsymbol{Z}_\epsilon(t)-\boldsymbol{X}_{\rm opt}\|_F\,\mathrm{d}t\\\le&\frac{\sqrt{5}}{\sigma_\ell}\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F\cdot\int_0^1 t\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F+\epsilon\|\boldsymbol{U}_1(t)(\boldsymbol{V}(t))^{\mathsf{T}}\|_F\,\mathrm{d}t\\=&\frac{\sqrt{5}}{\sigma_\ell}\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F\cdot\Big(\frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{X}_{\rm opt}\|_F+\epsilon\sqrt{\ell}\Big).\end{aligned}$$

Letting $\epsilon\to0^+$ *yields* (2).

For convenience, we let

$$\boldsymbol{H}=\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\ \text{ and }\ \boldsymbol{G}=\boldsymbol{C}^{\mathsf{T}}\boldsymbol{D}.$$

We now consider the distance between the optimal solution $\boldsymbol{X}_{\rm opt}$ and the search subspace $\mathcal{K}_k$.

Lemma 3 *Let*

$$\delta_k = \min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_k} \|\boldsymbol{X}_{\rm opt} - \boldsymbol{X}\|_{\mathsf{F}} = \|(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k^T)\boldsymbol{X}_{\rm opt}\|_{\mathsf{F}},\ k \geq 1.$$

Then we have

$$\delta_k \leq 2\|\boldsymbol{X}_{\rm opt}\|_{\mathsf{F}} \cdot \left(\frac{\sqrt{\varkappa}-1}{\sqrt{\varkappa}+1}\right)^k,$$

where $\varkappa$ is the 2-condition number of $\boldsymbol{H} + \frac{\mu}{\sigma_1}\boldsymbol{I}$.

Proof 2 *By (5),*

$$\boldsymbol{H}(\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1) - \boldsymbol{G}\boldsymbol{V}_1 = -\mu\boldsymbol{U}_1 = -\mu\boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^T\boldsymbol{V}_1\boldsymbol{\Sigma}_1^{-1} = -\mu(\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1)\boldsymbol{\Sigma}_1^{-1}.$$

It follows that

$$\boldsymbol{H}_i\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i = \boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i \quad \textit{for} \quad i = 1, 2, \ldots, r, \ \textit{where}\ \boldsymbol{H}_i := \boldsymbol{H} + \frac{\mu}{\sigma_i}\boldsymbol{I}.$$

Notice that

$$\mathcal{K}_k(\boldsymbol{H}_i, \boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i) = \mathcal{K}_k(\boldsymbol{H}, \boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i) = \mathcal{K}_k(\boldsymbol{H}, \boldsymbol{C}^T\boldsymbol{P}_1(\boldsymbol{R}_1\boldsymbol{V}_1\boldsymbol{e}_i)) \subseteq \mathcal{K}_k, i = 1, 2, \ldots, r.$$

Let $\mathcal{P}_j$ denote the set of polynomials of degree at most j. Then

$$\begin{aligned}
\min_{\boldsymbol{z}\in\mathcal{K}_k} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i - \boldsymbol{z}\|_2 &\leq \min_{\boldsymbol{z}\in\mathcal{K}_k(\boldsymbol{H}_i, \boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i)} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i - \boldsymbol{z}\|_2 \\
&= \min_{p\in\mathcal{P}_{k-1}} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i - p(\boldsymbol{H}_i)\boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i\|_2 \\
&\leq \min_{p\in\mathcal{P}_{k-1}} \|\boldsymbol{I} - p\left(\boldsymbol{H}_i\right)\boldsymbol{H}_i\| \left\|\boldsymbol{H}_i^{-1}\boldsymbol{G}\boldsymbol{V}_1\boldsymbol{e}_i\right\|_2 \\
&\leq \min_{p\in\mathcal{P}_{k-1}} \|\boldsymbol{I} - p\left(\boldsymbol{H}_i\right)\boldsymbol{H}_i\| \left\|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i\right\|_2.
\end{aligned}$$

Since $\boldsymbol{H}_i \succ \boldsymbol{0}$, it follows from [6, Section 3.1] that

$$\min_{p\in\mathcal{P}_{k-1}} \|\boldsymbol{I} - p\left(\boldsymbol{H}_i\right)\boldsymbol{H}_i\| \leq 2\left(\frac{\sqrt{\kappa_i}-1}{\sqrt{\kappa_i}+1}\right)^k \ \textit{where}\ \kappa_i = \left\|\boldsymbol{H}_i^{-1}\right\|_2 \|\boldsymbol{H}_i\|_2. \quad (11)$$

Hence,

$$\begin{aligned}
\delta_k^2 &= \|(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k^T)\boldsymbol{X}_{\rm opt}\|_F^2 = \|(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k^T)\boldsymbol{X}_{\rm opt}\begin{pmatrix}\boldsymbol{V}_1 & \boldsymbol{V}_2\end{pmatrix}\|_F^2 \\
&= \|(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k^T)\begin{pmatrix}\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1 & \boldsymbol{0}\end{pmatrix}\|_F^2 = \sum_{i=1}^r \|(\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{W}_k^T)\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i\|_2^2 \\
&= \sum_{i=1}^r \min_{\boldsymbol{z}\in\mathcal{K}_k} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i - \boldsymbol{z}\|_2^2 \leq 4\sum_{i=1}^r \left(\frac{\sqrt{\kappa_i}-1}{\sqrt{\kappa_i}+1}\right)^{2k} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\boldsymbol{e}_i\|_2^2 \\
&\leq 4\max_{1\leq i\leq r}\left(\frac{\sqrt{\kappa_i}-1}{\sqrt{\kappa_i}+1}\right)^{2k} \|\boldsymbol{X}_{\rm opt}\boldsymbol{V}_1\|_F^2 \leq 4\left(\frac{\sqrt{\varkappa}-1}{\sqrt{\varkappa}+1}\right)^{2k} \|\boldsymbol{X}_{\rm opt}\|_F^2,
\end{aligned}$$

which completes the proof.

Next, we analyze the convergence of both $f(\boldsymbol{X}_k)$ and $\boldsymbol{X}_k$.

Theorem 2 *Under the previous notation, the following assertions hold.*

(i) It holds that

$$0 \leq f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}) \leq \frac{1}{2}\|\boldsymbol{H}\|_2\delta_k^2 + \mu\sqrt{\ell}\delta_k.$$

(ii) If $\mathrm{rank}(\boldsymbol{X}_{\rm opt}) = \ell$, *then*

$$0 \leq f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}) \leq \frac{1}{2}\left(\|\boldsymbol{H}\|_2 + \frac{\sqrt{5}\mu}{\sigma_\ell}\right)\delta_k^2.$$

(iii) If $\mathrm{rank}(\boldsymbol{C}) = n$, *then*

$$\|\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt}\|_{\mathsf{F}} \leq \frac{\sigma_1(\boldsymbol{C})}{\sigma_\ell(\boldsymbol{C})}\sqrt{\left(\delta_k + \frac{2\mu\sqrt{\ell}}{\|\boldsymbol{H}\|_2}\right)\delta_k}.$$

(iv) If both $\boldsymbol{C}$ *and* $\boldsymbol{X}_{\rm opt}$ *have full column rank, then*

$$\|\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt}\|_{\mathsf{F}} \leq \frac{\sigma_1(\boldsymbol{C})}{\sigma_\ell(\boldsymbol{C})}\sqrt{1 + \frac{\sqrt{5}\mu}{\sigma_\ell\|\boldsymbol{H}\|_2}}\cdot\delta_k.$$

