Electronic Transactions on Numerical Analysis.
Volume 43, pp. 223-243, 2015.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

MATRIX DECOMPOSITIONS FOR TIKHONOV REGULARIZATION*

LOTHAR REICHEL† AND XUEBO YU†

Abstract. Tikhonov regularization is a popular method for solving linear discrete ill-posed problems with error-contaminated data. This method replaces the given linear discrete ill-posed problem by a penalized least-squares problem. The choice of the regularization matrix in the penalty term is important. We are concerned with the situation when this matrix is of fairly general form. The penalized least-squares problem can be conveniently solved with the aid of the generalized singular value decomposition, provided that the size of the problem is not too large. However, it is impractical to use this decomposition for large-scale penalized least-squares problems. This paper describes new matrix decompositions that are well suited for the solution of large-scale penalized least-square problems that arise in Tikhonov regularization with a regularization matrix of general form.

Key words. ill-posed problem, matrix decomposition, generalized Krylov method, Tikhonov regularization.

AMS subject classifications. 65F22, 65F10, 65R30

1. Introduction.

We are concerned with the solution of large-scale linear least-squares problems

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x}-\boldsymbol{b}\|, \tag{1.1}$$

with a matrix $A \in \mathbb{R}^{m\times n}$ that has many singular values of different orders of magnitude close to the origin. In particular, A is severely ill-conditioned. The vector $\boldsymbol{b} \in \mathbb{R}^m$ represents measured data and is assumed to be contaminated by an error $\boldsymbol{e} \in \mathbb{R}^m$ that stems from measurement inaccuracies. Throughout this paper, $\|\cdot\|$ denotes the Euclidean vector norm or the associated induced matrix norm.

We can express the data vector as

$$\boldsymbol{b} = \widehat{\boldsymbol{b}} + \boldsymbol{e}, \tag{1.2}$$

where $\widehat{\boldsymbol{b}}$ denotes the unknown error-free vector associated with $\boldsymbol{b}$. The linear system of equations with the unknown right-hand side,

$$A\boldsymbol{x} = \widehat{\boldsymbol{b}}, \tag{1.3}$$

is assumed to be consistent and we denote its solution of minimal Euclidean norm by $\widehat{\boldsymbol{x}}$. We would like to determine an accurate approximation of $\widehat{\boldsymbol{x}}$ by computing a suitable approximate solution of (1.1).

Least-squares problems (1.1) with a matrix whose singular values "cluster" at zero are commonly referred to as linear discrete ill-posed problems. They arise in image deblurring problems as well as from the discretization of linear ill-posed problems such as Fredholm integral equations of the first kind with a continuous kernel. Due to the ill-conditioning of A and the error $\boldsymbol{e}$ in $\boldsymbol{b}$, straightforward solution of (1.1) generally does not give a meaningful approximation of $\widehat{\boldsymbol{x}}$.

A common approach to determine a useful approximate solution of (1.1) is to employ Tikhonov regularization, i.e., to replace (1.1) by a penalized least-squares problem of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{\|A\boldsymbol{x}-\boldsymbol{b}\|^2 + \mu\|B\boldsymbol{x}\|^2\}, \tag{1.4}$$

*Received February 20, 2014. Accepted June 12, 2015. Published online on June 19, 2015. Recommended by H. Sadok. The research of L. R. and X. Y. is supported in part by NSF grant DMS-1115385.
†Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA ({reichel, xyu}@math.kent.edu).

ETNA
Kent State University
http://etna.math.kent.edu

where the matrix $B \in \mathbb{R}^{p\times n}$ is a regularization matrix and the scalar $\mu \geq 0$ is a regularization parameter. When B is the identity matrix, the Tikhonov minimization problem (1.4) is said to be in *standard form*; otherwise it is in *general form*. We assume that B is such that

$$\mathcal{N}(A) \cap \mathcal{N}(B) = \{\mathbf{0}\}, \tag{1.5}$$

where $\mathcal{N}(M)$ denotes the null space of the matrix M.

The normal equations associated with (1.4) are given by

$$(A^*A + \mu B^*B)\boldsymbol{x} = A^*\boldsymbol{b}, \tag{1.6}$$

where A^* and B^* denote the adjoints of A and B, respectively. It follows from (1.5) and (1.6) that (1.4) has the unique solution

$$\boldsymbol{x}_\mu = (A^*A + \mu B^*B)^{-1}A^*\boldsymbol{b}$$

for any $\mu > 0$. The value of μ determines how sensitive $\boldsymbol{x}_\mu$ is to the error $\boldsymbol{e}$ in $\boldsymbol{b}$ and to round-off errors introduced during the computations and how close $\boldsymbol{x}_\mu$ is to $\widehat{\boldsymbol{x}}$; see, e.g., Engl et al. [6], Groetsch [8], and Hansen [9] for discussions on Tikhonov regularization.

We would like to determine a suitable value of the regularization parameter $\mu > 0$ and an approximation of the associated solution $\boldsymbol{x}_\mu$ of (1.4). The determination of a suitable μ generally requires that the Tikhonov minimization problem (1.4) be solved for several μ-values. For instance, the discrepancy principle, the L-curve criterion, and the generalized cross validation method are popular approaches to determine a suitable value of μ, and all of them require that (1.4) be solved for several values of $\mu > 0$ in order to find an appropriate value; see, e.g., [6, 9, 14, 15, 17] and the references therein for discussions on these methods for determining μ. The repeated solution of (1.4) for different μ-values can be expensive when the matrices A and B are large and do not possess a structure that makes a fast solution possible.

When the matrices A and B are of small to moderate sizes, the Tikhonov minimization problem (1.4) is typically simplified by first computing the Generalized Singular Value Decomposition (GSVD) of the matrix pair $\{A, B\}$ or a related decomposition; see [3, 4, 9]. When one of the latter decompositions is available, the minimization problem (1.4) can be solved quite inexpensively for several different μ-values.

In this paper, we are interested in developing solution methods that can be applied when the matrices A and B are too large to compute the GSVD or a related decomposition of the matrix pair $\{A, B\}$. Moreover, B is not assumed to have a particular structure that makes the transformation of the problem (1.4) to standard form with the aid of the A-weighted generalized inverse of B feasible; see Eldén [5] for details on this transformation. We describe decomposition methods for the matrices A and B that are well suited for the approximate solution of large-scale Tikhonov minimization problems (1.4) in general form. These methods reduce a pair of large matrices $\{A, B\}$ to a pair of small matrices and, thereby, reduce the large-scale problem (1.4) to a small one. The GSVD or the decomposition described in [3] can be applied to solve the latter for several values of the regularization parameter. The reduction methods considered in this paper are modifications of decomposition schemes described in [12, 20]. The decomposition discussed in [12] is a generalization of Golub–Kahan bidiagonalization to matrix pairs. We describe a variant that allows the generation of more general solution subspaces than those considered in [12]. Computed examples illustrate that this extension may be beneficial. We also discuss an extension of the decomposition method described in [20], which is based on the flexible Arnoldi process introduced by Saad [21]. This decomposition method is designed for square matrices A and B of the same size. We

ETNA
Kent State University
http://etna.math.kent.edu

consider an extension that allows B to be rectangular. This extension is briefly commented on in the last computed example of [20]. The present paper discusses its implementation and illustrates its performance in two computed examples.

This paper is organized as follows. Sections 2 and 3 describe the new decomposition methods and discuss some of their properties. Numerical examples are presented in Section 4, and concluding remarks can be found in Section 5.

We conclude this section with a few comments on some available methods for the solution of large-scale Tikhonov minimization problems in general form. Kilmer et al. [13] describe an inner-outer iterative method. This method is based on the partial GSVD method described by Zha [24]. The latter method may require a fairly large number of matrix-vector product evaluations. We therefore are interested in developing alternative methods. A reduction method that forms a solution subspace that is independent of the matrix B is proposed in [11]. This approach is simple and works well for many problems, but as is illustrated in [20], it may be beneficial to use a solution subspace that incorporates information from both the matrices A and B. A generalization of the Arnoldi process that can be applied to the reduction of a pair of square matrices of the same sizes has been discussed by Li and Ye [16], and applications to Tikhonov regularization are described in [16, 18]. This reduction method requires the matrices A and B to be square.

We will use the following notation throughout the paper: $M_{k,\ell}$ denotes a matrix of size $k\times\ell$, its entries are $m_{i,j}$. We use MATLAB-type notation: $M_{:,j}$ is the jth column and $M_{i,:}$ the ith row of the matrix $M = M_{k,\ell}$. The submatrix consisting of rows i through j and columns k through ℓ is denoted by $M_{i:j,k:\ell}$. Sometimes the number of rows of a matrix is suppressed, i.e., we write $M_\ell = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_\ell]$ for a matrix with ℓ columns. Boldface letters stand for column vectors. The range of the matrix M is denoted by $\mathcal{R}(M)$. The condition number of the matrix M, denoted by $\kappa(M)$, is the quotient of the largest and smallest singular values of the matrix. Moreover, $(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^*\boldsymbol{v}$ stands for the standard inner product between the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

2. Golub–Kahan-type decomposition methods. The application of a few steps of Golub–Kahan bidiagonalization (also known as Lanczos bidiagonalization) is a popular approach to reduce a large matrix to a small bidiagonal one. Recently, Hochstenbach et al. [12] described an extension of Golub–Kahan bidiagonalization that can be applied to reduce a pair of large matrices $\{A, B\}$ with the same number of columns to a pair of small matrices. This extension builds up a solution subspace that is constructed by invoking matrix-vector products with A and A^* in essentially the same manner as matrix-vector products with B and B^*. Algorithm 2.1 below describes a modification of the method presented in [12] that allows the construction of more general solution subspaces. Computed examples presented in Section 4 illustrate that the method of this paper may determine approximations of $\widehat{\boldsymbol{x}}$ of higher quality than the method described in [12].

