Check for updates

Numerical Linear Algebra with Applications

WILEY

RESEARCH ARTICLE OPEN ACCESS

A Preconditioned Majorization-Minimization Method for ℓ^2-ℓ^q Minimization

A. Buccini[1] | M. Donatelli[2] | M. Ratto[2] | L. Reichel[3]

[1]University of Cagliari, Cagliari, Italy | [2]University of Insubria, Como, Italy | [3]Kent State University, Kent, OH, USA

Correspondence: M. Ratto (mratto1@uninsubria.it)

Received: 13 June 2025 | **Revised:** 3 December 2025 | **Accepted:** 25 April 2026

Keywords: ill-posed problem | inverse problem | ℓ^2-ℓ^q minimization | preconditioning | regularization

ABSTRACT

The need to minimize a linear combination of an expression that involves an ℓ^q-norm of a linear transformation of the computed solution and the ℓ^2-norm of the residual error arises in image restoration as well as in statistics. A solution method that is suited for large-scale problems is the iterative maximization-minimization method, which determines an approximate solution in a generalized Krylov subspace. This paper describes a preconditioner for accelerating the convergence of the iterative method. The design of the preconditioner is inspired by the iterated Tikhonov regularization method. Numerical examples in image deblurring and computerized tomography show that the preconditioner significantly reduces the required CPU time and gives computed solutions of about the same quality as the unpreconditioned iterative method.

1 | Introduction

Inverse problems are concerned with the determination of the cause of an observed effect. They arise in many areas of science and engineering, including image restoration. Many inverse problems are ill-posed, that is, the problems might not have a unique solution or the solution might not depend continuously on the data; see Engl et al. [1]. This paper is concerned with the solution of linear problems that are discretizations of ill-posed problems. They are of the form

$$A\mathbf{x} + \boldsymbol{\eta} = \mathbf{b}^{\delta}, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is a large very ill-conditioned matrix whose singular values decrease to zero with no significant gap between the small ones, $\mathbf{b}^{\delta} \in \mathbb{R}^m$ represents measured data, and $\mathbf{x} \in \mathbb{R}^n$ denotes the unknown solution that we would like to compute. We assume that the vector $\mathbf{b}^{\delta}$ is corrupted by an error $\boldsymbol{\eta} \in \mathbb{R}^m$ that may stem from measurement and discretization inaccuracies, and we will refer to the vector $\boldsymbol{\eta}$ as *noise*. The entries of $\boldsymbol{\eta}$ are assumed to be independent realizations of a random Gaussian variable with zero mean and with a certain variance. Thus, the noise is white Gaussian. However, the minimization problem considered can be easily modified to handle other types of noise (see [2]), and the proposed preconditioning strategy can also be applied to the solution of these modified problems.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Numerical Linear Algebra with Applications* published by John Wiley & Sons Ltd.

Let $\mathbf{b} = \mathbf{b}^\delta - \boldsymbol{\eta}$ denote the unknown noise-free data vector. We would like to determine the solution $\mathbf{x}^\dagger$ of minimal Euclidean norm of the linear system of equations

$$A\mathbf{x} = \mathbf{b}, \tag{2}$$

where we assume that $\mathbf{b} \in \mathcal{R}(A)$ and $\mathcal{R}(A)$ denotes the range of A. Since, usually, $\mathbf{b}^\delta \notin \mathcal{R}(A)$, one may consider the least-squares problem

$$\arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2, \tag{3}$$

where $\|\cdot\|_2$ denotes the Euclidean vector norm. However, the solution of this least-squares problem usually is a poor approximation of $\mathbf{x}^\dagger$ due to the presence of the noise $\boldsymbol{\eta}$ in the data and the ill-conditioning of the matrix A. Least-squares problems of this kind are commonly referred to as *linear discrete ill-posed problems*; see, for example, [3] for discussions of these kinds of problems.

To obtain an accurate approximation of $\mathbf{x}^\dagger$, one applies regularization, that is, one replaces the problem (3) by a nearby problem, whose solution is less sensitive to the error $\boldsymbol{\eta}$ in $\mathbf{b}^\delta$ than the solution of (3). A regularization method that has attracted considerable attention in recent years is the ℓ^2-ℓ^q minimization method that determines a solution of the problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \|L\mathbf{x}\|_q^q \right\}, \tag{4}$$

where the *regularization matrix* $L \in \mathbb{R}^{s\times n}$ is such that $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$, the *regularization parameter* $\mu > 0$ balances the influence of the two terms in the right-hand side of (4), and $0 < q \leq 2$; see [4–11]. Here $\mathcal{N}(M)$ denotes the null space of the matrix M. With a slight abuse of terminology, we will refer to the function

$$\mathbf{x} \to \|\mathbf{x}\|_q = \left(\sum_{i=1}^{n} |x_i|^q \right)^{1/q}$$

as the ℓ^q-norm of $\mathbf{x}$ for any $q \geq 0$, even though it is not a norm for $q < 1$. The first term on the right-hand side of (4) is referred to as the *fidelity term* and the second term as the *regularization term*. The matrix L is chosen to promote desirable features of the computed solution.

When $q = 2$, the minimization problem (4) is a Tikhonov minimization problem in general form [3, 12]. Letting $0 \leq q \leq 1$ promotes sparsity of $L\mathbf{x}^*$. We are particularly interested in these values of q; see [2] for a discussion on the choice of q. Note that the minimization problem (4) is non-convex when $0 \leq q < 1$.

We compute an approximate solution of (4) by a restarted version of an iterative majorization-minimization (MM) method that was first described in [9]. Recent modifications and an implementation are described in [7, 8]. This method determines an approximate solution in a generalized Krylov subspace that typically is of fairly low dimension. The dimension increases by one at each iteration until a satisfactory approximate solution has been determined or the method is restarted. Restarting prevents the dimension of the solution subspace from becoming large. This reduces the computational effort and the memory requirement of the method. Restarting is described in [7] and is implemented in the software presented in [8]. The MM method for the solution of ℓ^p-ℓ^q minimization problems produces extremely accurate reconstructions and is competitive with several state-of-the-art methods; we refer the interested reader to [9, 13] for a comparison between MM and other methods in the literature.

The MM method may require quite a few iterations to produce a satisfactory approximation of $\mathbf{x}^\dagger$. The overall computational cost therefore may be considerable for large-scale problems even with the use of restarts. It is the purpose of this paper to discuss the design and application of a preconditioner whose aim is to reduce the number of iterations required and, thereby, decrease the computational effort without compromising the accuracy of the computed solution.

Inspired by solution methods for Tikhonov regularization in general form, that is, by solution methods for the ℓ^2-ℓ^2 minimization problem (see, e.g., [14, 15]), we propose a preconditioner that uses the matrix $A^T A$ and is fairly well-conditioned. Here and throughout this paper, the superscript ^{T} denotes transposition. Well-conditioning is obtained by shifting the matrix $A^T A$ by a positive multiple of the identity. We describe the preconditioner in two steps: Section 2 introduces the

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

preconditioner, but ignores the computational cost of its use. Section 3 presents a variant of the preconditioner that is much cheaper to apply for large-scale problems. We show that, if the regularization parameter $\mu > 0$ is fixed, then the preconditioned iterative method for the solution of (4) converges to a stationary point. A non-stationary version of the algorithm that adaptively chooses the regularization parameter by exploiting the discrepancy principle (DP) is also described. This version is inspired by [6].

This paper is organized as follows. The preconditioned MM method is described in Section 2. The use of generalized Krylov subspaces to reduce the computational burden is discussed in Section 3 and Section 4 considers the choice of the regularization parameter. Section 5 reports a few numerical examples that illustrate the effectiveness of the preconditioned iterative method described when applied to image deblurring and computed tomography (CT) problems. Concluding remarks can be found in Section 6.

2 | Preconditioned Majorization-Minimization

We consider preconditioners of the form

$$P^{\frac{1}{2}} = \sqrt{A^T A + \alpha I}, \tag{5}$$

where I is the identity matrix and $\alpha > 0$ is a scalar to be determined. A discussion on a possible strategy to estimate α can be found at the end of the numerical results. Our implementation of this preconditioner is discussed in Section 3. The present section describes the iterative method in $\mathbb{R}^n$.

The operator P is inspired by the computations required for iterated Tikhonov regularization obtained by setting $q = 2$ in (4), and letting $\mu = \alpha$ and $L = I$; see [14, 15] for discussions on these iteration methods. For these choices of q, μ, and L, the solution of (4) can be written as

$$\left(A^T A + \alpha I\right)\mathbf{x} = A^T \mathbf{b}^\delta.$$

We may interpret the operator $P^{\frac{1}{2}}$ as a symmetrized and regularized version of A, in which the eigenvalues of $A^T A$ have been shifted by α. Since A is severely ill-conditioned, the eigenvalues of $A^T A$ "cluster" at the origin. Shifting the eigenvalues by $\alpha > 0$ regularizes the operator.

We will use $P^{\frac{1}{2}}$ as a right preconditioner in (4). The minimization problem becomes

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \left\| AP^{-\frac{1}{2}} P^{\frac{1}{2}} \mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \left\| LP^{-\frac{1}{2}} P^{\frac{1}{2}} \mathbf{x} \right\|_q^q \right\},$$

that is

$$\begin{cases} \mathbf{y}^* = \arg\min_{\mathbf{y}} \left\{ \frac{1}{2} \| AP^{-\frac{1}{2}} \mathbf{y} - \mathbf{b} \|_2^2 + \frac{\mu}{q} \| LP^{-\frac{1}{2}} \mathbf{y} \|_q^q \right\}, \\ \mathbf{x}^* = P^{-\frac{1}{2}} \mathbf{y}^*. \end{cases} \tag{6}$$

Denote the functional minimized in (6) by

$$\mathcal{J}(\mathbf{y}) = \frac{1}{2} \left\| AP^{-\frac{1}{2}} \mathbf{y} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \left\| LP^{-\frac{1}{2}} \mathbf{y} \right\|_q^q.$$

When $0 < q < 1$, this functional is neither convex nor differentiable, and when $q = 1$ the functional is convex, but not differentiable.

