Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems

A. H. Bentbib

Laboratory LAMAI, FSTG, University of Cadi Ayyad, Marrakesh, Morocco.

M. El Guide

Laboratory LAMAI, FSTG, University of Cadi Ayyad, Marrakesh, Morocco.

K. Jbilou

Université de Lille Nord de France, L.M.P.A, ULCO, 50 rue F. Buisson, BP699, F-62228 Calais-Cedex, France.

L. Reichel

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

Abstract

We consider the solution of large linear systems of equations that arise from the discretization of ill-posed problems. The matrix has a Kronecker product structure and the right-hand side is contaminated by measurement error. Problems of this kind arise, for instance, from the discretization of Fredholm integral equations of the first kind in two space-dimensions with a separable kernel and in image restoration problems. Regularization methods, such as Tikhonov regularization, have to be employed to reduce the propagation of the error in the right-hand side into the computed solution. We investigate the use of the global Golub–Kahan bidiagonalization method to reduce the given large problem to a small one. The small problem is solved by employing Tikhonov regularization. A regularization parameter determines the amount of regularization. The connection between global Golub–Kahan bidiagonalization and Gauss-type quadrature rules is exploited to inexpensively compute bounds that are useful for determining the regularization

Email addresses: a.bentbib@uca.ma (A. H. Bentbib), medelguide@gmail.com (M. El Guide), jbilou@lmpa.univ-littoral.fr (K. Jbilou), reichel@math.kent.edu (L. Reichel)

Preprint submitted to Journal of Computational and Applied Mathematics (JCAM)April 16, 2018

parameter by the discrepancy principle.

Keywords: Global Golub–Kahan bidiagonalization, Ill-posed problems, Gauss quadrature.

---

1. Introduction

Linear ill-posed problems arise in essentially every branch of science and engineering, including in remote sensing, computerized tomography, and image restoration. Discretization of these problems gives rise to linear systems of equations,

$$Hx = b, \qquad H \in \mathbb{R}^{N\times N}, \qquad x, b \in \mathbb{R}^N, \tag{1.1}$$

with a matrix that has many singular values of different orders of magnitude close to the origin; in particular, H may be singular. This makes the solution x of (1.1), if it exists, very sensitive to perturbations in the right-hand side b. In applications of interest to us, the vector b represents available data and is contaminated by an error $e \in \mathbb{R}^N$ that may stem from measurement and discretization errors. Therefore, straightforward solution of (1.1), generally, does not yield a useful result.

Let $\widehat{b} \in \mathbb{R}^N$ denote the unknown error-free vector associated with b, i.e.,

$$b = \widehat{b} + e. \tag{1.2}$$

We will assume the unavailable system of equations with error-free right-hand side,

$$Hx = \widehat{b}, \tag{1.3}$$

to be consistent and denote its solution of minimal Euclidean norm by $\widehat{x}$. It is our aim to determine an accurate approximation of $\widehat{x}$ by computing an approximate solution of the available linear system of equations (1.1). The first step in our solution process is to replace (1.1) by a nearby problem, whose solution is less sensitive to the error e in b. This replacement is commonly referred to as regularization. One of the most popular regularization methods is due to Tikhonov [7, 13]. In its simplest form, Tikhonov regularization replaces the linear system (1.1) by the minimization problem

$$\min_{x\in\mathbb{R}^N} \{\|Hx - b\|_2^2 + \mu^{-1}\|x\|_2^2\}. \tag{1.4}$$

Here $\mu > 0$ is a regularization parameter and $\|\cdot\|_2$ denotes the Euclidean vector norm. We will comment on the use of μ^{-1} instead of μ in (1.4) below. The minimization problem (1.4) has the unique solution

$$x_\mu := (H^T H + \mu^{-1} I_N)^{-1} H^T b \tag{1.5}$$

for any fixed $\mu > 0$. Here and throughout this paper I_N denotes the identity matrix of order N. The choice of μ affects how sensitive x_μ is to the error e in b, and how accurately x_μ approximates $\widehat{x}$. Many techniques for choosing a suitable value of μ have been analyzed and illustrated in the literature; see, e.g., [3, 7, 14, 21, 25] and references therein. In this paper we will use the *discrepancy principle*. It requires that a bound ε for $\|e\|_2$ be available and prescribes that $\mu > 0$ be determined so that $\|b - Hx_\mu\|_2 = \eta\varepsilon$ for a user chosen constant $\eta \geq 1$ that is independent of ε; see [7, 14, 25] for discussions on this parameter choice method. In the present paper, we will determine a value $\mu > 0$ such that

$$\varepsilon \leq \|b - Hx_\mu\|_2 \leq \eta\varepsilon, \tag{1.6}$$

where the constant $\eta > 1$ is independent of ε.

The computation of a μ-value such that the associated solution x_μ of (1.4) satisfies (1.6) generally requires the use of a zero-finder, see below, and typically $\|b - Hx_\mu\|_2$ has to be evaluated for several μ-values. This can be expensive when the matrix H is large. A solution method based on first reducing H to a small bidiagonal matrix with the aid of Golub–Kahan bidiagonalization (GKB) and then applying the connection between GKB and Gauss-type quadrature rules to determine an approximation of x_μ that satisfies (1.6) is discussed in [4].