We first discuss the standard Golub–Kahan method for partial bidiagonalization of one matrix $A \in \mathbb{R}^{m\times n}$. An outline of the bidiagonalization process is provided in the proof of Proposition 2.1 because related constructions are employed below. We introduce notation that is convenient for our subsequent generalization. Detailed discussions of Golub–Kahan bidiagonalization can be found in, e.g., [2, 7].

PROPOSITION 2.1. *Let $A \in \mathbb{R}^{m\times n}$ and $\boldsymbol{u}_1 \in \mathbb{R}^m$ be a unit vector. Then $k \le \min\{m, n\}$ steps of Golub–Kahan bidiagonalization applied to A with initial vector $\boldsymbol{u}_1$ yield the decompositions*

$$AV_k = U_{k+1}H_{k+1,k}, \tag{2.1}$$

$$A^*U_{k+1} = V_{k+1}K_{k+1,k+1}, \tag{2.2}$$

ETNA
Kent State University
http://etna.math.kent.edu

where the columns of the matrices $U_{k+1} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{k+1}] \in \mathbb{R}^{m\times(k+1)}$ *and* $V_{k+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)}$ *are orthonormal, the matrix* $H_{k+1,k} \in \mathbb{R}^{(k+1)\times k}$ *is lower bidiagonal, and the leading* $k \times (k+1)$ *submatrix of* $K_{k+1,k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$ *satisfies*

$$K_{k,k+1} = H_{k+1,k}^*. \tag{2.3}$$

The initial column $\boldsymbol{v}_1$ *of* V_{k+1} *is determined by* (2.2) *with* $k = 0$ *so that* $k_{1,1} > 0$. *Generically, the diagonal and subdiagonal entries of* $H_{k+1,k}$ *can be chosen to be positive. The decompositions* (2.1) *and* (2.2) *then are uniquely determined.*

Proof. The columns $\boldsymbol{u}_2, \boldsymbol{v}_2, \boldsymbol{u}_3, \boldsymbol{v}_3, \ldots$, of U_{k+1} and V_{k+1} are generated, in order, by alternatingly using equations (2.1) and (2.2) for increasing values of k. Thus, the column $\boldsymbol{u}_2$ is determined by requiring $\boldsymbol{u}_2$ to be of unit length, to be orthogonal to $\boldsymbol{u}_1$, and to satisfy (2.1) for $k = 1$ for a positive subdiagonal entry $h_{2,1}$ of $H_{2,1}$, where we note that $h_{1,1} = k_{1,1}$. This determines both $\boldsymbol{u}_2$ and $h_{2,1}$. The column $\boldsymbol{v}_2$ of V_2 is now defined by equation (2.2) for $k = 2$. The column is uniquely determined by the requirements that $\boldsymbol{v}_2$ be orthogonal to $\boldsymbol{v}_1$, of unit length, and such that the last diagonal entry of $K_{2,2}$ is positive. This entry equals $h_{2,2}$. The next vector to be evaluated is $\boldsymbol{u}_3$. Generically, the computations can be continued in the manner indicated until the decompositions (2.1) and (2.2) have been computed for some $k \leq \min\{m, n\}$.

In rare situations, the computations cannot be completed as described because the first, say j, generated columns $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j$ of V_{k+1} span an invariant subspace of A^*A. This situation is referred to as breakdown. The computations can be continued by letting the next column, $\boldsymbol{v}_{j+1}$, be an arbitrary unit vector that is orthogonal to $\operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j\}$. The situation when the first j generated columns of U_{k+1} span an invariant subspace of AA^* can be handled analogously and is also referred to as breakdown. In case of breakdown, suitable entries of $H_{k+1,k}$ and $K_{k+1,k+1}$ are set to zero so that the decompositions (2.1) and (2.2) are valid. These decompositions are not unique when breakdown occurs. □

In applications of partial Golub–Kahan bidiagonalization to the solution of least-squares problems (1.1), one generally chooses the initial vector $\boldsymbol{u}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$. Available descriptions of Golub–Kahan bidiagonalization exploit that $K_{k,k+1}$ can be expressed in terms of $H_{k+1,k}$, see (2.3), and do not explicitly use the matrix $K_{k+1,k+1}$. It is convenient for our discussion below to distinguish between the matrices $H_{k+1,k}$ and $K_{k,k+1}$.

We now turn to a modification of Golub–Kahan bidiagonalization that allows the choice of a fairly arbitrary column $\boldsymbol{v}_{i+1}$ in addition to the column $\boldsymbol{u}_1$. The matrices U_{k+1} and V_{k+1} generated will have orthonormal columns, similarly as in the decompositions (2.1) and (2.2), but the structure of the matrices analogous to $H_{k+1,k}$ and $K_{k+1,k+1}$ in (2.1) and (2.2) will be different. We assume for notational simplicity the generic situation that no breakdown takes place.

Let the decompositions (2.1) and (2.2) be available for $k = i - 1$, i.e., we have

$$AV_{i-1} = U_i H_{i,i-1}, \tag{2.4}$$

$$A^*U_i = V_i K_{i,i}, \tag{2.5}$$

with $K_{i-1,i} = H_{i,i-1}^*$. Determine the column $\boldsymbol{u}_{i+1}$ of U_{k+1} from (2.1) with $k = i$. This defines the entry $h_{i+1,i} > 0$ of $H_{i+1,i}$. Now let the column $\boldsymbol{v}_{i+1}$ of V_{k+1} be an arbitrary unit vector such that

$$\boldsymbol{v}_{i+1} \perp \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_i\}. \tag{2.6}$$

We proceed to compute the column $\boldsymbol{u}_{i+2}$ of U_{k+1} by using (2.1) with $k = i + 1$. This determines the last column of $H_{i+2,i+1}$. We will show below that all entries above the diagonal in

ETNA
Kent State University
http://etna.math.kent.edu

the column $H_{:,i+1}$ vanish. The column $\boldsymbol{v}_{i+2}$ of V_{i+2} is chosen to be of unit length, orthogonal to the columns of V_{i+1}, and such that the relation

$$A^*U_{i+1} = V_{i+2}K_{i+2,i+1}$$

holds. We then compute the columns $\boldsymbol{u}_{i+3}, \boldsymbol{v}_{i+3}, \ldots, \boldsymbol{u}_{k+1}, \boldsymbol{v}_{k+1}$, in order, from decompositions of the form

$$AV_j = U_{j+1}H_{j+1,j}, \tag{2.7}$$

$$A^*U_j = V_{j+1}K_{j+1,j}, \tag{2.8}$$

for $j = i+2, i+3, \ldots, k$. The following theorem describes the structure of the matrices $H_{k+1,k}$ and $K_{k+1,k}$.

THEOREM 2.2. *Let the decompositions* (2.7) *and* (2.8) *for* $j = k$ *be generated as described above and assume that no breakdown occurs. Then the columns of* $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ *and* $V_{k+1} \in \mathbb{R}^{n\times(k+1)}$ *are orthonormal, and the matrix* $H_{k+1,k} \in \mathbb{R}^{(k+1)\times k}$ *has the structure*

$$H_{k+1,k} = \begin{bmatrix} h_{1,1} & & & & & & & O \\ h_{2,1} & h_{2,2} & & & & & & \\ & h_{3,2} & \ddots & & & & & \\ & & \ddots & h_{i+1,i+1} & h_{i+1,i+2} & & & \\ & & & h_{i+2,i+1} & h_{i+2,i+2} & \ddots & & \\ & & & & h_{i+3,i+2} & \ddots & h_{k-1,k} & \\ & & & & & \ddots & h_{k,k} & \\ O & & & & & & h_{k+1,k} & \end{bmatrix}.$$

Thus, the leading principal $(i+2)\times(i+1)$ *submatrix is lower bidiagonal and the matrix* $H_{k+1,k}$ *is tridiagonal. Furthermore,* $K_{k,k} = H_{k,k}^*$.

Proof. Let the decompositions (2.4) and (2.5) be available. The matrix $H_{i,i-1}$ in (2.4) is lower bidiagonal by Proposition 2.1. The next step in the Golub–Kahan bidiagonalization method is to replace V_{i-1} by V_i in (2.4) and define the matrix U_{i+1} by appending a suitable column $\boldsymbol{u}_{i+1}$ to U_i. Append a zero row to $H_{i,i-1}$ and the column $[h_{1,i}, h_{2,i}, \ldots, h_{i+1,i}]^*$ to the matrix so obtained. This gives a decomposition of the form (2.4) with i replaced by $i+1$. The entries $h_{j,i}$ are defined by

$$A\boldsymbol{v}_i = \sum_{j=1}^{i+1} h_{j,i}\boldsymbol{u}_j, \tag{2.9}$$

where we choose $h_{i+1,i} > 0$ so that $\boldsymbol{u}_{i+1}$ is a unit vector that is orthogonal to $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_i$. It follows from (2.5) and the fact that $K_{i,i}$ is upper triangular that

$$A^*\boldsymbol{u}_j \in \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j\}, \qquad j = 1, 2, \ldots, i. \tag{2.10}$$

Therefore,

$$h_{j,i} = \boldsymbol{u}_j^* A\boldsymbol{v}_i = \boldsymbol{v}_i^*(A^*\boldsymbol{u}_j) = 0, \qquad j = 1, 2, \ldots, i-1.$$

The last diagonal entry of $H_{i+1,i}$ is determined by (2.5), i.e., by

$$h_{i,i} = \boldsymbol{u}_i^* A\boldsymbol{v}_i = k_{i,i}.$$

ETNA
Kent State University
http://etna.math.kent.edu

Thus, we have obtained the desired decomposition

$$AV_i = U_{i+1}H_{i+1,i},$$

where the columns of U_{i+1} are orthonormal and $H_{i+1,i}$ is lower bidiagonal.

