Numerical Linear Algebra with Applications

WILEY

RESEARCH ARTICLE OPEN ACCESS

Krylov Subspace Based FISTA-Type Methods for Linear Discrete Ill-Posed Problems

Alessandro Buccini[1] | Fei Chen[2] | Mirjeta Pasha[3] | Lothar Reichel[4]

[1]Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy | [2]School of Mathematics, Trinity College Dublin, Dublin, Ireland | [3]Department of Mathematics, Virgina Tech, Blacksburg, Virginia, USA | [4]Department of Mathematical Sciences, Kent State University, Kent, Ohio, USA

Correspondence: Alessandro Buccini (alessandro.buccini@unica.it)

Received: 22 January 2024 | **Revised:** 21 November 2024 | **Accepted:** 29 November 2024

Funding: This research was supported by the GNCS group of INdAM, of which Alessandro Buccini is a member, with the INdAM-GNCS 2024 Project "Algebra lineare numerica per problemi di grandi dimensioni: aspetti teorici e applicazioni" (CUP_E53C23001670001) and with the INdAM-GNCS 2023 Project "Tecniche numeriche per lo studio dei problemi inversi e l'analisi delle reti complesse" (CUP_E53C22001930001). Moreover, Alessandro Buccini is partially supported by the PRIN 2022 PNRR project "A mathematical approach to inverse problems arising in cultural heritage preservation and dissemination" (P2022PMEN2) and by the PRIN 2022 project "Inverse Problems in the Imaging Sciences (IPIS)" (2022ANC8HL), both financed by the European Union - NextGenerationEU and by the Italian Ministry of University and Research (MUR). Alessandro Buccini is also partially supported by Fondazione di Sardegna, Progetto biennale bando 2021, "Computational Methods and Networks in Civil Engineering (COMANCHE)". Mirjeta Pasha acknowledges support from NSF Award No. 2202846 and partial support from NSF DMS No. 2410699.

Keywords: image reconstruction | Krylov subspace | projected FISTA

ABSTRACT

Several iterative soft-thresholding algorithms, such as FISTA, have been proposed in the literature for solving regularized linear discrete inverse problems that arise in various applications in science and engineering. These algorithms are easy to implement, but their rates of convergence may be slow. This paper describes novel approaches to reduce the computations required for each iteration by using Krylov subspace techniques. Specifically, we propose to impose sparsity on the coefficients in the representation of the computed solution in terms of a Krylov subspace basis. Several numerical examples from image deblurring and computerized tomography are used to illustrate the efficiency and accuracy of the proposed methods.

1 | Introduction

We are concerned with the solution of large-scale minimization problems

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \tag{1}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$ is a large matrix, whose singular values decay to zero with increasing index number without a significant gap between the smallest nonvanishing singular values. This implies that the matrix $\mathbf{A}$ is severely ill-conditioned and may be rank-deficient. Minimization problems (1) with a matrix of this kind are commonly referred to as discrete ill-posed problems, and they arise, for instance, from the discretization of linear ill-posed problems such as Fredholm integral equations of the first kind; see, for example [1, 2]. Throughout this paper $\|\cdot\|_2$ denotes the Euclidean vector norm.

In many applications in science and engineering, the vector $\mathbf{b}\in\mathbb{R}^m$ in (1) represents data that is contaminated by an error $\mathbf{e}\in\mathbb{R}^m$. The data vector can be expressed as

$$\mathbf{b}=\mathbf{b}_{\text{true}}+\mathbf{e} \tag{2}$$

where $\mathbf{b}_{\text{true}}$ denotes the unknown error-free vector associated with $\mathbf{b}$. We will assume that $\mathbf{b}_{\text{true}}$ is in the range of $\mathbf{A}$, and would

like to determine the solution of minimal least-squares norm of

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}_{\text{true}}\|_2$$

We denote this solution by $\mathbf{x}_{\text{true}}$.

Let $\mathbf{A}^\dagger$ denote the Moore-Penrose pseudoinverse of $\mathbf{A}$. Then $\mathbf{x}_{\text{true}} = \mathbf{A}^\dagger\mathbf{b}_{\text{true}}$. However, due to the severe ill-conditioning of $\mathbf{A}$, the error $\mathbf{e}$ in $\mathbf{b}$ is amplified when evaluating $\mathbf{A}^\dagger\mathbf{b}$, that is, the solution of (1). It follows that a straightforward solution of (1) generally will not furnish a useful approximation of $\mathbf{x}_{\text{true}}$.

A common approach to compute a better approximation of $\mathbf{x}_{\text{true}}$ than $\mathbf{A}^\dagger\mathbf{b}$ is to replace the least-squares problem (1) by a nearby problem, whose solution is less sensitive to the error $\mathbf{e}$ in $\mathbf{b}$. This replacement is often referred to as *regularization*. In recent years, the replacement of (1) by a minimization problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n}\{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2+\mu\|\mathbf{x}\|_1\} \tag{3}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm, has received considerable attention; see, for example [3–8]. The parameter $\mu>0$ is known as the *regularization parameter*. It balances the influence of the first term (*the fidelity term*) and the second term (*the penalty term*) in (3). The use of the ℓ_1-norm in the penalty term promotes sparsity of the computed solution, that is, many entries in the computed solution may vanish. This is advantageous when the desired solution $\mathbf{x}_{\text{true}}$ is known to have many zero entries; see, for example, [9–11], for examples. In particular, sparsity-promoting methods are useful for image restoration when the image is represented by framelets, as it is well known that natural images have a sparse representation in the framelet domain; see, for example, [5], for illustrations.

Solution methods for (3) include fast iterative soft-thresholding algorithm (FISTA), first described by Beck and Teboulle [3], and iterative Bregman-type methods, which use the Bregman distance; see [4, 6–8, 12]. The latter methods also apply soft-thresholding. This paper focuses on FISTA-type methods and discusses approaches based on Krylov subspace techniques for reducing the computational effort required by these methods.

The main computational issues of FISTA-type methods, especially when applied to large-scale problems, are the need to carry out many iterations and that each iteration can be expensive in terms of computer time. This work explores how Krylov subspace techniques can be applied to speed up FISTA-type methods. Properties of the FISTA method described by Beck and Teboulle [3] carry over to the projected method discussed in this work. Our aim is to describe and compare several approaches to project the minimization problem (3) into Krylov subspaces to speed up the computations.

We will assume that a fairly accurate bound δ for the norm of the error $\mathbf{e}$ in $\mathbf{b}$,

$$\|\mathbf{e}\|_2\le\delta \tag{4}$$

is known. This bound will be helpful for determining the number of iterations to be carried out with FISTA-type methods. We remark that if such a bound is not available, an estimate often can be determined directly by error-estimation techniques such as those described in [13–15], or indirectly by generalized cross validation, the L-curve criterion [16–18], or related methods such as [19, 20].

Introduce the Krylov subspace

$$\begin{aligned}&\mathcal{K}_d(\mathbf{A}^T\mathbf{A},\mathbf{A}^T\mathbf{b})\\&\quad=\text{span}\{\mathbf{A}^T\mathbf{b},\mathbf{A}^T\mathbf{A}\mathbf{A}^T\mathbf{b},(\mathbf{A}^T\mathbf{A})^2\mathbf{A}^T\mathbf{b},\ldots,(\mathbf{A}^T\mathbf{A})^{d-1}\mathbf{A}^T\mathbf{b}\}\end{aligned} \tag{5}$$

where the superscript ^{T} denotes transposition, and assume that it is of dimension d. An orthonormal basis for this subspace can be generated by carrying out d steps of Golub-Kahan bidiagonalization applied to $\mathbf{A}$ with initial vector $\mathbf{b}$; see Section 3 for details. The dimension d is chosen large enough so that the subspace (5) contains a vector $\mathbf{x}^*$ that satisfies the discrepancy principle (DP),

$$\|\mathbf{A}\mathbf{x}^*-\mathbf{b}\|_2\le\tau\delta \tag{6}$$

where δ satisfies (4) and $\tau>1$ is a user-supplied constant that is independent of δ. Generally, $d\ll\min\{m,n\}$. It is shown in a Hilbert space setting by, for example, Engl et al. [1] that if $\|\mathbf{A}\mathbf{x}^*-\mathbf{b}\|_2\approx\tau\delta$, then $\mathbf{x}^*$ converges to $\mathbf{x}_{\text{true}}$ as the bound δ in (4) converges to zero; however, the rate of convergence may be slow. We are interested in the finite-dimensional situation.

This paper describes two approaches to applying Krylov subspace techniques in FISTA-type methods. In the KFISTAF method, we replace the matrix $\mathbf{A}$ by an approximation of rank d furnished by partial Golub-Kahan bidiagonalization of $\mathbf{A}$. The computed iterates are mapped from $\mathbb{R}^n$ to the Krylov subspace (5) and back in each iteration. The savings in computing time stem from the fact that a matrix-vector product evaluation with the projection of $\mathbf{A}$ in the Krylov subspace (5) is cheaper than a matrix-vector product evaluation with the matrix $\mathbf{A}$. Similar considerations apply to the evaluation of $\mathbf{A}^T$.

A significant computational cost of the KFISTAF method is the mapping of iterates from $\mathbb{R}^n$ to the Krylov subspace (5) and back in each iteration. These mappings are needed to apply soft-thresholding in $\mathbb{R}^n$. We also describe a Krylov subspace-based solution method that applies thresholding of the coefficient vectors of dimension d that represent the computed solutions in (5). This approach obviates the need to map iterates from $\mathbb{R}^n$ to (5) and back in each iteration, and this reduces the computing time required for each iteration. We will refer to this scheme as KFISTA. The minimization problem solved by KFISTA is different from the problem (3) solved by FISTA and KFISTAF; see (14) below for details. Nevertheless, numerous computed examples, a few of which are reported in Section 4 shows KFISTA to give computed approximate solutions of comparable quality as FISTA and KFISTAF. We also illustrate in Section 4 that KFISTA yields computed solutions of comparable quality as FISTA, but at a lower computational cost. Note that KFISTA can be combined with hard thresholding of the Krylov subspace coefficients; however, we will not dwell on this in the present work.

