

Solution methods for linear discrete ill-posed problems for color image restoration

A. H. Bentbib¹ · M. El Guide^{1,2} · K. Jbilou³ ·
E. Onunwor^{4,5} · L. Reichel⁴

Received: 8 June 2017 / Accepted: 26 March 2018 / Published online: 12 April 2018
© Springer Science+Business Media B.V., part of Springer Nature, corrected publication May 2018

Abstract This work discusses four algorithms for the solution of linear discrete ill-posed problems with several right-hand side vectors. These algorithms can be applied, for instance, to multi-channel image restoration when the image degradation model is described by a linear system of equations with multiple right-hand sides that are contaminated by errors. Two of the algorithms are block generalizations of the stan-

Communicated by Lars Eldén.

The original version of this article was revised.

✉ L. Reichel
reichel@math.kent.edu

A. H. Bentbib
a.bentbib@uca.ac.ma

M. El Guide
mohamed.elguide@um6p.ma; mohamed.elguide@edu.uca.ac.ma

K. Jbilou
jbilou@univ-littoral.fr

E. Onunwor
eonunwor@starkstate.edu

¹ Laboratoire de Mathématiques Appliquées et Informatique, Faculté des Sciences et Techniques-Gueliz, Marrakesh, Morocco

² Université Mohammed VI Polytechnique, FAB-LAB, UM6P, Bengueurir, Morocco

³ Université du Littoral Côte d'Opale, L.M.P.A., ULCO, 50 rue F. Buisson, BP699, 62228 Calais-Cedex, France

⁴ Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

⁵ Department of Mathematics, Stark State College, 6200 Frank Ave. NW, North Canton, OH 44720, USA

dard Golub–Kahan bidiagonalization method with the block size equal to the number of channels. One algorithm uses standard Golub–Kahan bidiagonalization without restarts for all right-hand sides. These schemes are compared to standard Golub–Kahan bidiagonalization applied to each right-hand side independently. Tikhonov regularization is used to avoid severe error propagation. Numerical examples illustrate the performance of these algorithms. Applications include the restoration of color images.

Keywords Golub–Kahan bidiagonalization · Block Golub–Kahan bidiagonalization · Global Golub–Kahan bidiagonalization · Tikhonov regularization · Ill-posed problem · Multiple right-hand sides · Color image restoration

Mathematics Subject Classification 6510 · 65F22

1 Introduction

This paper discusses the use of iterative methods based on standard or block Golub–Kahan-type bidiagonalization, combined with Tikhonov regularization, to the restoration of a multi-channel image from an available blur- and noise-contaminated version. Applications include the restoration of color images whose RGB (red, green, and blue) representation uses three channels; see [9, 17]. The methods described also can be applied to the solution of Fredholm integral equations of the first kind in two or more space dimensions and to the restoration of hyper-spectral images. The latter kind of images generalize color images in that they allow more than three “colors”; see, e.g. [21]. For definiteness, we focus in this section on the restoration of k -channel images that have been contaminated by blur and noise, and formulate this restoration task as a linear system of equations with k right-hand side vectors, where each spectral band corresponds to one channel. To simplify our notation, we assume the image to be represented by an array of $n \times n$ pixels in each one of the k channels, where $1 \leq k \ll n^2$. Let $b^{(i)} \in \mathbb{R}^{n^2}$ represent the available blur- and noise-contaminated image in channel i , let $e^{(i)} \in \mathbb{R}^{n^2}$ describe the noise in this channel, and let $\hat{x}^{(i)} \in \mathbb{R}^{n^2}$ denote the desired unknown blur- and noise-free image in channel i . The corresponding quantities for all k channels $b, \hat{x}, e \in \mathbb{R}^{n^2 k}$ are obtained by stacking the vectors $b^{(i)}, \hat{x}^{(i)}, e^{(i)}$ of each channel. For instance, $b = [(b^{(1)})^T, \dots, (b^{(k)})^T]^T$. Here and throughout this paper, the superscript T denotes transposition.

The degradation model is of the form

$$b = H\hat{x} + e \quad (1.1)$$

with blurring matrix

$$H = A_k \otimes A \in \mathbb{R}^{n^2 k \times n^2 k}.$$

Here \otimes denotes the Kronecker product, the matrix $A \in \mathbb{R}^{n^2 \times n^2}$ represents within-channel blurring, which is assumed to be the same in all channels, and the small matrix $A_k \in \mathbb{R}^{k \times k}$ models cross-channel blurring.

Sometimes it is convenient to gather images for the different channels in “block vectors.” Introduce the block vectors $B = [b^{(1)}, \dots, b^{(k)}] \in \mathbb{R}^{n^2 \times k}$, $\widehat{X} = [\widehat{x}^{(1)}, \dots, \widehat{x}^{(k)}] \in \mathbb{R}^{n^2 \times k}$, and $E = [e^{(1)}, \dots, e^{(k)}] \in \mathbb{R}^{n^2 \times k}$. Using properties of the Kronecker product, the model (1.1) can be expressed as

$$B = \mathcal{A}(\widehat{X}) + E, \quad (1.2)$$

where the linear operator \mathcal{A} is defined by

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{n^2 \times k} &\rightarrow \mathbb{R}^{n^2 \times k} \\ \mathcal{A}(X) &:= AXA_k^T. \end{aligned} \quad (1.3)$$

Its transpose is given by $\mathcal{A}^T(X) := A^T X A_k$. The model (1.2) is said to have cross-channel blurring when $A_k \neq I_k$; when $A_k = I_k$, there is no cross-channel blurring. In the latter situation, the blurring is said to be within-channel only, and the deblurring problem decouples into k independent deblurring problems. The degradation model (1.1) then can be expressed in the form

$$B = A\widehat{X} + E. \quad (1.4)$$

For notational simplicity, we denote in the following both the matrix A in (1.4) and the linear operator \mathcal{A} in (1.2) by A , and we write $\mathcal{A}(X)$ as AX .

The singular values of a blurring matrix or operator A typically “cluster” at the origin, i.e., A has many singular values of different orders of magnitude close to zero. It follows that the solution (if it exists) of the linear system of equations

$$AX = B \quad (1.5)$$

is very sensitive to the error E in B . Linear systems of equations with a matrix of this kind are commonly referred to as linear discrete ill-posed problems.

Let \widehat{B} denote the (unknown) noise-free block vector associated with B . The system of equations $AX = \widehat{B}$ is assumed to be consistent, and \widehat{X} stands for the solution of minimal Frobenius norm of this system. The Frobenius norm of a matrix M is defined by $\|M\|_F = \text{trace}(M^T M)^{1/2}$.

We would like to determine an accurate approximation of \widehat{X} given B and A . This generally is a difficult computational task due to the error E in B and the presence of tiny positive singular values of A . Tikhonov regularization reduces the sensitivity of the solution of (1.5) to the error E in B by replacing (1.5) by a penalized least-squares problem of the form

$$\min_{X \in \mathbb{R}^{n^2 \times k}} \{\|AX - B\|_F^2 + \mu^{-1}\|X\|_F^2\}, \quad (1.6)$$

where $\mu > 0$ is the regularization parameter. The normal equations associated with this minimization problem are given by

$$(A^T A + \mu^{-1} I)X = A^T B. \quad (1.7)$$

They have the unique solution

$$X_\mu = \left(A^T A + \mu^{-1} I \right)^{-1} A^T B \quad (1.8)$$

for any $\mu > 0$. The size of μ determines how sensitive X_μ is to the error in B and how close X_μ is to the desired solution \hat{X} . We will comment on the use of μ^{-1} in (1.6) instead of μ below.

