Numer Algor (2019) 81:33–55
https://doi.org/10.1007/s11075-018-0534-y

ORIGINAL PAPER

On the choice of subspace for large-scale Tikhonov regularization problems in general form

Guangxin Huang[1] · Lothar Reichel[2] · Feng Yin[3]

Received: 27 February 2018 / Accepted: 19 April 2018 / Published online: 3 May 2018

Abstract Many applications in science and engineering require the solution of large linear discrete ill-posed problems. The matrices that define these problems are very ill-conditioned and possibly numerically singular, and the right-hand sides, which represent the measured data, typically are contaminated by measurement error. Straightforward solution of these problems generally is not meaningful due to severe error propagation. Tikhonov regularization seeks to alleviate this difficulty by replacing the given linear discrete ill-posed problem by a penalized least-squares problem, whose solution is less sensitive to the error in the right-hand side. The penalty term is determined by a regularization matrix. A suitable choice of this matrix may result in a computed solution of higher quality than when the regularization matrix is the identity. Two iterative solution methods based on the global Arnoldi decomposition method have been proposed in the literature for the solution of large-scale penalized least-squares problems that stem from Tikhonov regularization. In one of these, the

In Memory of Sebastiano Seatzu.

✉ Lothar Reichel
reichel@math.kent.edu

Guangxin Huang
huangx@cdut.edu.cn

Feng Yin
fyin@suse.edu.cn

[1] Geomathematics Key Laboratory of Sichuan, Department of Information and Computing Science, Chengdu University of Technology, Chengdu, 610059, People's Republic of China

[2] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

[3] School of Mathematics and Statistics, Sichuan University of Science and Engineering, Zigong, 643000, People's Republic of China

regularization matrix influences the choice of the solution subspace; in the other one, it does not. This paper compares these approaches both with respect to the quality of the computed solution and computing time.

Keywords Linear discrete ill-posed problems · Global Arnoldi process · Matrix Krylov subspace · Standard Tikhonov problems · Regularization

Mathematics Subject Classification (2010) 65F10 · 65F22 · 65R30

1 Introduction

We consider the approximate solution of large-scale minimization problems of the form

$$\min_{X\in\mathbb{R}^{m\times n}} \|G - \sum_{i=1}^{p} A_i\, X\, B_i\|_F, \tag{1.1}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. At least one of the matrices $A_i \in \mathbb{R}^{m\times m}$ and $B_i \in \mathbb{R}^{n\times n}$ of each pair (A_i, B_i) is large and of ill-determined rank, i.e., its singular values decay to zero with increasing index without a significant gap. Minimization problems of the form (1.1) with matrices A_i and B_i of this kind are commonly referred to as discrete ill-posed problems. For notational simplicity, we assume the matrices A_i and B_i to be square; however, this restriction easily can be removed. The matrix $G \in \mathbb{R}^{m\times n}$ in (1.1) represents available error-contaminated data, such as a blurred and noise-contaminated image.

Let the matrix $E \in \mathbb{R}^{m\times n}$ represent the unknown noise-contamination of G. Thus,

$$G = \widehat{G} + E, \tag{1.2}$$

where $\widehat{G} \in \mathbb{R}^{m\times n}$ denotes the unknown error-free matrix associated with G. We assume the unavailable linear system of equations

$$\sum_{i=1}^{p} A_i\, X\, B_i = \widehat{G} \tag{1.3}$$

to be consistent and let $\widehat{X} \in \mathbb{R}^{m\times n}$ denote the solution of (1.3) of minimal Frobenius norm. We would like to determine an approximation of $\widehat{X}$ by computing a suitable approximate solution of (1.1). The consistency requirement may be dispensed with; see below.

The solution of (1.1) of minimal Frobenius norm, typically, is not a meaningful approximation of $\widehat{X}$ due to severe propagation of the error E in G into the computed solution. This depends on that at least one of the matrices in each pair (A_i, B_i) has some "tiny" positive singular values. Therefore, one often replaces (1.1) by a nearby problem, whose solution is less sensitive to the error in G than the solution of (1.1), and computes the solution of the modified problem so obtained. This replacement is known as regularization. The possibly most popular regularization method is due

to Tikhonov. This method replaces (1.1) by a penalized least-squares problem of the form

$$\min_{X\in\mathbb{R}^{m\times n}} \left\{ \left\| \sum_{i=1}^{p} A_i\, X\, B_i - G \right\|_F^2 + \mu \left\| \sum_{j=1}^{q} L_j^{(1)}\, X\, L_j^{(2)} \right\|_F^2 \right\}, \tag{1.4}$$

where the matrices $L_j^{(1)} \in \mathbb{R}^{s\times m}$ and $L_j^{(2)} \in \mathbb{R}^{n\times t}$ are referred to as regularization matrices and $\mu > 0$ is a regularization parameter. The purpose of this parameter is to balance the influence of the first term (the *fidelity term*) and the second term (the *regularization term*) on the solution X_μ of (1.4). We will let the matrices $L_j^{(i)}$ be discretizations of differential operators in one space dimension. Then, generally, the minimization problem (1.4) has a unique solution for any $\mu > 0$; see below.

We note that the choice of the matrices $L_j^{(i)}$ is important for the quality of the computed approximation X_μ of the desired solution $\widehat{X}$. The $L_j^{(i)}$ should be chosen so that important features of $\widehat{X}$ or $\widehat{X}^T$ are in the null spaces of the matrices $L_j^{(1)}$ or $(L_j^{(2)})^T$, respectively, and, therefore, are not damped when solving (1.4). Here the superscript ^{T} denotes transposition. Several techniques for constructing regularization matrices with desirable properties are described in the literature; see, e.g., [7, 9–11, 20, 28, 29]. Minimization problems (1.4) with at least one of the matrices $L_j^{(i)}$ different from the identity matrix are said to be in *general form*; when all the $L_j^{(i)}$ are identity matrices, the problem (1.4) is said to be in *standard form*.

The quality of the solution X_μ of (1.4) also depends on the value of the regularization parameter μ. A too large value gives an over-smoothed solution X_μ that lacks details that the desired solution $\widehat{X}$ may have; a too small value yields a solution X_μ that is contaminated by unnecessarily much propagated error. We will use the discrepancy principle [13, 16] to determine μ in the computed examples of Section 4. Its application requires (1.3) to be consistent. Many other techniques for determining a suitable value of μ, including the L-curve criterion and generalized cross validation, also can be used; see [2, 4–6, 17, 24, 30, 33] for discussion and illustrations. The latter methods do not demand (1.3) to be consistent. All methods mentioned for determining a suitable μ value require that several problems (1.4) be solved approximately for different μ values.

Let the vector $\boldsymbol{x} = \mathrm{vec}(X) \in \mathbb{R}^{mn}$ be obtained by stacking the columns of the matrix X in order. The Kronecker product $\otimes$ of two matrices $C \in \mathbb{R}^{m_c\times n_c}$ and $D \in \mathbb{R}^{m_d\times n_d}$ is defined as the $(m_c m_d)\times(n_c n_d)$ matrix $C\otimes D = [c_{ij}D]$; see, e.g., [19]. We can express the minimization problem (1.4) in the form

$$\min_{x\in\mathbb{R}^{mn}} \left\{ \|K\boldsymbol{x} - \boldsymbol{g}\|_2^2 + \mu\, \|L\boldsymbol{x}\|_2^2 \right\}, \tag{1.5}$$

with

$$K = \sum_{i=1}^{p} B_i^T \otimes A_i, \quad L = \sum_{j=1}^{q} \left(L_j^{(2)}\right)^T \otimes L_j^{(1)}, \quad \boldsymbol{g} = \mathrm{vec}(G),$$

where $\|\cdot\|_2$ stands for the Euclidean vector norm. The solutions X_μ of (1.4) and $\boldsymbol{x}_\mu$ of (1.5) are related by $\boldsymbol{x}_\mu = \mathrm{vec}(X_\mu)$. The minimization problem (1.5) has a unique

solution for any $\mu > 0$ if and only if $\mathcal{N}(K) \cap \mathcal{N}(L) = \{0\}$, where $\mathcal{N}(M)$ denotes the null space of the matrix M. This requirement implies an analogous condition for the unique solvability of (1.4).

Many iterative methods are available for the solution of problems of the form (1.5); see, e.g., [15, 18, 23, 25, 27, 31, 32, 34] and references therein. However, we will see in Section 4 that it is beneficial to solve the equivalent problem (1.4) instead. We are therefore interested in studying iterative solution methods for the latter problem. Several block iterative methods for the solution of minimization problems of the form (1.4) are described in [1, 12, 21, 22]. We are particularly interested in the application of the global Arnoldi method and variants thereof, which have been applied to image restoration in [2, 3, 11]. Block Arnoldi-type methods have the advantage over methods based on block Golub–Kahan bidiagonalization that they only require the evaluation of one matrix-block-vector product (instead of two) per iteration, and they only demand matrix-block-vector product evaluations with the matrix A (instead of with A and A^T).