It is the purpose of this paper to describe an analogous method for the situation when H is the Kronecker product of two matrices, $H_1 = [h_{i,j}^{(1)}] \in \mathbb{R}^{n\times n}$ and $H_2 \in \mathbb{R}^{m\times m}$, i.e.,

$$H = H_1 \otimes H_2 = \begin{bmatrix} h_{1,1}^{(1)}H_2 & h_{1,2}^{(1)}H_2 & \cdots & h_{1,n}^{(1)}H_2 \\ h_{2,1}^{(1)}H_2 & h_{2,2}^{(1)}H_2 & \cdots & h_{2,n}^{(1)}H_2 \\ \vdots & \vdots & & \vdots \\ h_{n,1}^{(1)}H_2 & h_{n,2}^{(1)}H_2 & \cdots & h_{n,n}^{(1)}H_2 \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{1.7}$$

with $N = mn$. Then the GKB method can be replaced by the global Golub–Kahan bidiagonalization (GGKB) method described by Toutounian and Karimi [26]. The latter method replaces matrix-vector product evaluations in the GKB method by matrix-matrix operations. It is well known that matrix-matrix operations execute efficiently on many modern computers; see, e.g., Dongarra et al. [6]. Iterative methods based on the GGKB method therefore can be expected to execute efficiently on many computers. We will exploit the relation between Gauss-type quadrature rules and the GGKB method to determine a value μ and an associated approximation of

the vector x_μ that satisfies (1.6). We remark that matrices H with a tensor product structure (1.7) arise in a variety of applications including when solving Fredholm integral equations of the first kind in two space-dimensions with a separable kernel, and in imaging restoration problems where the matrix H models a blurring operator. It is well known that many blurring matrices have Kronecker structure (1.7) or can be approximated well by a matrix with this structure; see [19, 20, 27].

In applications of our solution method described in Section 5 both the matrices H_1 and H_2 are square. Then H is square. This simplifies the notation and, therefore, only this situation will be considered. However, only minor modifications of the method are necessary to handle the situation when one or both of the matrices H_1 and H_2 are rectangular.

This paper continues our exploration of the application of global Krylov subspace methods to the solution of large-scale problems (1.1) with a Kronecker structure that was begun in [2]. There a scheme for computing an approximation of $\widehat{x}$ of prescribed norm is described. It was convenient to base this scheme on the global Lanczos tridiagonalization method and use its connection to Gauss-type quadrature rules. The paper focuses on the more common situation that a bound for the norm of the error e in b is available or can be estimated. Then the regularization parameter $\mu > 0$ can be determined by the discrepancy principle, i.e., so that the computed solution satisfies (1.6); see [7, 14]. The requirement (1.6) on the computed solution makes it natural to apply the GGKB method to develop an analogue of the approach in [4]. Timings and counts of arithmetic floating point operations (flops) show the structure-respecting method of the present paper to require less computing time and fewer flops than than the structure-igoring method described in [4], while giving an approximate solution of about the same quality. There are presently no other available methods that exploit the Kronecker structure and determine the regularization parameter μ by the discrepancy principle.

This paper is organized as follows. Section 2 discusses how the Kronecker product structure can be utilized when determining an approximate solution of (1.4) with the aid of the GGKB method. The connection between the GGKB method and Gauss-type quadrature rules is reviewed in Section 3, and the application of the GGKB method and Gauss-type quadrature to determine an approximate solution of (1.4) that satisfies (1.6) is described in Section 4. Numerical examples are presented in Section 5 and concluding remarks can be found in Section 6.

2. Kronecker structure

Introduce the operator vec, which transforms a matrix $A = [a_{i,j}] \in \mathbb{R}^{m\times n}$ to a vector $a \in \mathbb{R}^{mn}$ by stacking the columns of A from left to right, i.e,

$$a = [a_{1,1}, a_{2,1}, \ldots, a_{m,1}, a_{1,2}, a_{2,2}, \ldots, a_{m,2}, \ldots, a_{m,n}]^T. \tag{2.1}$$

We also need the inverse operator, mat, which transforms a vector (2.1) to an associated matrix $A = [a_{i,j}] \in \mathbb{R}^{m\times n}$. Thus,

$$\mathsf{vec}(A) = a, \qquad \mathsf{mat}(a) = A.$$

The Kronecker product satisfies the following relations for matrices A, B, C, D, X of suitable sizes,

$$\left.\begin{aligned} (A\otimes B)\mathsf{vec}(X) &= \mathsf{vec}(BXA^T), \\ (A\otimes B)^T &= A^T\otimes B^T, \\ (AB)\otimes(CD) &= (A\otimes C)(B\otimes D), \end{aligned}\right\} \tag{2.2}$$

see, e.g., [17] for proofs. For matrices $A, B \in \mathbb{R}^{m\times n}$, we define the inner product

$$\langle A, B\rangle_F := \mathsf{tr}(A^T B),$$

where $\mathsf{tr}(\cdot)$ denotes the trace, and we note that

$$\langle A, B\rangle_F = (\mathsf{vec}(A))^T \mathsf{vec}(B).$$

The Frobenius norm is associated with this inner product,

$$\|A\|_F := \langle A, A\rangle_F^{1/2},$$

and satisfies

$$\|A\|_F = \|\mathsf{vec}(A)\|_2. \tag{2.3}$$

Two matrices $A, B \in \mathbb{R}^{m\times n}$ are said to be F-orthogonal if

$$\langle A, B\rangle_F = 0.$$

Let H_1 and H_2 be Kronecker factors of the matrix H, cf. (1.7), and define the linear operator

$$\begin{aligned} \mathcal{A} : \mathbb{R}^{m\times n} &\to \mathbb{R}^{m\times n} \\ \mathcal{A}(X) &:= H_2 X H_1^T. \end{aligned}$$

Its transpose is given by $\mathcal{A}^T(X) := H_2^T X H_1$. We will need the symmetric linear operator

$$\breve{\mathcal{A}}(X) := (\mathcal{A}^T \circ \mathcal{A})(X),$$

where $\circ$ denotes composition. It can be expressed as

$$\breve{\mathcal{A}}(X) = H_2^T H_2 X H_1^T H_1. \tag{2.4}$$

Let $B := \mathsf{mat}(b) \in \mathbb{R}^{m\times n}$ and assume that $G := \mathcal{A}^T(B) \neq O$, where $O \in \mathbb{R}^{m\times n}$ denotes the zero matrix. Let $0 < \mu < \infty$. Then the equation

$$(\breve{\mathcal{A}} + \mu^{-1} I_m)(X) = G \tag{2.5}$$

has a unique solution $X_\mu \in \mathbb{R}^{m\times n}$. Using (2.4) this equation can be written as

$$H_2^T H_2 X H_1^T H_1 + \mu^{-1} X = H_2^T B H_1.$$

With the aid of (2.2), we can express the above equation in the form

$$((H_1 \otimes H_2)^T (H_1 \otimes H_2) + \mu^{-1} I_N)\mathsf{vec}(X) = (H_1 \otimes H_2)^T \mathsf{vec}(B).$$

This equation is equivalent to the normal equations associated with (1.4). Therefore, $X_\mu = \mathsf{mat}(x_\mu)$.

