BIT manuscript No.
(will be inserted by the editor)

GCV for Tikhonov regularization by partial SVD

Caterina Fenu · **Lothar Reichel** ·
Giuseppe Rodriguez · **Hassane Sadok**

Received: date / Accepted: date

Abstract Tikhonov regularization is commonly used for the solution of linear discrete ill-posed problems with error-contaminated data. A regularization parameter that determines the quality of the computed solution has to be chosen. One of the most popular approaches to choosing this parameter is to minimize the Generalized Cross Validation (GCV) function. The minimum can be determined quite inexpensively when the matrix A that defines the linear discrete ill-posed problem is small enough to rapidly compute its singular value decomposition (SVD). We are interested in the solution of linear discrete ill-posed problems with a matrix A that is too large to make the computation of its complete SVD feasible, and show how upper and lower bounds for the numerator and denominator of the GCV function can be determined fairly inexpensively for large matrices A by computing only a few of the largest singular values and associated singular vectors of A. These bounds are used to determine a suitable value of the regularization parameter. Computed examples illustrate the performance of the proposed method.

Keywords Generalized cross validation · Tikhonov regularization · partial singular value decomposition

Mathematics Subject Classification (2000) 65F22 · 65F20 · 65R32

C. Fenu
Dipartimento di Informatica, Università di Pisa, Largo Pontecorvo 3, 56127 Pisa, Italy.
E-mail: kate.fenu@unica.it

L. Reichel
Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.
E-mail: reichel@math.kent.edu

G. Rodriguez
Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy.
E-mail: rodriguez@unica.it

H. Sadok
Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral, Centre Universtaire de la Mi-Voix, Batiment H. Poincarré, 50 Rue F. Buisson, BP 699, 62228 Calais cedex, France.
E-mail: sadok@lmpa.univ-littoral.fr

1 Introduction

We are interested in the solution of large-scale least-squares problems

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

where $A \in \mathbb{R}^{m\times n}$ is a large matrix whose singular values decay gradually to zero without a significant gap. In particular, the Moore–Penrose pseudoinverse of A, denoted by $A^\dagger$, is of very large norm. Least-squares problems of this kind often are referred to as linear discrete ill-posed problems. The data vector $\mathbf{b} \in \mathbb{R}^m$ is assumed to stem from measurements and be contaminated by a measurement error $\mathbf{e} \in \mathbb{R}^m$ of unknown size. We will assume that $m \geq n$ for notational simplicity, but the method discussed also can be applied when $m < n$ after appropriate modifications.

Let $\widehat{\mathbf{b}}$ denote the unknown error-free vector associated with $\mathbf{b}$. Thus, $\mathbf{b} = \widehat{\mathbf{b}} + \mathbf{e}$. We would like to compute $\widehat{\mathbf{x}} := A^\dagger \widehat{\mathbf{b}}$. The solution

$$A^\dagger \mathbf{b} = \widehat{\mathbf{x}} + A^\dagger \mathbf{e}$$

of (1.1) generally is not a meaningful approximation of $\widehat{\mathbf{x}}$, because typically $\|A^\dagger \mathbf{e}\| \gg \|\widehat{\mathbf{x}}\|$. Here and throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm or the spectral matrix norm. To compute a useful approximation of $\widehat{\mathbf{x}}$, one often applies Tikhonov regularization, i.e., one replaces the problem (1.1) by the penalized least-squares problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \|A\mathbf{x}-\mathbf{b}\|^2 + \mu \|\mathbf{x}\|^2 \right\}, \tag{1.2}$$

where $\mu > 0$ is a regularization parameter. It is the purpose of the regularization term $\mu\|\mathbf{x}\|^2$ to damp the propagated error in the computed approximation of $\widehat{\mathbf{x}}$. However, it can be difficult to determine a suitable value of μ when no accurate bound for the norm of the error $\mathbf{e}$ in $\mathbf{b}$ is known. A too small value of μ gives a solution that is severely contaminated by the propagated error, and a too large value yields an unnecessarily poor approximation of $\widehat{\mathbf{x}}$; see, e.g., [13,15,20,26] for discussions on Tikhonov regularization and the choice of μ. For future reference, we note that the Tikhonov minimization problem (1.2) for any $\mu > 0$ has the unique solution

$$\mathbf{x}_\mu = (A^T A + \mu I)^{-1} A^T \mathbf{b}, \tag{1.3}$$

where the superscript ^{T} denotes transposition and I is the identity matrix.

One of the most popular methods for determining a suitable value of μ when no accurate bound for $\|\mathbf{e}\|$ is available is the Generalized Cross Validation (GCV) method; see [10,16,20,26,28]. This method chooses a regularization parameter that minimizes the *GCV function*

$$\mathscr{V}(\mu) := \frac{\|A\mathbf{x}_\mu - \mathbf{b}\|^2}{(\mathrm{trace}(I - A(\mu)))^2}, \tag{1.4}$$

where the *influence matrix* is defined by

$$A(\mu) := A(A^T A + \mu I)^{-1} A^T \tag{1.5}$$

and $\mathbf{x}_\mu$ is the Tikhonov solution (1.3). Thus, $A(\mu)\mathbf{b} = A\mathbf{x}_\mu$. Let $\mu^* > 0$ minimize (1.4). Numerical experiments indicate that the minimum of (1.4) generally is unique; in case of nonunicity, we let μ^* be the largest minimum. The GCV method prescribes that $\mathbf{x}_{\mu^*}$ be used as an approximate solution of (1.1). We remark that determining μ^* generally requires that the GCV function be evaluated for several μ-values.

When the matrix A is of small to moderate size, the evaluation of the singular value decomposition (SVD) of A is feasible. It is inexpensive to evaluate $\mathscr{V}(\mu)$ for different μ-values when the SVD of A is available. However, when A is large, the computation of its SVD is too expensive to be attractive. We describe how upper and lower bounds for the GCV function can be determined from a partial SVD that is made up of a few, say k, of the largest singular values of A and associated singular vectors. Computed examples in [29] illustrate that the computation of the $1 \leq k \ll n$ largest singular values and associated singular vectors of A is fairly inexpensive when A is the matrix of a linear discrete ill-posed problem; several examples taken from [21] show that the number of matrix-vector product evaluations with A and A^T required is only a small multiple of k. These computations apply an implicitly restarted Golub–Kahan bidiagonalization method; see, e.g., [3,4,25] for discussions on implicitly restarted bidiagonalization methods. Also other approaches to determining a few singular values and vectors of a large matrix, such as those described in [22,23,31] can be used. We remark that the evaluation of the matrix-vector products constitutes the dominating computational work.

The availability of a partial SVD of A makes it possible to compute an approximation of the Tikhonov solution (1.3), and allows the determination of upper and lower bounds for the numerator and denominator of the GCV function. These bounds yield useful bounds for the GCV function (1.4). We will use the upper bound of the GCV function, rather than the function itself, to determine a suitable value of the regularization parameter.

Several other methods for bounding or estimating the GCV function have been described in the literature. An approach to compute upper and lower bounds for $\mathscr{V}(\mu)$ based on the connection between standard and global Golub–Kahan bidiagonalization with Gauss quadrature is discussed in [14]. The method of the present paper is faster when the singular values of A decay to zero quickly. Golub and von Matt [19] estimate the denominator of (1.4) with the aid of Hutchinson's trace estimator [24] and bound the numerator by using the connection between Golub–Kahan bidiagonalization and Gauss quadrature; see also Bai et al. [5] for related methods. The computations described in [19] can be carried out efficiently for large-scale problems; however, Hutchinson's trace estimator is not reliable; see [14] for illustrations. Eldén [12] shows that it suffices to reduce A to bidiagonal form, instead of computing the SVD. Randomized algorithms for estimating the trace of a large matrix are surveyed by Avron and Toledo [1]. These algorithms typically require a large number of matrix-vector product evaluations to yield estimates of moderate to high accuracy. Novati and Russo [28] describe how carrying out a few steps of the Arnoldi process applied to A with initial vector $\mathbf{b}$ can give estimates of the GCV function (1.4). Approaches based on extrapolation and probing for approximating the trace of a large symmetric implicitly defined matrix are presented by Brezinski et al. [8,9], and Tang and Saad [32]. We remark that the method of the present paper may be attractive

to use for the applications discussed in [8,9,32] if the matrix whose trace is to be determined stems from a discrete ill-posed problem.

