An Arnoldi-based preconditioner for iterated Tikhonov regularization

Alessandro Buccini[*]   Lucas Onisk[†]   Lothar Reichel[‡]

Dedicated to Claude Brezinski on the occasion of his 80th birthday.

Abstract

Many problems in science and engineering give rise to linear systems of equations that are commonly referred to as large-scale linear discrete ill-posed problems. These problems arise, for instance, from the discretization of Fredholm integral equations of the first kind. The matrices that define these problems are typically severely ill-conditioned and may be rank-deficient. Because of this, the solution of linear discrete ill-posed problems may not exist or be very sensitive to perturbations caused by errors in the available data. These difficulties can be reduced by applying Tikhonov regularization. We describe a novel "approximate Tikhonov regularization method" based on constructing a low-rank approximation of the matrix in the linear discrete ill-posed problem by carrying out a few steps of the Arnoldi process. The iterative method so defined is transpose-free. Our work is inspired by a scheme by Donatelli and Hanke, whose approximate Tikhonov regularization method seeks to approximate a severely ill-conditioned block-Toeplitz matrix with Toeplitz-blocks by a block-circulant matrix with circulant-blocks. Computed examples illustrate the performance of our proposed iterative regularization method.

Keywords: discrete ill-posed inverse problems, iterated Tikhonov method, spectral equivalence.

Mathematics Subject Classification (2010): 65F10, 65F22, 90C06.

1   Introduction

We are interested in computing approximate solutions to linear least-squares problems of the form

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^{\boldsymbol{\delta}}\|, \tag{1}$$

---

[*]Department of Mathematics and Computer Science, University of Cagliari, 09124 Cagliari, Italy, `alessandro.buccini@unica.it`

[†]Department of Mathematical Sciences, Kent State University, Kent, 44242 OH, `lonisk@kent.edu`

[‡]Department of Mathematical Sciences, Kent State University, Kent, 44242 OH, `reichel@math.kent.edu`

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a square severely ill-conditioned or rank-deficient matrix, whose singular values "cluster" at the origin. Throughout, we let $\| \cdot \|$ denote the Euclidean norm. Least-squares problems with a matrix of this kind are commonly referred to as linear discrete ill-posed problems. They arise, for instance, by discretizing Fredholm integral equations of the first kind; see [7,12]. We focus on applications to image deblurring, but the method proposed also can be used with other applications.

The vector $\boldsymbol{b}^\delta$ in (1) often represents measurements that are corrupted by error; we denote this error by $\boldsymbol{e}$ so that

$$\boldsymbol{b}^\delta = \boldsymbol{b} + \boldsymbol{e},$$

where $\boldsymbol{b}$ is the unknown error-free vector associated with $\boldsymbol{b}^\delta$. We would like to determine the solution of minimal Euclidean norm, $\boldsymbol{x}^\dagger$, of the unavailable least-squares problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|. \tag{2}$$

Since $\boldsymbol{b}^\delta$ is contaminated by error and the singular values of the matrix $\boldsymbol{A}$ cluster at the origin, the solution of (1) of minimal Euclidean norm typically is a poor approximation of $\boldsymbol{x}^\dagger$. For this reason, we replace the problem (1) by a nearby problem whose solution is less sensitive to the error $\boldsymbol{e}$ in $\boldsymbol{b}^\delta$. This replacement is known as *regularization*. The most popular and well-understood regularization method is due to Tikhonov. The simplest form of Tikhonov regularization replaces the least-squares problem (1) by the penalized least-squares problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \{\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\delta\|^2 + \alpha\|\boldsymbol{x}\|^2\}, \tag{3}$$

where $\alpha > 0$ is a regularization parameter. The solution of (3) can be expressed as

$$\boldsymbol{x}^{(\alpha)} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \alpha I)^{-1}\boldsymbol{b}^\delta, \tag{4}$$

where the superscript ^{T} denotes transposition. The regularization parameter $\alpha > 0$ determines the sensitivity of the vector $\boldsymbol{x}^{(\alpha)}$ to the error in $\boldsymbol{b}^\delta$, as well as the closeness of $\boldsymbol{x}^{(\alpha)}$ to the desired solution $\boldsymbol{x}^\dagger$.

There are many techniques for determining a suitable value of α including generalized cross validation and the L-curve criterion; see [2,8,15,16,18] for discussions and illustrations. Another fairly common technique is the *discrepancy principle*. It requires knowledge of a fairly accurate error bound

$$\|\boldsymbol{e}\| \le \delta, \tag{5}$$

as well as that the system (2) be consistent. The discrepancy principle prescribes that $\alpha > 0$ be chosen so that

$$\|\boldsymbol{A}\boldsymbol{x}^{(\alpha)} - \boldsymbol{b}^\delta\| = \tau\delta,$$

where $\tau > 1$ is a user-specified constant that is independent of δ. It can be shown that $\boldsymbol{x}^{(\alpha)} \to \boldsymbol{x}^\dagger$ as $\delta \searrow 0$; see [7] for a proof in a Hilbert space setting.

We are interested in developing an iterative method for the computation of $\boldsymbol{x}^{(\alpha)}$ when $\boldsymbol{A}$ is a large matrix and matrix-vector products $\boldsymbol{A}\boldsymbol{w}$ for $\boldsymbol{w} \in \mathbb{R}^n$ can be evaluated fairly inexpensively, but matrix-vector products $\boldsymbol{A}^T\boldsymbol{w}$ cannot. This situation arises, for instance, when $\boldsymbol{A}$ represents a discretization of an integral operator and $\boldsymbol{A}\boldsymbol{w}$ is evaluated by a multipole method. It also arises when $\boldsymbol{A}$ represents the Jacobian of a nonlinear problem; see [5]. We therefore focus on an iterative method that only requires matrix-vector product evaluations with $\boldsymbol{A}$, but not with $\boldsymbol{A}^T$.

Let $\boldsymbol{x}^{(k)}$ denote the approximate solution of the least-squares problem (1) determined at step k of an iterative method. Define the (unknown) error

$$\boldsymbol{e}^{(k)} = \boldsymbol{x}^{\dagger} - \boldsymbol{x}^{(k)}.$$

We may approximate $\boldsymbol{e}^{(k)}$ by the solution $\boldsymbol{h}^{(k)}$ of the error equation

$$\boldsymbol{A}\boldsymbol{h}^{(k)} = \boldsymbol{r}^{(k)}, \qquad \boldsymbol{r}^{(k)} = \boldsymbol{b}^{\boldsymbol{\delta}} - \boldsymbol{A}\boldsymbol{x}^{(k)}. \tag{6}$$

As the matrix $\boldsymbol{A}$ is ill-conditioned and the residual $\boldsymbol{r}^{(k)}$ is contaminated by error, we determine an approximate solution of (6) by Tikhonov regularization

$$\boldsymbol{h}^{(k)} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{r}^{(k)}.$$

Then, generally, an improved approximation of $\boldsymbol{x}^{\dagger}$ is given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{h}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{r}^{(k)}. \tag{7}$$

The iterations may be repeated and the resulting solution method is known as the *iterated Tikhonov* method; see, e.g., [2,3,11] for discussions of this method. These iterations may be terminated by the discrepancy principle, i.e., the iterative process is stopped once the kth iterate, $\boldsymbol{x}^{(k)}$, satisfies

$$\|\boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}^{\boldsymbol{\delta}}\| \leq \tau\delta. \tag{8}$$

As the computed solution of (7) may be sensitive to the choice of the regularization parameter $\alpha^{(k)}$ at step k, Donatelli and Hanke use a modified Levenberg-Marquardt iteration, first suggested in [10], to solve

$$\boldsymbol{A}\boldsymbol{e}^{(k)} \approx \boldsymbol{r}^{(k)} \tag{9}$$

up to a certain relative error, i.e., they solve

$$\begin{aligned} q^{(k)}\|\boldsymbol{r}^{(k)}\| &= \|\boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{h}^{(k)}\| \\ &= \|\boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{r}^{(k)}\|, \end{aligned} \tag{10}$$

where the lower bound of $0 < q^{(k)} < 1$ is a user-specified constant. This approach to determine the regularization parameter using a damped version of the discrepancy principle has proved to be useful when $\boldsymbol{A}$ has a structure that allows fast execution of each step of iterated Tikhonov regularization.

A major drawback of the iterative method (7) is that if $\boldsymbol{A}$ is large, then a large system of linear equations with the matrix $\boldsymbol{A}\boldsymbol{A}^T + \alpha^{(k)}\boldsymbol{I}$ has to be solved in each iteration. Moreover, the transpose of $\boldsymbol{A}$ may be unavailable or expensive to compute. In [6] Donatelli and Hanke address these issues by replacing $\boldsymbol{A}$ in (7) by a matrix $\boldsymbol{C}$, whose structure allows for fast computation and transposition. The matrix $\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{C}^T + \alpha^{(k)}\boldsymbol{I})^{-1}$ so obtained is referred to as a *preconditioner*; see [11]. The algorithm in [6], termed the *approximated iterated Tikhonov* (AIT) method, was demonstrated to give accurate image reconstructions for a reasonable computational cost when $\boldsymbol{A}$ is a matrix that has dominant block-Toeplitz with Toeplitz-blocks (BTTB) substructure and $\boldsymbol{C}$ is a suitably chosen block-circulant with circulant-blocks (BCCB) matrix. Analyses of the AIT method are provided in [4, 6]. Under the assumption that $\boldsymbol{A}$ and $\boldsymbol{C}$ are spectrally equivalent (see below), it is shown in [6] that the AIT method is a regularization method in a well-defined sense. However, often times in applications, the spectral equivalence assumption is not satisfied; this issue is addressed in [4]. Irrespective of this, the AIT method provides fast and reliable image reconstructions when the available image, represented by the vector $\boldsymbol{b}^{\boldsymbol{\delta}}$, is contaminated by sufficiently much noise.