Proof 3 *Since* $f(\boldsymbol{X}) = \frac{1}{2}\mathit{trace}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}) - \mathit{trace}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{G}) + \frac{1}{2}\|\boldsymbol{D}\|_F^2 + \mu\|\boldsymbol{X}\|_*$, *it holds that*

$$\begin{aligned}
&f(\boldsymbol{X}) - f(\boldsymbol{X}_{\rm opt})\\
=&\frac{1}{2}\mathit{trace}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}) + \mathit{trace}\big[(\boldsymbol{X}_{\rm opt} - \boldsymbol{X})^{\mathsf{T}}\boldsymbol{G}\big]\\
&\quad + \mu(\|\boldsymbol{X}\|_* - \|\boldsymbol{X}_{\rm opt}\|_*) - \frac{1}{2}\mathit{trace}(\boldsymbol{X}_{\rm opt}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}_{\rm opt})\\
=&\frac{1}{2}\mathit{trace}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}) - \frac{1}{2}\mathit{trace}(\boldsymbol{X}_{\rm opt}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}_{\rm opt})\\
&\quad + \mathit{trace}\Big[(\boldsymbol{X}_{\rm opt} - \boldsymbol{X})^{\mathsf{T}}\big(\boldsymbol{G} - \mu\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} - \mu\boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}}\big)\Big]\\
&\quad + \mu\left(\|\boldsymbol{X}\|_* - \|\boldsymbol{X}_{\rm opt}\|_* - \mathit{trace}\Big[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^{\mathsf{T}}(\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}})\Big]\right)\\
\overset{(5)}{=}&\frac{1}{2}\mathit{trace}(\boldsymbol{X}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}) - \frac{1}{2}\mathit{trace}(\boldsymbol{X}_{\rm opt}^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}_{\rm opt}) + \mathit{trace}\Big[(\boldsymbol{X}_{\rm opt} - \boldsymbol{X})^{\mathsf{T}}\boldsymbol{H}\boldsymbol{X}_{\rm opt}\Big]\\
&\quad + \mu\left(\|\boldsymbol{X}\|_* - \|\boldsymbol{X}_{\rm opt}\|_* - \mathit{trace}\Big[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^{\mathsf{T}}(\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}})\Big]\right)\\
=&\frac{1}{2}\mathit{trace}\left[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^{\mathsf{T}}\boldsymbol{H}(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})\right]\\
&\quad + \mu\left(\|\boldsymbol{X}\|_* - \|\boldsymbol{X}_{\rm opt}\|_* - \mathit{trace}\Big[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^{\mathsf{T}}\big(\boldsymbol{U}_1\boldsymbol{V}_1^{\mathsf{T}} + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^{\mathsf{T}}\big)\Big]\right).
\end{aligned} \tag{12}$$

Let $\widetilde{\boldsymbol{X}}_k = \boldsymbol{W}_k\boldsymbol{W}_k^T\boldsymbol{X}_{\rm opt}$. *Then* $\|\widetilde{\boldsymbol{X}}_k\|_* \le \|\boldsymbol{X}_{\rm opt}\|_*$ *(see Lemma 1(i)). It follows from* (12) *that*

$$\begin{aligned}
&0 \le f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}) \overset{(9)}{\le} f(\widetilde{\boldsymbol{X}}_k) - f(\boldsymbol{X}_{\rm opt})\\
&\le \frac{1}{2}\mathit{trace}\left[(\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt})^T\boldsymbol{H}(\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt})\right] - \mu\mathit{trace}\Big[(\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt})^T\big(\boldsymbol{U}_1\boldsymbol{V}_1^T + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^T\big)\Big]\\
&\le \frac{1}{2}\mathit{trace}\left[(\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt})^T\boldsymbol{H}(\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt})\right] + \mu\|\boldsymbol{U}_1\boldsymbol{V}_1^T + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^T\|_F \cdot \|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F\\
&\le \frac{\|\boldsymbol{H}\|_2}{2}\|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F^2 + \mu\left\|\begin{pmatrix}\boldsymbol{I}_r & \\ & \boldsymbol{T}\end{pmatrix}\right\|_F \cdot \|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F\\
&\le \frac{\|\boldsymbol{H}\|_2}{2}\|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F^2 + \mu\sqrt{\ell}\cdot\|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F,
\end{aligned}$$

which implies assertion (i).

If $\mathrm{rank}(\boldsymbol{X}_{\rm opt}) = \ell$, *then*

$$\begin{aligned}
f(\boldsymbol{X}) - f(\boldsymbol{X}_{\rm opt}) =& \frac{1}{2}\mathit{trace}\left[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^T\boldsymbol{H}(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})\right]\\
&+ \mu\left(\|\boldsymbol{X}\|_* - \|\boldsymbol{X}_{\rm opt}\|_* - \mathit{trace}\Big[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^T\boldsymbol{U}_1\boldsymbol{V}^T\Big]\right)\\
\overset{\text{Lem.2}}{\le}& \frac{1}{2}\mathit{trace}\left[(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})^T\boldsymbol{H}(\boldsymbol{X} - \boldsymbol{X}_{\rm opt})\right] + \frac{\mu\sqrt{5}\|\boldsymbol{X} - \boldsymbol{X}_{\rm opt}\|_F^2}{2\sigma_\ell}\\
\le& \frac{1}{2}\left(\|\boldsymbol{H}\|_2 + \frac{\sqrt{5}\mu}{\sigma_\ell}\right)\|\boldsymbol{X} - \boldsymbol{X}_{\rm opt}\|_F^2.
\end{aligned}$$

It follows that

$$0 \le f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}) \le f(\widetilde{\boldsymbol{X}}_k) - f(\boldsymbol{X}_{\rm opt}) \le \frac{1}{2}\left(\|\boldsymbol{H}\|_2 + \frac{\sqrt{5}\mu}{\sigma_\ell}\right)\|\widetilde{\boldsymbol{X}}_k - \boldsymbol{X}_{\rm opt}\|_F^2,$$

which implies assertion (ii).

If $\mathrm{rank}(\boldsymbol{C}) = n$, *then*

$$\begin{aligned}
&\frac{\lambda_{\min}(\boldsymbol{H})}{2}\|\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt}\|_F^2 \le \frac{1}{2}\mathit{trace}(\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt})^T\boldsymbol{H}(\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt})\\
=& f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}) - \mu\left(\|\boldsymbol{X}_k\|_* - \|\boldsymbol{X}_{\rm opt}\|_* - \mathit{trace}\Big[(\boldsymbol{X}_k - \boldsymbol{X}_{\rm opt})^T\big(\boldsymbol{U}_1\boldsymbol{V}_1^T + \boldsymbol{U}_2\boldsymbol{T}\boldsymbol{V}_2^T\big)\Big]\right)\\
\le& f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\rm opt}),
\end{aligned}$$

where the last inequality follows from (4). *Combining* (i) *and* (ii) *yields* (iii) *and* (iv).

Remark 3.1 *The bound in Lemma 3 reveals a linear convergence rate of* δ_k, *and smaller* $\varkappa$ *translates into fast convergence of* δ_k. *Moreover, this upper bound is similar to the convergence upper bound of the conjugate gradient method.*

It can be seen from Theorem 2 that as $\delta_k \to 0$, $f(\boldsymbol{X}_k)$ *converges to* $f(\boldsymbol{X}_{\rm opt})$. *However, it is worth noting that when* $\boldsymbol{X}_{\rm opt}$ *is rank-deficient, the convergence*

rate of $f(\boldsymbol{X}_k)$ is the same as that of δ_k; whereas when $\boldsymbol{X}_{\rm opt}$ is of full rank, the convergence rate of $f(\boldsymbol{X}_k)$ is the same as that of δ_k^2. Consequently, Algorithm 2 performs more effectively when $\mathrm{rank}(\boldsymbol{X}_{\rm opt}) = \ell$.

In contrast, gradient-type algorithms (e.g., the proximal gradient method, the FISTA method) typically exhibit a convergence rate of $\mathcal{O}(k^{-1})$ or $\mathcal{O}(k^{-2})$ [2, Chapter 8 and Chapter 10]. Thus, if $\boldsymbol{H} + \frac{\mu}{\sigma_1}\boldsymbol{I}$ is well-conditioned, the convergence rate of Algorithm 2 will be significantly higher than that of gradient-type algorithms. In fact, if $\boldsymbol{C}$ is well-conditioned or $\frac{\mu}{\sigma_1}$ is not small, $\boldsymbol{H} + \frac{\mu}{\sigma_1}\boldsymbol{I}$ will be a well-conditioned matrix. For μ large, the solution $\boldsymbol{X}_{\rm opt}$ is near $\boldsymbol{0}$ [17, p. 3481]. In this case, $\frac{\mu}{\sigma_1}\boldsymbol{I}$ will dominate in $\boldsymbol{H} + \frac{\mu}{\sigma_1}\boldsymbol{I}$, and $\varkappa$ will approach 1. This implies that Algorithm 2 is effective when μ is large.