In most regularization methods, the determination of a suitable value of the regularization parameter $\mu>0$ is both important and

challenging. A too small value of μ results in a poor approximation of $\mathbf{x}_{\text{true}}$ due to an unnecessarily large propagated error stemming from the error $\mathbf{e}$ in $\mathbf{b}$, while a too large value μ gives a too smooth approximation that does not show details of $\mathbf{x}_{\text{true}}$ that a smaller value would reveal. This paper proposes nonstationary versions of both FISTA and KFISTA to avoid the estimation of μ. Following the idea proposed for iterated Tikhonov regularization in [21], we use a sequence of parameters $\mu_k \searrow 0$ instead of a fixed parameter. We terminate the iterations with the aid of the DP (6), that is, we stop the iterations as soon as $\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\| \leq \tau\delta$, where $\mathbf{x}^{(k)}$ denotes the kth iterate of either nonstationary FISTA (NFISTA) or nonstationary KFISTA (NKFISTA). Moreover, denoting by k_{DP} the stopping iteration determined by the DP applied to a reduced problem resulting from a projection into a Krylov subspace, we propose to use $\mu_{\text{DP}} := \mu_{k_{\text{DP}}}$ as a regularization parameter in FISTA and KFISTA; See Section 4 for details. Extensive numerical examples have shown that this approach is able to provide values of μ that are close to the optimal one, that is, the one that minimizes the reconstruction error (defined by (21) below), with little additional computational effort, especially for KFISTA.

We remark that the minimization problem (3) is convex and, typically, strictly convex. It has a unique solution in the latter case. The algorithms discussed minimize (3) with the solution restricted to a Krylov subspace of a suitable dimension. This minimization problem is also convex and, generally, strictly convex. Hence, a unique solution exists for the latter case too. It follows from the properties of FISTA that this solution is obtained when the iterations are continued indefinitely. However, this solution is of no interest to us because it is typically contaminated by a significant propagated error that stems from the error in the data vector b.

Most of the computed examples presented in Section 4 are concerned with the restoration of grayscale images that have been contaminated by blur and noise. An image is a collection of nonnegative pixels. Therefore, it is meaningful to require the desired image to have nonnegative pixels. We achieve this by setting all negative pixels in the reconstructions obtained by KFISTA to zero. The algorithm so obtained is referred to as post-nonnegative projection KFISTA (PPKFISTA). PPKFISTA first runs NKFISTA with the DP as a stopping criterion to determine μ_{DP}. Next, we run KFISTA with μ_{DP} as a fixed regularization parameter, and finally, we set all negative entries of the computed solution to zero. PPKFISTA is our algorithm of choice for the image restoration problems considered. The main computational burden is the determination of an orthonormal basis for the Krylov subspace. This basis is already computed when applying NKFISTA; running KFISTA afterward merely increases the computational time by a fraction of a second. The reason we apply the projection to the solution determined by KFISTA instead of to the solution computed by NKFISTA is that the latter method may terminate prematurely, and this results in over-smoothed solutions. This is illustrated in Tables 3 and 5 of Section 4.

The main attraction of PPKFISTA is that the projection onto the nonnegative orthant is simple to implement and fast to execute. A possible drawback of PPKFISTA is that the solution produced is not guaranteed to be an accurate approximate solution of the minimization problem (3). However, this typically is not a major concern since the choice of penalty term is somewhat arbitrary. If the computed nonnegative solution must be an accurate approximate solution of (3), then the method of the present paper can be modified to project the computed approximate solution into the first orthant in each iteration. An analogous extension of Bregman iteration is described in [5, Section 3.2]. Such an extension generally gives a slightly more accurate approximate solution of (3) than PPKFISTA, but generally requires more matrix-vector product evaluations with $\mathbf{A}$ and $\mathbf{A}^T$.

This paper is organized as follows: Section 2 briefly discusses the minimization problem (3). We describe our Krylov subspace-based FISTA-type methods in Section 3 and discuss their convergence. The application of Krylov subspaces to accelerate FISTA may be considered preconditioning in the sense that it reduces the number of matrix-vector product evaluations required with the matrices $\mathbf{A}$ and $\mathbf{A}^T$. Section 4 presents computed examples that illustrate the performance of the FISTA-type methods discussed. Concluding remarks can be found in Section 5.

It is with great sadness that we mourn the passing of Owe Axelsson, a remarkable researcher in the field of Scientific Computing and, in particular, in Numerical Linear Algebra. His profound contributions left an indelible mark, evident by his books [22, 23] and many papers, among them [24–26], as well as in the accomplishments of his numerous Ph.D. students. In addition, Axelsson launched the journal Numerical Linear Algebra with Applications. The last author was first exposed to Numerical Linear Algebra as an undergraduate student by lecture notes by Axelsson on iterative methods used at the Department of Information Processing at the University of Lund in the mid '70s. These lecture notes were subsequently expanded to the book [22].

2 | FISTA-Type Methods

FISTA-type methods were introduced by Beck and Teboulle [3] to compute the solution of convex optimization problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n}\{\mathcal{J}(\mathbf{x}) : \mathcal{J}(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\} \tag{7}$$

with $f : \mathbb{R}^n \to \mathbb{R}$ a smooth convex function of the type $C^{1,1}$, that is,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq l_f\|\mathbf{x} - \mathbf{y}\|_2, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

where $l_f > 0$ is a Lipschitz constant of the gradient ∇f, and $g : \mathbb{R}^n \to \mathbb{R}$ is a continuous, possibly non-smooth, convex function. The general idea of FISTA is that after having determined the point $\mathbf{x}^{(k)} \in \mathbb{R}^n$ in iteration k, an additional point $\mathbf{u}^{(k)} \in \mathbb{R}^n$ is chosen as a linear combination of $\mathbf{x}^{(k)}$ and the previous point $\mathbf{x}^{(k-1)} \in \mathbb{R}^n$. Then

$$\mathbf{x}^{(k+1)} = T_{l_f}(\mathbf{u}^{(k)}) := \arg\min_{\mathbf{x}\in\mathbb{R}^n}\left\{ Q_{l_f}(\mathbf{x}, \mathbf{u}^{(k)}) + g(\mathbf{x}) \right\}$$

that is, $\mathbf{x}^{(k+1)}$ is the unique minimizer of $Q_{l_f}(\mathbf{x}, \mathbf{u}) + g(\mathbf{x})$ at $\mathbf{u} = \mathbf{u}^{(k)}$, where

$$\mathbf{x} \to Q_{l_f}(\mathbf{x}, \mathbf{u}) := f(\mathbf{u}) + \langle \mathbf{x} - \mathbf{u}, \nabla f(\mathbf{u})\rangle + \frac{l_f}{2}\|\mathbf{x} - \mathbf{u}\|_2^2$$

is a quadratic upper bound of $f(\mathbf{x})$. Since the term $f(\mathbf{u}^{(k)})$ in $T_{l_f}(\mathbf{u}^{(k)})$ is independent of $\mathbf{x}$, we have

$$\begin{aligned}\mathbf{x}^{(k+1)} &= T_{l_f}(\mathbf{u}^{(k)}) \\ &= \arg\min_{\mathbf{x}\in\mathbb{R}^n}\left\{ Q_{l_f}(\mathbf{x}, \mathbf{u}^{(k)}) + g(\mathbf{x}) \right\} \\ &= \arg\min_{\mathbf{x}\in\mathbb{R}^n}\left\{ \langle \mathbf{x}-\mathbf{u}^{(k)}, \nabla f(\mathbf{u}^{(k)})\rangle + \frac{l_f}{2}\|\mathbf{x}-\mathbf{u}^{(k)}\|_2^2 + g(\mathbf{x}) \right\} \\ &= \arg\min_{\mathbf{x}\in\mathbb{R}^n}\left\{ \frac{l_f}{2}\left\|\mathbf{x}-(\mathbf{u}^{(k)}-\frac{1}{l_f}\nabla f(\mathbf{u}^{(k)}))\right\|_2^2 + g(\mathbf{x}) \right\}\end{aligned}$$

where in the third equality we have ignored the term $f\left(\mathbf{u}^{(k)}\right)$ as it is independent of $\mathbf{x}$. We note that FISTA implements Nesterov acceleration.

2.1 | Stationary FISTA

FISTA uses the two most recently computed iterates for evaluating the next iterate. We consider the following functions

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2, \qquad g(\mathbf{x}) = \mu\|\mathbf{x}\|_1 \tag{8}$$

in (7), which are of interest when minimizing (3).

Let $\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ denote the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. Then $l_f = 2\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ is the smallest Lipschitz constant of ∇f, where f is defined in (8). The quotient

$$L_f = \frac{l_f}{s}$$

serves as step size in FISTA, where $0 < s \le 1$. Beck and Teboulle [3] show that with constant step size $L_f = l_f$, FISTA achieves a convergence rate of $\mathcal{O}(1/k^2)$, where k denotes the number of iterations performed.

The operator $T_{L_f}(\mathbf{u})$ is a proximal regularization operator for the non-smooth ℓ_1-regularized problem (3). Since the $\|\mathbf{x}\|_1$-term in (3) is separable, we obtain

$$\mathbf{x}^{(k+1)} = T_{L_f}(\mathbf{u}^{(k)}) = \Phi_{s\mu/l_f}\left(\mathbf{u}^{(k)} - \frac{s}{l_f}\nabla f(\mathbf{u}^{(k)})\right)$$

where $\Phi_\alpha(\mathbf{x})$, for $\alpha > 0$, denotes the soft-thresholding operator, that is,

$$\Phi_\alpha(\mathbf{x}) = [\phi_\alpha(x_1), \phi_\alpha(x_2), \ldots, \phi_\alpha(x_n)]^T \in \mathbb{R}^n$$

with $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and

$$\phi_\alpha(x) := \begin{cases} 0 & \text{if } |x| \le \alpha \\ \text{sign}(x)(|x|-\alpha) & \text{if } |x| > \alpha \end{cases}$$

see [3] for details.

ALGORITHM 1 | Fast Iterative Soft-Thresholding Algorithm (FISTA).

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mu>0$, $l_f = 2\lambda_{\max}(\mathbf{A}^T\mathbf{A})$, $s \in (0,1]$, $t_0 = t_1 = 1$, and $tol>0$.
$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} = \mathbf{A}^T\mathbf{b}$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} + \dfrac{t_{k-1}-1}{t_k}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$
  $\mathbf{u}^{(k)} = \mathbf{z}^{(k)} - \dfrac{2s}{l_f}\mathbf{A}^T(\mathbf{A}\mathbf{z}^{(k)} - \mathbf{b})$
  $\mathbf{x}^{(k+1)} = \Phi_{s\mu/l_f}(\mathbf{u}^{(k)})$
  if $\dfrac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \le tol$ **then**
    exit
  end if
  $t_{k+1} = \dfrac{1+\sqrt{1+4t_k^2}}{2}$
end for
Output: Approximate solution $\mathbf{x}^{(k+1)}$.