In this paper, we allow $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ to be a fairly general large matrix whose singular values "cluster" at the origin, and let the matrix $\boldsymbol{C}$ be a low-rank matrix that is constructed by applying a few steps of the Arnoldi process to the the matrix $\boldsymbol{A}$. We will refer to the iterative method so obtained as the *iterated Arnoldi-Tikhonov* (IAT) method. Our analysis of this method is closely related to the analysis of the AIT method in [6]. The main advantage of our IAT method, when compared to the AIT method, is that we do not require $\boldsymbol{A}$ to have a particular structure.

The rest of this paper is structured as follows: Section 2 describes the AIT method by Donatelli and Hanke [6] and reviews its properties. In Section 3 we introduce the IAT method and present its analysis. Section 4 provides a brief background on image deblurring problems and reports some numerical examples. We conclude in Section 5 with some remarks.

2 The Approximated Iterated Tikhonov Method

This section reviews the AIT method and the theoretical results shown by Donatelli and Hanke [6]. A modification of this method is discussed in [4]; we will not consider this modification in the present paper. As mentioned above, the AIT method is a regularization method in a well-defined sense under the assumption that $\boldsymbol{A}$ and $\boldsymbol{C}$ are spectrally equivalent; this condition is stated as follows:

Assumption 1 (Spectral Equivalence)**.** *The matrices* $\boldsymbol{A}, \boldsymbol{C} \in \mathbb{R}^{m \times n}$ *are said to be* spectrally equivalent *if for some constant* $0 < \rho < 1/2$ *it holds*

$$\|(\boldsymbol{A} - \boldsymbol{C})\boldsymbol{z}\| \leq \rho \|\boldsymbol{A}\boldsymbol{z}\|. \tag{11}$$

Note that condition (11) implies that $\mathcal{N}(\boldsymbol{A}) \subseteq \mathcal{N}(\boldsymbol{C})$. The following algorithm describes the AIT method.

Algorithm 1 (Approximated Iterated Tikhonov (AIT) method). *Let $\boldsymbol{A}$ and $\boldsymbol{C}$ satisfy Assumption 1 for some $0 < \rho < 1/2$. Fix $q \in (2\rho, 1)$. Let $\delta > 0$ satisfy (5), and let $\boldsymbol{x}^{(0)}$ be an initial guess for $\boldsymbol{x}^{\dagger}$.*

> $\boldsymbol{r}^{(0)} = \boldsymbol{b}^{\delta} - \boldsymbol{A}\boldsymbol{x}^{(0)}$;
> $\tau = \frac{1+2\rho}{1-2\rho}$;
> **for** $k = 0, 1, \ldots$ **do**
> > $\tau^{(k)} = \|r^{(k)}\|/\delta$;
> > $q^{(k)} = \max\{q, 2\rho + (1+\rho)/\tau^{(k)}\}$;
> > Determine $\alpha^{(k)}$ such that
> > $\quad q^{(k)}\|\boldsymbol{r}^{(k)}\| = \|\boldsymbol{r}^{(k)} - \boldsymbol{C}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{C}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{r}^{(k)}\|$;
> > $\boldsymbol{h}^{(k)} = \boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{C}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{r}^{(k)}$;
> > $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{h}^{(k)}$;
> > $\boldsymbol{r}^{(k+1)} = \boldsymbol{b}^{\delta} - \boldsymbol{A}\boldsymbol{x}^{(k+1)}$;
> > **if** $\|\boldsymbol{r}^{(k+1)}\| \leq \tau\delta$ **then**
> > > Stop;
> > **end**
> **end**

We review the essential results by Donatelli and Hanke [6] regarding the AIT method without proofs.

Proposition 1. *Let Assumption 1 hold for some $0 < \rho < 1/2$, and let $\tau_* = (1 + \rho)/(1 - 2\rho)$. Then, if $\tau^{(k)} = \|\boldsymbol{r}^{(k)}\|/\delta > \tau_*$, it follows that*

$$\|\boldsymbol{r}^{(k)} - \boldsymbol{C}\boldsymbol{e}^{(k)}\| \leq \left(\rho + \frac{1+\rho}{\tau^{(k)}}\right)\|\boldsymbol{r}^{(k)}\| < (1 - \rho)\|\boldsymbol{r}^{(k)}\|,$$

where $\boldsymbol{x}^{(k)}$ is the kth iterate generated by the AIT method, $\boldsymbol{e}^{(k)} = \boldsymbol{x}^{\dagger} - \boldsymbol{x}^{(k)}$, and $\boldsymbol{r}^{(k)} = \boldsymbol{b}^{\delta} - \boldsymbol{A}\boldsymbol{x}^{(k)}$.

Proposition 2. *Let Assumption 1 hold. Then the norm of the error, $\boldsymbol{e}^{(k)}$, in the kth iterate, $\boldsymbol{x}^{(k)}$, generated by Algorithm 1 decreases monotonically as k increases,*

$$\|\boldsymbol{e}^{(k)}\|^2 - \|\boldsymbol{e}^{(k+1)}\|^2 \geq 2\rho\|(\boldsymbol{C}\boldsymbol{C}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\|\|\boldsymbol{r}^{(k)}\|,$$

as long as $\|\boldsymbol{r}^{(k)}\| > \tau\delta$, where $\boldsymbol{r}^{(k)} = \boldsymbol{b}^{\delta} - \boldsymbol{A}\boldsymbol{x}^{(k)}$.

Corollary 1. *Under the assumptions of Proposition 2, let k_{δ} denote the index of the last iterate determined by Algorithm 1. Then*

$$\|\boldsymbol{e}_0\|^2 \geq 2\rho \sum_{i=0}^{k_{\delta}-1} \|(\boldsymbol{C}\boldsymbol{C}^T + \alpha^{(i)}\boldsymbol{I})^{-1}\|\|\boldsymbol{r}^{(i)}\| \geq c \sum_{i=0}^{k_{\delta}-1} \|\boldsymbol{r}^{(i)}\|^2,$$

for some constant $c > 0$ that only depends on ρ and q.

Theorem 1. *Assume that $\delta = 0$ and that $\boldsymbol{x}^{(0)}$ is not a solution of (2). Then the sequence of iterates $\boldsymbol{x}^{(k)}$, $k = 0, 1, 2, \ldots$, generated by Algorithm 1 converges as $k \to \infty$ to the solution of (1) that is closest to $\boldsymbol{x}^{(0)}$.*

The following final result states that the AIT algorithm is an iterative regularization method.

Theorem 2. *Let Assumption 1 be valid for some $0 < \rho < 1/2$, and let $\delta \mapsto \boldsymbol{b}^{\boldsymbol{\delta}}$ be a function such that (5) holds for all $\delta > 0$. For fixed parameters τ and q, let $\boldsymbol{x}^{\delta}$ denote the approximate solution computed by Algorithm 1. Then, as $\delta \to 0$, $\boldsymbol{x}^{\delta}$ converges to the solution of (1) that is closest to $\boldsymbol{x}^{(0)}$.*

3 The Iterated Arnoldi–Tikhonov Method

We now turn our attention to the IAT method and its relation to the AIT method. The use of a BCCB matrix $\boldsymbol{C}$ to approximate $\boldsymbol{A}$ in [6] was justified by the ease of computation of matrix-vector products as well as inversion with such matrices; see Section 4. In a similar light, we utilize the Arnoldi process and the orthogonality of the vectors generated by this process to simplify the computations.

The pth step of the Arnoldi process applied to the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with initial vector $\boldsymbol{b}^{\boldsymbol{\delta}}$ generically yields the decomposition

$$\boldsymbol{A}\boldsymbol{V}_p = \boldsymbol{V}_{p+1}\boldsymbol{H}_{p+1,p}, \tag{12}$$

where the matrix $\boldsymbol{V}_{p+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{p+1}] \in \mathbb{R}^{n \times (p+1)}$ has orthonormal columns $\boldsymbol{v}_j$ for $j = 1, 2, \ldots, p+1$ with $\boldsymbol{v}_1 = \boldsymbol{b}^{\boldsymbol{\delta}}/\|\boldsymbol{b}^{\boldsymbol{\delta}}\|$, and $\boldsymbol{V}_p \in \mathbb{R}^{n \times p}$ is made up of the first p columns of $\boldsymbol{V}_{p+1}$. Further, $\boldsymbol{H}_{p+1,p} \in \mathbb{R}^{(p+1) \times p}$ is of upper Hessenberg form. The range of $\boldsymbol{V}_p$ is the Krylov subspace

$$\mathbb{K}_p = \mathrm{span}\{\boldsymbol{b}^{\boldsymbol{\delta}}, \boldsymbol{A}\boldsymbol{b}^{\boldsymbol{\delta}}, \ldots, \boldsymbol{A}^{p-1}\boldsymbol{b}^{\boldsymbol{\delta}}\};$$

see, e.g., Saad [19] for details on the Arnoldi process.