The remainder of this paper is organized as follows. Section 2 reviews two kinds of global Arnoldi methods and discusses some properties of these methods. Application of these methods to the solution of the minimization problem (1.4) is described in Section 3. Computed examples with application to image restoration can be found in Section 4, and Section 5 contains concluding remarks.

2 Global iteration schemes based on the Arnoldi method

This section reviews the global Arnoldi method by Jbilou et al. [21, 22] and the generalized global Arnoldi method due to Bouhamidi et al. [3]. We will need the following definitions: The Frobenius inner product is given by

$$\langle V, W \rangle_F = \text{trace}(V^T W)$$

for matrices $V, W \in \mathbb{R}^{m\times n}$. Matrices V and W that satisfy $\langle V, W \rangle_F = 0$ are said to be F-orthogonal. Moreover, a sequence of matrices $V_1, V_2, V_3, \ldots$ is said to be F-orthonormal if

$$\langle V_i, V_j \rangle_F = \begin{cases} 0 & i \neq j, \\ 1 & i = j. \end{cases}$$

2.1 The global Arnoldi method

We discuss the global Arnoldi method for the computation of solution subspaces determined by the linear operator and data matrix G in (1.1). The global Arnoldi method was introduced and analyzed by Jbilou et al. [21, 22]; see also Elbouyahyaout et al. [12] and Frommer et al. [14].

Define the linear operator

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{m\times n} &\longrightarrow \mathbb{R}^{m\times n} \\ X &\longrightarrow \mathcal{A}(X) = \sum_{i=1}^{p} A_i \, X \, B_i. \end{aligned} \tag{2.1}$$

The application of k steps of the global Arnoldi method to this operator with initial matrix G yields an F-orthonormal basis $\{V_1, V_2, \ldots, V_{k+1}\}$ of block vectors $V_j \in \mathbb{R}^{m\times n}$ for the block Krylov subspace

$$\mathcal{K}_{k+1}(\mathcal{A}, G) = \text{span}\{G, \mathcal{A}(G), \ldots, \mathcal{A}^k(G)\}. \tag{2.2}$$

In particular, $V_1 = G/\|G\|_F \in \mathbb{R}^{m\times n}$. We define the matrix $\mathcal{V}_{k+1} = [V_1, V_2, \ldots, V_{k+1}]$. Algorithm 1 implements the global Arnoldi method; see [12, 21, 22] for further discussions of this and other block methods. Application of k steps of the algorithm requires k evaluations of the operator $\mathcal{A}$. This is typically the dominating computational effort of the algorithm.

Algorithm 1 The global Arnoldi method

1. Let $V_1 = G/\|G\|_F \in \mathbb{R}^{m\times n}$;
2. `for` $j = 1, \ldots, k$ `do`
   2.1. $V = \mathcal{A}(V_j)$;
   2.3. `for` $i = 1, \ldots, j$ `do`
   $h_{i,j} = \langle V, V_i\rangle_F$;
   $V = V - h_{i,j}V_i$;
   2.4. `end for`
   2.5. $h_{j+1,j} = \|V\|_F$;
   2.6. `if` $h_{j+1,j} > 0$ `then`
   $V_{j+1} = V/h_{j+1,j}$;
   `else`
   `exit`;
   2.7. `end if`
3. `end for`

It follows from the recursions of Algorithm 1 that

$$[\mathcal{A}(V_1), \ldots, \mathcal{A}(V_k)] = \mathcal{V}_{k+1}\,(\widetilde{H}_k \otimes I_n), \tag{2.3}$$

where $\widetilde{H}_k = [h_{i,j}] \in \mathbb{R}^{(k+1)\times k}$ is an upper Hessenberg matrix, whose nontrivial entries are defined by the algorithm. The following result is closely related to [3, Proposition 1].

Proposition 2.1 *Let $\mathcal{V}_k = [V_1, V_2, \ldots, V_k]$ be determined by Algorithm 1. For any $\boldsymbol{z}, \boldsymbol{g} \in \mathbb{R}^k$, we have*

1. $\langle \mathcal{V}_k(\boldsymbol{z}\otimes I_n), \mathcal{V}_k(\boldsymbol{g}\otimes I_n)\rangle_F = \langle \boldsymbol{z}, \boldsymbol{g}\rangle_2$. *In particular,* $\|\mathcal{V}_k(\boldsymbol{z}\otimes I_n)\|_F = \|\boldsymbol{z}\|_2$.
2. $\langle \mathcal{V}_k(\boldsymbol{z}\otimes I_n), G\rangle_F = z_1\|G\|_F$, *where* $\boldsymbol{z} = [z_1, z_2, \ldots, z_k]^T \in \mathbb{R}^k$.
3. $\|\mathcal{V}_k(\boldsymbol{z}\otimes I_n) - G\|_F = \|\boldsymbol{z} - \|G\|_F\boldsymbol{e}_1\|_2$,
   where $\langle \boldsymbol{z}, \boldsymbol{g}\rangle_2 = \boldsymbol{z}^T\boldsymbol{g}$ *denotes the standard inner product between vectors, and* $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T \in \mathbb{R}^k$ *is the first axis vector.*

Proof Property 1 follows from the fact that

$$\langle \mathcal{V}_k(\boldsymbol{z} \otimes I_n), \mathcal{V}_k(\boldsymbol{g} \otimes I_n) \rangle_F = \sum_{i,j=1}^{k} \langle z_i V_i, g_j V_j \rangle_F = \sum_{i=1}^{k} z_i g_i = \langle \boldsymbol{z}, \boldsymbol{g} \rangle_2$$

for any $\boldsymbol{z} = [z_i] \in \mathbb{R}^k$ and $\boldsymbol{g} = [g_i] \in \mathbb{R}^k$. The F-orthonormality of the block vectors $V_1, V_2, \ldots, V_k$ and the fact that $G = \|G\|_F V_1$ give Property 2.

From Properties 1 and 2, we obtain

$$\begin{aligned} \|\mathcal{V}_k(\boldsymbol{z} \otimes I_n) - G\|_F^2 &= \|\mathcal{V}_k(\boldsymbol{z} \otimes I_n)\|_F^2 - 2\langle \mathcal{V}_k(\boldsymbol{z} \otimes I_n), G \rangle_F + \|G\|_F^2 \\ &= \|\boldsymbol{z}\|_2^2 - 2z_1\|G\|_F + \|G\|_F^2 \\ &= \|\boldsymbol{z} - \|G\|_F \boldsymbol{e}_1\|_2^2, \end{aligned}$$

which shows Property 3. □

2.2 The generalized global Arnoldi method

This subsection reviews the generalized global Arnoldi method introduced in [3]. Let $s = m, t = n$ in (1.4), and let the operator $\mathcal{A}$ be defined by (2.1). Introduce the linear regularization operator

$$\begin{aligned} \mathcal{L} : \mathbb{R}^{m \times n} &\longrightarrow \mathbb{R}^{m \times n} \\ X &\longrightarrow \mathcal{L}(X) = \sum_{j=1}^{q} L_j^{(1)} X L_j^{(2)}. \end{aligned}$$

Application of k steps of the generalized global Arnoldi method to the operator pair $(\mathcal{A}, \mathcal{L})$ with initial matrix $G \in \mathbb{R}^{m \times n}$ determines the matrix

$$\mathcal{V}_k = [V_1, V_2, \ldots, V_k], \tag{2.4}$$

where the F-orthonormal block vectors $V_j \in \mathbb{R}^{m \times n}$ form a basis for the generalized block Krylov subspace spanned by the first k of the block vectors

$$G, \mathcal{A}(G), \mathcal{L}(G), \mathcal{A}^2(G), \mathcal{A}\mathcal{L}(G), \mathcal{L}\mathcal{A}(G), \mathcal{L}^2(G), \ldots .$$

Algorithm 2 provides a MATLAB-like description of the computations required. The algorithm requires k applications of each of the operators $\mathcal{A}$ and $\mathcal{L}$. Since the matrices that make up $\mathcal{L}$ are very sparse (banded with a small bandwidth), the dominating computational effort is the application of the operator $\mathcal{A}$. The algorithm also determines matrices H_A and H_L such that

$$[\mathcal{A}(V_1), \mathcal{A}(V_2), \ldots, \mathcal{A}(V_k)] = \mathcal{V}(:, 1 : \alpha_k n)\,(H_{A,k} \otimes I_n)$$

and

$$[\mathcal{L}(V_1), \mathcal{L}(V_2), \ldots, \mathcal{L}(V_k)] = \mathcal{V}(:, 1 : \beta_k n)\,(H_{L,k} \otimes I_n),$$

where $H_{A,k} = H_A(1 : \alpha_k, 1 : k)$ and $H_{L,k} = H_L(1 : \beta_k, 1 : k)$, and α_k and β_k are the values of the parameter N at the end of lines 2.7 and 2.13 of Algorithm 2, respectively, at the last iteration; see [3] for details.