For the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ given by (8), the iterations with FISTA, with $\mathbf{x}^{(0)} = \mathbf{x}^{(1)} = 0$ and $t_0 = t_1 = 1$, can be written as

$$\begin{cases} \mathbf{z}^{(k)} = \mathbf{x}^{(k)} + \left(\frac{t_{k-1}-1}{t_k}\right)(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}), \\ \mathbf{u}^{(k)} = \mathbf{z}^{(k)} - \frac{2s}{l_f}\mathbf{A}^T(\mathbf{A}\mathbf{z}^{(k)} - \mathbf{b}), \\ \mathbf{x}^{(k+1)} = \Phi_{s\mu/l_f}(\mathbf{u}^{(k)}), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \end{cases} \qquad k = 1, 2, \ldots$$

see [3]. The computations are described by Algorithm 1. It is specified in Section 4 how we choose the parameters and the stopping criteria for the algorithms tested in this paper.

It is worth mentioning that the regularization parameter μ in FISTA is not always easy to determine; see, for example, [4], for a related discussion on Bregman iteration. One approach to find a suitable value of μ is to use a nonstationary algorithm. We describe in the next subsection a nonstationary version of FISTA (NFISTA) that automatically selects a suitable value of μ. NFISTA also depends on some other parameters, but in our experience they are not difficult to determine.

The computational cost of FISTA is dominated by the evaluation of matrix-vector products with the matrices $\mathbf{A}$ and $\mathbf{A}^T$. In Section 3 we therefore approximate $\mathbf{A}$ and $\mathbf{A}^T$ by matrices that allow less expensive evaluation of matrix-vector products.

2.2 | Nonstationary FISTA

The quality[1] of the computed solution by a regularization method depends on the choice of the regularization parameter μ. A number of rules have been discussed in the literature for choosing a suitable value of μ; see, for example, [1, 2, 18]. In this paper, we

use nonstationary iterative methods, for which the regularization parameter is not fixed, but is changed during the iterations. For nonstationary iterative Tikhonov methods, a common choice of the sequence of regularization parameters $\mu = \mu_k$ is

$$\mu_k = \mu_0 q^{k-1}, \qquad k = 1, 2, \ldots \tag{9}$$

for some suitable values of $\mu_0 > 0$ and $0 < q < 1$; see, for example, [21, 27, 28]. We will employ the same sequence for nonstationary FISTA, that is, at iteration k, the regularization parameter is defined by (9).

The value of μ_k depends on the two parameters μ_0 and q. In our numerical experiments reported in Section 4, we let $\mu_0 = 10$. Hence, the iterations will start with a fairly large value of the regularization parameter. This parameter should be chosen large enough so that the discrepancy principle is not satisfied right away. The exact value of μ_0 is not critical for the performance of the methods discussed. We let $q = 0.99$ so that the value of μ_k is reduced slowly as k increases. This choice generally results in that many iterations in the Krylov subspace have to be carried out. Since each one of these iterations is very cheap, the exact value of q is not important. We found these choices of μ_0 and q to perform well for a large number of restoration problems. We start iteration k by computing the regularization parameter.

Beck and Teboulle [3] discuss different choices of the sequence of t_k, $k = 0, 1, \ldots$, in Algorithm 1, which correspond to using constant step size or backtracking. In this paper we only consider constant step size. A reason why the authors of [3] propose a backtracking technique is because the constant l_f required by Algorithm 1 is expensive to compute for large-scale problems, for example, when the matrix $\mathbf{A}^T\mathbf{A}$ is large. Our solution methods for large-scale problems overcome this difficulty since the matrix $\mathbf{A}^T\mathbf{A}$ is projected to a small matrix by a Krylov subspace method; the largest eigenvalue of the projected matrix easily can be computed; see Section 3 for more details.

Let $\mu > 0$ be fairly large. Then the error $\|\mathbf{x}_\mu - \mathbf{x}_{\text{true}}\|_2$ in the solution $\mathbf{x}_\mu$ of the minimization problem (3) first decreases as μ decreases but subsequently increases when μ decreases further. This behavior is commonly referred to as semiconvergence. Since $\mu_k \searrow 0$ as $k \to \infty$, we expect NFISTA to be semiconvergent, that is, the approximation error $\|\mathbf{x}^{(k)} - \mathbf{x}_{\text{true}}\|_2$ decreases during the first few iterations, but subsequently increases with k. In particular, $\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\|_2^2 + \mu_k\|\mathbf{x}^{(k)}\|_1$ being small does not guarantee that $\mathbf{x}^{(k)}$ is close to $\mathbf{x}_{\text{true}}$. We therefore terminate the computations as soon as an approximate solution $\mathbf{x}^{(k)}$ has been found that satisfies the DP (6), that is, as soon as

$$\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\|_2 \le \tau\delta$$

holds. The left-hand side is inexpensive to evaluate; see below.

3 | Krylov Subspace Based FISTA

This section first introduces a projected FISTA method for the solution of (3) and then discusses some variations.

3.1 | Projection Into a Krylov Subspace

As already mentioned, a difficulty when applying FISTA is that the algorithm requires the parameter $l_f = 2\lambda_{\max}(\mathbf{A}^T\mathbf{A})$, which may be expensive to compute when the matrix $\mathbf{A}$ is large. Another difficulty is, as we will illustrate with numerical experiments, that it is typically hard to choose a suitable stopping criterion. To circumvent these difficulties, we develop Krylov subspace-based FISTA methods. We generate a low-dimensional Krylov subspace (5) with $d = \ell$ by applying ℓ steps of Golub-Kahan bidiagonalization to the matrix $\mathbf{A}$ with starting vector $\mathbf{b}$; see [29]. This gives the decomposition

$$\mathbf{A}\mathbf{V}_\ell = \mathbf{U}_{\ell+1}\mathbf{B}_{\ell+1,\ell} \tag{10}$$

where the matrices $\mathbf{U}_{\ell+1} \in \mathbb{R}^{m\times(\ell+1)}$ and $\mathbf{V}_\ell \in \mathbb{R}^{n\times\ell}$ have orthonormal columns, and the first columns of $\mathbf{U}_{\ell+1}$ and $\mathbf{V}_\ell$ are $\mathbf{b}/\|\mathbf{b}\|_2$ and $\mathbf{A}^T\mathbf{b}/\|\mathbf{A}^T\mathbf{b}\|_2$, respectively. The matrix $\mathbf{B}_{\ell+1,\ell} \in \mathbb{R}^{(\ell+1)\times\ell}$ is lower bidiagonal. The columns of $\mathbf{V}_\ell$ span the Krylov subspace $\mathcal{K}_\ell(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$, cf. (5), which generically is of dimension ℓ. We will assume this to be the case; otherwise, the computations simplify.

An approximate solution of (1) in $\mathcal{K}_\ell(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$ can easily be computed as follows. Let $\mathbf{x} = \mathbf{V}_\ell\mathbf{y}$. Then, using (10), we obtain

$$\min_{\mathbf{x}\in\mathcal{K}_\ell(\mathbf{A}^T\mathbf{A},\mathbf{A}^T\mathbf{b})} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \min_{\mathbf{y}\in\mathbb{R}^\ell} \|\mathbf{A}\mathbf{V}_\ell\mathbf{y} - \mathbf{b}\|_2^2 \tag{11}$$

$$\begin{aligned} &= \min_{\mathbf{y}\in\mathbb{R}^\ell} \|\mathbf{U}_{\ell+1}\mathbf{B}_{\ell+1,\ell}\mathbf{y} - \mathbf{b}\|_2^2 \\ &= \min_{\mathbf{y}\in\mathbb{R}^\ell} \|\mathbf{B}_{\ell+1,\ell}\mathbf{y} - \beta\mathbf{e}_1\|_2^2 \end{aligned} \tag{12}$$

where $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{\ell+1}$ and $\beta = \|\mathbf{b}\|_2$. The last equality follows from the fact that $\mathbf{U}_{\ell+1}^T\mathbf{b} = \beta\mathbf{e}_1$. Since the subspace we minimize over increases in dimension with ℓ and $\mathcal{K}_{\ell-1}(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b}) \subseteq \mathcal{K}_\ell(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$, it follows that (12) decreases (generally strictly) as ℓ increases.

Let $\mathbf{y}_\ell$ denote the solution of (12). Then $\mathbf{x}_\ell = \mathbf{V}_\ell\mathbf{y}_\ell$ solves the left-hand side of (11). Let d denote the smallest integer ℓ, such that

$$\|\mathbf{B}_{\ell+1,\ell}\mathbf{y}_d - \beta\mathbf{e}_1\|_2 \le \tau\delta$$

Then

$$\|\mathbf{A}\mathbf{x}_d - \mathbf{b}\|_2 \le \tau\delta \tag{13}$$

cf. (6). Typically, $d \ll \min\{m, n\}$.

3.2 | The KFISTAF Method

Instead of applying FISTA to solve the minimization problem (3), we apply FISTA to the problem

$$\begin{aligned} &\min_{\mathbf{y}\in\mathbb{R}^d} \{\|\mathbf{B}_{d+1,d}\mathbf{V}_d^T\mathbf{V}_d\mathbf{y} - \beta\mathbf{e}_1\|_2^2 + \mu\|\mathbf{V}_d\mathbf{y}\|_1\} \\ &\quad = \min_{\mathbf{x}\in\mathcal{K}_d(\mathbf{A}^T\mathbf{A},\mathbf{A}^T\mathbf{b})} \{\|\mathbf{B}_{d+1,d}\mathbf{V}_d^T\mathbf{x} - \beta\mathbf{e}_1\|_2^2 + \mu\|\mathbf{x}\|_1\} \end{aligned} \tag{14}$$

where we have used that $\mathbf{V}_d^T\mathbf{V}_d = \mathbf{I}$ and $\mathbf{x} = \mathbf{V}_d\mathbf{y}$. This obviates the need to evaluate matrix-vector products with the matrices $\mathbf{A}$

and $\mathbf{A}^T$ during the execution of FISTA. The application of FISTA to the minimization problem (14) can be expressed as

$$\begin{cases} \mathbf{z}^{(k)} = \mathbf{x}^{(k)} + (\frac{t_{k-1}-1}{t_k})(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}), \\ \mathbf{u}^{(k)} = \mathbf{z}^{(k)} - \frac{2s}{l_f}\mathbf{V}_d\mathbf{B}_{d+1,d}^T(\mathbf{B}_{d+1,d}\mathbf{V}_d^T\mathbf{z}^{(k)} - \beta\mathbf{e}_1), \\ \mathbf{x}^{(k+1)} = \Phi_{s\mu/l_f}(\mathbf{u}^{(k+1)}), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \end{cases} \quad k = 1, 2, \ldots \tag{15}$$

where $l_f = 2\lambda_{\max}(\mathbf{B}_{d+1,d}^T\mathbf{B}_{d+1,d})$. Since the matrix $\mathbf{B}_{d+1,d}$ typically is small, we can easily compute its largest singular value to determine l_f. We refer to FISTA described by (15) as KFISTAF, since it is Krylov subspace-based and soft-thresholds "full-sized" vectors $\mathbf{u}^{(k+1)} \in \mathbb{R}^n$.