The computation of an accurate approximation X_μ of \hat{X} requires that a suitable value of the regularization parameter μ be used. Several methods for determining such a μ -value have been suggested in the literature. These include so-called heuristic methods that do not require knowledge of the size of the error E in B , such as the L-curve criterion, generalized cross validation, and the quasi-optimality criterion; see, e.g. [2, 8, 11, 15, 19, 20, 24] for discussions and illustrations. We will use the discrepancy principle, discussed, e.g. in [6], to determine μ in the computed examples reported in Sect. 5. The discrepancy principle requires that a bound $\varepsilon > 0$ of $\|E\|_F$ be available and prescribes that $\mu > 0$ be chosen so that the solution (1.8) of (1.6) satisfies

$$\|B - AX_\mu\|_F = \eta\varepsilon, \quad (1.9)$$

where $\eta > 1$ is a user-specified constant independent of ε . A zero-finder can be applied to determine a μ -value such that the associated Tikhonov solution (1.8) satisfies (1.9). When the matrix A is of small to moderate size, the left-hand side of (1.9) easily can be evaluated by using the singular value decomposition (SVD) of A . However, computation of the SVD is impractical when the matrix A is large.

We will discuss how an approximate solution of (1.6) can be computed by first evaluating a partial block Golub–Kahan bidiagonalization (BGKB) of A and then solving (1.6) in a subspace so defined.

Alternatively, we may reduce A to a small bidiagonal matrix with the aid of global Golub–Kahan bidiagonalization (GGKB), which also is a block method, and then apply the connection between GGKB and Gauss-type quadrature rules to determine upper and lower bounds for the left-hand side of (1.9). This allows the computation of a suitable value of μ in a simple manner. This approach has previously been applied in [1]; the GGKB method was first described in [26]. The BGKB and GGKB block methods are compared to the application of Golub–Kahan bidiagonalization (with block size one) in two ways. One approach applies Golub–Kahan bidiagonalization with initial vector $b^{(1)}$ and generates a solution subspace that is large enough to solve all systems of equations

$$Ax^{(i)} = b^{(i)}, \quad i = 1, \dots, k, \quad (1.10)$$

with Tikhonov regularization. The other approach is to simply solve each one of the k systems of equations (1.10) independently with Golub–Kahan bidiagonalization and Tikhonov regularization, i.e., by using the algorithm described in [3] k times.

This paper is organized as follows. Section 2 describes the BGKB method and discusses its application to the solution of (1.6). The determination of a regularization

parameter such that the computed solution satisfies the discrepancy principle is also described. Section 3 reviews the use of the GGKB method to reduce A . The connection between this reduction and Gauss-type quadrature rules is exploited to compute bounds for the left-hand side of (1.9). The solution of (1.6) by applying Golub–Kahan bidiagonalization (with block size one) determined by A and the initial vector $b^{(1)}$ is discussed in Sect. 4. Sufficiently many bidiagonalization steps are carried out so that all systems (1.10) can be solved with solution subspaces determined by A and $b^{(1)}$. We also consider the solution of the k systems (1.10) independently by Golub–Kahan bidiagonalization and Tikhonov regularization as described in [3]. Section 5 presents a few numerical examples. Concluding remarks can be found in Sect. 6.

2 Solution by partial block Golub–Kahan bidiagonalization

Introduce for $\mu > 0$ the function

$$\phi(\mu) = \|B - AX_\mu\|_F^2. \quad (2.1)$$

Substituting (1.8) into (2.1) and using the identity

$$I - A(A^T A + \mu^{-1}I)^{-1}A^T = (\mu AA^T + I)^{-1} \quad (2.2)$$

shows that (2.1) can be written as

$$\phi(\mu) = \text{tr} \left(B^T f_\mu(AA^T)B \right) \quad (2.3)$$

with

$$f_\mu(t) = (\mu t + 1)^{-2}.$$

The determination of a value of the regular parameter $\mu > 0$ such that (1.9) holds generally requires the function ϕ to be evaluated for several μ -values. Each evaluation of ϕ is very expensive for large-scale problems. We therefore approximate the expression $B^T f_\mu(AA^T)B$ by a simpler one, which we determine with a few steps of block Golub–Kahan bidiagonalization as follows. Introduce the QR factorization $B = P_1 R_1$, where $P_1 \in \mathbb{R}^{n^2 \times k}$ has orthonormal columns and $R_1 \in \mathbb{R}^{k \times k}$ is upper triangular. Then ℓ steps of the BGKB method applied to A with initial block vector P_1 gives the decompositions

$$AQ_\ell^{(k)} = P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)}, \quad A^T P_\ell^{(k)} = Q_\ell^{(k)} C_\ell^{(k)T}, \quad (2.4)$$

where the matrices $P_\ell^{(k)} = [P_1, \dots, P_\ell] \in \mathbb{R}^{n^2 \times \ell k}$, $P_{\ell+1}^{(k)} = [P_1, \dots, P_{\ell+1}] \in \mathbb{R}^{n^2 \times (\ell+1)k}$, and $Q_\ell^{(k)} = [Q_1, \dots, Q_\ell] \in \mathbb{R}^{n^2 \times \ell k}$ have orthonormal columns, and

$$\bar{C}_\ell^{(k)} := \begin{bmatrix} L_1 & & & & \\ R_2 & L_2 & & & \\ & \ddots & \ddots & & \\ & & R_\ell & L_\ell & \\ & & & R_{\ell+1} & \end{bmatrix} \in \mathbb{R}^{k(\ell+1) \times k\ell}$$

is lower block bidiagonal with lower triangular diagonal blocks $L_j \in \mathbb{R}^{k \times k}$ and upper triangular blocks $R_j \in \mathbb{R}^{k \times k}$. Moreover, $C_\ell^{(k)}$ is the leading $k\ell \times k\ell$ submatrix of $\bar{C}_\ell^{(k)}$. In case A denotes the operator \mathcal{A} defined by (1.3), the expressions $AQ_\ell^{(k)}$ and $A^T P_\ell^{(k)}$ in the left-hand sides of (2.4) should be replaced by $[\mathcal{A}(Q_1), \dots, \mathcal{A}(Q_\ell)]$ and $[\mathcal{A}^T(P_1), \dots, \mathcal{A}^T(P_\ell)]$, respectively. When the block size is $k = 1$, the decompositions (2.4) simplify to the decompositions computed by the algorithm `bidiag1` by Paige and Saunders [23]. In particular, the decompositions (2.4) differ from the ones described by Golub et al. [13], who compute an upper block bidiagonal matrix. In our discussion, we will assume that ℓ is small enough so that the triangular matrices L_j , $j = 1, \dots, \ell$, and R_j , $j = 2, \dots, \ell + 1$, are nonsingular.