Following the approach in [9], we introduce a surrogate functional $\mathcal{J}_\varepsilon$ that is differentiable and approximates $\mathcal{J}$. Let $\mathbf{z}_j$ denote the jth entry of the vector $\mathbf{z}$ and define

$$\mathcal{J}_\varepsilon(\mathbf{y}) = \frac{1}{2} \sum_{j=1}^{m} \left(AP^{-\frac{1}{2}} \mathbf{y} - \mathbf{b}^\delta \right)_j^2 + \frac{\mu}{q} \sum_{j=1}^{s} \Phi_{q,\varepsilon} \left((LP^{-\frac{1}{2}} \mathbf{y})_j \right), \tag{7}$$

where $\varepsilon > 0$ is a smoothing parameter and

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\Phi_{q,\varepsilon}(t) = \begin{cases} |t|^q & \text{for } q > 1, \\ (t^q + \varepsilon^2)^{q/2} & \text{for } 0 < q \leq 1. \end{cases}$$

We compute a stationary point of $\mathcal{J}_\varepsilon$ by the restarted MM method described in [7, 8]. Each iteration with this iterative method consists of two steps: Let $\mathbf{y}^{(k)}$ denote the current approximation of the solution $\mathbf{y}^*$ of (6). We first determine a quadratic functional $\mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)})$ that majorizes the smoothed functional $\mathcal{J}_\epsilon$ and is tangent to it at $\mathbf{y}^{(k)}$. Then we compute a new iterate $\mathbf{y}^{(k+1)}$ by minimizing $\mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)})$ with respect to $\mathbf{y}$. The functional $\mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)})$ is a quadratic tangent majorant for $\mathcal{J}_\varepsilon$. It satisfies

(i) $\mathbf{y} \mapsto \mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)})$ is a quadratic function;

(ii) $\mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)}) \geq \mathcal{J}_\varepsilon(\mathbf{y})$ for all $\mathbf{y}$;

(iii) $\mathcal{Q}(\mathbf{y}^{(k)}, \mathbf{y}^{(k)}) = \mathcal{J}_\varepsilon(\mathbf{y}^{(k)})$;

(iv) $\nabla\mathcal{Q}(\mathbf{y}^{(k)}, \mathbf{y}^{(k)}) = \nabla\mathcal{J}_\varepsilon(\mathbf{y}^{(k)})$, where $\nabla\mathcal{Q}(\mathbf{y}^{(k)}, \mathbf{y}^{(k)})$ denotes the gradient of $\mathcal{Q}(\mathbf{y}, \mathbf{y}^{(k)})$ computed with respect to $\mathbf{y}$ and evaluated at $\mathbf{y}^{(k)}$.

One can construct various quadratic tangent majorants with the above properties. In this paper, we use one that in [9] is referred to as "fixed". However, our preconditioner can also be applied with other choices of quadratic tangent majorants. The fixed quadratic majorant is defined by

$$Q(\mathbf{y}, \mathbf{y}^{(k)}) = \frac{1}{2}\|AP^{-\frac{1}{2}}\mathbf{y} - \mathbf{b}\|_2^2 + \frac{\mu}{2}\varepsilon^{q-2}\left(\|LP^{-\frac{1}{2}}\mathbf{y}\|_2^2 - 2\left\langle \boldsymbol{\omega}_{\text{reg}}^{(k)}, LP^{-\frac{1}{2}}\mathbf{y}\right\rangle\right) + c,$$

where the term c is independent of $\mathbf{y}$ and the *weight vector* is given by

$$\boldsymbol{\omega}_{\text{reg}}^{(k)} = \mathbf{u}^{(k)}\left(1 - \left(\frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right), \qquad \mathbf{u}^{(k)} = LP^{-\frac{1}{2}}\mathbf{y}^{(k)}.$$

All the operations in the above expression are element-wise.

Discarding terms that are independent of $\mathbf{y}$ shows that the iterate $\mathbf{y}^{(k+1)}$ satisfies

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}}\left\{\left\|AP^{-\frac{1}{2}}\mathbf{y} - \mathbf{b}\right\|_2^2 + \eta\left\|LP^{-\frac{1}{2}}\mathbf{y} - \boldsymbol{\omega}_{\text{reg}}^{(k)}\right\|_2^2\right\}, \tag{8}$$

where

$$\eta = \mu\varepsilon^{q-2}. \tag{9}$$

Thus, η is a scaling of the regularization parameter μ.

The iterations described in this section can be implemented straightforwardly when the matrices A and L are small or have an exploitable structure, for example, both A and L are circulant matrices. An example of the latter is reported in [5] when $P = I$. We are concerned with the situation when the matrices A and L are large and A does not have a structure that can be exploited to reduce the computational effort. The following sections describe how the computations can be carried out fairly inexpensively by projecting the minimization problem (4) into generalized Krylov subspaces of fairly small dimensions.

Our solution method for (4) determines a suitable value of the regularization parameter $\mu > 0$ during the solution process. When an appropriate value of μ is known before beginning the computation of the solution of (4), other solution methods than the one described in this paper may be competitive. For instance, when $q = 1$, FISTA by Beck and Teboulle [16] can be applied and may be competitive. This is illustrated in [7].

3 | Krylov Subspace Preconditioning

When the matrix A is large and without exploitable structure, the solution of the minimization problem (8) may be expensive to compute. To reduce the computational effort, we seek to determine an approximate solution in a fairly

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

low-dimensional subspace. Similarly as in [9, 17], we compute approximate solutions in generalized Krylov subspaces. They are defined as follows. Restrict $\mathbf{y}$ in the minimization problem (8) to live in the subspace $\mathcal{V}_k$ (to be defined below) of dimension $\widehat{k} \gtrapprox k$, and let the columns of the matrix $V_k \in \mathbb{R}^{n\times\widehat{k}}$ form an orthonormal basis for $\mathcal{V}_k$. Then

$$\mathbf{y}^{(k)} = V_k \mathbf{t}^{(k)} \tag{10}$$

for some $\mathbf{t}^{(k)} \in \mathbb{R}^{\widehat{k}}$. To reduce the computational effort when applying the preconditioner P defined in (5), we "project" P into the subspace $\mathcal{V}_k$ as follows. Introduce the economical QR factorization

$$AV_k = Q_A R_A, \tag{11}$$

where the matrix $Q_A \in \mathbb{R}^{n\times\widehat{k}}$ has orthonormal columns and the matrix $R_A \in \mathbb{R}^{\widehat{k}\times\widehat{k}}$ is upper-triangular. We approximate A by

$$A \approx Q_A R_A V_k^T,$$

and approximate the square of the preconditioner (5) by

$$P_k := V_k R_A^T R_A V_k^T + \alpha I,$$

where we note that the matrix P_k is symmetric positive definite when $\alpha > 0$.

Let the columns of $V_k^\perp \in \mathbb{R}^{n\times(n-\widehat{k})}$ form an orthonormal basis for the subspace $\mathcal{V}_k^\perp$, which is the orthogonal complement of $\mathcal{V}_k$ in $\mathbb{R}^n$. The matrix $V_k^\perp$ does not have to be computed in our iterative method, but it is convenient to use it in our analysis. We have

$$P_k = \begin{bmatrix} V_k, & V_k^\perp \end{bmatrix} \begin{bmatrix} R_A^T R_A + \alpha I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} V_k^T \\ \left(V_k^\perp\right)^T \end{bmatrix}. \tag{12}$$

Since $R_A^T R_A$ is symmetric positive semidefinite, it has a spectral factorization of the form

$$R_A^T R_A = W \Lambda W^T, \tag{13}$$

where $W \in \mathbb{R}^{\widehat{k}\times\widehat{k}}$ is an orthogonal matrix and $\Lambda \in \mathbb{R}^{\widehat{k}\times\widehat{k}}$ is a diagonal matrix with nonnegative diagonal entries. Note that this decomposition can be computed without forming $R_A^T R_A$ by using the singular value decomposition (SVD) of R_A. We recall that the right singular vectors of R_A are the eigenvectors of $R_A^T R_A$, and the square of the singular values of R_A are the eigenvalues of $R_A^T R_A$.

Using the decomposition (13), we can write P_k in (12) as

$$\begin{aligned} P_k &= \begin{bmatrix} V_k, & V_k^\perp \end{bmatrix} \begin{bmatrix} W\Lambda W^T + \alpha W W^T & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} V_k^T \\ \left(V_k^\perp\right)^T \end{bmatrix} \\ &= \begin{bmatrix} V_k W, & V_k^\perp \end{bmatrix} \begin{bmatrix} \Lambda + \alpha I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} W^T V_k^T \\ \left(V_k^\perp\right)^T \end{bmatrix}. \end{aligned} \tag{14}$$

In particular, it holds

$$P_k V_k = V_k W(\Lambda + \alpha I) W^T, \tag{15}$$

since $\left(V_k^\perp\right)^T V_k = O$, where $O \in \mathbb{R}^{\widehat{k}\times\widehat{k}}$ denotes the zero matrix.

We turn to the evaluation of matrix-vector products with the preconditioning matrix $P_k^{-\frac{1}{2}}$. First, consider the situation when the vector $\mathbf{y}$ belongs to the subspace $\mathcal{V}_k$. Then, using (15), we obtain

$$P_k^{-\frac{1}{2}} \mathbf{y} = P_k^{-\frac{1}{2}} V_k \mathbf{t} = V_k W(\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t}. \tag{16}$$

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

For a general vector $\mathbf{s} \in \mathbb{R}^n$, using (14), we have

$$\begin{aligned} P_k^{-\frac{1}{2}}\mathbf{s} &= \begin{bmatrix} V_k W & V_k^{\perp} \end{bmatrix} \begin{bmatrix} (\Lambda + \alpha I)^{-\frac{1}{2}} & 0 \\ 0 & \alpha^{-\frac{1}{2}} I \end{bmatrix} \begin{bmatrix} W^T V_k^T \\ \left(V_k^{\perp}\right)^T \end{bmatrix} \mathbf{s} \\ &= V_k W \left[(\Lambda + \alpha I)^{-\frac{1}{2}} \right] W^T V_k^T \mathbf{s} + \alpha^{-\frac{1}{2}} V_k^{\perp} \left(V_k^{\perp}\right)^T \mathbf{s} \\ &= V_k W \left[(\Lambda + \alpha I)^{-\frac{1}{2}} \right] W^T V_k^T \mathbf{s} + \alpha^{-\frac{1}{2}} \left(I - V_k V_k^T \right) \mathbf{s} \\ &= V_k W \left[(\Lambda + \alpha I)^{-\frac{1}{2}} - \alpha^{-\frac{1}{2}} I \right] W^T V_k^T \mathbf{s} + \alpha^{-\frac{1}{2}} \mathbf{s}. \end{aligned}$$

We also have to evaluate matrix-vector products of the matrices $AP_k^{-\frac{1}{2}}V_k$ and $LP_k^{-\frac{1}{2}}V_k$ and vectors $\mathbf{t} \in \mathbb{R}^{\hat{k}}$. Combining Equations (16) and (11), we obtain

$$AP_k^{-\frac{1}{2}} V_k \mathbf{t} = Q_A R_A W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t}. \tag{17}$$