Algorithm 2 The generalized global Arnoldi method

1. $V_1 := G/\|G\|_F$; $N := 1$;
2. `for` $j = 1, 2, \ldots, k$ `do`
   2.1. `if` $j > N$ `then exit`;
   2.2. $\tilde{V} := \mathcal{A}(V_j)$;
   2.3. `for` $i = 1, \ldots, N$ `do`
   $H_A(i, j) := \langle \tilde{V}, V_i \rangle_F$;
   $\tilde{V} := \tilde{V} - H_A(i, j)V_i$;
   2.4. `end for`
   2.5. $H_A(N + 1, j) := \|\tilde{V}\|_F$;
   2.6. `if` $H_A(N + 1, j) > 0$ `then`
   $N := N + 1$;
   $V_N := \tilde{V}/H_A(N, j)$;
   `else`
   `exit`;
   2.7. `end if`
   2.8. $\tilde{V} := \mathcal{L}(V_j)$;
   2.9. `for` $i = 1, \ldots, N$ `do`
   $H_L(i, j) := \langle \tilde{V}, V_i \rangle_F$;
   $\tilde{V} := \tilde{V} - H_L(i, j)V_i$;
   2.10. `end for`
   2.11. $H_L(N + 1, j) := \|\tilde{V}\|_F$;
   2.12. `if` $H_L(N + 1, j) > 0$ `then`
   $N := N + 1$;
   $V_N := \tilde{V}/H_L(N, j)$;
   `else`
   `exit`;
   2.13. `end if`
3. `end for`

3 Tikhonov regularization methods

This section describes iterative methods for the solution of (1.4) based on Algorithms 1 and 2, as well as a solution method based on the application of the standard Arnoldi method to the solution of the minimization problem (1.5). We begin with a discussion on the application of Algorithm 1.

3.1 Tikhonov regularization based on the global Arnoldi method

At step k of Algorithm 3 below, the approximation X_k of the solution of (1.4) in the block Krylov subspace $\mathcal{K}_k(\mathcal{A}, G)$ is of the form

$$X_k = \sum_{i=1}^{k} y_k^{(i)} V_i = \mathcal{V}_k\,(\boldsymbol{y}_k \otimes I_n), \tag{3.1}$$

where $y_k^{(i)}$ denotes the ith component of the vector $\boldsymbol{y}_k \in \mathbb{R}^k$. The following result has been shown in [21, Theorem 2] for the situation when $\mathcal{A}$ is a square matrix. We present a proof for the convenience of the reader.

Proposition 3.1 *Let the upper Hessenberg matrix $\widetilde{H}_k \in \mathbb{R}^{(k+1)\times k}$ and the F-orthonormal bases $\{V_1, V_2, \ldots, V_j\}$ of the block Krylov subspaces $\mathcal{K}_j(\mathcal{A}, G)$ for $j \in \{k, k+1\}$ be produced by Algorithm 1. Let X_k and $\boldsymbol{y}_k$ be related by (3.1). Then,*

$$\|\mathcal{A}(X_k) - G\|_F = \left\|\widetilde{H}_k\,\boldsymbol{y}_k - \|G\|_F \boldsymbol{e}_1\right\|_2. \tag{3.2}$$

Proof Equations (2.3) and (3.1) give

$$\begin{aligned}
\mathcal{A}(X_k) &= \sum_{i=1}^{k} y_k^{(i)} \mathcal{A}(V_i) \\
&= [\mathcal{A}(V_1), \mathcal{A}(V_2), \ldots, \mathcal{A}(V_k)]\,(\boldsymbol{y}_k \otimes I_n) \\
&= \mathcal{V}_{k+1}\,(\widetilde{H}_k \otimes I_n)\,(\boldsymbol{y}_k \otimes I_n) \\
&= \mathcal{V}_{k+1}\,(\widetilde{H}_k\,\boldsymbol{y}_k \otimes I_n).
\end{aligned}$$

Using the above relations, together with Property 3 of Proposition 2.1, we obtain

$$\begin{aligned}
\|\mathcal{A}(X_k) - G\|_F^2 &= \left\|\mathcal{V}_{k+1}\left(\widetilde{H}_k\,\boldsymbol{y}_k \otimes I_n\right) - G\right\|_F^2 \\
&= \left\|\widetilde{H}_k\,\boldsymbol{y}_k - \|G\|_F \boldsymbol{e}_1\right\|_2^2.
\end{aligned}$$

□

Proposition 3.2 *Let X_k be of the form (3.1). Then, there is a matrix $\widetilde{L}_k^T \in \mathbb{R}^{k\times k}$ such that*

$$\|\sum_{j=1}^{q} L_j^{(1)} X_k L_j^{(2)}\|_F^2 = \|\widetilde{L}_k^T \boldsymbol{y}_k\|_2^2.$$

Proof Let

$$M_i = \sum_{j=1}^{q} L_j^{(1)} V_i L_j^{(2)}, \qquad 1 \le i \le k. \tag{3.3}$$

Then,

$$\left\| \sum_{j=1}^{q} L_j^{(1)} X_k L_j^{(2)} \right\|_F^2 = \left\| \sum_{j=1}^{q} L_j^{(1)} \left(\sum_{i=1}^{k} y_k^{(i)} V_i \right) L_j^{(2)} \right\|_F^2$$

$$= \left\| \sum_{i=1}^{k} y_k^{(i)} \left(\sum_{j=1}^{q} L_j^{(1)} V_i L_j^{(2)} \right) \right\|_F^2$$

$$= \left\| \sum_{i=1}^{k} y_k^{(i)} M_i \right\|_F^2 = \operatorname{trace}\left(\left(\sum_{i=1}^{k} y_k^{(i)} M_i^T \right) \left(\sum_{j=1}^{k} y_k^{(j)} M_j \right) \right) \tag{3.4}$$

$$= \sum_{i,j=1}^{k} y_k^{(i)} y_k^{(j)} \operatorname{trace}(M_i^T M_j) = \boldsymbol{y}_k^T N_k \boldsymbol{y}_k, \tag{3.5}$$

where

$$N_k = [n_{i,j}] \in \mathbb{R}^{k\times k}, \quad n_{i,j} = \operatorname{trace}(M_i^T M_j), \quad \boldsymbol{y}_k = [y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(k)}]^T. \tag{3.6}$$

The symmetric matrix N_k is a Gram matrix. It is easy to see that it is positive semidefinite. This follows from the fact that $\boldsymbol{y}_k \in \mathbb{R}^k$ is an arbitrary vector and combining (3.4) and (3.5) shows that $\boldsymbol{y}_k^T N_k \boldsymbol{y}_k \geq 0$.

If N_k is positive definite, then we can let $\widetilde{L}_k$ be its lower triangular Cholesky factor. The proposition now follows from the observation that

$$\boldsymbol{y}_k^T N_k \boldsymbol{y}_k = \boldsymbol{y}_k^T \widetilde{L}_k \widetilde{L}_k^T \boldsymbol{y}_k = \|\widetilde{L}_k^T \boldsymbol{y}_k\|_2^2.$$

The Cholesky factor $\widetilde{L}_k$ is easy to update when k is increased.

When N_k is singular, it may be attractive to instead define $\widetilde{L}_k^T$ with the aid of the spectral factorization $N_k = Q_k D_k Q_k^T$, where $Q_k \in \mathbb{R}^{k\times k}$ is an orthogonal matrix and the diagonal matrix $D_k \in \mathbb{R}^{k\times k}$ has nonnegative diagonal entries. We may choose $\widetilde{L}_k^T = D_k^{1/2} Q_k^T$. □

Combining Propositions 3.1 and 3.2 yields the following result.

Theorem 3.3 *Let* $X_k = \sum_{i=1}^{k} y_k^{(i)} V_i = \mathcal{V}_k(\boldsymbol{y}_k \otimes I_n)$ *denote the solution of* (1.4) *restricted to the block Krylov subspace* $\mathcal{K}_k(\mathcal{A}, G)$, *where* $\boldsymbol{y}_k = [y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(k)}]^T$. *Then,* $\boldsymbol{y}_k$ *solves the reduced Tikhonov minimization problem*

$$\min_{\boldsymbol{y}\in\mathbb{R}^k} \left\{ \left\| \widetilde{H}_k \boldsymbol{y} - \|G\|_F \boldsymbol{e}_1 \right\|_2^2 + \mu \left\| \widetilde{L}_k^T \boldsymbol{y} \right\|_2^2 \right\}. \tag{3.7}$$

Typically, the matrix $\widetilde{L}_k^T$ is not very ill-conditioned. This holds, for instance, for the computed examples in Section 4. We will assume this to be the case and let $\boldsymbol{z} = \widetilde{L}_k^T \boldsymbol{y}$ and

$$\hat{H}_k = \tilde{H}_k \widetilde{L}_k^{-T}. \tag{3.8}$$

Then, the Tikhonov minimization problem (3.7) can be expressed in standard form,

$$\min_{z\in\mathbb{R}^k}\{\,\|\hat{H}_k\,z-\|G\|_F\,\boldsymbol{e}_1\|_2^2+\mu\|z\|_2^2\,\}. \tag{3.9}$$

We solve (3.9) by computing the solution $z_{\mu,k}$ of the least-square problem

$$\min_{z\in\mathbb{R}^k}\left\|\begin{bmatrix}\hat{H}_k\\ \mu^{1/2}I_k\end{bmatrix}z-\begin{bmatrix}\|G\|_F\,\boldsymbol{e}_1\\ 0\end{bmatrix}\right\|_2^2. \tag{3.10}$$

In the present paper, we determine the regularization parameter $\mu>0$ by the discrepancy principle. This requires that a bound for the error E in G be available. Thus, assume that a bound

$$\|E\|_F\le\varepsilon \tag{3.11}$$

is known. The discrepancy principle prescribes that the parameter $\mu>0$ be chosen so that

$$\left\|G-\sum_{i=1}^{p}A_i\,X_{\mu,k}\,B_i\right\|_F^2=\|\|G\|_F\,\boldsymbol{e}_1-\hat{H}_k z_{\mu,k}\|_2^2=\eta\varepsilon, \tag{3.12}$$

where the first equality above stems from (3.2) and $\eta\ge1$ is a user-chosen constant independent of ε. The second equality provides a nonlinear equation for μ with small matrices. It can be solved by a variety of methods such as by Newton's method. Let $v=1/\mu$, and introduce the function

$$\phi(v)=\|\boldsymbol{f}-\hat{H}_k z_{1/v,k}\|_2^2, \tag{3.13}$$

where $\boldsymbol{f}=\|G\|_F\,\boldsymbol{e}_1$. The following proposition, shown in [34], sheds some light on the properties of this function.