4 Restart method

In Algorithm 2, the number of columns of matrix $\boldsymbol{W}_k$ increases as k grows, resulting in increased memory consumption. Furthermore, the dimension of the reduced NNRLS problem (8) also grows with k, which significantly increases the time required to compute $\boldsymbol{X}_k$, thereby rendering Algorithm 2 computationally inefficient. To address these issues, it is necessary to constrain the dimension of $\mathcal{K}_k$. Accordingly, to enhance the practical utility of Algorithm 2, a restarted block PSB method is introduced in this section. The proposed algorithm is presented in Algorithm 3.

Next we describe how to solve NNRLS (15). Let $\begin{pmatrix}\boldsymbol{G} & \boldsymbol{X}^{(i)}\end{pmatrix} = \boldsymbol{P}_1^{(i)}\boldsymbol{R}_1^{(i)}$ be the economy-size QR decomposition of $\begin{pmatrix}\boldsymbol{G} & \boldsymbol{X}^{(i)}\end{pmatrix}$ with $\boldsymbol{P}_1^{(i)} \in \mathbb{R}^{n\times\tilde{t}}, \tilde{t} = 2\ell$. Similar to (6) and (7), we applying the s-step block PSB process (see Algorithm 1) to $\boldsymbol{C}^{\mathsf{T}}$ and $\boldsymbol{P}_1^{(i)}$ yields

$$\boldsymbol{C}^{\mathsf{T}}\boldsymbol{W}_s^{(i)} = \boldsymbol{U}_{s+1}^{(i)}\boldsymbol{B}_{s+1}^{(i)}, \quad \boldsymbol{C}\boldsymbol{U}_s^{(i)} = \boldsymbol{W}_s^{(i)}(\boldsymbol{B}_s^{(i)})^{\mathsf{T}},$$

where $\boldsymbol{U}_{s+1}^{(i)} = \left(\boldsymbol{P}_1^{(i)}, \ldots, \boldsymbol{P}_{s+1}^{(i)}\right) \in \mathbb{R}^{n\times\tilde{t}(s+1)}$ and $\boldsymbol{W}_s^{(i)} = \left(\boldsymbol{Z}_1^{(i)}, \ldots, \boldsymbol{Z}_s^{(i)}\right) \in \mathbb{R}^{m\times s\tilde{t}}$ have orthonormal columns and $\boldsymbol{B}_s^{(i)} \in \mathbb{R}^{s\tilde{t}\times s\tilde{t}}$ is lower block bidiagonal

Algorithm 3 A framework of the restarted block PSB method for NNRLS (1).

Input: $\boldsymbol{C} \in \mathbb{R}^{m\times n}$, $\boldsymbol{D} \in \mathbb{R}^{m\times \ell}$, k, s, and $Iter_{\max}$.
Output: $\boldsymbol{X}^{(i)}$ and $f(\boldsymbol{X}^{(i)})$.

1: Perform the k-step PSB process (6) on $\boldsymbol{C}$ and $\boldsymbol{D}$ obtain $\boldsymbol{B}_{k+1}$ and $\boldsymbol{W}_k$. Obtain the solution $\boldsymbol{Y}_k$ of NNRLS (8). Let $\boldsymbol{X}^{(1)} := \boldsymbol{X}_k = \boldsymbol{W}_k\boldsymbol{Y}_k$ and $i = 1$;
2: **while** $i \le Iter_{\max}$
3: Let $\begin{pmatrix}\boldsymbol{G} & \boldsymbol{X}^{(i)}\end{pmatrix} = \boldsymbol{P}_1^{(i)}\boldsymbol{R}_1^{(i)}$ be the economy-size QR decomposition of $\begin{pmatrix}\boldsymbol{G} & \boldsymbol{X}^{(i)}\end{pmatrix}$ with $\boldsymbol{P}_1^{(i)} \in \mathbb{R}^{n\times \bar{t}}$.
4: Let $\mathcal{K}_s^{(i)} := \mathcal{K}_s(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}, \boldsymbol{P}_1^{(i)}) = \mathrm{span}\{\boldsymbol{P}_1^{(i)}, (\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})\boldsymbol{P}_1^{(i)}, \ldots, (\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})^{s-1}\boldsymbol{P}_1^{(i)}\}$;
5: Obtain the solution $\boldsymbol{X}^{(i+1)}$ of

$$\boldsymbol{X}^{(i+1)} = \arg\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_s^{(i)}} f(\boldsymbol{X}); \tag{15}$$

6: If the termination criteria are met, output the result $\boldsymbol{X}^{(i+1)}$ and $f(\boldsymbol{X}^{(i+1)})$;
7: $i = i + 1$;
8: **end while**

and $\mathcal{R}(\boldsymbol{U}_s^{(i)}) = \mathcal{K}_s^{(i)} := \mathcal{K}_s(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}, \boldsymbol{P}_1^{(i)})$. It follows that

$$\begin{aligned}
&\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_s^{(i)}} f(\boldsymbol{X}) = \min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_s^{(i)}} \|\boldsymbol{C}\boldsymbol{X} - \boldsymbol{D}\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{X}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{s\bar{t}\times\ell}} \left\|\boldsymbol{C}\boldsymbol{U}_s^{(i)}\boldsymbol{Y} - \boldsymbol{D}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{U}_s^{(i)}\boldsymbol{Y}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{s\bar{t}\times\ell}} \left\|(\boldsymbol{B}_s^{(i)})^{\mathsf{T}}\boldsymbol{Y} - (\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D}\right\|_{\mathsf{F}}^2 + \left\|\left(\boldsymbol{I} - \boldsymbol{W}_s^{(i)}(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\right)\boldsymbol{D}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_* \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{s\bar{t}\times\ell}} \left\|\begin{pmatrix}(\boldsymbol{B}_s^{(i)})^{\mathsf{T}} \\ \boldsymbol{0}\end{pmatrix}\boldsymbol{Y} - \begin{pmatrix}(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D} \\ \left\|\left(\boldsymbol{I} - \boldsymbol{W}_s^{(i)}(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\right)\boldsymbol{D}\right\|_{\mathsf{F}}\end{pmatrix}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_* \qquad (13) \\
&= \min_{\boldsymbol{Y}\in\mathbb{R}^{s\bar{t}\times\ell}} \left\|\begin{pmatrix}(\boldsymbol{B}_s^{(i)})^{\mathsf{T}} \\ \boldsymbol{0}\end{pmatrix}\boldsymbol{Y} - \begin{pmatrix}(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D} \\ \sqrt{\|\boldsymbol{D}\|_{\mathsf{F}}^2 - \left\|(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D}\right\|_{\mathsf{F}}^2}\end{pmatrix}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_*.
\end{aligned}$$

Let

$$\boldsymbol{Y}^{(i)} = \arg\min_{\boldsymbol{Y}\in\mathbb{R}^{s\bar{t}\times\ell}} \left\|\begin{pmatrix}(\boldsymbol{B}_s^{(i)})^{\mathsf{T}} \\ \boldsymbol{0}\end{pmatrix}\boldsymbol{Y} - \begin{pmatrix}(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D} \\ \sqrt{\|\boldsymbol{D}\|_{\mathsf{F}}^2 - \left\|(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D}\right\|_{\mathsf{F}}^2}\end{pmatrix}\right\|_{\mathsf{F}}^2 + \mu\|\boldsymbol{Y}\|_*. \tag{14}$$

Then $\boldsymbol{X}^{(i+1)} = \boldsymbol{U}_s^{(i)}\boldsymbol{Y}^{(i)} = \arg\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_s^{(i)}} f(\boldsymbol{X})$. Note that both $\|(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D}\|_{\mathsf{F}}^2$ and $(\boldsymbol{W}_s^{(i)})^{\mathsf{T}}\boldsymbol{D}$ can be computed recursively.

We now present the convergence analysis of Algorithm 3.