At step p we utilize the Arnoldi decomposition (12) to approximate $\boldsymbol{A}$. Because of this, we seek the projected solution of $\boldsymbol{x}^{\dagger}$ in the Krylov subspace $\mathbb{K}_p$ using the column space of $\boldsymbol{V}_p$. We will denote this projected solution in $\mathbb{K}_p$ by $\boldsymbol{x}_p^{\dagger}$. The iterated Tikhonov method may be simplified by substituting (12) into (7) to obtain

$$\begin{aligned} \boldsymbol{x}_p^{(k+1)} &= \boldsymbol{x}_p^{(k)} + \boldsymbol{h}_p^{(k)} \\ &= \boldsymbol{x}_p^{(k)} + \boldsymbol{V}_p(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha_p^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_{p+1}^T\boldsymbol{r}_p^{(k)} \end{aligned} \tag{13}$$

as an improved approximation to $\boldsymbol{x}_p^{\dagger}$. Since we may select a different regularization parameter $\alpha_p^{(k)}$ at each iteration, we refer to the iterative method

(13) as a *nonstationary preconditioned iterative* method, analogously with the terminology used in [6].

The following theorem connects Assumption 1 of Section 2 with our proposed method.

Theorem 3. *Let the matrices $\boldsymbol{A}$ and*

$$\boldsymbol{A}_p = \boldsymbol{V}_{p+1} \boldsymbol{H}_{p+1,p} \boldsymbol{V}_p^T \tag{14}$$

be nonvanishing. Then $\boldsymbol{A}$ and $\boldsymbol{A}_p$ are spectrally equivalent for any $\rho \in \mathbb{R}$ and for all $\boldsymbol{z} \in \mathbb{K}_p$.

Proof. Define $\boldsymbol{z} = \boldsymbol{V}_p \boldsymbol{x}$ for any fixed $\boldsymbol{x} \in \mathbb{R}^p$. We note that for any $\boldsymbol{z} \in \mathbb{K}_p$ that

$$\boldsymbol{A}_p \boldsymbol{z} = \boldsymbol{V}_{p+1} \boldsymbol{H}_{p+1,p} \boldsymbol{V}_p^T \boldsymbol{V}_p \boldsymbol{x} = \boldsymbol{V}_{p+1} \boldsymbol{H}_{p+1,p} \boldsymbol{x} = \boldsymbol{A} \boldsymbol{z}.$$

In this case Assumption 1 is satisfied trivially for any $\rho \in \mathbb{R}$

$$\|(\boldsymbol{A} - \boldsymbol{A}_p)\boldsymbol{z}\| = 0 \leq \rho \|\boldsymbol{A}\boldsymbol{z}\|.$$

□

We remark that Theorem 3 only holds when the Arnoldi decomposition (12) is determined in exact arithmetic. In particular, the columns of the matrix $\boldsymbol{V}_{p+1}$ are required to be exactly orthonormal. This requirement is typically not satisfied when finite-precision floating-point arithmetic is used as in the computed examples reported in Section 4. We have not experienced any difficulty due to this issue. If the error $\boldsymbol{e}$ is of very small norm and, therefore, many steps of the Arnoldi process will be carried out, then it may be advantageous to implement this process with reorthogonalization. This has not been necessary for the computed examples reported in this paper.

Theorem 3 allows us to replace the BCCB matrix $\boldsymbol{C}$ from Section 2 by the matrix (14) determined by p steps of the Arnoldi process because it satisfies Assumption 1 for any $\rho \in \mathbb{R}$. To utilize the general algorithmic structure of the AIT method and its theory we will require $\rho \in (0, \frac{1}{2})$ as in [6].

To determine an apt regularization parameter at step k of the iterative process, we now introduce a result adapted from [4, Lemma 1].

Proposition 3. *Let $\boldsymbol{x}^{(k)}$ live in the Krylov subspace $\mathbb{K}_p$, and assume that the residual vector $\boldsymbol{r}^{(k)} = \boldsymbol{b}^\delta - \boldsymbol{A}\boldsymbol{x}^{(k)}$ and the matrix $\boldsymbol{A}$ are nonvanishing. Then by Theorem 3, the equation*

$$
\begin{aligned}
q^{(k)} \|\boldsymbol{r}^{(k)}\| &= \|\boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{h}^{(k)}\| \\
&= \|\boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{V}_p(\boldsymbol{H}_{p+1,p}^T \boldsymbol{H}_{p+1,p} + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T \boldsymbol{V}_{p+1}^T \boldsymbol{r}^{(k)}\|
\end{aligned}
\tag{15}
$$

has a unique solution for $0 < \alpha^{(k)} < \infty$, when $0 < q^{(k)} < 1$ is close enough to unity, and the elements of $\boldsymbol{r}^{(k)}$ satisfy a pair of conditions specified in the proof.

Proof. We argue first that $q^{(k)} > 0$. Assume that $q^{(k)} = 0$ and $\alpha^{(k)} > 0$. Then we have from (15)

$$\boldsymbol{r}^{(k)} = \boldsymbol{A}\boldsymbol{V}_p(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_{p+1}^T\boldsymbol{r}^{(k)}.$$

This equation only holds if $\alpha^{(k)}\boldsymbol{r}^{(k)} = \boldsymbol{0}$. Thus, we have that $q^{(k)} > 0$.

Before proceeding, we simplify (15). Using the Arnoldi decomposition (12), we may rewrite the right-hand side of (15) as

$$\|\boldsymbol{r}^{(k)} - \boldsymbol{V}_{p+1}\boldsymbol{H}_{p+1,p}(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_{p+1}^T\boldsymbol{r}^{(k)}\|.$$

Multiply the vector inside the norm by the matrix $\boldsymbol{V}_{p+1}^T$ from the left. This gives

$$\|\boldsymbol{V}_{p+1}^T\boldsymbol{r}^{(k)} - \boldsymbol{H}_{p+1,p}\boldsymbol{H}_{p+1,p}^T(\boldsymbol{H}_{p+1,p}\boldsymbol{H}_{p+1,p}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\boldsymbol{V}_{p+1}^T\boldsymbol{r}^{(k)}\|. \qquad (16)$$

Define the singular value decomposition (SVD) $\boldsymbol{H}_{p+1,p} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^T$, where the matrices $\boldsymbol{U} \in \mathbb{R}^{(p+1)\times(p+1)}$ and $\boldsymbol{W} \in \mathbb{R}^{p\times p}$ are orthogonal, and $\boldsymbol{\Sigma} \in \mathbb{R}^{(p+1)\times p}$ is diagonal; the diagonal entries are the singular values of $\boldsymbol{H}_{p+1,p}$ in non-increasing order. Using the SVD of $\boldsymbol{H}_{p+1,p}$ and defining $\tilde{\boldsymbol{r}}^{(k)} := \boldsymbol{V}_{p+1}^T\boldsymbol{r}^{(k)}$, the expression (16) simplifies to

$$\|\tilde{\boldsymbol{r}}^{(k)} - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\boldsymbol{U}^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\boldsymbol{U}^T + \alpha^{(k)}\boldsymbol{I})^{-1}\tilde{\boldsymbol{r}}^{(k)}\|.$$

Let $\bar{\boldsymbol{\Sigma}} := \boldsymbol{\Sigma}\boldsymbol{\Sigma}^T \in \mathbb{R}^{(p+1)\times(p+1)}$, whose first p diagonal elements $\bar{\sigma}_j$, $1 \leq j \leq p$, are the squared singular values of $\boldsymbol{H}_{p+1,p}$ and the last diagonal entry $\bar{\sigma}_{p+1}$ vanishes. Finally, defining $\hat{\boldsymbol{r}}^{(k)} := \boldsymbol{U}^T\tilde{\boldsymbol{r}}^{(k)}$ and using orthogonality, we arrive at the simplified form of (15):

$$q^{(k)}\|\hat{\boldsymbol{r}}^{(k)}\| = \|\hat{\boldsymbol{r}}^{(k)} - \bar{\boldsymbol{\Sigma}}(\bar{\boldsymbol{\Sigma}} + \alpha^{(k)}\boldsymbol{I})^{-1}\hat{\boldsymbol{r}}^{(k)}\|. \qquad (17)$$

Squaring both sides of (17) gives

$$\sum_{j=1}^p \left(1 - \frac{\bar{\sigma}_j}{\bar{\sigma}_j + \alpha^{(k)}}\right)^2 \left(\hat{r}_j^{(k)}\right)^2 + \left(\hat{r}_{p+1}^{(k)}\right)^2 = \left(q^{(k)}\right)^2 \sum_{j=1}^{p+1} \left(\hat{r}_j^{(k)}\right)^2, \qquad (18)$$

where $\hat{r}_j^{(k)}$ denotes the jth entry of the the vector $\hat{\boldsymbol{r}}^{(k)}$.