3.3 | The KFISTA Method

The most time-consuming part in each iteration with KFISTAF is the mapping of vectors from $\mathbb{R}^n$ to $\mathcal{K}_d(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$ and back when the matrix $\mathbf{V}_d \in \mathbb{R}^{n\times d}$ has many rows. We therefore propose a new model to increase the efficiency, where one soft-thresholds the much smaller vector $\mathbf{y} \in \mathbb{R}^d$:

$$\min_{y\in\mathbb{R}^d}\{\|\mathbf{B}_{d+1,d}\mathbf{y} - \beta\mathbf{e}_1\|_2^2 + \mu\|\mathbf{y}\|_1\} \tag{16}$$

FISTA applied to the solution of (16) can be written as

$$\begin{cases} \mathbf{z}^{(k)} = \mathbf{y}^{(k)} + (\frac{t_{k-1}-1}{t_k})(\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}), \\ \mathbf{u}^{(k)} = \mathbf{z}^{(k)} - \frac{2s}{l_f}\mathbf{B}_{d+1,d}^T(\mathbf{B}_{d+1,d}\mathbf{z}^{(k)} - \beta\mathbf{e}_1), \\ \mathbf{y}^{(k+1)} = \Phi_{s\mu/l_f}(\mathbf{u}^{(k+1)}), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \end{cases} \quad k = 1, 2, \ldots \tag{17}$$

where $l_f = 2\lambda_{\max}(\mathbf{B}_{d+1,d}^T\mathbf{B}_{d+1,d})$. Let FISTA applied to (17) carry out K steps. Then $\mathbf{x} = \mathbf{V}_d\mathbf{y}^{(K)}$ is an approximate solution of (1). We call the algorithm determined by the iterations (17) Krylov-FISTA (KFISTA). Since KFISTA promotes sparsity of the Krylov subspace coefficients, but not of the computed solution, the computed approximate solution $\mathbf{x} = \mathbf{V}_d\mathbf{y}^{(K)}$ of (1) is usually not sparse.

We remark that the columns of the matrix $\mathbf{V}_d = [v_1, v_2, \ldots, v_d]$ can be considered discretizations of piecewise smooth functions of an increasing variable. These functions typically oscillate more with increasing index. The computed solution is a linear combination of the vectors $v_1, v_2, \ldots, v_d$. Setting coefficients of columns with large indexes in this linear combination to zero generally reduces the oscillations in the computed solution. This is often desirable when the desired solution is known to be piecewise smooth and thus justifies the shrinkage of the coefficient vectors.

3.4 | The PPKFISTA Method

We will apply all the algorithms mentioned to image restoration. Then the entries of the computed solutions represent pixel values of the restored image. As already mentioned, pixel values are nonnegative. We therefore project the solutions computed with all methods in our comparison to the nonnegative orthant. For instance, for the approximate solution determined by K steps of KFISTA, we let

$$\mathbf{x}_+ = \max\left\{\mathbf{V}_d\mathbf{y}^{(K)}, 0\right\} \tag{18}$$

where the max operator is applied element-wise.

Numerical experiments reported in Section 4 show that the quality of the restored images determined by KFISTA in combination with orthogonal projection (18) (PPKFISTA) are comparable to those determined by KFISTAF with orthogonal projection (PPKFISTAF), but PPKFISTA requires less computing time per iteration, and it is easier to determine a suitable number of iterations for this algorithm; see details in Section 4.

Let $d_2 > d_1$. If d_1 and d_2 are small enough, then $\mathcal{K}_{d_1}(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b}) \subsetneq \mathcal{K}_{d_2}(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$. This holds in all examples reported in Section 4. It is natural to expect more accurate restorations, the more Golub-Kahan bidiagonalization steps are carried out since the matrix $\mathbf{U}_{d_2+1}\mathbf{B}_{d_2+1,d_2}\mathbf{V}_{d_2}^T$ usually furnishes a more accurate approximation of $\mathbf{A}$ than the matrix $\mathbf{U}_{d_1+1}\mathbf{B}_{d_1+1,d_1}\mathbf{V}_{d_1}^T$. However, increasing the dimension of the Krylov subspace makes the execution of KFISTAF and KFISTA slower. Numerical experiments suggest that it is often beneficial with respect to the quality of the computed restoration to carry out $d + 3$ steps of Golub-Kahan bidiagonalization, where d is the minimal number of steps such that (13) holds. We will use this choice of number of steps for all FISTA-type algorithms discussed.

3.5 | The NKFISTA Method

Nonstationary FISTA restricted to a Krylov subspace (NKFISTA) is described by Algorithm 2. We obtain the stationary version, KFISTA, by setting $q = 1$. Then the regularization parameter μ_k does not change during the iterations. We obtain the stationary version, KFISTA, by setting $q = 1$. In this version the regularization parameter μ_k does not change during the iterations.

Since the dimension of the minimization problem (16) is much smaller than the dimension of the problem (3), the dominant computational effort of KFISTA is the construction of the Krylov subspace; see Table 4 for illustrations. This situation is also observed for the Krylov APLB method described below.

3.6 | The KAPLB Method

Inspired by the efficiency of KFISTA, we also modify the accelerated projected linearized Bregman (APLB) method proposed in [5] so that the modified method soft-thresholds Krylov subspace coefficients. We call the method so obtained the Krylov APLB (KAPLB) method. It is described by Algorithm 3. We remark that the APLB method was inspired by the accelerated Bregman method described by Huang et al. [30].

There are two main differences between FISTA-type and APLB-type methods: Firstly, FISTA-type methods use in each iteration a linear combination of two previous iterates that are both soft-thresholded to calculate the next iterate, while

ALGORITHM 2 | Nonstationary FISTA on a Krylov Subspace (NKFISTA).

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\delta \geq \|\mathbf{e}\|_2$, $\beta = \|\mathbf{b}\|_2$, $\mu_0>0$, $0<q<1$, $\tau>1$, $s \in (0, 1]$, $t_0 = t_1 = 1$, and $tol>0$.
for $k = 1, 2, \ldots$ **do**
  Compute the decompositions (10) $\mathbf{AV}_k = \mathbf{U}_{k+1}\mathbf{B}_{k+1,k}$.
  $\mathbf{y}_k^* = \beta\mathbf{B}_{k+1,k}^{\dagger}\mathbf{e}_1$, where $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{k+1}$
  if $\|\mathbf{B}_{k+1,k}\mathbf{y}_k^* - \beta\mathbf{e}_1\|_2 \leq \tau\delta$ **then**
    $d = k + 3$
  end if
end for
Compute the decompositions (10) $\mathbf{AV}_d = \mathbf{U}_{d+1}\mathbf{B}_{d+1,d}$.
$l_f = 2\lambda_{\max}(\mathbf{B}_{d+1,d}^T\mathbf{B}_{d+1,d})$
$\mathbf{x}^{(0)} = \mathbf{A}^T\mathbf{b}$
$\mathbf{y}^{(0)} = \mathbf{y}^{(1)} = \mathbf{V}_d^T\mathbf{x}^{(0)}$
for $k = 1, 2, \ldots$ **do**
  $\mu_k = \mu_0 q^{k-1}$
  $\mathbf{z}^{(k)} = \mathbf{y}^{(k)} + \dfrac{t_{k-1} - 1}{t_k}(\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)})$
  $\mathbf{u}^{(k)} = \mathbf{z}^{(k)} - \dfrac{2s}{l_f}\mathbf{B}_{d+1,d}^T(\mathbf{B}_{d+1,d}\mathbf{z}^{(k)} - \beta\mathbf{e}_1)$
  $\mathbf{y}^{(k+1)} = \Phi_{\mu_k/L_f}(\mathbf{u}^{(k)})$
  if $\dfrac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\|_2}{\|\mathbf{y}^{(k)}\|_2} \leq tol$ **then**
    exit
  end if
  $t_{k+1} = \dfrac{1 + \sqrt{1 + 4t_k^2}}{2}$
end for
Output: Approximate solution $\mathbf{x}^{(k+1)} = \mathbf{V}_d\mathbf{y}^{(k+1)}$

APLB-type methods use a linear combination of two previous iterates and their non-soft-thresholded versions. Secondly, one chooses $s = 1$ for FISTA-type methods for fastest convergence, while in order to make sure that APLB-type methods converge, one picks a value $0 < s < 1$. Numerical experiments, some of which are reported in Section 4, show computed solutions determined by KFISTA to be of higher quality than those determined by KAPLB.

We turn to the convergence of KFISTA-type methods.

Proposition 1. *Let $\mathbf{y}^*$ be a solution of the minimization problem (16), and let $\mathbf{y}^{(k)}$, $k = 1, 2, \ldots$, be the sequence of vectors with Krylov subspace coefficients generated by KFISTA. The number of iterations of KFISTA required to obtain an ϵ-optimal solution, that is, a vector $\tilde{\mathbf{y}}$ such that $F(\tilde{\mathbf{y}}) - F(\mathbf{y}^*) \leq \epsilon$, is at most $\lceil C/\sqrt{\epsilon} - 1\rceil$, where C is a positive constant, $F(\mathbf{y}) := \|\mathbf{B}_{d+1,d}\mathbf{y} - \beta\mathbf{e}_1\|_2^2 + \mu\|\mathbf{y}\|_1$, and $\lceil\alpha\rceil$ denotes the smallest integer larger or equal to $\alpha > 0$.*

Proof. The proposition is a consequence of [3, Theorem 4.4]. □

4 | Numerical Experiments

This section presents a few numerical examples that illustrate the performance of the methods discussed in this paper. We consider

ALGORITHM 3 | Krylov-coefficient APLB (KAPLB).