Proposition 2.1. *Let $0 < \mu < \infty$ and let X_μ be the unique solution of (2.5). Introduce the function*

$$\phi(\mu) := ||B - \mathcal{A}(X_\mu)||_F^2. \tag{2.6}$$

Let $\eta > 1$ be the same as in (1.6). Then

$$\varepsilon^2 \leq \phi(\mu) \leq \eta^2 \varepsilon^2 \tag{2.7}$$

is equivalent to that $x_\mu = \mathsf{vec}(X_\mu)$ satisfies (1.6) with $b = \mathsf{vec}(B)$.

Proof. Using (2.3) and (2.2), in order, yields

$$\phi(\mu) = \|B - H_2 X_\mu H_1^T\|_F^2 = \|\mathsf{vec}(B) - \mathsf{vec}(H_2 X_\mu H_1^T)\|_2^2 = \|b - (H_1 \otimes H_2) x_\mu\|_2^2,$$

and the proposition follows from $H = H_1 \otimes H_2$. □

The proposition implies that we can use X_μ instead of x_μ when determining a value of μ such that (1.6) holds. We will apply the following implementation of the GGKB method to determine an approximation of X_μ. The GGKB method generates two sequences of F-orthogonal matrices as well as a bidiagonal matrix.

Algorithm 1: The GGKB method

1. Set $\sigma_1 = ||B||_F$, $U_1 = B/\sigma_1$, $V_1 = O$
2. For $j = 1, 2, \ldots, k$
   (a) $\tilde{V}_j = \mathcal{A}^T(U_j) - \sigma_j V_j$
   (b) $\rho_j = \|\tilde{V}_j\|_F$
   (c) $V_j = \tilde{V}_j/\rho_j$
   (d) $\tilde{U}_{j+1} = \mathcal{A}(V_j) - \rho_j U_j$
   (e) $\sigma_{j+1} = ||\tilde{U}_{j+1}||_F$
   (f) $U_{j+1} = \tilde{U}_{j+1}/\sigma_{j+1}$

   EndFor

The above algorithm determines the decompositions

$$\left.\begin{aligned} \mathcal{U}_{k+1}(\sigma_1 e_1 \otimes I_n) &= B, \\ \big[\mathcal{A}(V_1), \mathcal{A}(V_2), \ldots, \mathcal{A}(V_k)\big] &= \mathcal{U}_{k+1}(\bar{C}_k \otimes I_n), \\ \big[\mathcal{A}^T(U_1), \mathcal{A}^T(U_2), \ldots, \mathcal{A}^T(U_k)\big] &= \mathcal{V}_k(C_k^T \otimes I_n), \end{aligned}\right\} \tag{2.8}$$

where $e_j = [1, 0, \ldots, 0]^T$ denotes the first axis vector. The matrix

$$\mathcal{V}_k = [V_1, V_2, \ldots, V_k] \in \mathbb{R}^{m\times kn}, \qquad \mathcal{U}_{k+1} = [U_1, U_2, \ldots, U_{k+1}] \in \mathbb{R}^{m\times(k+1)n},$$

have F-orthonormal "matrix columns" $V_j \in \mathbb{R}^{m\times n}$ and $U_j \in \mathbb{R}^{m\times n}$, respectively, i.e.,

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

Finally, the matrix $C_k \in \mathbb{R}^{k\times k}$ is bidiagonal,

$$C_k = \begin{bmatrix} \rho_1 & & & & \\ \sigma_2 & \rho_2 & & & \\ & \ddots & \ddots & & \\ & & \sigma_{k-1} & \rho_{k-1} & \\ & & & \sigma_k & \rho_k \end{bmatrix} \tag{2.9}$$

and so is

$$\bar{C}_k = \begin{bmatrix} C_k \\ \sigma_{k+1} e_k^T \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}.$$

We remark that the matrix columns $U_1, U_2, U_3, \ldots$, are only required to advance the computations of Algorithm 1. Therefore, they all can be stored in the same memory location. Also, the matrix column $\tilde{U}_{j+1}$ can use this location. The matrix columns $V_1, V_2, \ldots, V_k$ will be used to determine an approximate solution of (1.1) and, therefore, cannot be overwritten. However, the matrix columns $\tilde{V}_j$ and V_{j+1} may share the same storage location.

3. Gauss quadrature

This section discusses how the GGKB method is related to quadrature rules of Gauss-type. The connection follows via the relation between global Lanczos tridiagonalization and Gauss quadrature. Combining the equations (2.8) yields

$$[\breve{\mathcal{A}}(U_1), \breve{\mathcal{A}}(U_2), \ldots, \breve{\mathcal{A}}(U_k)] = \mathcal{U}_k(C_k C_k^T \otimes I_n) + \sigma_{k+1}[O, \ldots, O, U_{k+1}], \quad (3.1)$$

where O is a zero matrix. This decomposition also can be determined by applying k steps of the global symmetric Lanczos (GSL) method to the matrix $\breve{\mathcal{A}}$ with initial block vector B. The GSL method is described in [18] and its relation to Gauss quadrature is discussed in [2]. We therefore only provide an outline. We remark that the relation between the standard symmetric Lanczos method and Gauss-type quadrature is well known; see, e.g., [4, 11, 12].