Proposition 3.4 *Assume that $\hat{H}_k^T\boldsymbol{f}\neq0$. Then, the function (3.13) can be expressed as*

$$\phi(v)=\boldsymbol{f}^T(v\hat{H}_k\hat{H}_k^T+I_k)^{-2}\boldsymbol{f},$$

and ϕ is decreasing and convex for $v>0$. Furthermore, the equation $\phi(v)=\tau^2$ has a unique solution $0<v<+\infty$ for any τ with

$$\|P_{\mathcal{N}(\hat{H}_k\hat{H}_k^T)}(\boldsymbol{f})\|_2^2<\tau^2<\|\boldsymbol{f}\|_2^2,$$

where $P_{\mathcal{N}(\hat{H}_k\hat{H}_k^T)}$ denotes the orthogonal projector onto the null space of $\hat{H}_k\hat{H}_k^T$.

Algorithm 3 summarizes the global Arnoldi–Tikhonov regularization method for the solution of (1.4). The solution subspace (2.2) is independent of the orthogonal projectors that determine the regularization term in (1.4). The matrix $\widetilde{L}_k^T$ is assumed to be fairly well conditioned. In certain situations, such as when the solution subspace is required to contain a component of the null space of the regularization operator, this may not be the case. It may then be attractive to define $\widetilde{L}_k^T$ with the aid of the spectral factorization of the matrix N_k as described in the proof of Proposition 3.2.

Algorithm 3 Global Arnoldi–Tikhonov regularization method

1. input: $A_i \in \mathbb{R}^{m\times m}$, $B_i \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^{m\times n}$, $L_j^{(1)} \in \mathbb{R}^{s\times m}$, $L_j^{(2)} \in \mathbb{R}^{n\times t}$;
2. initialize: ε, $\eta > 1$;
3. `for` $k = 1, 2, \ldots$ until convergence `do`
   3.1. construct $\mathcal{V}_k = [V_1, V_2, \ldots, V_k]$ and $\widetilde{H}_k$ by Algorithm 1;
   3.2. compute M_i $(i = 1, 2, \ldots, k)$ by (3.3) and N by (3.6);
   3.3. compute $\widetilde{L}_k^T$ as described by Proposition 3.2;
   3.4. compute $\hat{H}_k$ by (3.8);
   3.5. compute the zero v of (3.13) by the discrepancy principle;
   3.6. define the regularization parameter $\mu = 1/v$;
   3.7. compute $z_{\mu,k}$ by (3.10), and let $y_k = \widetilde{L}_k^{-T} z_{\mu,k}$;
4. `end for`
5. output: approximation solution $X_k^* = \sum_{i=1}^k V_i y_k^{(i)}$ of (1.4);

3.2 Tikhonov regularization based on the generalized global Arnoldi method

We turn to the application of Algorithm 2 to the solution of (1.4). Let the matrix $\mathcal{V}_k$ be defined by (2.4). Restricting the solution of (1.4) to the space spanned by the block columns of $\mathcal{V}_k$ gives the minimization problem

$$\min_{X\in \mathrm{span}\{V_1,V_2,\ldots,V_k\}} \left\{ \|\mathcal{A}(X) - G\|_F^2 + \mu \|\mathcal{L}(X)\|_F^2 \right\}. \tag{3.14}$$

An analog of Proposition 2.1 is valid; see Bouhamidi et al. [3]. This can be used to show that

$$\begin{aligned} \|\mathcal{A}(X_k) - G\|_F^2 &= \left\| \mathcal{V}_{\alpha_k} \left(H_{A,k}\, y_k \otimes I_n \right) - G \right\|_F^2 \\ &= \left\| H_{A,k}\, y_k - z_G \right\|_2^2 + \|G\|_F^2 - \|z_G\|_2^2 \end{aligned}$$

and

$$\|\mathcal{L}(X_k)\|_F = \left\| \mathcal{V}_{\beta_k} \left(H_{L,k}\, y_k \otimes I_n \right) \right\|_F = \left\| H_{L,k}\, y_k \right\|_2.$$

We conclude that problem (3.14) is equivalent to the low-dimensional Tikhonov regularization problem

$$\min_{y_k\in\mathbb{R}^k} \{ \|H_{A,k}\, y_k - z_G\|_2^2 + \mu \|H_{L,k}\, y_k\|_2^2 \}. \tag{3.15}$$

We refer to Bouhamidi et al. [3] for details. The problem (3.15) can be transformed to standard form similarly as (3.7), and the transformed problem can be solved by an algorithm analogous to Algorithm 3. We omit the details.

3.3 Tikhonov regularization based on the standard Arnoldi method

This subsection discusses the application of the standard Arnoldi method to the Tikhonov regularization problem (1.5). The Arnoldi method applied to the matrix

K with initial vector $\boldsymbol{g}$ yields at step k the matrix $V_{k+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{mn\times(k+1)}$, whose columns form an orthonormal basis for the Krylov subspace

$$\mathcal{K}_{k+1}(K, \boldsymbol{g}) = \text{span}\{\boldsymbol{g}, K\boldsymbol{g}, \ldots, K^k\boldsymbol{g}\},$$

and the upper Hessenberg matrix $\widetilde{H}_k \in \mathbb{R}^{(k+1)\times k}$. These matrices satisfy

$$K V_k = V_{k+1}\widetilde{H}_k; \tag{3.16}$$

see, e.g., [6, 15, 26, 35]. We assume here that the Arnoldi method does not break down. This is the generic situation. Matrix-vector product evaluations with K should be carried out without explicitly forming K.

Algorithm 4 The standard Arnoldi–Tikhonov regularization method

1. input: $A_i \in \mathbb{R}^{m\times m}$, $B_i \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^{m\times n}$, $L_j^{(1)} \in \mathbb{R}^{s\times m}$, $L_j^{(2)} \in \mathbb{R}^{n\times t}$;
2. initialize: ε, $\eta > 1$;
3. let K, L, and $\boldsymbol{g}$ be as defined in (1.5); the matrices K and L do not have to be explicitly formed;
4. `for` $k = 1, 2, \ldots$ until convergence `do`
   4.1. construct $V_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k]$ and $\widetilde{H}_k$ by the Arnoldi process;
   4.2. compute $LV_k = [LV_{k-1}, L\boldsymbol{v}_k]$;
   4.3. compute $R_{L,k}$ by (3.17);
   4.4. compute $\hat{H}_k$ by (3.8);
   4.5. compute the zero v of (3.13) by the discrepancy principle;
   4.6. define the regularization parameter $\mu = 1/v$;
   4.7. compute $\boldsymbol{z}_k$ by (3.10), and let $\boldsymbol{y}_k = R_{L,k}^{-1}\boldsymbol{z}_k$;
5. `end for`
6. output: approximation solution $\boldsymbol{x}_k^* = V_k\boldsymbol{y}_k$ of (1.5);

Let $LV_k = [LV_{k-1}, L\boldsymbol{v}_k]$ and compute the QR factorization

$$LV_k = Q_{L,k}\, R_{L,k}, \tag{3.17}$$

where the matrix $Q_{L,k} \in \mathbb{R}^{st\times k}$ has orthonormal columns and $R_{L,k} \in \mathbb{R}^{k\times k}$ is upper triangular. This factorization can be evaluated by updating the available QR factorization of LV_{k-1}; see Daniel et al. [8] for a discussion on updating methods.

By using (3.16) and (3.17), it is not difficult see that when restricting the solution of the problem (1.5) to the Krylov subspace $\mathcal{K}_k(K, \boldsymbol{g})$, the minimization problem so obtained can be expressed as the low-dimensional Tikhonov regularization problem

$$\min_{\boldsymbol{y}\in\mathbb{R}^k}\{\,\|\widetilde{H}_k\,\boldsymbol{y} - \|\boldsymbol{z}_G\|_2\boldsymbol{e}_1\|_2^2 + \mu\|R_{L,k}\boldsymbol{y}\|_2^2\,\}.$$

This minimization problem can be transformed to standard form (3.9) by using (3.8). Algorithm 4 summarizes this solution method for problem (1.5).