In some applications of Tikhonov regularization, the minimization problem (1.2) is replaced by a problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^n}\{\|A\mathbf{x}-\mathbf{b}\|^2+\mu\|L\mathbf{x}\|^2\} \tag{1.6}$$

with a matrix $L\in\mathbb{R}^{p\times n}$ different from the identity. Common choices of the matrix L include scaled discretizations of differential operators such as the tridiagonal matrix

$$L=\begin{bmatrix} -1 & 2 & -1 & & & \mathrm{O}\\ & 1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ \mathrm{O} & & & -1 & 2 & -1 \end{bmatrix}\in\mathbb{R}^{(n-2)\times n}.$$

The choice of an orthogonal projection $L\in\mathbb{R}^{n\times n}$ is discussed in [27]. Björck [6] and Eldén [11] describe how the minimization problem (1.6) can be brought into the form (1.2). We will therefore only consider Tikhonov regularization problems of the form (1.2) in the present paper.

This paper is organized as follows. Section 2 discusses how bounds for the denominator of the GCV function can be computed by using a partial SVD of the matrix A, and the computation of bounds for the numerator is described in Section 3. The evaluation of bounds for the GCV function is considered in Section 4, where also an algorithm is presented. The partial SVD of A allows the computation of an approximate solution of the Tikhonov minimization problem (1.2). Computed examples in Section 5 illustrate the performance of the proposed method and Section 6 contains concluding remarks.

2 Bounding the denominator of the GCV function

This section describes how upper and lower bounds for the denominator of the GCV function (1.4) can be computed with the aid of a partial SVD of the matrix A. Introduce the (full) SVD of A,

$$A=U\Sigma V^T, \tag{2.1}$$

where $U=[\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_m]\in\mathbb{R}^{m\times m}$ and $V=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n]\in\mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma=\mathrm{diag}[\sigma_1,\sigma_2,\ldots,\sigma_n]\in\mathbb{R}^{m\times n}$ is a (possibly rectangular) diagonal matrix with nonnegative diagonal entries. The diagonal entries are the singular values of A; they are ordered according to

$$\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_n\geq 0.$$

The columns $\mathbf{u}_j$ of U are commonly referred to as the left singular vectors of A, and the columns $\mathbf{v}_j$ of V as the right singular vectors. We refer to the singular triplets $\{\sigma_j,\mathbf{u}_j,\mathbf{v}_j\}_{j=1}^k$ associated with the k largest singular values as the k largest singular triplets of A. The bounds of the following proposition only require the k largest singular values to be known.

Proposition 2.1 *Let the matrix $A \in \mathbb{R}^{m\times n}$ have the singular value decomposition* (2.1) *and let $A(\mu)$ denote the influence matrix* (1.5)*. Then for $\mu > 0$ and $1 \le k \le n$, we have*

$$w_k - s_k \le \operatorname{trace}(I - A(\mu)) \le w_k, \tag{2.2}$$

where

$$w_k = m - \sum_{j=1}^{k} \frac{\sigma_j^2}{\sigma_j^2 + \mu}, \qquad s_k = (n-k)\frac{\sigma_k^2}{\sigma_k^2 + \mu}.$$

Proof Substituting (2.1) into (1.5) yields

$$A(\mu) = U\Sigma(\Sigma^T\Sigma + \mu I)^{-1}\Sigma^T U^T.$$

Therefore,

$$\operatorname{trace}(I - A(\mu)) = m - \sum_{j=1}^{n} \frac{\sigma_j^2}{\sigma_j^2 + \mu}. \tag{2.3}$$

This shows the right-hand side inequality of (2.2). To show the left-hand side inequality, we introduce the function

$$f(t) = \frac{t}{t+\mu}. \tag{2.4}$$

It is strictly increasing for $t \ge 0$ (and fixed $\mu > 0$). Therefore

$$\operatorname{trace}(A(\mu)) \le \sum_{j=1}^{k} \frac{\sigma_j^2}{\sigma_j^2 + \mu} + (n-k)\frac{\sigma_k^2}{\sigma_k^2 + \mu}. \tag{2.5}$$

This completes the proof.

The bound (2.5) can be sharpened by replacing σ_k by σ_{k+1} in the trailing term, and this would give a sharper lower bound than (2.2). We are interested in the bounds (2.2) because they can be evaluated when the first k singular values of A are known.

The accuracy of the approximation induced by the bounds (2.2) is proportional to the ratio $\sigma_k^2/(\sigma_k^2 + \mu)$. The following proposition investigates how this accuracy is influenced by the presence of errors in the singular values.

Proposition 2.2 *Let the computed singular values $\tilde{\sigma}_j$ be perturbed by errors, such that $|\tilde{\sigma}_j - \sigma_j| \le \tau$, $j = 1, 2, \ldots, n$. Let $\tilde{s}_k$ be the difference between the corresponding perturbed upper and lower bounds* (2.2)*. Then*

$$\tilde{s}_k \le s_k + \frac{(n-k)\tau}{\sigma_k^2 + \mu}. \tag{2.6}$$

Proof We can assume that the perturbed singular values $\tilde{\sigma}_j$ are the exact singular values of a perturbed matrix $A + E$, with $\|E\| \le \tau$. The stability theorem for the singular values [7] states that

$$|\tilde{\sigma}_j - \sigma_j| \le \|E\|,$$

which implies

$$\frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 + \mu} \le \frac{\sigma_k^2 + \|E\|^2}{\sigma_k^2 + \|E\|^2 + \mu} \le \frac{\sigma_k^2}{\sigma_k^2 + \mu} + \frac{\|E\|^2}{\sigma_k^2 + \mu}.$$

Inequality (2.6) follows immediately.

The above result shows that, when the singular values are perturbed, the accuracy of the bounds changes significantly only if the singular value σ_k, corresponding to the truncation parameter k, is comparable, or smaller, than the maximum absolute error τ in the singular values. This situation does not take place when the singular values decrease to zero quickly with increasing index number and there is a fairly large error $\mathbf{e}$ in the data $\mathbf{b}$.

Proposition 2.1 assumes that the k largest singular values of A are available. However, due to the fact that any algorithm used for their computation is terminated after finitely many steps and because of round-off errors introduced during the calculations, only approximations of these singular values are known. The following result complements Proposition 2.2 and is concerned with the sensitivity of the bounds of Proposition 2.1 to errors in the singular values.

Corollary 2.1 *Assume that not the singular values σ_j, $1 \leq j \leq k$, of A, but only lower and upper bounds σ_j^- and σ_j^+ for these singular values are known. Let these bounds be such that*

$$\sigma_j^- \leq \sigma_j \leq \sigma_j^+, \qquad 1 \leq j \leq k. \tag{2.7}$$

Then, for $\mu > 0$ and $1 \leq k \leq n$, we have

$$m - \sum_{j=1}^{k} \frac{(\sigma_j^+)^2}{(\sigma_j^+)^2+\mu} - (n-k)\frac{(\sigma_k^+)^2}{(\sigma_k^+)^2+\mu} \leq \mathrm{trace}(I - A(\mu)) \leq m - \sum_{j=1}^{k} \frac{(\sigma_j^-)^2}{(\sigma_j^-)^2+\mu}.$$

Proof The inequalities can be shown similarly as those of Proposition 2.1. For instance, using (2.3) and the fact that the function (2.4) is increasing, we obtain

$$\mathrm{trace}(I - A(\mu)) \leq m - \sum_{j=1}^{n} \frac{(\sigma_j^-)^2}{(\sigma_j^-)^2+\mu}.$$

The desired upper bound for $\mathrm{trace}(I - A(\mu))$ now follows. Similarly,

$$\mathrm{trace}(I - A(\mu)) \geq m - \sum_{j=1}^{n} \frac{(\sigma_j^+)^2}{(\sigma_j^+)^2+\mu},$$

which yields the lower bound for $\mathrm{trace}(I - A(\mu))$.

In applications of the bounds of the present paper, generally, all one knows about the matrix A are its $k \ll n$ largest singular triplets. It is therefore natural to replace A by its best rank-k approximation

$$A_k = U_k \Sigma_k V_k^T, \tag{2.8}$$

which is defined by these triplets. Thus, $U_k \in \mathbb{R}^{m\times k}$ is made up of the k first columns of the matrix U in (2.1), $V_k \in \mathbb{R}^{n\times k}$ is made up of the k first columns of V, and $\Sigma_k \in \mathbb{R}^{k\times k}$ is the leading principal submatrix of Σ of order k. The analogues of the bounds of Proposition 2.1 and Corollary 2.1 for A_k read as follows. Their proofs are similar to the proofs already shown and therefore are omitted.

