Applied Numerical Mathematics 164 (2021) 72–88

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A novel modified TRSVD method for large-scale linear discrete ill-posed problems

Xianglan Bai [a], Guang-Xin Huang [b], Xiao-Jun Lei [c], Lothar Reichel [a,*], Feng Yin [d]

[a] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA
[b] Geomathematics Key Laboratory of Sichuan, Chengdu University of Technology, Chengdu, 610059, PR China
[c] Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Graduate School of China Academy of Engineering Physics, Beijing, 100088, PR China
[d] College of Mathematical and Statistics, Sichuan University of Science and Engineering, Zigong, 643000, PR China

ARTICLE INFO

Article history:
Received 1 June 2020
Received in revised form 28 August 2020
Accepted 31 August 2020
Available online 11 September 2020

Keywords:
Linear discrete ill-posed problems
TSVD
Modified TSVD
TRSVD

ABSTRACT

The truncated singular value decomposition (TSVD) is a popular method for solving linear discrete ill-posed problems with a small to moderately sized matrix A. This method replaces the matrix A by the closest matrix A_k of low rank k, and then computes the minimal norm solution of the linear system of equations with a rank-deficient matrix so obtained. The modified TSVD (MTSVD) method improves the TSVD method, by replacing A by a matrix that is closer to A than A_k in a unitarily invariant matrix norm and has the same spectral condition number as A_k. Approximations of the SVD of a large matrix A can be computed quite efficiently by using a randomized SVD (RSVD) method. This paper presents a novel modified truncated randomized singular value decomposition (MTRSVD) method for computing approximate solutions to large-scale linear discrete ill-posed problems. The rank, k, is determined with the aid of the discrepancy principle, but other techniques for selecting a suitable rank also can be used. Numerical examples illustrate the effectiveness of the proposed method and compare it to the truncated RSVD method.

1. Introduction

This paper is concerned with the computation of approximate solutions of large minimization problems of the form

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2, \tag{1.1}$$

where $A\in\mathbb{R}^{m\times n}$ is a large matrix whose singular values gradually decay to zero without a significant gap. In particular, A is severely ill-conditioned and may be rank-deficient. Minimization problems (1.1) with a matrix of this kind often are referred to as discrete ill-posed problems. They arise, for example, from the discretization of linear ill-posed problems, such as Fredholm integral equations of the first kind with a smooth kernel. Throughout this paper $\|\cdot\|_2$ denotes the Euclidean vector norm or the spectral matrix norm. We allow $m\geq n$ as well as $m<n$.

* Corresponding author.
E-mail addresses: xbai@kent.edu (X. Bai), huangx@cdut.edu.cn (G.-X. Huang), 3292694852@qq.com (X.-J. Lei), reichel@math.kent.edu (L. Reichel), fyin@suse.edu.cn (F. Yin).

https://doi.org/10.1016/j.apnum.2020.08.019
0168-9274/

The vector $b \in \mathbb{R}^m$ in (1.1) represents measured data that are contaminated by an error $e \in \mathbb{R}^m$, which may stem from measurement or discretization errors. Let $\widehat{b} \in \mathbb{R}^m$ denote the unknown error-free vector associated with b, i.e.,

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

We will assume $\widehat{b}$ to be in the range of A, and that a fairly accurate bound for the relative error

$$\frac{\|e\|_2}{\|\widehat{b}\|_2} \leq \epsilon, \tag{1.3}$$

in b is known. Then a regularized solution of (1.1) can be determined with the aid of the discrepancy principle; see below.

We are interested in computing an approximation of the solution $\widehat{x}$ of minimal Euclidean norm of the unknown error-free least-squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - \widehat{b}\|_2.$$

Let $A^\dagger \in \mathbb{R}^{n \times m}$ denote the Moore-Penrose pseudoinverse of A. Then

$$\widehat{x} = A^\dagger \widehat{b}. \tag{1.4}$$

Because A has many positive singular values close to zero, the matrix $A^\dagger$ is of very large norm, and the solution of the available least-squares problem (1.1), given by

$$\check{x} = A^\dagger b = A^\dagger(\widehat{b} + e) = \hat{x} + A^\dagger e,$$

typically is dominated by the propagated error $A^\dagger e$, and then is meaningless. This difficulty can be mitigated by replacing the matrix A by a nearby matrix that does not have tiny positive singular values. This replacement commonly is referred to as *regularization*. One of the most popular regularization methods for discrete ill-posed problem (1.1) of small to moderate size is the truncated singular value decomposition (TSVD); see, e.g., [2,3,7]. Introduce the singular value decomposition

$$A = U \Sigma V^\top, \tag{1.5}$$

where $U = [u_1, u_2, \ldots, u_m] \in \mathbb{R}^{m \times m}$ and $V = [v_1, v_2, \ldots, v_n] \in \mathbb{R}^{n \times n}$ are orthogonal matrices, the superscript $^\top$ denotes transposition, and the nontrivial entries (known as the singular values) of the (possibly rectangular) diagonal matrix

$$\Sigma = \operatorname{diag}[\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}}] \in \mathbb{R}^{m \times n} \tag{1.6}$$

are ordered according to

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_{\min\{m,n\}} = 0.$$

Here r is the rank of A. Define the matrix

$$\Sigma_k = \operatorname{diag}[\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0] \in \mathbb{R}^{m \times n}$$

for $k \leq r$ by setting the singular values $\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_{\min\{m,n\}}$ in (1.6) to zero. Then the matrix

$$A_k = U \Sigma_k V^\top \tag{1.7}$$

is a best rank-k approximation of A in any unitarily invariant matrix norm, such as the spectral and Frobenius norms. We have

$$\|A_k - A\|_2 = \sigma_{k+1}, \quad \|A_k - A\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i{}^2}, \quad 0 \leq k \leq r,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and we define $A_0 = 0$ and $\sigma_{n+1} = 0$.

The TSVD method replaces the matrix A in (1.1) by A_k. Let $x_k \in \mathbb{R}^n$ denote the solution of minimal Euclidean norm of

$$\min_{x \in \mathbb{R}^n} \|A_k x - b\|_2.$$

It is given by $x_k = A_k^\dagger b$. The *discrepancy principle* prescribes that the truncation index $k \geq 0$ in (1.7) be chosen as the smallest integer such that

$$\frac{\|A_k x_k - b\|_2}{\|\widehat{b}\|_2} \leq \tau \epsilon, \tag{1.8}$$

where $\tau > 1$ is a user-chosen constant that is independent of the bound ϵ in (1.3); see [3,7]. We will use the discrepancy principle in the computed examples reported in Section 4. However, other methods for determining k also can be used when no accurate estimate of $\|e\|$ is available, such as the L-curve criterion and generalized cross validation; see, e.g., [2,7,11,12,14].

The modified truncated singular value decomposition (MTSVD) method proposed in [13] replaces the matrix A_k in (1.8) by a matrix that (generally) is closer to A in a unitarily invariant matrix norm with the same condition number as A_k. This replacement often results in more accurate approximations of $\widehat{x}$ when the discrepancy principle is used to determine the truncation index k; see [13] for computed examples.

The TSVD and MTSVD methods are not suitable for application to the solution of large-scale discrete ill-posed problems (1.1) due to the high cost of evaluating the SVD of a large matrix A; see, e.g., [5, p. 493] for counts of arithmetic floating point operations. However, the computational effort can be reduced by computing an approximation of the singular value decomposition by a randomized method and applying the modified truncated singular value decomposition to the computed approximate SVD. This yields the modified truncated randomized singular value decomposition (MTRSVD).

In recent years several randomized algorithms have been proposed for computing approximate factorizations of a large matrix, such as an approximate SVD; see, e.g., Halko et al. [6]. These factorizations have been used to compute approximate solutions to ill-posed problems (1.1) by Tikhonov regularization; see, e.g., Jia and Yang [10] and Xiang and Zou [15,16].

It is the purpose of the present paper to discuss the use of a randomized singular value decomposition (RSVD) in conjunction with the MTSVD described in [13]. We refer to this scheme as the MTRSVD method. Our reason for regularizing by a truncated singular value decomposition instead of using Tikhonov regularization is that the former method is easier to implement. Moreover, we base our method on the MTSVD instead of on the standard TSVD, because the former typically yields approximations of the desired solution (1.4) of higher quality when the required singular vectors have been computed with high enough accuracy.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the MTSVD, RSVD, and TRSVD methods. The proposed MTRSVD method is described in Section 3, where also its computational complexity and error bounds are presented. Section 4 shows several numerical examples that illustrate the efficiency of the MTRSVD method. Section 5 contains concluding remarks.

2. The MTSVD, RSVD, and TRSVD methods

2.1. The MTSVD method

We review the modified SVD (MTSVD) method introduced in [13]. First consider the TSVD method [3,7]. It replaces the matrix A in (1.1) by the matrix A_k in (1.8), and determines the least-squares solution of minimal Euclidean norm. We denote this solution by x_k and assume that $k \leq r = \operatorname{rank}(A)$. The vector x_k can be expressed as

$$x_k = A_k^{\dagger} b = \sum_{j=1}^{r} \phi_j^{(k)} \frac{u_j^{\top} b}{\sigma_j} v_j,$$

where the u_j and v_j are columns of the matrices U and V in (1.5), respectively, and the filter factors $\phi_j^{(k)}$ are defined by

$$\phi_j^{(k)} = \begin{cases} 1, & 1 \leq j \leq k, \\ 0, & k < j \leq r. \end{cases}$$

The condition number of A_k, given by

$$\kappa_2(A_k) = \frac{\sigma_1}{\sigma_k}, \tag{2.1}$$

grows with k. The larger the condition number (2.1), the more sensitive the vector x_k can be to the error e in b; see, e.g., [3,5,7] for discussions.