Lemma 4 *Define*

$$\mathcal{H}(\boldsymbol{X}) := \boldsymbol{H}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{\Lambda}, \ \textit{where} \ \ \boldsymbol{\Lambda} = \mu \boldsymbol{V}_1 \boldsymbol{\Sigma}_1^{-1} \boldsymbol{V}_1^{\mathsf{T}}$$

If $\operatorname{rank}(\boldsymbol{X}_{\rm opt}) = \ell$. *Then*

$$\mathcal{M}_s^{(i)} := \{\mathcal{R}(\boldsymbol{X}) \mid \boldsymbol{X} = \boldsymbol{X}^{(i)} + p(\mathcal{H})\mathcal{H}(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt}), \forall p \in \mathcal{P}_{s-2}\} \subseteq \mathcal{K}_s^{(i)}, \ s > 2.$$

Proof 4 *We have*

$$\begin{aligned}\mathcal{H}(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt}) &= \boldsymbol{H}(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt}) + (\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt})\boldsymbol{\Lambda} \\ &= \boldsymbol{H}\boldsymbol{X}^{(i)} + \boldsymbol{X}^{(i)}\boldsymbol{\Lambda} - (\boldsymbol{H}\boldsymbol{X}_{\rm opt} + \boldsymbol{X}_{\rm opt}\boldsymbol{\Lambda}) \\ &\overset{(5)}{=} \boldsymbol{H}\boldsymbol{X}^{(i)} + \boldsymbol{X}^{(i)}\boldsymbol{\Lambda} - \boldsymbol{G}.\end{aligned}$$

Hence, $\mathcal{R}\left(\mathcal{H}(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt})\right) \subseteq \mathcal{K}_s^{(i)}$ *and*

$$\begin{aligned}&\mathcal{H}^2(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt}) = \mathcal{H}(\boldsymbol{H}\boldsymbol{X}^{(i)} + \boldsymbol{X}^{(i)}\boldsymbol{\Lambda} - \boldsymbol{G}) \\ =&\boldsymbol{H}^2\boldsymbol{X}^{(i)} + 2\boldsymbol{H}\boldsymbol{X}^{(i)}\boldsymbol{\Lambda} - \boldsymbol{H}\boldsymbol{G} + \boldsymbol{X}^{(i)}\boldsymbol{\Lambda}^2 - \boldsymbol{G}\boldsymbol{\Lambda}.\end{aligned}$$

It follows that $\mathcal{R}\left(\mathcal{H}^2(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt})\right) \subseteq \mathcal{K}_s^{(i)}$. *By mathematical induction,*

$$\mathcal{R}\left(\mathcal{H}^j(\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt})\right) \subseteq \mathcal{K}_s^{(i)}, j = 1, 2, \ldots, s-2,$$

which implies that $\mathcal{M}_s^{(i)} \subseteq \mathcal{K}_s^{(i)}$.

Theorem 3 *Let* $\boldsymbol{X}^{(1)} = \boldsymbol{X}_k$ *and*

$$\delta_s^{(i)} = \min_{\mathcal{R}(\boldsymbol{X}) \subseteq \mathcal{K}_s^{(i)}} \|\boldsymbol{X} - \boldsymbol{X}_{\rm opt}\|, \ i = 1, 2, \ldots.$$

If $\operatorname{rank}(\boldsymbol{X}_{\rm opt}) = \ell$. *Then*

$$\delta_s^{(i)} \leq 2\left(\frac{\sqrt{\xi} - 1}{\sqrt{\xi} + 1}\right)^{s-1} \|\boldsymbol{X}^{(i)} - \boldsymbol{X}_{\rm opt}\|_{\mathsf{F}}, \tag{16}$$

where $\xi = \|\boldsymbol{I}_\ell \otimes \boldsymbol{H} + \boldsymbol{\Lambda} \otimes \boldsymbol{I}_n\|_2 \|(\boldsymbol{I}_\ell \otimes \boldsymbol{H} + \boldsymbol{\Lambda} \otimes \boldsymbol{I}_n)^{-1}\|_2$ *and "* $\otimes$ *" denotes the Kronecker product.*

Proof 5 *Let* $\boldsymbol{H}_\Lambda = \boldsymbol{I}_\ell \otimes \boldsymbol{H} + \boldsymbol{\Lambda} \otimes \boldsymbol{I}_n$. *Note that* $\boldsymbol{H}_\Lambda \succ \boldsymbol{0}$ *and* $\mathsf{vec}(\mathcal{H}(\boldsymbol{X})) = \boldsymbol{H}_\Lambda \mathsf{vec}(\boldsymbol{X})$. *Then we have*

$$\mathsf{vec}\left(\mathcal{H}^j(\boldsymbol{X})\right) = \boldsymbol{H}_\Lambda \mathsf{vec}\left(\mathcal{H}^{j-1}(\boldsymbol{X})\right) = \cdots = \boldsymbol{H}_\Lambda^{j-1} \mathsf{vec}\left(\mathcal{H}(\boldsymbol{X})\right) = \boldsymbol{H}_\Lambda^j \mathsf{vec}\left(\boldsymbol{X}\right), \quad j \geq 1,$$

Algorithm 4 A restarted block PSB method for NNRLS (1).

Input: $\boldsymbol{C} \in \mathbb{R}^{m\times n}$, $\boldsymbol{D} \in \mathbb{R}^{m\times \ell}$, k and $Iter_{\max}$.
Output: $\boldsymbol{X}^{(i)}$ and $f(\boldsymbol{X}^{(i)})$.

1: Perform the k-step block PSB process (6) on $\boldsymbol{C}$ and $\boldsymbol{D}$ to obtain $\boldsymbol{B}_{k+1}$ and $\boldsymbol{W}_k$. Obtain the solution $\boldsymbol{Y}_k$ to NNRLS (8). Let $\boldsymbol{X}^{(1)} := \boldsymbol{X}_k = \boldsymbol{W}_k\boldsymbol{Y}_k$, $\boldsymbol{Y}^{(0)} = \boldsymbol{Y}_k$, and $i = 1$;
2: **while** $i \le Iter_{\max}$
3: Let $s = \frac{k}{2}$. Perform the s-step block PSB process (6) on $\boldsymbol{C}^{\mathsf{T}}$ and $\boldsymbol{P}_1^{(i)}$ to obtain $\boldsymbol{B}_s^{(i)}$, $\boldsymbol{U}_s^{(i)}$ and $\boldsymbol{W}_s^{(i)}$;
4: Obtain the solution $\boldsymbol{Y}^{(i)}$ to NNRLS (14) via the PAPG method with initial value $\boldsymbol{Y}^{(i-1)}$;
5: Set $\boldsymbol{X}^{(i+1)} = \boldsymbol{U}_s^{(i)}\boldsymbol{Y}^{(i)}$;
6: If the termination criteria are met, output the results $\boldsymbol{X}^{(i+1)}$ and $f(\boldsymbol{X}^{(i+1)})$;
7: $i = i + 1$;
8: **end while**

where $\mathsf{vec}(\cdot)$ *denotes the vectorization operator. By Lemma 4, we obtain*

$$\begin{aligned}
&\min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{K}_s^{(i)}} \|\boldsymbol{X}-\boldsymbol{X}_{\rm opt}\|_F \le \min_{\mathcal{R}(\boldsymbol{X})\subseteq\mathcal{M}_s^{(i)}} \|\boldsymbol{X}-\boldsymbol{X}_{\rm opt}\|_F\\
&= \min_{p\in\mathcal{P}_{s-2}} \|(\boldsymbol{I}-p(\mathcal{H})\mathcal{H})(\boldsymbol{X}^{(i)}-\boldsymbol{X}_{\rm opt})\|_F\\
&= \min_{p\in\mathcal{P}_{s-2}} \left\|\mathsf{vec}\left[(\boldsymbol{I}-p(\mathcal{H})\mathcal{H})(\boldsymbol{X}^{(i)}-\boldsymbol{X}_{\rm opt})\right]\right\|_2\\
&= \min_{p\in\mathcal{P}_{s-2}} \left\|(\boldsymbol{I}-p(\boldsymbol{H}_\Lambda)\boldsymbol{H}_\Lambda)\mathsf{vec}(\boldsymbol{X}^{(i)}-\boldsymbol{X}_{\rm opt})\right\|_2\\
&\le \min_{p\in\mathcal{P}_{s-2}} \|\boldsymbol{I}-p(\boldsymbol{H}_\Lambda)\boldsymbol{H}_\Lambda\|_2\,\|\boldsymbol{X}^{(i)}-\boldsymbol{X}_{\rm opt}\|_F.
\end{aligned}$$