Let $\boldsymbol{v}_{i+1}$ be a unit vector that satisfies (2.6). We proceed as above to determine a new unit vector $\boldsymbol{u}_{i+2}$ that we append to U_{i+1} to determine the matrix U_{i+2} with orthonormal columns. Append a zero row to $H_{i+1,i}$ and the column $[h_{1,i+1}, h_{2,i+1}, \ldots, h_{i+2,i+1}]^*$ to the matrix so obtained. Our aim is to determine a decomposition of the form (2.4) with i replaced by $i+1$. Therefore, analogously to (2.9), we let

$$A\boldsymbol{v}_{i+1} = \sum_{j=1}^{i+2} h_{j,i+1}\boldsymbol{u}_j$$

and choose $h_{i+2,i+1} > 0$ so that $\boldsymbol{u}_{i+2}$ is a unit vector that is orthogonal to the vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{i+1}$. It follows from (2.10) that

$$h_{j,i+1} = \boldsymbol{u}_j^* A\boldsymbol{v}_{i+1} = \boldsymbol{v}_{i+1}^*(A^*\boldsymbol{u}_j) = 0, \qquad j = 1, 2, \ldots, i.$$

The remaining entry of the last column of $H_{i+1,i}$ is defined by $h_{i+1,i+1} = \boldsymbol{u}_{i+1}^* A\boldsymbol{v}_{i+1}$. Thus, we have determined the decomposition

$$AV_{i+1} = U_{i+2}H_{i+2,i+1}, \tag{2.11}$$

where the columns of U_{i+2} and V_{i+1} are orthonormal and $H_{i+2,i+1}$ is lower bidiagonal.

To proceed, let $\boldsymbol{v}_{i+2}$ be a unit vector that is orthogonal to $\operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{i+1}\}$ and satisfies

$$A^*\boldsymbol{u}_{i+1} = \sum_{j=1}^{i+2} k_{j,i+1}\boldsymbol{v}_j$$

with $k_{i+2,i+1} > 0$. It follows from (2.11) and the structure of $H_{i+2,i+1}$ that $k_{j,i+1} = 0$ for $j = 1, 2, \ldots, i-1$. We first append two zero rows to the matrix K_i and then the column $[k_{1,i+1}, k_{2,i+1}, \ldots, k_{i+2,i+1}]^*$ to the matrix so obtained. This defines the matrix $K_{i+2,i+1}$. By construction, it satisfies

$$A^*U_{i+1} = V_{i+2}K_{i+2,i+1}. \tag{2.12}$$

Hence, the matrix V_{i+2} has orthonormal columns, the last column of $K_{i+2,i+1}$ has at most three nonvanishing entries, and the submatrix $K_{i+1,i+1}$ is upper bidiagonal.

We continue to define the column $\boldsymbol{u}_{i+3}$ of the matrix U_{i+3} with the aim of obtaining a decomposition of the form

$$AV_{i+2} = U_{i+3}H_{i+3,i+2}.$$

Specifically, we let $\boldsymbol{u}_{i+3}$ be of unit length, orthogonal to $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{i+2}$, and such that

$$A\boldsymbol{v}_{i+2} = \sum_{i=1}^{i+3} h_{j,i+2}\boldsymbol{u}_j$$

with $h_{i+3,i+2} > 0$. It follows from (2.12) and the structure of $K_{i+2,i+1}$ that

$$h_{j,i+2} = \boldsymbol{u}_j^* A\boldsymbol{v}_{i+2} = 0, \qquad j = 1, 2, \ldots, i.$$

ETNA
Kent State University
http://etna.math.kent.edu

Hence, only the last three entries of the vector $[h_{1,i+2}, h_{2,i+2}, \ldots, h_{i+3,i+2}]^*$, which is the last column of the matrix $H_{i+3,i+2}$, may be nonvanishing.

We proceed by defining new columns of the matrices U_{k+1} and V_{k+1} in this manner until the decompositions (2.7) and (2.8) have been determined for $j = k$. ∎

Results analogous to Theorem 2.2 can be obtained by letting the column $\boldsymbol{u}_{i+1}$ of U_{k+1} be an arbitrary unit vector that is orthogonal to the preceding columns $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_i$. We will not dwell on this situation since it is of little interest for our numerical method for Tikhonov regularization.

The special case of Theorem 2.2 when both the initial columns of U_{k+1} and V_{k+1} are chosen to be arbitrary unit vectors is described by the following corollary. It has previously been discussed in [19, 22].

COROLLARY 2.3. *Let the initial columns of the matrices $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ and $V_{k+1} \in \mathbb{R}^{n\times(k+1)}$ be arbitrary unit vectors. Determine the remaining columns similarly as in Theorem 2.2 and assume that no breakdown occurs. Then the matrices U_{k+1} and V_{k+1} satisfy the relations*

$$AV_k = U_{k+1}H_{k+1,k},$$
$$A^*U_k = V_{k+1}K_{k+1,k},$$

where $H_{k+1,k} \in \mathbb{R}^{(k+1)\times k}$ is tridiagonal and $K_{k,k} = H_{k,k}^*$.

Proof. The result is a consequence of Theorem 2.2. Breakdown of the recursions is discussed in [19]. ∎

We can extend Theorem 2.2 to allow the inclusion of several arbitrary orthonormal columns in the matrix V_{k+1}.

THEOREM 2.4. *Let the indices i_j be ordered so that $1 \le i_1 < i_2 < \ldots < i_s \le k$, and let $\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \ldots, \boldsymbol{v}_{i_s}$ be arbitrary unit vectors such that $\boldsymbol{v}_{i_\ell}$ is orthogonal to all preceding columns $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{i_\ell-1}$ of V_{k+1} for $\ell = 1, 2, \ldots, s$. Introducing these columns similarly as the column $\boldsymbol{v}_{i+1}$ in Theorem 2.2 yields the decompositions*

$$AV_k = U_{k+1}H_{k+1,k},$$
$$A^*U_{k+1-s} = V_{k+1}K_{k+1,k+1-s},$$

where $U_{k+1} \in \mathbb{R}^{m\times(k+1)}$ and $V_{k+1} \in \mathbb{R}^{n\times(k+1)}$ have orthonormal columns. The matrices $H_{k+1,k} \in \mathbb{R}^{(k+1)\times k}$ and $K_{k+1,k+1-s} \in \mathbb{R}^{(k+1)\times(k+1-s)}$ are banded and satisfy $(H_{k+1-s,k})^* = K_{k,k+1-s}$. Moreover, $H_{k+1,k}$ is upper Hessenberg and such that

- *all entries except possibly $h_{j+1,j}$ and $h_{j,j}$ of the column $H_{:,j}$ vanish for $j \le i_1$,*
- *all entries except possibly $h_{j+1,j}, h_{j,j}, \ldots, h_{j-t,j}$ of the column $H_{:,j}$ vanish for $i_t < j \le i_{t+1}$, where $1 \le t \le s-1$,*
- *all entries except possibly $h_{j+1,j}, h_{j,j}, \ldots, h_{j-s,j}$ of the column $H_{:,j}$ vanish for $j > i_s$.*

Proof. Theorem 2.2 shows that when introducing an arbitrary unit vector $\boldsymbol{v}_{i_1}$ that is orthogonal to the preceding vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{i_1-1}$, the upper bandwidth of the matrix $H_{k+1,k}$ increases by one, starting at column $i_1 + 1$. A slight modification of the proof of Theorem 2.2 shows that if a new arbitrary unit vector $\boldsymbol{v}_{i_2}$ that is orthogonal to the preceding vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{i_2-1}$ is introduced, then the upper bandwidth of $H_{k+1,k}$ is increased by one, starting at column $i_2 + 1$. Repeating this process for all vectors $\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \ldots, \boldsymbol{v}_{i_s}$ shows the theorem. ∎

The above theorem forms the basis for our generalized Golub–Kahan reduction method for matrix pairs $\{A, B\}$. We first present an outline of this method. A detailed algorithm is

ETNA
Kent State University
http://etna.math.kent.edu

presented in Algorithm 2.1. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, and let the first column $\boldsymbol{u}_1$ of the matrix $U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots\ ]$ be an arbitrary unit vector. Define the first column of the matrix $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots\ ]$ by $\boldsymbol{v}_1 = A^*\boldsymbol{u}_1/\|A^*\boldsymbol{u}_1\|$. Let the matrix U_j consist of the first j columns of U; similar notation is used for the matrices V and W. Further, $H^{(A)}_{j+1,j}$ denotes the leading principal $(j+1)\times j$ submatrix of the matrix $H^{(A)}$ defined below. We use the same notation for the matrices $H^{(B)}$, $K^{(A)}$, and $K^{(B)}$ also defined below. Let the $(1,1)$-entries of $H^{(A)}$ and $K^{(A)}$ be given by $h^{(A)}_{1,1} = k^{(A)}_{1,1} = \|A^*\boldsymbol{u}_1\|$. The index sets $\mathbb{P}_A$ and $\mathbb{P}_B$ keep track of how new vectors in the solution subspace $\mathcal{R}(V)$ are generated; the integers s_A and s_B are associated counters. We generate successive columns of the matrices U, V, W, $H^{(A)}$, $H^{(B)}$, $K^{(A)}$, and $K^{(B)}$ in the manner described in the Algorithm Outline 2.1.

ALGORITHM OUTLINE 2.1.

Initialization:

$s_A = 1$; $s_B = 0$; $\mathbb{P}_A = \{1\}$; $\mathbb{P}_B = \emptyset$; define the vectors $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ as described above.

Iteration:

for $j = 1, 2, 3, \dots$:

- Determine the new $(j+1)$st and jth columns $\boldsymbol{u}_{j+1}$ and $\boldsymbol{w}_j$, respectively, of the matrices U and W by equating

$$AV_j = U_{j+1}H^{(A)}_{j+1,j},$$
$$BV_j = W_jH^{(B)}_{j,j},$$

  so that the matrices U_{j+1} and W_j have orthonormal columns, $H^{(A)}_{j+1,j} \in \mathbb{R}^{(j+1)\times j}$ is upper Hessenberg, and $H^{(B)}_{j,j} \in \mathbb{R}^{j\times j}$ is upper triangular.