Proposition 3.1. *Assume that $H^T b \neq 0$. Then the function* (2.6) *can be expressed as*

$$\phi(\mu) = b^T(\mu H H^T + I_{mn})^{-2} b. \quad (3.2)$$

Proof. The proof of Proposition 2.1 shows that the function (2.6) can be written as

$$\phi(\mu) = \|b - Hx_\mu\|_2^2.$$

Substituting (1.5) into the right-hand side and using the identity, for $\mu > 0$,

$$H(H^T H + \mu^{-1} I_{mn})^{-1} H^T = I_{mn} - (\mu H H^T + I_{mn})^{-1} \quad (3.3)$$

gives (3.2). □

Substituting the spectral factorization $HH^T = W\Lambda W^T$, where $W \in \mathbb{R}^{mn\times mn}$ is orthogonal and $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_{mn}] \in \mathbb{R}^{mn\times mn}$, into the right-hand side of (3.2) yields

$$\phi(\mu) = \sum_{j=1}^{mn} f_\mu(\lambda_j) w_j^2. \quad (3.4)$$

Here $[w_1, w_2, \ldots, w_{mn}]^T := W^T b$ and

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

The expression (3.4) is a Stieltjes integral, which we write as

$$\phi(\mu) = \int f_\mu(t) d\omega(t). \quad (3.5)$$

The distribution function ω associated with the measure $d\omega$ can be chosen as a nondecreasing piecewise constant function with nonnegative jumps w_j^2 at the eigenvalues λ_j. Since HH^T is positive semidefinite, the support of the measure $d\omega$ lives on the nonnegative real axis.

Define the integral

$$\mathcal{I}f := \int f(t)d\omega(t)$$

for suitable functions f and let $\mathcal{G}_k$ denote the k-point Gauss quadrature rule associated with $d\omega$. It is characterized by the property that

$$\mathcal{G}_k p = \mathcal{I}p \qquad \forall p \in \mathbb{P}_{2k-1},$$

where $\mathbb{P}_{2k-1}$ denotes the set of polynomials of degree at most $2k-1$. The global Lanczos decomposition (3.1) provides a way to evaluate this Gauss rule without explicit knowledge of the measure $d\omega$. We have that for suitable functions f,

$$\mathcal{G}_k f = \|B\|_F^2 e_1^T f(T_k) e_1,$$

where the matrix $T_k := C_k C_k^T$ is symmetric and tridiagonal; see [2] for a proof. Analogous results that relate standard Golub–Kahan bidiagonalization to Gauss quadrature have been shown by Golub and Meurant [11, 12] and are applied in [4]. Gautschi [10] provides a nice fairly recent discussion on Gauss quadrature and orthogonal polynomials; see also [9].

We are particularly interested in the Gauss rule

$$\mathcal{G}_k f_\mu = \|B\|_F^2 e_1^T (\mu C_k C_k^T + I_k)^{-2} e_1. \tag{3.6}$$

Using the remainder formula for Gauss quadrature and the fact that all even-order derivatives of the integrand $t \to f_\mu(t)$ in (3.5) are positive, one can show that, generically,

$$\mathcal{G}_1 f_\mu < \cdots < \mathcal{G}_{k-1} f_\mu < \mathcal{G}_k f_\mu < \phi(\mu); \tag{3.7}$$

see, e.g., [2, 23] for details.

Let $\mathcal{R}_{k+1}$ denote the $(k+1)$-point Gauss–Radau rule for the measure $d\omega$ with a fixed node $t_0 = 0$. Then

$$\mathcal{R}_{k+1} p = \mathcal{I}p \qquad \forall p \in \mathbb{P}_{2k}.$$

This rule, when applied to the integration of f_μ, can be can expressed as

$$\mathcal{R}_{k+1} f_\mu = \|B\|_F^2 e_1^T (\mu \bar{C}_k \bar{C}_k^T + I_{k+1})^{-2} e_1;$$

see [2]. The remainder formula for Gauss–Radau quadrature rules and the fact that all odd-order derivatives of $t \to f_\mu(t)$ are negative yield that, generically,

$$\phi(\mu) < \mathcal{R}_{k+1} f_\mu < \mathcal{R}_k f_\mu < \cdots < \mathcal{R}_1 f_\mu; \tag{3.8}$$

see, e.g., [2, 23].

We conclude that pairs of Gauss and Gauss–Radau quadrature rules $\mathcal{G}_k f_\mu$ and $\mathcal{R}_{k+1} f_\mu$ yield lower and upper bounds, respectively, for $\phi(\mu)$. We evaluate these rules by first executing Algorithm 1 and then solving the least-squares problem

$$\min_{z \in \mathbb{R}^k} \left\| \begin{bmatrix} \mu^{1/2} C_k^T \\ I_k \end{bmatrix} z - e_{k+1} \right\|_2^2. \tag{3.9}$$

The solution, denoted by z_k, satisfies

$$(\mu C_k C_k^T + I_k) z_k = e_1. \tag{3.10}$$

It follows from (3.6) that

$$\mathcal{G}_k f_\mu = \|B\|_F^2 z_k^T z_k.$$

The special structure of the least-squares problem (3.9) makes it possible to evaluate $\mathcal{G}_k f_\mu$ in only $\mathcal{O}(k)$ arithmetic floating-point operations for each value of μ. Typically $\mathcal{G}_k f_\mu$ has to be evaluated for several μ-values; see Section 4. The evaluation of $\mathcal{R}_{k+1} f_\mu$ can be carried out analogously; the matrix C_k^T in (3.9) has to be replaced by $\bar{C}_k^T$; see [2]. The reason for solving the least-squares problem instead of the associated normal equations (3.10) is that the solution of the former generally is less sensitive to errors in the data and to round-off errors introduced during the computations.

4. Parameter selection and computation of an approximate solution

This section describes how the bounds for $\phi(\mu)$ described in the previous section can be used to determine a suitable number of steps k of the GGKB method, a value μ_k of the regularization parameter, and an approximation $x_{\mu_k,k}$ of the vector x_{μ_k}, defined by (1.5) with $\mu = \mu_k$, that satisfies (1.6).

For a given value of $k \geq 2$, we solve the nonlinear equation

$$\mathcal{G}_k f_\mu = \varepsilon^2 \tag{4.1}$$

for μ. Because we use the parameter μ in (1.4), instead of $1/\mu$, the left-hand side is a decreasing convex function of μ. There is a unique solution, denoted by μ_ε, of

$$\phi(\mu) = \varepsilon^2 \tag{4.2}$$

for almost all values of $\varepsilon > 0$ of practical interest and therefore also of (4.1) for k sufficiently large; see [2, 4] for analyses. Various zero-finders can be applied, including Newton's method; see [22]. The evaluation of each iterate requires the solution of a least-squares problem (3.9). The following result shows that the regularization parameter determined by solving (4.1) provides more regularization than the parameter obtained by solving (4.2).