It follows from (2.4) that the range of the matrix $P_\ell^{(k)}$ is the block Krylov subspace

$$\mathbb{K}_\ell(AA^T, B) = \text{range}[P_1, AA^T P_1, (AA^T)^2 P_1, \dots, (AA^T)^{\ell-1} P_1].$$

Similarly, the range of the matrix $Q_\ell^{(k)}$ is the block Krylov subspace

$$\mathbb{K}_\ell(A^T A, A^T B) = \text{range}[A^T P_1, A^T AA^T P_1, (A^T A)^2 A^T P_1, \dots, (A^T A)^{\ell-1} A^T P_1].$$

Multiplying the rightmost equation in (2.4) by A from the left yields

$$AA^T P_\ell^{(k)} = P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)} C_\ell^{(k)T}.$$

Therefore,

$$P_\ell^{(k)T} AA^T P_\ell^{(k)} = C_\ell^{(k)} C_\ell^{(k)T}.$$

This suggests that $f_\mu(AA^T)$ may be approximated by evaluating $f_\mu(C_\ell^{(k)} C_\ell^{(k)T})$, which is much easier to compute than $f_\mu(AA^T)$ when A is large. Let E_1 denotes the block vector of appropriate dimensions with blocks of size $k \times k$, with the first block equal to I_k and all other blocks equal to 0. It follows from results by Golub and Meurant [14] on the symmetric block Lanczos algorithm that the expression

$$\mathcal{G}_\ell f_\mu = R_1^T E_1^T f_\mu \left(C_\ell^{(k)} C_\ell^{(k)T} \right) E_1 R_1 \quad (2.5)$$

can be interpreted as an ℓ -block Gauss quadrature rule for the approximation of $B^T f_\mu(AA^T)B$, i.e.,

$$\mathcal{G}_\ell f = B^T f(AA^T)B \quad \forall f \in \mathbb{P}_{2\ell-1},$$

where $\mathbb{P}_{2\ell-1}$ denotes the set of all polynomials of degree at most $2\ell - 1$; see also [7] for related discussions. We therefore approximate (2.3) by

$$\phi_\ell(\mu) = \text{tr}(\mathcal{G}_\ell f_\mu) \quad (2.6)$$

and let the regularization parameter be the solution of

$$\phi_\ell(\mu) = \eta^2 \varepsilon^2. \quad (2.7)$$

The following result shows that $\phi_\ell(\mu)$ is decreasing and convex. This makes it convenient to compute the solution μ_ℓ of (2.7) by Newton's method; see below.

Proposition 2.1 *The functions $\phi(\mu)$ and $\phi_\ell(\mu)$, defined by (2.3) and (2.6) for $\mu > 0$, respectively, satisfy*

$$\phi'(\mu) < 0, \quad \phi''(\mu) > 0, \quad \phi'_\ell(\mu) < 0, \quad \phi''_\ell(\mu) > 0.$$

Proof The derivative of $\phi(\mu)$ is given by

$$\phi'(\mu) = -2 \text{tr}(B^T (\mu A A^T + I)^{-3} A A^T B).$$

It follows from $(\mu A A^T + I)^{-1} A = A(\mu A^T A + I)^{-1}$ that

$$\phi'(\mu) = -2 \text{tr}(B^T A (\mu A^T A + I)^{-3} A^T B).$$

Substituting the spectral factorization $A^T A = S \Lambda S^T$, $S^T S = I$, into the above expression and letting $W = [w_1, \dots, w_k] = S^T A^T B$ yields

$$\phi'(\mu) = -2 \text{tr}(W^T (\mu \Lambda + I)^{-3} W) = -2 \sum_{j=1}^k w_j^T (\mu \Lambda + I)^{-3} w_j < 0.$$

Thus, $\phi(\mu)$ is a decreasing function of μ . Turning to the second derivative, we have

$$\phi''(\mu) = 6 \text{tr}(B^T A A^T (\mu A A^T + I)^{-4} A A^T B),$$

and can proceed similarly as above to show that $\phi''(\mu) > 0$.

The derivative of $\phi_\ell(\mu)$ is given by

$$\phi'_\ell(\mu) = \text{tr}(R_1^T E_1^T C_\ell^{(k)} (\mu C_\ell^{(k)T} C_\ell^{(k)} + I)^{-3} C_\ell^{(k)T} E_1 R_1), \quad (2.8)$$

where we again use the identity $(\mu C_\ell^{(k)} C_\ell^{(k)T} + I)^{-1} C_\ell^{(k)} = C_\ell^{(k)} (\mu C_\ell^{(k)T} C_\ell^{(k)} + I)^{-1}$. The stated properties of $\phi'_\ell(\mu)$ and $\phi''_\ell(\mu)$ can be shown by substituting the spectral factorization of $C_\ell^{(k)T} C_\ell^{(k)}$ into (2.8). \square

Since $\phi_\ell(\mu)$ is decreasing and convex, Newton's method converges monotonically and quadratically to the solution μ_ℓ of (2.7) for any initial approximate solution $\mu_{\text{init}} < \mu_\ell$. This makes it easy to implement the Newton method. For instance, we may use $\mu_{\text{init}} = 0$ when ϕ_ℓ and its derivative are suitably defined at $\mu = 0$; see [3] for a detailed discussion of the case when the block size is one.

We note that the function $\mu \rightarrow \phi_\ell(1/\mu)$, which corresponds to the regularization term $\mu \|X\|_F^2$ in (1.6), is not guaranteed to be convex. Therefore, Newton's method has to be safeguarded when applied to the solution of $\phi_\ell(1/\mu) = \varepsilon^2$. This is the reason for considering Tikhonov regularization of the form (1.6).

Proposition 2.2 *Let $P_{\mathcal{N}(M)}$ denote the orthogonal projector onto the null space $\mathcal{N}(M)$ of the matrix M . Then*

$$\begin{aligned}\phi(0) &= \text{tr}(B^T B), & \lim_{\mu \rightarrow \infty} \phi(\mu) &= \text{tr}(B^T P_{\mathcal{N}(AA^T)} B), \\ \phi_\ell(0) &= \text{tr}(B^T B), & \lim_{\mu \rightarrow \infty} \phi_\ell(\mu) &= \text{tr}(R_1^T E_1^T P_{\mathcal{N}(R_\ell R_\ell^T)} E_1 R_1).\end{aligned}$$

Proof The value at zero and limit of ϕ follow from (2.3). The expression (2.5) and the definition of the upper triangular matrix R_1 in the QR factorization $B = P_1 R_1$ yield

$$\phi_\ell(0) = \text{tr}\left(R_1^T E_1^T f_0\left(C_\ell^{(k)} C_\ell^{(k)T}\right) E_1 R_1\right) = \text{tr}(R_1^T R_1) = \text{tr}(B^T B).$$

The result for $\phi_\ell(\mu)$ as $\mu \rightarrow \infty$ follows similarly as for ϕ . \square

Let the regularization parameter μ_ℓ be computed by Newton's method. We then determine the corresponding approximate solution by projecting the normal equations (1.7) with $\mu = \mu_\ell$ onto a smaller space determined by the decompositions (2.4). We seek to determine an approximate solution of the form

$$X_{\mu_\ell} = Q_\ell^{(k)} Y_{\mu_\ell}, \quad Y_{\mu_\ell} \in \mathbb{R}^{k\ell \times k\ell}, \quad (2.9)$$

by solving the normal equations (1.7) with $\mu = \mu_\ell$ by a Galerkin method,

$$(Q_\ell^{(k)})^T (A^T A + \mu_\ell^{-1} I) Q_\ell^{(k)} Y_{\mu_\ell} = (Q_\ell^{(k)})^T A^T B, \quad (2.10)$$

which simplifies to

$$(\bar{C}_\ell^{(k)T} \bar{C}_\ell^{(k)} + \mu_\ell^{-1} I) Y_{\mu_\ell} = \bar{C}_\ell^{(k)T} E_1 R_1. \quad (2.11)$$

We compute the solution $Y_{\ell, \mu}$ by solving a least-squares problem for which (2.11) are the normal equations

$$\min_{Y \in \mathbb{R}^{k\ell \times k\ell}} \left\| \begin{bmatrix} \bar{C}_\ell^{(k)} \\ \mu_\ell^{-1/2} I \end{bmatrix} Y - \begin{bmatrix} E_1 R_1 \\ 0 \end{bmatrix} \right\|_F^2. \quad (2.12)$$

Our reason for computing the solution of (2.12) instead of (2.11) is that solving the least-squares problem is less sensitive to errors for small values of $\mu_\ell > 0$.