- Determine the new $(j+1)$st column $\boldsymbol{v}_{j+1}$ of the matrix V by equating one of the following formulas that define decompositions and by carrying out the other required computations

  (i): $A^*U_{j+1-s_B} = V_{j+1}K^{(A)}_{j+1,j+1-s_B}$; $\quad s_A = s_A+1$; $\mathbb{P}_A = \mathbb{P}_A \cup \{j+1\}$;

  (ii): $B^*W_{j+1-s_A} = V_{j+1}K^{(B)}_{j+1,j+1-s_A}$; $\quad s_B = s_B+1$; $\mathbb{P}_B = \mathbb{P}_B \cup \{j+1\}$;

  so that V_{j+1} has orthonormal columns. Here, the matrices $K^{(A)}_{j+1,j+1-s_B} \in \mathbb{R}^{(j+1)\times(j+1-s_B)}$ and $K^{(B)}_{j+1,j+1-s_A} \in \mathbb{R}^{(j+1)\times(j+1-s_A)}$ have zero entries below the diagonal. The indices s_A and s_B count the number of columns of V that have been determined by equating (i) and (ii), respectively. Thus, $s_A + s_B = j$. The index sets $\mathbb{P}_A$ and $\mathbb{P}_B$ are used in Theorem 2.5 below.

end j-loop

THEOREM 2.5. *Let $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, and let the first columns of the matrices U and V be defined as in the Algorithm Outline 2.1. Then, assuming that no breakdown occurs, k iteration steps described by Algorithm Outline 2.1 yield the decompositions*

$$AV_k = U_{k+1}H^{(A)}_{k+1,k}, \tag{2.13}$$

$$BV_k = W_kH^{(B)}_{k,k}, \tag{2.14}$$

$$A^*U_{k+1-s_B} = V_{k+1}K^{(A)}_{k+1,k+1-s_B}, \tag{2.15}$$

$$B^*W_{k+1-s_A} = V_{k+1}K^{(B)}_{k+1,k+1-s_A}. \tag{2.16}$$

ETNA
Kent State University
http://etna.math.kent.edu

Assume that the vectors $\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \ldots, \boldsymbol{v}_{i_{s_B}}$ are generated by using equation (ii). Then $i_1 > 1$, and $H^{(A)}$ has the structure described by Theorem 2.4 with the indices $i_1 < i_2 < \ldots < i_s$ defined by $\mathbb{P}_B = \{i_j\}_{j=1}^{s}$.

Let the indices $k_1 < k_2 < \ldots < k_t$ be defined by $\mathbb{P}_A = \{k_j\}_{j=1}^{t}$. Then $k_1 = 1$, and the matrix $H^{(B)}$ is upper triangular and such that the column $H_{:,j}^{(B)}$ has at most $1 + q$ nonzero entries $h_{j,j}^{(B)}, h_{j+1,j}^{(B)}, \ldots, h_{j+q,j}^{(B)}$ for $i_q < j \leq i_{q+1}$.

Proof. The structure of $H^{(A)}$ is obtained from Theorem 2.4 by replacing s by s_B and by letting the vectors $\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \ldots, \boldsymbol{v}_{i_s}$ of Theorem 2.4 be $\boldsymbol{v}_{i_1}, \boldsymbol{v}_{i_2}, \ldots, \boldsymbol{v}_{i_{s_B}}$ of the present theorem.

The structure of $H^{(B)}$ can be shown similarly as the structure of $H^{(A)}$ as follows. Consider the two-part iteration (2.14) and (2.16) to generate the first $i-1$ columns of V. This is Golub–Kahan bidiagonalization with the initial vector $\boldsymbol{v}_1$. The matrix $H^{(B)}$ determined is upper bidiagonal. Now let $\boldsymbol{v}_i$ be determined by (2.15) for $i = \ell_1, \ell_2, \ldots, \ell_{s_A}$. We apply Theorem 2.2 repeatedly, similarly as in the proof of Theorem 2.4, to show the structure of $H^{(B)}$. ∎

Algorithm 2.1 below describes a particular implementation of the Algorithm Outline 2.1, in which a parameter $\rho > 0$ determines whether step (i) or step (ii) should be executed to determine a new column of V. The value of ρ affects the solution subspace $\mathcal{R}(V)$ generated by the algorithm. This is illustrated below. Algorithm 2.1 generalizes [12, Algorithm 2.2] by allowing step (i) to be executed a different number of times than step (ii). The algorithm in [12] corresponds to the case $\rho = 1$.

The counters $N(\boldsymbol{u})$ and $N(\boldsymbol{w})$ in Algorithm 2.1 are indices used when generating the next column of V.

EXAMPLE 2.6. Let $\rho = 1$. Then ℓ steps with Algorithm 2.1 generates the matrix V_ℓ with range

$$\mathcal{R}(V_\ell) = \operatorname{span}\{A^*\boldsymbol{b}, B^*BA^*\boldsymbol{b}, A^*AA^*\boldsymbol{b}, (B^*B)^2A^*\boldsymbol{b}, A^*AB^*BA^*\boldsymbol{b}, B^*BA^*AA^*\boldsymbol{b}, (A^*A)^2A^*\boldsymbol{b}, \ldots\}.$$

This space also is determined by the generalized Golub–Kahan reduction method described by [12, Algorithm 2.2].

EXAMPLE 2.7. Let $\rho = 1/2$. Then each application of step (i) is followed by two applications of step (ii). This yields a subspace of the form

$$\mathcal{R}(V_\ell) = \operatorname{span}\{A^*\boldsymbol{b}, B^*BA^*\boldsymbol{b}, (B^*B)^2A^*\boldsymbol{b}, A^*AA^*\boldsymbol{b}, (B^*B)^3A^*\boldsymbol{b}, B^*BA^*AA^*\boldsymbol{b}, A^*AB^*BA^*\boldsymbol{b}, \ldots\}.$$

The computation of the matrix V_ℓ in this example requires more matrix-vector product evaluations with the matrix B^* than the determination of the corresponding matrix V_ℓ of Example 2.6. In many applications of Tikhonov regularization, B^* represents a discretization of a differential operator and is sparse. Typically, the evaluation of matrix-vector products with B^* is cheaper than with A^*. Therefore, the computation of a solution subspace of dimension ℓ generally is cheaper when $\rho = 1/2$ than when $\ell = 1$. Moreover, computed examples of Section 4 show that $\rho < 1$ may yield more accurate approximations of the desired solution $\widehat{\boldsymbol{x}}$ than $\rho = 1$.

Algorithm 2.1 is said to break down if an entry $h_{j+1,j}$, $r_{j,j}$, or α_j in lines 10, 16, or 26 vanishes. Breakdown is very unusual in our application to Tikhonov regularization. If breakdown occurs in lines 10 or 16, then we may terminate the computations with the algorithm and solve the available reduced problem. When breakdown takes place in line 26, we ignore the computed vector $\boldsymbol{v}$ and generate a new vector $\boldsymbol{v}$ via either line 18 or line 20. Breakdown

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHM 2.1 (Extension of Golub–Kahan-type reduction to matrix pairs $\{A, B\}$).

1. Input: matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times n}$, unit vector $\boldsymbol{u}_1 \in \mathbb{R}^n$, ratio $\rho \geq 0$, and number of steps ℓ
2. $\widehat{\boldsymbol{v}} := A^*\boldsymbol{u}_1$; $h_{1,1} := \|\widehat{\boldsymbol{v}}\|$; $\boldsymbol{v}_1 := \widehat{\boldsymbol{v}}/h_{1,1}$;
3. $N(\boldsymbol{u}) := 1$; $N(\boldsymbol{w}) := 1$
4. **for** $j = 1, 2, \ldots, \ell$ **do**
5. $\quad\widehat{\boldsymbol{u}} := A\boldsymbol{v}_j$ — new $\boldsymbol{u}$-vector
6. $\quad$**for** $i = 1, 2, \ldots, j$ **do**
7. $\quad\quad h_{i,j} := \boldsymbol{u}_i^*\widehat{\boldsymbol{u}}$; $\widehat{\boldsymbol{u}} := \widehat{\boldsymbol{u}} - \boldsymbol{u}_i h_{i,j}$
8. $\quad$**end for**
9. $\quad h_{j+1,j} := \|\widehat{\boldsymbol{u}}\|$
10. $\quad\boldsymbol{u}_{j+1} := \widehat{\boldsymbol{u}}/h_{j+1,j}$ — if $h_{j+1,j} = 0$: see text
11. $\quad\widehat{\boldsymbol{w}} := B\boldsymbol{v}_j$ — new $\boldsymbol{w}$-vector
12. $\quad$**for** $i = 1, 2, \ldots, j-1$ **do**
13. $\quad\quad r_{i,j} := \boldsymbol{w}_i^*\widehat{\boldsymbol{w}}$; $\widehat{\boldsymbol{w}} := \widehat{\boldsymbol{w}} - \boldsymbol{w}_i r_{i,j}$
14. $\quad$**end for**
15. $\quad r_{j,j} := \|\widehat{\boldsymbol{w}}\|$
16. $\quad\boldsymbol{w}_j := \widehat{\boldsymbol{w}}/r_{j,j}$ — if $r_{j,j} = 0$: see text
17. $\quad$**if** $N(\boldsymbol{w})/N(\boldsymbol{u}) > 1/\rho$
18. $\quad\quad N(\boldsymbol{u}) := N(\boldsymbol{u}) + 1$; $\boldsymbol{v} := A^*\boldsymbol{u}_{N(\boldsymbol{u})}$
19. $\quad$**else**
20. $\quad\quad \boldsymbol{v} := B^*\boldsymbol{w}_{N(\boldsymbol{w})}$; $N(\boldsymbol{w}) := N(\boldsymbol{w}) + 1$
21. $\quad$**end**
22. $\quad$**for** $i = 1, 2, \ldots, j$ **do**
23. $\quad\quad \boldsymbol{v} := \boldsymbol{v} - (\boldsymbol{v}_i^*\boldsymbol{v})\boldsymbol{v}_i$;
24. $\quad$**end for**
25. $\quad\alpha_j := \|\boldsymbol{v}\|$;
26. $\quad\boldsymbol{v}_{j+1} := \boldsymbol{v}/\alpha_j$; — new $\boldsymbol{v}$-vector, if $h_{j+1,j} = 0$: see text
27. **end for**

also could be handled in other ways. The occurrence of breakdown may affect the structure of the matrices $H^{(A)}$ and $H^{(B)}$.