Similarly, computing and using the economical QR factorization $LV_k = Q_L R_L$, we get

$$LP_k^{-\frac{1}{2}} V_k \mathbf{t} = Q_L R_L W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t}. \tag{18}$$

Finally, looking for a solution of (8) in $\mathcal{V}_k$, we write $\mathbf{y}^{(k)}$ in the form (10) and approximate P by P_k. This gives

$$\mathbf{t}^{(k+1)} = \arg\min_{\mathbf{t} \in \mathbb{R}^{\hat{k}}} \left\{ \left\| AP_k^{-\frac{1}{2}} V_k \mathbf{t} - \mathbf{b}^{\delta} \right\|_2^2 + \eta \left\| LP_k^{-\frac{1}{2}} V_k \mathbf{t} - \boldsymbol{\omega}_{\text{reg}}^{(k)} \right\|_2^2 \right\}. \tag{19}$$

Combining equation (19) with (17) and (18), we get

$$\begin{aligned} \mathbf{t}^{(k+1)} &= \arg\min_{\mathbf{t} \in \mathbb{R}^{\hat{k}}} \left\| Q_A R_A W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t} - \mathbf{b}^{\delta} \right\|_2^2 + \eta \left\| Q_L R_L W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t} - \boldsymbol{\omega}_{\text{reg}}^{(k)} \right\|_2^2 \\ &= \arg\min_{\mathbf{t} \in \mathbb{R}^{\hat{k}}} \left\| \begin{bmatrix} R_A W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \\ \sqrt{\eta} R_L W (\Lambda + \alpha I)^{-\frac{1}{2}} W^T \end{bmatrix} \mathbf{t} - \begin{bmatrix} Q_A^T \mathbf{b}^{\delta} \\ \sqrt{\eta} Q_L^T \boldsymbol{\omega}_{\text{reg}}^{(k)} \end{bmatrix} \right\|_2^2. \end{aligned}$$

We will return to this minimization problem below.

Once we have computed $\mathbf{y}^{(k+1)} = V_k \mathbf{t}^{(k+1)}$, we enlarge the solution subspace as follows: Compute the residual error of the normal equations associated with (8), that is, let for $\eta = \eta_k$,

$$\mathbf{r}^{(k+1)} = P_k^{-\frac{1}{2}} A^T \left(AP_k^{-\frac{1}{2}} \mathbf{y}^{(k+1)} - \mathbf{b} \right) + \eta P_k^{-\frac{1}{2}} L^T \left(LP_k^{-\frac{1}{2}} \mathbf{y}^{(k+1)} - \boldsymbol{\omega}_{\text{reg}}^{(k)} \right).$$

This vector is orthogonal to the columns of the matrix V_k (at least in exact arithmetic). Its normalization

$$\mathbf{v}_{\text{new}} = \mathbf{r}^{(k+1)} / \|\mathbf{r}^{(k+1)}\|_2$$

is used to update the basis for the solution subspace $\mathcal{V}_{k+1}$ as

$$V_{k+1} = \begin{bmatrix} V_k, & \mathbf{v}_{\text{new}} \end{bmatrix}.$$

We refer to the subspace spanned by the columns of V_{k+1} as a *generalized Krylov subspace* (GKS).

The QR factorizations of AV_k and LV_k are updated to determine the QR factorizations of AV_{k+1} and LV_{k+1},

$$AV_{k+1} = \begin{bmatrix} Q_A & \widetilde{\mathbf{q}}_A \end{bmatrix} \begin{bmatrix} R_A & \mathbf{r}_A \\ \mathbf{0}^T & \tau_A \end{bmatrix},$$

$$LV_{k+1} = \begin{bmatrix} Q_L & \widetilde{\mathbf{q}}_L \end{bmatrix} \begin{bmatrix} R_L & \mathbf{r}_L \\ \mathbf{0}^T & \tau_L \end{bmatrix},$$

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $\mathbf{0}$ is the zero vector of appropriate dimension and

$$\begin{aligned}
\mathbf{r}_A &= Q_A^T(A\mathbf{v}_{\text{new}}), & \mathbf{q}_A &= A\mathbf{v}_{\text{new}} - Q_A\mathbf{r}_A,\\
\tau_A &= \|\mathbf{q}_A\|_2, & \widetilde{\mathbf{q}}_A &= \mathbf{q}_A/\tau_A,\\
\mathbf{r}_L &= Q_L^T(L\mathbf{v}_{\text{new}}), & \mathbf{q}_L &= L\mathbf{v}_{\text{new}} - Q_L\mathbf{r}_L,\\
\tau_L &= \|\mathbf{q}_L\|_2, & \widetilde{\mathbf{q}}_L &= \mathbf{q}_L/\tau_L;
\end{aligned}$$

see [18] for details. In the computations reported in Section 5, we set $V_0 = A^T\mathbf{b}^\delta / \left\|A^T\mathbf{b}^\delta\right\|_2$. Then $\widehat{k} = k+1$.

When many iterations are carried out with the MM method, the dimension of the GKS becomes large and this increases both the computational effort and the memory requirement for the iterative method. To avoid having to work with generalized Krylov subspaces of large dimension, a restarting strategy is proposed in [7]. After having carried out K iterations, for a user-specified positive integer K, the GKS is reinitialized. Thus, if $k \equiv 0 \bmod K$, then

$$V_{k+1} = \mathbf{y}^{(k+1)}/\|\mathbf{y}^{(k+1)}\|_2 \in \mathbb{R}^n.$$

The computations are summarized in Algorithm 1 below.

The iterates $\mathbf{x}_1, \mathbf{x}_2, \ldots$ generated by Algorithm 1 are guaranteed to converge to a stationary point of the functional (7) if the generalized Krylov subspace is not reinitialized, that is, if $K = \infty$, but not if K is finite. Nevertheless, numerous computed examples reported in [7, 8] indicate that restarting the generalized Krylov subspace periodically generally yields restorations of the same high quality at a significantly lower computational cost as when no restarts are carried out.

Theorem 1. *With the notation and assumptions above, let $K = \infty$. Then the iterates generated by Algorithm 1 converge to a stationary point of (7). If $q \geq 1$, then they converge to a global minimum of (7).*

Proof. The result follows from the observation that, after at most n iterations, we have $P_k = P$. This, together with [9, theorem 5], where we have to replace A and L with $AP^{-1/2}$ and $LP^{-1/2}$, respectively, yields the desired result. □

Theorem 2. *The functional (7) satisfies $\mathcal{J}_\varepsilon(\mathbf{y}^{(k+1)}) \leq \mathcal{J}_\varepsilon(\mathbf{y}^{(k)})$. It follows that there is a convergent subsequence $\{\mathbf{y}^{(k_j)}\}$.*

Proof. The result is shown by [8, proposition 1 and theorem 1] for the situation when $P = I$. The proof carries over to the case when P is symmetric positive definite. □

We conclude this section by observing that the proposed preconditioned method modifies both the projected problem and the approximation subspace, rather than just the latter as it would typically be the case for preconditioned Krylov solvers.

4 | The Choice of Regularization Parameter

The performance of Algorithm 1 depends on the choice of the regularization parameters $\mu > 0$ or η defined in (9). An imprudent choice of the regularization parameter may yield results of unnecessarily poor quality. To determine a suitable value of the regularization parameter μ, we proceed similarly as in [6]. Thus, we assume that a fairly accurate bound for the norm of the noise is available, that is, that a value of δ such that

$$\left\|\mathbf{b}^\delta - \mathbf{b}\right\|_2 \leq \delta \tag{20}$$

is known and that $\mathbf{b} \in \mathcal{R}(A)$. We propose a nonstationary version of Algorithm 1, where at each iteration k a value of the regularization parameter η_k is determined so that

$$\left\|AP^{-\frac{1}{2}}\mathbf{y} - \mathbf{b}^\delta\right\|_2 \leq \tau\delta. \tag{21}$$

Here $\tau > 1$ is a user-defined parameter. It will be chosen close to one in the computed examples presented in Section 5. The parameter η_k is a particular value of the parameter η. The latter parameter is a scaling of the regularization parameter μ; cf. (9). This criterion for choosing η_k is known as the discrepancy principle (DP); see, for example, [1, 19] for discussions. We will comment on the situation when no bound (20) is known.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ALGORITHM 1 | Preconditioned ℓ^2-ℓ^q.

Let $A \in \mathbb{R}^{m\times n}$, $L \in \mathbb{R}^{s\times n}$ and $\mathbf{b}^\delta \in \mathbb{R}^m$. Fix $0 < q \leq 2$, restarting iteration K, a maximum number of iterations $K_{\max}$, a smoothing parameter $\varepsilon > 0$, and a tolerance $\gamma > 0$;

Let $\mathbf{y}^{(0)}$ be an initial guess for $P^{\frac{1}{2}}\mathbf{x}^\dagger$;

$V_0 = \mathbf{y}^{(0)}/\|\mathbf{y}^{(0)}\|_2$;

Compute and store $A_0 = AV_0$ and $L_0 = LV_0$;

Compute the economic QR factorizations $\begin{cases} A_0 = Q_A R_A; \\ L_0 = Q_L R_L; \end{cases}$ $\mathbf{t}^{(0)} = \|\mathbf{y}^{(0)}\|_2$;

$\eta = \mu\varepsilon^{2-q}$;

for $k = 0, 1, \ldots, K_{\max}$ **do**

  Compute the eigendecomposition $R_A^T R_A = W\Lambda W^T$ using the SVD of R_A;

  $\mathbf{u}^{(k)} = Q_L R_L W(\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t}^{(k)}$;

  $\boldsymbol{\omega}_{\text{reg}}^{(k)} = \mathbf{u}^{(k)}\left(1 - \frac{\left((\mathbf{u}^{(k)})^2+\varepsilon^2\right)^{q/2-1}}{\varepsilon^2}\right)$;

  $\widehat{\mathbf{t}}^{(k+1)} = \arg\min_{\hat{\mathbf{t}}\in\mathbb{R}^{\hat{k}}}\left\{\left\|\begin{bmatrix} R_A \\ \sqrt{\eta}R_L \end{bmatrix}\hat{\mathbf{t}} - \begin{bmatrix} Q_A^T\mathbf{b}^\delta \\ \sqrt{\eta}Q_L^T\boldsymbol{\omega}_{\text{reg}}^{(k)} \end{bmatrix}\right\|_2^2\right\}$;

  $\mathbf{t}^{(k+1)} = W\left(\Lambda + \alpha I\right)^{\frac{1}{2}} W^T\widehat{\mathbf{t}}^{(k+1)}$;

  if $k > 1$ *and* $\|\mathbf{t}^{(k+1)} - \mathbf{t}^{(k)}\|_2 < \gamma\|\mathbf{t}^{(k)}\|_2$ **then**

    break;

  if $k \equiv 0 \bmod K$ **then**

    $V_{k+1} = V_k\mathbf{t}^{(k+1)}/\left\|V_k\mathbf{t}^{(k+1)}\right\|_2$;