The MTSVD method in [13] gives a closest matrix $\widetilde{A}_{\widetilde{k}}$ to A in the spectral or Frobenius norms with a specified smallest singular value σ_k. Define the decomposition

$$\widetilde{A}_{\widetilde{k}} = U \widetilde{\Sigma}_{\widetilde{k}} V^{\top}, \tag{2.2}$$

where U and V are the orthogonal matrices in the SVD (1.5) of A, and the entries of $\widetilde{\Sigma}_{\widetilde{k}} = \operatorname{diag}[\widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_k, 0, \ldots, 0]$ are given by

$$\widetilde{\sigma}_j = \begin{cases} \sigma_j, & 1 \leq j \leq k, \\ \sigma_k, & k < j \leq \widetilde{k}, \end{cases}$$

where $\widetilde{k} \geq k$ is determined by the inequalities $\sigma_{\widetilde{k}} \geq \sigma_k/2$ and $\sigma_{\widetilde{k}+1} < \sigma_k/2$. It is shown in [13] that

$$\|A - \widetilde{A}_{\widetilde{k}}\|_2 < \|A - A_k\|_2, \quad \|A - \widetilde{A}_{\widetilde{k}}\|_F < \|A - A_k\|_F$$

when $\sigma_{k+1} > \sigma_k/2$. Moreover,

$$\kappa_2(\widetilde{A}_{\widetilde{k}}) = \kappa_2(A_k).$$

By replacing the matrix A in (1.1) by $\widetilde{A}_{\widetilde{k}}$ in (2.2), we can get a more accurate approximation of the desired solution (1.4),

$$\widetilde{x}_{\widetilde{k}} = \widetilde{A}_{\widetilde{k}}^{\dagger} b = \sum_{j=1}^{r} \widetilde{\phi}_j^{(\widetilde{k})} \frac{u_j{}^{\top} b}{\sigma_j} v_j$$

with filter factors

$$\widetilde{\phi}_j^{(\widetilde{k})} = \begin{cases} 1, & 1 \leq j \leq k, \\ \sigma_j/\sigma_k, & k < j \leq \widetilde{k}, \\ 0, & \widetilde{k} < j \leq r, \end{cases}$$

where we assume that $\widetilde{k} \leq r$. The definition of $\widetilde{x}_{\widetilde{k}}$ can be extended to allow $k \leq r$.

2.2. The RSVD and TRSVD methods

Algorithm 1 describes the randomized SVD (RSVD) method considered in [6]. The decomposition determined by the algorithm can be used to compute a low-rank approximation of a matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. The index ℓ in the algorithm has to be chosen large enough. We let $\ell = k_{target} + p$, where $k_{target} \ll \min\{m, n\}$ is the required rank of the low-rank approximation of A when seeking to satisfy the discrepancy principle, and p is an over-sampling parameter. The choices of k_{target} and p are illustrated in Section 4. In our analysis below, we set $k_{target} = k$. Algorithm 1 computes an approximate SVD of A by computing an orthogonal matrix Q_ℓ, whose range approximates that of A_k, in Step 3, and the SVD of a small matrix B_ℓ in Step 5. We refer to [6] for more details.

Algorithm 1 (RSVD). Given $A \in \mathbb{R}^{m\times n} (m \geq n)$, compute an approximate SVD: $A \approx \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top}$ with $\check{U}_\ell \in \mathbb{R}^{m\times \ell}$, $\check{\Sigma}_\ell \in \mathbb{R}^{\ell\times\ell}$, and $\check{V}_\ell \in \mathbb{R}^{n\times\ell}$, with $1 \leq \ell \ll n$.

1: Generate a Gaussian random matrix $\Omega_\ell \in \mathbb{R}^{n\times\ell}$.
2: Form the matrix $Y_\ell = A\Omega_\ell \in \mathbb{R}^{m\times\ell}$.
3: Compute the skinny QR factorization $Y_\ell = Q_\ell R_\ell$, where $Q_\ell \in \mathbb{R}^{m\times\ell}$ has orthonormal columns and $R_\ell \in \mathbb{R}^{\ell\times\ell}$ is upper triangular.
4: Form the matrix $B_\ell = Q_\ell^{\top} A \in \mathbb{R}^{\ell\times n}$.
5: Compute the SVD of the small matrix B_ℓ: $B_\ell = \check{W}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top}$.
6: Form the matrix $\check{U}_\ell = Q_\ell \check{W}_\ell \in \mathbb{R}^{m\times\ell}$. Then $A \approx \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top}$.

Algorithm 2 describes the randomized SVD for a matrix $A \in \mathbb{R}^{m\times n}$ with $m < n$. The algorithm is equivalent to applying Algorithm 1 to $A^{\top}$.

Algorithm 2 (RSVD*). Given $A \in \mathbb{R}^{m\times n} (m < n)$, compute an approximate SVD: $A \approx \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top}$ with $\check{U}_\ell \in \mathbb{R}^{m\times\ell}$, $\check{\Sigma}_\ell \in \mathbb{R}^{\ell\times\ell}$, and $\check{V} \in \mathbb{R}^{n\times\ell}$, with $1 \leq \ell \ll m$.

1: Generate a Gaussian random matrix $\Omega_\ell \in \mathbb{R}^{\ell\times m}$.
2: Form the matrix $Y_\ell = \Omega_\ell A \in \mathbb{R}^{\ell\times n}$.
3: Compute the skinny QR factorization $Y_\ell^{\top} = Q_\ell R_\ell$, where $Q_\ell \in \mathbb{R}^{n\times\ell}$ has orthonormal columns and $R_\ell \in \mathbb{R}^{\ell\times\ell}$ is upper triangular.
4: Form the matrix $B_\ell = AQ_\ell \in \mathbb{R}^{m\times\ell}$.
5: Compute the SVD of the small matrix B_ℓ: $B_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{W}_\ell^{\top}$.
6: Form the matrix $\check{V}_\ell = Q_\ell \check{W}_\ell \in \mathbb{R}^{n\times\ell}$. Then $A \approx \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top}$.

Algorithm 1 generates the approximation

$$\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^{\top} \tag{2.3}$$

of at most rank ℓ of A, where

$$\check{U}_\ell = [\check{u}_1, \check{u}_2, \ldots, \check{u}_\ell] \in \mathbb{R}^{m\times\ell}, \quad \check{V}_\ell = [\check{v}_1, \check{v}_2, \ldots, \check{v}_\ell] \in \mathbb{R}^{n\times\ell},$$

and $\check{\Sigma}_k = \mathrm{diag}[\check{\sigma}_1, \check{\sigma}_2, \ldots, \check{\sigma}_\ell] \in \mathbb{R}^{\ell\times\ell}$ with $\check{\sigma}_1 \geq \check{\sigma}_2 \geq \ldots \geq \check{\sigma}_\ell \geq 0$. As mentioned above, ℓ is generally chosen somewhat larger than the smallest positive integer k that allows the discrepancy principle to be satisfied; cf. (1.8). Let

$\check{U}_k = [\check{u}_1, \check{u}_2, \ldots, \check{u}_k] \in \mathbb{R}^{m\times k}$, $\check{V}_k = [\check{v}_1, \check{v}_2, \ldots, \check{v}_k] \in \mathbb{R}^{n\times k}$, and $\check{\Sigma}_k = \mathrm{diag}[\check{\sigma}_1, \check{\sigma}_2, \ldots, \check{\sigma}_k] \in \mathbb{R}^{k\times k}$ be submatrices of the matrices in the decomposition (2.3). Then

$$\check{A}_k = \check{U}_k \check{\Sigma}_k \check{V}_k^\top \tag{2.4}$$

is a best rank-k approximation of the matrix $\check{A}_\ell$. The following result gives an error bound for the TRSVD method.

Theorem 1. *([6]) Let $\check{A}_k$ be the rank-k TRSVD approximation to A defined by (2.4). Then the approximation error is*

$$||A - \check{A}_k||_2 \le \sigma_{k+1} + ||A - Q_\ell Q_\ell^\top A||_2. \tag{2.5}$$

3. The MTRSVD method and error bounds

3.1. The MTRSVD method

This subsection describes a novel modified truncated randomized singular value decomposition (MTRSVD) method for solving large-scale discrete ill-posed problems (1.1). The following result gives a closest matrix to the matrix $\check{A}_\ell$ in (2.3) in the spectral and Frobenius norms.

Theorem 2. *Let $\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^\top$ be the approximate truncated SVD (2.3) of A, and let $1 \le k \le \widehat{k} \le \ell$. A closest matrix $\widehat{A}_{\widehat{k}}$ to the matrix $\check{A}_\ell$ in the spectral and Frobenius norms with smallest singular value $\check{\sigma}_k$ is given by*

$$\widehat{A}_{\widehat{k}} = \check{U}_{\widehat{k}} \widehat{\Sigma}_{\widehat{k}} \check{V}_{\widehat{k}}^\top, \tag{3.1}$$

where the matrix $\check{U}_{\widehat{k}}$ is made up by the first $\widehat{k}$ columns of the matrix $\check{U}_\ell$ and the matrix $\check{V}_{\widehat{k}}$ is made up of the first $\widehat{k}$ columns of the matrix $\check{V}_\ell$. Moreover, the entries of $\widehat{\Sigma}_{\widehat{k}} = \mathrm{diag}[\widehat{\sigma}_1, \widehat{\sigma}_2, \ldots, \widehat{\sigma}_{\widehat{k}}] \in \mathbb{R}^{\widehat{k}\times\widehat{k}}$ are given by

$$\widehat{\sigma}_j = \begin{cases} \check{\sigma}_j, & 1 \le j \le k, \\ \check{\sigma}_k, & k < j \le \widehat{k}, \end{cases}$$

where $\widehat{k}$ is determined by the inequalities $\check{\sigma}_{\widehat{k}} \ge \check{\sigma}_k/2$ and $\check{\sigma}_{\widehat{k}+1} < \check{\sigma}_k/2$, and the $\check{\sigma}_j$ are the singular values of $\check{A}_\ell$.