Let $\boldsymbol{u}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$ in Algorithm 2.1 and assume that no breakdown occurs during the execution of the algorithm. Execution of ℓ steps of Algorithm 2.1 then yields the decompositions (2.13) and (2.14) for $k = \ell$. These decompositions determine the reduced Tikhonov minimization problem

$$\min_{\boldsymbol{x}\in\mathcal{R}(V_\ell)} \{\|A\boldsymbol{x} - \boldsymbol{b}\|^2 + \mu\|B\boldsymbol{x}\|^2\} = \min_{\boldsymbol{y}\in\mathbb{R}^\ell} \{\|H_{\ell+1,\ell}^{(A)}\boldsymbol{y} - \boldsymbol{e}_1\|\boldsymbol{b}\|\,\|^2 + \mu\|H_{\ell,\ell}^{(B)}\boldsymbol{y}\|^2\}. \tag{2.17}$$

It follows from (1.5) that

$$\mathcal{N}(H_{\ell+1,\ell}^{(A)}) \cap \mathcal{N}(H_{\ell,\ell}^{(B)}) = \{\boldsymbol{0}\},$$

and therefore the reduced Tikhonov minimization problem on the right-hand side of (2.17) has a unique solution $\boldsymbol{y}_{\ell,\mu}$ for all $\mu > 0$. The corresponding approximate solution of (1.4) is given by $\boldsymbol{x}_{\ell,\mu} = V_\ell\boldsymbol{y}_{\ell,\mu}$. Since

$$\|A\boldsymbol{x}_{\ell,\mu} - \boldsymbol{b}\| = \|H_{\ell+1,\ell}^{(A)}\boldsymbol{y}_{\ell,\mu} - \boldsymbol{e}_1\|\boldsymbol{b}\|\,\|, \tag{2.18}$$

we can evaluate the norm of the residual error $A\boldsymbol{x}_{\ell,\mu} - \boldsymbol{b}$ by computing the norm of the residual error of the reduced problem on the right-hand side of (2.18). This is helpful when

ETNA
Kent State University
http://etna.math.kent.edu

determining a suitable value of the regularization parameter μ by the discrepancy principle or L-curve criterion; see, e.g., [9] for the latter. In the computed examples of Section 4, we assume that an estimate of the norm of the error $\boldsymbol{e}$ in $\boldsymbol{b}$ is known. Then μ can be determined by the discrepancy principle, i.e., we choose μ so that

$$\|H_{\ell+1,\ell}^{(A)}\boldsymbol{y}_{\ell,\mu} - \boldsymbol{e}_1\|\boldsymbol{b}\|\,\| = \|\boldsymbol{e}\|.$$

This value of μ is the solution of a nonlinear equation, which can be solved conveniently by using the GSVD of the matrix pair $\{H_{\ell+1,\ell}^{(A)}, H_{\ell,\ell}^{(B)}\}$. An efficient method for computing the GSVD is described in [4]; see also [3].

3. A decomposition method based on flexible Arnoldi reduction. Many commonly used regularization matrices $B \in \mathbb{R}^{p\times n}$ are rectangular with either $p < n$ or $p > n$. The reduction method for the matrix pairs $\{A, B\}$ described in [20], which is based on the flexible Arnoldi process due to Saad [21], requires the matrix B to be square. This section describes a simple modification of the method in [20] that allows B to be rectangular. Differently from the method in [20], the method of the present paper requires the evaluation of matrix-vector products with B^*. The matrix A is required to be square.

We first outline the flexible Arnoldi decomposition for a matrix pair $\{A, B\}$ in the Algorithmic Outline 3.1. A detailed algorithm is presented below.

ALGORITHM OUTLINE 3.1.
Input: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{p\times n}$, $\boldsymbol{b} \in \mathbb{R}^n$, ratio $\rho \geq 0$, and number of steps ℓ

Initialization:
$h_{1,1} := \|\boldsymbol{b}\|$; $\boldsymbol{u}_1 := \boldsymbol{b}/h_{1,1}$; $\boldsymbol{v}_1 := \boldsymbol{u}_1$;
Iteration:
for $j = 1, 2, \ldots, \ell$:

- Determine the new columns $\boldsymbol{u}_{j+1}$ and $\boldsymbol{w}_j$ of the matrices U and W, respectively, by equating the right-hand sides and left-hand sides of the expressions

$$AV_j = U_{j+1}H_{j+1,j},$$
$$BV_j = W_jR_{j,j},$$

  in a such a manner that the matrices U_{j+1} and W_j have orthonormal columns, $H_{j+1,j} \in \mathbb{R}^{(j+1)\times j}$ is upper Hessenberg, and $R_{j,j} \in \mathbb{R}^{j\times j}$ is upper triangular.
- Determine the new column $\boldsymbol{v}_{j+1}$ of the matrix V. This column should be linearly independent of the already available columns $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j$. In Algorithm 3.1 below, we will let $\boldsymbol{v}_{j+1}$ be a unit vector that is orthogonal to the columns $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_j$. It is constructed by matrix-vector product evaluations $A\boldsymbol{v}_j$ or $B^*B\boldsymbol{v}_j$ depending on the input parameter ρ.

end j-loop

The Algorithm Outline 3.1 generates the decompositions

$$AV_\ell = U_{\ell+1}H_{\ell+1,\ell}, \tag{3.1}$$
$$BV_\ell = W_\ell R_{\ell,\ell}, \tag{3.2}$$

which we apply to solve the Tikhonov minimization problem (1.4). Details of the outlined computations are described in Algorithm 3.1.

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHM 3.1 (Reduction of matrix pair $\{A, B\}$; A square, B rectangular).

1. Input: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{p\times n}$, $\boldsymbol{b} \in \mathbb{R}^n$, ratio $\rho \geq 0$, and number of steps ℓ
2. $h_{1,1} := \|\boldsymbol{b}\|$; $\boldsymbol{u}_1 := \boldsymbol{b}/h_{1,1}$; $\boldsymbol{v}_1 := \boldsymbol{u}_1$;
3. $N(\boldsymbol{u}) := 1$; $N(\boldsymbol{w}) := 1$
4. **for** $j = 1, 2, \ldots, \ell$ **do**
5. $\quad \widehat{\boldsymbol{u}} := A\boldsymbol{v}_j$ — new $\boldsymbol{u}$-vector
6. $\quad$ **for** $i = 1, 2, \ldots, j$ **do**
7. $\quad\quad h_{i,j} := \boldsymbol{u}_i^* \widehat{\boldsymbol{u}}$; $\widehat{\boldsymbol{u}} := \widehat{\boldsymbol{u}} - \boldsymbol{u}_i h_{i,j}$
8. $\quad$ **end for**
9. $\quad h_{j+1,j} := \|\widehat{\boldsymbol{u}}\|$
10. $\quad \boldsymbol{u}_{j+1} := \widehat{\boldsymbol{u}}/h_{j+1,j}$ — if $h_{j+1,j} = 0$: see text
11. $\quad \widehat{\boldsymbol{w}} := B\boldsymbol{v}_j$ — new $\boldsymbol{w}$-vector
12. $\quad$ **for** $i = 1, 2, \ldots, j-1$ **do**
13. $\quad\quad r_{i,j} := \boldsymbol{w}_i^* \widehat{\boldsymbol{w}}$; $\widehat{\boldsymbol{w}} := \widehat{\boldsymbol{w}} - \boldsymbol{w}_i r_{i,j}$
14. $\quad$ **end for**
15. $\quad r_{j,j} := \|\widehat{\boldsymbol{w}}\|$
16. $\quad \boldsymbol{w}_j := \widehat{\boldsymbol{w}}/r_{j,j}$ — if $r_{j,j} = 0$: see text
17. $\quad$ **if** $N(\boldsymbol{w})/N(\boldsymbol{u}) > 1/\rho$
18. $\quad\quad N(\boldsymbol{u}) := N(\boldsymbol{u}) + 1$; $\boldsymbol{v} := \boldsymbol{u}_{N(\boldsymbol{u})}$
19. $\quad$ **else**
20. $\quad\quad \boldsymbol{v} := B^* \boldsymbol{w}_{N(\boldsymbol{w})}$; $N(\boldsymbol{w}) := N(\boldsymbol{w}) + 1$
21. $\quad$ **end**
22. $\quad$ **for** $i = 1, 2, \ldots, j$ **do**
23. $\quad\quad \boldsymbol{v} := \boldsymbol{v} - (\boldsymbol{v}_i^* \boldsymbol{v})\boldsymbol{v}_i$;
24. $\quad$ **end for**
25. $\quad \alpha_j := \|\boldsymbol{v}\|$;
26. $\quad \boldsymbol{v}_{j+1} := \boldsymbol{v}/\alpha_j$; — new $\boldsymbol{v}$-vector
27. **end for**

The elements $h_{i,j}$ and $r_{i,j}$ in Algorithm 3.1 are the nontrivial entries of the matrices $H_{\ell+1\ell}$ and $R_{\ell,\ell}$ determined by the algorithm. Algorithm 3.1 differs from the flexible Arnoldi reduction algorithm presented in [20, Algorithm 2.1] only insofar as line 20 in this algorithm has been changed from $\boldsymbol{v} := \boldsymbol{w}_{N(\boldsymbol{w})}$ to $\boldsymbol{v} := B^* \boldsymbol{w}_{N(\boldsymbol{w})}$. Most of the properties of [20, Algorithm 2.1] carry over to Algorithm 3.1. The structure of the matrix R determined by Algorithm 3.1 is similar to the structure of the matrix $H^{(B)}$ computed by Algorithm 2.1 of the present paper. Moreover, if the matrix A in Algorithm 3.1 is symmetric, then the structure of the computed matrix H is similar to the structure of the matrix $H^{(A)}$ determined by Algorithm 2.1. The structure of H and R can be shown similarly to the analogous results for the matrices $H^{(A)}$ and $H^{(B)}$ in Section 2.