    $A_{k+1} = A_k V_{k+1}$;

    $L_{k+1} = L_k V_{k+1}$;

    Compute the economic QR factorizations $\begin{cases} A_{k+1} = Q_A R_A; \\ L_{k+1} = Q_L R_L; \end{cases}$

  else

    $\mathbf{r}_A = A^T\left(Q_A R_A W(\Lambda + \alpha I)^{-\frac{1}{2}} W^T\mathbf{t}^{(k+1)} - \mathbf{b}^\delta\right)$;

    $\mathbf{r}_L = \eta L^T\left(Q_L R_L W(\Lambda + \alpha I)^{-\frac{1}{2}} W^T\mathbf{t}^{(k+1)} - \boldsymbol{\omega}_{\text{reg}}^{(k)}\right)$;

    $\widehat{\mathbf{r}} = \mathbf{r}_A + \mathbf{r}_L$;

    $\mathbf{r}^{(k+1)} = V_k W\left[(\Lambda + \alpha I)^{-\frac{1}{2}} - \alpha^{-\frac{1}{2}} I\right] W^T V_k^T\widehat{\mathbf{r}} + \alpha^{-\frac{1}{2}}\widehat{\mathbf{r}}$;

    $\mathbf{v}_{\text{new}} = \mathbf{r}^{(k+1)}/\|\mathbf{r}^{(k+1)}\|_2$;

    $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$;

    $A_{k+1} = [A_k, A\mathbf{v}_{\text{new}}]$;

    $L_{k+1} = [L_k, L\mathbf{v}_{\text{new}}]$;

    Update Q_A, Q_L, R_A, and R_L as in (20);

  $\mathbf{x}^{(k+1)} = V_k W\left(\Lambda + \alpha I\right)^{-\frac{1}{2}} W^T\mathbf{t}^{(k+1)}$;

Let $\mathbf{x}_{\eta_k}^{(k)}$ denote the kth iterate produced by Algorithm 1 with regularization parameter $\eta = \eta_k$. To satisfy the DP at the kth iteration, we choose η_k such that

$$\left\|AP_k^{-\frac{1}{2}}\mathbf{y}_{\eta_k}^{(k)} - \mathbf{b}^\delta\right\|_2 = \tau\delta,$$

since $\mathbf{x}_{\eta_k}^{(k)} = P_k^{-\frac{1}{2}}\mathbf{y}_{\eta_k}^{(k)}$.

Let $\beta = 1/\eta$. To determine the regularization parameter η, we have to compute the smallest positive zero of the function

$$f(\beta) = \left\|AP^{-\frac{1}{2}}\mathbf{y}_\beta^{(k)} - \mathbf{b}^\delta\right\|_2^2 - \tau^2\delta^2. \tag{22}$$

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

This function typically has a unique positive zero. We compute it by explicitly writing the expression for $\mathbf{y}^{(k+1)}$,

$$\begin{cases} \mathbf{y}^{(k+1)} = V_k \mathbf{t}^{(k+1)}, \\ \mathbf{t}^{(k+1)} = \arg\min_{\mathbf{t}} \left\{ \frac{1}{2} \left\| A\widetilde{P}^{-\frac{1}{2}} V_k \mathbf{t} - \mathbf{b}^{\delta} \right\|_2^2 + \frac{1}{2\beta} \left\| L\widetilde{P}^{-\frac{1}{2}} V_k \mathbf{t} - \boldsymbol{\omega}_{\text{reg}}^{(k)} \right\|_2^2 \right\}. \end{cases}$$

Hence,

$$\mathbf{t}^{(k+1)} = \arg\min_{\mathbf{t}} \left\| \begin{bmatrix} R_A(\Lambda + \alpha I)^{-\frac{1}{2}} W^T \\ \beta^{-1/2} R_L(\Lambda + \alpha I)^{-\frac{1}{2}} W^T \end{bmatrix} \mathbf{t} - \begin{bmatrix} Q_A^T \mathbf{b}^{\delta} \\ \beta^{-1/2} Q_L^T \boldsymbol{\omega}_{\text{reg}}^{(k)} \end{bmatrix} \right\|_2^2.$$

Substituting $\mathbf{z} = W(\Lambda + \alpha I)^{-\frac{1}{2}} W^T \mathbf{t}$ into the above expression gives

$$\mathbf{z}^{(k+1)} = \arg\min_{\mathbf{z}} \left\| \begin{bmatrix} R_A \\ \beta^{-1/2} R_L \end{bmatrix} \mathbf{z} - \begin{bmatrix} Q_A^T \mathbf{b}^{\delta} \\ \beta^{-1/2} Q_L^T \boldsymbol{\omega}_{\text{reg}}^{(k)} \end{bmatrix} \right\|_2^2. \tag{23}$$

Introduce the generalized singular value decomposition (GSVD) of the matrix pair $\{R_A, R_L\}$:

$$\begin{cases} R_A = U \Sigma X^T, \\ R_L = V \Delta X^T, \end{cases} \tag{24}$$

where $U, V \in \mathbb{R}^{\hat{k} \times \hat{k}}$ are orthogonal matrices, $X \in \mathbb{R}^{\hat{k} \times \hat{k}}$ is a non-singular matrix, and $\Sigma, \Delta \in \mathbb{R}^{\hat{k} \times \hat{k}}$ are diagonal matrices, whose diagonal elements, denoted by σ_j and δ_j, respectively, are nonnegative and scaled so that $\sigma_j^2 + \delta_j^2 = 1$ for $j = 1, 2, \ldots \hat{k}$. This scaling is the standard one used in the GSVD; see, for example, [20]. An alternative scaling that may yield somewhat higher accuracy is described in [21]. Solution methods for (23) that circumvent the computation of the GSVD are discussed in [22, 23]. We will not dwell on these details.

Assume that the GSVD (24) is available. We will describe how the GSVD can be applied to simplify the right-hand side of (22). Substituting the decompositions (24) into (22), we obtain

$$\begin{aligned} f(\beta) &= \left\| Q_A R_A \left(R_A^T R_A + \frac{1}{\beta} R_L^T R_L \right)^{-1} \left(R_A^T Q_A^T \mathbf{b}^{\delta} + \frac{1}{\beta} R_L^T Q_L^T \boldsymbol{\omega}_{\text{reg}}^{(k)} \right) - \mathbf{b}^{\delta} \right\|_2^2 - \tau^2 \delta^2 \\ &= \left\| \Sigma \left(\Sigma^2 + \frac{1}{\beta} \Delta^2 \right)^{-1} \left(\Sigma \hat{\mathbf{b}} + \frac{1}{\beta} \Delta \hat{\boldsymbol{\omega}} \right) - \hat{\mathbf{b}} \right\|_2^2 + \left\| \mathbf{b}^{\delta} \right\|_2^2 - \left\| Q_A^T \mathbf{b}^{\delta} \right\|_2^2 - \tau^2 \delta^2, \end{aligned}$$

where $\hat{\mathbf{b}} = U^T Q_A^T \mathbf{b}^{\delta}$ and $\hat{\boldsymbol{\omega}} = V^T Q_L^T \boldsymbol{\omega}_{\text{reg}}^{(k)}$. This gives

$$f(\beta) = \sum_{j=1}^{\hat{k}} \left(\frac{\left(\sigma_j \delta_j \hat{\boldsymbol{\omega}}_j - \delta_j^2 \hat{b}_j \right)^2}{\left(\beta \sigma_j^2 + \delta_j^2 \right)} \right) + \left\| \mathbf{b}^{\delta} \right\|_2^2 - \left\| Q_A^T \mathbf{b}^{\delta} \right\|_2^2 - \tau^2 \delta^2. \tag{25}$$

It is easy to see that the function f is a monotonically decreasing convex function of β and that, under mild assumptions, it has a unique zero. We will, in the computed examples of the following section determine the zero of f with a few iterations of Newton's method, however, other zero-finders also can be used; see, for example, [24, 25]. We note that the change of variable $\beta = 1/\eta$ in (22) secures that f is a convex function of its argument.

Remark 1. For the sake of brevity, this paper does not present convergence results related to the non-stationary version of our proposed algorithm. Nevertheless, one may expect similar results to the ones in [6] to hold. In [6], the authors showed that, if A is of full column rank, then the sequence $\mathbf{x}^{(k)}$ admits a converging subsequence, and the regularization property holds. That is, if we denote the computed solution with noise level δ by $\mathbf{x}^{\delta}$, then

$$\lim_{\delta \to 0} \left\| \mathbf{x}^{\delta} - A^{\dagger} \boldsymbol{b} \right\|_2 = 0.$$

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Vast numerical experience suggests that, both in [6] and in this work, the whole sequence $\mathbf{x}^{(k)}$ converges and that, when the noise level is small enough, the reconstructions computed by the method are accurate approximations of $A^{\dagger}\boldsymbol{b}$.

We finally remark that when no bound (20) for the error in $\mathbf{b}^{\delta}$ is known or if $\mathbf{b} \notin \mathcal{R}(A)$, the discrepancy principle cannot be applied to determine the regularization parameter η. A variety of so-called heuristic methods for determining η, including generalized cross-validation, cross-validation, and the L-curve criterion can be applied; see [3, 26–28] for discussions and illustrations of these methods.

5 | Numerical Examples

This section reports a few computed examples that illustrate the performance of Algorithm 1 applied to image deblurring and computerized tomography (CT). We measure the quality of the computed solutions with the relative reconstruction error

$$\mathrm{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}^{\dagger}\|_2}{\|\mathbf{x}^{\dagger}\|_2},$$

where $\mathbf{x}$ denotes a vectorization of the computed image and $\mathbf{x}^{\dagger}$ is the solution of (2) of minimal Euclidean norm, which is a vectorization of the desired image. The latter image is assumed to be represented by a matrix of $n_1 \times n_2$ pixels. Thus, $\mathbf{x}^{\dagger} \in \mathbb{R}^n$ with $n = n_1 n_2$.

We first consider the discretized gradient regularization matrix

$$L_{\nabla} = \begin{bmatrix} I_{n_1} \otimes L_1^{(2)} \\ L_1^{(1)} \otimes I_{n_2} \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product, $I_{n_j} \in \mathbb{R}^{n_j \times n_j}$ is the identity matrix of order n_j, and

$$L_1^{(j)} = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 1 & & & -1 \end{bmatrix} \in \mathbb{R}^{n_j \times n_j}, \quad j \in \{1, 2\}$$

is a circulant matrix with nontrivial diagonal and superdiagonal entries; see, for example, [29].