Proposition 2.3 *Let μ_ℓ solve (2.7), and let Y_{μ_ℓ} solve (2.10). Then the associated approximate solution $X_{\mu_\ell} = Q_\ell^{(k)} Y_{\mu_\ell}$ of (1.6) satisfies*

$$\|AX_{\mu_\ell} - B\|_F^2 = \text{tr} \left(R_1^T E_1^T f_{\mu_\ell} (\bar{C}_\ell^{(k)} \bar{C}_\ell^{(k)T}) E_1 R_1 \right). \quad (2.13)$$

Proof Using the expression of $X_{\ell,\mu}$ and applying (2.4) shows that

$$\begin{aligned} AX_{\mu_\ell} - B &= A Q_\ell^{(k)} Y_{\mu_\ell} - B \\ &= P_{\ell+1}^{(k)} \bar{C}_\ell^{(k)} Y_{\mu_\ell} - P_1 R_1 \\ &= P_{\ell+1}^{(k)} \left(\bar{C}_\ell^{(k)} Y_{\mu_\ell} - E_1 R_1 \right), \end{aligned}$$

where we recall that $B = P_1 R_1$. It follows from (2.11) that

$$P_{\ell+1}^{(k)} \left(\bar{C}_\ell^{(k)} Y_{\mu_\ell} - E_1 R_1 \right) = P_{\ell+1}^{(k)} \left[\left(\bar{C}_\ell^{(k)} \left(\bar{C}_\ell^{(k)T} \bar{C}_\ell^{(k)} + \mu_\ell^{-1} I \right)^{-1} \bar{C}_\ell^{(k)T} - I \right) E_1 R_1 \right].$$

The identity (2.2) with A replaced by $\bar{C}_\ell^{(k)}$ now yields (2.13). \square

Algorithm 1 The BGKB-Tikhonov method.

Input: $A, B, k, \varepsilon, \eta \geq 1$.

1. Compute the QR factorization $B = P_1 R_1$.

2. **For** $\ell = 1, 2, \dots$ until $\|AX_{\mu_\ell} - B\|_F \leq \eta\varepsilon$

 (a) Determine $Q_\ell^{(k)}$ and $P_{\ell+1}^{(k)}$ and block bidiagonal matrix $C_\ell^{(k)}$ by BGKB.

 (b) Update the value μ_ℓ by solving (2.7) with Newton's method.

3. Determine Y_{μ_ℓ} by solving (2.12) and then X_{μ_ℓ} by (2.9).

3 The GGKB method and Gauss-type quadrature

We discuss the application of the GGKB method to the computation of an approximate solution of (1.6) and review how the method can be used to compute inexpensive upper and lower bounds for the discrepancy (1.9). These bounds help us to determine the regularization parameter. This approach of solving (1.6) and determining bounds for the discrepancy has recently been described in [1], where further details can be found.

Introduce the inner product

$$\langle F, G \rangle = \text{tr}(F^T G), \quad F, G \in \mathbb{R}^{n^2 \times k}.$$

We have $\|F\|_F = \langle F, F \rangle^{1/2}$. Application of ℓ steps of the GGKB method to A with initial block vector B determines the lower bidiagonal matrix

$$\bar{C}_\ell = \begin{bmatrix} \rho_1 & & & & & \\ \sigma_2 & \rho_2 & & & & \\ & \ddots & \ddots & & & \\ & & \sigma_{\ell-1} & \rho_{\ell-1} & & \\ & & & \sigma_\ell & \rho_\ell & \\ & & & & \sigma_{\ell+1} & \end{bmatrix} \in \mathbb{R}^{(\ell+1) \times \ell}$$

as well as the matrices

$$U_{\ell+1}^{(k)} = [U_1, U_2, \dots, U_{\ell+1}] \in \mathbb{R}^{n^2 \times (\ell+1)k}, \quad V_\ell^{(k)} = [V_1, V_2, \dots, V_\ell] \in \mathbb{R}^{n^2 \times \ell k}$$

with orthonormal block columns $U_i, V_j \in \mathbb{R}^{n^2 \times k}$, where $U_1 = s_1 B$ and $s_1 > 0$ is a scaling factor. Thus,

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

We assume that ℓ is small enough so that all nontrivial entries of the matrix \bar{C}_ℓ are positive. This is the generic situation. Denote the leading $\ell \times \ell$ submatrix of \bar{C}_ℓ by C_ℓ . The matrices determined satisfy

$$A[V_1, V_2, \dots, V_\ell] = U_{\ell+1}^{(k)}(\bar{C}_\ell \otimes I_k), \quad (3.1)$$

$$A^T[U_1, U_2, \dots, U_\ell] = V_\ell^{(k)}(C_\ell^T \otimes I_k). \quad (3.2)$$

When A is the operator \mathcal{A} defined by (1.3), one should replace $A[V_1, V_2, \dots, V_\ell]$ and $A^T[U_1, U_2, \dots, U_\ell]$ on the left-hand sides of (3.1) and (3.2) by the expressions $[\mathcal{A}(V_1), \mathcal{A}(V_2), \dots, \mathcal{A}(V_\ell)]$ and $[\mathcal{A}^T(U_1), \mathcal{A}^T(U_2), \dots, \mathcal{A}^T(U_\ell)]$, respectively.

The functions (of μ)

$$\begin{aligned} \mathcal{G}_\ell f_\mu &= \|B\|_F^2 e_1^T (\mu C_\ell C_\ell^T + I_\ell)^{-2} e_1, \\ \mathcal{R}_{\ell+1} f_\mu &= \|B\|_F^2 e_1^T (\mu \bar{C}_\ell \bar{C}_\ell^T + I_{\ell+1})^{-2} e_1 \end{aligned}$$

can be interpreted as Gauss-type quadrature rules for the approximation of the expression $\phi(\mu)$ defined by (2.3). The remainder formulas for these quadrature rules yield the bounds

$$\mathcal{G}_\ell f_\mu \leq \phi(\mu) \leq \mathcal{R}_{\ell+1} f_\mu; \quad (3.3)$$

see [1] for details.

We determine a suitable value of μ and an associated approximate solution of (1.6) as follows. For $\ell \geq 2$, we seek to solve the nonlinear equation

$$\mathcal{G}_\ell f_\mu = \varepsilon^2 \quad (3.4)$$

for $\mu > 0$ by Newton's method. If the solution μ_ℓ of (3.4) satisfies

$$\mathcal{R}_{\ell+1} f_{\mu_\ell} \leq \eta^2 \varepsilon^2, \quad (3.5)$$

then it follows from (3.3) that there is a solution X_{μ_ℓ} of (1.6) such that

$$\varepsilon \leq \|B - AX_{\mu_\ell}\|_F \leq \eta \varepsilon.$$

If either (3.4) does not have a solution or (3.5) does not hold, then we increase ℓ . Generally, it suffices to choose ℓ quite small.

Assume that (3.4) and (3.5) hold for $\mu = \mu_\ell$. We then compute the approximate solution

$$X_{\mu_\ell, \ell} = V_\ell^{(k)}(y_{\mu_\ell} \otimes I_k) \quad (3.6)$$

of (1.6), where y_{μ_ℓ} solves

$$(\bar{C}_\ell^T \bar{C}_\ell + \mu_\ell^{-1} I_\ell)y = d_1 \bar{C}_\ell^T e_1, \quad d_1 = \|B\|_F. \quad (3.7)$$

The vector y_{μ_ℓ} is computed by solving a least-squares problem for which (3.7) are the associated normal equations. The following result shows an important property of the approximate solution (3.6). We include a proof for completeness.