Assume for the moment the generic situation that no breakdown occurs during the execution of Algorithm 3.1. Then the algorithm yields the decompositions (3.1) and (3.2) and we obtain

$$\min_{\boldsymbol{x}\in\mathcal{R}(V_\ell)} \{\|A\boldsymbol{x} - \boldsymbol{b}\|^2 + \mu\|B\boldsymbol{x}\|^2\} = \min_{\boldsymbol{y}\in\mathbb{R}^\ell} \{\|H_{\ell+1,\ell}\boldsymbol{y} - \boldsymbol{e}_1\|\boldsymbol{b}\|\,\|^2 + \mu\|R_{\ell,\ell}\boldsymbol{y}\|^2\}. \tag{3.3}$$

It follows from (1.5) that the reduced minimization problem on the right-hand side has the unique solution $\boldsymbol{y}_{\ell,\mu}$ for any $\mu > 0$, from which we obtain the approximate solutions $\boldsymbol{x}_{\ell,\mu} = V_\ell \boldsymbol{y}_{\ell,\mu}$ of (1.4). Similarly to (2.18), we have

$$\|A\boldsymbol{x}_{\ell,\mu} - \boldsymbol{b}\| = \|H_{\ell+1,\ell}\boldsymbol{y}_{\ell,\mu} - \boldsymbol{e}_1\|\boldsymbol{b}\|\,\|,$$

ETNA
Kent State University
http://etna.math.kent.edu

and this allows us to determine μ with the aid of the discrepancy principle by solving a nonlinear equation that depends on the matrix pair $\{H_{\ell+1,\ell}, R_{\ell,\ell}\}$. We proceed analogously as outlined at the end of Section 2.

4. Numerical examples. We present a few examples that illustrate the application of decompositions computed by Algorithms 2.1 and 3.1 to Tikhonov regularization. We compare with results obtained by using decompositions determined the GSVD and by [20, Algorithm 2.1]. In all examples, the error vector $\boldsymbol{e}$ has normally distributed pseudorandom entries with mean zero; cf. (1.2). The vector is scaled to correspond to a chosen noise level

$$\delta = \frac{\|\boldsymbol{e}\|}{\|\hat{\boldsymbol{b}}\|}.$$

We assume the noise level to be known and therefore may apply the discrepancy principle to determine the regularization parameter $\mu > 0$; see the discussions at the end of Sections 2 and 3. The methods of this paper, of course, also can be applied in conjunction with other schemes for determining the regularization parameter such as the L-curve criterion.

We tabulate the relative error $\|\boldsymbol{x}_{\ell,\mu_\ell} - \widehat{\boldsymbol{x}}\|/\|\widehat{\boldsymbol{x}}\|$, where $\widehat{\boldsymbol{x}}$ is the desired solution of the unknown error-free linear system of equations (1.3). All computations are carried out in MATLAB with about 15 significant decimal digits.

EXAMPLE 4.1. Consider the inverse Laplace transform

$$\int_0^\infty \exp(-st) f(t)\,dt = \frac{1}{s+1/2}, \qquad 0 \le s < \infty,$$

with solution $f(t) = \exp(-t/2)$. This problem is discussed, e.g., by Varah [23]. We use the function `i_laplace` from [10] to determine a discretization $A \in \mathbb{R}^{n\times n}$ of the integral operator and a discretized scaled solution $\widehat{\boldsymbol{x}} \in \mathbb{R}^n$ for $n = 1000$. The error vector $\boldsymbol{e} \in \mathbb{R}^n$ has noise level $\delta = 10^{-1}$. The error-contaminated data vector $\boldsymbol{b}$ in (1.1) is defined by (1.2).

We will use the regularization matrix

$$B = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix},$$

where

$$L_1 = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & \mathrm{O} \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ \mathrm{O} & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n} \tag{4.1}$$

is a bidiagonal scaled finite difference approximation of the first-derivative operator and

$$L_2 = \frac{1}{4}\begin{bmatrix} -1 & 2 & -1 & & & \mathrm{O} \\ & 1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ \mathrm{O} & & & -1 & 2 & -1 \end{bmatrix} \in \mathbb{R}^{(n-2)\times n} \tag{4.2}$$

is a tridiagonal scaled finite difference approximation of the second-derivative operator. The matrix B damps finite differences that approximate both the first and second derivatives in the computed approximate solution.

ETNA
Kent State University
http://etna.math.kent.edu

We apply Algorithm 2.1 with $\rho = 1$, $\rho = 0.5$, and $\rho = 0.1$. The results are reported in Table 4.1. When $\rho = 1$, the best approximation of $\widehat{\boldsymbol{x}}$ is achieved after $\ell = 20$ iterations. If instead $\rho = 0.5$, then the best approximation of $\widehat{\boldsymbol{x}}$ is obtained after only $\ell = 13$ iterations, and $\rho = 0.1$ gives the best approximation after $\ell = 29$ iterations. This example illustrates that Algorithm 2.1 may yield approximate solutions of higher quality and require less storage and computational work when $\rho < 1$.

We do not presently have a complete understanding of how ρ should be chosen. In [20], we observed that when $\|B\boldsymbol{x}_{\ell,\mu_\ell}\|$ is small, the computed solution $\boldsymbol{x}_{\ell,\mu_\ell}$ often is of high quality and that choosing $\rho < 1$ seems to be beneficial for achieving this. It is also important that the condition number of the reduced matrix

$$\begin{bmatrix} H_{\ell+1,\ell}^{(A)} \\ H_{\ell,\ell}^{(B)} \end{bmatrix} \tag{4.3}$$

is not very large; a very large condition number could make the accurate numerical solution of the reduced Tikhonov minimization problem (2.17) problematic. We have found that for linear discrete ill-posed problems, the condition number of the matrix (4.3) generally, but not always, is reduced by decreasing ρ (for fixed ℓ). Table 4.2 displays condition numbers of the present example, and Figure 4.1 provides a graphical illustration.

We also solve the Tikhonov minimization problem of this example with the aid of the GSVD in the following manner. First we determine the QR factorization $B = QR$, where $Q \in \mathbb{R}^{(2n-3)\times n}$ has orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is upper triangular, and then we compute the GSVD of the matrix pair $\{A, R\}$. Table 4.1 shows this approach to yield the least accurate approximation of $\widehat{\boldsymbol{x}}$. Thus, it may be appropriate to use Algorithm 2.1 also for problems that are small enough to allow the application of the GSVD. Figure 4.2 shows the desired solution $\widehat{\boldsymbol{x}}$ (black dash-dotted curve) and the approximation $\boldsymbol{x}_{13,\mu_{13}}$ computed by Algorithm 2.1 with $\rho = 0.5$ (red solid curve). They are very close. The figure also displays the approximate solution determined by the GSVD (blue dashed curve).

TABLE 4.1
Example 4.1. Relative errors in computed approximate solutions for the noise level 10^{-1}.

Method	ρ	ℓ	$\|\boldsymbol{x}_{\ell,\mu_\ell} - \widehat{\boldsymbol{x}}\|/\|\widehat{\boldsymbol{x}}\|$
Algorithm 2.1	1	20	$3.71 \cdot 10^{-2}$
Algorithm 2.1	0.5	13	$3.16 \cdot 10^{-2}$
Algorithm 2.1	0.1	29	$3.29 \cdot 10^{-2}$
GSVD			$1.16 \cdot 10^{-1}$

EXAMPLE 4.2. The Fredholm integral equation of the first kind,

$$\int_0^{\pi/2} \kappa(\sigma,\tau)x(\sigma)d\sigma = b(\tau), \qquad 0 \le \tau \le \pi, \tag{4.4}$$

with $\kappa(\sigma,\tau) = \exp(\sigma\cos(\tau))$, $b(\tau) = 2\sinh(\tau)/\tau$, and solution $x(\tau) = \sin(\tau)$, is discussed by Baart [1]. We use the MATLAB function `baart` from [10] to discretize (4.4) by a Galerkin method with $n = 1000$ orthonormal box functions as test and trial functions. The function `baart` produces the nonsymmetric matrix $A \in \mathbb{R}^{n\times n}$ and the scaled discrete approximation $\widehat{\boldsymbol{x}} \in \mathbb{R}^n$ of $x(\tau)$, with which we compute the error-free right-hand side $\widehat{\boldsymbol{b}} := A\widehat{\boldsymbol{x}}$. The error vector $\boldsymbol{e} \in \mathbb{R}^n$ corresponds to the noise level $\delta = 1\cdot 10^{-2}$. The data vector $\boldsymbol{b}$ in (1.1) is obtained from (1.2).

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.2
Example 4.1. Condition number of the matrix (4.3) as a function of ℓ and ρ.