Proof. The proof is similar to the proof of [13, Theorem 2.3] and therefore is omitted. □

Remark 1. Let $\ell > \widehat{k}$ and assume that $\check{\sigma}_k - \check{\sigma}_{\widehat{k}} < \check{\sigma}_{k+1}$ and $\check{\sigma}_{\widehat{k}+1} < \check{\sigma}_{k+1}$. The latter inequalities impose conditions on the decay of the singular values $\check{\sigma}_k, \check{\sigma}_{k+1}, \ldots, \check{\sigma}_{\widehat{k}+1}$ and hold for many linear discrete ill-posed problems. Then

$$\|\check{A}_\ell - \widehat{A}_{\widehat{k}}\|_2 < \|\check{A}_\ell - \check{A}_k\|_2, \tag{3.2}$$

i.e., $\widehat{A}_{\widehat{k}}$ is a more accurate approximation of $\check{A}_\ell$ in (2.3) than $\check{A}_k$. The inequality (3.2) follows from the observations that

$$\|\check{A}_\ell - \widehat{A}_{\widehat{k}}\|_2 = \max\{\check{\sigma}_k - \check{\sigma}_{\widehat{k}}, \check{\sigma}_{\widehat{k}+1}\}, \qquad \|\check{A}_\ell - \check{A}_k\|_2 = \check{\sigma}_{k+1}.$$

Moreover,

$$\kappa_2(\widehat{A}_{\widehat{k}}) = \kappa_2(\check{A}_k).$$

The matrix (3.1) can be used to determine a regularized approximate solution of (1.1).

Theorem 3. *Let $\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^\top$ be the approximate partial SVD in (2.3) of A and let $\widehat{A}_{\widehat{k}}$ be a closest matrix defined by (3.1) with $1 \le k \le \widehat{k} \le \ell$. Then an approximate regularized solution of (1.1) is given by*

$$\widehat{x}_{\widehat{k}} = \sum_{j=1}^{\ell} \widehat{\phi}_j^{(\widehat{k})} \frac{\check{u}_j^\top b}{\check{\sigma}_j} \check{v}_j, \tag{3.3}$$

with the filter factors

$$\widehat{\phi}_j^{(\widehat{k})} = \begin{cases} 1, & 1 \le j \le k, \\ \check{\sigma}_j/\check{\sigma}_k, & k < j \le \widehat{k}, \\ 0, & \widehat{k} < j \le \ell. \end{cases}$$

Proof. Replacing A by $\widehat{A}_{\widehat{k}}$ in (1.1) yields

$$\min_{x\in\mathbb{R}^n} \|\widehat{A}_{\widehat{k}}x - b\|_2.$$

The minimal-norm least-squares solution of this minimization problem is given by (3.3), which is an approximate regularized solution of (1.1). □

We remark that the approximate solution (3.3) typically is of higher quality than the approximate solution determined by replacing A by $\widetilde{A}_k$ and using the TRSVD method, because $\widehat{A}_{\widehat{k}}$ often approximates $\widetilde{A}_\ell$ more accurately than $\breve{A}_k$, cf. (3.2), provided that the singular values $\{\breve{\sigma}_j\}_{j=1}^k$ and singular vectors $\{\breve{u}_j, \breve{v}_j\}_{j=1}^{\widehat{k}}$ of $\breve{A}$ approximate the corresponding singular values and vectors of A with sufficient accuracy.

Algorithm 3 implements an application of Theorem 3. The algorithm consists of three steps: Step 1 computes a partial approximate SVD of A by Algorithm 1, and Step 2 gives the rank-k TRSVD approximation of A by the TRSVD in (2.4). The regularization parameter k is determined by the discrepancy principle similarly as in [13]. Step 3 uses the MTSVD method in [13] to compute the approximate solution $\widehat{x}_{\widehat{k}}$ of the large-scale linear discrete ill-posed problems (1.1). Algorithm 3 does not require $m \geq n$.

Algorithm 3 (MTRSVD). Given $A \in \mathbb{R}^{m\times n}$, $\ell \ll \min\{m,n\}$. Compute an approximate regularized solution of (1.1).

1: Compute the approximate partial RSVD $\breve{A}_\ell = \breve{U}\breve{\Sigma}\breve{V}^\top$ of A by Algorithm 1.
2: Compute the rank-k TRSVD approximation $\breve{A}_k$ of the matrix A by using the discrepancy principle.
3: Compute the approximate regularized solution $\widehat{x}_{\widehat{k}}$ of (1.1) by (3.3). We assume that $1 \leq k \leq \widehat{k} \leq \ell$.

3.2. Computational complexity and error bounds

We discuss the computational complexity of the MTRSVD method. For a given matrix $A \in \mathbb{R}^{m\times n}$ in (1.1) with $m \geq n \gg \ell$, the count of arithmetic floating point operations (flops) for Algorithm 1 is about $4mn\ell$, see [15], while the computation of the standard SVD of A is about $6mn^2 + 20n^3$ flops; see [5]. Other computations required by Algorithm 3 have negligible flop counts.

At the end of this section, we analyze the error in approximate solutions determined by Algorithm 3. We first need the following estimates shown in [6].

Lemma 1. *Suppose that $A \in \mathbb{R}^{m\times n}$ has the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0$. Let $\Omega_\ell \in \mathbb{R}^{n\times(k+p)}$ be a Gaussian matrix with $\ell := k + p \leq \min\{m,n\}$ and $p \geq 4$. Let the columns of Q_ℓ form an orthonormal basis for the range of the matrix $A\Omega_\ell$; see Algorithm 1. Then*

$$\|A - Q_\ell Q_\ell^\top A\|_2 \leq (1 + 6\sqrt{\ell p \log p})\sigma_{k+1} + 3(\ell\Sigma_{j>k}\sigma_j^2)^{1/2} \tag{3.4}$$

with probability not less than $1 - 3p^{-p}$.

Under a mild assumption on p, the error bound (3.4) can be simplified to

$$\|A - Q_\ell Q_\ell^\top A\|_2 \leq (1 + 9\sqrt{\ell \min\{m,n\}})\sigma_{k+1};$$

see [6] for details.

The following result gives an error bound for the approximation determined by the MTRSVD method.

Theorem 4. *Let $\widehat{A}_{\widehat{k}}$ in (3.1) be an approximation of A determined by Algorithm 3. Then, using the notation of Lemma 1, the approximation error is bounded by*

$$||A - \widehat{A}_{\widehat{k}}||_2 \leq \sigma_{k+1} + \sigma_k + (1 + 6\sqrt{\ell p \log p})\sigma_{k+1} + 3(\ell\Sigma_{j>k}\sigma_j^2)^{1/2}. \tag{3.5}$$

Proof. We have

$$||A - \widehat{A}_{\widehat{k}}||_2 \leq ||A - \breve{A}_k||_2 + ||\breve{A}_k - \widehat{A}_{\widehat{k}}||_2, \quad ||\breve{A}_k - \widehat{A}_{\widehat{k}}||_2 = \sigma_k$$

by Theorem 1 and (3.4). □

Similarly as above, we can simplify the bound of Theorem 4 under mild condition on p to

$$\|A - \widehat{A}_{\widehat{k}}\|_2 \leq (2 + 9\sqrt{\ell \min\{m,n\}})\sigma_{k+1} + \sigma_k. \tag{3.6}$$

The following theorem gives a sharper bound than (3.6).

Theorem 5. *Let $\widehat{A}_{\widehat{k}} \in \mathbb{R}^{m\times n}$ in (3.1) with $m \geq n$ be an approximation of A determined by Algorithm 3. Then, using the notation of Lemma 1,*

$$||A - \widehat{A}_{\widehat{k}}||_2 \leq \widetilde{\sigma}_{k+1} + \sigma_k + (1 + 6\sqrt{\ell p \log p})\sigma_{k+1} + 3(\ell \Sigma_{j>k}\sigma_j^2)^{1/2}, \tag{3.7}$$

where $\widetilde{\sigma}_{k+1}$ is the $(k+1)$st singular value of $Q_\ell^\top A$ and satisfies

$$\sigma_{m-q+1} \leq \widetilde{\sigma}_{k+1} \leq \sigma_{k+1}$$

with the definition $\sigma_{n+1} = \ldots = \sigma_m = 0$.

The bound of Theorem 5 with obvious modifications also holds for $m < n$.

3.3. The randomized power method

The MTRSVD method as implemented by Algorithm 3 performs well when applied to many discrete ill-posed problems that arise from the discretization or linear ill-posed problems in one space-dimension. This is illustrated in Section 4. However, for some discrete ill-posed problems the MTRSVD method is not able to determine an approximation of the desired solution $\widehat{x}$ with satisfactory accuracy. This is true, for instance, for several discrete ill-posed problems that stem from the discretization of ill-posed problems in two space-dimensions.

Poor performance of the MTRSVD method, when the MTSVD method performs well, stems from the fact that the former method does not determine a suitable solution subspace. This difficulty with the MTRSVD method typically arises when the singular values of the matrix A in (1.1) decay to zero fairly slowly with increasing index. Slow convergence of the singular values to zero with increasing index is fairly common for linear discrete ill-posed problems in two or more space-dimensions.

It is well known that approximations of the singular vectors determined by the RSVD method are less accurate when the singular values decay to zero slowly with increasing index number than when they decay to zero quickly. This behavior is suggested by Lemma 1.

The randomized power method described by Halko et al. [6] provides a remedy. This method determines the singular vectors of $M = (AA^\top)^q A$ for a small integer $q \geq 1$. We observe that the matrix M has the same singular vectors as A, while its singular values $\sigma_j(M) = \sigma_j(A)^{2q+1}$, $j = 1, 2, \ldots$, decay faster to zero than the singular values $\sigma_j(A)$ of A when the latter are small. Therefore, the RSVD method applied to M is able to determine the singular vectors associated with the largest singular values more accurately than when the RSVD method is applied to A. This observation suggests that we modify Step 1 in Algorithm 3 by applying the RSVD method to the matrix M instead of to A.

It is undesirable to explicitly form the possibly very large matrix M. Algorithm 4 determines approximations of a partial SVD of A by applying a randomized power method to M without explicitly forming M. In the computed examples of Section 4, we set $q = 1$. Algorithm 4 assumes that $m \geq n$. It is straightforward to modify the algorithm to be applicable when $m < n$.