Proposition 3.1 *Let μ_ℓ solve (3.4) and let y_{μ_ℓ} solve (3.7). Then the associated approximate solution (3.6) of (1.6) satisfies*

$$\|AX_{\mu_\ell, \ell} - B\|_F^2 = \mathcal{R}_{\ell+1} f_{\mu_\ell}.$$

Proof The representation (3.6) and (3.1) show that

$$AX_{\mu_\ell, \ell} = U_{\ell+1}^{(k)}(\bar{C}_\ell \otimes I_k)(y_{\mu_\ell} \otimes I_k) = U_{\ell+1}^{(k)}(\bar{C}_\ell y_{\mu_\ell} \otimes I_k).$$

Using the above expression gives

$$\begin{aligned} \|AX_{\mu_\ell, \ell} - B\|_F^2 &= \|U_{\ell+1}^{(k)}(d_1 e_1 \otimes I_k) - U_{\ell+1}^{(k)}(\bar{C}_\ell y_{\mu_\ell} \otimes I_k)\|_F^2 \\ &= \|(d_1 e_1 \otimes I_k) - (\bar{C}_\ell y_{\mu_\ell} \otimes I_k)\|_F^2 \\ &= \|d_1 e_1 - \bar{C}_\ell y_{\mu_\ell}\|_F^2, \end{aligned}$$

where we recall that $d_1 = \|B\|_F$. We now express y_{μ_ℓ} with the aid of (3.7), and apply the identity (2.2) with A replaced by \bar{C}_ℓ , to obtain

$$\begin{aligned}\|AX_{\mu_\ell, \ell} - B\|_F^2 &= d_1^2 \|e_1 - \bar{C}_\ell(\bar{C}_\ell^T \bar{C}_\ell + \mu_\ell^{-1} I_\ell)^{-1} \bar{C}_\ell^T e_1\|_F^2 \\ &= d_1^2 e_1^T (\mu_\ell \bar{C}_\ell \bar{C}_\ell^T + I_{\ell+1})^{-2} e_1 \\ &= \mathcal{R}_{\ell+1} f_{\mu_\ell}.\end{aligned}$$

□

The following algorithm outlines the main steps for computing μ_ℓ and $X_{\mu_\ell, \ell}$ that satisfy (1.9).

Algorithm 2 The GGKB-Tikhonov method.

Input: $A, B, k, \varepsilon, \eta \geq 1$.

1. Let $U_1 := B/\|B\|_F$.

2. **For** $\ell = 1, 2, \dots$ until $\|AX_{\mu_\ell} - B\|_F \leq \eta\varepsilon$

(a) Determine $U_{\ell+1}^{(k)}$ and $V_\ell^{(k)}$, and the bidiagonal matrices C_ℓ and \bar{C}_ℓ with GGKB algorithm.

(b) Determine μ_ℓ that satisfies (3.4) with Newton's method.

3. Determine y_{μ_ℓ} by solving a least-squares problem for which (3.7) are the associated normal equations and then compute $X_{\mu_\ell, \ell}$ by (3.6).

4 Golub–Kahan bidiagonalization for problems with multiple right-hand sides

We may consider (1.5) as k linear discrete ill-posed problems that have the same matrix A and different right-hand side vectors $b^{(1)}, \dots, b^{(k)}$; cf. (1.10). The solution of linear systems of equations with multiple right-hand sides that might not be known simultaneously and a matrix that stems from the discretization of a well-posed problem has received considerable attention in the literature; see e.g., [4, 5, 18, 22, 25] and references therein. However, the solution of linear discrete ill-posed problems with multiple right-hand sides that might not be available simultaneously has not. The method described in this section is based on the analysis and numerical experience reported in [12], where it is shown that it often suffices to apply only a few steps of (standard) Golub–Kahan bidiagonalization (GKB) to a matrix A of a linear discrete ill-posed problem to gain valuable information of subspaces spanned by the right and left singular vectors of A associated with the dominant singular values.

Consider the first system of (1.10),

$$Ax^{(1)} = b^{(1)}, \quad (4.1)$$

where the right-hand side is the sum of an unknown error-free vector $\widehat{b}^{(1)}$ and an error-vector $e^{(1)}$. Thus, $b^{(1)} = \widehat{b}^{(1)} + e^{(1)}$. A bound $\|e^{(1)}\| \leq \varepsilon^{(1)}$ is assumed to be known. Let $\widehat{x}^{(1)}$ denote the first column of the matrix \widehat{X} in (1.4). We seek to compute

an approximation of $\hat{x}^{(1)}$ by using (standard) partial Golub–Kahan bidiagonalization (GKB) of A with initial vector $b^{(1)}$.

To explain some properties of the bidiagonalization computed, we introduce the SVD of A ,

$$A = W \Sigma Z^T, \quad (4.2)$$

where $W, Z \in \mathbb{R}^{n^2 \times n^2}$ are orthogonal matrices and

$$\begin{aligned} \Sigma &= \text{diag} [\sigma_1, \sigma_2, \dots, \sigma_{n^2}] \in \mathbb{R}^{n^2 \times n^2}, \\ \sigma_1 &\geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_{n^2} = 0. \end{aligned}$$

Here r is the rank of A . Let $1 \leq s \leq r$ and let Z_s and W_s consist of the first s columns of Z and W , respectively. Moreover, Σ_s denotes the leading $s \times s$ principal submatrix of Σ . This gives the best rank- s approximation

$$A_s = W_s \Sigma_s Z_s^T$$

of A in the spectral and Frobenius norms.

The computation of the full SVD (4.2) is too expensive for large-scale problems without a particular structure to be practical. The computation of a partial GKB is much cheaper. Application of ℓ steps of GKB yields the decompositions

$$A V_\ell = U_{\ell+1} \tilde{C}_\ell, \quad A^T U_\ell = V_\ell C_\ell^T, \quad (4.3)$$

where the matrices $V_\ell = [v_1, v_2, \dots, v_\ell] \in \mathbb{R}^{n^2 \times \ell}$ and $U_{\ell+1} = [u_1, u_2, \dots, u_{\ell+1}] \in \mathbb{R}^{n^2 \times (\ell+1)}$ have orthonormal columns, and U_ℓ consists of the first ℓ columns of $U_{\ell+1}$. Further, $\tilde{C}_\ell \in \mathbb{R}^{(\ell+1) \times \ell}$ is lower bidiagonal and C_ℓ is the leading $\ell \times \ell$ submatrix of \tilde{C}_ℓ . We apply reorthogonalization of the columns of $U_{\ell+1}$ and V_ℓ to secure their numerical orthogonality. It is shown in [12] that for sufficiently many steps ℓ , the spaces $\text{range}(U_{\ell+1})$ and $\text{range}(V_\ell)$ contain to high accuracy the subspaces $\text{range}(W_s)$ and $\text{range}(Z_s)$, respectively, for $s \geq 1$ fixed and not too large. Computed examples in [12] indicate that it often suffices to choose $\ell \leq 3s$. Moreover, the columns of the noise-free right-hand side matrix \hat{B} , generally, can be approximated quite well by only the first few columns of the matrix W in the SVD (4.2) of A . This follows from the discrete Picard condition [15]. These columns, in turn, typically can be approximated fairly accurately by the first few columns of the matrix $U_{\ell+1}$ in the partial Golub–Kahan bidiagonalization (4.3). It is therefore unlikely that many steps of this bidiagonalization process have to be carried out in order to be able to compute useful approximations of the columns of the desired solution matrix \hat{X} .

Consider the Tikhonov regularization problem

$$\min_{x \in \text{range}(V_\ell)} \{\|Ax - b^{(1)}\|_2^2 + \mu \|x\|_2^2\} = \min_{y \in \mathbb{R}^\ell} \{\|\tilde{C}_\ell y - U_{\ell+1}^T b^{(1)}\|_2^2 + \mu \|y\|_2^2\}, \quad (4.4)$$

where $x = V_\ell y$. We determine the regularization parameter $\mu > 0$ so that the computed solution y_μ satisfies the discrepancy principle

$$\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(1)}\|_2 = \eta \varepsilon^{(1)}. \quad (4.5)$$

If no such μ -value exists, then we increase ℓ by one and try to solve (4.5) with ℓ replaced by $\ell + 1$ in (4.4) and (4.5). The small least-squares problem on the right-hand side of (4.4) is solved as described in [3]. We remark that the vector $U_{\ell+1}^T b^{(1)}$ can be simplified to $e_1 \|b^{(1)}\|_2$. The solution y_μ of (4.4) determines the approximate solution $x_\mu^{(1)} = V_\ell y_\mu$ of (4.1).