Introduce the function

$$\phi\left(\alpha^{(k)}\right) = \sum_{j=1}^p \left(1 - \frac{\bar{\sigma}_j}{\bar{\sigma}_j + \alpha^{(k)}}\right)^2 \left(\hat{r}_j^{(k)}\right)^2.$$

Then we have

$$\phi(0) = 0, \qquad \lim_{\alpha^{(k)}\to\infty} \phi\left(\alpha^{(k)}\right) = \sum_{j=1}^p \left(\hat{r}_j^{(k)}\right)^2,$$

and the derivative satisfies $\phi'(0) = 0$ and $\phi'(\alpha^{(k)}) > 0$ for $\alpha^{(k)} > 0$. Thus, the function $\phi(\alpha^{(k)})$ is monotonically increasing for $\alpha^{(k)} > 0$. It follows that equation (18) has a unique finite solution if and only if the following pair of conditions are satisfied

$$
\begin{cases}
(\hat{r}_{p+1}^{(k)})^2 < (q^{(k)})^2 \sum_{j=1}^{p+1} (\hat{r}_j^{(k)})^2, \\
\sum_{j=1}^{p+1} (\hat{r}_j^{(k)})^2 > (q^{(k)})^2 \sum_{j=1}^{p+1} (\hat{r}_j^{(k)})^2.
\end{cases}
\tag{19}
$$

The bottom inequality holds when $0 < q^{(k)} < 1$ and the top inequality holds when $q^{(k)}$ is large enough and p is sufficiently large (see Remark 2 below). $\qquad \square$

Remark 1. For the constants q and $q^{(k)}$ that occur in both Algorithms 1 and 2, we note that by construction $q \le q^{(k)}$. The conditions in Proposition 3 on $q^{(k)}$ can then be satisfied by choosing q large enough. In numerical experiments reported in [4,6], the value $q = 0.7$ worked well. We will use this value in the examples reported in the next section. The computed results are not sensitive to small changes in the q-value, however, we note that using a value very close to one results in an increased number of iterations, while a value very close to zero gives fewer iterations and worse quality of the computed reconstructions.

Remark 2. In computations, the top condition of (19) usually fails to be satisfied after a single Tikhonov step has been carried out in $\mathbb{K}_p$, because the first p entries of the residual vector $(\hat{\boldsymbol{r}}^{(k)})^2$ are small compared to the $(p+1)^{st}$ entry. When this situation occurs, a positive regularization parameter with which to carry out the next Tikhonov step in $\mathbb{K}_p$ cannot be found using the Levenberg-Marquardt iteration. To alleviate this issue, one need only expand the Krylov space by performing an additional step of the Arnoldi process. By doing so, the entries of $(\hat{\boldsymbol{r}}^{(k)})^2$ become larger in magnitude. This process is repeated as necessary until the top inequality of (19) holds. We demonstrate in our numerical experiments that it is typically unnecessary to carry out multiple expansions in early Tikhonov steps of the IAT method. We visualize and comment further on this issue in Section 4.

We now present the IAT algorithm.

Algorithm 2 (Iterated Arnoldi Tikhonov (IAT) method)**.** *For a given real $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ select $0 < \rho < 1/2$ and fix $q \in (2\rho, 1)$. Let $\delta > 0$ satisfy (5) and let $\boldsymbol{x}_1^{(0)}$ be an initial guess for $\boldsymbol{x}^\dagger \in \mathbb{R}^n$.*

$\boldsymbol{r}_1^{(0)} = \boldsymbol{b}^\delta - \boldsymbol{A}\boldsymbol{x}_1^{(0)};$
$\tau = \frac{1+2\rho}{1-2\rho};$
$\boldsymbol{V}_1 = \boldsymbol{b}^\delta / \|\boldsymbol{b}^\delta\|;$
for $p = 1, 2, \ldots$ **do**
   $k = 0;$
   $\boldsymbol{A}\boldsymbol{V}_p = \boldsymbol{V}_{p+1}\boldsymbol{H}_{p+1,p};$
   $\tau_p^{(k)} = \|\boldsymbol{r}_p^{(k)}\|/\delta;$
   $q_p^{(k)} = \max\{q, 2\rho + (1+\rho)/\tau_p^{(k)}\};$
   while $\left(\hat{r}_{p+1}^{(k)}\right)^2 < q_p^2 \sum_{j=1}^{p+1} \left(\hat{r}_j^{(k)}\right)^2$ **do**
      Determine $\alpha_p^{(k)}$ such that
      $q_p^2 \|\hat{\boldsymbol{r}}_p^{(k)}\|^2 = \|\hat{\boldsymbol{r}}_p^{(k)} - \bar{\boldsymbol{\Sigma}}(\bar{\boldsymbol{\Sigma}} + \alpha_p^{(k)}\boldsymbol{I})^{-1}\hat{\boldsymbol{r}}_p^{(k)}\|^2;$
      $\boldsymbol{h}_p^{(k)} = \boldsymbol{V}_p(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha_p^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_{p+1}^T\boldsymbol{r}_p^{(k)};$
      $\boldsymbol{x}_p^{(k+1)} = \boldsymbol{x}_p^{(k)} + \boldsymbol{h}_p^{(k)};$
      $\boldsymbol{r}_p^{(k+1)} = \boldsymbol{b}^\delta - \boldsymbol{A}\boldsymbol{x}_p^{(k+1)};$
      if $\|\boldsymbol{r}_p^{(k+1)}\| \le \tau\delta$ **then**
         Stop;
      end
      $k = k + 1;$
      $\tau_p^{(k)} = \|\boldsymbol{r}_p^{(k)}\|/\delta;$
      $q_p = \max\{q, 2\rho + (1+\rho)/\tau_p^{(k)}\};$
   end
   $\boldsymbol{x}_{p+1}^{(0)} := \boldsymbol{x}_p^{(k)}$
end

Remark 3. With regard to computational efficiency, we note that the computation of the $(k+1)^{st}$ residual vector $\boldsymbol{r}_p^{(k+1)}$ in Algorithm 2 does not require an additional matrix-vector product with $\boldsymbol{A}$. Instead, this vector may be computed as follows:

$$
\begin{aligned}
\boldsymbol{r}_p^{(k+1)} &= \boldsymbol{b}^\delta - \boldsymbol{A}\boldsymbol{x}_p^{(k+1)} \\
&= \boldsymbol{b}^\delta - \boldsymbol{A}\big(\boldsymbol{x}_p^{(k)} + \boldsymbol{h}_p^{(k)}\big) \\
&= \boldsymbol{r}_p^{(k)} - \boldsymbol{A}\boldsymbol{V}_p(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha_p^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_p^T\boldsymbol{r}_p^{(k)} \\
&= \Big(\boldsymbol{I} - \boldsymbol{V}_{p+1}\boldsymbol{H}_{p+1,p}(\boldsymbol{H}_{p+1,p}^T\boldsymbol{H}_{p+1,p} + \alpha_p^{(k)}\boldsymbol{I})^{-1}\boldsymbol{H}_{p+1,p}^T\boldsymbol{V}_p^T\Big)\boldsymbol{r}_p^{(k)},
\end{aligned}
$$

where in the last equality we have used the Arnoldi decomposition (12) to avoid the evaluation of further matrix-vector products with the matrix $\boldsymbol{A}$.

The following results are analogous to those at the end of Section 2; most of them can be found in [6].

Proposition 4. *For some $0 < \rho < 1/2$, let $\tau_* = (1 + \rho)/(1 - 2\rho)$. Then, by Theorem 3 if $\tau_p^{(k)} = \|\mathbf{r}_p^{(k)}\|/\delta > \tau_*$, it follows that*

$$\|\mathbf{r}_p^{(k)} - \mathbf{A}_p \mathbf{e}_p^{(k)}\| \leq \left(\rho + \frac{1+\rho}{\tau_p^{(k)}}\right)\|\mathbf{r}_p^{(k)}\| < (1-\rho)\|\mathbf{r}_p^{(k)}\|,$$

where $\mathbf{x}_p^{(k)}$ is the kth iterate generated by the outer p^{th} loop of Algorithm 2, $\mathbf{e}_p^{(k)} = \mathbf{x}_p^{\dagger} - \mathbf{x}_p^{(k)}$, and $\mathbf{r}_p^{(k)} = \mathbf{b}^{\delta} - \mathbf{A}\mathbf{x}_p^{(k)}$.

Proposition 5. *Under the results and assumptions of Theorem 3, the norm of the iteration error $\mathbf{e}_p^{(k)} = \mathbf{x}_p^{\dagger} - \mathbf{x}_p^{(k)}$ in the subspace $\mathbb{K}_p$ decreases monotonically for $k = 0, 1, 2, \ldots, k_p^{\delta} - 1$:*

$$\|\mathbf{e}_p^{(k)}\|^2 - \|\mathbf{e}_p^{(k+1)}\|^2 \geq 2\rho\|(\mathbf{A}_p \mathbf{A}_p^T + \alpha_p^{(k)}\mathbf{I})^{-1}\|\|\mathbf{r}_p^{(k)}\| \tag{20}$$

provided that $\|\mathbf{r}_p^{(k)}\| > \tau\delta$.

Corollary 2. *Under the assumptions and notation of Proposition 5, and for sufficiently large p, there exists a first iterate $\mathbf{x}_p^{(k)}$ with $k = k_p^{\delta}$ such that*

$$\|\mathbf{e}_p^{(0)}\|^2 \geq 2\rho \sum_{k=0}^{k_p^{\delta}-1} \|(\mathbf{A}_p \mathbf{A}_p^T + \alpha_p^{(k)}\mathbf{I})^{-1}\|\|\mathbf{r}_p^{(k)}\| \geq c \sum_{k=0}^{k_p^{\delta}-1} \|\mathbf{r}_p^{(k)}\|^2 \tag{21}$$

for some constant $c > 0$ that only depends on the parameter ρ and the value of q used in Algorithm 2.