Algorithm 4 (RSVD(q)). Given $A \in \mathbb{R}^{m\times n}$, $m \geq n$, and $\ell \ll n$. Compute an approximation $\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}^\top$ of a partial SVD of A, with $\check{U}_\ell \in \mathbb{R}^{m\times\ell}$, $\check{\Sigma}_\ell \in \mathbb{R}^{\ell\times\ell}$, and $\check{V}_\ell \in \mathbb{R}^{n\times\ell}$ by q steps of the power method applied to $AA^\top$.

1: Generate a Gaussian random matrix $\Omega_\ell \in \mathbb{R}^{n\times\ell}$.
2: Form the matrix $Y_0 = A\Omega_\ell \in \mathbb{R}^{m\times\ell}$.
3: Compute the matrix $Q_\ell^{(0)} \in \mathbb{R}^{m\times\ell}$ with orthonormal columns by compact QR factorization $Y_0 = Q_\ell^{(0)} R_\ell^{(0)}$.
4: **for** $j = 1, 2, \ldots, q$ **do**
5: Form the matrix $\tilde{Y}_j = A^\top Q_\ell^{(j-1)}$, and compute the compact QR factorization $\tilde{Y}_j = \tilde{Q}_\ell^{(j)} \tilde{R}_\ell^{(j)}$ with $\tilde{Q}_\ell^{(j)} \in \mathbb{R}^{n\times\ell}$ and $\tilde{R}_\ell^{(j)} \in \mathbb{R}^{\ell\times\ell}$.
6: Form the matrix $Y_j = A\tilde{Q}_\ell^{(j)}$, and compute the compact QR factorization $Y_j = Q_\ell^{(j)} R_\ell^{(j)}$ with $Q_\ell^{(j)} \in \mathbb{R}^{m\times\ell}$ and $R_\ell^{(j)} \in \mathbb{R}^{\ell\times\ell}$.
7: **end for**
8: Let $Q_\ell = Q_\ell^{(q)}$.
9: Form the matrix $B_\ell = (Q_\ell)^\top A \in \mathbb{R}^{\ell\times n}$.
10: Compute the SVD of the small matrix B_ℓ: $B_\ell = \check{W}_\ell \check{\Sigma}_\ell \check{V}_\ell^\top$.
11: Form the matrix $\check{U}_\ell = Q_\ell^{(q)} \check{W}_\ell \in \mathbb{R}^{m\times\ell}$. Then $\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}_\ell^\top$ is the desired approximate partial SVD of A.

When the singular vectors are not approximated accurately enough, the MTRSVD method may determine worse approximations of the desired solution $\widehat{x}$ of (1.1) than the TRSVD method. This depends on that the error in the singular vectors often increases with their index, and the MTRSVD typically includes more (approximate) singular vectors in the computed solution than the TRSVD method. Thus, the MTRSVD method may include more poorly approximated singular vectors in the computed solution than the TRSVD method. Some computed examples in Section 4 with discrete ill-posed problems that arise from the discretization of ill-posed problems in two space-dimensions illustrate this. The good performance of the MTRSVD method reported in [13] is achieved, because the singular values and vectors are computed accurately in the examples reported there.

The shortcoming of the MTRSVD method mentioned above can be remedied by applying the power method (Algorithm 4) to determine improved approximations of the singular vectors required to represent a regularized approximate solution of (1.1). Algorithm 5 describes such a method, which we refer to as MTRSVD(q).

Algorithm 5 (MTRSVD(q)). Given $A \in \mathbb{R}^{m\times n}$. Let $\ell = k + p \ll \min\{m, n\}$ and let $q > 0$ be a small integer. Compute an approximate regularized solution x_k of (1.1).

1: When $m \geq n$, compute an approximate partial SVD $\check{A}_\ell = \check{U}_\ell \check{\Sigma}_\ell \check{V}^\top$ of A by q steps of power method applied to $AA^\top$ as described by Algorithm 4; an analogous algorithm can be applied when $m < n$.
2: Compute the rank-k (with $k \leq \ell$) TRSVD approximation $\check{A}_k$ of $\check{A}$, where k is determined by the discrepancy principle.
3: Compute the approximate regularized solution $\widehat{x}_k$ of (1.1) by (3.3) with the matrix $\widetilde{A}_\ell$ in Theorem 3 replaced by $\check{A}_\ell$.

4. Numerical experiments

We first consider three linear discrete ill-posed problems that arise from the discretization of linear ill-posed problems in one space-dimension. These problems stem from Hansen's Regularization Tools [8]. The performance of the MTRSVD and TRSVD methods are reported in Subsection 4.1. Linear discrete ill-posed problems that stem from the discretization of linear ill-posed problems in two space-dimensions are discussed in Subsection 4.2. One of the problems is from IR Tools [4]. The problems in this subsection illustrate that it may be necessary to apply the power method (Algorithm 4) to obtain useful approximations of the desired solution $\widehat{x}$ of (1.1). In all our examples $m \geq n$; however, similar results are achieved when $m < n$.

4.1. Ill-posed problems in one space-dimension

All calculations reported in this subsection were carried out in MATLAB R2016b on a laptop computer with 2.2 GHz Intel Core i5-5200U CPU and 4 GB RAM.

We consider the test problems *deriv2*, *gravity*, and *heat* from [8]. They are linear discrete ill-posed problems, whose singular values converge to zero with increasing index slowly, quickly, and moderately fast, respectively. All examples arise from the discretization of Fredholm integral equations of the first kind of the form

$$\int_a^b \kappa(s,t)x(t)dt = g(s), \quad c \leq s \leq d, \tag{4.1}$$

with a smooth kernel κ. MATLAB functions in [8] are used to generate the matrices A and the exact solutions $\widehat{x}$. We then determine $\widehat{b} = A\widehat{x}$, and the error contaminated right-hand data vector b in (1.1) by (1.2), where e models white Gaussian noise with different noise levels $\epsilon = \{0.1, 0.01, 0.001\}$; cf. (1.3).

Let x_k denote a computed approximation of the desired solution $\widehat{x}$. The *relative error* is defined as

$$Err = \frac{||x_k - \widehat{x}||_2}{||\widehat{x}||_2}. \tag{4.2}$$

The parameter ℓ in our algorithms is important for the good performance of the randomized SVD method. We would like to choose ℓ in Algorithm 1 so that

$$\frac{\|(I - Q_\ell Q_\ell^\top)A\|_2}{\|\widehat{b}\|_2} \leq \epsilon, \tag{4.3}$$

where the matrix $Q_\ell \in \mathbb{R}^{m\times\ell}$ has random orthogonal columns, like in Algorithm 1, and ϵ is an available bound for the error e in b; cf. (1.3). Then, if σ_{k+1} is small, $\|A - \check{A}_\ell\|_2$ is close to $\epsilon\|\widehat{b}\|_2$; cf. (2.5). Our numerical experience suggests that for many problems ℓ has to be somewhat larger than the smallest k-value, such that the discrepancy principle (1.8) holds.

Theorem 6. *Let the matrix $A \in \mathbb{R}^{m\times n}$ have the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{m,n\}} \geq 0$, and let $1 \leq \ell \leq \min\{m, n\}$. Then $\sigma_{\ell+1}$ is a lower bound for the expression in the left-hand side of (4.3), i.e., $e_\ell := \|(I - Q_\ell Q_\ell^\top)A\|_2 \geq \sigma_{\ell+1}$. This bound is achieved for $Q_\ell = [u_1, u_2, \ldots, u_\ell]$, where the vectors u_j are left singular vectors of A; cf. (1.5).*

Proof. First let $Q_\ell = U_\ell = [u_1, u_2, \ldots, u_\ell] \in \mathbb{R}^{m\times\ell}$, where the vectors u_j are left singular vectors of A; cf. (1.5). Using the SVD (1.5) of A, we obtain

$$\|A - U_\ell U_\ell^\top A\|_2 = \|U\Sigma V^\top - U_\ell U_\ell^\top U\Sigma V^\top\|_2 = \|\Sigma - U^\top U_\ell U_\ell^\top U\Sigma V^\top\|_2 = \sigma_{\ell+1}.$$