We turn to the problem

$$Ax^{(2)} = b^{(2)} \quad (4.6)$$

and compute an approximate solution by solving (4.4) with the vector $b^{(1)}$ replaced by $b^{(2)}$. The vector $U_{\ell+1}^T b^{(2)}$ has to be explicitly computed. Therefore it is important that the columns of the matrix $U_{\ell+1}$ are numerically orthonormal. If no $\mu > 0$ can be determined so that (4.5) can be satisfied with $b^{(1)}$ replaced by $b^{(2)}$, then we carry out one more step of Golub–Kahan bidiagonalization (4.3); otherwise, we compute the solution y_μ of (4.4) with the available decomposition.

Let μ be such that the discrepancy principle holds. Then we obtain the approximate solution $x_\mu^{(2)} = V_\ell y_\mu$ of (4.6). We proceed in the same manner to solve $Ax^{(i)} = b^{(i)}$ for $i = 3, 4, \dots, k$. We will compare this algorithm and Algorithms 1 and 2

Algorithm 3 The GKB-Tikhonov method.

Input: $A, k, b^{(1)}, b^{(2)}, \dots, b^{(k)}, \varepsilon^{(1)}, \varepsilon^{(2)}, \dots, \varepsilon^{(k)}, \eta \geq 1$.

1. Let $u_1 := b^{(1)} / \|b^{(1)}\|_2$.

2. Compute $AV_\ell = U_{\ell+1} \bar{C}_\ell$, $A^T U_\ell = V_\ell C_\ell^T$

3. **For** $i = 1, 2, \dots, k$

(a) Compute $\min_{y_\mu \in \mathbb{R}^\ell} \{\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2^2 + \mu \|y_\mu\|_2^2\}$

(b) If $\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2 > \eta \varepsilon^{(i)}$

i. $\ell := \ell + 1$

ii. Return to step (a).

(c) Compute $x_\mu^{(i)} = V_\ell y_\mu$

to the following “trivial” method that is based on solving each one of the linear discrete ill-posed problems (1.10) independently with the aid of (standard) Golub–Kahan bidiagonalization. Thus, we apply Algorithm 2 with block size one to each one of the k linear discrete ill-posed problems (1.10) independently. We refer to this scheme as Algorithm 4. We expect it to require the most matrix-vector product evaluations of the methods in our comparison because we compute a new partial standard Golub–Kahan bidiagonalization for each one of the vectors $b^{(j)}$, $j = 1, \dots, k$. Moreover, this method does not benefit from the fact that on many modern computers the evaluation of matrix-block-vector products with a large matrix A does not require much more

time than the evaluation of a matrix-vector product with a single vector for small block sizes; see e.g., [10] for discussions on this and related issues.

Algorithm 4 “Trivial” method.

Input: $A, k, b^{(1)}, b^{(2)}, \dots, b^{(k)}, \varepsilon^{(1)}, \varepsilon^{(2)}, \dots, \varepsilon^{(k)}, \eta \geq 1$.

1. For $i = 1, 2, \dots, k$

(a) Let $u_1 := b^{(i)} / \|b^{(i)}\|_2$.

(b) Compute Golub–Kahan bidiagonalization $AV_\ell = U_{\ell+1}\bar{C}_\ell, \quad A^T U_\ell = V_\ell C_\ell^T$

(c) Compute $\min_{y_\mu \in \mathbb{R}^\ell} \{\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2^2 + \mu \|y_\mu\|_2^2\}$

(d) If $\|\bar{C}_\ell y_\mu - U_{\ell+1}^T b^{(i)}\|_2 > \eta \varepsilon^{(i)}$

i. $\ell := \ell + 1$

ii. Return to step (b).

(e) Compute $x_\mu^{(i)} := V_\ell y_\mu$

5 Numerical results

This section provides some numerical results that illustrate the performance of Algorithms 1–4 when applied to the solution of linear discrete ill-posed problems with the same matrix and different right-hand sides. The first example applies these algorithms to the solution of a linear discrete ill-posed problem with several right-hand sides defined by matrices that stem from Regularization Tools by Hansen [16]; the remaining examples discuss applications to image restoration. We consider the restoration of RGB images that have been contaminated by within-channel and cross-channel blur and noise, as well as the restoration of a sequence of images from a video. All computations were carried out using the MATLAB environment on an Intel(R) Core(TM) i5-4590 CPU computer with 8 GB of RAM. The computations were done with approximately 15 decimal digits of relative accuracy.

5.1 Example 1

We would like to solve linear discrete ill-posed problems (1.10) with the matrix $A \in \mathbb{R}^{70^2 \times 70^2}$ determined by the function `phillips` in Regularization Tools [16]. The matrix is a discretization of a Fredholm integral equation of the first kind that describes a convolution on the interval $-6 \leq t \leq 6$. The function `phillips` also determines the error-free data vector $\hat{b}^{(1)} \in \mathbb{R}^{70^2}$ and the associated error-free solution $\hat{x}^{(1)} \in \mathbb{R}^{70^2}$. The other error-free data vectors $\hat{b}^{(i)} \in \mathbb{R}^{70^2}$, $i = 2, \dots, k$, are obtained by setting $\hat{x}^{(i)} = \hat{x}^{(i-1)} + y/2$ for $i = 2, \dots, k$, where y is a vector obtained by discretizing a function of the form $\frac{1}{2} \cos(\frac{t}{3}) + \frac{1}{4}$ at equidistant points on the interval $-6 \leq t \leq 6$. The error-free right-hand sides are obtained from $\hat{b}^{(i)} = A\hat{x}^{(i)}$ for $i = 2, \dots, k$. A noise vector $e^{(i)} \in \mathbb{R}^{70^2}$ with normally distributed random entries with zero mean is added to each data vector $\hat{b}^{(i)}$ to obtain the error-contaminated data vectors $b^{(i)}$, $i = 1, \dots, k$, in (1.10). The error-vectors $e^{(i)}$ are scaled to correspond to a specified noise level. This is simulated by

Table 1 Results for the phillips test problem

Noise level	Method	MVP	Relative error	CPU-time (s)
10^{-3}	Algorithm 1	100	1.46×10^{-2}	0.30
	Algorithm 2	200	1.31×10^{-2}	0.43
	Algorithm 3	16	2.28×10^{-2}	0.31
	Algorithm 4	162	1.43×10^{-2}	2.08
10^{-2}	Algorithm 1	80	2.54×10^{-2}	0.24
	Algorithm 2	120	2.61×10^{-2}	0.30
	Algorithm 3	10	2.52×10^{-2}	0.19
	Algorithm 4	140	2.60×10^{-2}	1.32

$$e^{(i)} := \tilde{\delta} \|\widehat{b}^{(i)}\|_2 \tilde{e}^{(i)},$$

where $\tilde{\delta}$ is the noise level and the vectors $\tilde{e}^{(i)} \in \mathbb{R}^{70^2}$ have normally distributed random entries with mean zero and variance one.

When the data vectors $b^{(i)}$, $i = 1, \dots, k$, are available sequentially, the linear discrete ill-posed problems (1.10) can be solved one by one by Algorithms 3 or 4. If the data vectors are available simultaneously, then Algorithms 1 and 2 also can be used to solve (1.10). The latter algorithms require that the noise level for each discrete ill-posed problem (1.10) is about the same. This is a reasonable assumption for many applications.