Theorem 4. *Assume that $\delta = 0$ and that $\mathbf{x}^{(0)}$ is not a solution of (2). Then for sufficiently large p, the sequence of iterates $\mathbf{x}_p^{(k)}$, $k = 1, 2, \ldots$, generated by the outer p^{th} loop of Algorithm 2 converges as $k \to \infty$ to a solution of (1) that is closest to $\mathbf{x}_p^{(0)}$.*

Proof. If $\delta = 0$ then the stopping criterion (8) can only be satisfied with $k = k_p^{\delta}$ for a solution $\mathbf{x}_p^{(k)}$ of (1). Here, if $k > 0$ then $\mathbf{h}_p^{(k-1)}$ must coincide with $\mathbf{e}_p^{(k-1)}$ up to an element in the null space of $\mathbf{A}$, that is, in the null space of $\mathbf{A}_p$ by Theorem 3. Accordingly, it follows from (10) with the pth Arnoldi approximation of $\mathbf{A}$ substituted and Proposition 4 that

$$q_p^{(k-1)}\|\mathbf{r}_p^{(k-1)}\| = \|\mathbf{r}_p^{(k-1)} - \mathbf{A}_p \mathbf{h}_p^{(k-1)}\| = \|\mathbf{r}_p^{(k-1)} - \mathbf{A}_p \mathbf{e}_p^{(k-1)}\|$$
$$\leq \left(\rho + \frac{1+\rho}{\tau_p^{(k-1)}}\right)\|\mathbf{r}_p^{(k-1)}\|.$$

However, this contradicts the definition of $q_p^{(k-1)} := \max\{q, 2\rho + (1+\rho)/\tau_p^{(k-1)}\}$ that is used in Algorithms 1 and 2. Thus, the iteration does not terminate for

11

exact data unless $\boldsymbol{x}_p^{(0)}$ is already a solution of (1). An analogous argument to that in [6] can be used to show that for p sufficiently large each sequence $\{\boldsymbol{x}_p^{(k)}\}$ is a Cauchy sequence that converges as $k \to \infty$ to a solution of (1) that is closest to $\boldsymbol{x}_p^{(0)}$. $\qquad\qquad\square$

Theorem 5. *By Theorem 3 and for some $0 < \rho < 1/2$, let $\delta \mapsto \boldsymbol{b}^\delta$ be a function such that (5) holds for all $\delta > 0$. For fixed parameters τ and q, let $\boldsymbol{x}_p^\delta$ denote the approximate solution generated by the outer p^{th} loop of Algorithm 2. Then, as $\delta \to 0$ and for sufficiently large p, $\boldsymbol{x}_p^\delta$ converges to a solution of (1) that is closest to $\boldsymbol{x}_p^{(0)}$.*

A variant of Theorem 5 has been used previously in [7, 10]. Necessary results include the monotonicity property of the iterates and their continuous dependence on $\boldsymbol{b}^\delta$.

4 Numerical Results

We illustrate the properties of the IAT method and compare its performance to the AIT and range restricted generalized minimum residual (rrGMRES) when applied to image deblurring. The latter method is described in [17]. Both spatially variant and invariant blur are considered. Both kinds of blur can be modeled by a Fredholm integral equation of the first kind,

$$\int_{\boldsymbol{\Omega}} \boldsymbol{\kappa}(u, s, v, t)\, \boldsymbol{x}(u, v)\, \mathrm{d}u\, \mathrm{d}v = \boldsymbol{b}(s, t) \qquad (s, t) \in \boldsymbol{\Omega}, \qquad (22)$$

where $\boldsymbol{b}$ represents the blurred image, $\boldsymbol{\kappa}$ the point spread function (PSF), and $\boldsymbol{\Omega}$ is the domain of the exact image represented by $\boldsymbol{x}$. When the blur is spatially invariant, the kernel $\boldsymbol{\kappa}$ in the integral equation above is given by $\boldsymbol{\kappa}(u, s, v, t) = \boldsymbol{\kappa}(u - s, v - t)$. Upon discretization of (22) we have a problem of the form (1), where the structure of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ depends on the the properties of $\boldsymbol{\kappa}$ and on the boundary conditions (BC); see [1, 13]. For common BCs including zero, periodic, and reflexive, the matrix $\boldsymbol{A}$ may be decomposed as

$$\boldsymbol{A} = \boldsymbol{T} + \boldsymbol{R} + \boldsymbol{E},$$

where $\boldsymbol{T}$ is a BTTB matrix, $\boldsymbol{R}$ is a matrix of low-rank, and $\boldsymbol{E}$ is a matrix of small norm. The approximation of $\boldsymbol{A}$ by a BCCB matrix is attractive, because the structure of $\boldsymbol{C}$ allows diagonalization by the discrete Fourier transform. Therefore, the vector $(\boldsymbol{C}\boldsymbol{C}^T + \alpha_p \boldsymbol{I})^{-1} \boldsymbol{r}_p$ can be computed in $O(n \log n)$ floating point operations using a fast Fourier transform algorithm.

To evaluate the quality of the reconstructed images, we compute the relative reconstructive error (RRE), which is defined by

$$\mathrm{RRE}(\boldsymbol{x}_p^{(k)}) = \frac{\|\boldsymbol{x}_p^{(k)} - \boldsymbol{x}^\dagger\|}{\|\boldsymbol{x}^\dagger\|},$$
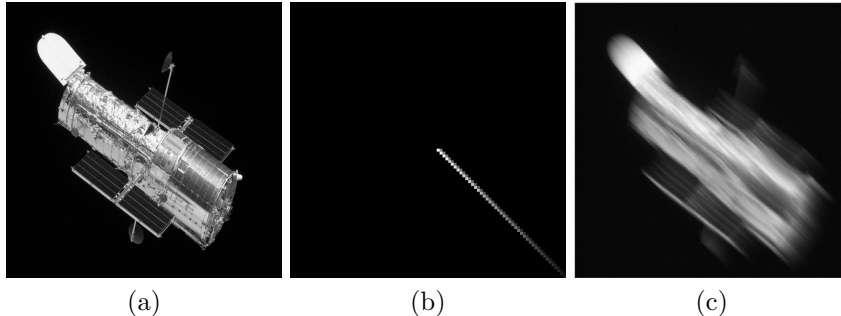
Figure 1: *Hubble* test case: (a) true image (512 x 512 pixels), (b) enlarged PSF from IR Tools, (c) blurred and 3% noised image (512 x 512 pixels).

where $\boldsymbol{x}_p^{(k)}$ denotes the computed approximate solution obtained when the discrepancy principle (8) is satisfied, and $\boldsymbol{x}^{\dagger}$ represents the exact solution. We will refer to $\boldsymbol{x}_p^{(k)}$ as the solution at breakout. Because of the "double" iterative structure of our IAT method, it is necessary to differentiate between the Tikhonov steps taken and the number of Arnoldi iterations computed. The Tikhonov steps of the IAT method are measured by the number of successive residual vectors computed, and the Arnoldi iterations are tracked by the iteration counter, p, in the algorithm. We will see that the number of Arnoldi iterations required to achieve breakout according to the discrepancy principle is larger than or equal to the number of Tikhonov steps. The number of Arnoldi iterations of the IAT method are compared to the number of iterations with the AIT and rrGMRES methods. For further comparative studies of the AIT and closely related methods, we refer the interested reader to [2].

Our computational work was carried out in the IR Tools toolbox environment by Gazzola et al. [9] using MATLAB R2020b on a MacBook Pro laptop computer running MacOS Catalina with an i5 Dual-Core Intel processor @2.7 GHz and 8 GB of RAM. The computations were carried out with about 15 significant decimal digits. Standard images from MATLAB's image processing toolbox as well as constructed PSFs were used. For proper comparison between the IAT and AIT methods, we set $\rho = 10^{-3}$ and $q = 0.7$ in our examples, as was done in [6] and [4].

Our first two examples below compare the three methods for two different spatially invariant non-symmetric motion blurs for images that naturally require zero and reflexive BCs, respectively. The last example considers a spatially variant rotational motion blur. The first example, *hubble*, also emphasizes the reconstructive process of the IAT method for one level of noise by considering the solution basis and computed residuals.

Hubble We begin by considering a non-symmetric problem using a PSF from IR Tools that models motion blur; see Figure 1. Here, we chose the most severe option of this PSF from IR Tools, where a high level of severity corresponds to a

faster rate of decay of the singular values of the blurring matrix to zero. Because the image is black near the border, we impose zero BCs. We contaminate the blurred image by Gaussian noise with noise levels 3%, 1%, and 0.1%. We found that the best initial vector choice for the IAT method was the zero vector (i.e., $x_0 = 0$). Interestingly, the AIT method also required fewer iterations and displayed lower RRE values for solutions with $x_0 = 0$ as its starting vector for all noise levels. This is in contrast with the examples considered in [6] and [4]. We comment on this further below.

Before comparing the computational aspects and the accuracy of the computed reconstructions of the methods considered, we discuss the manner by which the IAT method arrives at its reconstructions. By viewing the 19 reshaped basis vectors in Figure 2 that define the solution subspace for the *hubble* problem with 3% noise, we can see that the initial basis vectors of the solution subspace represent low frequencies, while later basis vectors represent higher frequencies. This can be observed by the lack of entry-to-entry pixel change in the early reshaped basis vectors. The change in the progression of the reshaped iterative residual vectors in Figure 3 for the same problem also illustrates this.