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\delta \geq \|\mathbf{e}\|_2$, $\beta = \|\mathbf{b}\|_2$, $\mu>0$, $\tau>1$, $s \in (0, 1)$, $t_0 = t_1 = 1$, and $tol>0$.
for $k = 1, 2, \ldots$ **do**
  Compute the decompositions (10) $\mathbf{AV}_k = \mathbf{U}_{k+1}\mathbf{B}_{k+1,k}$.
  $\mathbf{y}_k^* = \beta\mathbf{B}_{k+1,k}^{\dagger}\mathbf{e}_1$, where $\mathbf{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^{k+1}$
  if $\|\mathbf{B}_{k+1,k}\mathbf{y}_k^* - \beta\mathbf{e}_1\|_2 \leq \tau\delta$ **then**
    $d = k + 3$
  end if
end for
Compute the decompositions (10) $\mathbf{AV}_d = \mathbf{U}_{d+1}\mathbf{B}_{d+1,d}$.
$l_f = 2\lambda_{\max}(\mathbf{B}_{d+1,d}^T\mathbf{B}_{d+1,d})$
$\theta_0 = 1, \mathbf{v}^{(1)} = \mathbf{z}^{(1)} = \mathbf{y}^{(1)} = \mathbf{0} \in \mathbb{R}^d$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{v}^{(k+1)} = \mathbf{z}^{(k)} - \mathbf{B}_{d+1,d}^T(\mathbf{B}_{d+1,d}\mathbf{y}^{(k)} - \beta\mathbf{e}_1)$
  $\theta_k = \frac{2}{k+2}$
  $\alpha_k = 1 + \theta_k(\theta_{k-1}^{-1} - 1)$
  $\mathbf{z}^{(k+1)} = \alpha_k\mathbf{v}^{(k+1)} + (1 - \alpha_k)\mathbf{v}^{(k)}$
  $\mathbf{y}^{(k+1)} = \frac{2s}{l_f}\Phi_\mu(\mathbf{z}^{(k+1)})$
  if $\dfrac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\|_2}{\|\mathbf{y}^{(k)}\|_2} \leq tol$ **then**
    exit
  end if
end for
Output: Approximate solution $\mathbf{x}^{(k+1)} = \mathbf{V}_d\mathbf{y}^{(k+1)}$

the restoration of images that have been blurred and contaminated by noise, as well as the inversion of tomographic data.

Continuous space-invariant image blurring can be formulated with the aid of a Fredholm integral equation of the first kind,

$$g(\mathbf{x}, \mathbf{y}) = \int_\Omega \kappa(\mathbf{s} - \mathbf{x}, \mathbf{t} - \mathbf{y}) f(\mathbf{s}, \mathbf{t}) d\mathbf{s}d\mathbf{t}, \qquad (\mathbf{x}, \mathbf{y}) \in \Omega \qquad (19)$$

where g represents a blurred noise-free image, f represents the unknown image to be recovered, and κ is a smooth kernel with compact support. The kernel is referred to as a point-spread function and defines the blur. The integral operator in (19) is compact with Ω a square or rectangle. Seeking the solution f of (19) is an ill-posed problem; see, for example, Engl et al. [1]. Discretizing (19) gives a problem of the form (1). In our applications, the function g is already known in discretized form and is represented by a matrix, whose entries are pixel values of an available blur- and noise-contaminated two-dimensional image. This matrix is vectorized by stacking its columns. This defines the data vector $\mathbf{b}$ in (1). The matrix $\mathbf{A}$ in (1) is obtained by discretizing the integral operator (19). Due to the shift-invariant structure of the kernel κ, the matrix $\mathbf{A}$ is the sum of a block Toeplitz with Toeplitz block (BTTB) matrix, a correction of small norm, and a correction of small rank due to the boundary conditions that are imposed; see, for example, [31, 32] for more details on image deblurring.

Tomography data is obtained by applying the Radon transform. An object is shined on with X-rays, and after each ray has passed through the sample, we measure its residual intensity. The difference between the initial and residual intensities provides

the amount of radiation absorbed along the line. We collect the data by shining rays on different points from various angles. This results in an integral equation; see, for example, [33], for a discussion.

We turn to the stopping criteria (SC) employed in the algorithms. For stationary methods, we terminate the iterations as soon as the relative difference between two consecutive iterates is smaller than or equal to a given tolerance tol > 0, that is, when

$$r := \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \leq \text{tol} \tag{20}$$

Nonstationary iterative methods are terminated with the discrepancy principle (DP), that is, the iterations are stopped as soon as

$$\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\|_2 \leq \tau\delta$$

We set $\tau = 1.01$ in our experiments. The error vector $\mathbf{e}$ in (2) represents white Gaussian noise. We refer to the ratio $\|\mathbf{e}\|_2/\|\mathbf{b}\|_2$ as the noise level.

The quality of the computed restoration $\mathbf{x}$ is measured by the relative restoration error

$$\text{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2} \tag{21}$$

We also compute the normalized structural similarity index (SSIM). The definition of this index is somewhat involved; see [34]., and we do not give it here. We just recall that a larger SSIM value corresponds to a more accurate reconstruction and the maximum value is 1.

To ensure convergence of the APLB and KAPLB methods, we set the step size scalar to $s = 0.9$; the same value as was used in the experiments reported in [5]. For all FISTA-type methods, we let $s = 1$. For nonstationary algorithms, the initial regularization parameter is set to $\mu_0 = 10$ and we let $q = 0.99$; c.f. (9). For stationary algorithms, we determine the regularization parameter either by hand-tuning to minimize the RRE (the smallest value of μ we consider is 0.01 to prevent the errors in the data from being severely magnified) or we use the parameter μ_k at the breakout iteration of the corresponding nonstationary method. This is the iteration at which the DP is satisfied for the first time. We terminate the iteration then. In detail, we proceed as follows:

1. Run the nonstationary algorithm and terminate the iterations with the DP.
2. Let k_{DP} denote the breakout iteration of the nonstationary algorithm.
3. Run the stationary algorithm with $\mu = \mu_{k_{\text{DP}}} := \mu_0 q^{k_{\text{DP}}}$.

Note that the last step does not require much computational effort when using KFISTA, since the main computational burden of steps 1–3 is the computation of the Krylov subspace basis, which already is available.

All the examples were run on an Apple Silicon 8-Core 3.49 GHz MacBook Air laptop computer with 8 GB of RAM, using MacOS Sonoma 14.5 and 64-bit arithmetic. The computations were carried out in MATLAB R2023a Update 2 (9.14.0.2254940) with about 15 significant decimal digits.

4.1 | Hubble

In this example, we consider a synthetic astronomical image of the Hubble telescope of size 496×496 pixels as shown in Figure 1a. We construct the example by blurring the exact image of size 512×512 pixels with the motion-blur PSF displayed in Figure 1b. The PSF simulates the linear motion of a camera by 15 pixels, with an angle of 15° in the counter-clockwise direction. We add 2% Gaussian noise and obtain the blurred and noisy image in Figure 1c. Reflexive boundary conditions are imposed. To simulate realistic data, we crop the boundaries of the images by 8 pixels.

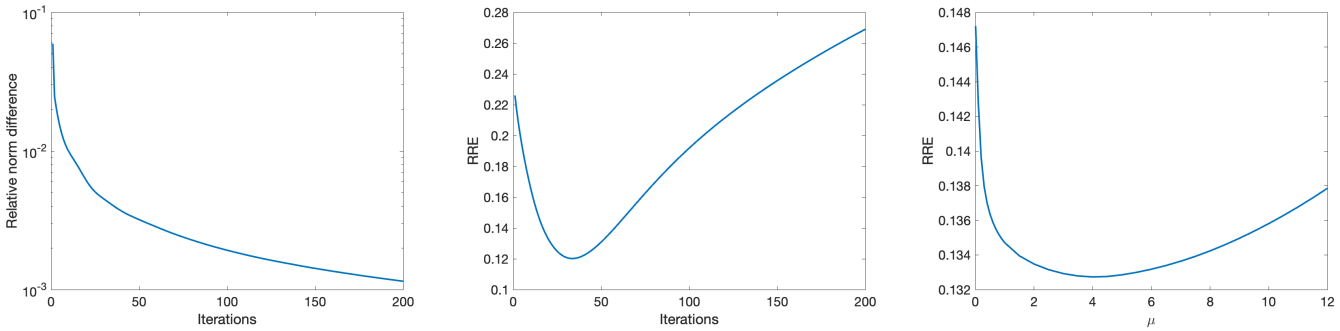
We first look into the issue of choosing a stopping criterion for FISTA. To this aim we run the considered methods using a fixed value of the regularization parameter μ, in particular, we set $\mu = 4$. Figure 2a displays the value of r, defined in (20), at each iteration. We can observe that this value decreases rapidly in the first iterations, but in the subsequent iterations, the rate of decrease reduces significantly. If we would require the iterations to stop as soon as $r \leq 10^{-3}$, then the algorithm would not terminate within 200 iterations, while, if we require that the iterations terminate as soon as $r \leq 10^{-2}$, the algorithm terminates after only 10 iterations; see Table 1. We can observe in Figure 2b, which shows the evolution of the RRE versus the number of iterations that FISTA behaves like a semiconvergent method, that is, the RRE, defined in (21), first decreases quickly and then rapidly increases. Finally, in Figure 2c we display the RRE obtained at the 20th iteration as a function of μ. As expected, we can observe that there is a value of μ that minimizes the RRE. Moreover, the RRE is quite insensitive with respect to small changes in μ when a small number of iterations is carried out. We display the computed solutions by FISTA in Figure 3. Panel (a) depicts the restoration obtained at the 200th iteration, while panel (b) shows the restoration determined when $r \leq 10^{-2}$. Visual inspection shows that neither restorations are satisfactory. The first one is corrupted by noise, while the second one is over-smoothed. This illustrates that a suitable stopping criterion is required for FISTA to produce satisfactory results. We remark that our FISTA code produces the same results as the FISTA code in IR Tools [35] in this as well as in the following examples.