Proposition 4.1. *Let μ_k solve* (4.1) *and let μ_ε solve* (4.2). *Then, generically, $\mu_k < \mu_\varepsilon$.*

Proof. It follows from (3.2) that $\phi(\mu)$ is a decreasing and convex function for $\mu \geq 0$ with $\phi(0) = \|b\|_2^2$. Similarly, by (3.6), $\mathcal{G}_k f_\mu$ is a decreasing and convex function for $\mu \geq 0$ with $\mathcal{G}_k f_0 = \|b\|_2^2$. Generically, $\phi(\mu) > \mathcal{G}_k f_\mu$ for $\mu > 0$; cf. (3.7). Therefore, typically, $\mu_k < \mu_\varepsilon$. We have equality in the rare event of breakdown of the recursion formulas for the GGKB method. Assume that the $\rho_k > 0$ in (2.9). Then the matrix (2.9) is nonsingular and the solution μ_k of (4.1) exists if $0 < \varepsilon < \|b\|_2$. Let $P_{\mathcal{N}(HH^T)}$ denote the orthogonal projector onto the null space of HH^T. Then the solution μ_ε of (4.2) exist if $\|P_{\mathcal{N}(HH^T)}b\|_2 < \varepsilon < \|b\|_2$. □

Having computed μ_k, we check whether

$$\mathcal{R}_{k+1} f_\mu \leq \eta^2 \varepsilon^2 \tag{4.3}$$

holds for $\mu = \mu_k$. If this is the case, then it follows from (3.7) and (3.8) that (2.7) is valid for $\mu = \mu_k$. If, on the other hand, the inequality (4.3) is violated for $\mu = \mu_k$, then we increase k by one, compute μ_{k+1}, and check whether (4.3) holds with $k+1$ replaced by $k+2$ for $\mu = \mu_{k+1}$. For most problems of interest, the Gauss and Gauss-Radau approximations (3.7) and (3.8) converge quite rapidly to $\phi(\mu_k)$ as k increases. Therefore, the bound (4.3) typically holds already for a fairly small value of k.

Assume that μ_k satisfies (4.1) and (4.3). Then we determine an approximate solution of (1.4) with the aid of the global Golub–Kahan decomposition (2.8) as follows. First we determine the solution y_{k,μ_k} of

$$(\bar{C}_k^T \bar{C}_k + \mu_k^{-1} I_k) y = \sigma_1 \bar{C}_k^T e_1, \qquad \sigma_1 = \|B\|_F. \tag{4.4}$$

It is computed by solving the least-squares problem

$$\min_{y\in\mathbb{R}^k} \left\| \begin{bmatrix} \mu_k^{1/2}\bar{C}_k \\ I_k \end{bmatrix}_k y - \sigma_1 \mu_k^{1/2} e_1 \right\|_2 .$$

Similarly as above, we solve this least-squares problem instead of the associated normal equations (4.4) because of the better numerical properties of the latter. Finally, our approximate solution of (2.5) is determined by

$$X_{k,\mu_k} = \mathcal{V}_k(y_{k,\mu_k} \otimes I_n). \tag{4.5}$$

Proposition 4.2. *The approximate solution* (4.5) *of* (2.5) *satisfies*

$$\varepsilon \leq \|B - \mathcal{A}(X_{k,\mu_k})\|_F \leq \eta\varepsilon. \tag{4.6}$$

Proof. Using the representation (4.5), and applying (2.8) as well as (2.2), shows that

$$\mathcal{A}(X_{k,\mu_k}) = \mathcal{U}_{k+1}(\bar{C}_k \otimes I_n)(y_{k,\mu_k} \otimes I_n) = \mathcal{U}_{k+1}(\bar{C}_k y_{k,\mu_k} \otimes I_n).$$

Substituting the above expression into (4.6) and again using (2.8) yields

$$\begin{aligned} & \|\mathcal{U}_{k+1}(\sigma_1 e_1 \otimes I_n) - \mathcal{U}_{k+1}(\bar{C}_k y_{k,\mu_k} \otimes I_n)\|_F^2 \\ = \;& \|(\sigma_1 e_1 \otimes I_n) - (\bar{C}_k y_{k,\mu_k} \otimes I_n)\|_F^2 \\ = \;& \|\sigma_1 e_1 - \bar{C}_k y_{k,\mu_k}\|_2^2, \end{aligned}$$

where we recall that $\sigma_1 = \|B\|_F$. We now express y_{k,μ_k} with the aid of (4.4), and apply the identity (3.3) with H replaced by $\bar{C}_k$, to obtain

$$\begin{aligned} & \|B - \mathcal{A}(X_{k,\mu_k})\|_F^2 \\ = \;& \sigma_1^2 \|e_1 - \bar{C}_k(\bar{C}_k^T\bar{C}_k + \mu_k^{-1} I_k)^{-1}\bar{C}_k^T e_1\|_2^2 \\ = \;& \sigma_1^2 e_1^T(\mu_k \bar{C}_k\bar{C}_k^T + I_{k+1})^{-2} e_1 \\ = \;& \mathcal{R}_{k+1} f_{\mu_k}. \end{aligned}$$

The proposition now follows from (4.3) and the fact that $\varepsilon^2 = \mathcal{G}_k f_{\mu_k} \leq \mathcal{R}_{k+1} f_{\mu_k}$. □

The following algorithm summarizes the computation of (4.5).

Algorithm 2. The GGKB-Tikhonov method

Input: H_1, H_2, B, ε, $\eta \geq 1$

1. Set $k = 2$ (k is the number of global Lanczos steps.)
   Let $U_1 := B/||B||_F$;
2. Determine the F-orthonormal bases $\{U_j\}_{j=1}^{k}$ and $\{V_j\}_{j=1}^{k}$, and the bidiagonal matrices C_k and $\bar{C}_k$ with Algorithm 1.
3. Determine μ_k that satisfies (4.1) and (4.3) as described above. This may require k to be increased, in which case one returns to step 2.
4. Determine y_{k,μ_k} and X_{k,μ_k} as described above.