As mentioned in Remark 2, not every Tikhonov step of the IAT method requires more than one Arnoldi iteration. We illustrate this in Figure 3 by denoting below each reshaped residual vector the number of sequential basis vectors from Figure 2 that were needed to determine a unique positive regularization parameter and thus achieve the next Tikhonov step. We note that the first residual vector is determined at the start of the algorithm using A. In our experience, later Tikhonov steps often require more Arnoldi iterations. We see this from the later reshaped residual vectors in Figure 3 by noting the larger number of required Arnoldi iterations with A. In an image deconvolution setting this point of view makes intuitive sense as more high frequency basis information is needed to resolve finer details of an image.

The iteration requirements and breakout RRE values for the *hubble* example for all three investigated noise levels are shown in Table 1. Figure 4 displays the final reconstructions at breakout for the corresponding noise levels. We note that the IAT method produces the lowest RRE values amongst the three methods for all levels of noise. Furthermore, the number of Arnoldi iterations necessary for the IAT method to achieve breakout is smaller than the number of iterations required to breakout for rrGMRES in all cases.

The AIT method did not perform well in this example compared with the other two methods. This is evident from the quick breakouts in the 3% and 1% noise cases, and the poor RRE values compared to the other methods. We hasten to add that the initial vector $x_0 = A^T b^\delta$ used in [6] and [4] provided worse results in all noise cases for the AIT method, and therefore the initial vector played no major role. This result was particularly surprising given that in the aforementioned studies it performed well for problems with higher noise levels. We surmise that the chosen severity of this PSF from IR Tools contributed significantly to this outcome.

The iterative evolution of the RREs and residual plots for the methods considered are shown in Figure 5 for all noise levels. In the left-hand column
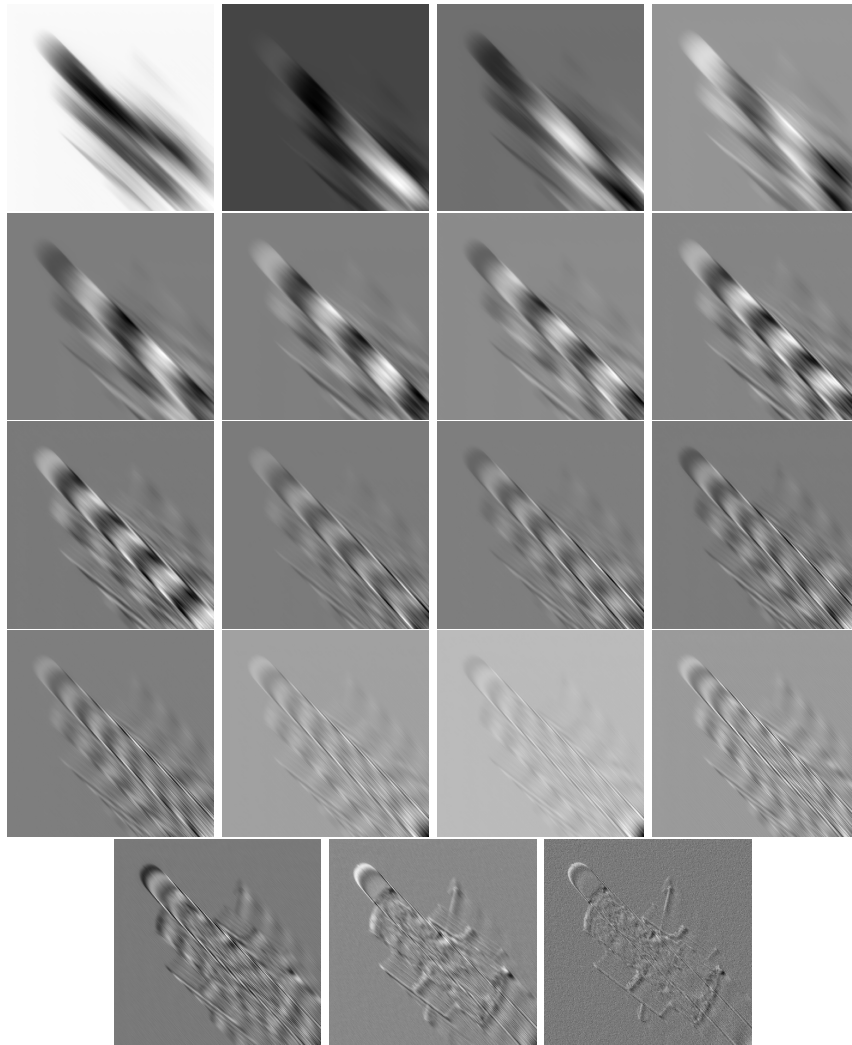
14

Figure 2: The 19 reshaped basis vectors that define the solution space for the reconstructed image in the *hubble* example with 3% noise. The top left entry corresponds to the first reshaped basis vector and the bottom right the last.
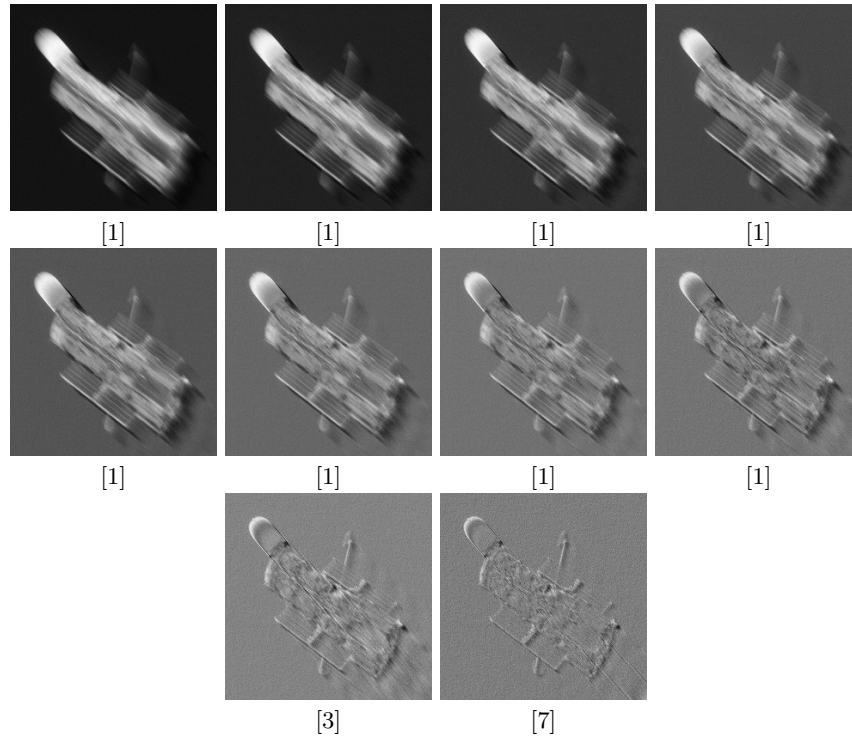
Figure 3: The 10 reshaped residual vectors resulting during the IAT method in the *hubble* example with 3% noise. The top left entry is the first residual after the first iterative improvement has been made and the bottom right is the last when breakout occurs. Bracketed enumerations below each pane corresponds to the number of Arnoldi iterations needed for the corresponding residual to be computed. The numbers in brackets show the number of Arnoldi steps p in Algorithm **??**. Images with the same p-values corresponds to inceasing k-values in the algorithm (from left to right).
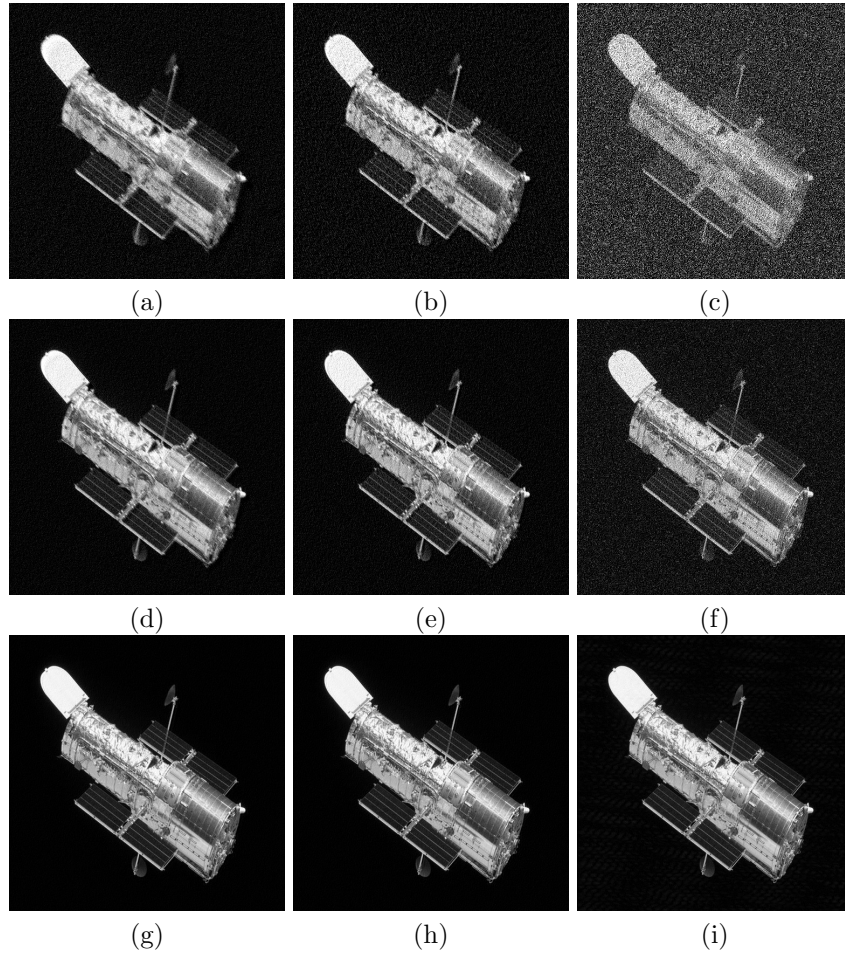
Figure 4: *Hubble* test case reconstructions: The first row reports the 3% cases for IAT (left), rrGMRES (middle), and AIT (right) methods. The second row reports the 1% cases, and the third row reports the 0.1% cases in the same method ordering as the first row.