Table 1 compares the number of matrix-vector product evaluations and the CPU time required by Algorithms 1–4 for $k = 10$ and noise-contaminated data vectors $b^{(i)}$ corresponding to the noise levels $\tilde{\delta} = 10^{-2}$ and $\tilde{\delta} = 10^{-3}$. In all examples of this section, the regularization parameter is determined with the aid of the discrepancy principle with $\eta = 1.1$; cf. (1.9). The displayed relative error in the computed solutions is the maximum error for each one of the k problems (1.10). The number of matrix-vector products (MVP) shown is the number of matrix-vector product evaluations with A and A^T with a single vector. Thus, each iteration step of Algorithms 1 and 2 adds $2k$ matrix-vector product evaluations to the count. The number of matrix-vector product evaluations does not give an accurate idea of the computing time required. We therefore also present timings for the algorithms.

Table 1 shows Algorithm 3 to require the fewest matrix-vector product evaluations and to give approximate solutions of comparable or higher quality than the other algorithms. Algorithms 2 and 4 require about the same number of matrix-vector product evaluations, but the former algorithm demands less CPU time because it implements a block method.

5.2 Example 2

This example illustrates the performance of Algorithms 1–4 when applied to the restoration of a 3-channel RGB color image that has been contaminated by blur and

noise. The corrupted image is stored in a block vector B with three columns (one for each channel). The desired (and assumed unavailable) image is stored in the block vector \widehat{X} with three columns. The blur-contaminated, but noise-free image associated with \widehat{X} , is stored in the block vector \widehat{B} . The block vector E represents the noise in B , i.e., $B := \widehat{B} + E$. We define the noise level

$$\nu = \frac{\|E\|_F}{\|\widehat{B}\|_F}.$$

To determine the effectiveness of our solution methods, we evaluate the relative error

$$\text{Relative error} = \frac{\|\widehat{X} - X_{\mu_\ell}\|_F}{\|\widehat{X}\|_F},$$

where X_{μ_ℓ} denotes the computed restoration.

We consider within-channel blurring. Hence, the matrix A_3 in (1.3) is the 3×3 identity matrix, and the matrix A in (1.3), which describes the blurring within each channel, models Gaussian blur and is determined by the Gaussian PSF,

$$h_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

Thus, A is a symmetric block Toeplitz matrix with Toeplitz blocks. It is generated with the MATLAB function `blur` from [16]. This function has two parameters, the half-bandwidth of the Toeplitz blocks, r , and the variance, σ , of the Gaussian PSF. We let $\sigma = 4$ and $r = 6$.

The original (unknown) RGB image $\widehat{X} \in \mathbb{R}^{256 \times 256 \times 3}$ is the `papav256` image from MATLAB. It is shown on the left-hand side of Fig. 1. The associated blurred and noisy image $B = A\widehat{X} + E$ is shown on the right-hand side of the figure. The noise level is $\nu = 10^{-3}$. Given the contaminated image B , we would like to recover



Fig. 1 Example 2: Original image (left), blurred and noisy image (right)

Table 2 Results for Example 2

Noise level	Method	MVP	Relative error	CPU-time (s)
10^{-3}	Algorithm 1	492	6.93×10^{-2}	3.86
	Algorithm 2	558	6.85×10^{-2}	3.95
	Algorithm 3	112	2.64×10^{-1}	1.66
	Algorithm 4	632	1.29×10^{-1}	6.55
10^{-2}	Algorithm 1	144	9.50×10^{-2}	1.13
	Algorithm 2	156	9.44×10^{-2}	1.12
	Algorithm 3	20	2.91×10^{-1}	0.32
	Algorithm 4	112	1.58×10^{-1}	1.10

**Fig. 2** Example 2: Restored image by Algorithm 1 (left), and restored image by Algorithm 2 (right)

an approximation of the original image \hat{X} . Table 2 compares the number of matrix-vector product evaluations, the computing time, and the relative errors in the computed restorations.

The restoration obtained with Algorithm 1 for the noise level $\nu = 10^{-3}$ is shown on the left-hand side of Fig. 2. The discrepancy principle is satisfied when $\ell = 82$ steps of the BGKB method have been carried out. This corresponds to $3 \times 2 \times 82$ matrix-vector product evaluations.

The restoration determined by Algorithm 2 is shown on the right-hand side of Fig. 2. The GGKB method requires $\ell = 93$ steps to satisfy the discrepancy principle. Algorithm 3 is the fastest, but yields restorations of lower quality than the other algorithms for this example.

5.3 Example 3

The previous example illustrates the restoration of an image that has been contaminated by noise and within-channel blur, but not by cross-channel blur. This example shows



Fig. 3 Example 3: Cross-channel blurred and noisy image (left), restored image by Algorithm 2 (right)

the restoration of an image that has been contaminated by noise, within-channel blur, and cross-channel blur. We use the same within-channel blur as in Example 2. The cross-channel blur is defined by the cross-channel blur matrix

$$A_3 = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.25 & 0.5 & 0.25 \\ 0.15 & 0.1 & 0.75 \end{bmatrix}$$

from [17]. The blurred and noisy image is represented by $B = A\widehat{X}A_3^T + E$, where the noise level is $\nu = 10^{-3}$. It is shown on the left-hand side of Fig. 3. We restore this image with Algorithms 1–4. The quality of the restored images obtained with Algorithm 1, 2, and 4 is about the same, while the restoration determined by Algorithm 3 is of poor quality. The best attainable restoration is shown on the right-hand side of Fig. 3. Table 3 compares the algorithms for this example.

5.4 Example 4

We evaluate the effectiveness of Algorithms 1–4 when applied to the restoration of a video defined by a sequence of black and white images. Video restoration is the

Table 3 Results for Example 3

Noise level	Method	MVP	Relative error	CPU-time (s)
10^{-3}	Algorithm 1	354	7.56×10^{-2}	2.74
	Algorithm 2	702	6.97×10^{-2}	4.99
	Algorithm 3	112	2.64×10^{-1}	1.63
	Algorithm 4	556	1.35×10^{-1}	5.77

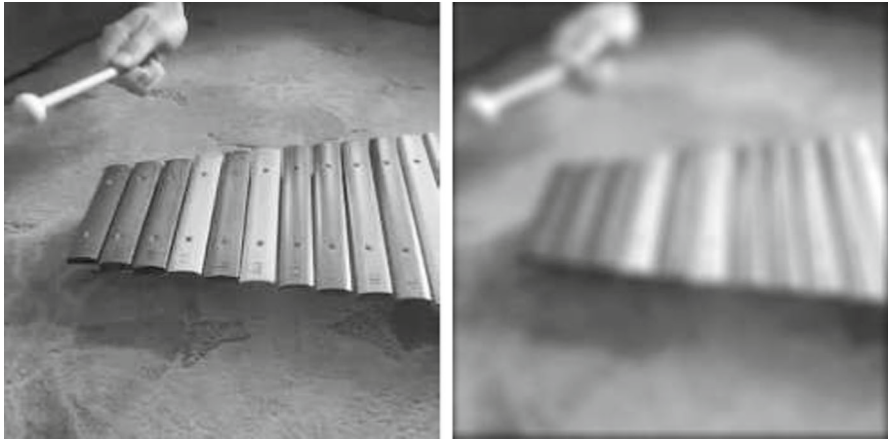


Fig. 4 Frame no. 3: Original frame (left), blurred and noisy frame (right)