Proposition 2.3 *Let the matrix $A_k \in \mathbb{R}^{m\times n}$ be defined by* (2.8) *and let $A_k(\mu)$ denote the associated influence matrix (obtained by replacing A by A_k in* (1.5)*). Then for $\mu > 0$, we have*

$$\operatorname{trace}(I - A_k(\mu)) = m - \sum_{j=1}^{k} \frac{\sigma_j^2}{\sigma_j^2 + \mu}.$$

Assume that the singular values σ_j of A are not known, but lower and upper bounds σ_j^- and σ_j^+ that satisfy (2.7) *are available. Then for $\mu > 0$, we have*

$$m - \sum_{j=1}^{k} \frac{(\sigma_j^+)^2}{(\sigma_j^+)^2 + \mu} \le \operatorname{trace}(I - A_k(\mu)) \le m - \sum_{j=1}^{k} \frac{(\sigma_j^-)^2}{(\sigma_j^-)^2 + \mu}.$$

It is interesting to observe that the upper bound for the trace is the same as in Corollary 2.1.

Remark 2.1 When the block Lanczos bidiagonalization algorithm (also known as the block Golub–Kahan bidiagonalization algorithm) [17] is applied to the computation of the k largest singular triplets, the following bounds for the singular values hold,

$$\sigma_j^- = \theta_j \qquad \text{and} \qquad \sigma_j^+ = \theta_j + \frac{\varepsilon_j^2}{\sigma_j} \le \theta_j + \frac{\varepsilon_j^2}{\theta_j},$$

where θ_j is the jth singular value of the bidiagonal matrix computed at the sth step of the algorithm; see [17, Theorem 3.2]. The parameter ε_j^2, which can be expressed in terms of the reciprocal of the square of a Chebyshev polynomial of the first kind evaluated outside the interval $[-1,1]$, usually is very small for the largest singular values. We therefore can expect the largest singular values of A to be computed with high accuracy; see [17, Example 3.2] and the end of Section 5 for illustrations.

3 Bounding the numerator of the GCV function

Let $\mathbf{x}_\mu$ be given by (1.3). This section derives bounds for $\|A\mathbf{x}_\mu - \mathbf{b}\|^2$ whose evaluation requires the first k singular values $\sigma_1, \sigma_2, \ldots, \sigma_k$ of A and associated left singular vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$. Substituting the SVD (2.1) of A into $\|A\mathbf{x}_\mu - \mathbf{b}\|^2$ and letting

$$\widetilde{\mathbf{b}} = [\widetilde{b}_1, \widetilde{b}_2, \ldots, \widetilde{b}_m]^T := U^T\mathbf{b}, \tag{3.1}$$

where the matrix U is from the SVD of A, yields

$$\|A\mathbf{x}_\mu - \mathbf{b}\|^2 = \mu^2 \widetilde{\mathbf{b}}^T (\Sigma\Sigma^T + \mu I)^{-2} \widetilde{\mathbf{b}} = \sum_{j=1}^{n} \frac{\mu^2 \widetilde{b}_j^2}{(\sigma_j^2 + \mu)^2} + \sum_{j=n+1}^{m} \widetilde{b}_j^2. \tag{3.2}$$

Proposition 3.1 *Let $\mu > 0$ and let $\mathbf{x}_\mu$ be the solution* (1.3) *of the Tikhonov minimization problem* (1.2)*. Then the following upper bound holds*

$$\|A\mathbf{x}_\mu - \mathbf{b}\|^2 \le u_k := \sum_{j=1}^{k} \frac{\mu^2 \widetilde{b}_j^2}{(\sigma_j^2 + \mu)^2} + c_k, \tag{3.3}$$

where

$$c_k := \sum_{j=k+1}^{m} \widetilde{b}_j^2 = \|\mathbf{b}\|^2 - \sum_{j=1}^{k} \widetilde{b}_j^2, \tag{3.4}$$

can be evaluated when the first k singular values $\sigma_1, \sigma_2, \ldots, \sigma_k$ *of A and the associated left singular vectors* $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ *are available.*

Proof The function

$$g(t) = (t+\mu)^{-2}, \qquad \mu > 0, \qquad t \geq 0, \tag{3.5}$$

is strictly decreasing for $0 \leq t < \infty$ and fixed $\mu > 0$. In particular, its maximum is $g(0) = 1/\mu^2$ and $\lim_{t\to\infty} g(t) = 0$. It follows that

$$0 \leq \frac{\mu^2 \widetilde{b}_j^2}{(\sigma_j^2+\mu)^2} \leq \widetilde{b}_j^2.$$

Substituting this inequality into (3.2) gives (3.3). The relation (3.4) is a consequence of $\|\widetilde{\mathbf{b}}\| = \|\mathbf{b}\|$.

We remark that when k in (3.3) is large enough so that σ_{k+1}^2 is small, the bound is quite sharp. The following lower bound generally is less sharp.

Proposition 3.2 *Let* $\mu > 0$ *and let* $\mathbf{x}_\mu$ *be the solution* (1.3) *of the Tikhonov minimization problem* (1.2)*. Then*

$$\|A\mathbf{x}_\mu - \mathbf{b}\|^2 \geq u_k - r_k, \tag{3.6}$$

where

$$r_k = \frac{\sigma_k^2(\sigma_k^2+2\mu)}{(\sigma_k^2+\mu)^2} c_k,$$

and u_k *and* c_k *are defined by* (3.3) *and* (3.4)*, respectively. The lower bound* (3.6) *can be evaluated by using only the first k singular triplets of A.*

Proof It follows from the properties of the function (3.5) that the right-hand side of (3.2) can be bounded below by

$$\|A\mathbf{x}_\mu - \mathbf{b}\|^2 \geq \sum_{j=1}^{k} \frac{\mu^2 \widetilde{b}_j^2}{(\sigma_j^2+\mu)^2} + \frac{\mu^2}{(\sigma_k^2+\mu)^2} \sum_{j=k+1}^{n} \widetilde{b}_j^2 + \sum_{j=n+1}^{m} \widetilde{b}_j^2. \tag{3.7}$$

Using the fact that $\mu^2/(\sigma_k^2+\mu)^2 \leq 1$, we obtain the lower bound (3.6) for the right-hand side of (3.7).

A looser but simpler bound can be obtained as follows.

Corollary 3.1 *Under the assumptions of the preceding proposition, it holds*

$$\|A\mathbf{x}_\mu - \mathbf{b}\|^2 \geq u_k - \frac{2\sigma_k^2}{\sigma_k^2+\mu} c_k.$$

Proof The result follows from

$$\frac{\sigma_k^2(\sigma_k^2+2\mu)}{(\sigma_k^2+\mu)^2} = \frac{\sigma_k^2}{\sigma_k^2+\mu}\left(1+\frac{\mu}{\sigma_k^2+\mu}\right) \leq \frac{2\sigma_k^2}{(\sigma_k^2+\mu)}.$$

Other bounds can be derived by relating k to the size of μ. The following proposition furnishes an example.

Proposition 3.3 *We use the notation of Proposition 3.2 and let $m = n$. Assume that k is chosen so that $\sigma_{k+1}^2 \leq \mu$. Then it holds*

$$\frac{1}{4}\sum_{j=k+1}^{n}\widetilde{b}_j^2 \leq \|A\mathbf{x}_\mu - \mathbf{b}\|^2 - \sum_{j=1}^{k}\frac{\mu^2\widetilde{b}_j^2}{(\sigma_j^2+\mu)^2} \leq \sum_{j=k+1}^{n}\widetilde{b}_j^2. \tag{3.8}$$

Proof The right-hand side inequality follows from (3.3). To show the left-hand side inequality, we note that $\sigma_j^2 \leq \mu$ implies

$$\frac{\mu^2\widetilde{b}_j^2}{(\sigma_j^2+\mu)^2} \geq \frac{\mu^2\widetilde{b}_j^2}{(\mu+\mu)^2} = \frac{1}{4}\widetilde{b}_j^2.$$

Therefore,

$$\sum_{j=k+1}^{n}\frac{\mu^2\widetilde{b}_j^2}{(\sigma_j^2+\mu)^2} \geq \frac{1}{4}\sum_{j=k+1}^{n}\widetilde{b}_j^2.$$

The left-hand side inequality of (3.8) now follows from (3.2).

When the singular values are perturbed by errors, the following result can be shown similarly as Proposition 2.2.