We now turn to KFISTA, and we wish to illustrate how the dimension of the Krylov subspace influences the results. We consider dimensions of the Krylov subspace ranging from 8, the minimum dimension required to satisfy the DP, to 16. In these experiments, we hand-tune μ to minimize the RRE and terminate the iterations as soon as $r \leq 10^{-4}$. The results are reported in Table 2. We can observe that as the dimension increases, the RRE decreases. However, increasing the dimension d over 11 does not substantially improve the results, while the computational cost increases with d. The latter is to be expected as the computations are performed in higher-dimensional spaces. Visual inspection of the computed solutions shown in Figure 4 confirms our observations.

Table 3 compares the KFISTA, FISTA, and APLB methods; the latter method is described in [5]. For the stationary algorithms,

(a) (b) (c)

FIGURE 1 | Hubble test case: (a) True image (502 × 502 pixels), (b) Motion-blur PSF: the linear motion of a camera by 15 pixels, with an angle of 15° in the counter-clockwise direction (5 × 15 pixels), (c) Blurred image with 2% of white Gaussian noise (502 × 502 pixels).



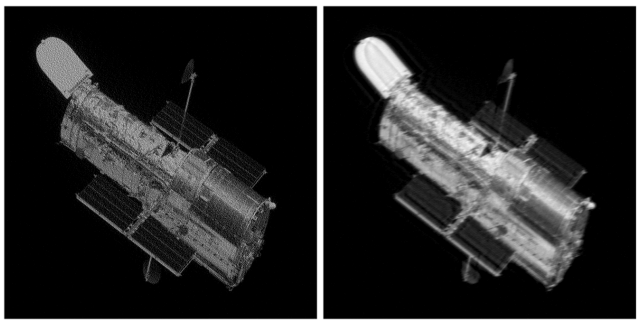
(a) (b) (c)

FIGURE 2 | Hubble test case: Behavior of the FISTA algorithm: (a) Relative norm difference between estimates of two consecutive iterations for $\mu = 4$, (b) RRE as a function of iterations with $\mu = 4$, (c) RRE at the 20th iteration as a function of μ.

TABLE 1 | Hubble test case: FISTA algorithm with $\mu = 4$.

SC	Time (s)	Iter.	RRE	SSIM
$r \leq 10^{-3}$	1.6943	200	0.2690	0.5883
$r \leq 10^{-2}$	0.1345	10	0.1655	0.6603

Notes: We report the stopping criteria, the time elapsed in seconds, the number of iterations performed to satisfy the stopping criterion, the RRE, and the SSIM at the stopping iteration. The underlined number of iterations indicates that the algorithm did not stop within that amount of iterations.



(a) (b)

FIGURE 3 | Hubble test case: Restorations obtained with FISTA with $\mu = 4$ and different stopping criteria: (a) $r \leq 10^{-3}$, (b) $r \leq 10^{-2}$.

we both hand-tune μ and use the value of μ obtained by their nonstationary counterparts. We can observe that FISTA suffers the same convergence issue as shown in Table 1, that is, the algorithm does not stop within 200 iterations. On the other hand, NFISTA successfully satisfies its stopping criterion, that is, the DP, however, the RRE of the solution determined by NFISTA is quite large. We observe that KFISTAF does not terminate within 200 iterations either. Nevertheless, the computed solution at the 200th iteration is accurate. Finally, KFISTA satisfies the stopping criterion in 52 iterations.

TABLE 2 | Hubble test case: KFISTA algorithm with hand-tuned μ and stopped as soon as $r \leq 10^{-4}$ for different dimensions of the Krylov subspace.

d	Time(s)	Iter.	RRE	SSIM	μ
8	0.0666	39	0.1559	0.6800	0.01
9	0.0724	44	0.1522	0.6557	0.01
10	0.0831	48	0.1498	0.6295	0.01
11	0.0899	52	0.1482	0.6036	0.1
12	0.1019	57	0.1473	0.5905	5
13	0.1129	62	0.1468	0.5873	9.5
14	0.1311	68	**0.1466**	0.5808	12
15	0.1294	74	**0.1466**	0.5829	14
16	0.1431	78	**0.1466**	0.5803	14.5

Notes: We report the dimension of Krylov subspace, the time elapsed in seconds, the number of iterations performed, RRE, and SSIM of the computed solutions, and the regularization parameter μ. The smallest error is in boldface.

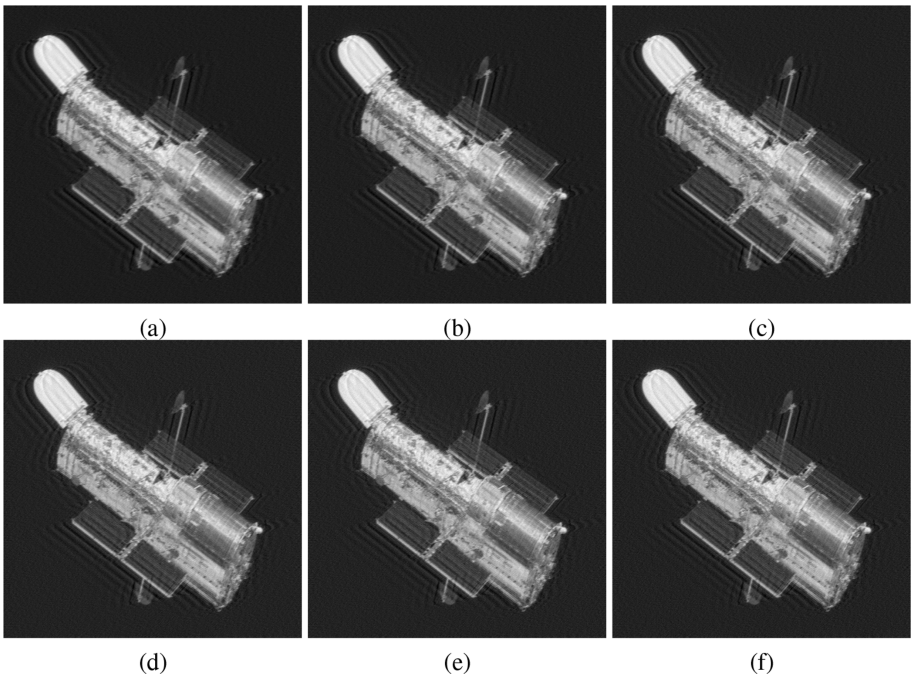


FIGURE 4 | Hubble test case: Restorations obtained with KFISTA with hand-tuned μ: Dimension of Krylov subspace for (a) through (f) increases from 8 to 13.

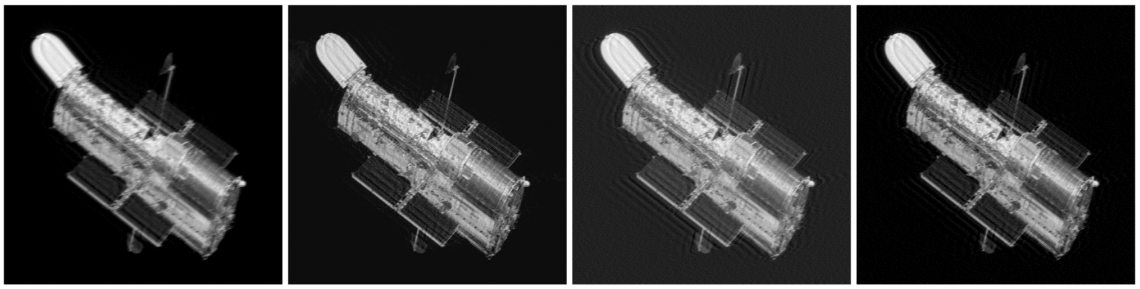
TABLE 3 | Hubble test case: Comparison of the considered methods.

SC	Method	Time (s)	Iter.	RRE	SSIM	μ
$r \leq 10^{-3}$	FISTA	1.7297	200	0.2713	0.6274	2.1704[a]
$r \leq 10^{-2}$	FISTA	0.1286	10	0.1656	0.7605	2.1704[a]
DP	NFISTA	1.2514	153	0.2405	0.6378	2.1704
$r \leq 10^{-4}$	APLB	0.2203	60	**0.1342**	0.6789	12
$r \leq 10^{-4}$	APLB	0.2256	56	0.1405	0.6776	3.3103[a]
$r \leq 10^{-4}$	NAPLB	0.5890	111	0.1405	0.6671	3.3103
$r \leq 10^{-4}$	KAPLB	0.1042	55	0.1482	0.6032	3
$r \leq 10^{-4}$	KFISTA	0.1064	52	0.1488	0.6308	8.6875[a]
DP	NKFISTA	0.1062	15	0.1569	0.7041	8.6875
—	PPKFISTA	—	—	0.1432	0.7259	8.6875[a]
$r \leq 10^{-4}$	KFISTAF	0.3976	200	0.1392	0.6340	0.01

Notes: For each algorithm, we report the employed stopping criterion (SC), time elapsed in seconds, number of iterations performed, RRE and SSIM of the computed solution, and value of the regularization parameter. The smallest error is in boldface. The underlined number of iterations indicates that the algorithm did not stop within that amount of iterations.

[a] μ value is obtained by the corresponding nonstationary version with $\mu_0 = 10$ and $q = 0.99$.

Figure 5a–c display the approximate solutions computed by FISTA and KFISTAF with hand-tuned μ-values, and KFISTA with μ determined by NKFISTA, respectively. We choose $tol = 10^{-2}$ in (20) for FISTA since the method does not stop within 200 iterations when tol $= 10^{-3}$. For all other stationary methods, we let tol $= 10^{-4}$. FISTA and KFISTAF are able to accurately restore the black background, while KFISTA struggles in this task. This can be expected since FISTA and KFISTAF promote sparsity of the restored images, while KFISTA does not. However, the FISTA restoration appears over-smoothed, and details of the restored image are blurry. The restoration computed at the 200th iteration with KFISTAF has the second smallest RRE among all the tested methods. The reconstructed image looks sharper than the one determined by FISTA and shows negligible ringing effect. This suggests that KFISTAF is able to produce accurate restored images when coupled with a suitable stopping criterion. We turn to the restored image determined by KFISTA. This restoration looks sharper than the restoration obtained by FISTA. However, it displays some ringing effects in the restored background. This can be expected since KFISTA promotes sparsity in Krylov coefficients, but not in the restored image. Projecting the KFISTA solution to the nonnegative orthant (as implemented by PPKFISTA) reduces the ringing. This is confirmed by the decrease of the RRE and by the increase of the SSIM; see Figure 5d.



