Numerical Algorithms (2020) 84:1535–1563
https://doi.org/10.1007/s11075-020-00911-y

ORIGINAL PAPER

Golub–Kahan bidiagonalization for ill-conditioned tensor equations with applications

Fatemeh P. A. Beik[1] · Khalide Jbilou[2,3] · Mehdi Najafi-Kalyani[1] · Lothar Reichel[4]

Received: 16 August 2019 / Accepted: 14 February 2020 / Published online: 2 April 2020

Abstract

This paper is concerned with the solution of severely ill-conditioned linear tensor equations. These kinds of equations may arise when discretizing partial differential equations in many space-dimensions by finite difference or spectral methods. The deblurring of color images is another application. We describe the tensor Golub–Kahan bidiagonalization (GKB) algorithm and apply it in conjunction with Tikhonov regularization. The conditioning of the Stein tensor equation is examined. These results suggest how the tensor GKB process can be used to solve general linear tensor equations. Computed examples illustrate the feasibility of the proposed algorithm.

Keywords Linear tensor operator equation · Ill-posed problem · Tikhonov regularization · Golub–Kahan bidiagonalization

Mathematics Subject Classification (2010) 65F10 · 15A24

Dedicated to Gérard Meurant on the occasion of his 70th birthday.

✉ Lothar Reichel
reichel@math.kent.edu

Fatemeh P. A. Beik
f.beik@vru.ac.ir

Khalide Jbilou
Khalide.Jbilou@univ-littoral.fr

Mehdi Najafi-Kalyani
mehdi.najafi@stu.vru.ac.ir

[1] Department of Mathematics, Vali-e-Asr University of Rafsanjan, PO Box 518, Rafsanjan, Iran

[2] Laboratory LMPA, 50 Rue F. Buisson, ULCO calais cedex, Lyon, France

[3] Laboratory CSEHS, University UM6P, Benguérir, Morocco

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

1 Introduction

This paper is concerned with the numerical solution of severely ill-conditioned tensor equations. We are particularly interested in the solution of Sylvester and Stein tensor equations. The proposed iterative schemes also can be used to solve equations of the form

$$\mathcal{L}(\mathscr{X}) = \mathscr{C}, \tag{1}$$

where $\mathcal{L} : \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N} \to \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is a linear tensor operator. Severely ill-conditioned tensor equations arise in color image restoration, video restoration, and when solving certain partial differential equations in several space-dimensions by collocation methods; see, e.g., [3, 21–24]. Throughout this work, vectors and matrices are denoted by lowercase and capital letters, respectively, and tensors of order three (or higher) are represented by Euler script letters.

Before discussing the problems to be solved, we recall the definition of an n-mode product from [19]:

Definition 1 The n-mode (matrix) product of a tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathscr{X} \times_n U$. It is of size

$$I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N,$$

and has the elements

$$(\mathscr{X} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

The n-mode (vector) product of a tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a vector $v \in \mathbb{R}^{I_n}$ is of order $N-1$ and is denoted by $\mathscr{X} \bar{\times}_n v$; its size is $I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N$.

The Sylvester and Stein tensor equations are given by

$$\mathscr{X} \times_1 A^{(1)} + \mathscr{X} \times_2 A^{(2)} + \ldots + \mathscr{X} \times_N A^{(N)} = \mathscr{D} \tag{2}$$

and

$$\mathscr{X} - \mathscr{X} \times_1 A^{(1)} \times_2 A^{(2)} \ldots \times_N A^{(N)} = \mathscr{F}, \tag{3}$$

respectively, where the right-hand side tensors $\mathscr{D}, \mathscr{F} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and the coefficient matrices $A^{(n)} \in \mathbb{R}^{I_n \times I_n}$ $(n = 1, 2, \ldots, N)$ are known, and $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the unknown tensor to be determined.

Many discretized linear partial differential equations in several space-dimensions by finite differences [2, 3, 9] or spectral methods [3, 21–23, 27] can be expressed with the aid of a Sylvester tensor equation. A discussion on the conditioning of (2) under certain conditions is provided by Najafi et al. [24], who proposed the application of Tikhonov regularization in conjunction with the global Hessenberg process in tensor form to solve (2) with a perturbed right-hand side. Some perturbation results for (3) are provided by Liang and Zheng [20] and by Xu and Wang [28], who solve (3) by using the tensor forms of the BiCG and BiCR iterative methods. Liang and Zheng [20] present perturbation results for (3) for the case when N is even and $A^{(1)} = \cdots = A^{(N)} = A$ is Schur stable, i.e., when all eigenvalues of A lie in the open unit disc in the complex plane. These results rely on the matrix spectral norm of $(I - A^{(N)} \otimes \cdots \otimes A^{(2)} \otimes A^{(1)})^{-1}$.

Recently, Huang et al. [16] proposed to apply the global form of well-known iterative methods in their tensor forms to the solution of a class of tensor equations via the Einstein product. The iterative methods in the present work are well suited to solve problems discussed in [16] when they are severely ill-conditioned; Huang et al. [16] do not consider this situation.