ℓ	ρ	condition number of the matrix (4.3)
13	1	$4.66 \cdot 10^1$
13	0.5	$3.77 \cdot 10^1$
13	0.1	$1.73 \cdot 10^1$
20	1	$2.18 \cdot 10^2$
20	0.5	$1.38 \cdot 10^2$
20	0.1	$3.59 \cdot 10^2$
29	1	$6.93 \cdot 10^2$
29	0.5	$5.81 \cdot 10^2$
29	0.1	$3.90 \cdot 10^2$

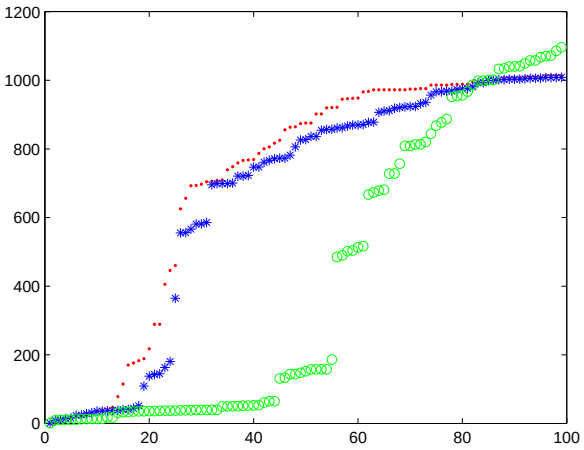

FIG. 4.1. *Example 4.1. Condition number of the matrices (4.3) as a function of ℓ and ρ. The vertical axis displays ℓ and the horizontal axis the condition number. The red dots show the condition number for $\rho = 1$, the blue stars the condition number for $\rho = 0.5$, and the green circles the condition number for $\rho = 0.1$.*

We seek to determine an approximation of $\widehat{\boldsymbol{x}}$ by using a decomposition determined by Algorithm 3.1. The regularization matrix L_2 is defined by (4.2). This approach is compared to [20, Algorithm 2.1]. The latter algorithm requires the regularization matrix to be square. We therefore pad the regularization matrix (4.2) with two zero rows when applied in Algorithm 2.1 of [20]. An approximate solution is also computed by using the GSVD of the matrix pair $\{A, L_2\}$.

The results are listed in Table 4.3. Both [20, Algorithm 2.1] and Algorithm 3.1 of the present paper with $\rho = 0.5$ yield better approximations of $\widehat{\boldsymbol{x}}$ than the GSVD; the best approximation of $\widehat{\boldsymbol{x}}$ can be seen to be determined by Algorithm 3.1 with $\rho = 0.5$; the relative error is $6.58 \cdot 10^{-3}$. This approximate solution is shown in Figure 4.3 (red solid curve). The figure also displays $\widehat{\boldsymbol{x}}$ (black dash-dotted curve) and the GSVD solution (blue solid curve).

Both Algorithm 3.1 of this paper and [20, Algorithm 2.1] yield approximations of $\widehat{\boldsymbol{x}}$ of higher quality when $\rho = 1/2$ than when $\rho = 1$. We therefore do not show results for $\rho = 1$. We also note that Algorithm 3.1 with $\rho = 0.25$ and $\rho = 0.20$ gives computed approximate solutions with relative error $8.7 \cdot 10^{-3}$ after $\ell = 58$ and $\ell = 79$ iterations, respectively. This relative error is smaller than the relative error of the GSVD solution. We finally remark that

ETNA
Kent State University
http://etna.math.kent.edu

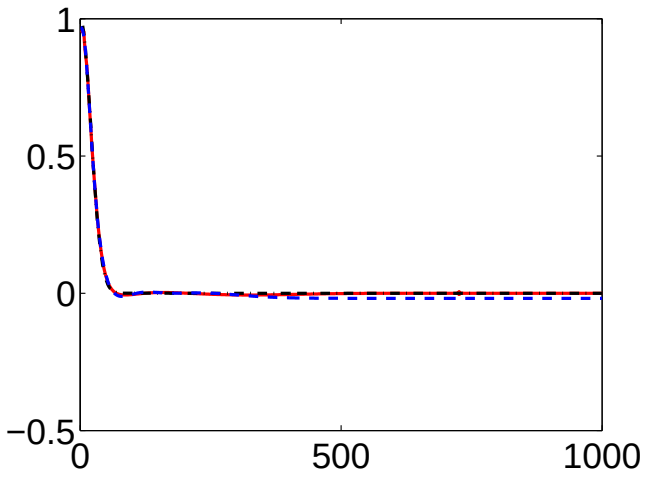

FIG. 4.2. *Example 4.1. The red solid curve displays the computed approximate solution $\boldsymbol{x}_{13,\mu_{13}}$ determined by Algorithm 2.1 with $\rho = 1/2$, the blue dashed curve shows the solution computed via GSVD, and the black dash-dotted curve depicts the desired solution $\widehat{\boldsymbol{x}}$.*

the relative error for $\rho = 0.1$ can be reduced to $1.46 \cdot 10^{-2}$ by carrying out more than 27 iterations. Hence, Algorithm 3.1 can determine approximate solutions with a smaller relative error than GSVD for many ρ-values smaller than or equal to 0.5.

Similarly as for Example 4.1, we display the condition number of the matrices

$$\begin{bmatrix} H_{\ell+1,\ell} \\ R_{\ell,\ell} \end{bmatrix} \tag{4.5}$$

as a function of ℓ and ρ. This matrix defines the reduced problem (3.3). Figure 4.4 shows the condition number for several ρ values and increasing ℓ. The condition number is seen to decrease as ρ increases.

TABLE 4.3
Example 4.2. Relative errors in computed approximate solutions for the noise level 10^{-3}. The parameter ℓ denotes the number of steps with Algorithm 3.1 of this paper and with [20, Algorithm 2.1].

Method	ρ	ℓ	$\|\boldsymbol{x}_{\ell,\mu_\ell} - \widehat{\boldsymbol{x}}\|/\|\widehat{\boldsymbol{x}}\|$
Algorithm 2.1 from [20]	0.5	16	$9.44 \cdot 10^{-3}$
Algorithm 3.1	0.5	26	$6.58 \cdot 10^{-3}$
Algorithm 3.1	0.1	27	$3.97 \cdot 10^{-2}$
GSVD			$2.76 \cdot 10^{-2}$

EXAMPLE 4.3. Our last example illustrates the performance of Algorithm 3.1 when applied to the restoration of a two-dimensional gray-scale image that has been contaminated by blur and noise. The gray-scale image `rice` from MATLAB's Image Processing Toolbox is represented by an array of 256×256 pixels. Each pixel is stored as an 8-bit unsigned integer with a value in the interval $[0, 255]$. The pixels are ordered row-wise and stored in a vector of dimension $n = 256^2$. Let $\widehat{\boldsymbol{x}} \in \mathbb{R}^n$ represent the blur- and noise-free image (which is assumed not to be available). We generate an associated blurred and noise-free image, $\widehat{\boldsymbol{b}}$, by multiplying $\widehat{\boldsymbol{x}}$ by a blurring matrix $A \in \mathbb{R}^{n\times n}$ that models Gaussian blur. This matrix is generated by the function `blur` from [10] with parameters `band` $= 9$ and `sigma` $= 2$.

ETNA
Kent State University
http://etna.math.kent.edu

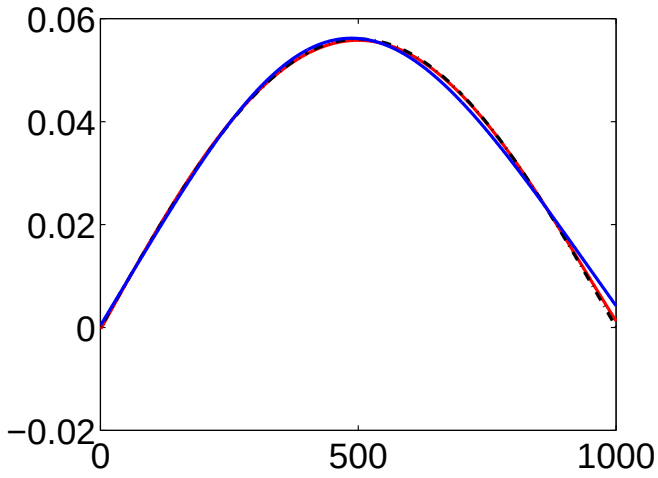

FIG. 4.3. *Example 4.2. Approximate solution $\boldsymbol{x}_{26,\mu_{26}}$ determined by Algorithm 3.1 of this paper with $\rho = 1/2$ with noise level 10^{-3} (red solid curve), approximate solution computed with GSVD (blue dashed curve), and desired solution $\widehat{\boldsymbol{x}}$ (black dash-dotted curve)*

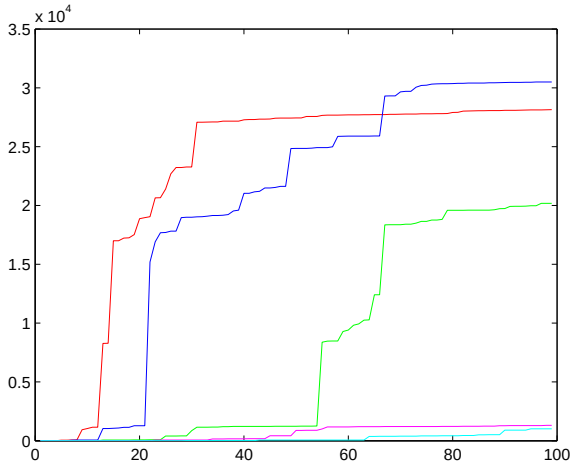

FIG. 4.4. *Example 4.2. Condition number of the matrices (4.5) as a function of ℓ and ρ. The vertical axis displays ℓ and the horizontal axis the condition number. The red graph shows the condition number for $\rho = 1$, the blue graph for $\rho = 1/3$, the green graph for $\rho = 1/5$, the magenta graph for $\rho = 1/10$, and the cyan graph for $\rho = 1/20$.*

The parameter `band` controls the bandwidth of the submatrices that comprise A and the parameter `sigma` controls the shape of the Gaussian point spread function. The blur- and noise-contaminated image $\boldsymbol{b} \in \mathbb{R}^n$ is obtained by adding a "noise vector" $\boldsymbol{e} \in \mathbb{R}^n$ to $\widehat{\boldsymbol{b}}$ with noise level $\delta = 10^{-2}$; cf. (1.2). Our task is to restore the image $\boldsymbol{b}$. The desired image $\widehat{\boldsymbol{x}}$ and the blur- and noise-contaminated image $\boldsymbol{b}$ are shown in Figures 4.5 and 4.6, respectively. We assume the blurring matrix A, the contaminated image $\boldsymbol{b} \in \mathbb{R}^n$, and the noise level δ to be available.