(a) (b) (c) (d)

FIGURE 5 | Hubble test case: Restorations obtained with some of the considered methods: (a) FISTA with hand-tuned μ, (b) KFISTAF with hand-tuned μ, (c) KFISTA with μ determined by NKFISTA, (d) PPKFISTA (obtained projecting the solution displayed in (c) to the non-negative orthant).

TABLE 4 | Hubble test case: Breakdown of the computing time for the selected methods listed in Table 3.

Method	Time for G.-K. (s)	Iteration time	Iter.
APLB (DP param.)	0.1016	0.1240	56
NAPLB	0.1016	0.4874	111
KAPLB	0.1016	0.0026	55
KFISTA (DP param.)	0.1016	0.0048	52
NKFISTA	0.1016	0.0046	15
KFISTAF	0.1016	0.2960	200

Notes: The time required for the construction of the Krylov subspace is denoted by "Time for G.-K.", while the time required for the iterations is denoted by "Iteration time." Finally, in the column "Iter." we report the number of iterations performed. For all methods, the dimension of the Krylov subspace is 11.

The results in Table 3 show that the Krylov subspace-projected algorithms are significantly faster than the non-projected ones. We mentioned in the previous section that the dominant computational effort of KFISTA-type methods is the construction of the Krylov subspace basis. This is confirmed by the fact that the CPU time required to carry out 15 iterations with NKFISTA is about the same as the CPU time required for 52 iterations with KFISTA. In Table 4 we break down the computational time required for the projected methods into two parts, the time required to construct the Krylov subspace basis and the time required to run the iterations. We can observe that for the methods that perform soft-thresholding of the Krylov coefficients, that is, KAPLB, KFISTA, and NKFISTA, the time required to construct the Krylov subspace basis dominates the overall time. On the other hand, for the algorithms that carry out soft-thresholding of the signal, the opposite holds.

4.2 | Tomography

This example is concerned with a synthetic tomography problem. In tomography, the data are obtained as the Radon transform of the attenuation coefficients of a given scanned object. For more details on computed tomography; see, for example, [33]. We consider parallel beam tomography, where J parallel X-ray beams are shined through an object at different angles θ_k with $k = 1, 2, \ldots, K$. The data $b_{j,k}$, the so-called sinogram, is the line integral of the attenuation coefficients of the scanned object along the jth beam at angle θ_k. We generate the data using the MATLAB toolbox IR Tools [35]. The dimensions of the image are set to 512×512, and we consider 90 equally spaced angles between 0 and 179 degrees with 724 beams for each angle. This gives an underdetermined system of equations with a matrix $\mathbf{A} \in \mathbb{R}^{65{,}165 \times 262{,}144}$. Panel (a) of Figure 6 shows the exact attenuation coefficients, panel (b) displays the computed noise-free sinogram, and panel (c) depicts the sinogram corrupted by 5% of white Gaussian noise.

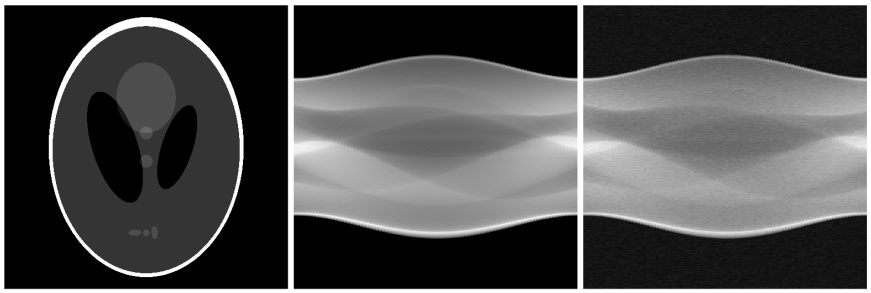
We report the obtained results in Table 5 and show some computed reconstructions in Figure 7. The computed solution in Figure 7a and the results reported in Table 5 show that FISTA terminates too late when we stop the iterations when $r \leq 10^{-2}$, in fact, the computed RRE is fairly large, and the restoration is far from satisfactory even after the computed solution is projected into the nonnegative cone (PPFISTA); see Figure 7b. Our proposed methods consistently generate accurate restorations. Moreover, we can observe that PPKFISTA reduces the RRE, significantly increases the SSIM, and eliminates the artificial grid effect in the black background of the restoration determined by KFISTA; see Figure 7c, d. KFISTAF gives a restoration with a smaller RRE and larger SSIM than KFISTA, however, the computational cost is significantly larger than the cost of KFISTA.

4.3 | Axelsson

Our next example is a photo of Owe Axelsson of size 452×500 pixels shown in Figure 8a. The original image, of size 470×518 pixels, is blurred by the non-symmetric PSF shown in Figure 8b and corrupted by white Gaussian noise with a noise level of 0.01. This gives the blurred and noisy image in Figure 8c. We impose reflexive boundary conditions [31, 32, 36]. To simulate realistic data, we crop the boundaries by 18 pixels to avoid artificial effects by the boundary conditions. We list the results obtained with different methods in Table 6 and show the computed solution determined by PPKFISTA in Figure 8d. Similarly to the results in the previous examples, KFISTA outperforms other methods both in terms of accuracy and computational cost.

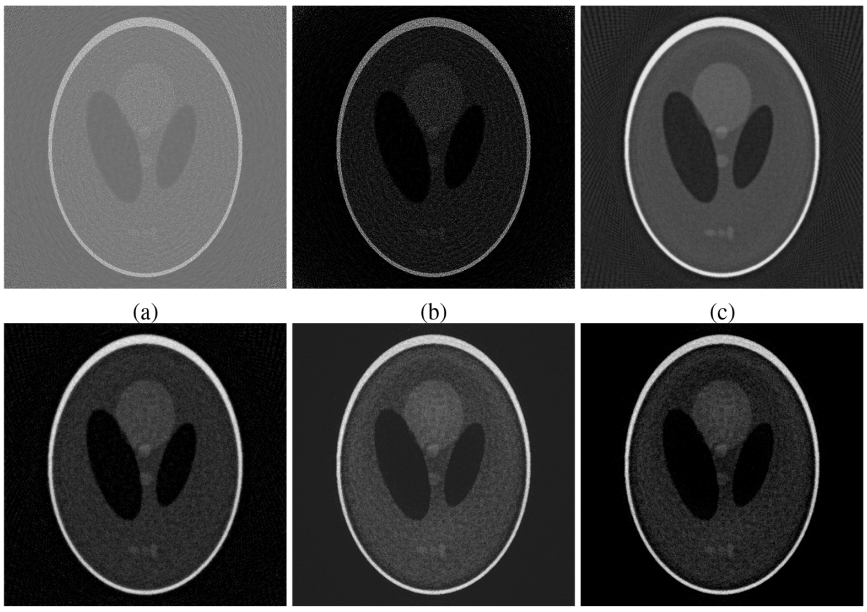
4.4 | Cameraman

For our last example, we consider the cameraman image shown in Figure 9a. The original image, of size 512×512 pixels, is



(a) (b) (c)

FIGURE 6 | Tomography test case: (a) True image (512×512 pixels), (b) noise-free sinogram (724×90 pixels), (c) sinogram corrupted by 5% of white Gaussian noise (724×90 pixels).



(a) (b) (c)

(d) (e) (f)

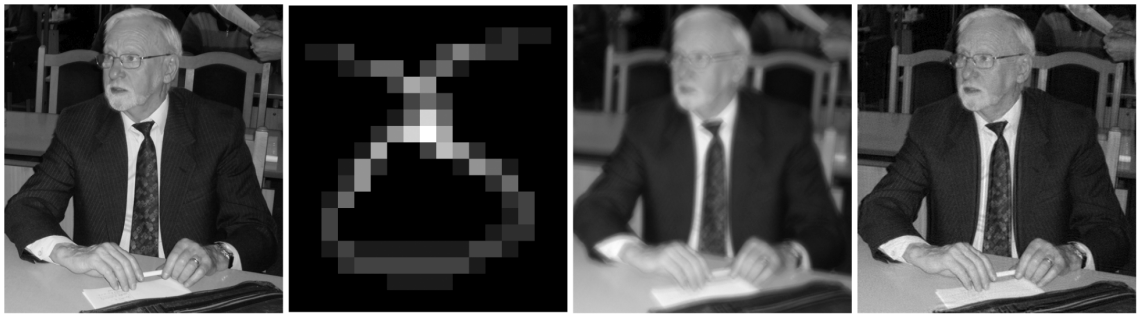
FIGURE 7 | Tomography test image restorations: (a) FISTA with hand-tuned μ, (b) PPFISTA, (c) KFISTA with μ obtained with NKFISTA, (d) PPKFISTA, (e) KFISTAF with hand-tuned μ, (f) PPKFISTAF.

TABLE 5 | Tomography test case: Comparison of the considered methods.

SC	Method	Time (s)	Iter.	RRE	SSIM	μ
$r \leq 10^{-2}$	FISTA	18.7620	466	0.6334	0.0928	7.5
—	PPFISTA	—	—	0.5644	0.0934	—
$r \leq 10^{-2}$	FISTA	16.8289	418	1.1394	0.0588	0.3450[a]
DP	NFISTA	13.4647	336	1.1550	0.0594	0.3450
$r \leq 10^{-4}$	KAPLB	0.2618	133	0.3055	0.2209	3
$r \leq 10^{-4}$	KFISTA	0.2726	426	0.2960	0.2731	35
$r \leq 10^{-4}$	KFISTA	0.2706	535	0.3005	0.2343	1.8667[a]
DP	NKFISTA	0.2712	168	0.3022	0.2367	1.8667
—	PPKFISTA	—	—	0.2720	0.2510	—
$r \leq 10^{-4}$	KFISTAF	4.2897	2868	0.2718	0.5677	0.024
—	PPKFISTAF	—	—	**0.2714**	0.5726	—

Notes: For each algorithm, we report the employed stopping criterion, time elapsed in seconds, number of iterations performed, RRE and SSIM of the computed solution, and value of the regularization parameter. The smallest error is in boldface.

[a] μ-value is obtained by the corresponding nonstationary version with $\mu_0 = 10$ and $q = 0.99$.



(a) (b) (c) (d)

FIGURE 8 | Axelsson test case: (a) True image (452×500 pixels), (b) PSF (17×17 pixels), (c) Blurred and noisy image with $\delta = 0.01\|b\|_2$, (d) PPKFISTA restoration.