In all our experiments, we set $q = 0.1$, $\epsilon = 1$, and choose the tolerance $\gamma = 10^{-4}$ in Algorithm 1. We remark that this value of ϵ is well-suited for grayscale images that are represented by 8-bit pixel values in the interval $[0, 255]$. To illustrate the performance of Algorithm 1 without restarts, we set $K = \infty$ in some runs of the algorithm.

All computations were carried out in MATLAB version R2021b running in Windows 11 on a laptop computer with an 11th Gen Intel Core i7 processor with 16 GB of RAM. We used "double precision" arithmetic, that is, all floating point computations were carried out with about 15 significant decimal digits.

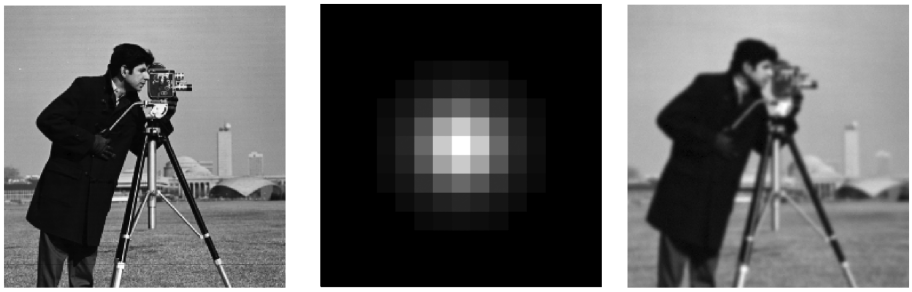
5.1 | Stationary μ

We compare Algorithm 1 with the corresponding unpreconditioned algorithm (i.e., Algorithm 1 with $P = I$) and first consider the stationary version of these algorithms. In these versions, the value of μ is kept fixed during the iterations. We let $\mu = 0.1$ since this value yields quite accurate computed solutions. The restorations obtained are quite similar regardless of the choice of the operator L. We let $L = L_{\nabla}$ in the experiments.

Our first example is concerned with deblurring an available blur- and noise-contaminated version of the Cameraman image. Figure 1a shows the "exact" blur- and noise-free image. We blur this image with the PSF shown in Figure 1b and then add 1% of white Gaussian noise $\boldsymbol{\eta}$; cf. (1). Thus,

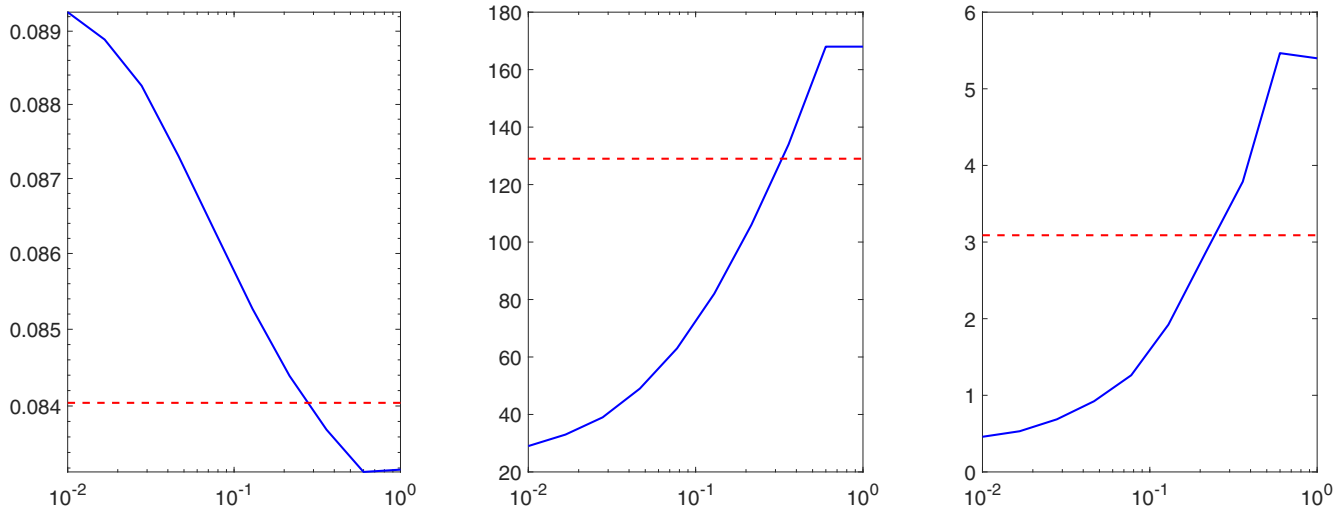
$$\|\boldsymbol{\eta}\|_2 = 0.01\|\mathbf{b}\|_2.$$

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) (b) (c)

FIGURE 1 | Cameraman example: (a) True image (240 × 240 pixels), (b) PSF (15 × 15 pixels), (c) Blurred and noisy image with 1% of white Gaussian noise (240 × 240 pixels).



(a) RRE vs α (b) Iterations vs α (c) CPU time vs α

FIGURE 2 | Cameraman example with $L = L_\nabla$ and 1% of noise: Comparison of the unpreconditioned ℓ^2-ℓ^q method (dashed red graphs) and Algorithm 1 (solid blue graphs) for $\mu = 0.1$: (a) RRE, (b) number of iterations, and (c) CPU time in seconds as functions of α.



(a) (b) (c)

FIGURE 3 | Cameraman example with $L = L_\nabla$ and 1% of noise: Reconstructions obtained for $\mu = 0.1$ by (a) the unpreconditioned ℓ^2-ℓ^q method, (b) Algorithm 1 with $\alpha = 0.1$, and (c) Algorithm 1 with $\alpha = 0.01$.

This gives the blurred and noisy image shown in Figure 1c.

We ran Algorithm 1 for a large number of α-values. Figure 2 shows the RRE, the number of iterations demanded to satisfy the stopping criterion, and the CPU time required as a function of α, and compares these values to those obtained with the unpreconditioned ℓ^2-ℓ^q algorithm. We can observe that smaller values of α reduce the CPU time required by Algorithm 1 significantly, but give restorations with slightly larger RRE values. Note that the CPU time needed to apply the preconditioner is negligible. Figure 3 shows reconstructions obtained with the unpreconditioned ℓ^2-ℓ^q algorithm and

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

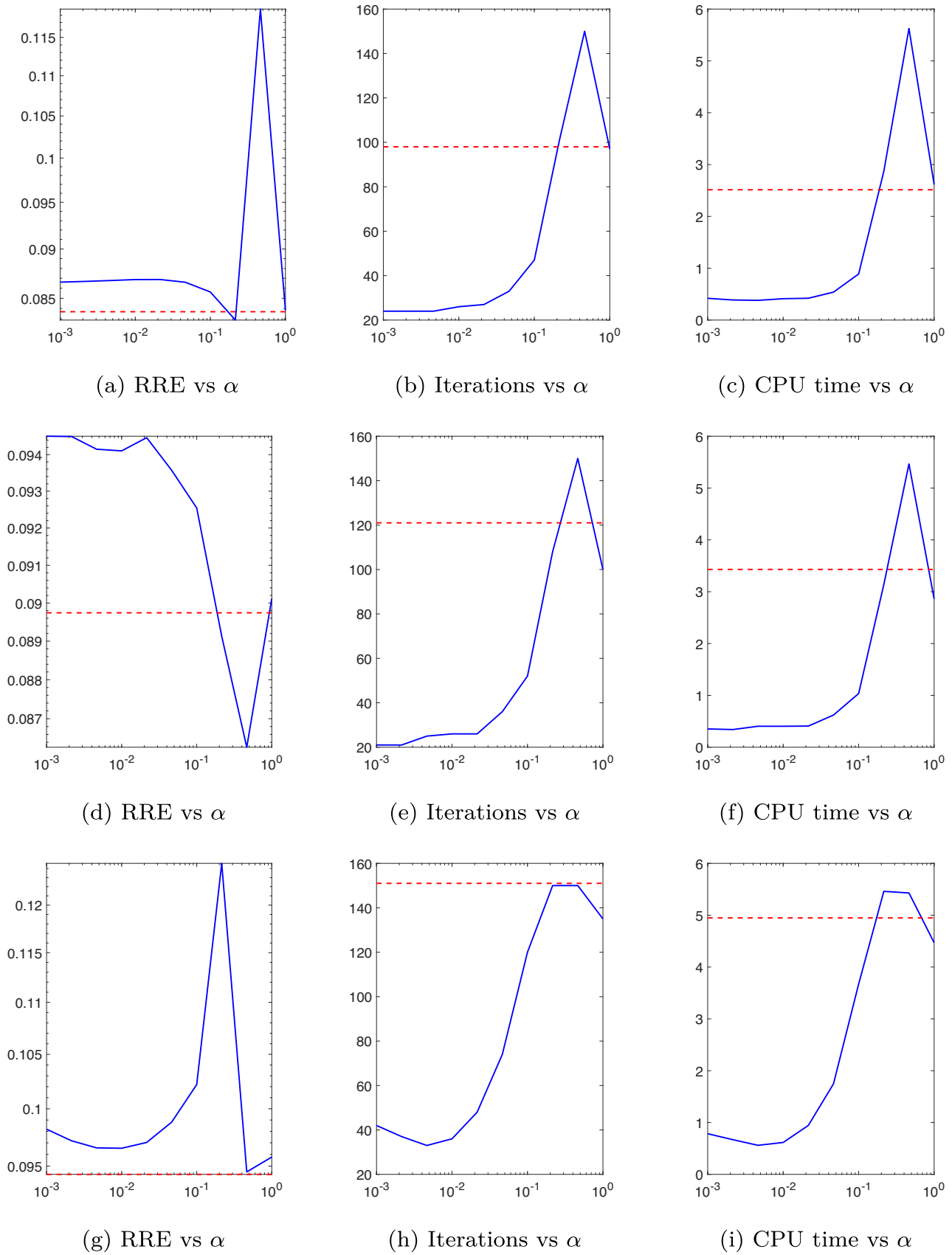


FIGURE 4 | Cameraman example with $L = L_{\nabla}$: Comparison of the non-stationary unpreconditioned ℓ^2-ℓ^q algorithm (dashed red graphs) and Algorithm 1 (solid blue graphs) for several values of α. The left-hand side panels report the RRE, the middle panels show the number of iterations, and the right-hand side panels depict the CPU time in seconds as a function of α. The first row reports results for 0.5% Gaussian noise, the second for 1% Gaussian noise, and the third for 2% Gaussian noise.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 | Cameraman example with $L = L_{\mathrm{V}}$: Comparison of the nonstationary unpreconditioned ℓ^2-ℓ^q method and Algorithm 1 for two values of α and three noise levels.