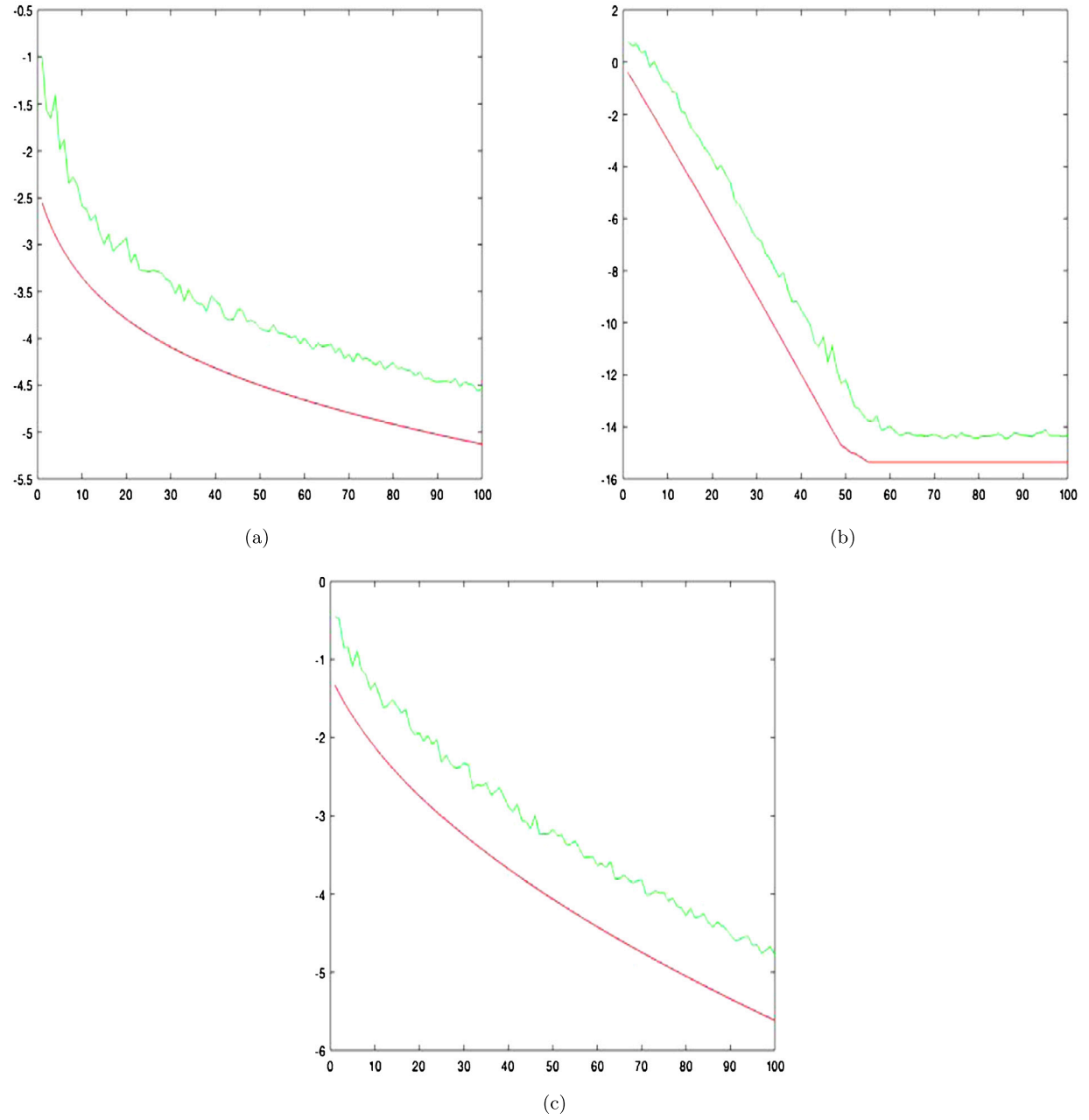

Fig. 1. Example 4.1: Approximation errors $e_k = ||(I - Q_k Q_k^\top)A||_2$ for random matrix $Q_k \in \mathbb{R}^{m\times k}$ with orthonormal columns (green jagged graph) and the $(k+1)$st singular value of $A \in \mathbb{R}^{200\times 200}$ (red smooth graph), which is a lower bound for e_k, as functions of k. The graphs show the logarithm in base 10 of these quantities for the test problems (a) *deriv2*, (b) *gravity*, and (c) *heat*. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Let $B = U^\top U_\ell U_\ell^\top U \Sigma V^\top \in \mathbb{R}^{m\times n}$. Then B is of rank at most ℓ and has the singular value decomposition $B = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^\top$ with

$$\widetilde{\Sigma} = \mathrm{diag}[\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_{\min\{m,n\}}], \qquad \widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \ldots \geq \widetilde{\sigma}_{\min\{m,n\}} \geq 0$$

and $\widetilde{\sigma}_j = 0$ for $j \geq \ell$. It follows from [9, Theorem 3.4.5] and the Fan dominance theorem contained in [9, Corollary 3.5.9] that

$$\|A - B\|_2 \geq \|\Sigma - \widetilde{\Sigma}\|_2 = \max\{\max_{1\leq j\leq \ell} |\sigma_j - \widetilde{\sigma}_j|, \sigma_{\ell+1}\}.$$

This shows the theorem. □

Table 1
Example 4.1: Comparison of CPU times (in seconds) and relative errors for the MTRSVD, TSVD, MTSVD, and TRSVD methods for matrices $A \in \mathbb{R}^{n\times n}$ of different sizes n and for three noise levels ϵ.

Method	$\epsilon = 0.1$		$\epsilon = 0.01$		$\epsilon = 0.001$	
	Time	Err	Time	Err	Time	Err
$n = 1000$						
TSVD	0.727	0.3451	0.734	0.2347	0.723	0.1608
MTSVD	0.729	0.3364	0.735	0.2203	0.724	0.1480
TRSVD	0.031	0.3461	0.033	0.2342	0.044	0.1512
MTRSVD	0.033	0.3364	0.035	0.2191	0.046	0.1457
$n = 2500$						
TSVD	12.463	0.3392	12.141	0.2165	11.835	0.1503
MTSVD	12.464	0.3174	12.143	0.1999	11.837	0.1331
TRSVD	0.111	0.3188	0.104	0.2084	0.142	0.1480
MTRSVD	0.113	0.2878	0.106	0.1862	0.145	0.1426
$n = 20000$						
TSVD	–	–	–	–	–	–
MTSVD	–	–	–	–	–	–
TRSVD	2.788	0.282	2.291	0.1821	4.022	0.1208
MTRSVD	2.981	0.2523	2.373	0.1719	4.100	0.1117

Fig. 1 displays σ_{k+1}, the $(k+1)$st largest singular value of A (red smooth graphs), and the quantity e_k of Theorem 6 (green jagged graphs) as functions of k for three linear discrete ill-posed problems. The latter graphs are jagged, because the matrix Q_k has random orthonormal columns. The figure illustrates that in order for the inequality (4.3) to hold, the parameter ℓ has to be larger than the smallest k such that $\sigma_{k+1} \leq \epsilon$.

Let k_{target} be the smallest k-value so that the discrepancy principle (1.8) is satisfied. We seek to choose ℓ in Algorithm 1 that is somewhat larger than k_{target}, i.e., we let $\ell = k_{target} + p$ for some moderate positive integer p.

The value k_{target} depends both on the problem being solved and on the size of ϵ, and is rarely known in advance. In practice, ℓ is often determined in an adaptive way, i.e., we seek to determine a computed solution that satisfies the discrepancy principle and check whether (4.3) holds. The parameter ℓ usually is not determined very carefully, because using an ℓ-value that is somewhat larger than necessary does not reduce the accuracy in the computed solution and the additional cost of such a choice of ℓ often is negligible. Therefore, adaptive methods for selecting ℓ often start with a fairly large initial value of ℓ, and then increase ℓ, if necessary, with not very small "steps".

The determination of a suitable ℓ is not the main focus of this paper. We therefore, for simplicity, in our experiments choose fairly large values of ℓ. In this subsection, we set $\ell = 120$ when $\epsilon = 0.001$, and let $\ell = 70$ when $\epsilon = 0.1$ or $\epsilon = 0.01$.

We report the CPU time required in seconds (CPU) and the relative error in the computed approximate solution (4.2) (Err) for each method and several noise levels. To gain insight into the average behavior of the solution methods, we report for every example the average of the relative errors in the computed approximate solutions x_k over 100 runs for each noise level, with exception for experiments with $n = 20000$. The latter results are from a single trial.

Example 4.1. Consider the Fredholm integral equations of the first kind (4.1) with

$$\kappa(s,t) = \begin{cases} s(t-1), & s < t, \\ t(s-1), & s \geq t, \end{cases} \qquad x(t) = t, \quad g(s) = \frac{s^3 - s}{6},$$

and $a = c = 0$, $b = d = 1$. We use the MATLAB code *deriv2* from [8] to generate the matrix $A \in \mathbb{R}^{n\times n}$ for $n \in \{1000, 2500, 20000\}$. The code also generates the desired solution $\widehat{x} \in \mathbb{R}^n$. We let $e \in \mathbb{R}^n$ model white Gaussian noise, scaled such that $\|e\| = \epsilon$, and determine the "available" error-contaminated data vector b using (1.2).

Table 1 presents CPU times and relative errors for the MTRSVD method and compares this method to the TSVD, MTSVD, and TRSVD methods for matrices $A \in \mathbb{R}^{n\times n}$ for $n \in \{1000, 2500, 20000\}$ and noise levels $\epsilon \in \{0.1, 0.01, 0.001\}$. The table shows the CPU time for MTRSVD to be lower than for the TSVD and MTSVD methods, and the relative errors achieved with the MTRSVD method to be smaller than those for the TRSVD method. When n is large, say $n = 20000$, the TSVD and MTSVD cannot be evaluated on the laptop computer used for the experiments in this subsection, while the MTRSVD method performs well. The table entries for the TSVD and MTSVD methods for $n = 20000$ therefore are marked by $-$.

Fig. 2 displays the exact and computed approximate solutions determined by the MTSVD and MTRSVD methods for $n = 2500$ and noise level $\epsilon = 0.001$. The approximate solution determined by the MTRSVD method shown in Fig. 2(b) is closer to the exact solution than the approximate solution computed by MTSVD and depicted in Fig. 2(a). Table 1 and Fig. 2 illustrate the benefit of using the MTRSVD method. □

Example 4.2. This test example uses the MATLAB code *gravity* in [8] to determine the matrix $A \in \mathbb{R}^{n\times n}$ and the desired solution $\widehat{x} \in \mathbb{R}^n$. The error vector and error-contaminated data vector $b \in \mathbb{R}^n$ are generated similarly as in Example 4.1. Table 2 displays the CPU time (in seconds) and the relative errors in the approximate solutions computed by the MTRSVD,

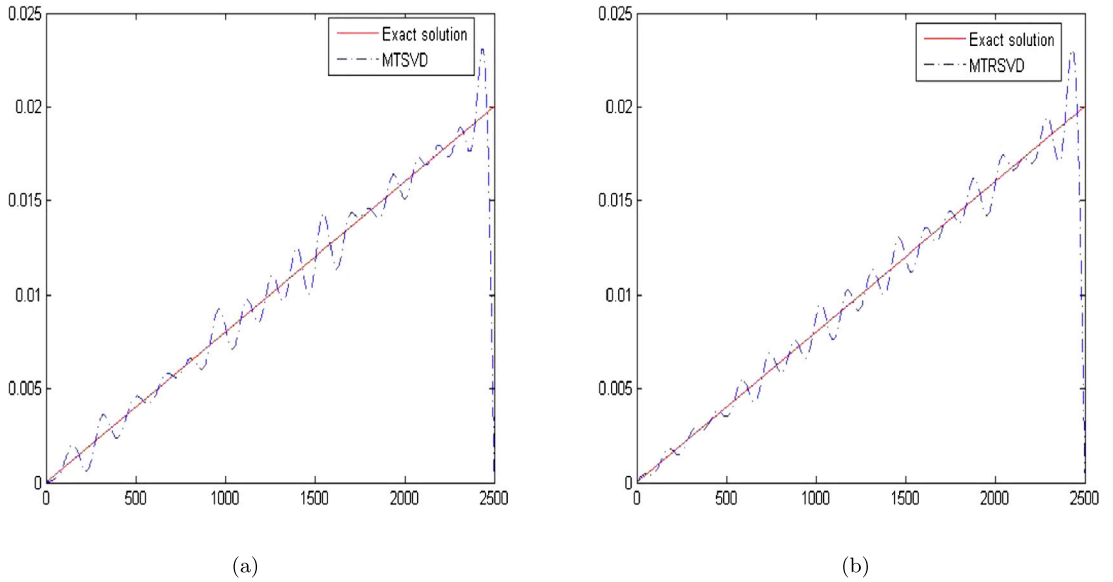

Fig. 2. Example 4.1: Exact solution and approximate solutions computed by (a) MTSVD and (b) MTRSVD for $n = 2500$ and noise level $\epsilon = 0.001$.