Proposition 3.4 *Let the computed singular values $\tilde{\sigma}_j$ be perturbed by errors, such that $|\tilde{\sigma}_j - \sigma_j| \leq \tau$, $j = 1,\ldots,n$. Let $\tilde{r}_k$ be the difference between the corresponding perturbed upper and lower bounds given in Proposition 3.1 and Corollary 3.1. Then*

$$\tilde{r}_k \leq \frac{2\sigma_k^2}{\sigma_k^2+\mu}c_k + \frac{2\tau}{\sigma_k^2+\mu}c_k.$$

We conclude this section by discussing some inequalities that are analogous to those of Propositions 3.1 and 3.2, and that can be applied when only approximations of the k first singular values of A and of the k first entries of the vector (3.1) are known. It is natural that bounds in terms of computed quantities only should involve these quantities. This suggests that the matrix A_k be used in the bounds, and that only part of the residual error be considered.

Corollary 3.2 *Let, for $1 \leq j \leq k$, the singular value bounds σ_j^- and σ_j^+ satisfy* (2.7)*, and let $|\widetilde{b}_j|^+$ and $|\widetilde{b}_j|^-$ be available upper and lower bounds for $|\widetilde{b}_j|$. Then*

$$\|A_k\mathbf{x}_{k,\mu} - U_kU_k^T\mathbf{b}\|^2 \leq \sum_{j=1}^{k}\frac{\mu^2(|\widetilde{b}_j|^+)^2}{((\sigma_j^-)^2+\mu)^2}, \tag{3.9}$$

where U_k is defined in (2.8) *and*

$$\mathbf{x}_{k,\mu} = V_k \Sigma_k (\Sigma_k^2 + \mu I)^{-1} \Sigma_k U_k^T \mathbf{b}.$$

Further,

$$\|A_k \mathbf{x}_{k,\mu} - U_k U_k^T \mathbf{b}\|^2 \geq \sum_{j=1}^{k} \frac{\mu^2 (|\widetilde{b}_j|^-)^2}{((\sigma_j^+)^2 + \mu)^2}, \tag{3.10}$$

Proof The upper bound (3.9) follows from (2.8) and the form of solution $\mathbf{x}_{k,\mu}$. The derivation of the bound assumes that the columns of the matrices U_k and V_k are orthonormal, but they are not required to be exact left and right singular vectors, respectively. The orthogonality requirement can be satisfied by reorthogonalization, which generally is used in the computation of partial singular value decompositions (2.8); see, e.g., [3,4]. When the orthogonal projector $U_k U_k^T$ is ignored in (3.9), the term $\|(I - U_k U_k^T)\mathbf{b}\|^2$ has to be added to the right-hand side. This term is independent of μ and therefore does not affect the location of the minimum (or minima) of the GCV function.

The lower bound (3.10) follows similarly. This bound is simpler than the corresponding bound (3.6) of Proposition 3.2.

As explained in the following section, the difference between available upper and lower bounds for the GCV function is used to determine how many singular triplets to compute. The location of the (a) minimum of the upper bound determines the value of the regularization parameter.

4 Bounding the GCV function

From the upper and lower bounds for the denominator and numerator of the GCV function derived in Sections 2 and 3, it is immediate to obtain bounds for the ratio. Let u_k and $\ell_k := u_k - r_k$ denote the upper and lower bounds, respectively, of the numerator of the GCV function of Propositions 3.1 and 3.2. Thus, we have

$$\ell_k \leq \|A\mathbf{x}_\mu - \mathbf{b}\|^2 \leq u_k.$$

Moreover, let

$$v_k \leq \operatorname{trace}(I - A(\mu)) \leq w_k,$$

where w_k and $v_k := w_k - s_k$ are the bounds of Proposition 2.1. Then,

$$\mathscr{L}_k \leq \mathscr{V}(\mu) \leq \mathscr{U}_k, \tag{4.1}$$

with $\mathscr{L}_k = \ell_k / w_k^2$ and $\mathscr{U}_k = u_k / v_k^2$. Note that these bounds depend on $\mu > 0$.

We remark that it is straightforward to use the bounds for the approximations of singular triplets described in Sections 2 and 3 instead of the bounds of Propositions 3.1 and 3.2. Therefore, we will only discuss the use of the latter bounds in this section.

In order to understand the behavior of the bounds (4.1), we consider the approximation errors r_k and s_k induced by the bounds for the numerator and the denominator;

see also Corollary 3.1. Both the errors depend on the ratio $\sigma_k^2/(\sigma_k^2+\mu)$. For a fixed tolerance τ, we have that

$$\frac{\sigma_k^2}{\sigma_k^2+\mu} < \tau \quad \text{implies} \quad \sigma_k^2 < \frac{\tau}{1-\tau}\mu.$$

Therefore, the bounds will be sharp for μ in some interval $[\mu_0, \mu_f]$ only for k such that $\sigma_k \ll \mu_0$. The interval will expand towards the left when k increases. This effect is observed in Figure 5.1 and 5.2 of Section 5.

In real-world applications, the error $\mathbf{e}$ in the data $\mathbf{b}$ is generally quite large, say 0.5% and larger. The minimum of the GCV function then is achieved at a not very small value of μ. If the singular values decay rapidly, as is the case for many discrete ill-posed problems, then we expect to obtain reliable bounds for $\mathscr{V}(\lambda)$ in a neighborhood of the minimum already for a fairly small value of k. This is illustrated in Section 5.

Let the singular triplets $\{\sigma_j, \mathbf{u}_j, \mathbf{v}_j\}_{j=1}^k$ be available. Then we evaluate the upper and lower bounds $\mathscr{U}_k$ and $\mathscr{L}_k$, respectively, for a discrete set of values of the regularization parameter μ. We store the elements of this set in the vector $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_p]^T$. The remainder of this section provides a discussion of our scheme for determining an approximation of the minimum of the GCV function. Details are described by Algorithm 1 below. In the algorithm, vectors are in bold, MATLAB notation is used for vector operations, and the square of a vector means squaring its components. The expressions in square brackets at line 19 produce the logical results *true* or *false*. The variable S, which is initialized at line 5 and updated at line 13, contains successive values of the quantity (3.4). Due to propagated round-off errors, this quantity may achieve a small negative value when many iterations are carried out. This situation is remedied at line 14, where the variable S is set to zero if it is found to be of magnitude smaller than a tiny positive constant eps.

We have observed in numerous numerical examples that the minimum of the upper bound quickly converges to the minimum of $\mathscr{V}(\mu)$, i.e., the minimum of the upper bound yields a quite accurate location of the minimum of the GCV function already for a fairly small number, k, of singular triplets. However, when $\mu > 0$ is tiny, the lower bound may be much smaller than the global minimum of the GCV function; see Figures 5.1 and 5.2 for illustrations. Algorithm 1 therefore estimates the location of the minimum of the GCV function from the upper bound; the lower bound is primarily used to determine whether the number of available singular triplets is large enough. Thus, at each step ℓ we identify the minimum μ_ℓ of the upper bound for $\mathscr{V}(\mu)$, and we stop the iteration if either the bounds converge near the minimum, i.e., if for a given tolerance $\tau > 0$,

$$\frac{\mathscr{U}_\ell - \mathscr{L}_\ell}{\mathscr{U}_\ell + \mathscr{L}_\ell} \leq \tau,$$

or if the minimum does not change much, that is, if

$$|\mu_\ell - \mu_{\ell-1}| \approx |\mu_{\ell-1} - \mu_{\ell-2}|. \tag{4.2}$$

To avoid that the iterations be terminated while the convergence is still erratic, we apply the condition (4.2) only after the difference $|\mu_\ell - \mu_{\ell-1}|$ has decreased for at least two subsequent iterations; see lines 19–21 of Algorithm 1.

The algorithm is initially applied to a vector λ with 12 components λ_k that are logarithmically spaced in $[10^{-10}, 10]$. In case the minimum is at one end point, we shift the interval and repeat the computation. If the minimum is at an internal point λ_k, we apply the algorithm to 100 values in $[\lambda_{k-1}, \lambda_{k+1}]$.

Algorithm 1 Bounding the GCV function by partial SVD.