Noise Level	Method	Iteration Count	Breakout RRE
3%	IAT	10[18]	0.3488
	rrGMRES	58	0.4073
	AIT	2	1.7181
1%	IAT	13[32]	0.2133
	rrGMRES	80	0.2427
	AIT	2	0.5745
0.1%	IAT	20[91]	0.0561
	rrGMRES	179	0.0615
	AIT	15	0.1436

Table 1: Tabular information for the *hubble* example for the three methods at each level of noise investigated. Bracketed iterative information for the IAT corresponds to the number of Arnoldi expansions needed. Breakout RRE values presented correspond to the pth iterative solution when the discrepancy principle is satisfied.

of the relative residual plots, we note the quick termination of the AIT method after two iterations for 3% and 1% noise levels. For 0.1% noise, the AIT method fares better as it does not immediately terminate after the first iteration. In general, these plots provide visualization of the rapid relative residual decay of the IAT and AIT methods compared to the rrGMRES method. The RRE plots in the right-hand column provide a visual trajectory of the RRE values of each algorithm. We note that the IAT and rrGMRES methods demonstrate semiconvergent behavior in the 3% and 1% cases as both methods have their RRE values increase before breakout occurs. However, the relative quickness of termination of the IAT method stymies this effect compared with the rrGMRES method.

Brezinski We next consider the *Brezinski* image blurred with constructed non-symmetric motion blur, where the PSF diagonal elements are computed by the Softmax function, $\mathcal{F}$, whose entries are given by

$$\mathcal{F}(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{n} e^{y_i}}.$$

Here, the y_i come from a linearly spaced vector, $\boldsymbol{y}$, whose first and last entries are 0 and 1, respectively. The off-diagonal entries are defined to be zero. We note that the singular values associated with this problem decay less rapidly compared to the *hubble* example. We impose reflexive boundary conditions and introduce 0.1% Gaussian noise; see Figure 6. Similarly to the *hubble* example, the selection of $\boldsymbol{x}_0 = \boldsymbol{A}^T \boldsymbol{b}^\delta$ was found to cause the magnitude of the normalized starting residual $\boldsymbol{r}_0$ to be large, which resulted in a poor solution. Because of this, we chose our initial vector for the IAT algorithm to again be $\boldsymbol{x}_0 = \boldsymbol{0}$. The
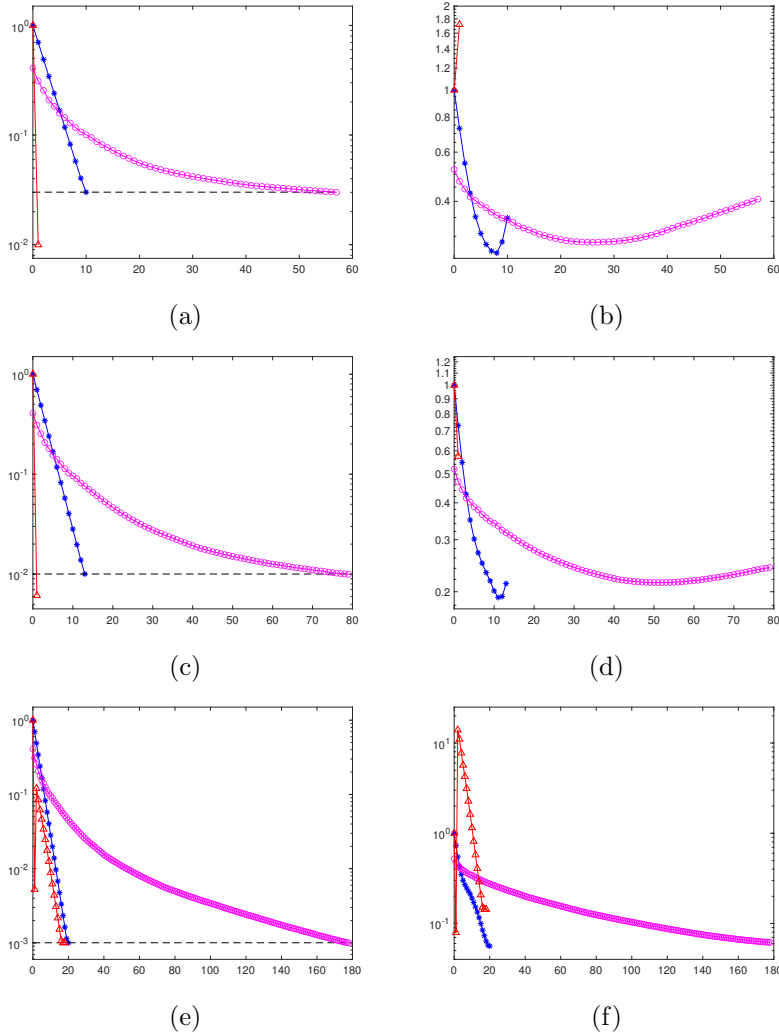
Figure 5: *Hubble* test case: left-hand column reports the relative residual plots and the right-hand column the RRE plots plotted against the iteration number. The top to bottom rows correspond to the 3%, 1%, and 0.1% test cases, respectively. The IAT method is represented by blue stars, the rrGMRES by magenta circles, and the AIT by red triangles. The black dashed horizontal lines in the relative residual plots correspond to the breakout level according to the discrepancy principle.
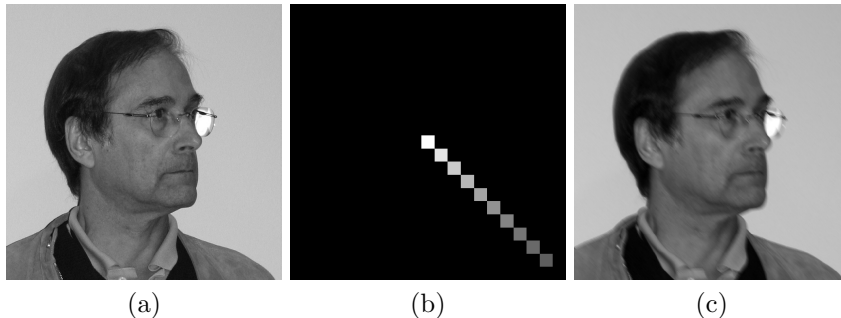
<div align="center">(a)   (b)   (c)</div>

Figure 6: *Brezinski* test case: (a) true image (700 x 700 pixels), (b) PSF (21 x 21 pixels), (c) blurred and 0.1% noised image (700 x 700 pixels).

Method	Iteration Count	Breakout RRE
IAT	20[23]	0.0103
rrGMRES	42	0.0111
AIT	23	0.0111

Table 2: Tabular information for the *Brezinski* example for the three methods investigated. Bracketed iterative information for the IAT corresponds to the number of Arnoldi expansions needed. Breakout RRE values presented correspond to the pth iterative solution when the discrepancy principle is satisfied.

AIT algorithm also performed best when the starting vector was the zero vector.

The iteration requirements and breakout RRE values for the *Brezinski* example are tabulated in Table 2. Figure 7 displays the final image reconstructions at breakout for each of the methods. We note a couple items: the first being that the IAT method had the lowest RRE at breakout. The second is that the number of iterations required to achieve breakout for the IAT method is smaller than for the other two method. The iterative evolution of the RREs and relative residual plots are provided in Figure 8. The left-hand side plot displays the relative residual progression and the right-hand side plot displays the RRE values for each iteration of the three methods.

Satellite In our last example we consider a spatially variant rotational motion blurring problem described in [14] and implemented in the IR Tools toolbox. The singular values of the blurring matrix decay to zero more slowly than in the *hubble* example. Figure 9 displays both the true image as well as the blurred image with 3% imposed noise. Here, again, we impose zero BC's. The blurred image is created by taking an average of a series of images, each of which is rotated slightly with respect to the center of the image (see Figure 1 in [14]). The IAT method uses $x_0 = 0$ as its starting vector. We note that the blurring matrix for this spatially variant motion blur problem does not have BTTB

(a)   (b)   (c)
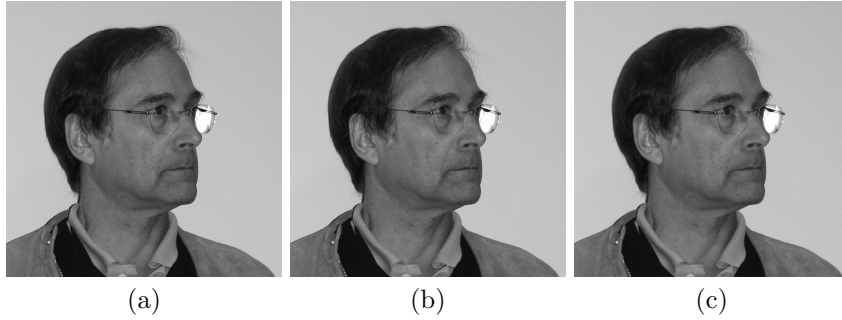
Figure 7: *Brezinski* test case with 0.1% noise: (a) IAT, (b) rrGMRES, and (c) AIT method reconstructions at breakout.