Noise level	Method	RRE	Iterations	CPU time
0.5%	Unpreconditioned ℓ^2-ℓ^q	0.0837	98	2.5028
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.0856	47	0.9380
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0869	26	0.4230
1%	Unpreconditioned ℓ^2-ℓ^q	0.0897	121	3.6922
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.0924	54	1.1843
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0944	25	0.4477
2%	Unpreconditioned ℓ^2-ℓ^q	0.0942	153	5.7077
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.1022	120	3.8819
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0965	36	0.6505

with Algorithm 1 for $\alpha = 0.1$ and $\alpha = 0.01$. We observe that the reconstructed images obtained with these algorithms look the same to the naked eye, even though the RRE of the reconstruction produced by Algorithm 1 is slightly larger. The CPU time required by Algorithm 1 is significantly smaller than for the unpreconditioned algorithm. Specifically, the RRE obtained with the unpreconditioned ℓ^2-ℓ^q algorithm is 0.0840, while Algorithm 1 with $\alpha = 0.1$ produces a reconstruction with RRE = 0.0858. The former method demanded 129 iterations to satisfy the stopping criterion and this required 3.2078 s, while Algorithm 1 carried out 71 iterations in only 1.5127 s.

5.2 | Application of the Discrepancy Principle

We consider a nonstationary version of Algorithm 1, in which the parameter μ is determined by the discrepancy principle (21). To use this parameter choice rule, we need to set the parameter $\tau > 1$ in (21). Numerical experience suggests that a value of τ close to 1 is appropriate. We set $\tau = 1.01$ in the computed examples.

As described in Section 4, we can determine a new value of the regularization parameter μ at each iteration by computing the zero of the function f in (25) by a few iterations of Newton's method. However, in our experience, this approach to determine μ, or equivalently η, for the preconditioned ℓ^2-ℓ^q algorithms may display instability and determine values of μ (and thereby of η) that are too small. This happens in particular during the first iterations when the dimension of the Krylov subspace is small. We therefore introduce a safeguard: if the computed value of $\eta = \eta_k$ is smaller than 10^{-8}, then we set $\eta = \eta_k = 0.1$. This safeguarded zero-finder avoided numerical instability and furnished accurate image reconstructions in all our computed examples.

We first consider the image reconstruction problem described in Section 5.1. The first row of Figure 4 reports the RRE, the number of iterations, and the CPU time required by Algorithm 1 for 0.5% Gaussian noise as functions of α. The first lines of Table 1 reports results obtained for $\alpha \in \{0.01,\ 0.1\}$ for the same amount of noise. These results are compared to those achieved with the unpreconditioned ℓ^2-ℓ^q algorithms. We can observe that the quality of the reconstruction is about the same for the preconditioned and unpreconditioned algorithms, and that the use of the preconditioner decreases the CPU time. The last two rows of Figure 4 and the last six rows of Table 1 show results for the noise levels 1% and 2%. The relative performance of the methods for these noise levels is similar to that for 0.5% noise.

Figure 3 shows the reconstruction obtained for $\alpha = 0.1$ and $\alpha = 0.01$, with 1% noise.

5.3 | A Framelet Regularization Operator

Consider the framelet analysis operator $L_{\mathrm{W}} \in \mathbb{R}^{9n\times n}$associated with a two-level linear B-spline decomposition. We refer the interested reader to, for example, [30, 31], for more details. Here we only note that $L_{\mathrm{W}}^T L_{\mathrm{W}} = I_n$. When $\mathbf{x}$ is a vectorization of a natural image, the vector $L_{\mathrm{W}}\mathbf{x}$ typically is very sparse.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

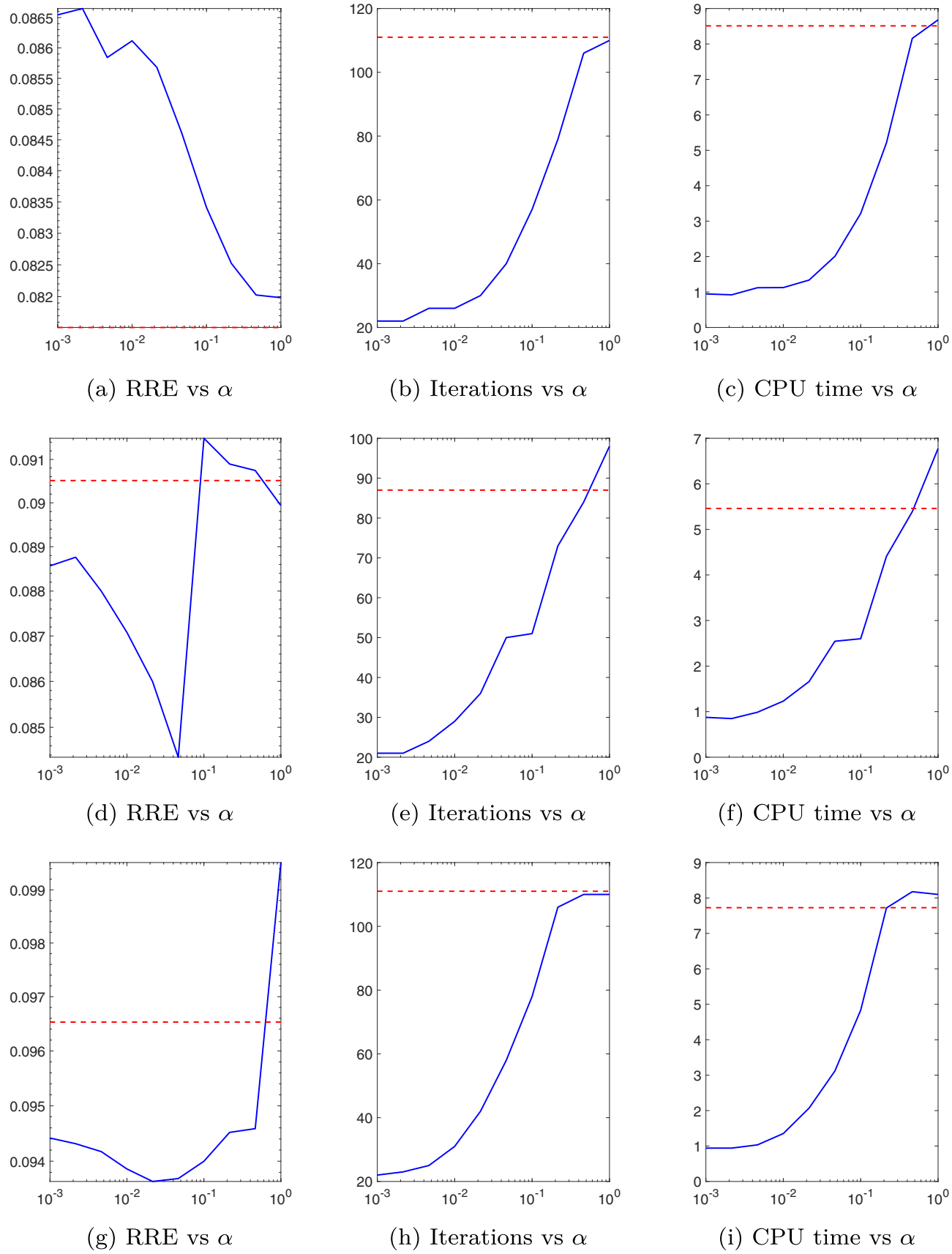
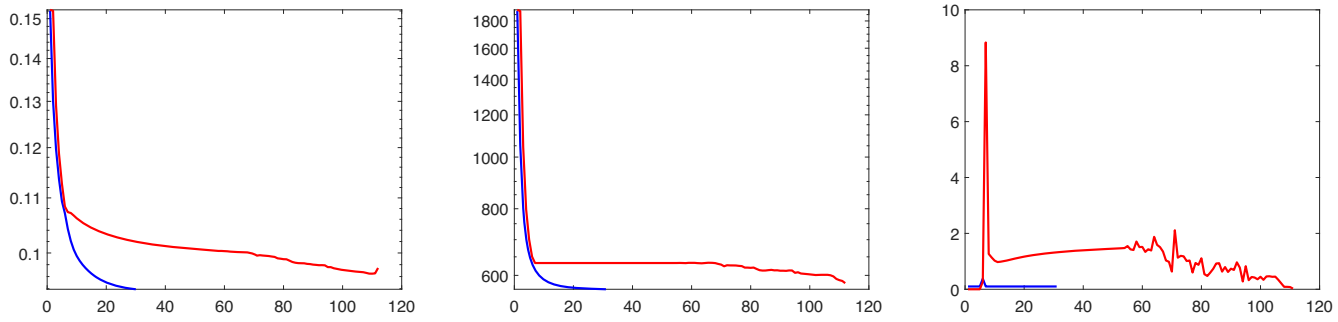


FIGURE 5 | Cameraman example with $L = L_{\mathrm{W}}$: Comparison of the non-stationary unpreconditioned ℓ^2-ℓ^q method (dashed red graphs) and Algorithm 1 (solid blue graphs). The left-hand side panels report the RRE, the middle panels show the number of iterations, and the right-hand side panels display the CPU time in seconds as a function of α. The first row reports results for 0.5% Gaussian noise, the second row for 1% Gaussian noise, and the third row for 2% Gaussian noise.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 | Cameraman example with $L = L_W$: Comparison of the nonstationary standard ℓ^2-ℓ^q method and the preconditioned method for two values of α and three noise levels.

Noise level	Method	RRE	Iterations	CPU time
0.5%	Unprecondiotioned ℓ^2-ℓ^q	0.0815	111	8.5771
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.0834	57	3.2810
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0861	26	1.0901
1%	Unpreconditioned ℓ^2-ℓ^q	0.0905	87	5.8017
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.0915	51	2.7286
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0871	29	1.2809
2%	Unpreconditioned ℓ^2-ℓ^q	0.0965	111	8.4351
	Prec. ℓ^2-ℓ^q with $\alpha = 0.1$	0.0940	78	5.0533
	Prec. ℓ^2-ℓ^q with $\alpha = 0.01$	0.0939	31	1.4526



(a) RRE vs iterations (b) Residual vs iterations (c) μ vs iterations

FIGURE 6 | Cameraman example with $L = L_W$, 2% Gaussian noise. Comparison of the non-stationary unpreconditioned ℓ^2-ℓ^q algorithm (red lines) and Algorithm 1 with $\alpha = \sigma_1/100 = 0.01$ (blue lines). In all plots on the horizontal axis, we report the number of iterations. On the left we show the evolution of the RRE, in the middle panel the residual norm, and in the right one the values of μ computed by the discrepancy principle.