Table 2
Example 4.2: Comparison of CPU times and relative errors using MTRSVD, TSVD, MTSVD, and TRSVD for different matrix sizes n and noise levels ϵ.

Method	$\epsilon = 0.1$		$\epsilon = 0.01$		$\epsilon = 0.001$	
	Time	Err	Time	Err	Time	Err
$n = 1000$						
TSVD	0.655	0.0753	0.612	0.0322	0.617	0.0144
MTSVD	0.657	0.0676	0.614	0.0276	0.618	0.0122
TRSVD	0.032	0.0752	0.031	0.0318	0.031	0.0146
MTRSVD	0.034	0.0678	0.033	0.0275	0.032	0.0123
$n = 2500$						
TSVD	10.807	0.0410	10.713	0.0289	10.779	0.0136
MTSVD	11.808	0.0338	10.714	0.0208	10.780	0.0121
TRSVD	0.097	0.0612	0.103	0.0195	0.110	0.0143
MTRSVD	0.099	0.0536	0.105	0.0176	0.112	0.0110
$n = 20000$						
TSVD	–	–	–	–	–	–
MTSVD	–	–	–	–	–	–
TRSVD	2.361	0.0613	2.354	0.0217	2.221	0.0095
MTRSVD	2.447	0.0554	2.439	0.0201	2.314	0.0074

TSVD, MTSVD, and TRSVD methods for problems of different sizes and for three noise levels. Table 2 shows the MTRSVD method to require less CPU time than the TSVD and MTSVD methods, and determines an approximate solution with a smaller relative error than TRSVD for most noise levels and problem sizes. Again, when n is large, such as $n = 20000$, the TSVD and MTSVD methods cannot be used.

Fig. 3 displays the exact and computed approximate solutions determined by the MTSVD and MTRSVD methods for $n = 2500$ and noise level $\epsilon = 0.001$. The approximate solution determined by MTRSVD shown in Fig. 3(b) is seen to be closer to the exact solution $\widehat{x}$ than the approximate solution computed by MTSVD depicted in Fig. 3(a). Table 2 and Fig. 3 illustrate the benefit of using the MTRSVD method. □

Example 4.3. This example arises from the discretization of a first kind Volterra integral equation on the interval $[0, 1]$ with a convolution kernel. The MATLAB code *heat* in [8] is used to generate the matrix $A \in \mathbb{R}^{n\times n}$ and the desired solution $\widehat{x}$. The noise vector $e \in \mathbb{R}^n$ and the error-contaminated data vector $b \in \mathbb{R}^n$ are generated analogously as in the previous examples. Table 3 compares the CPU times and relative errors for the MTRSVD, TSVD, MTSVD, and TRSVD methods for problems of different sizes n and with different noise levels. Fig. 4 displays $\widehat{x}$ and approximate solutions computed by the MTSVD and MTRSVD methods for $n = 2500$ and noise level $\epsilon = 0.001$. The MTRSVD method is seen to be competitive both with regard to accuracy and CPU time. □

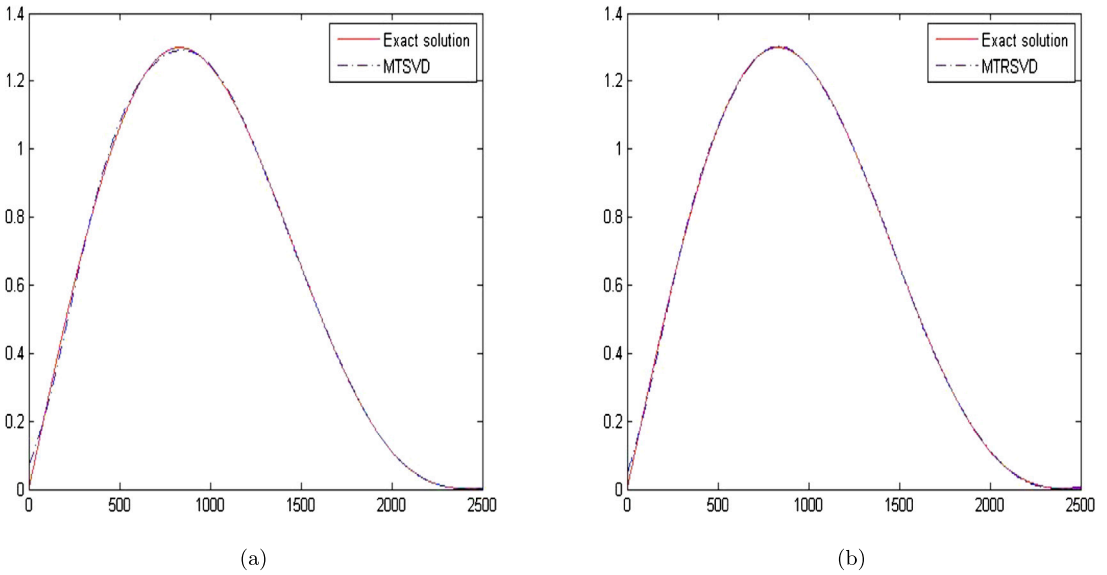

Fig. 3. Example 4.2: Exact solution $\widehat{x}$ and approximate solutions computed by (a) MTSVD and (b) MTRSVD for $n = 2500$ and noise level $\epsilon = 0.001$.

Table 3
Example 4.3: Comparison of CPU times and relative errors using MTRSVD, TSVD, MTSVD, and TRSVD for different sizes and noise levels.

Method	$\epsilon = 0.1$		$\epsilon = 0.01$		$\epsilon = 0.001$	
	Time	Err	Time	Err	Time	Err
$n = 1000$						
TSVD	0.663	0.2504	0.652	0.1048	0.688	0.0296
MTSVD	0.664	0.2132	0.653	0.0746	0.689	0.0223
TRSVD	0.030	0.2479	0.032	0.0999	0.031	0.0276
MTRSVD	0.032	0.2107	0.035	0.0628	0.034	0.0228
$n = 2500$						
TSVD	10.800	0.1845	10.872	0.0703	10.808	0.0265
MTSVD	10.802	0.1461	10.873	0.0575	10.810	0.0193
TRSVD	0.098	0.1894	0.106	0.0570	0.130	0.0264
MTRSVD	0.100	0.1536	0.109	0.0444	0.133	0.0218
$n = 20000$						
TSVD	–	–	–	–	–	–
MTSVD	–	–	–	–	–	–
TRSVD	2.905	0.1385	2.377	0.0416	2.686	0.0167
MTRSVD	2.981	0.1120	2.458	0.0279	2.766	0.0138

We conclude this subsection with a comment on the bounds furnished by Theorems 4 and 5 for the above examples. For all examples, the right-hand sides (3.5) and (3.7) are close. However, these bounds are far from sharp. They are about a factor 10 to 100 larger than the left-hand side of (3.5).

4.2. Ill-posed problems in two space-dimensions

All examples in this subsection are solved using MATLAB R2019b on a laptop computer with 1.4 GHz Dual-Core Intel Core i5 and 4 GB RAM. The first example is an image deblurring problem from *IR Tools* [4] and the second example is a discretization of a Fredholm integral equation of the first kind in two space-dimensions. The discretized problem is determined with the aid of the code *baart* from [8]. The examples illustrate the performance of Algorithm 5 for $q = 0$ and $q = 1$.

For the examples of this subsection, the choice of the subspace dimension ℓ requires some attention. We will use different ℓ-values for different methods when necessary. A method may require ℓ to be quite large to be able to satisfy the discrepancy principle, and we do not want to use the same large value of ℓ for the other methods if this is not needed, because this would make the latter methods unnecessarily slow. Similarly as in the previous subsection, we determine the required ℓ-values only roughly; if a certain ℓ-value is found not to be large enough to be able to satisfy the discrepancy principle, then we generously increase ℓ in steps of several hundred or even a thousand until we are able to satisfy the discrepancy principle. The specific ℓ-value used for each example and method is stated in the tables. We also tabulate the

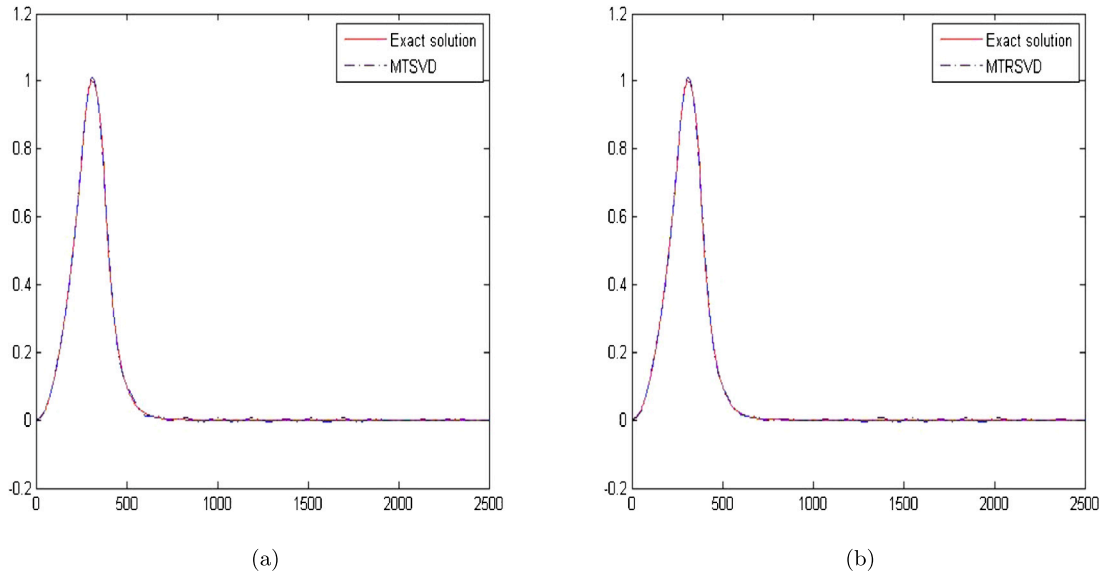

Fig. 4. Example 4.3: Exact solution $\widehat{x}$ and approximate solutions computed by (a) MTSVD and (b) MTRSVD for $n = 2500$ and noise level $\epsilon = 0.001$.