1: **Input:** Singular triplets $(\sigma_j, \mathbf{u}_j, \mathbf{v}_j)$ of A, $\mathbf{u}_j \in \mathbb{R}^m$, $\mathbf{v}_j \in \mathbb{R}^n$, $j = 1\ldots, k$;
2: noisy right hand side $\mathbf{b}$; vector of Tikhonov parameters $\lambda \in \mathbb{R}^p$;
3: tolerances τ and η.
4: $s = \sigma_1^2$, $\beta = (\mathbf{u}_1^T \mathbf{b})^2$
5: $\mathbf{t} = \lambda^2./(s+\lambda)^2$, $\mathbf{w} = \beta\mathbf{t}$, $S = \|\mathbf{b}\|^2 - \beta$
6: $\mathbf{U}_1 = \mathbf{w} + S$, $\mathbf{L}_1 = \mathbf{w} + S\mathbf{t}$
7: $\mathbf{t} = s./(s+\lambda)$, $\mathbf{U}_2 = m - \mathbf{t}$, $\mathbf{L}_2 = \mathbf{U}_2 - (n-1)\mathbf{t}$
8: $\mathscr{U} = \mathbf{U}_1/\mathbf{L}_2^2$, $\mathscr{L} = \mathbf{L}_1/\mathbf{U}_2^2$
9: $\omega_\ell = \min(\mathscr{U})$, $\mu = \lambda_\ell$
10: $j = 1$, $\varepsilon = 0$, $c = 0$, flag = true
11: **while** flag
12: $j = j + 1$, $\widetilde{\mu} = \mu$, $\widetilde{\varepsilon} = \varepsilon$, $s = \sigma_j^2$, $\beta = (\mathbf{u}_j^T \mathbf{b})^2$
13: $\mathbf{t} = \lambda^2./(s+\lambda)^2$, $\mathbf{w} = \mathbf{w} + \beta\mathbf{t}$, $S = S - \beta$
14: **if** $|S| <$ eps **then** $S = 0$ **end**
15: $\mathbf{U}_1 = \mathbf{w} + S$, $\mathbf{L}_1 = \mathbf{w} + S\mathbf{t}$
16: $\mathbf{t} = s./(s+\lambda)$, $\mathbf{U}_2 = \mathbf{U}_2 - \mathbf{t}$, $\mathbf{L}_2 = \mathbf{U}_2 - (n-j)\mathbf{t}$
17: $\mathscr{U} = \mathbf{U}_1/\mathbf{L}_2^2$, $\mathscr{L} = \mathbf{L}_1/\mathbf{U}_2^2$
18: $\omega_\ell = \min(\mathscr{U})$, $\mu = \lambda_\ell$, $\varepsilon = |\mu - \widetilde{\mu}|$
19: flag = $[(j < k)$ **and** $((\mathscr{U}_\ell - \mathscr{L}_\ell)/(\mathscr{U}_\ell + \mathscr{L}_\ell) > \tau)]$
20: **if** $(\varepsilon < \widetilde{\varepsilon})$ **then** $c = c + 1$ **else** $c = 0$ **end**
21: **if** $(c \geq 2)$ **then** flag = [flag **and** $(|\varepsilon - \widetilde{\varepsilon}| > \eta)$] **end**
22: **end while**
23: **Output:** Tikhonov parameter μ, lower and upper bounds $\mathscr{L}_\ell$, $\mathscr{U}_\ell$ for $\mathscr{V}(\mu)$.

Typically, the number of singular triplets of the matrix A in (1.1) required to satisfy the stopping criterion of Algorithm 1 is not known before application of the algorithm. Therefore, when applying the algorithm to large-scale problems, it has to be combined with a suitable procedure for computing the required singular triplets. To this end, we adopted the same computational scheme that was applied in [2] in the context of complex network analysis. We first compute the p largest singular triplets $\{\sigma_j, \mathbf{u}_j, \mathbf{v}_j\}_{j=1}^p$ of A by the augmented implicitly restarted Golub–Kahan bidiagonalization method described in [3]. If more singular triplets are required to satisfy the stopping criterion of Algorithm 1, we use a slight modification of the MATLAB function from [3], which allows the computation to be restarted, to determine a new batch of triplets $\{\sigma_j, \mathbf{u}_j, \mathbf{v}_j\}_{j=p+1}^{2p}$. We repeat this process as many times as necessary to satisfy the stopping criterion.

We found in numerous numerical experiments that when k is the index that satisfies the stopping criterion of Algorithm 1, the difference $\|\mathbf{x}_\mu - \mathbf{x}_{k,\mu}\|$ between the Tikhonov solution $\mathbf{x}_\mu$ and the corresponding truncated SVD approximation $\mathbf{x}_{k,\mu}$ is

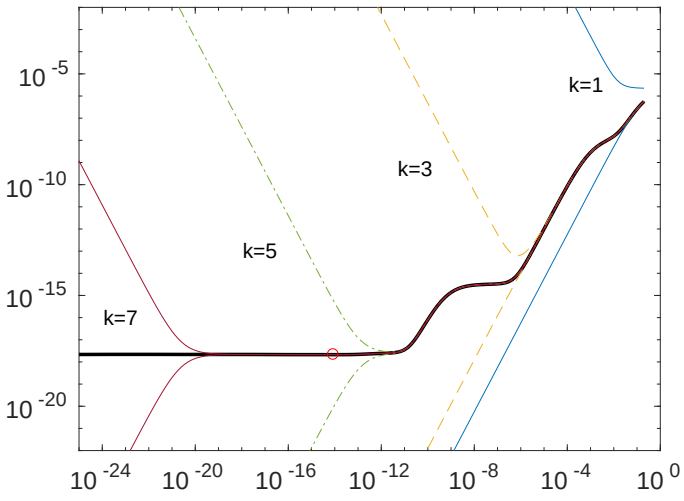

Fig. 5.1 Upper and lower bounds for the Wing test problem of size 100×100 with noise level 10^{-6}, as functions of μ. The thick line represents the GCV function $\mathscr{V}(\mu)$, while the thin lines are the computed bounds $\mathscr{U}_k$ and $\mathscr{L}_k$, $k = 1,3,5,7$.

negligible compared with the error in the Tikhonov solution $\|\widehat{\mathbf{x}} - \mathbf{x}_\mu\|$, where $\widehat{\mathbf{x}} = A^\dagger \widehat{\mathbf{b}}$ denotes the desired solution of the error-free least-squares problem associated with (1.1).

5 Computed examples

We implemented Algorithm 1 in MATLAB to illustrate the effectiveness of the bounds computed by the algorithm for estimating the minimum of the GCV function and in this manner determining a suitable value of the parameter for Tikhonov regularization. All computations were carried out with about 16 significant decimal digits on an Intel Core i7/860 computer with 8Gb RAM running Linux. The algorithm was applied to the discrete ill-posed problems listed in Table 5.1, some of which are from Hansen's Regularization Tools [21] and some from the MATLAB `gallery` function. While the problems from [21] come with a model solution $\widehat{\mathbf{x}}$, the ones from the `gallery` function do not. We associated with the latter the exact solution $\widehat{\mathbf{x}}_{\text{Baart}}$ of the Baart test problem from Regularization Tools and determined the exact right-hand side according to $\widehat{\mathbf{b}} = A\widehat{\mathbf{x}}_{\text{Baart}}$. For all test problems, we generate an error-contaminated right-hand side $\mathbf{b}$ according to

$$\mathbf{b} = \widehat{\mathbf{b}} + \frac{\delta \|\widehat{\mathbf{b}}\|}{\sqrt{m}} \mathbf{n},$$

where $\mathbf{n} \in \mathbb{R}^m$ is a random white noise vector, with zero mean and variance one, and δ is the noise level.

We first consider some small-scale examples, for which we can compute the (full) singular value decomposition by the `svd` function of MATLAB. Figure 5.1 displays the bounds obtained from (4.1), as functions of μ, for the linear discrete ill-posed problem (1.1) Wing from [21] with $A \in \mathbb{R}^{100\times 100}$ and the small noise level $\delta = 10^{-6}$. Algorithm 1 carries out 7 iterations and correctly determines the location of the minimum of $\mathscr{V}(\mu)$, displayed by a red circle in the graph. The upper and lower bounds $\mathscr{U}_k$ and $\mathscr{L}_k$ approximate the GCV function $\mathscr{V}(\mu)$ well over larger μ-intervals the larger k is.

Figure 5.2 is analogous to Figure 5.1 and shows the upper and lower bounds from (4.1) for a linear discrete ill-posed problem (1.1) with a rectangular matrix A. Specifically, A is defined by the first 40 columns of a Shaw matrix of order 60. The latter is generated by software in [21]. The noise level in this example is $\delta = 10^{-2}$; this noise level is representative for many linear discrete ill-posed problems that have to be solved in science and engineering. Algorithm 1 carried out 13 steps for the present example before satisfying the stopping criterion. The computed minimum gives a value of the the regularization parameter that is very close to the best possible.