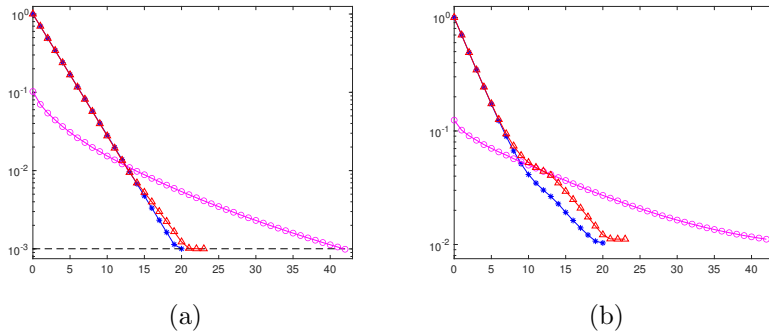(a)   (b)

Figure 8: *Brezinski* test case: (a) relative residual and (b) RRE plots plotted against the iteration number. The IAT method is represented by blue stars, the rrGMRES by magenta circles, and the AIT by red triangles. The black dashed horizontal line in the relative residual plot corresponds to the breakout level according to the discrepancy principle.
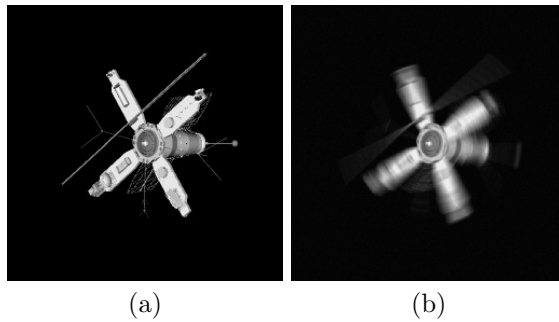

(a)   (b)

Figure 9: *Satellite* test case: (a) true image (256 x 256 pixels) and (b) blurred and 3% noised image (256 x 256 pixels).

Method	Iteration Count	Breakout RRE
IAT	10[183]	0.3268
rrGMRES	300*	0.3462

Table 3: Tabular information for the *satellite* example for the two methods investigated. Bracketed iterative information for the IAT corresponds to the number of Arnoldi expansions needed. Breakout RRE values presented correspond to the pth iterative solution when the discrepancy principle is satisfied. The asterisk, *, denotes the manual termination of the method due to prolonged runtime.
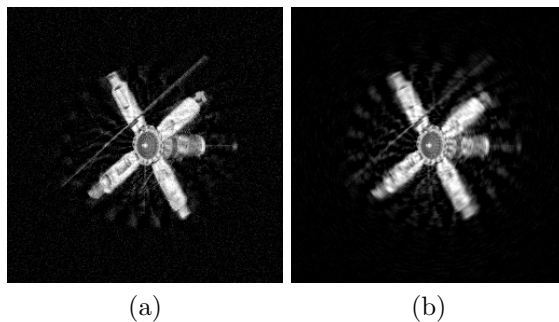

(a)   (b)

Figure 10: *Satellite* test case with 3% noise: (a) IAT and (b) rrGMRES method reconstructions at breakout.

substructure. As such, we only compare the IAT and rrGMRES methods since the AIT algorithm cannot be applied in this situation.

Iteration requirements and RRE breakout values for the IAT and rrGMRES methods are provided in Table 3. The final image reconstructions at breakout are shown in Figure 10. We found similarly to the *hubble* example that many more Arnoldi iterations were required to be able to determine unique regularization parameters for later Tikhonov steps than for earlier ones. The number of Arnoldi iterations were largest for the final two Tikhonov steps. The number of iterations, while large compared to the *hubble* and *Brezinski* examples, did result in successful breakout. The rrGMRES method was manually terminated at the 300th iteration due to long run-time. In the left-hand side pane of Figure 11, one may estimate that the rrGMRES method might have required an additional 100 iterations to achieve breakout. The RRE progression of each method is also shown in the right-hand pane of Figure 11. We especially note the erratic behavior of the rrGMRES progression. Finally, we note in Figure 10 the pictorial differences between the IAT and the rrGMRES method reconstructions. While there is some minor background noise in the IAT reconstruction, the wings of the satellite are much better defined than in the reconstruction obtained by rrGMRES.
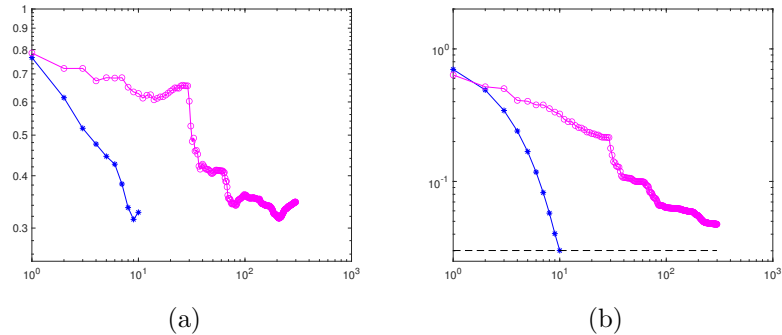
Figure 11: *Satellite* test case: (a) relative residual and (b) RRE plots plotted against the iteration number. The IAT method is represented by blue stars and the rrGMRES by magenta circles. The black dashed horizontal line in the relative residual plot corresponds to the breakout level according to the discrepancy principle.

5 Conclusions

We have introduced a transpose-free preconditioned iterated Tikhonov method using the Arnoldi iteration which we refer to as the *iterated Arnoldi-Tikhonov* (IAT) method based on the AIT method formulated by Donatelli and Hanke in [6]. In Section 3, we presented the IAT algorithm and showed that it is a regularization method provided the original matrix A is approximated sufficiently accurately by the Arnoldi process. Section 4 showcased the effectiveness of the IAT method at reconstructing images that have been contaminated by blurs of various severities and Gaussian noise of different levels.

Acknowledgments

The authors would like to thank a referee for comments. They also would like to thank Marco Donatelli for precious discussions that helped improve the quality of this paper. A.B. is a member of the GNCS group of INdAM.

Competing interests

The authors declare that they have no conflict of interest.

Data availability

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

References

[1] Berisha, S., Nagy, J.G.: Iterative methods for image restoration. Tech. rep., Department of Mathematics and Computer Science, Emory University (2012). http://www.mathcs.emory.edu/ nagy/RestoreTools/IR.pdf

[2] Buccini, A.: Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term. Applied Numerical Mathematics **116**, 64–81 (2017)

[3] Buccini, A., Donatelli, M., Reichel, L.: Iterated Tikhonov with general penalty term. Numerical Linear Algebra and Applications **24**, e2089 (2017)

[4] Buccini, A., Donatelli, M., Reichel, L., Zhang, W.: A new non-stationary preconditioned iterative method for linear discrete ill-posed problems. Numerical Linear Algebra with Applications **28**(2), e2353 (2021)

[5] Chan, T.F., Jackson, K.R.: Nonlinearly preconditioned krylov subspace methods for discrete newton algorithms. SIAM Journal on Scietific and Statistical Computing **5**, 533–542 (1984)

[6] Donatelli, M., Hanke, M.: Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring. Inverse Problems **29**(9), 095,008 (2013)

[7] Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Doordrecht (1996)

[8] Fenu, C., Reichel, L., Rodriguez, G.: GCV for Tikhonov regularization via global Golub-Kahan decomposition. Numerical Linear Algebra with Applications **23**(3), 467–484 (2016)

[9] Gazzola, S., Hansen, P.C., Nagy, J.G.: IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numerical Algorithms **81**, 773–811 (2019)

[10] Hanke, M.: A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. Inverse problems **13**(1), 79 (1997)

[11] Hanke, M., Groetsch, C.W.: Nonstationary iterated Tikhonov regularization. Journal of Optimization Theory and Applications **98**(1), 37–53 (1998)

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

[13] Hansen, P.C., Nagy, J., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)

[14] Hansen, P.C., Nagy, J.G., Tigkos, K.: Rotational image deblurring with sparse matrices. BIT Numerical Mathematics **54**, 649–671 (2014)

[15] Kindermann, S.: Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems. Electronic Transactions on Numerical Analysis **38**, 233–257 (2011)

[16] Kindermann, S., Raik, K.: A simplified L-curve as error estimator. Electronic Transactions on Numerical Analysis **53**, 213–238 (2020)

[17] Neuman, A., Reichel, L., Sadok, H.: Algorithms for range restricted iterative methods for linear discrete ill-posed problems. Numerical Algorithms **59**(2), 325–331 (2012)

[18] Reichel, L., Rodriguez, G.: Old and new parameter choice rules for discrete ill-posed problems. Numerical Algorithms **63**, 65–87 (2013)

[19] Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd. ed. SIAM, Philadelphia (2003)