Table 4
Example 4.4: Comparison of CPU times, relative errors, and truncation index values of the TSVD, MTSVD, TRSVD(1), and MTRSVD(1) methods applied to the restoration of speckle blur and noise contaminated images. When $n = 4900$, we let $\ell = 3000$ and $\ell = 4000$ for $\epsilon = 0.01$ and $\epsilon = 0.001$, respectively; when $n = 10000$, we let $\ell = 6000$ and $\ell = 7000$ for the noise levels $\epsilon = 0.01$ and $\epsilon = 0.001$, respectively.

Method	$\epsilon = 0.01$			$\epsilon = 0.001$		
	Time	Err	k or $\widehat{k}$	Time	Err	k or $\widehat{k}$
$n = 4900$						
TSVD	124.4375	0.7688	2	88.9597	0.7698	2
MTSVD	124.4521	0.7688	2	88.9643	0.7698	2
TRSVD(1)	76.1313	0.2264	1763	108.1081	0.1586	3289
MTRSVD(1)	76.494	0.2124	2484	108.1686	0.1483	3521
$n = 10000$						
TSVD	917.3022	0.7733	2	834.3533	0.7686	2
MTSVD	917.4021	0.7733	2	834.3971	0.7686	2
TRSVD(1)	494.9429	0.2187	3050	713.3095	0.1411	5890
MTRSVD(1)	495.5029	0.1962	4848	713.6283	0.1288	6695

truncation index for TSVD and TRSVD, denoted by k, as well as the truncation index for MTSVD and MTRSVD, denoted by $\widehat{k}$. These truncation indices may depend on the subspace dimension ℓ; a larger value of ℓ may decrease the relative error (4.2) as well as the truncation indices k and $\widehat{k}$.

MTRSVD(1) in the table denotes Algorithm 5 with $q = 1$; the method MTRSVD(0) stands for the basic MTRSVD method without power iteration. All the results are averages based on 30 experiments unless otherwise specified. □

Example 4.4. This example uses the IR Tools package [4]. We blur an image of the Hubble space telescope included in the package (of different sizes, 70×70 and 100×100 pixels) with medium speckle blur or severe motion blur. The image sizes correspond to $n = 4900$ and $n = 10000$, respectively. Zero boundary conditions are imposed. These boundary conditions are appropriate due to the black background of the image. The blurring matrix A generated by IR Tools is a `psfMatrix` object for the blur considered. These objects have many favorable features and can be convenient to apply. However, in this example, we convert the object determined by IR Tools to a matrix A. This allows us to compute the SVD of the blurring matrix with the MATLAB function `svd`. The evaluation of the SVD of a `psfMatrix` object is not possible. We do not exploit any structure of A in these computations. The timings therefore are representative for general matrices. Moreover, matrix-block-vector products can be evaluated quite efficiently on many computers (by the use of level 3 BLAS), while `psfMatrix` objects only allow multiplication by one vector at a time.

The exact blurred Hubble space telescope image is contaminated by white Gaussian noise with noise levels $\epsilon \in \{0.01, 0.001\}$.

Table 4 demonstrates the performance of the methods TRSVD(1) and MTRSVD(1) with one step of power iteration when applied to problems of various sizes and noise levels, and compares these methods to the standard TSVD and MTSVD meth-

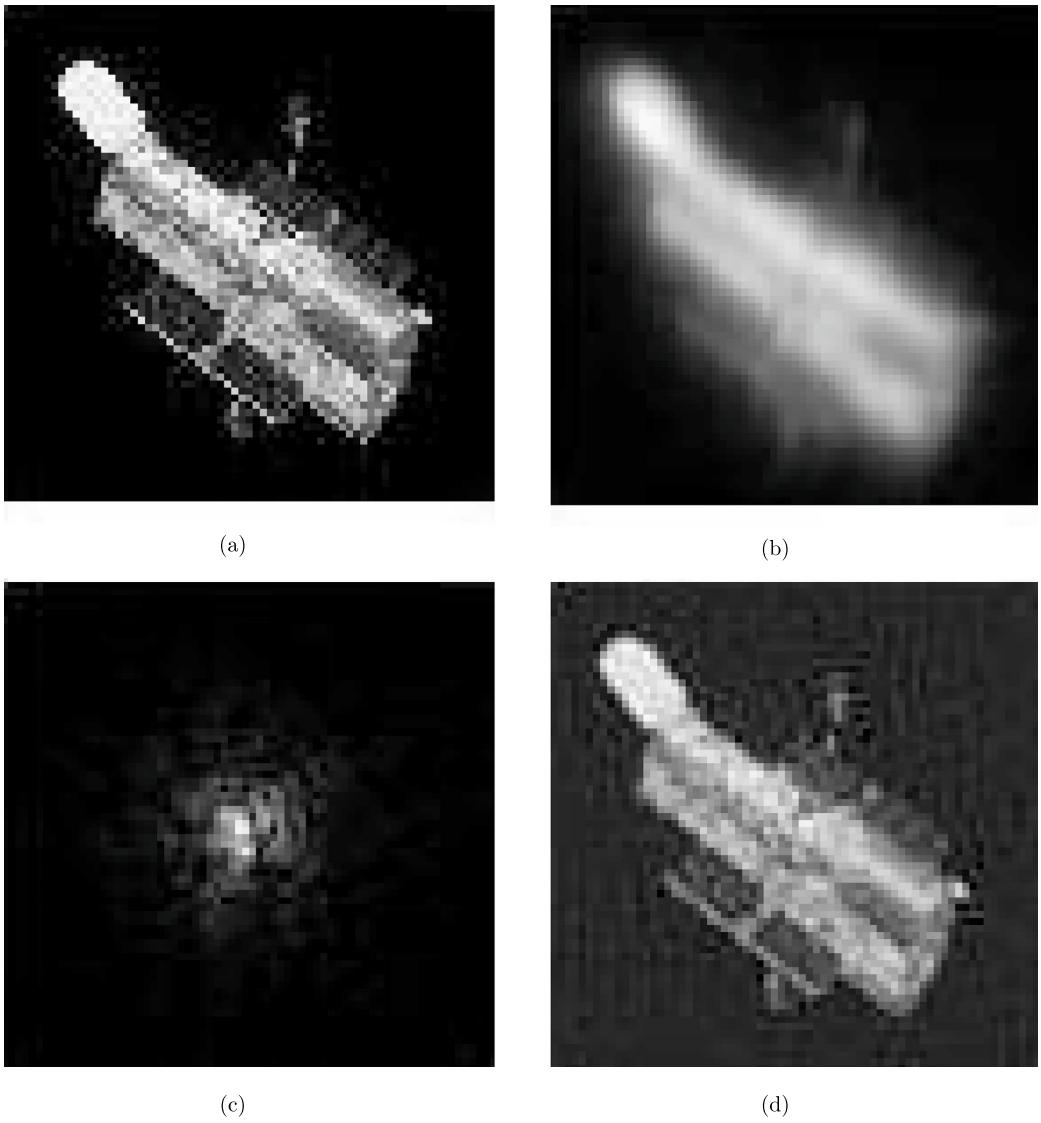

Fig. 5. Example 4.4: (a) True image (70×70 pixels, i.e., $n = 4900$), (b) blur- and noise-contaminated image (noise level $\epsilon = 0.01$), (c) speckle blur point spread function displayed on a square root scale, and (d) restored image determined by MTRSVD(1).

ods. Fig. 5 provides visual illustrations. We note that the values of the parameter ℓ used for the computations for Table 4 and Fig. 5 are quite large. This is necessary to achieve fairly accurate restorations. Smaller values of ℓ yield restorations of worse quality.

Table 5 compares the basic MTRSVD method without power iteration, denoted by MTRSVD(0), to the basic MTRSVD method complemented by j steps of power iterations. The latter schemes are denoted by MTRSVD(j). We found that at most a few steps of power iteration is necessary, and typically we only apply at most one step. This table also illustrates that power iterations are necessary when the singular values decay slowly with increasing index. The left-hand side of Table 5 shows results for a fairly small speckle blur matrix. The MTRSVD(0) method does not give as accurate restorations as TRSVD(0), because the former method uses more error-contaminated singular vectors than the latter. The right-hand side of the table shows results for motion blur. The singular values for this blur decay to zero slower with increasing index than for speckle blur. Therefore, more singular vectors are required to determine a restoration that satisfies the discrepancy principle. The MTRSVD(0) method is seen to perform much worse than TRSVD(0), since the former method uses more error contaminated singular vectors to construct the restoration.

Table 6 supplements Table 5 and illustrates that when matrix size increases, it becomes even more important to apply power iterations. Results in this table show averages of 10 experiments.

The MTRSVD method seeks to determine a partial SVD of matrix A. Fig. 6 displays the exact singular values and the approximate singular values computed by MTRSVD(1) for a speckle medium blurring matrix of size $n = 2500$ and noise level

Table 5
Example 4.4: Comparison of CPU times, relative errors, and truncation index values for TRSVD and MTRSVD with different numbers of power iterations, q. Left columns are results for speckle blur and right columns are for motion severe blur. In all experiments $n = 2500$ and $\epsilon = 0.1$.