4 Numerical examples

This section presents a few examples that compare the solution methods described in Sections 3.1, 3.2, and 3.3 when applied to image restoration problems. The solution method defined by Algorithm 3 is referred to as the global Arnoldi–Tikhonov (GAT) method. We refer to the method discussed in Section 3.2 as the generalized global Arnoldi (GGA) method. These methods are compared to the standard Arnoldi (SA) method outlined in Section 3.3. The regularization parameter μ is determined with the aid of the discrepancy principle for all methods. Thus, we assume that a bound (3.11) for the error E in G is known; cf. (1.2). The matrix E has normally distributed entries with zero mean and is scaled to correspond to a specific noise level

$$\nu := \frac{\|E\|_F}{\|\widehat{G}\|_F},$$

which is assumed to be known. Thus, we use $\varepsilon = \nu\|\widehat{G}\|_F$ in (3.12) and let $\eta = 1.01$.

For each example, we choose a pair of regularization matrices $(L_j^{(1)}, L_j^{(2)})$, where $L_1^{(\ell)}$ is of the form

$$L_1 = \begin{bmatrix} 1 & -1 & & & 0 \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & \ddots & \ddots \\ & & & & 1 \ -1 \\ 0 & & & & 0 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{4.1}$$

and $L_2^{(\ell)}$ is of the form

$$L_2 = \begin{bmatrix} 0 & 0 & 0 & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ 0 & & & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

We note that the matrices L_1 and L_2 do not have to be square for the GAT and SA methods. Here we choose square regularization matrices because this is required by the GGA method. The zero-padding of L_1 and L_2 does not affect the solutions computed by the GAT and SA methods.

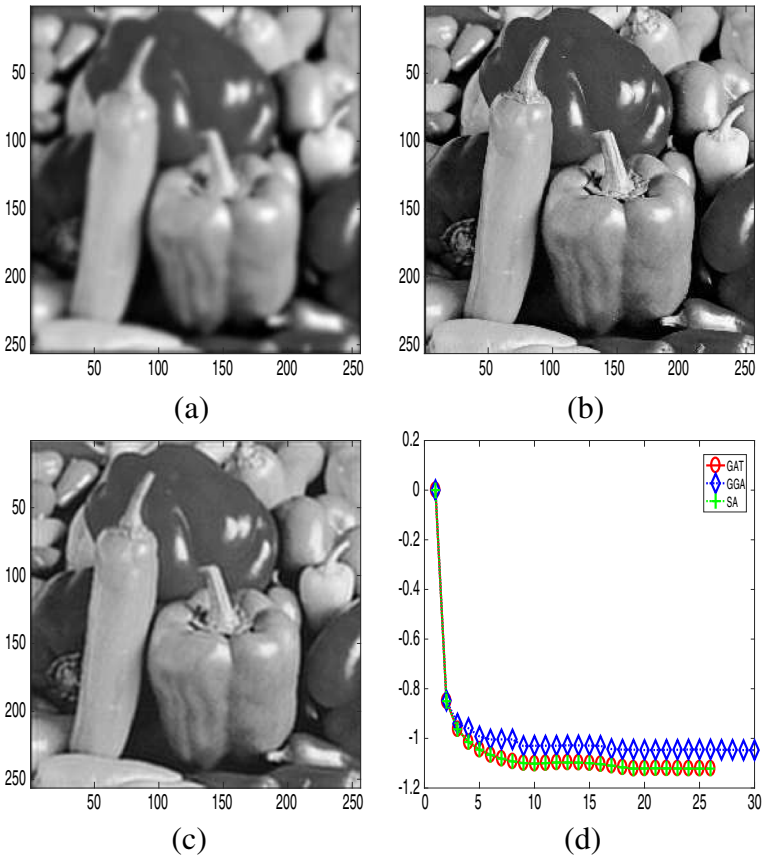

Fig. 1 Example 4.1: **a** available blur- and noise-contaminated `peppers` image represented by the matrix G, **b** desired image, **c** restored image for the noise level $\nu = 1 \cdot 10^{-3}$ and regularization matrix pair (L_1, L_1), and **d** convergence of the relative error e_k (in logarithmic scale) as a function of k for the SA, GGA, and GAT methods

The quality of the computed approximate solutions $X_{\mu,k}$ of (1.4) of the general form (3.1) is measured with the relative error norm

$$e_k := \frac{\|X_{\mu_k,k} - \widehat{X}\|_F}{\|\widehat{X}\|_F},$$

where $\widehat{X}$ is the desired solution of the unknown error-free problem (1.3).

The iterations with the algorithms in our comparison are terminated as soon as the relative change in an iterate satisfies

$$\frac{\|X_{\mu_k,k} - X_{\mu_{k-1},k-1}\|_F}{\|X_{\mu_{k-1},k-1}\|_F} \leq \tau \tag{4.2}$$

or the number of iterations reaches $k_{\max} = 40$. We set $\tau = 1 \cdot 10^{-4}$ for Example 4.1 and $\tau = 5 \cdot 10^{-4}$ for the remaining examples. The parameters $k_{\max}$ and τ are chosen

Table 1 Example 4.1: number of iteration k, regularization parameter μ_k, CPU time in second, and relative error e_k in the computed approximate solutions $X_{\mu_k,k}$ determined by different methods with different noise levels and regularization matrices

Method	$(L_1^{(1)}, L_1^{(2)})$	k	μ_k	CPU time (s)	Relative error e_k
Noise level $\nu = 1 \cdot 10^{-2}$					
SA	(L_1, L_1)	16	$8.79 \cdot 10^{-2}$	2.34	$9.59 \cdot 10^{-2}$
GGA	(L_1, L_1)	34	$2.94 \cdot 10^{-2}$	13.57	$9.88 \cdot 10^{-2}$
GAT	(L_1, L_1)	16	$8.79 \cdot 10^{-2}$	1.42	$9.59 \cdot 10^{-2}$
SA	(L_1, L_2)	15	$1.70 \cdot 10^{-1}$	2.18	$9.64 \cdot 10^{-2}$
GGA	(L_1, L_2)	35	$5.98 \cdot 10^{-1}$	13.76	$1.03 \cdot 10^{-1}$
GAT	(L_1, L_2)	15	$1.70 \cdot 10^{-1}$	1.13	$9.64 \cdot 10^{-2}$
SA	(L_2, L_2)	14	$4.86 \cdot 10^{-1}$	2.14	$9.70 \cdot 10^{-2}$
GGA	(L_2, L_2)	25	$7.31 \cdot 10^{-2}$	8.30	$9.82 \cdot 10^{-2}$
GAT	(L_2, L_2)	14	$4.87 \cdot 10^{-1}$	1.08	$9.70 \cdot 10^{-2}$
Noise level $\nu = 1 \cdot 10^{-3}$					
SA	(L_1, L_1)	25	$1.03 \cdot 10^{-3}$	4.17	$7.56 \cdot 10^{-2}$
GGA	(L_1, L_1)	29	$6.77 \cdot 10^{-2}$	10.37	$8.97 \cdot 10^{-2}$
GAT	(L_1, L_1)	25	$1.03 \cdot 10^{-3}$	2.55	$7.56 \cdot 10^{-2}$
SA	(L_1, L_2)	24	$1.17 \cdot 10^{-3}$	4.12	$7.59 \cdot 10^{-2}$
GGA	(L_1, L_2)	34	$8.75 \cdot 10^{-1}$	13.16	$8.88 \cdot 10^{-2}$
GAT	(L_1, L_2)	24	$1.19 \cdot 10^{-3}$	2.47	$7.59 \cdot 10^{-2}$
SA	(L_2, L_2)	23	$1.87 \cdot 10^{-3}$	3.78	$7.60 \cdot 10^{-2}$
GGA	(L_2, L_2)	20	$2.35 \cdot 10^{-3}$	6.02	$8.47 \cdot 10^{-2}$
GAT	(L_2, L_2)	22	$2.48 \cdot 10^{-3}$	2.17	$7.65 \cdot 10^{-2}$

so that the computed solution does not change much with the iteration number, k, when the iterations are terminated. All computations were carried out in MATLAB R2017a with about 15 significant decimal digits on a laptop computer with an Intel Core i7-6700HQ CPU @ 2.60GHz processor and 16GB RAM.

Example 4.1 We consider the restoration of the test image `peppers`, which is represented by an array of 256×256 pixels. We let $p = 1$ and $q = 1$ in (1.4). The available blur- and noise-contaminated image is represented by the matrix $G \in \mathbb{R}^{256\times256}$. It is corrupted by Gaussian blur and additive zero-mean white Gaussian noise. The blurring matrix $A_1 = [a_{ij}] \in \mathbb{R}^{256\times256}$ is of Toeplitz form and given by

$$a_{ij} = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(i-j)^2}{2\sigma^2}\right), & |i-j| \le d, \\ 0, & \text{otherwise.} \end{cases}$$

It is generated by the MATLAB function blur from [17] using the parameters `band` $= d = 7$ and `sigma` $= \sigma = 2$. We let $B_1 = A_1$.