According to (11)*, inequality* (16) *holds.*

Remark 4.1 *If* $\mathrm{rank}(\boldsymbol{C}) = n$ *and* $\mathrm{rank}(\boldsymbol{X}_{\rm opt}) = \ell$*, it follows from Theorem 2 (iv) that*

$$\delta_s^{(i+1)} \le \underbrace{2\frac{\sigma_1(\boldsymbol{C})}{\sigma_\ell(\boldsymbol{C})}\sqrt{1+\frac{\sqrt{5}\mu}{\sigma_\ell\|\boldsymbol{H}\|_2}}\left(\frac{\sqrt{\xi}-1}{\sqrt{\xi}+1}\right)^{s-1}}_{:=\alpha_s}\cdot\delta_s^{(i)}.$$

If s *is sufficiently large such that* $\alpha_s < 1$*, then the bounds in* (16) *imply that the sequence* $\{\delta_s^{(i)}\}$ *converges linearly. In this case, Theorem 2 yield that* $f(\boldsymbol{X}^{(i+1)})$ *converges to* $f(\boldsymbol{X}_{\rm opt})$ *and* $\boldsymbol{X}^{(i+1)}$ *converges to* $\boldsymbol{X}_{\rm opt}$*. Furthermore, the closer* ξ *is to 1, the faster the convergence rate.*

5 Numerical experiments

In this section, we perform some numerical experiments to illustrate the numerical behavior of the proposed algorithms. All the numerical experiments were run on a AMD R7 5800H CPU 3.20 GHz with 16GB RAM under Windows 11 operation system. The experimental results are obtained from using MATLAB R2022a implementation with machine precision $u_{\text{machine}} \approx 2.22 \times 10^{-16}$.

To demonstrate the efficiency of Algorithm 4, we compare it with four approaches for solving (1), including the DGP method, the DAGP method, the PAPG method, and PAPG$_{\text{avg}}$ [17]. In all the experiments, we terminate the iteration once the following condition is satisfied [17]

$$\chi_i := \frac{\|\widetilde{\boldsymbol{X}}^{(i+1)} - \widetilde{\boldsymbol{X}}^{(i)}\|_{\mathsf{F}}}{\|\widetilde{\boldsymbol{X}}^{(i+1)}\|_{\mathsf{F}}} \leq 10^{-8}, \tag{17}$$

where $\widetilde{\boldsymbol{X}}^{(i)}$ denotes the approximate solution obtained at the i-th iteration of the algorithms, and we set $k = 20$ and $s = \frac{k}{2}$. In the numerical experiments, we employ the PAPG method to solve NNRLS (14). To enhance computational efficiency, we use $\boldsymbol{Y}^{(i-1)}$ as the initial value for calculating $\boldsymbol{Y}^{(i)}$. In summary, the final algorithm we proposed is presented in Algorithm 4. For Algorithm 4, we set the maximum number of iterations $Iter_{\max} = 1000$, and for the remaining algorithms we set $Iter_{\max} = 10000$.

Following the suggestions in [17, Section 6], we report the results for the DGP method with $\mathbf{0}$ as the initial value and step size $\alpha = 1$ (DGP-Z) , the DAGP method with $\text{Proj}_{\|\boldsymbol{S}\|_2 \leq \mu}(-\boldsymbol{C}^{\mathsf{T}}\boldsymbol{D})$ as the initial value (DAGP-P), the PAPG method with $\mathbf{0}$ as the initial value (PAPG-Z), the PAPG method with the least squares solution $\boldsymbol{C}^{\dagger}\boldsymbol{D}$ as the initial value (PAPG-LS), and the PAPG$_{\text{avg}}$ method. Here, $\text{Proj}_{\|\boldsymbol{S}\|_2 \leq \mu}(\cdot)$ denotes the nearest-point projection onto $\{\boldsymbol{S} \mid \|\boldsymbol{S}\|_2 \leq \mu\}$.

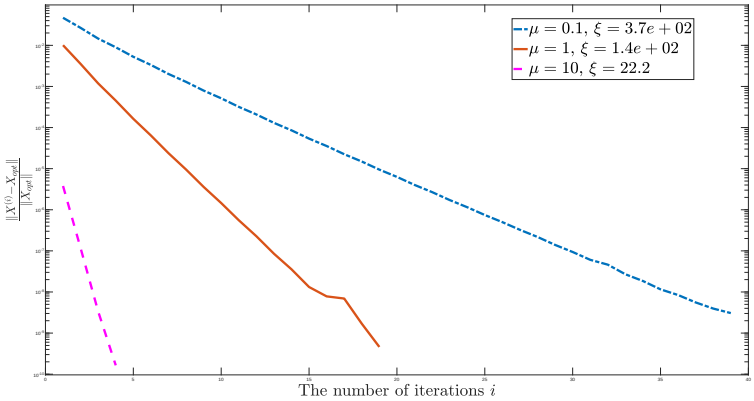
For the computation of $\boldsymbol{C}^{\dagger}\boldsymbol{D}$, we adopt the MATLAB's built-in function `lsqr` with the stopping criterion set to $tol = 10^{-8}$. In addition, at each iteration of the DGP method and DAGP method, it is required to solve a multiple right-hand side linear system with $\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}$ as the coefficient matrix (containing ℓ right-hand sides); for this purpose, we use MATLAB's division commend "\" to perform the computation. For the sake of fairness, we include the computation time for the initial values $\text{Proj}_{\|\boldsymbol{S}\|_2 \leq \mu}(-\boldsymbol{C}^{\mathsf{T}}\boldsymbol{D})$ and $\boldsymbol{C}^{\dagger}\boldsymbol{D}$ in the running time of the algorithms.

5.1 Impact of ξ on Algorithm 4 convergence

Within this section, we present one illustrative example to demonstrate the impact of $\xi = \|\boldsymbol{I}_\ell \otimes \boldsymbol{H} + \boldsymbol{\Lambda} \otimes \boldsymbol{I}_n\|_2 \|(\boldsymbol{I}_\ell \otimes \boldsymbol{H} + \boldsymbol{\Lambda} \otimes \boldsymbol{I}_n)^{-1}\|_2$ on Algorithm 4. Let

$$\boldsymbol{C} = \texttt{sprandn}(5000, 4000, 0.05) \quad \text{and} \quad \boldsymbol{X}_{\text{opt}} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{4000 \times 2},$$

Figure 1: Numerical results on Section 5.1: Convergence of $\frac{\|\boldsymbol{X}^{(i)}-\boldsymbol{X}_{\rm opt}\|}{\|\boldsymbol{X}_{\rm opt}\|}$.



where `sprandn` is MATLAB's built-in function. By Theorem 5, $\boldsymbol{D}$ should be set to

$$\boldsymbol{D} = (\boldsymbol{C}^{\mathsf{T}})^{\dagger}(\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}\boldsymbol{X}_{\rm opt} + \mu\boldsymbol{X}_{\rm opt}(\boldsymbol{X}_{\rm opt}^{\mathsf{T}}\boldsymbol{X}_{\rm opt})^{-\frac{1}{2}}).$$

Here, we compute $\boldsymbol{D}$ using MATLAB's built-in function `lsqrmin`. The experimental results are presented in Figure 5.1. It can be observed from the figure that as μ increases, ξ decreases, and the convergence rate of $\boldsymbol{X}^{(i)}$ rises significantly. This experimental observation is consistent with the theoretical analysis given in Remark 4.1.

5.2 Test on synthetic data

In this subsection, we conduct experiments on synthetic data generated with MATLAB's built-in function `sprandn` and `randn`:

$$\boldsymbol{C} = \texttt{sprandn}(m, n, density) \quad \text{and} \quad \boldsymbol{D} = \texttt{randn}(m, \ell),$$

where $density = 0.05$, and $\ell = 10, 20$, and 40, respectively. CPU(s) reports the output of MATLAB `cputime`.