(a) (b) (c)

FIGURE 7 | Cameraman example with $L = L_W$, 2% Gaussian noise: (a) Noisy and blurred observed image, (b) reconstruction obtained with the unpreconditioned ℓ^2-ℓ^q method, (c) Algorithm 1 with $\alpha = 0.01$.

We consider the noise levels 0.5%, 1%, and 2%, and report the computed results in Figure 5 and Table 2. The quality of the computed reconstructions determined is similar to what was obtained with $L = L_V$.

The acceleration of the convergence is shown in Figure 6, where it can be observed that both the RRE and the norm of the residual decrease faster for the preconditioned method than for the standard one. Figure 7 shows the computed reconstructions.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

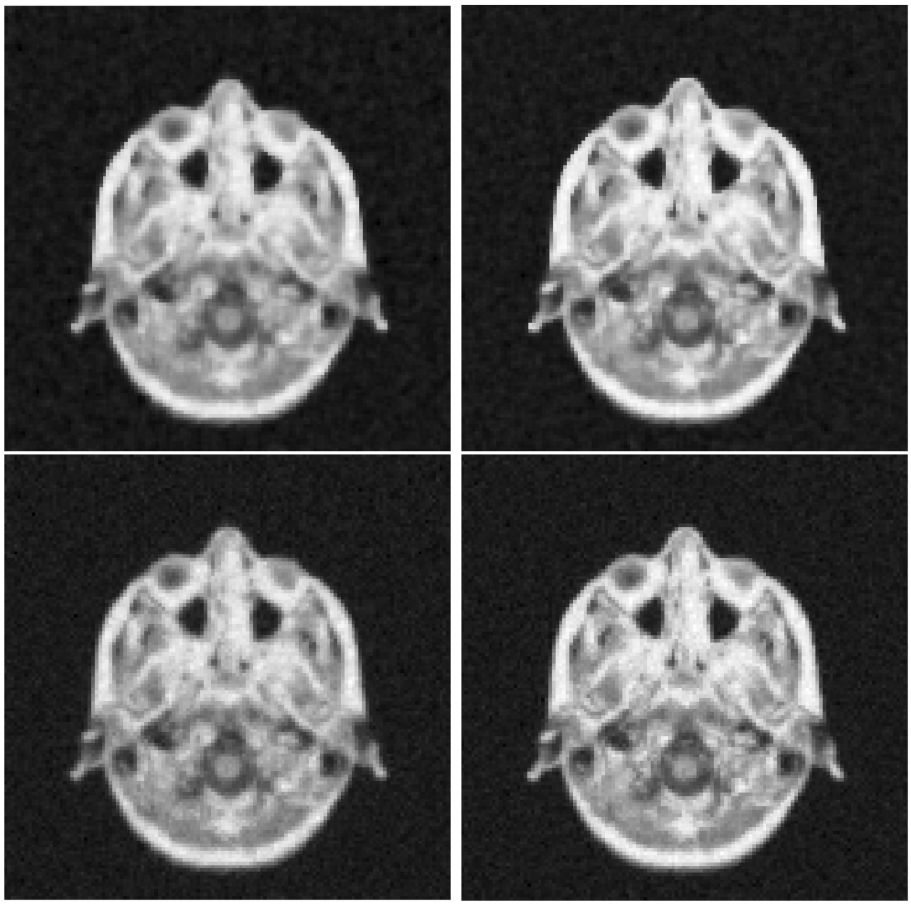


FIGURE 8 | MRI test image with 3% Gaussian noise. On the left-hand side: reconstructions obtained with the unpreconditioned $[\ell^2]$-$[\ell^q]$ method. On the right-hand side: reconstructions determined by Algorithm 1 with $\alpha = \tilde{\sigma}_1/100 = 1.496$. For the first row, we used $L = L_\nabla$, while for the second row we used framelets and $L = L_W$.

5.4 | Computerized Tomography and the Choice of α

We consider a CT example. The phantom is a 128×128 grayscale slice of a human head taken from the MRI dataset provided by MATLAB. We start by shining 181 X-rays on the phantom at 180 equispaced angles in $[0, \pi]$ using the MATLAB package IR Tools [32]. This produces a 181×180 sinogram, which we corrupt with Gaussian noise. In the same framework, we consider a limited angle example, where the X-rays are only shone from 90 equispaced angles in $[0, \pi/2]$. This simulates a scenario where the CT scan cannot fully rotate around the object.

Our preconditioner P in (5) requires the choice of the parameter $\alpha > 0$, which shifts the spectrum of $A^T A$. The choice of α is important to speed up the convergence without introducing instability. Indeed, as observed in the previous numerical results, choosing α too large does not accelerate the method, while a too small value of α leads to fast convergence, but may negatively affect the quality of the computed restoration. We found that the choice of α should depend on the eigenvalues of $A^T A$, that is, on the singular values of A. According to the previous numerical results on image deblurring, the choice $\alpha = 0.01$ usually gives fast convergence without spoiling the quality of the reconstructed image. For image deblurring problems, the PSF yields a weighted average of the exact image and so the largest singular value σ_1 of A is about one. Therefore, we propose the choice

$$\alpha = \frac{\tilde{\sigma}_1}{100},$$

where $\tilde{\sigma}_1$ is an estimate σ_1.

For the present computerized tomography examples, we carried out the same comparisons as for the image deblurring example above. For the full-angle case, the discussion on the choice of α for image deblurring problems suggests that

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

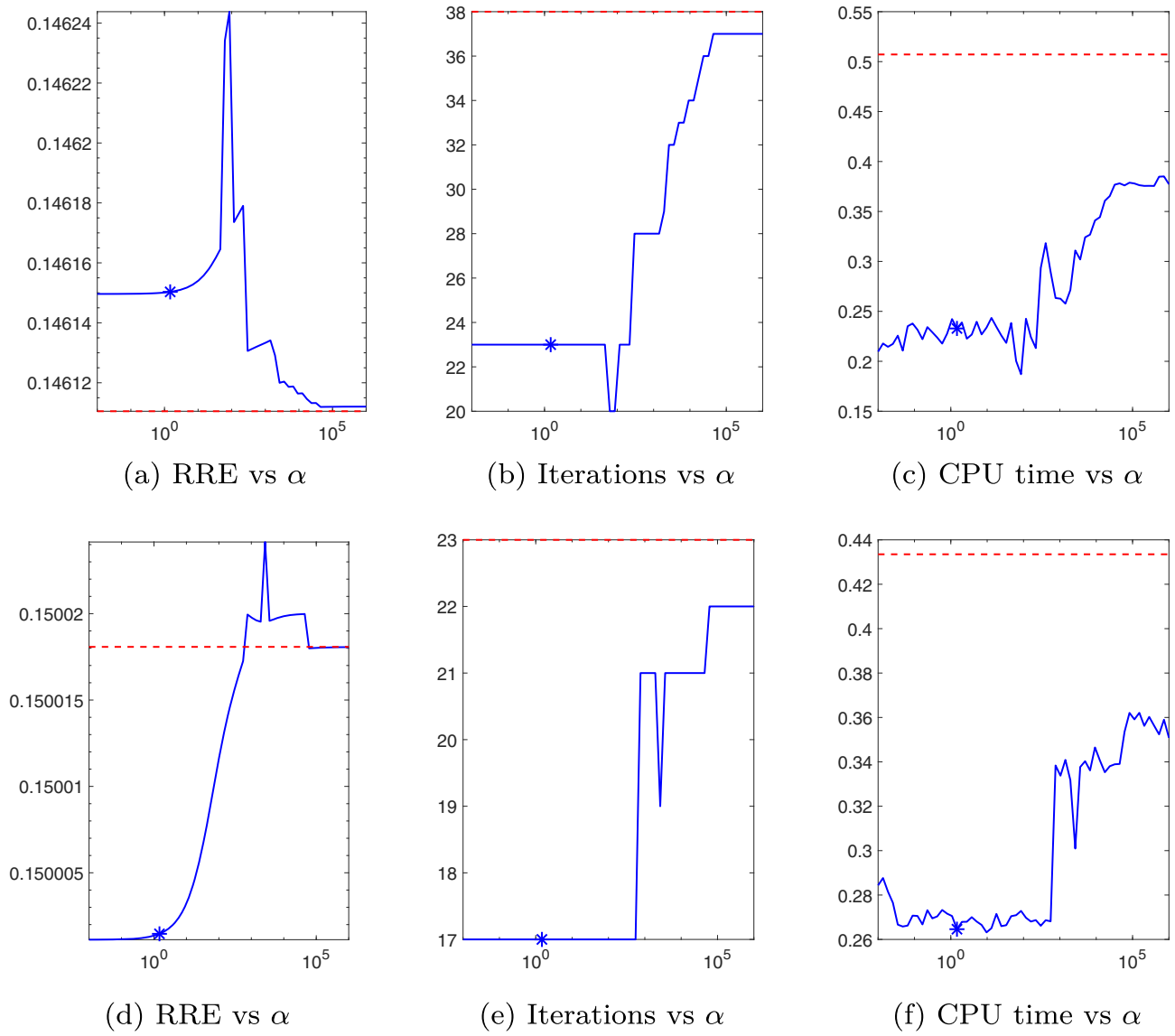


FIGURE 9 | MRI test image with 3% Gaussian noise: comparison of the non-stationary unpreconditioned ℓ^2-ℓ^q method (dashed red graphs) and Algorithm 1 (solid blue graphs). The left-hand side panels report the RRE, the middle panels show the number of iterations, and the right-hand side panels display the CPU time in seconds as a function of α. The first row reports results for $L = L_\nabla$ and the second row for $L = L_W$. The $*$ marks the results for $\alpha = \tilde{\sigma}_1/100 = 1.496$.