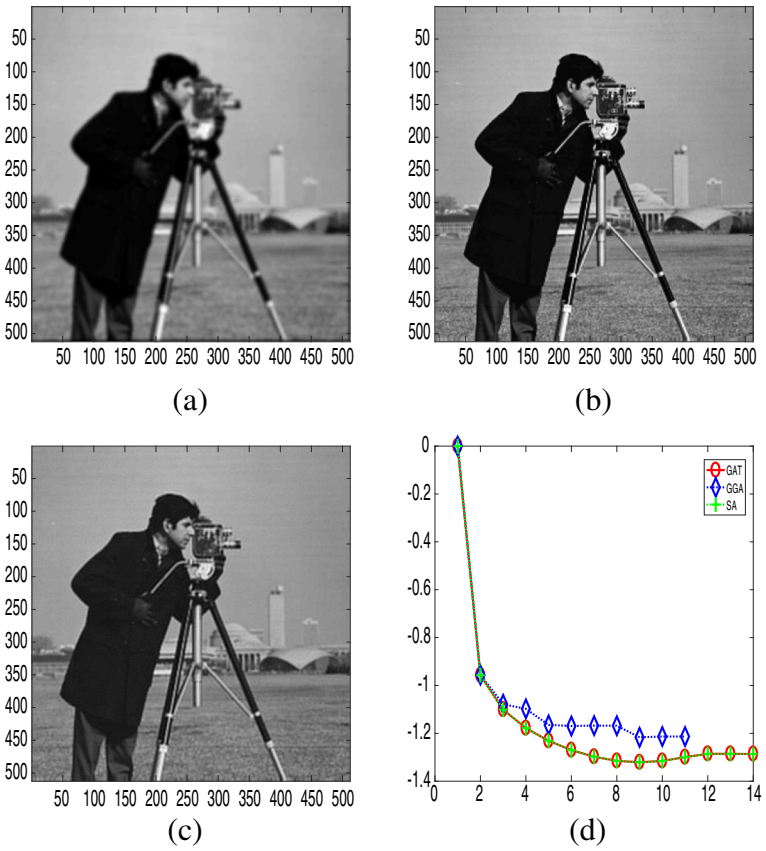

Fig. 2 Example 4.2: **a** available blur- and noise-contaminated `cameraman` image represented by the matrix G, **b** desired image, **c** restored image for the noise level $\nu = 1 \cdot 10^{-3}$ and regularization matrix pair (L_1, L_1), and **d** convergence of the relative error e_k (in logarithmic scale) as a function of k for the SA, GGA, and GAT methods

The available blur- and noise-contaminated image is displayed in Fig. 1a. The noise corresponds to the noise level $\nu = 1 \cdot 10^{-3}$. Table 1 shows results for the restoration of this image as well as for an image that has been contaminated by noise of level $\nu = 1 \cdot 10^{-2}$ and the same blur. Figure 1b depicts the desired blur- and noise-free image. It is represented by the matrix $\widehat{X} \in \mathbb{R}^{256\times 256}$ and is assumed not to be known. Figure 1c displays the restoration that is obtained with the regularization matrix pair $(L_1^{(1)}, L_1^{(2)}) = (L_1, L_1)$. Finally, Fig. 1d shows the convergence history of the relative error in the computed iterates as a function of the number of iterations. The iterations are terminated as soon as the stopping rule (4.2) is satisfied. Table 1 shows the number of iterations, denoted by k, the value of the regularization

Table 2 Example 4.2: number of iteration k, regularization parameter μ_k, CPU time in second, and relative error e_k in the computed approximate solutions $X_{\mu_k,k}$ determined by different methods with different noise levels and regularization matrices

Method	$(L_1^{(1)}, L_1^{(2)})$	k	μ_k	CPU time (s)	Relative error e_k
Noise level $\nu = 1 \cdot 10^{-2}$					
SA	(L_1, L_1)	10	$1.86 \cdot 10^{-1}$	4.01	$6.38 \cdot 10^{-2}$
GGA	(L_1, L_1)	6	$8.86 \cdot 10^{-2}$	2.49	$7.91 \cdot 10^{-2}$
GAT	(L_1, L_1)	10	$1.86 \cdot 10^{-1}$	2.43	$6.38 \cdot 10^{-2}$
SA	(L_1, L_2)	14	$4.89 \cdot 10^{-1}$	6.33	$6.52 \cdot 10^{-2}$
GGA	(L_1, L_2)	14	$5.97 \cdot 10^{-2}$	9.27	$7.75 \cdot 10^{-2}$
GAT	(L_1, L_2)	14	$4.89 \cdot 10^{-1}$	4.04	$6.52 \cdot 10^{-2}$
SA	(L_2, L_2)	14	2.17	6.49	$6.53 \cdot 10^{-2}$
GGA	(L_2, L_2)	11	$5.09 \cdot 10^{-2}$	6.22	$7.24 \cdot 10^{-2}$
GAT	(L_2, L_2)	14	2.25	3.85	$6.53 \cdot 10^{-2}$
Noise level $\nu = 1 \cdot 10^{-3}$					
SA	(L_1, L_1)	13	$2.84 \cdot 10^{-5}$	5.65	$5.17 \cdot 10^{-2}$
GGA	(L_1, L_1)	10	$1.94 \cdot 10^{-1}$	5.22	$6.12 \cdot 10^{-2}$
GAT	(L_1, L_1)	13	$2.84 \cdot 10^{-5}$	3.35	$5.17 \cdot 10^{-2}$
SA	(L_1, L_2)	13	$1.01 \cdot 10^{-5}$	6.16	$5.14 \cdot 10^{-2}$
GGA	(L_1, L_2)	10	$2.78 \cdot 10^{-1}$	5.19	$6.15 \cdot 10^{-2}$
GAT	(L_1, L_2)	13	$1.01 \cdot 10^{-5}$	3.41	$5.14 \cdot 10^{-2}$
SA	(L_2, L_2)	21	$8.19 \cdot 10^{-3}$	11.40	$4.11 \cdot 10^{-2}$
GGA	(L_2, L_2)	7	$2.83 \cdot 10^{-3}$	3.21	$6.63 \cdot 10^{-2}$
GAT	(L_2, L_2)	20	$1.57 \cdot 10^{-2}$	7.22	$4.23 \cdot 10^{-2}$

parameter at iteration k, denoted by μ_k, the CPU time, and the relative error, e_k, in the computed approximate solutions $X_{\mu_k,k}$ determined by the GAT, GGA, and SA methods. The GAT and SA methods are seen to yield more accurate approximations of $\widehat{X}$ than the GGA method for both noise levels and all choices of regularization matrix pairs. We also note that the regularization matrices $L_1^{(1)} = L_1^{(2)} = L_1$ give somewhat more accurate approximations of $\widehat{X}$ than the other choices of regularization matrices.

The GAT and SA methods yield restorations of the highest quality, but the CPU times required by these methods differ. Step 4.3 of Algorithm 4 demands more CPU time than steps 3.2–3.3 of Algorithm 3. These methods use the same solution subspaces, but their implementation differs. The GAT method works with block vectors in $\mathbb{R}^{n\times n}$, while the SA method works with the associated vectors in $\mathbb{R}^{n^2}$. The handling of the regularization term differs and this may affect the computed solutions and when the stopping criterion is satisfied by the GAT and SA methods. The timings do not include the time for matrix-vector product evaluations, because by using the structure of the operators $\mathcal{A}$ and $\mathcal{L}$, the evaluation of these products takes the same time for all methods.

Table 3 Example 4.3: number of iteration k, regularization parameter μ_k, CPU time in second, and relative error e_k in the computed approximate solutions $X_{\mu_k,k}$ determined by different methods with different noise levels and regularization matrices