The numerical results are reported in Table 5.2. It can be seen from Table 5.2 that in terms of CPU time, Algorithm 4 performs significantly better than PAPG-Z, PAPG-LS and PAPG_{avg} in most cases. When $(m, n, \mu) = (10000, 8000, 1)$ and $(m, n, \mu, \ell) = (10000, 8000, 10, 40)$, Algorithm 4 performs worse than or on par with DGP-Z and DAGP-P; in most other cases, Algorithm 4 outperforms DGP-Z and DAGP-P. In particular, when $\mu = 100$, Algorithm 4 performs significantly better than the other five methods. As analyzed in Remark 3.1, when μ is large, $\boldsymbol{X}_{\rm opt}$ is close to $\boldsymbol{0}$, which implies that $\varkappa$ is small. Therefore, Algorithm 4 converges faster when $\mu = 100$.

Table 1: Numerical results on subsection 5.2.

	Methods	$\ell = 10$		$\ell = 20$		$\ell = 40$	
		CPU(s)	χ_i	CPU(s)	χ_i	CPU(s)	χ_i
$m = 8000$ $n = 3000$ $(\mu = 1)$	Alg. 4	3.5	6.9e-9	3.0	8.7e-9	5.1	2.8e-9
	DGP-Z	5.0	8.1e-9	4.8	8.2e-9	5.0	8.3e-9
	DAGP-P	5.3	7.6e-9	5.5	7.6e-9	6.1	8.1e-9
	PAPG-Z	9.9	9.9e-9	17.8	9.9e-9	9.7	1.0e-8
	PAPG-LS	6.3	9.6e-9	10.8	9.7e-9	7.8	9.5e-9
	PAPG_{avg}	24.4	1.0e-8	41.9	1.0e-8	24.4	1.0e-8
$m = 8000$ $n = 6000$ $(\mu = 1)$	Alg. 4	52.3	4.6e-9	36.8	7.6e-9	74.5	5.4e-9
	DGP-Z	77.9	5.7e-9	78.5	5.4e-9	82.7	5.4e-9
	DAGP-P	87.0	3.9e-9	87.7	3.9e-9	92.6	4.1e-9
	PAPG-Z	2.8e+2	9.9e-9	5.0e+2	1.0e-8	2.9e+2	9.9e-9
	PAPG-LS	1.6e+2	1.0e-8	2.8e+2	9.9e-9	1.8e+2	9.9e-9
	PAPG_{avg}	7.0e+2	1.0e-8	1.2e+3	1.0e-8	7.6e+2	1.0e-8
$m = 10000$ $n = 6000$ $(\mu = 1)$	Alg. 4	25.2	4.0e-9	18.4	1.9e-9	28.8	4.2e-9
	DGP-Z	49.2	1.3e-10	48.5	1.3e-10	48.3	1.0e-10
	DAGP-P	53.9	1.3e-9	54.6	1.4e-9	55.2	1.4e-9
	PAPG-Z	1.1e+1	9.8e-9	2.0e+2	9.8e-9	1.1e+2	9.8e-9
	PAPG-LS	62.3	9.9e-9	1.1e+2	1.0e-8	69.7	1.0e-8
	PAPG_{avg}	3.4e+2	1.0e-8	6.2e+2	1.0e-8	3.4e+2	1.0e-8
$m = 10000$ $n = 8000$ $(\mu = 1)$	Alg. 4	1.3e+2	8.9e-9	1.0e+2	8.0e-9	2.2e+2	9.5e-9
	DGP-Z	1.1e+2	8.5e-9	1.1e+2	7.0e-9	1.1e+2	7.0e-9
	DAGP-P	1.0e+2	5.5e-9	1.3e+2	5.5e-9	1.3e+2	5.0-9
	PAPG-Z	7.7e+2	9.9e-9	1.4e+3	1.0e-8	9.2e+2	1.0e-8
	PAPG-LS	4.2e+2	9.9e-9	7.6e+2	1.0e-8	4.8e+2	1.0e-8
	PAPG_{avg}	1.3e+3	1.8e-8	2.4e+3	1.7e-8	1.5e+3	1.7e-8
$m = 8000$ $n = 3000$ $(\mu = 10)$	Alg. 4	3.5	4.4e-9	3.0	7.4e-9	5.7	2.8e-9
	DGP-Z	5.7	4.5e-10	6.0	4.4e-10	5.8	4.4e-10
	DAGP-P	6.9	3.2e-9	7.4	3.5e-9	7.0	3.9e-9
	PAPG-Z	10.6	9.8e-9	17.7	9.2e-9	10.6	9.8e-9
	PAPG-LS	8.3	9.5e-9	14.4	9.2e-9	10.0	9.4e-9
	PAPG_{avg}	47.1	9.9e-9	84.4	1.0e-8	48.1	1.0e-8
$m = 8000$ $n = 6000$ $(\mu = 10)$	Alg. 4	48.5	6.5e-9	36.2	8.0e-9	75.1	6.3e-9
	DGP-Z	83.5	5.1e-9	84.5	4.4e-9	84.3	4.9e-9
	DAGP-P	99.4	6.2e-9	99.3	7.1e-9	1.0e+2	6.0e-9
	PAPG-Z	2.6e+2	1.0e-8	4.8e+2	1.0e-8	2.9e+2	9.0e-9
	PAPG-LS	2.2e+2	1.0e-8	3.8e+2	1.0e-8	2.4e+2	1.0e-8
	PAPG_{avg}	5.2e+2	1.0e-8	9.6e+2	1.0e-8	5.8e+2	1.0e-8
$m = 10000$ $n = 6000$ $(\mu = 10)$	Alg. 4	23.5	9.2e-9	19.4	5.0e-9	36.7	3.4e-9
	DGP-Z	52.3	4.4e-9	52.0	4.7e-9	53.0	4.8e-9
	DAGP-P	62.6	4.3e-9	64.0	4.3e-9	64.5	4.2e-9
	PAPG-Z	1.2e+2	9.9e-9	1.9e+2	9.9e-9	1.2e+2	1.0e-8
	PAPG-LS	85.0	1.0e-8	1.5e+2	9.9e-9	97.8	1.0e-8
	PAPG_{avg}	3.8e+2	1.0e-8	6.7e+2	1.0e-8	3.9e+2	1.0e-8
$m = 10000$ $n = 8000$ $(\mu = 10)$	Alg. 4	1.1e+2	8.6e-9	1.0e+2	4.8e-9	1.6e+2	8.4e-9
	DGP-Z	1.4e+2	2.0e-9	1.5e+2	1.9e-9	1.5e+2	2.4e-9
	DAGP-P	1.8e+2	6.8e-10	1.9e+2	1.0e-9	1.9e+2	1.7e-9
	PAPG-Z	6.8e+2	1.0e-8	1.2e+3	1.0e-8	7.3e+2	1.0e-8
	PAPG-LS	5.7e+2	9.9e-9	9.8e+2	1.0e-8	6.2e+2	9.9e-9
	PAPG_{avg}	1.1e+3	1.0e-8	2.0e+3	1.0e-8	1.0e+03	1.0e-8
$m = 8000$ $n = 3000$ $(\mu = 100)$	Alg. 4	2.8	8.2e-9	2.8	3.5e-9	4.0	5.6e-9
	DGP-Z	11.2	7.4e-9	11.9	7.0e-9	12.2	6e-9
	DAGP-P	12.0	6.2e-9	12.7	9.3e-9	13.3	8.2e-9
	PAPG-Z	7.7	9.2e-9	13.0	9.7e-9	7.7	9.2e-9
	PAPG-LS	7.7	9.0e-9	13.4	9.5e-9	10.3	9.9e-9
	PAPG_{avg}	42.6	1.0e-8	78.6	1.0e-8	43.3	1.0e-8
$m = 8000$ $n = 6000$ $(\mu = 100)$	Alg. 4	23.3	3.6e-9	18.5	6.2e-9	30.1	2.5e-9
	DGP-Z	1.4e+2	7.7e-9	1.4e+2	7.8e-9	1.4e+2	5.7e-9
	DAGP-P	2.8e+2	9.8e-9	2.8e+2	9.4e-9	2.9e+2	2.4e-9
	PAPG-Z	1.1e+2	9.9e-9	2.0e+2	9.9e-9	1.1e+2	9.7e-9
	PAPG-LS	1.2e+2	9.9e-9	2.2e+2	9.8e-9	1.4e+2	1.0e-8
	PAPG_{avg}	3.0e+2	1.0e-8	5.1e+2	9.9e-9	5.0e+2	1.0e-8
$m = 10000$ $n = 6000$ $(\mu = 100)$	Alg. 4	21.3	4.3e-9	16.4	5.2e-9	30.1	6.2e-9
	DGP-Z	71.8	3.2e-9	76.6	4.5e-9	73.7	4.3e-9
	DAGP-P	1.3e+2	1.7e-8	1.4e+2	4.3e-9	1.3e+2	5.6e-9
	PAPG-Z	80.9	9.9e-9	1.4e+2	9.9e-9	1.2e+2	9.9e-9
	PAPG-LS	81.7	9.8e-9	1.4e+2	1.0e-8	1.3e+2	9.9e-9
	PAPG_{avg}	3.0e+2	1.0e-8	5.4e+2	1.0e-8	3.6e+2	9.9e-9
$m = 10000$ $n = 8000$ $(\mu = 100)$	Alg. 4	54.2	3.0e-9	40.1	2.4e-9	70.1	7.3e-9
	DGP-Z	2.6e+2	7.9e-9	2.6e+2	7.0e-9	2.6e+2	7.2e-9
	DAGP-P	6.1e+2	8.7e-9	6.2e+2	9.2e-9	6.2e+2	9.6e-9
	PAPG-Z	2.7e+2	1.0e-8	4.9e+2	9.9e-9	3.0e+2	9.9e-9
	PAPG-LS	3.0e+2	9.9e-9	5.3e+2	9.9e-9	3.2e+2	9.8e-9
	PAPG_{avg}	6.4e+2	1.0e-8	1.1e+3	1.0e-8	7.2e+2	1.0e-8