We comment on the complexity of Algorithm 2. First note that the overall computational cost for Algorithm 2 is dominated by the work required to determine U_j and V_j at step 2 (here we ignore the computational cost of evaluating $\mathcal{A}^T(U_j)$ and $\mathcal{A}(V_j)$ in Algorithm 1). The computational effort required to determine U_j and V_j is dominated by the evaluation of four matrix-matrix products, which demands approximately $4(m + n)N$ flops. These matrix-matrix operations need fewer flops than structure-ignoring evaluation of matrix-vector products with the large matrix H and its transpose, which requires approximately $4N^2$ flops.

5. Numerical examples

This section presents a few representative numerical experiments. All computations were carried out using the MATLAB environment on an Intel(R) Core (TM) 2 Duo CPU T5750 computer with 3 GB of RAM. The computations were done with approximately 15 decimal digits of relative accuracy.

Let $\widehat{x} := \mathsf{vec}(\widehat{X})$ denote the error-free exact solution of the linear system of equations (1.1), let $\widehat{B} := H_2\widehat{X}H_1^T$ and $B := \widehat{B} + E$, and define

$$\widehat{b} := \mathsf{vec}(\widehat{B}), \qquad b := \mathsf{vec}(B), \qquad e := \mathsf{vec}(E),$$

where the error matrix E has normally distributed entries with zero mean and is normalized to correspond to a specific noise level

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

The sizes of the matrices is specified in the examples below.

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

$$\frac{||\widehat{X} - X_k||_F}{||\widehat{X}||_F}$$

of the computed approximate solution $X_k = X_{k,\mu_k}$ determined by Algorithm 2. The first three examples are concerned with the solution of Fredholm integral equations of the first kind in two space-dimensions with a separable kernel. Discretization gives matrices that are severely ill-conditioned. The last two examples discuss image restoration problems. Overviews of image restoration problems can be found in [5, 16].

Example 1.

Let the nonsymmetric matrix H_1 and the symmetric matrix H_2, both of size 1500×1500, be determined by the MATLAB programs `baart` and `foxgood`, respectively, in the Regularization Tools package by Hansen [15]. These programs provide discretizations of Fredholm integral equations of the first kind described in [1, 8]. The computed condition numbers of these matrices are $\kappa(H_1) = 3.72 \times 10^{19}$ and $\kappa(H_2) = 1.03 \times 10^{21}$. Since $\kappa(H) = \kappa(H_1)\kappa(H_2)$, the matrix H is numerically singular. The right-hand side matrix $\widehat{B}$ and the exact solution $\widehat{X}$ also are generated with the aid of the codes `baart` and `foxgood`. Table 5.1 displays the computed regularization parameters and the relative error in computed approximate solutions determined by Algorithm 2 with $\eta = 1.1$ for different noise levels, as well as the number of iterations required to satisfy the stopping criterion.

Table 5.1: Results for Example 1.

Noise level	Iterations	μ	Relative error
0.01	4	5.77×10^2	2.08×10^{-1}
0.001	7	2.64×10^4	1.22×10^{-1}

Example 2.

Instead of determining an approximate solution of (1.1) with Algorithm 2, one could compute the singular value decompositions (SVDs) of the matrices H_1 and H_2. The SVD of the matrix H then can be determined from the SVDs of H_1 and H_2. We can apply the SVD of H to determine a suitable regularization parameter $\mu > 0$. Specifically, we solve the equation

$\|b - Hx_\mu\|_2 = \eta\varepsilon$ for μ by Newton's method without explicitly forming the matrices in the SVD of H. Knowing μ allows us to compute the Tikhonov solution (1.5) without explicitly forming the matrices in the SVD of H. This approach of determining the regularization parameter and computing the corresponding regularized solution (1.5) is attractive due to its simplicity when the matrices H_1 and H_2 are small. However, the approach is slow for large matrices H_1 and H_2. To illustrate this, we solve the same problem as in Example 1 with a finer discretization. Thus, let $H_1, H_2 \in \mathbb{R}^{2000\times 2000}$ be determined by the MATLAB codes `baart` and `foxgood`, respectively, from [15]. The matrices $\widehat{B}$ and $\widehat{X}$ are generated in the same manner as in Example 1 with the matrix B such that the noise level is $\nu = 0.01$. We let the safety factor for the discrepancy principle be $\eta = 1.1$. Then Algorithm 2 terminates after 4 iterations with the regularization parameter $\mu_4 = 5.77 \times 10^2$ and an approximate solution, defined by (4.5) with $k = 4$, with relative error 2.09×10^{-1}. The computing time for this experiment is 37.96 seconds.

When we instead use the SVD of H, we obtain the regularization parameter $\mu = 22.8$ and an approximate solution with relative error 2.14×10^{-1}. The computing time for this method is 47.95 seconds. The difference in the values of the regularization parameter for these approaches depends on that in Algorithm 2 the regularization parameter is determined for the solution in a subspace of low dimension, while the solution determined by the SVD approach lives in the whole space.

The computations of this example illustrate that the computations with the global Golub–Kahan method are faster than using the SVD of H, even when the structure of the latter is exploited. The difference in computing time is even more pronounced for larger problems. We remark that also for problems of the size considered in Example 1, the global Golub–Kahan method is faster than using the SVD of H.

Example 3.