Method	$(L_1^{(1)}, L_1^{(2)})$	k	μ_k	CPU time (s)	Relative error e_k
Noise level $\nu = 1 \cdot 10^{-2}$					
SA	(L_1, L_1)	8	$5.46 \cdot 10^{-1}$	3.07	$4.39 \cdot 10^{-2}$
GGA	(L_1, L_1)	6	$1.88 \cdot 10^{-1}$	2.70	$5.54 \cdot 10^{-2}$
GAT	(L_1, L_1)	8	$5.45 \cdot 10^{-1}$	1.98	$4.39 \cdot 10^{-2}$
SA	(L_1, L_2)	12	$9.31 \cdot 10^{-1}$	5.14	$4.50 \cdot 10^{-2}$
GGA	(L_1, L_2)	14	$1.23 \cdot 10^{-1}$	9.64	$5.24 \cdot 10^{-2}$
GAT	(L_1, L_2)	12	$9.31 \cdot 10^{-1}$	3.50	$4.50 \cdot 10^{-2}$
SA	(L_2, L_2)	15	2.63	7.31	$4.58 \cdot 10^{-2}$
GGA	(L_2, L_2)	10	$1.68 \cdot 10^{-3}$	5.64	$5.23 \cdot 10^{-2}$
GAT	(L_2, L_2)	15	2.63	5.07	$4.58 \cdot 10^{-2}$
Noise level $\nu = 1 \cdot 10^{-3}$					
SA	(L_1, L_1)	17	$1.02 \cdot 10^{-2}$	8.36	$2.21 \cdot 10^{-2}$
GGA	(L_1, L_1)	6	$7.13 \cdot 10^{-1}$	2.78	$4.30 \cdot 10^{-2}$
GAT	(L_1, L_1)	17	$1.02 \cdot 10^{-2}$	6.10	$2.21 \cdot 10^{-2}$
SA	(L_1, L_2)	16	$9.23 \cdot 10^{-3}$	7.70	$2.22 \cdot 10^{-2}$
GGA	(L_1, L_2)	11	$4.20 \cdot 10^{-1}$	6.50	$3.62 \cdot 10^{-2}$
GAT	(L_1, L_2)	16	$9.23 \cdot 10^{-3}$	5.45	$2.22 \cdot 10^{-2}$
SA	(L_2, L_2)	9	$1.33 \cdot 10^{-4}$	3.59	$2.66 \cdot 10^{-2}$
GGA	(L_2, L_2)	10	$4.84 \cdot 10^{-3}$	5.72	$3.34 \cdot 10^{-2}$
GAT	(L_2, L_2)	9	$1.33 \cdot 10^{-4}$	2.31	$2.66 \cdot 10^{-2}$

Example 4.2 We consider the restoration of the test image `cameraman`, which is represented by an array of 512×512 pixels. We let $p = 1$ and $q = 1$ in (1.4). The available image, represented by the matrix $G \in \mathbb{R}^{512\times512}$, is corrupted by Gaussian blur and additive zero-mean white Gaussian noise. The blurring matrices A_1 and B_1 are of the same type as in Example 4.1 with `band` $= d = 7$ and `sigma` $= \sigma = 2.5$. A larger σ yields more blur than a smaller one. The blur- and noise-contaminated image with noise level $1 \cdot 10^{-3}$ is shown in Fig. 2a. Figure 2b displays the desired blur- and noise-free image, which is represented by the matrix $\widehat{X} \in \mathbb{R}^{512\times512}$ and is assumed not to be known.

Table 2 is analogous to Table 1. Also, for this example, the GAT and SA methods yield the most accurate restorations with about the same error, with GAT requiring the least CPU time. The regularization matrix pair (L_1, L_1) gives slightly higher accuracy than pairs of other regularization matrices for $\nu = 1 \cdot 10^{-2}$.

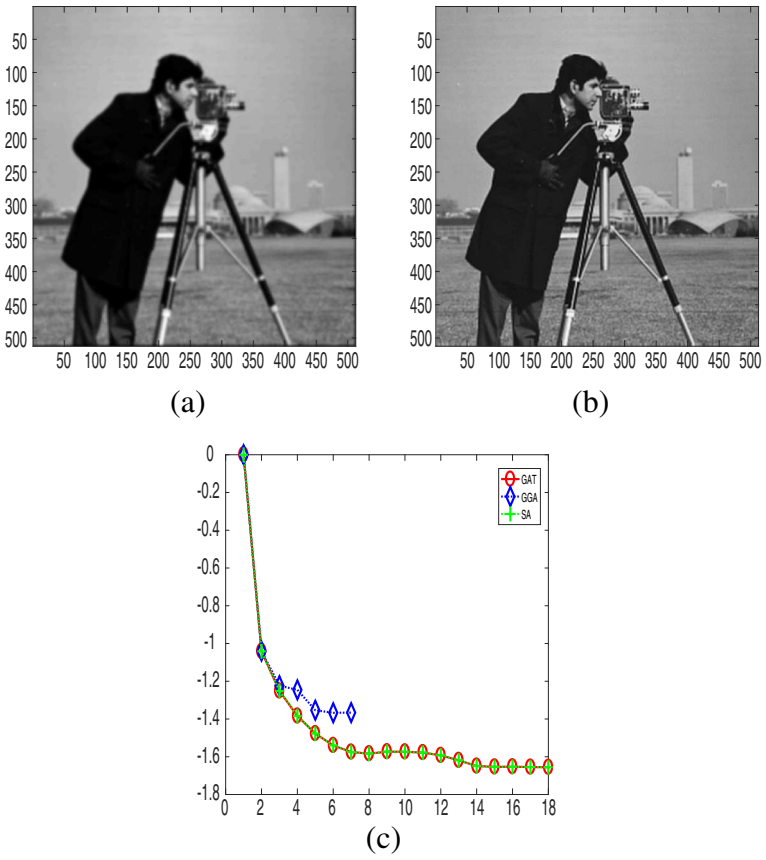

Fig. 3 Example 4.3: **a** available blur- and noise-contaminated `cameraman` image represented by the matrix G, **b** restored image for the noise level $\nu = 1 \cdot 10^{-3}$ and regularization matrix pair (L_1, L_1), and **c** convergence of the relative error e_k (in logarithmic scale) as a function of k for the SA, GGA, and GAT methods

Example 4.3 This example differs from Example 4.2 only in that the available image is contaminated by two kinds of Gaussian blur as well as by Gaussian noise. We let $p = 2$ and $q = 1$ in (1.1) define the blurring operator

$$\mathcal{A}(X) = A_1 X B_1 + A_2 X B_2,$$

where A_i and B_i are matrices of the same form as in Example 4.2. The matrices A_1 and B_1 are determined by the parameters $d = 5$ and $\sigma = 1.5$, and the matrices A_2 and B_2 are defined by the parameters $d = 7$ and $\sigma = 2.5$. The regularization matrices are the same as in Example 4.2. Table 3 and Fig. 3 are analogous to Table 2 and Fig. 2, respectively. Similarly as in the previous examples, the GAT and SA methods yield restorations of the highest quality with the former method requiring the least CPU time. The most accurate restorations are obtained when regularization matrix pair (L_1, L_1) is used with the GAT and SA methods.

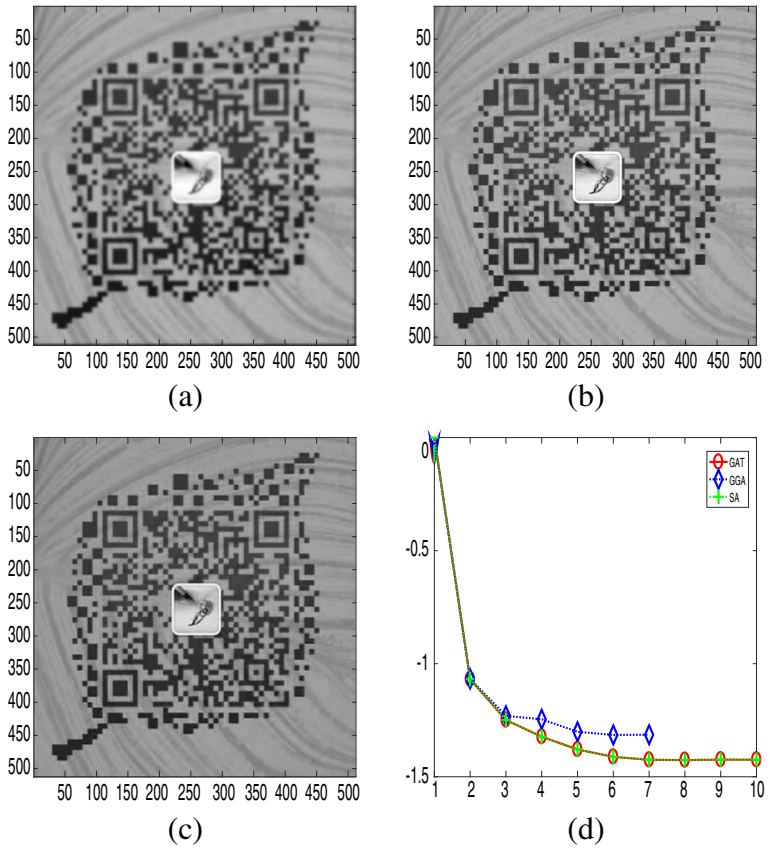

Fig. 4 Example 4.4: **a** available blur- and noise-contaminated `hgxwechat` image represented by the matrix G, **b** desired image, **c** restored image for the noise level $\nu = 1 \cdot 10^{-3}$ and regularization matrix pair (L_1, L_1), and **d** convergence of the relative error e_k (in logarithmic scale) as a function of k for the SA, GGA, and GAT methods

Example 4.4 This example differs form Example 4.3 only in the image to be restored. Here we consider the restoration of the image `hgxwechat`, which is represented by an array of 512×512 pixels and has been corrupted by Gaussian blur and additive zero-mean white Gaussian noise. The blurring operator and matrices A_i and B_i used for this example are the same as in Example 4.3 and so are the noise levels. Figure 4 and Table 4 are analogous to Fig. 3 and Table 3, respectively.

Table 4 shows both GAT and SA to determine the best approximations of $\widehat{X}$ for all noise levels, but SA needs more CPU time than GAT. All regularization matrix pairs give restorations of about the same quality.