Table 2: Summary of test matrices.

$\boldsymbol{A}$	row dimension	column dimension	sparsity
relat8	345688	12347	0.03%
kneser_8_3_1	15737	15681	0.02%
GL7d14	171375	47271	0.02%
flower_5_4	5226	14721	0.06%
cat_ears_4_4	19020	44448	0.005%
ns3Da	20414	20414	0.40%
EX6	5226	14721	0.69%
D6-6	120576	23740	0.005%

5.3 Tests on real-world matrices

Consider the following matrices:

$$\boldsymbol{C} = \begin{cases} \boldsymbol{A} & \text{if row dimension} \geq \text{column dimension} \\ \boldsymbol{A}^{\mathsf{T}} & \text{if row dimension} < \text{column dimension} \end{cases},$$

where the selection of $\boldsymbol{A}$ refers to Table 2. The test matrices listed in Table 2 are all from the University of Florida sparse matrix collection[1]. The construction of $\boldsymbol{D}$ coincides with that in subsection 5.2 and $\mu = 1$.

The numerical results are presented in Table 5.3. The symbol "-" in the table indicates that the solver failed to obtain an approximate solution within 3600 seconds. In almost all test problems, the CPU time of Algorithm 4 is lower than that of PAPG-Z, PAPG-LS, and PAPG_{avg} when $\ell = 10, 20$. When $\ell = 40$, Algorithm 4 does not perform well in terms of time. This is because, as ℓ increases, not only does the dimension of NNRLS (14) increase significantly, but the time consumed by the QR decomposition in the block PSB procedure also grows. The accuracy index χ_i of Algorithm 3 is generally higher than that of PAPG-Z, PAPG-LS, and PAPG_{avg}.

The DGP-Z method fails to produce valid results for most test problems, and DAGP-P performs significantly better than Algorithm 3 only in kneser_8_3_1 ($\ell = 40$) and flower_5_4 ($\ell = 40$), while failing to converge in relat8, GL7d14, ns3Da, and D6-6. This indicates that the applicability and stability of both DGP-Z and DAGP-P are limited. This is because, at each iteration of the DGP-Z and DAGP-P methods, one must solve a linear system with multiple right-hand sides (containing ℓ right-hand sides) where $\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}$ serves as the coefficient matrix. When $\boldsymbol{C}^{\mathsf{T}}\boldsymbol{C}$ is of large dimension or ill-conditioned, solving such a system becomes extremely time-consuming. Overall, Algorithm 4 performs the best in both efficiency and robustness, especially when ℓ is small.

[1] https://sparse.tamu.edu.

Table 3: Numerical results on subsection 5.3.

$\mu = 1$	Methods	$\ell = 10$		$\ell = 20$		$\ell = 40$	
		CPU(s)	χ_i	CPU(s)	χ_i	CPU(s)	χ_i
relat8	Alg. 4	4.1	1.9e-9	7.7	5.4e-9	23.0	1.3e-9
	DGP-Z	-	-	-	-	-	-
	DAGP-P	-	-	-	-	-	-
	PAPG-Z	11.0	9.5e-9	29.3	9.9e-9	35.8	8.8e-9
	PAPG-LS	9.7	9.8e-9	24.4	7.8e-9	32.9	9.2e-9
	PAPG_{avg}	25.0	1.0e-8	59.6	1.0e-8	76.8	1.0e-8
kneser_8_3_1	Alg. 4	9.5	6.6e-9	27.8	8.8e-9	94.4	8.9e-9
	DGP-Z	-	-	-	-	-	-
	DAGP-P	72.2	4.6e-11	72.1	4.9e-11	71.9	5.0e-11
	PAPG-Z	23.4	1.0e-8	60.3	1.0e-8	98.8	1.0e-8
	PAPG-LS	18.0	1.0e-8	46.5	9.9e-9	75.8	1.0e-8
	PAPG_{avg}	30.3	1.4e-8	71.0	1.4e-8	1.1e+2	1.4e-8
GL7d14	Alg. 4	16.4	3.0e-9	20.9	2.0e-9	44.8	6.1e-10
	DGP-Z	-	-	-	-	-	-
	DAGP-P	-	-	-	-	-	-
	PAPG-Z	88.5	9.7e-9	1.7e+2	9.4e-9	1.6e+2	9.9e-9
	PAPG-LS	41.3	9.9e-9	80.4	9.6e-9	88.5	9.3e-9
	PAPG_{avg}	1.5e+2	1.0e-8	3.4e+2	1.0e-8	3.3e+2	1.0e-8
flower_5_4	Alg. 4	1.5	7.5e-9	4.7	3.1e-9	16.8	6.5e-9
	DGP-Z	1.3	8.4e-9	1.4	5.6e-9	1.6	4.3e-9
	DAGP-P	3.1	1.3e-9	3.3	1.5e-9	3.5	2.6e-9
	PAPG-Z	2.1	9.9e-9	5.3	9.5e-9	11.5	9.9e-9
	PAPG-LS	1.6	9.2e-9	5.1	9.8e-9	10.8	9.8e-7
	PAPG_{avg}	3.3	1.0e-8	8.4	1.0e-8	21.0	1.0e-8
cat_ears_4_4	Alg. 4	11.4	8.5e-9	30.0	6.8e-9	1.0e+2	8.6e-9
	DGP-Z	40.0	2.7e-9	98.5	2.0e-9	1.0e+2	1.4e-9
	DAGP-P	1.1e+2	9.9e-9	1.1e+2	9.5e-9	1.2e+2	8.4e-9
	PAPG-Z	27.4	9.9e-9	68.0	1.0e-8	1.2e+2	1.2e-8
	PAPG-LS	23.1	1.0e-8	57.0	9.9e-9	1.1e+2	9.9e-9
	PAPG_{avg}	31.0	1.0e-8	82.3	1.0e-8	1.2e+2	1.0e-8
ns3Da	Alg. 4	39.0	9.6e-9	43.5	5.8e-9	1.1e+2	7.9e-8
	DGP-Z	-	-	-	-	-	-
	DAGP-P	-	-	-	-	-	-
	PAPG-Z	2.5e+2	1.0e-8	4.7e+2	1.0e-8	3.6e+2	1.0e-8
	PAPG-LS	3.5e+2	2.0e-8	6.9e+2	1.3e-8	5.7e+2	2.0e-8
	PAPG_{avg}	2.8e+2	1.0e-8	5.8e+2	1.0e-8	4.2e+2	1.0e-8
EX6	Alg. 4	17.2	7.9e-9	21.3	3.0e-9	52.5	7.1e-9
	DGP-Z	-	-	-	-	-	-
	DAGP-P	36.4	1.8e-9	37.1	1.5e-9	40.0	1.4e-9
	PAPG-Z	1.0e+2	1.0e-6	2.0e+2	9.2e-7	1.3e+2	8.9e-7
	PAPG-LS	1.0e+2	5.9e-9	2.0e+2	5.2e-8	1.5e+2	5.2e-8
	PAPG_{avg}	1.0e+2	9.4e-8	2.1e+2	9.5e-8	1.3e+2	1.0e-7
D6-6	Alg. 4	1.6	8.3e-9	4.7	1.0e-8	16.2	7.2e-9
	DGP-Z	-	-	-	-	-	-
	DAGP-P	-	-	-	-	-	-
	PAPG-Z	3.0	9.3e-9	8.0	9.6e-9	14.5	1.0e-8
	PAPG-LS	2.0	5.3e-9	5.2	5.6e-9	9.3	5.9e-9
	PAPG_{avg}	15.0	1.0e-8	39.2	1.0e-8	68.5	1.0e-8