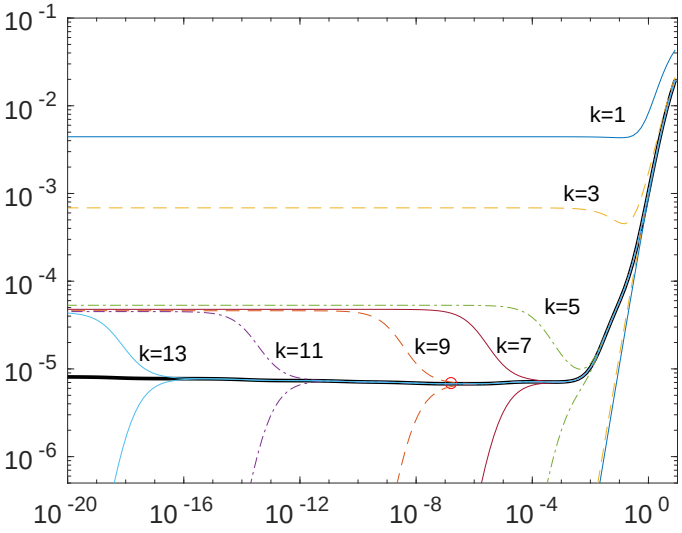

Fig. 5.2 Upper and lower bounds for the Shaw test problem of size 60×40 with noise level 10^{-2}, as functions of μ. The thick line represents the GCV function $\mathscr{V}(\mu)$, while the thin lines are the computed bounds $\mathscr{U}_k$ and $\mathscr{L}_k$, $k = 1,3,\ldots,13$.

To investigate the accuracy of the bounds when determining a suitable value of the Tikhonov regularization parameter, we constructed for each test problem a set of 60 least squares problems, both square (200×200) and rectangular (400×200) ones, by letting the noise level take on the values 10^{-4}, 10^{-3}, and 10^{-2}. For each problem and noise level, we generated 10 random noise vectors $\mathbf{e}$ and determined for each one of the 60 numerical examples so defined the regularization parameter μ by Algorithm 1 as well as the regularization parameter μ_{best} that gives the smallest

relative error in the Euclidean norm, i.e.,

$$E_{\text{opt}} := \frac{\|\mathbf{x}_{\mu_{\text{best}}} - \widehat{\mathbf{x}}\|}{\|\widehat{\mathbf{x}}\|} = \min_{\mu>0} \frac{\|\mathbf{x}_\mu - \widehat{\mathbf{x}}\|}{\|\widehat{\mathbf{x}}\|}, \tag{5.1}$$

where $\widehat{\mathbf{x}}$ is the desired solution of the error-free problem associated with (1.1). Let F_κ denote the number of experiments for which the computed regularized solution $\mathbf{x}_\mu$ with μ determined by Algorithm 1 yields a relative error larger than κ times the optimal error, that is, the number of times the following inequality holds

$$\frac{\|\mathbf{x}_\mu - \widehat{\mathbf{x}}\|}{\|\mathbf{x}\|} > \kappa \frac{\|\mathbf{x}_{\mu_{\text{best}}} - \widehat{\mathbf{x}}\|}{\|\mathbf{x}\|}. \tag{5.2}$$

Table 5.1 shows the results obtained for solutions determined by Algorithm 1, denoted by `gcvpsvd`, and compares them to results produced by the `gcv` function from [21]. We report, for each group of 60 test problems, the average E_{opt} of the optimal error (5.1) and the values of F_κ, $\kappa = 5, 10$, for the two methods. In the table, E_κ denotes the average of the relative errors that do not exceed the limit imposed by (5.2). The results show Algorithm 1 to perform essentially as well as the standard approach for minimizing $\mathscr{V}(\mu)$ as implemented by the `gcv` function from [21]. The slightly better performance of Algorithm 1 is probably due to the fact that the GCV function for some problems is extremely flat near the minimum, and this may lead to inaccuracy in the computation of the minimum. The table indicates that the minimization of the upper bound of the GCV function, as implemented by Algorithm 1, performs somewhat better than the minimization of the GCV function as implemented by the `gcv` function from [21].

Table 5.1 Comparison of two GCV minimization algorithms across the solution of 10 test problems, each composed by 60 numerical examples obtained by varying the size of the problem, the noise level, and by considering different realizations of the noise; see text.

		`gcv`				`gcvpsvd`			
matrix	E_{opt}	F_5	E_5	F_{10}	E_{10}	F_5	E_5	F_{10}	E_{10}
Baart	1.8e-01	14	5.9e-01	14	5.9e-01	4	5.4e-01	4	5.4e-01
Deriv2(2)	2.6e-01	1	5.6e-01	0	1.8e+00	0	2.7e-01	0	2.7e-01
Foxgood	5.7e-02	12	1.2e-01	3	4.5e-01	9	1.2e-01	2	4.4e-01
Gravity	3.7e-02	16	4.7e-02	13	1.9e-01	14	4.7e-02	9	2.4e-01
Heat(1)	1.3e-01	0	3.7e-01	0	3.7e-01	0	1.7e-01	0	1.7e-01
Hilbert	4.5e-01	4	6.4e-01	4	6.4e-01	2	6.4e-01	0	6.4e-01
Lotkin	4.6e-01	4	1.2e+00	2	1.2e+00	3	7.1e-01	1	7.1e-01
Phillips	3.7e-02	4	1.3e-01	2	1.3e-01	0	8.4e-02	0	8.4e-02
Shaw	1.3e-01	18	1.2e-01	14	3.5e-01	6	1.9e-01	5	1.9e-01
Wing	6.1e-01	12	1.6e+00	10	4.1e+00	6	1.6e+00	4	4.4e+00

We now illustrate the performance of Algorithm 1 when applied to a few examples of larger size. We construct a partial singular value decomposition, as described at the end of Section 4, computing the singular triplets in batches of 10, and compare our algorithm to two other methods for approximating the GCV function designed

for large scale problems, namely, the Quadrature method, described in [14], and the routine `gcv_lanczos` from [19]. The former approach combines the global Lanczos method with techniques described in [18] to determine upper and lower bounds for the GCV function; the latter determines an estimate of the trace in the denominator of (1.4) with the aid of Hutchinson's trace estimator. Computed examples reported in [14] show the routine `gcv_lanczos` to be fast, but the trace estimates determined not to be reliable. This is confirmed by examples below.

The first example is the Shaw test problem from [21] of size 2048, with noise level $\delta = 10^{-2}$. The solutions produced by the three methods are depicted in the left-hand side graph of Figure 5.3, while Table 5.2 reports the relevant parameters for this and related experiments: the value of the Tikhonov parameter μ, the relative error with respect to the exact solution E_r, and the computing time in seconds. For our method, labeled `gcvpsvd`, we also show the value of the truncation parameter k used for the bounds, that is the value of j in Algorithm 1 for which the stopping criterion is satisfied. The three methods are roughly equivalent in terms of accuracy. The Quadrature method is rather slow, compared to the other methods, as already remarked in [14], but gives exact bounds for the GCV function. The new approach shares the latter property, but it is much faster, as the value of k is small. We also note that its implementation is simpler than the implementation of the other methods in our comparison.

Table 5.2 Parameters which characterize the numerical experiments of Figures 5.3, 5.4, and 5.5.

	(m,σ)	Shaw $(2048, 10^{-2})$	Baart $(1024, 10^{-1})$	Hilbert $(65536, 10^{-4})$	AlgDec $(65536, 10^{-2})$
`gcvpsvd`	μ	$6.8 \cdot 10^{-3}$	$3.7 \cdot 10^{-2}$	$1.2 \cdot 10^{-5}$	$1.5 \cdot 10^{-2}$
	E_r	$6.1 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	$6.3 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$
	time	0.29	0.14	5	372
	k	14	7	25	300
`Quadrature`	μ	$6.1 \cdot 10^{-2}$	$3.7 \cdot 10^{-1}$	-	-
	E_r	$1.3 \cdot 10^{-1}$	$2.2 \cdot 10^{-1}$	-	-
	time	1.6	2.3	-	-
`gcv_lanczos`	μ	$3.0 \cdot 10^{-3}$	$3.0 \cdot 10^{-6}$	$8.3 \cdot 10^{-7}$	$2.5 \cdot 10^{-3}$
	E_r	$6.1 \cdot 10^{-2}$	$3.3 \cdot 10^{1}$	$2.2 \cdot 10^{-1}$	$3.5 \cdot 10^{-2}$
	time	0.32	0.17	22	3

The graph on the right-hand side of Figure 5.3 displays the results of a particular noise realization for the Baart test problem from [21] of order 1024 and with $\delta = 10^{-1}$. While for many runs the three methods behave similarly, in a significant number of experiments, such as this one, the `gcv_lanczos` routine underestimates the regularization parameter μ and yields a computed approximate solution with a large propagated error. The Quadrature method and Algorithm 1 yield about the same regularization parameter values, but the latter is more than 10 times faster, since only 7 singular triplets are required to compute the bounds.