TABLE 6 | Axelsson test case, comparison between the considered methods.

SC	Method	Time (s)	Iter.	RRE	SSIM	μ
DP	NFISTA	2.2114	174	0.2129	0.3444	1.7575
$r \leq 10^{-2}$	FISTA	0.1639	8	0.0950	0.8331	1.7575[a]
DP	NAPLB	1.0614	161	0.0750	0.7830	2.002
$r \leq 10^{-4}$	APLB	0.2611	59	0.0740	0.7908	2.002[a]
DP	NKAPLB	0.2442	16	0.0745	0.8424	8.6006
$r \leq 10^{-4}$	KAPLB	0.1700	59	0.0739	0.7909	8.6006[a]
DP	NKFISTA	0.1773	17	0.0738	0.8410	8.5146
$r \leq 10^{-4}$	KFISTA	0.1755	55	0.0719	0.8150	8.5146[a]
—	PPKFISTA	—	—	**0.0716**	0.8159	8.5146[a]

Notes: For each algorithm, we report the employed stopping criterion, time elapsed in seconds, number of iterations performed, RRE and SSIM of the computed solution, and value of the regularization parameter. The smallest error is in boldface.

[a] μ-value is obtained by the corresponding nonstationary version with $\mu_0 = 10$ and $q = 0.99$.

(a) (b) (c) (d)

FIGURE 9 | Cameraman test case: (a) True image (512×512 pixels), blurred and noisy images with (b) $\delta = 0.01\|b\|_2$ (506×506 pixels), (c) $\delta = 0.03\|b\|_2$ (504×504 pixels), (d) $\delta = 0.05\|b\|_2$ (500×500 pixels).

blurred by a selection of rotationally symmetric Gaussian lowpass filters of different sizes and standard deviations σ; see Table 7. The blurred images are then added with different levels of Gaussian noise. Reflexive boundary conditions are imposed [31, 32, 36]. To simulate realistic data, we crop the boundaries to avoid artificial effects due to the boundary conditions; see Figure 9b–d. We compare the results obtained with KFISTA and FISTA as follows. For each method, we run the nonstationary versions first, with the discrepancy principle as stopping criterion, and with $\mu_0 = 10$ and $q = 0.99$. We then run the stationary versions of these algorithms with the values of μ obtained from the nonstationary methods. The stopping criterion for KFISTA is $r \leq 10^{-4}$. Since iterations with FISTA do not terminate within 200 iterations when choosing $r \leq 10^{-3}$ as the stopping criterion, we employ the stopping criterion $r \leq 10^{-2}$ instead. For KFISTA, the dimensions of Krylov subspaces determined by the algorithm are all 7. We exclude the time elapsed for generating the Krylov subspaces since they have already been generated by NKFISTA. The results are listed in Table 7, which illustrates that KFISTA is a good general-purpose method that consistently outperforms FISTA both with regard to computational cost and quality of the restored image.

TABLE 7 | Cameraman test case.

SC	Method	Time (s)	Iter.	RRE	SSIM	μ	Noise level	h	σ
DP	NKFISTA	0.0599	5	0.0211	0.9733	9.606	0.01	5	1
$r \leq 10^{-4}$	KFISTA	0.003	17	**0.0204**	0.9461	9.606	0.01	5	1
DP	NFISTA	1.1225	146	0.1922	0.2653	2.3286	0.01	5	1
$r \leq 10^{-2}$	FISTA	0.0234	3	0.0277	0.9705	2.3286	0.01	5	1
DP	NKFISTA	0.0613	6	0.0685	0.8425	9.5099	0.03	7	3
$r \leq 10^{-4}$	KFISTA	0.0032	33	**0.0658**	0.7529	9.5099	0.03	7	3
DP	NFISTA	0.6228	71	0.2661	0.1785	4.9484	0.03	7	3
$r \leq 10^{-2}$	FISTA	0.0429	5	0.0721	0.8429	4.9484	0.03	7	3
DP	NKFISTA	0.0652	6	0.1049	0.7455	9.5099	0.05	11	5
$r \leq 10^{-4}$	KFISTA	0.0032	34	**0.0996**	0.6517	9.5099	0.05	11	5
DP	NFISTA	0.439	54	0.2723	0.1634	5.8704	0.05	11	5
$r \leq 10^{-2}$	FISTA	0.0376	5	0.1088	0.744	5.8704	0.05	11	5

Notes: For each algorithm, we report the employed stopping criterion, dimension of the Krylov subspace, time elapsed in seconds, number of iterations performed, RRE and SSIM of the computed solution, value of the regularization parameter, noise level, filter size h, and standard deviation σ. The smallest error for each restoring case is in boldface. The elapsed time for generating Krylov subspace is excluded for KFISTA.

5 | Conclusions

This paper proposes to reduce the computational effort required by FISTA by combining this method with projection into a low-dimensional Krylov subspace. Several approaches are described, and numerical experiments illustrate their performance. One of these variants, KFISTA, is found to perform well and require less CPU time than other approaches considered.

Author Contributions

All authors contributed equally.

Acknowledgments

The authors would like to thank the referees for comments that lead to clarifications of the presentation. Open access publishing facilitated by Universita degli Studi di Cagliari, as part of the Wiley - CRUI-CARE agreement.

Disclosure

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

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

Endnotes

[1] Throughout this paper we say that the quality of an image is higher if its distance in the least-squares norm from the desired image $\mathbf{x}_{\text{true}}$ is smaller.

References

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

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

3. A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences* 2, no. 1 (2009): 183–202.

4. A. Buccini, Y. Park, and L. Reichel, "Numerical Aspects of the Nonstationary Modified Linearized Bregman Algorithm," *Applied Mathematics and Computation* 377 (2018): 386–398.

5. A. Buccini, M. Pasha, and L. Reichel, "Linearized Krylov Subspace Bregman Iteration With Nonnegativity Constraint," *Numerical Algorithms* 87 (2021): 1177–1200.

6. J. F. Cai, S. Osher, and Z. Shen, "Linearized Bregman Iterations for Compressed Sensing," *Mathematics of Computation* 78, no. 267 (2009): 1515–1536.

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

8. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An Iterative Regularization Method for Total Variation-Based Image Restoration," *Multiscale Modeling and Simulation* 4, no. 2 (2005): 460–489.

9. E. J. Candes, J. K. Romberg, and T. Tao, "Stable Signal Recovery From Incomplete and Inaccurate Measurements," *Communications on Pure and Applied Mathematics* 59, no. 8 (2006): 1207–1223.

10. M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging," *Magnetic Resonance in Medicine* 58, no. 6 (2007): 1182–1195.

11. Y. Tsaig and D. L. Donoho, "Extensions of Compressed Sensing," *Signal Processing* 86, no. 3 (2006): 549–571.

12. L. M. Bregman, "The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming," *USSR Computational Mathematics and Mathematical Physics* 7, no. 3 (1967): 200–217.

13. A. Buades, B. Coll, and J. M. Morel, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Review* 52 (2010): 123–147.

14. T. A. Hearn and L. Reichel, "Application of Denoising Methods to Regularization of Ill-Posed Problems," *Numerical Algorithms* 66 (2014): 761–777.

15. T. A. Hearn and L. Reichel, "Image Denoising via Residual Kurtosis Minimization," *Numerical Mathematics Theory Methods and Applications* 8 (2015): 403–422.

16. P. C. Hansen, T. K. Jensen, and G. Rodriguez, "An Adaptive Pruning Algorithm for the Discrete L-Curve Criterion," *Journal of Computational and Applied Mathematics* 198, no. 2 (2007): 483–492.

17. P. C. Hansen and D. P. O'Leary, "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems," *SIAM Journal on Scientific Computing* 14, no. 6 (1993): 1487–1503.

18. 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.

19. J. Chung, J. G. Nagy, and D. P. O'Leary, "A Weighted GCV Method for Lanczos Hybrid Regularization," *Electronic Transactions on Numerical Analysis* 28 (2008): 149–167.

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

21. M. Hanke and C. W. Groetsch, "Nonstationary Iterated Tikhonov Regularization," *Journal of Optimization Theory and Applications* 98, no. 1 (1998): 37–53.

22. O. Axelsson, *Iterative Solution Methods* (Cambridge: Cambridge University Press, 1984).

23. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation* (London: Academic Press, 1984).

24. O. Axelsson and J. Karatson, "Krylov Improvements of the Uzawa Method for Stokes Type Operator Matrices," *Numerische Mathematik* 148 (2021): 611–631.

25. Z. Z. Liang and O. Axelsson, "Exact Inverse Solution Techniques for a Class of Complex Valued Block Two-By-Two Linear Systems," *Numerical Algorithms* 90 (2022): 79–98.

26. Z. Z. Liang, O. Axelsson, and M. Neytcheva, "A Robust Structured Preconditioner for Time-Harmonic Parabolic Optimal Control Problems," *Numerical Algorithms* 79 (2018): 575–596.

27. M. Brill and E. Schock, "Iterative Solution of Ill-Posed Problems-a Survey," in *Model Optimization in Exploration Geophysics*, ed. A. Vogel (Vieweg, Braunschweig: 1987), 17–37.

28. A. Buccini, M. Donatelli, and L. Reichel, "Iterated Tikhonov Regularization With a General Penalty Term," *Numerical Linear Algebra With Applications* 24, no. 4 (2017): e2089.

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

30. B. Huang, S. Ma, and D. Goldfarb, "Accelerated Linearized Bregman Method," *Journal of Scienific Computing* 54 (2013): 428–453.

31. M. Donatelli and L. Reichel, "Square Smoothing Regularization Matrices With Accurate Boundary Conditions," *Journal of Computatioal and Applied Mathematics* 272 (2014): 334–349.

32. P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering* (Philadelphia: SIAM, 2006).

33. T. M. Buzug, "Computed Tomography," in *Springer Handbook of Medical Technology*, eds. R. Kramme, K. P. Hoffmann, and R. S. Pozos (Berlin, Heidelberg: Springer, 2011), 311–342.

34. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing* 13, no. 4 (2004): 600–612.

35. 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.

36. M. K. Ng, R. H. Chan, and W. C. Tang, "A Fast Algorithm for Deblurring Models With Neumann Boundary Conditions," *SIAM Journal on Scientific Computing* 21, no. 3 (1999): 851–866.