6 Conclusions

This paper investigates the large-scale NNRLS problem. We propose a framework based on the block PSB process, and further develop a restarted block PSB algorithm to solve NNRLS.

First, the original high-dimensional NNRLS problem is projected onto the block Krylov subspace $\mathcal{K}_k$ generated by the k-step block PSB process, which reduces the original problem to a small-scale NNRLS problem. The projected small-scale NNRLS problem is then solved by the PAPG method.

Second, we conduct a convergence analysis of the proposed algorithm: we prove that the distance δ_k between the optimal solution $\boldsymbol{X}_{\text{opt}}$ and the search subspace $\mathcal{K}_k$ converges linearly with respect to the iteration number k; when the optimal solution is of full column rank (i.e., $\text{rank}(\boldsymbol{X}_{\text{opt}}) = \ell$), the convergence rate of the objective function error $f(\boldsymbol{X}_k) - f(\boldsymbol{X}_{\text{opt}})$ can achieve $\mathcal{O}\left(\left(\frac{\sqrt{\varkappa}-1}{\sqrt{\varkappa}+1}\right)^{2k}\right)$, which generally outperforms the $\mathcal{O}(k^{-1})$ or $\mathcal{O}(k^{-2})$ rates of gradient-type methods such as the proximal gradient method.

Third, to constrain the dimension of the subspace and improve the efficiency of the algorithm, we further propose a restarted block PSB algorithm and subsequently prove its convergence.

Finally, to verify the performance of Algorithm 4, we present a series of numerical experiments and compare the algorithm with DGP-Z, DAGP-P, PAPG-Z, PAPG-LS and PAPG_{avg}. The experimental results demonstrate that Algorithm 4 exhibits remarkable effectiveness and robustness in most test cases, especially when ℓ is small. The method proposed in this paper is effective for small values of ℓ. How to develop Krylov subspace methods for large ℓ deserves future research.

Acknowledgement

The work of this author was supported by the Youth Fund of the National Natural Science Foundation of China (Grant No. 12501529), the Basic Research Program of Jiangsu Province (Youth Program of Natural Science Foundation of Jiangsu Province, Grant No. BK20250892), and the China Postdoctoral Science Foundation (Grant No. 2025M783141).

Data Availability

Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest: The author declares that they have no conflict of interest.

References

[1] A. Alqahtani, S. Gazzola, L. Reichel, and G. Rodriguez, *On the block Lanczos and block Golub-Kahan reduction methods applied to discrete ill-posed problems*, Numer Linear Algebra Appl., 28 (2021), e2376.

[2] A. Beck, *First-Order Methods in Optimization*, SIAM, Philadelphia, 2017.

[3] R. Caruana, *Multitask learning*, Mach. Learn., 28 (1997), pp. 41–75.

[4] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the American Control Conference, 2001, pp. 4734–4739.

[5] T. Geng, G. Sun, Y. Xu, and J. He, *Truncated nuclear norm minimization based group sparse representation for image restoration* SIAM Journal on Imaging Sciences, 11 (2018), pp. 1878–1897.

[6] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers in Appl. Math. 17, SIAM, Philadephia, 1997.

[7] Y. Guo, G. Chen, T. Zeng, Q. Jin, and M. Ng, *Quaternion nuclear norm minus Frobenius norm minimization for color image reconstruction*, Pattern Recognition, 158 (2025), 110986.

[8] C. Huang, Z. Li, Y. Liu, T. Wu, and T. Zeng, *Quaternion-based weighted nuclear norm minimization for color image restoration*, Pattern Recognition, 128 (2022), 108665.

[9] R-C. Li, *New perturbation bounds for the unitary polar factor*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 327–332.

[10] K. Li, H. Li, R. Chan and Y. Wen, *Selecting regularization parameters for nuclear norm–type minimization problems*, SIAM J. Sci. Comput., 44(2022), pp. A2204–A2225.

[11] H. Li, Y. Lu, X. Shu, and Q. Liu, *Trace norm regularized canonical correlation analysis*, Proceedings of the 2016 International Conference on Machine Learning and Cybernetics, Jeju, South Korea, 10–13 July 2016.

[12] Z. Lu, R. Monteiro, and M. Yuan, *Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression*, Math. Program., Ser. A, 131(2012) pp. 163–194.

[13] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence* $\mathcal{O}(1/k^2)$, Doklady AN SSSR, 269(1983) pp. 543–547. Translated as Soviet Math. Docl.

[14] Y. Nesterov, *Smooth minimization of nonsmooth functions*, Math. Program., 103 (2005), pp. 127–152.

[15] C.C. PAIGE AND M.A. SAUNDERS, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8 (1982), pp .43–71.

[16] J. PAN AND M. K. NG, *Orthogonal nonnegative matrix factorization by sparsity and nuclear norm optimization*, SIAM Journal on Matrix Analysis and Applications, 39(2018), pp. 856–875.

[17] T. K. PONG, P. TSENG , S. W. JI, AND J. P. YE, *Trace norm regularization: reformulations, algorithms, and multi-task learning*, SIAM J. Optim., 20 (2010), pp. 3465–3489.

[18] B. RECHT, M. FAZEL, AND P.A. PARRILO, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Review, 52 (2010), pp. 471–501.

[19] J. RENNIE AND N. SREBRO, *Fast maximum margin matrix factorization for collaborative prediction*, in Proceedings of the International Conference on Machine Learning, 2005, pp. 713–719.

[20] P. SHANG AND L. KONG, *Safe subspace screening for the adaptive nuclear norm regularized trace regression*, J. Oper. Res. Soc. China (2025).

[21] N. SREBRO, J. RENNIE, AND T. JAAKKOLA, *Maximum-margin matrix factorization*, in Proceedings of the Conference on Neural Information Processing Systems, 2005, pp. 1329–1336.

[22] G.W. STEWART AND G.J. SUN, *Matrix Perturbation Theory*, Academic, Boston, 1990.

[23] P. TSENG, *On accelerated proximal gradient methods for convex-concave optimization*, Manuscript, Department of Mathematics, University of Washington, May 2008.

[24] G. WATSON, *Characterization of the subdifferential of some matrix norms*, Linear Algebra Appl., 170 (1992), pp. 1039–1053.

[25] M. WEIMER, A. KARATZOGLOU, Q. LE, AND A. SMOLA, $COFI^{rank}$ - *maximum margin matrix factorization for collaborative ranking*, in Proceedings of the Conference on Neural Information Processing Systems, 2008, pp. 1593–1600.

[26] M. WEIMER, A. KARATZOGLOU, AND A. SMOLA, *Improving maximum margin matrix factorization*, Mach. Learn., 72 (2008), pp. 263–276.

[27] Y. YU, Y. ZHANG, AND S. YUAN, *Quaternion-based weighted nuclear norm minimization for color image denoising*, Neurocomputing, 332 (2019), pp. 283–297.

[28] M. YUAN, A. EKICI, Z. LU, AND R. MONTEIRO, *Dimension reduction and coefficient estimation in multivariate linear regression*, J. R. Stat. Soc. Series B Stat. Methodol., 69(2007), pp. 329–346.