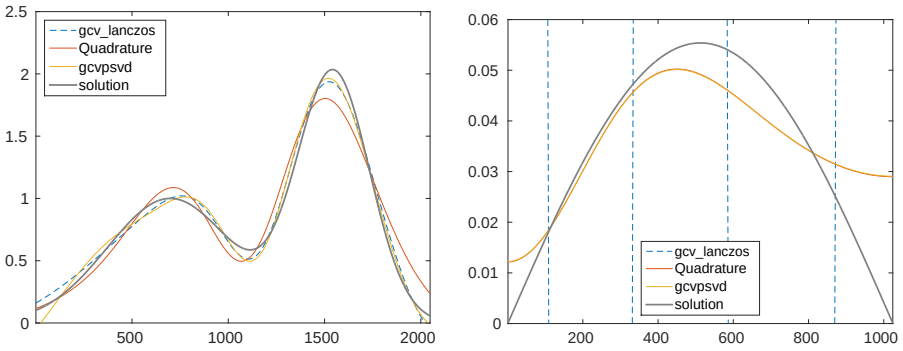

Fig. 5.3 Solution of Shaw test problem (left-hand side graph, $m = n = 2048$, $\delta = 10^{-2}$) and of the Baart example (right-hand side graph, $m = n = 1024$, $\delta = 10^{-1}$) by the three methods considered. The near-vertical lines depict parts of the solution computed by the `gcv_lanczos` method. This solution is underregularized and oscillates with large amplitude. Therefore only parts of it are visible.

We finally consider two large-scale linear discrete ill-posed problems. Figure 5.4 displays results for a Hilbert matrix $A \in \mathbb{R}^{65536\times 65536}$. We use the noise level $\delta = 10^{-4}$. The Hilbert matrix is a well known Hankel matrix. We generate A by the `smt` library [30], which provides fast matrix-vector multiplication routines and compressed storage for circulant and Toeplitz matrices. An extension to this library, available at the web page of the package, implements an additional class for Hankel matrices. The model solution is the one of the Baart test problem. We compare Algorithm 1 to the `gcv_lanczos` method. The two computed solutions are fairly satisfactory, but the one produced by our algorithm is clearly more accurate; see Table 5.2 and the left-hand side graph of Figure 5.4. The `gcvpsvd` routine is particularly fast in this example, because of the rapid decay of the singular values of the Hilbert matrix. This is illustrated by the right-hand side graph of Figure 5.4. The method of the present paper is faster in this and the following examples than the Quadrature method described in [14]. We therefore do not compare with the latter method in Table 5.2.

The last linear system of equations that we discuss has a Toeplitz matrix $A = [a_{ij}]$ of order $n = 65536$, whose elements are defined by

$$a_{ij} = \frac{2\pi}{\sigma(4\sigma^2 + (i-j)^2)}.$$

This test matrix is denoted AlgDec in [30]. We let $\sigma = 10$. It is shown in [33] that the asymptotic condition number, as $n \to \infty$, is about 10^{27}. We consider the model solution from the Shaw problem and use the noise level $\delta = 10^{-2}$. This defines the vector $\mathbf{b}$ in (1.1). The parameters of this experiment are reported in Table 5.2, while Figure 5.5 displays the solution, a plot of the GCV function, and the decay of the singular values of A. As in the previous example, Algorithm 1 is compared to the `gcv_lanczos` method. Table 5.2 shows this example to be particularly difficult for Algorithm 1. The algorithm reached the last iteration allowed ($k = 300$) without satisfying the convergence criterion. The long computation time is due to the very slow decay of the singular values of the Gaussian matrix, but the solution obtained by the `gcvpsvd` routine is much smoother than the one produced by `gcv_lanczos`.

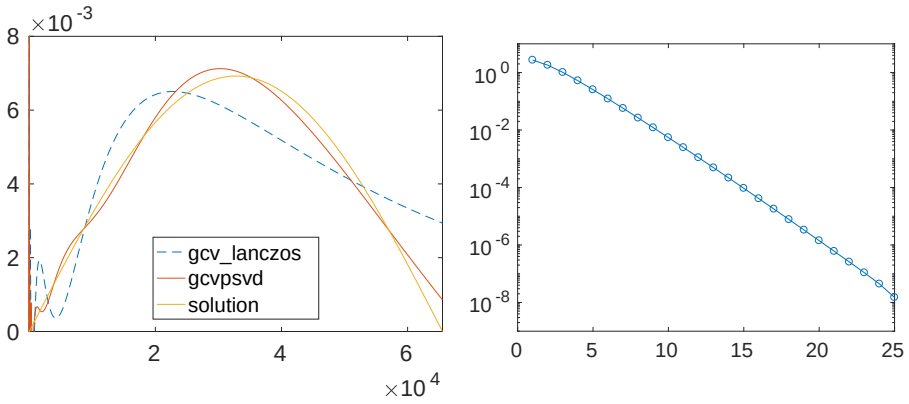

Fig. 5.4 Solution of the Hilbert test problem (left-hand side graph) with $m = n = 65536$ and $\delta = 10^{-4}$. The graph on the right-hand side shows the values of the first 25 singular values of the Hilbert matrix, computed during the execution of the `gcvpsvd` routine.

To conclude this section, we illustrate the behavior of the parameters ε_j^2 of Remark 2.1. These parameters bound the difference between the (exact) singular values of the matrix A and their approximations computed with the block Golub–Kahan algorithm [17]. Figure 5.6 shows the semi-logarithmic plot of the parameters ε_j^2, $j = 1,3,5,7,9$, as a function of the number of steps of the algorithm. We used block size 10. The left-hand side plot of Figure 5.6 displays the values of the ε_j^2 for the Shaw matrix of order 100, and the right-hand side plot shows the same quantities for the AlgDec matrix of the same size. It is worth noting that already a small number of steps yields very accurate approximations of the singular values. This is due to the fact that, in the special case of discrete ill-posed problems, the values of ε_j^2 are determined by the reciprocal of the square of a Chebyshev polynomial of the first kind evaluated at points very far away from the interval $[-1,1]$; see [17] for the definition of the ε_j^2.

6 Conclusion

The exact evaluation of the GCV function for large-scale problems can be expensive. To circumvent this difficulty Golub and von Matt described a method based on the stochastic trace estimator by Hutchinson. Recently an approach to compute bounds for the GCV function has been proposed in [14]. While the former method is fast, it is not reliable. The latter method is cheaper than computing the full SVD, but it may nevertheless be expensive to apply to some problems. This paper describes a novel approach for bounding the GCV function; it is based on the observation that for many linear discrete ill-posed problems with a large matrix, it is quite inexpensive to compute a few of the largest singular triplets. This allows us to compute upper and lower bounds for the GCV function. A suitable value of the regularization parameter is determined by minimizing the upper bound. For small problems, this approach is found to work at least as well or better than minimizing the (exact) GCV function. The