We consider the Fredholm integral equation

$$\int\int_\Omega K(x,y,s,t)f(s,t)dsdt = g(x,y), \qquad (x,y) \in \Omega, \tag{5.1}$$

where $\Omega = [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$. The kernel is given by

$$K(x,y,s,t) = k(x,s)k(y,t), \quad (x,y),(s,t) \in \Omega,$$

where

$$k(s,x) = (\cos(s) + \cos(x))^2(\sin(\xi)/\xi)^2, \quad \xi = \pi(\sin(s) + \sin(x)).$$

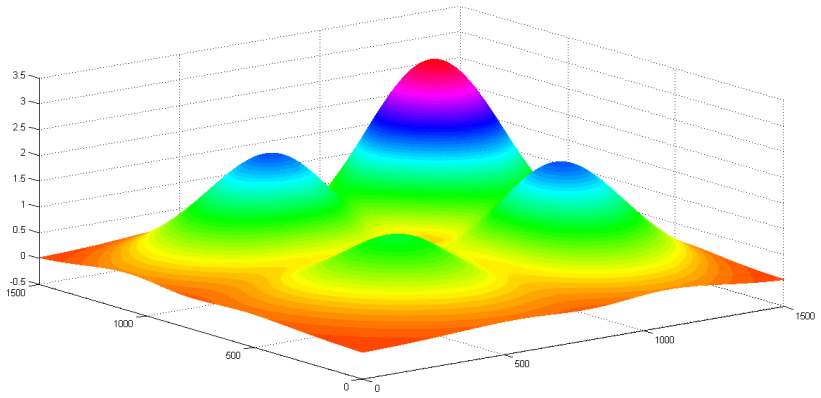

Figure 5.1: Example 3: Approximation X_{13} of $\hat{X}$ determined by Algorithm 2 for noise level 0.01.

The solution is the sum of two Gaussians in each space dimension.

We discretize (5.1) by a Nyström method based on the composite trapezoidal rule with equidistant nodes in each space-dimension. Code is available at [24]. Specifically, we determine the nonsymmetric matrix $H_1 \in \mathbb{R}^{1500\times 1500}$, from which we obtain the discretization $H = H_1 \otimes H_1$ of the integral operator (5.1), as well as the exact solution $\widehat{X} \in \mathbb{R}^{1500\times 1500}$ of the discretized problem and the associated exact right-hand side $\widehat{B}$. Table 5.2 shows the computed regularization parameters and the relative error in approximate solutions determined by Algorithm 2 with $\eta = 1.01$ and different noise levels, as well as the number of iterations required to satisfy the stopping criterion. Figure 5.1 displays the computed approximate solution X_{13} obtained when the noise level of the available data (right-hand side) is 0.01.

Table 5.2: Results for Example 3.

Noise level	Iterations	μ	Relative error
0.01	13	2.46×10^{2}	1.59×10^{-1}
0.001	32	2.67×10^{4}	6.97×10^{-2}

Example 4.

This and the following examples are concerned with the restoration of images that have been contaminated by blur and noise. Let the entries of the vector $\widehat{x}$ be pixel values for a desired, but unknown, image. The matrix H is a discretization of a blurring operator and equation (1.3) shows that $\widehat{b}$ represents a blurred, but noise-free, image. The vector b in (1.2) represents the available blur- and noise-contaminated image associated with $\widehat{x}$. The blurring matrix H is determined by a point-spread function (PSF), which determines how each pixel is smeared out (blurred), and by the boundary conditions, which specify our assumptions on the scene just outside the available image; see [5, 16] for details.

In some cases the horizontal and vertical components of the PSF can be written as a product of two functions, one depending on the horizontal coordinate and the other one on the vertical coordinate, only. In this situation, the blurring matrix H can be expressed as a Kronecker product $H = H_1 \otimes H_2$. Let the matrix $\widehat{X} = \mathsf{mat}(\widehat{x})$ be of suitable size. Then the blurred image can be represented as $H_2\widehat{X}H_1^T$; cf. (2.2). The blur- and noise-contaminated image is represented by the matrix $H_2\widehat{X}H_1^T + E$, where E is the noise matrix. Also when H cannot be written as a Kronecker product of two matrices, it may be possible to approximate H well by such a Kronecker product; see [19, 20, 27].

In this example, we seek to restore an image that has been contaminated by blur that is defined by a Gaussian PSF,

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

and by noise. The Dirichlet zero boundary condition is imposed. The blurring matrix then is the Kronecker product of a symmetric Toeplitz matrices with itself $H = H_1 \otimes H_1$, where $H_1 = [h_{ij}]$ with

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

The matrix H models atmospheric turbulence blur. We let $\sigma = 2.5$ and $r = 6$.

The original image $\widehat{X} \in \mathbb{R}^{256\times 256}$ is the `Enamel` image from MATLAB. It is shown in the left-hand side of Figure 5.2. The associated blurred and noisy image $B := H_2\widehat{X}H_1^T + E$ is shown in the middle of Figure 5.2; the noise level is 0.001. The restoration determined by Algorithm 2 with $\eta = 1.1$ is displayed in the right-hand side of the figure.

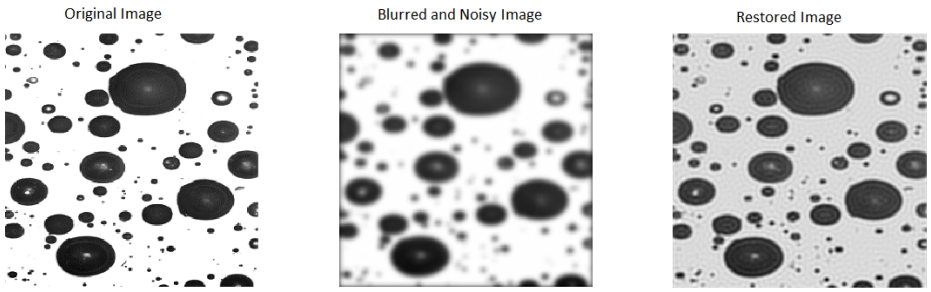

Figure 5.2: Example 4: Original image (left), degraded image (center), and restored image (right) for noise of level 0.001.

For comparison, we determine a regularization parameter and an approximate solution using the numerical method described in [4]. This method uses (standard) Golub–Kahan bidiagonalization instead of global Golub–Kahan bidiagonalization, and explores the connection between (standard) Golub–Kahan bidiagonalization and Gauss quadrature rules for solving large ill-conditioned linear systems of equations (1.1) without exploiting the structure of the matrix H. We refer to this method as GKB in Table 5.3. The table compares results obtained by Algorithm 2 and GKB, including the relative errors of the restorations, the number of iterations, and the CPU times required for two noise levels. Algorithm 2 is seen to require less CPU time than GKB and give about the same quality of the computed restoration as GKB.