Method	Speckle blur $n = 2500$			Motion blur $n = 2500$		
	Time	Err	k or $\widehat{k}$	Time	Err	k or $\widehat{k}$
$q = 0$, $\ell = 1500$						
TRSVD(0)	6.3056	0.3145	487	4.5657	0.3803	527.8667
MTRSVD(0)	6.3258	0.3299	1021	4.5788	0.5745	1078
$q = 1$, $\ell = 1000$						
TRSVD(1)	4.1258	0.3183	410.6333	3.0376	0.3539	290.4667
MTRSVD(1)	4.1432	0.3169	921.8333	3.046	0.3431	580.0333
$q = 2$, $\ell = 1000$						
TRSVD(2)	5.7218	0.3184	409.0333	4.0023	0.3538	290.4
MTRSVD(2)	5.7407	0.3162	940.2667	4.0107	0.3440	591.1333
$q = 3$, $\ell = 1000$						
TRSVD(3)	7.3551	0.3184	408.9667	5.2317	0.3538	290.4333
MTRSVD(3)	7.3745	0.3164	949.8667	5.2403	0.3442	593.6

Table 6
Example 4.4: Comparison of CPU times, relative errors, and truncation index values for the TRSVD(0), MTRSVD(0), TRSVD(1) and MTRSVD(1) methods applied to the restoration of images that have been contaminated by severe motion blur and noise.

Method	Motion blur $n = 4900$			Motion blur $n = 10000$		
	Time	Err	k or $\widehat{k}$	Time	Err	k or $\widehat{k}$
$q = 0$, $\ell = 2500$				$q = 0$, $\ell = 4000$		
TRSVD(0)	36.2741	0.3761	838.2	178.9941	0.4148	1560.5
MTRSVD(0)	36.3312	0.5664	1614.7	179.4288	0.6714	2865.6
$q = 1$, $\ell = 1500$				$q = 1$, $\ell = 3000$		
TRSVD(1)	22.1102	0.3416	438.5	167.3661	0.3338	738
MTRSVD(1)	22.1417	0.3296	875.4	167.623	0.3251	1483.6

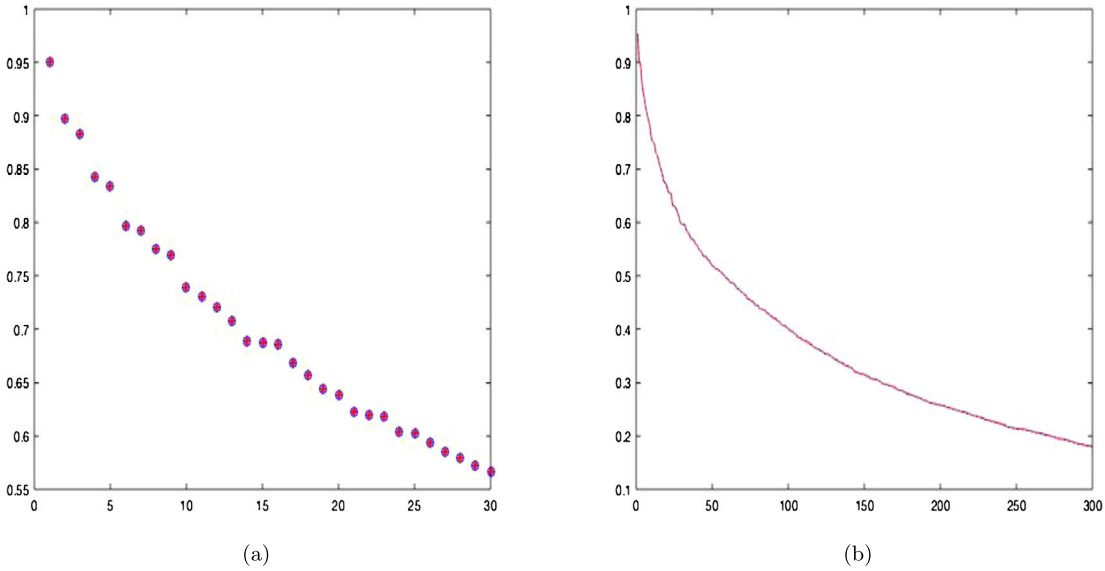

Fig. 6. Example 4.4: Exact singular values and approximate singular values computed by MTRSVD(1): (a) The first 30 singular values: The exact singular values are marked by (red) stars, while approximate ones are marked by (blue) circles. (b) The first 300 singular values: The exact singular values are on a (blue) continuous curve, while approximate ones are marked by (red) dots.

$\epsilon = 0.01$. Fig. 6(a) shows the first 30 singular values and Fig. 6(b) depicts the first 300. We can see that the approximate singular values are fairly accurate approximations. □

Table 7
Example 4.5: Comparison of CPU times, relative errors, and truncation index values for TRSVD(q) and MTRSVD(q) and $q \in \{0, 1\}$. The order of the matrix A is n. For all experiments, we let $\ell = 100$.

Method	$\epsilon = 0.1$			$\epsilon = 0.001$		
	Time	Err	k or $\widehat{k}$	Time	Err	k or $\widehat{k}$
$n = 2500$						
TRSVD(0)	0.0315	0.3869	5	0.108	0.1999	11
MTRSVD(0)	0.0324	0.3707	6	0.1094	0.1809	13
TRSVD(1)	0.1672	0.3716	6	0.2406	0.1783	12
MTRSVD(1)	0.1677	0.3708	6	0.2418	0.1731	13
$n = 10000$						
TRSVD(0)	1.5632	0.3654	6	1.7336	0.1736	13
MTRSVD(0)	1.5639	0.3654	6	1.7353	0.1727	13
TRSVD(1)	2.9248	0.3646	6	3.5356	0.1718	13
MTRSVD(1)	2.9256	0.3646	6	3.537	0.1708	13

Example 4.5. This example is constructed using the `baart` test problem from [8], which stems from the discretization of a Fredholm integral equation of the first kind in one space-dimension discussed in [1]. Let $\tilde{A} \in \mathbb{R}^{m\times m}$ be a matrix from the `baart` test problem, and define

$$A = \tilde{A} \otimes \tilde{A} \in \mathbb{R}^{m^2\times m^2},$$

where $\otimes$ denotes the Kronecker product. The singular values of A are products of the singular values of $\tilde{A}$. Since the latter singular values decay to zero rapidly with increasing index number, so do the singular values of A. In fact, A has only a few significant nonvanishing singular values, which well serves the purpose of illustrating that the basic MTRSVD method without power scheme may perform well for 2D problems when the singular values decay rapidly.

The true solution, $\widehat{x}$, is the tensor product of the exact solution of the `baart` problem with itself. The noise-free right-hand side $\widehat{b}$ equals $A\widehat{x}$. The noise-contaminated vector b in (1.1) is defined by (1.2), where e models Gaussian white noise with different noise levels $\epsilon \in \{0.1, 0.01, 0.001\}$. For all experiments in this example, we set $\ell = 100$.

Table 7 compares the performance TRSVD(q) and MTRSVD(q) for $q \in \{0, 1\}$. Letting $q = 1$ gives computed approximations of $\widehat{x}$ of higher accuracy and computing cost than $q = 0$. However, differently from the situation in Example 4.4, $q = 0$ gives quite accurate results, especially for small noise levels and large problem sizes. □

5. Conclusion

The application of truncated random singular value decomposition methods to the solution of large-scale linear discrete ill-posed problems is discussed. Several methods are described and compared. The choice of method should depend on how quickly the singular values of the problem decay to zero with increasing index. When the singular values decay slowly, application of one step of power iteration is found to be beneficial, because this enhances the accuracy of the computed approximate singular vectors. Numerical examples illustrate the performance the methods discussed.

Acknowledgements

The authors would like to thank Silvia Gazzola for helpful comments on the use of IR Tools. They also would like to thank Silvia Noschese and the referees for comments. Research by G.H. was supported in part by Application Fundamentals Foundation of STD of Sichuan (2020YJ0366) and Key Laboratory of bridge nondestructive testing and engineering calculation Open fund projects (2020QZJ03), and research by L.R. was supported in part by NSF grants DMS-1729509 and DMS-1720259, and research by F.Y. was supported in part by NNSF (11501392) and the Talent Project of SUSE (2019RC09).

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) 241–247.
[2] C. Brezinski, G. Rodriguez, S. Seatzu, Error estimates for linear systems with applications to regularization, Numer. Algorithms 49 (2008) 85–104.
[3] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.
[4] S. Gazzola, P.C. Hansen, J.G. Nagy, I.R. Tools, A MATLAB package of iterative regularization methods and large-scale test problems, Numer. Algorithms 81 (2019) 773–811.
[5] G.H. Golub, C.F. Van Loan, Matrix Computations, 4th Edition, The John Hopkins University Press, Baltimore, MD, 2013.
[6] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2011) 217–288.
[7] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems, SIAM, Philadelphia, 1998.
[8] P.C. Hansen, Regularization tools, Version 4.0 for MATLAB 7.3, Numer. Algorithms 46 (2007) 189–194.
[9] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Cambridge, 1991.

[10] Z. Jia, Y. Yang, Modified truncated randomized singular value decomposition (MTRSVD) algorithms for large scale discrete ill-posed problems with general-form regularization, Inverse Probl. 34 (2018), Art. 055013.
[11] S. Kindermann, Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems, Electron. Trans. Numer. Anal. 38 (2011) 233–257.
[12] S. Kindermann, K. Raik, A simplified L-curve method as error estimator, Electron. Trans. Numer. Anal. 53 (2020) 217–238.
[13] S. Noschese, L. Reichel, A modified truncated singular value decomposition method for discrete ill-posed problem, Numer. Linear Algebra Appl. 21 (2014) 813–822.
[14] L. Reichel, G. Rodriguez, Old and new parameter choice rules for discrete ill-posed problems, Numer. Algorithms 63 (2013) 65–87.
[15] H. Xiang, J. Zou, Regularization with randomized SVD for large-scale discrete inverse problems, Inverse Probl. 29 (2013), Art. 085008.
[16] H. Xiang, J. Zou, Randomized algorithms for large-scale inverse problems with general Tikhonov regularizations, Inverse Probl. 31 (2015), Art. 085008.