Table 4 Results for Example 4

Noise level	Method	MVP	Relative error	CPU-time (s)
10^{-3}	Algorithm 1	1152	5.76×10^{-2}	8.72
	Algorithm 2	1188	5.66×10^{-2}	6.23
	Algorithm 3	130	1.19×10^{-1}	1.69
	Algorithm 4	1190	5.67×10^{-2}	10.79
10^{-2}	Algorithm 1	264	9.48×10^{-2}	1.65
	Algorithm 2	228	9.53×10^{-2}	1.21
	Algorithm 3	34	1.40×10^{-1}	0.44
	Algorithm 4	250	9.48×10^{-2}	2.22

problem of restoring a sequence of k images (frames). Each frame is represented by a matrix of $n \times n$ pixels. In the present example, we are interested in restoring 6 consecutive frames of a contaminated video. We consider the xylophone video from MATLAB. The video clip is in MP4 format with each frame having 240×240 pixels. The (unknown) blur- and noise-free frames are stored in the block vector $\widehat{X} \in \mathbb{R}^{240^2 \times 6}$. These frames are blurred by a blurring matrix A of the same kind and with the same parameters as in Example 2. Figure 4 shows the exact (original) frame and the contaminated version, which is to be restored. Blurred and noisy frames are generated by $B = A\widehat{X} + E$, where the matrix E represents white Gaussian noise of levels $\nu = 10^{-3}$ or $\nu = 10^{-2}$. Table 4 displays the performance of the algorithms. Algorithms 1 and 2 are seen to yields fairly accurate restorations, with the latter algorithm requiring the least CPU time. Figure 5 shows restorations of frame 3 obtained with two of the algorithms for noise level $\nu = 10^{-3}$.

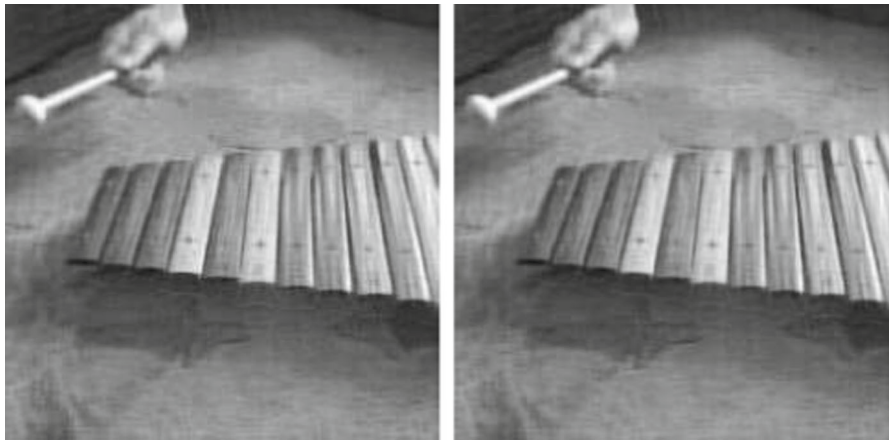


Fig. 5 Frame no. 3: Restored frame by Algorithm 1 (left), and restored frame by Algorithm 2 (right)

6 Conclusion

This paper discusses four approaches to the solution of linear discrete ill-posed problems with multiple right-hand sides. Algorithm 4 is clearly the least attractive of the algorithms considered. Algorithm 3 is the fastest for all examples, but determines approximate solutions of worse quality than Algorithms 1 and 2 for all image restoration examples. The accuracy achieved by Algorithm 3 depends on how well-suited the Krylov subspace generated by the matrix A and the first right-hand side $b^{(1)}$ is to represent the desired solutions associated with the other right-hand sides $b^{(2)}, \dots, b^{(k)}$. Algorithms 1 and 2 are attractive compromises between high accuracy and speed. Their relative speed depends on the computer architecture.

Acknowledgements The authors would like to thank Lars Eldén and the referee for carefully reading the manuscript and for comments that improved the presentation. The authors also would like to thank Alessandro Buccini for comments. Research by L.R. is supported in part by NSF Grants DMS-1729509 and DMS-1720259.

References

1. Bentbib, A.H., El Guide, M., Jbilou, K., Reichel, L.: Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems. *J. Comput. Appl. Math.* **322**, 46–56 (2017)
2. Calvetti, D., Hansen, P.C., Reichel, L.: L-curve curvature bounds via Lanczos bidiagonalization. *Electron. Trans. Numer. Anal.* **14**, 134–149 (2002)
3. Calvetti, D., Reichel, L.: Tikhonov regularization of large linear problems. *BIT* **43**, 263–283 (2003)
4. Chan, T.F., Wan, W.L.: Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM J. Stat. Comput.* **18**, 1698–1721 (1997)
5. El Guennouni, A., Jbilou, K., Sadok, H.: A block version of BiCGSTAB for linear systems with multiple right-hand sides. *Electron. Trans. Numer. Anal.* **16**, 129–142 (2003)
6. Engl, H.W., Hanke, M., Neubauer, A.: *Regularization of Inverse Problems*. Kluwer, Dordrecht (1996)
7. Fenu, C., Martin, D., Reichel, L., Rodriguez, G.: Block Gauss and anti-Gauss quadrature with application to networks. *SIAM J. Matrix Anal. Appl.* **34**, 1655–1684 (2013)
8. Fenu, C., Reichel, L., Rodriguez, G.: GCV for Tikhonov regularization via global Golub–Kahan decomposition. *Numer. Linear Algebra Appl.* **23**, 467–484 (2016)

9. Galatsanos, N.P., Katsaggelos, A.K., Chin, R.T., Hillary, A.D.: Least squares restoration of multichannel images. *IEEE Trans. Signal Proc.* **39**, 2222–2236 (1991)
10. Gallivan, K., Heath, M., Ng, E., Peyton, B., Plemmons, R., Ortega, J., Romine, C., Sameh, A., Voigt, R.: *Parallel Algorithms for Matrix Computations*. SIAM, Philadelphia (1990)
11. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. *Electron. Trans. Numer. Anal.* **44**, 83–123 (2015)
12. Gazzola, S., Onunwor, E., Reichel, L., Rodriguez, G.: On the Lanczos and Golub–Kahan reduction methods applied to discrete ill-posed problems. *Numer. Linear Algebra Appl.* **23**, 187–204 (2016)
13. Golub, G.H., Luk, F.T., Overton, M.L.: A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Trans. Math. Softw.* **7**, 149–169 (1981)
14. Golub, G.H., Meurant, G.: *Matrices, Moments and Quadrature with Applications*. Princeton University Press, Princeton (2010)
15. Hansen, P.C.: *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia (1998)
16. Hansen, P.C.: Regularization tools version 4.0 for MATLAB 7.3. *Numer. Algorithms* **46**, 189–194 (2007)
17. Hansen, P.C., Nagy, J., O’Leary, D.P.: *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia (2006)
18. 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)
19. 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)
20. Kindermann, S.: Discretization independent convergence rates for noise level-free parameter choice rules for the regularization of ill-conditioned problems. *Electron. Trans. Numer. Anal.* **40**, 58–81 (2013)
21. Li, F., Ng, M.K., Plemmons, R.J.: Coupled segmentation and denoising/deblurring for hyperspectral material identification. *Numer. Linear Algebra Appl.* **19**, 15–17 (2012)
22. Meng, J., Zhu, P.-Y., Li, H.-B.: A block GCROT(m, k) method for linear systems with multiple right-hand sides. *J. Comput. Appl. Math.* **255**, 544–554 (2014)
23. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.* **8**, 43–71 (1982)
24. Reichel, L., Rodriguez, G.: Old and new parameter choice rules for discrete ill-posed problems. *Numer. Algorithms* **63**, 65–87 (2013)
25. Saad, Y.: On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comp.* **48**, 651–662 (1987)
26. Toutounian, F., Karimi, S.: Global least squares method (GI-LSQR) for solving general linear systems with several right-hand sides. *Appl. Math. Comput.* **178**, 452–460 (2006)