Table 5.3: Results for Example 4.

Noise level	Iterations	Method	μ	Relative error	CPU-time (sec)
0.01	14	Algorithm 2	4.66×10^3	1.02×10^{-1}	0.50
	14	GKB	4.66×10^3	1.02×10^{-1}	1.35
0.001	62	Algorithm 2	1.71×10^4	8.00×10^{-2}	2.23
	62	GKB	1.71×10^4	8.02×10^{-2}	11.44

Example 5.

The original image is the `iograyBorder` image of dimension 256×256 from MATLAB. It is shown on the left-hand side of Figure 5.3. The blurring

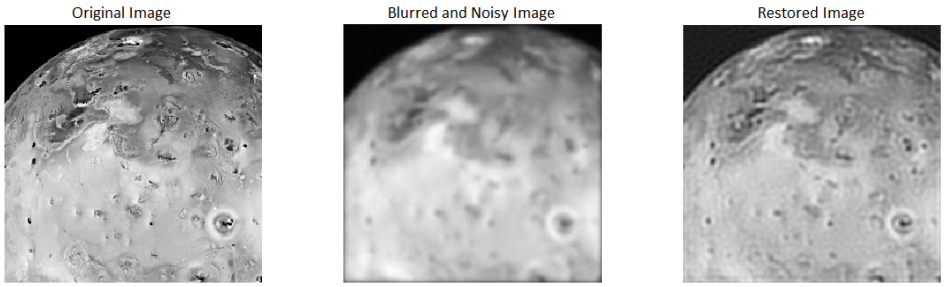

Figure 5.3: Example 5: Original image (left), degraded image (center), and restored image (right) for noise of level 0.01.

matrix $H = H_1 \otimes H_2$ is the same as in Example 3. The blurred and noisy image shown in the middle of Figure 5.3 has noise level 0.01. The restored image determined with Algorithm 2 with $\eta = 1$ is shown in the right-hand side of Figure 5.3. The number of iterations, relative errors of the restored images, and computed regularization parameters are shown in Table 5.4 for two noise levels.

Table 5.4: Results for Example 5.

Noise level	Iterations	μ	Relative error	CPU time (sec)
0.01	19	4.80×10^4	8.85×10^{-2}	1.09
0.001	75	2.60×10^4	7.58×10^{-2}	2.13

6. Conclusion

This paper describes an iterative scheme based on the global Golub–Kahan bidiagonalization method for the approximate solution of the Tikhonov minimization problem (1.4) when the matrix H has a Kronecker structure. The method exploits the relation between global Golub–Kahan bidiagonalization and Gauss-type quadrature to inexpensively determine the regularization parameter so that the discrepancy principle is satisfied. The Kronecker structure makes it possible to replace matrix-vector product evaluations in the scheme [4] by matrix-matrix product computations. The latter products execute more efficiently than the former on many computers.

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] A. H. Bentbib, M. El Guide, K. Jbilou, and L. Reichel, *A global Lanczos method for image restoration*, J. Comput. Appl. Math., 300 (2016), pp. 233–244.

[3] C. Brezinski, G. Rodriguez, and S. Seatzu, *Error estimates for the regularization of least squares problems*, Numer. Algorithms, 51 (2009), pp. 61–76.

[4] D. Calvetti and L. Reichel, *Tikhonov regularization of large linear problems*, BIT, 43 (2003), pp. 263–283.

[5] B. Chalmond, *Modeling and Inverse Problems in Image Analysis*, Springer, New York, 2003.

[6] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, 1998.

[7] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.

[8] L. Fox and E. T. Goodwin, *The numerical solution of non-singular linear integral equations*, Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Eng. Sci., 245:902 (1953), pp. 501–534.

[9] W. Gautschi, *The interplay between classical analysis and (numerical) linear algebra – a tribute to Gene H. Golub*, Electron. Trans. Numer. Anal., 13 (2002), pp. 113–147.

[10] W. Gautschi, *Orthogonal Polynomials: Approximation and Computation*, Oxford University Press, Oxford, 2004.

[11] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D. F. Griffiths and G. A. Watson, eds., Longman, Essex, 1994, pp. 105–156.

[12] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, 2010.

[13] C. W. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Integral Equations of the First Kind*, Pitman, Boston, 1984.

[14] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998

[15] P. C. Hansen, *Regularization tools version 4.0 for MATLAB 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.

[16] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

[17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[18] K. Jbilou, H. Sadok, and A. Tinzefte, *Oblique projection methods for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 20 (2005), pp. 119–138.

[19] J. Kamm and J. G. Nagy, *Kronecker product and SVD approximations in image restoration*, Linear Algebra Appl., 284 (1998), pp. 177–192.

[20] J. Kamm and J. G. Nagy, *Kronecker product approximations for restoration image with reflexive boundary conditions*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 829–841.

[21] 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.

[22] J. Lampe, L. Reichel, and H. Voss, *Large-scale Tikhonov regularization via reduction by orthogonal projection*, Linear Algebra Appl., 436 (2012), pp. 2845–2865.

[23] G. Lopez Lagomasino, L. Reichel, and L. Wunderlich, *Matrices, moments, and rational quadrature*, Linear Algebra Appl., 429 (2008), pp. 2540–2554.

[24] A. Neuman, L. Reichel, and H. Sadok, *Algorithms for range restricted iterative methods for linear discrete ill-posed problems*, Numer. Algorithms, 59 (2012), pp. 325–331. Code is available in Netlib at `http://www.netlbib.org/numeralgo/` in the package na33.

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

[26] F. Toutounian and S. Karimi, *Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides*, Appl. Math. Comput., 178 (2006), pp. 452–460.

[27] C. F. Van Loan and N. P. Pitsianis, *Approximation with Kronecker products*, in Linear Algebra for Large Scale and Real Time Applications, M. S. Moonen and G. H. Golub, eds., Kluwer, Dordrecht, 1993, pp. 293–314.