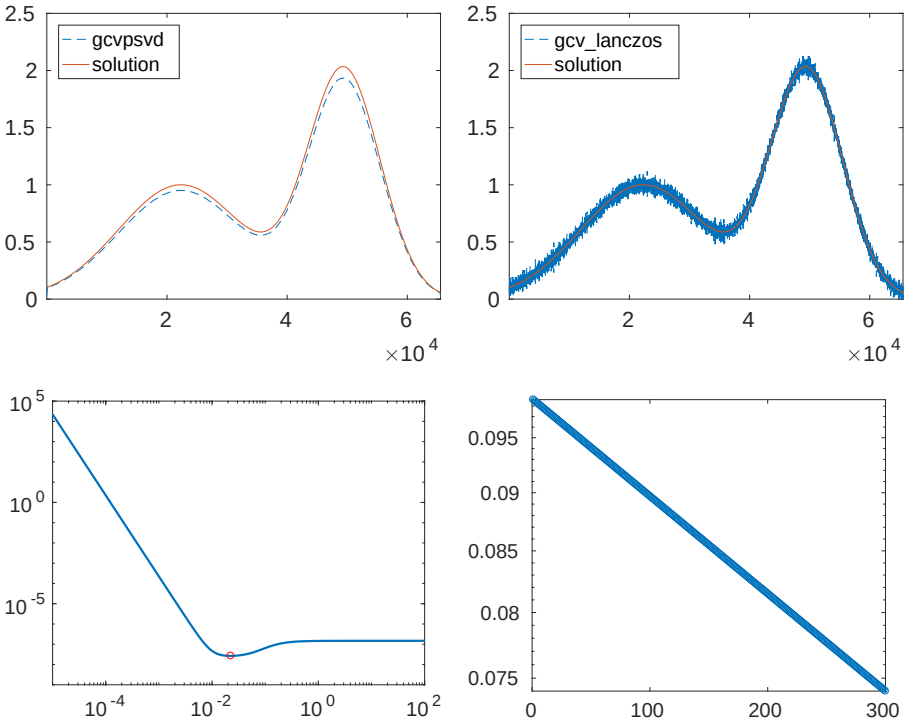

Fig. 5.5 Solution of the AlgDec test problem (graphs in the top row) with $m = n = 65535$ and $\delta = 10^{-2}$. The graph on the left-hand side in the bottom row shows the GCV function $\mathscr{V}(\mu)$ computed by Algorithm 1; the graph on the right-hand side shows the first 300 singular values of the AlgDec matrix, computed during the execution of the `gcvpsvd` routine.

proposed method performs particularly well when applied to the solution of linear discrete ill-posed problems with a matrix, whose singular values decay to zero fairly rapidly with increasing index number.

Acknowledgements The authors would like to thank Michiel Hochstenbach and a referee for comments. Work of C.F and G.R. was partially supported by INdAM-GNCS.

References

1. H. Avron and S. Toledo, *Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix*, J. ACM, 58, pp. 8:1–8:34 (2011)
2. J. Baglama, C. Fenu, L. Reichel, and G. Rodriguez, *Analysis of directed networks via partial singular value decomposition and Gauss quadrature*, Linear Algebra Appl., 456, pp. 93–121 (2014)
3. J. Baglama and L. Reichel, *Restarted block Lanczos bidiagonalization methods*, Numer. Algorithms, 43, pp. 251–272 (2006)
4. J. Baglama and L. Reichel, *An implicitly restarted block Lanczos bidiagonalization method using Leja shifts*, BIT, 53, pp. 285–310 (2013)
5. Z. Bai, M. Fahey, and G. Golub, *Some large-scale matrix computation problems*, J. Comput. Appl. Math., 74, pp. 71–89 (1996)

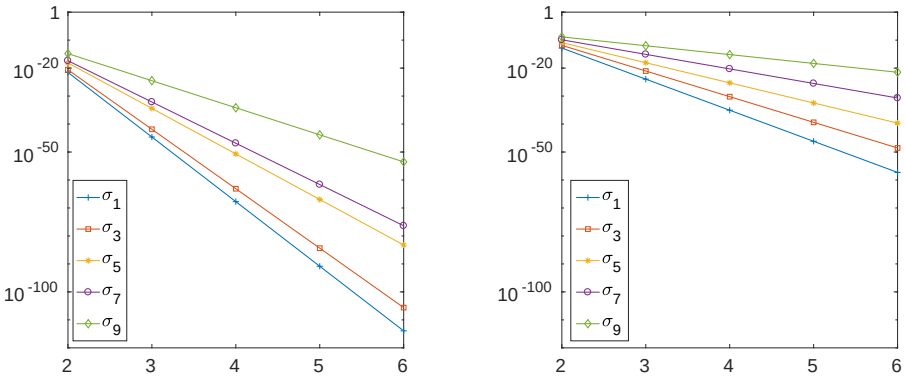

Fig. 5.6 Semi-logarithmic plot of the parameter ε_j^2, $j = 1,3,5,7,9$, vs. the number of steps of the Block Golub-Kahan algorithm. The left picture displays the values of the parameter for a Shaw matrix and the right picture for the AlgDec matrix both of size 100.

6. Å. BJÖRCK, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 18, pp. 659–670 (1988)
7. Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia (1996)
8. C. BREZINSKI, P. FIKA, AND M. MITROULI, *Estimations of the trace of powers of self-adjoint operators by extrapolation of the moments*, Electron. Trans. Numer. Anal., 39, pp. 144–159 (2012)
9. C. BREZINSKI, P. FIKA, AND M. MITROULI, *Moments of a linear operator on a Hilbert space, with applications to the trace of the inverse of matrices and the solution of equations*, Numer. Linear Algebra Appl., 19, pp. 937–953 (2012)
10. P. CRAVEN AND G. WAHBA, *Smoothing noisy data with spline functions*, Numer. Math., 31, pp. 377–403 (1979)
11. L. ELDÉN, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22, pp. 487–501 (1982)
12. L. ELDÉN, *A note on the computation of the generalized cross-validation function for ill-conditioned least squares problems*, BIT, 24, pp. 467–472 (1984)
13. H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, Dordrecht (1996)
14. C. FENU, L. REICHEL, AND G. RODRIGUEZ, *GCV for Tikhonov regularization via global Golub–Kahan decomposition*, Numer. Linear Algebra Appl., 23, pp. 467–484 (2016)
15. S. GAZZOLA, P. NOVATI, AND M. R. RUSSO, *On Krylov projection methods and Tikhonov regularization*, Electron. Trans. Numer. Anal., 44, pp. 83–123 (2015)
16. G. H. GOLUB, M. HEATH, AND G. WAHBA, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21, pp. 215–223 (1979)
17. G. H. GOLUB, F. T. LUK, AND M. L. OVERTON, *A block Lanczos method for computing the singular values and corresponding singular Vectors of a matrix*, ACM Trans. Math. Software, 7, pp. 149–169 (1981)
18. G. H. GOLUB AND G. MEURANT, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton (2010)
19. G. H. GOLUB AND U. VON MATT, *Generalized cross-validation for large-scale problems*, J. Comput. Graph. Stat., 6, pp. 1–34 (1997)
20. P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia (1998)
21. P. C. HANSEN, *Regularization tools version 4.0 for MATLAB 7.3*, Numer. Algorithms, 46, pp. 189–194 (2007)
22. M. E. HOCHSTENBACH, *A Jacobi–Davidson type SVD method*, SIAM J. Sci. Comput., 23, pp. 606–628 (2001)
23. M. E. HOCHSTENBACH, *Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems*, BIT, 44, pp. 721–754 (2004)

24. M. F. HUTCHINSON, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Statist. Simula., 18, pp. 1059–1076 (1989)
25. Z. JIA AND D. NIU, *An implicitly restarted refined bidiagonalization Lanczos method for computing a partial singular value decomposition*, SIAM J. Matrix Anal. Appl., 25, pp. 246–265 (2003)
26. S. KINDERMANN, *Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems*, Electron. Trans. Numer. Anal., 38, pp. 233–257 (2011)
27. S. MORIGI, L. REICHEL, AND F. SGALLARI, *Orthogonal projection regularization operators*, Numer. Algorithms, 44, pp. 99–114 (2007)
28. P. NOVATI AND M. R. RUSSO, *A GCV based Arnoldi–Tikhonov regularization method*, BIT, 54, pp. 501–521 (2014)
29. E. ONUNWOR AND L. REICHEL, *On the computation of a truncated SVD of a large linear discrete ill-posed problem*, Numer. Algorithms, in press.
30. M. REDIVO–ZAGLIA AND G. RODRIGUEZ, *smt: a Matlab toolbox for structured matrices*, Numer. Algorithms, 59, pp. 639–659 (2012)
31. M. Stoll, A Krylov–Schur approach to the truncated SVD, Linear Algebra Appl., 436, pp. 2795–2806 (2012)
32. J. TANG AND Y. SAAD, *A probing method for computing the diagonal of the matrix inverse*, Numer. Linear Algebra Appl., 19, pp. 485–501 (2012)
33. C. V. M. VAN DER MEE AND S. SEATZU, *A method for generating infinite positive self-adjoint test matrices and Riesz bases*, SIAM J. Matrix Anal. Appl., 26, pp. 1132–1149 (2005)