ETNA
Kent State University
http://etna.math.kent.edu

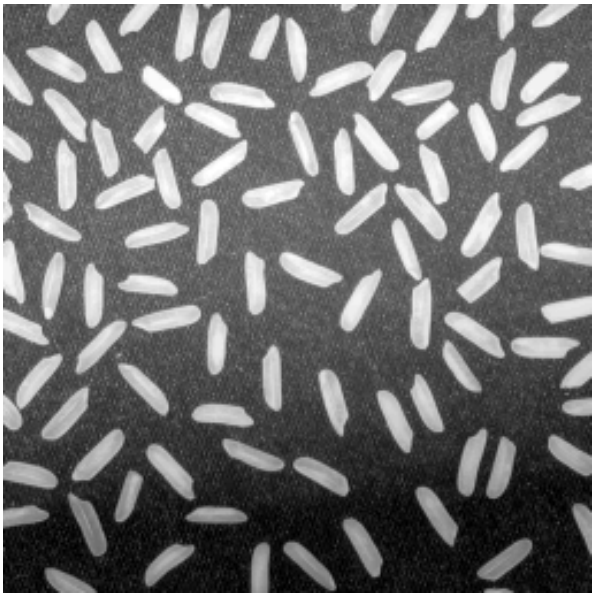

FIG. 4.5. *Example 4.3. Exact image.*

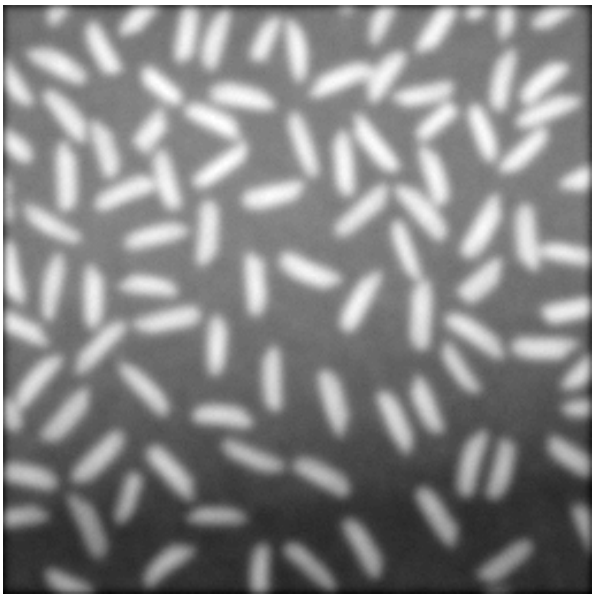

FIG. 4.6. *Example 4.3. Blur- and noise-contaminated image.*

ETNA
Kent State University
http://etna.math.kent.edu

The peak signal-to-noise ratio (PSNR) is commonly used to measure the quality of a restored image $\boldsymbol{x}$. It is defined as

$$\mathrm{PSNR}(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = 20 \log_{10} \left(\frac{255}{\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|} \right),$$

where the numerator 255 stems from the fact that each pixel is stored with 8 bits. A larger PSNR generally indicates that the restoration is of higher quality, but in some cases this may not agree with visual judgment. We therefore also display the restored image.

Let the regularization matrix B be defined by

$$B = \begin{bmatrix} I & \otimes & L_1 \\ L_1 & \otimes & I \end{bmatrix}, \tag{4.6}$$

where L_1 is given by (4.1) with $n = 256$. The matrix $B \in \mathbb{R}^{130560 \times 65536}$ has almost twice as many rows as columns. Therefore, we cannot use Algorithm 3.1 of [20]. This regularization matrix also is used in [13].

Table 4.4 reports results achieved with Algorithm 3.1 for several values of the parameter ρ. For each ρ, we carry out 30 iterations and select the approximation in the 30-dimensional solution subspace with the largest PSNR value. The restoration with the largest PSNR value is determined by Algorithm 3.1 with $\rho = 0.1$ and is displayed by Figure 4.7. We see that the best restoration is achieved with the smallest number of iterations.

TABLE 4.4
Example 4.3. PSNR-values of restorations computed by Algorithm 3.1 with B defined by (4.6).

Method	ρ	ℓ	PSNR
Algorithm 3.1	1	25	28.213
Algorithm 3.1	0.5	27	28.222
Algorithm 3.1	0.2	22	28.223
Algorithm 3.1	0.1	22	28.297

This example illustrates that Algorithm 3.1 with B given by (4.6) can yield quite accurate restorations with only 3 matrix-vector product evaluations with the matrix A. The development of a black box algorithm requires criteria for deciding on how many iterations ℓ to carry out and how to choose ρ. The discussion in [20] on the choice of ℓ carries over to Algorithm 3.1.

5. Conclusion and extension. We have described extensions of the generalized Golub–Kahan reduction method for matrix pairs described in [12] and of the reduction method based on the generalized Arnoldi process introduced in [20]. Computed examples illustrate the benefits of both these extensions. In particular, Examples 4.1–4.2 show that letting $0 < \rho < 1$ in Algorithm 2.1 may give a more accurate approximation of $\widehat{\boldsymbol{x}}$ than $\rho = 1$. The reduction methods of this paper can be generalized to matrix q-tuplets with $q \geq 3$ in a similar fashion as the methods in [12, 20].

Acknowledgement. The authors would like to thank Stefan Kindermann for carefully reading the manuscript.

ETNA
Kent State University
http://etna.math.kent.edu

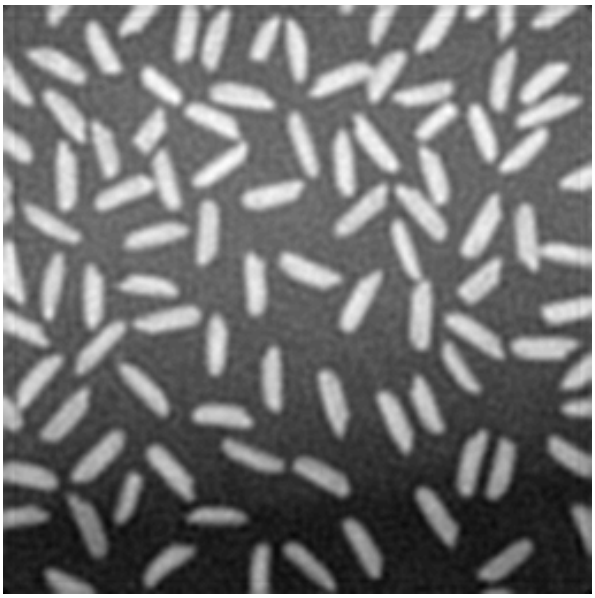

FIG. 4.7. *Example 4.3. Restored image* $\boldsymbol{x}_{22,\mu_{22}}$ *obtained by Algorithm 3.1 with* $\rho = 0.1$ *and regularization matrix* (4.6).

REFERENCES

[1] M. L. BAART, *The use of auto-correlation for pseudo-rank determination in noisy ill-conditioned least-squares problems*, IMA J. Numer. Anal., 2 (1982), pp. 241–247.
[2] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[3] L. DYKES, S. NOSCHESE, AND L. REICHEL, *Rescaling the GSVD with application to ill-posed problems*, Numer. Algorithms, 68 (2015), pp. 531–545.
[4] L. DYKES AND L. REICHEL, *Simplified GSVD computations for the solution of linear discrete ill-posed problems*, J. Comput. Appl. Math., 255 (2013), pp. 15–17.
[5] L. ELDÉN, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–501.
[6] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.
[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
[8] C. W. GROETSCH, *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, 1984.
[9] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.
[10] ———, *Regularization tools version 4.0 for Matlab 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.
[11] M. E. HOCHSTENBACH AND L. REICHEL, *An iterative method for Tikhonov regularization with a general linear regularization operator*, J. Integral Equations Appl., 22 (2010), pp. 463–480.
[12] M. E. HOCHSTENBACH, L. REICHEL, AND X. YU, *A Golub–Kahan-type reduction method for matrix pairs*, J. Sci. Comput., in press.
[13] M. E. KILMER, P. C. HANSEN, AND M. I. ESPAÑOL, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330.
[14] S. KINDERMANN, *Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems*, Electron. Trans. Numer. Anal., 38 (2011), pp. 233–257. http://etna.mcs.kent.edu/vol.38.2011/pp233-257.dir
[15] ———, *Discretization independent convergence rates for noise level-free parameter choice rules for the regularization of ill-conditioned problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 58–81. http://etna.mcs.kent.edu/vol.40.2013/pp58-81.dir

ETNA
Kent State University
http://etna.math.kent.edu

[16] R.-C. LI AND Q. YE, *A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 405–428.
[17] L. REICHEL AND G. RODRIGUEZ, *Old and new parameter choice rules for discrete ill-posed problems*, Numer. Algorithms, 63 (2013), pp. 65–87.
[18] L. REICHEL, F. SGALLARI, AND Q. YE, *Tikhonov regularization based on generalized Krylov subspace methods*, Appl. Numer. Math., 62 (2012), pp. 1215–1228.
[19] L. REICHEL AND Q. YE, *A generalized LSQR algorithm*, Numer. Linear Algebra Appl., 15 (2008), pp. 643–660.
[20] L. REICHEL AND X. YU, *Tikhonov regularization via flexible Arnoldi reduction*, BIT, in press.
[21] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[22] M. A. SAUNDERS, H. D. SIMON, AND E. L. YIP, *Two conjugate-gradient-type methods for unsymmetric linear equations*, SIAM J. Numer. Anal, 25 (1988), pp. 927–940.
[23] J. M. VARAH, *Pitfalls in the numerical solution of linear ill-posed problems*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 164–176.
[24] H. ZHA, *Computing the generalized singular values/vectors of large sparse or structured matrix pairs*, Numer. Math., 72 (1996), pp. 391–417.