Table 4 Example 4.4: number of iteration k, regularization parameter μ_k, CPU time in second, and relative error e_k in the computed approximate solutions $X_{\mu_k,k}$ determined by different methods with different noise levels and regularization matrices

Method	$(L_1^{(1)}, L_1^{(2)})$	k	μ_k	CPU time (s)	Relative error e_k
Noise level $\nu = 1 \cdot 10^{-2}$					
SA	(L_1, L_1)	12	1.17	5.11	$5.12 \cdot 10^{-2}$
GGA	(L_1, L_1)	6	$1.85 \cdot 10^{-1}$	2.74	$5.80 \cdot 10^{-2}$
GAT	(L_1, L_1)	12	1.17	3.43	$5.12 \cdot 10^{-2}$
SA	(L_1, L_2)	11	2.06	4.61	$5.15 \cdot 10^{-2}$
GGA	(L_1, L_2)	12	$1.22 \cdot 10^{-1}$	7.59	$5.64 \cdot 10^{-2}$
GAT	(L_1, L_2)	11	2.06	3.29	$5.15 \cdot 10^{-2}$
SA	(L_2, L_2)	11	4.95	4.64	$5.16 \cdot 10^{-2}$
GGA	(L_2, L_2)	6	$1.07 \cdot 10^{-1}$	2.77	$5.71 \cdot 10^{-2}$
GAT	(L_2, L_2)	11	4.94	2.98	$5.16 \cdot 10^{-2}$
Noise level $\nu = 1 \cdot 10^{-3}$					
SA	(L_1, L_1)	9	$1.21 \cdot 10^{-3}$	3.51	$3.76 \cdot 10^{-2}$
GGA	(L_1, L_1)	6	$8.48 \cdot 10^{-1}$	2.74	$4.85 \cdot 10^{-2}$
GAT	(L_1, L_1)	9	$1.21 \cdot 10^{-3}$	2.32	$3.76 \cdot 10^{-2}$
SA	(L_1, L_2)	9	$4.07 \cdot 10^{-4}$	3.55	$3.76 \cdot 10^{-2}$
GGA	(L_1, L_2)	11	$4.40 \cdot 10^{-1}$	6.59	$4.33 \cdot 10^{-2}$
GAT	(L_1, L_2)	9	$4.07 \cdot 10^{-4}$	2.28	$3.76 \cdot 10^{-2}$
SA	(L_2, L_2)	9	$1.37 \cdot 10^{-4}$	3.80	$3.76 \cdot 10^{-2}$
GGA	(L_2, L_2)	5	$5.00 \cdot 10^{-1}$	2.15	$4.78 \cdot 10^{-2}$
GAT	(L_2, L_2)	9	$1.37 \cdot 10^{-4}$	2.34	$3.76 \cdot 10^{-2}$

5 Concluding remarks

This paper compares three iterative methods based on the Arnoldi process for the solution of large-scale discrete ill-posed problems. They differ in the choice of solution subspace, choice of regularization matrices, and implementation. A global Arnoldi method, referred to as the GAT method, that generates the solution subspace independent of the regularization matrices is found to generally yield restorations of the highest quality and to typically require the least CPU time for various regularization matrix pairs $(L_1^{(1)}, L_1^{(2)})$.

While other solution subspaces may yield higher accuracy for certain discrete ill-posed problems, our investigation suggests that for many problems, the GAT method may be competitive. This method allows rectangular regularization matrices and our experiments suggest that the regularization matrix (4.1) or its rectangular analog are good default choices for linear discrete ill-posed problems in two space dimensions.

Funding information Research by GH was supported in part by the Fund of Application Foundation of Science and Technology Department of the Sichuan Province (2016JY0249) and by NNSF

(2017YFC0601505, 41672325), research by LR was supported in part by NSF grants DMS-1729509 and DMS-1720259, and research by FY was supported in part by NNSF (11501392).

References

1. Bentbib, A.H., El Guide, M., Jbilou, K., Reichel, L.: Global Golub–Kahan bidiagonalization for image restoration. J. Comput. Appl. Math. **300**, 233–244 (2016)
2. Bouhamidi, A., Jbilou, K.: Sylvester Tikhonov-regularization methods in image restoration. J. Comput. Appl. Math. **206**, 86–98 (2007)
3. Bouhamidi, A., Jbilou, K., Reichel, L., Sadok, H.: A generalized global Arnoldi method for ill-posed matrix equations. J. Comput. Appl. Math. **236**, 2078–2089 (2012)
4. Brezinski, C., Redivo–Zaglia, M., Rodriguez, G., Seatzu, S.: Extrapolation techniques for ill-conditioned linear systems. Numer. Math. **81**, 1–29 (1998)
5. Brezinski, C., Rodriguez, G., Seatzu, S.: Error estimates for the regularization of least squares problems. Numer. Algorithms **51**, 61–76 (2009)
6. Calvetti, D., Morigi, S., Reichel, L., Sgallari, F.: Tikhonov regularization and the L-curve for large, discrete ill-posed problems. J. Comput. Appl. Math. **123**, 423–446 (2000)
7. Calvetti, D., Reichel, L., Shuibi, A.: Invertible smoothing preconditioners for linear discrete ill-posed problems. Appl. Numer. Math. **54**, 135–149 (2005)
8. Daniel, J.W., Gragg, W.B., Kaufman, L., Stewart, G.W.: Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. Math. Comp. **30**, 772–795 (1976)
9. Donatelli, M., Reichel, L.: Square smoothing regularization matrices with accurate boundary conditions. J. Comput. Appl. Math. **272**, 334–349 (2014)
10. Donatelli, M., Neuman, A., Reichel, L.: Square regularization matrices for large linear discrete ill-posed problems. Numer. Linear Algebra Appl. **19**, 896–913 (2012)
11. Dykes, L., Huang, G., Noschese, S., Reichel, L.: Regularization matrices for discrete ill-posed problems in several space-dimensions. Numerical Linear Algebra Appl. (2018, in press)
12. Elbouyahyaout, L., Messaoudi, A., Sadok, H.: Algebraic properties of the block GMRES and block Arnoldi methods. Electron. Trans. Numer. Anal. **33**, 207–220 (2009)
13. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Dordrecht (1996)
14. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017)
15. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electron. Trans. Numer. Anal. **44**, 83–123 (2015)
16. Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. SIAM, Philadelphia (1998)
17. Hansen, P.C.: Regularization tools version 4.0 for MATLAB 7.3. Numer. Algorithms **46**, 189–194 (2007)
18. Hochstenbach, M.E., Reichel, L.: An iterative method for Tikhonov regularization with a general linear regularization operator. J. Integral Equations Appl. **22**, 463–480 (2010)
19. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)
20. Huang, G., Noschese, S., Reichel, L.: Regularization matrices determined by matrix nearness problems. Linear Algebra Appl. **502**, 41–57 (2016)
21. Jbilou, K., Messaoudi, A., Sadok, H.: Global FOM and GMRES algorithms for matrix equations. Appl. Numer. Math. **31**, 49–63 (1999)
22. Jbilou, K., Sadok, H., Tinzefte, A.: Oblique projection methods for linear systems with multiple right-hand sides. Electron. Trans. Numer. Anal. **20**, 119–138 (2005)
23. Kilmer, M.E., Hansen, P.C., Español, M.I.: A projection-based approach to general-form Tikhonov regularization. SIAM J. Sci. Comput. **29**, 315–330 (2007)
24. Kindermann, S.: Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems. Electron. Trans. Numer. Anal. **38**, 233–257 (2011)
25. Lampe, J., Reichel, L., Voss, H.: Large-scale Tikhonov regularization via reduction by orthogonal projection. Linear Algebra Appl. **436**, 2845–2865 (2012)
26. Lewis, B., Reichel, L.: Arnoldi–Tikhonov regularization methods. J. Comput. Appl. Math. **226**, 92–102 (2009)

27. Li, R.C., Ye, Q.: A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. SIAM J. Matrix Anal. Appl. **25**, 405–428 (2003)
28. Morigi, S., Reichel, L., Sgallari, F.: Orthogonal projection regularization operators. Numer. Algorithms **44**, 99–114 (2007)
29. Noschese, S., Reichel, L.: Inverse problems for regularization matrices. Numer. Algorithms **60**, 531–544 (2012)
30. Reichel, L., Rodriguez, G.: Old and new parameter choice rules for discrete ill-posed problems. Numer. Algorithms **63**, 65–87 (2013)
31. Reichel, L., Yu, X.: Matrix decompositions for Tikhonov regularization. Electron. Trans. Numer. Anal. **43**, 223–243 (2015)
32. Reichel, L., Yu, X.: Tikhonov regularization via flexible Arnoldi reduction. BIT **55**, 1145–1168 (2015)
33. Reichel, L., Rodriguez, G., Seatzu, S.: Error estimates for large-scale ill-posed problems. Numer. Algorithms **51**, 341–361 (2009)
34. Reichel, L., Sgallari, F., Ye, Q.: Tikhonov regularization based on generalized Krylov subspace methods. Appl. Numer. Math. **62**, 1215–1228 (2012)
35. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)