$$\alpha = \frac{\tilde{\sigma}_1}{100} = 1.496, \tag{26}$$

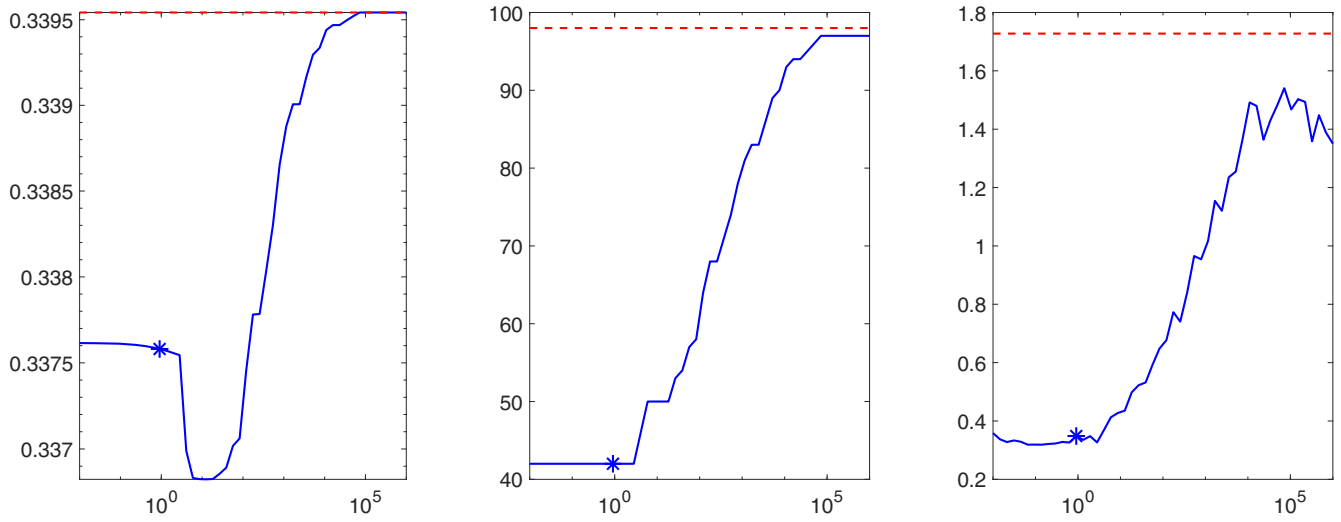
where $\tilde{\sigma}_1$ is an approximation of the largest singular value of the matrix of the regularized least-squares problem solved. This matrix is not explicitly formed. Instead, we compute $\tilde{\sigma}_1$ by using the MATLAB command svds$(A, 1)$. This command uses an implicitly restarted Golub-Kahan bidiagonalization method described in [33] to evaluate $\tilde{\sigma}_1$.

Figure 8 shows the reconstructed images obtained for α given by (26) and Figure 9 displays the RRE, the number of iterations, and the CPU time for various values of α. Results for the value (26) of α are marked with $*$.

We note that the results obtained when using $L = L_\nabla$ and $L = L_W$ (with framelets) are similar in quality, with a small advantage for L_∇ in terms of the RRE.

Finally, we report in Figure 10 analogous plots, that show that the advantages given by the preconditioning still hold in a much more ill-posed problem, that is, a limited angle example. The reconstructions are omitted since, as in the previous example, the standard and preconditioned methods produce visually identical images.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(a) RRE vs α (b) Iterations vs α (c) CPU time vs α

FIGURE 10 | MRI test image with limited angle, 3% Gaussian noise, $L = L_\nabla$: comparison of the non-stationary unpreconditioned ℓ^2-ℓ^q method (dashed red graphs) and Algorithm 1 (solid blue graphs). The left panel reports the RRE, the middle panel shows the number of iterations, and the right panel displays the CPU time in seconds as a function of α. The $*$ marks the results for $\alpha = \tilde{\sigma}_1/100 = 1.081$.

6 | Conclusions

This paper introduces a preconditioning strategy for ℓ^2-ℓ^q minimization. The aim of the preconditioner is to speed up the convergence of the iterative method and thereby reduce the computational effort, while maintaining the high quality of the reconstructed images achieved with the unpreconditioned method. Numerical examples of image deblurring and computerized tomography illustrate that the proposed preconditioner satisfies these properties. We have shown that the choice of the parameter α is crucial to obtain a fast and accurate method. Although we have provided some insight into how to choose this parameter, automatic rules and non-stationary sequences α_k will be a matter of future research. In the latter case, additional assumptions on α_k, for example, the convergence to a sufficiently large value $\overline{\alpha} > 0$ may be required; see [34].

Acknowledgments

The authors would like to thank the referees for their comments that led to improvements in the presentation. The first three authors are members of the GNCS group of INdAM. Open access publishing facilitated by Universita degli Studi dell'Insubria, as part of the Wiley - CRUI-CARE agreement.

Funding

This work was supported by the PRIN 2022 project "Italian Ministry of University and Research (MUR)" (Grant No. 2022ANC8HL), the INdAM-GNCS 2024 Project "Italian National Institute of Advanced Mathematics (INdAM)" (Grant No. CUP_E53C24001950001), and the PRIN 2022 PNRR project "Italian Ministry of University and Research (MUR)" (Grant No. P2022PMEN2).

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems* (Kluwer, 1996).

2. A. Buccini, O. De Cruz Cabrera, M. Donatelli, A. Martinelli, and L. Reichel, "Large-Scale Regression With Non-Convex Loss and Penalty," *Applied Numerical Mathematics* 157 (2020): 590–601.

3. P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion* (SIAM, 1998).

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4. A. Buccini, G. Huang, L. Reichel, and F. Yin, "On the Choice of Regularization Matrix for an ℓ_2-ℓ_q Minimization Method for Image Restoration," *Applied Numerical Mathematics* 164 (2021): 211–221.

5. A. Buccini, M. Pasha, and L. Reichel, "Modulus-Based Iterative Methods for Constrained ℓ_p–ℓ_q Minimization," *Inverse Problems* 36, no. 8 (2020): 084001.

6. A. Buccini and L. Reichel, "An ℓ^2-ℓ^q Regularization Method for Large Discrete Ill-Posed Problems," *Journal of Scientific Computing* 78 (2019): 1526–1549.

7. A. Buccini and L. Reichel, "Limited Memory Restarted ℓ^p–ℓ^q Minimization Methods Using Generalized Krylov Subspaces," *Advances in Computational Mathematics* 49 (2023): 26.

8. A. Buccini and L. Reichel, "Software for Limited Memory Restarted ℓ^p–ℓ^q Minimization Methods Using Generalized Krylov Subspaces," *Electronic Transactions on Numerical Analysis* 61 (2024): 66–91.

9. G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, "Majorization-Minimization Generalized Krylov Subspace Methods for ℓ_p–ℓ_q Optimization Applied to Image Restoration," *BIT Numerical Mathematics* 57 (2017): 351–378.

10. A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, "A Generalized Krylov Subspace Method for ℓ_p–ℓ_q Minimization," *SIAM Journal on Scientific Computing* 37 (2015): S30–S50.

11. M. Pasha, A. K. Saibaba, S. Gazzola, M. I. Español, and E. de Sturler, "A Computational Framework for Edge-Preserving Regularization in Dynamic Inverse Problems," *Electronic Transactions on Numerical Analysis* 58 (2023): 486–516.

12. S. Gazzola, P. Novati, and M. R. Russo, "On Krylov Projection Methods and Tikhonov Regularization," *Electronic Transactions on Numerical Analysis* 44 (2015): 83–123.

13. A. Buccini and L. Reichel, "Generalized Cross Validation for ℓ^p–ℓ^q Minimization," *Numerical Algorithms* 88 (2021): 1595–1616.

14. A. Buccini, M. Donatelli, and L. Reichel, "Theoretical and Numerical Aspects of a Non-Stationary Preconditioned Iterative Method for Linear Discrete Ill-Posed Problems," *Journal of Computational and Applied Mathematics* 423 (2023): 114940.

15. A. Buccini, S. Gazzola, L. Onisk, M. Pasha, and L. Reichel, "Projected Iterated Tikhonov in General Form With Adaptive Choice of the Regularization Parameter," *Numerical Algorithms* 100 (2025): 1617–1637.

16. A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences* 2 (2009): 183–202.

17. J. Lampe, L. Reichel, and H. Voss, "Large-Scale Tikhonov Regularization via Reduction by Orthogonal Projection," *Linear Algebra and Its Applications* 436 (2012): 2845–2865.

18. J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, "Reorthogonalization and Stable Algorithms for Updating the Gram–Schmidt QR Factorization," *Mathematics of Computation* 30 (1976): 772–795.

19. A. Buccini, M. Pragliola, L. Reichel, and F. Sgallari, "A Comparison of Parameter Choice Rules for ℓ_p–ℓ_q Minimization," *Annali dell'Universita' di Ferrara* 68 (2022): 441–463.

20. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins University Press, 2013).

21. L. Dykes, S. Noschese, and L. Reichel, "Rescaling the GSVD With Application to Ill-Posed Problems," *Numerical Algorithms* 68 (2015): 531–545.

22. L. Dykes and L. Reichel, "On the Reduction of Tikhonov Minimization Problems and the Construction of Regularization Matrices," *Numerical Algorithms* 60 (2012): 683–696.

23. L. Dykes and L. Reichel, "Simplified GSVD Computations for the Solution of Linear Discrete Ill-Posed Problems," *Journal of Computational and Applied Mathematics* 255 (2013): 15–27.

24. A. Buccini, M. Pasha, and L. Reichel, "Generalized Singular Value Decomposition With Iterated Tikhonov Regularization," *Journal of Computational and Applied Mathematics* 373 (2020): 112276.

25. L. Reichel and A. Shyshkov, "A New Zero-Finder for Tikhonov Regularization," *BIT Numerical Mathematics* 48 (2008): 627–643.

26. S. Kindermann, "Convergence Analysis of Minimization-Based Noise Level-Free Parameter Choice Rules for Linear Ill-Posed Problems," *Electronic Transactions on Numerical Analysis* 38 (2011): 233–257.

27. S. Kindermann and K. Raik, "A Simplified L-Curve as Error Estimator," *Electronic Transactions on Numerical Analysis* 53 (2020): 213–238.

28. L. Reichel and G. Rodriguez, "Old and New Parameter Choice Rules for Discrete Ill-Posed Problems," *Numerical Algorithms* 63 (2013): 65–87.

29. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D* 60, no. 1–4 (1992): 259–268.

30. J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman Methods and Frame Based Image Restoration," *Multiscale Modeling and Simulation* 8, no. 2 (2009): 337–369.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

31. J.-F. Cai, S. Osher, and Z. Shen, "Linearized Bregman Iterations for Frame-Based Image Deblurring," *SIAM Journal on Imaging Sciences* 2, no. 1 (2009): 226–252.

32. S. Gazzola, P. C. Hansen, and J. G. Nagy, "IR Tools: A MATLAB Package of Iterative Regularization Methods and Large-Scale Test Problems," *Numerical Algorithms* 81 (2019): 773–811.

33. J. Baglama and L. Reichel, "Augmented Implicitly Restarted Lanczos Bidiagonalization Methods," *SIAM Journal on Scientific Computing* 27, no. 1 (2005): 19–42.

34. J. Huang, M. Donatelli, R. H. Chan, et al., "Nonstationary Iterated Thresholding Algorithms for Image Deblurring," *Inverse Problems and Imaging* 7, no. 3 (2013): 717–736.

10991506, 2026, 3, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nla.70093 by Alessandro Buccini - Bibliothèque de Sorbonne Université , Wiley Online Library on [27/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License