This paper first establishes some results on the conditioning of (3) motivated by [20, 28]. Then the tensor form of the Golub–Kahan bidiagonalization (GKB) process for the solution of severely ill-conditioned tensor equations is described. In particular, we consider the solution of severely ill-conditioned tensor equations of the forms (2) and (3). To this end, we apply results in [3] and generalize techniques described in [5]. We remark that the results discussed in Section 3 also can be applied to the solution of severely ill-conditioned problems of the form (1).

The remainder of this section introduces notation used in throughout this paper. We also recall the concept of the contracted product between two tensors. Section 2 presents some results on the sensitivity of the solution of (3), and in Section 3 we describe a tensor form of the GKB process and discuss the use of Gauss-type quadrature to determine quantities of interest for Tikhonov regularization. Section 4 presents some numerical results, and Section 5 contains concluding remarks.

1.1 Notation

Let $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an N-mode tensor, and let $x_{i_1 i_2 \ldots i_N}$ denote the element $(i_1, i_2, \ldots, i_N)$ of $\mathscr{X}$. For a real square matrix A with real eigenvalues, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ stand for its smallest and largest eigenvalues, respectively. The set of all eigenvalues of A is denoted by $\sigma(A)$. The symmetric and skew-symmetric parts of A are given by

$$\mathcal{H}(A) = \frac{1}{2}\left(A + A^T\right) \quad \text{and} \quad \mathcal{S}(A) = \frac{1}{2}\left(A - A^T\right),$$

respectively, where the superscript ^{T} denotes transposition. The condition number of an invertible matrix A is defined by

$$\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2,$$

where $\|\cdot\|_2$ stands for the spectral norm. The largest and smallest singular values of a matrix A are denoted by $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$, respectively. In particular, for an invertible matrix it holds

$$\text{cond}(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}.$$

We use the notation $\bigotimes_{i=1}^{\ell} x_i := x_1 \otimes x_2 \otimes \cdots \otimes x_\ell$ for the multi-dimensional Kronecker product. The vector $\text{vec}(\mathscr{X})$ is obtained by using the standard vectorization operator with respect to frontal slices of $\mathscr{X}$. The mode-n matricization of a tensor

$\mathscr{X}$ is denoted by $X_{(n)}$; it arranges the mode-n fibers to be the columns of the resulting matrix. Recall that a fiber of a tensor is defined by fixing all indices but one; see [19] for more details.

1.2 Contracted product

The $\boxtimes^N$ product between two N-mode tensors

$$\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times I_N} \quad \text{and} \quad \mathscr{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N-1} \times \tilde{I}_N}$$

is defined as the $I_N \times \tilde{I}_N$ matrix, whose (i, j)th entry is given by

$$[\mathscr{X} \boxtimes^N \mathscr{Y}]_{ij} = \operatorname{tr}\left(\mathscr{X}_{::\cdots:i} \boxtimes^{N-1} \mathscr{Y}_{::\cdots:j}\right), \qquad N = 3, 4, \ldots,$$

where

$$\mathscr{X} \boxtimes^2 \mathscr{Y} = \mathscr{X}^T \mathscr{Y}, \qquad \mathscr{X} \in \mathbb{R}^{I_1 \times I_2}, \mathscr{Y} \in \mathbb{R}^{I_1 \times \tilde{I}_2},$$

and $\operatorname{tr}(\cdot)$ denotes the trace of its argument. The $\boxtimes^N$ product is a special case of the contracted product [10]. Specifically, $\mathscr{X} \boxtimes^N \mathscr{Y}$ is the contracted product of the N-mode tensors $\mathscr{X}$ and $\mathscr{Y}$ along the first $N-1$ modes. For $\mathscr{X}, \mathscr{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we have

$$\langle \mathscr{X}, \mathscr{Y} \rangle = \operatorname{tr}(\mathscr{X} \boxtimes^N \mathscr{Y}), \qquad N = 2, 3, \ldots,$$

and $\|\mathscr{X}\|^2 = \operatorname{tr}(\mathscr{X} \boxtimes^N \mathscr{X}) = \mathscr{X} \boxtimes^{(N+1)} \mathscr{X}$ for $\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. We conclude this section by recalling the following two results from [3].

Lemma 1 *Let $\mathscr{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$, $A \in \mathbb{R}^{J_n \times I_n}$, and $y \in \mathbb{R}^{J_n}$. Then*

$$\mathscr{X} \times_n A \bar{\times}_n y = \mathscr{X} \bar{\times}_n (A^T y).$$

Proposition 1 *Let $\mathscr{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times m}$ be an $(N+1)$-mode tensor with the column tensors $\mathscr{B}_1, \mathscr{B}_2, \ldots, \mathscr{B}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $z = (z_1, z_2, \ldots, z_m)^T \in \mathbb{R}^m$. Then for an arbitrary $(N+1)$-mode tensor $\mathscr{A}$ with N-mode column tensors $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_m$, we have*

$$\mathscr{A} \boxtimes^{(N+1)} (\mathscr{B} \bar{\times}_{N+1} z) = (\mathscr{A} \boxtimes^{(N+1)} \mathscr{B}) z.$$

2 Sensitivity analysis of the Stein tensor equation

This section mainly discusses the conditioning of the Stein tensor (3). To this end, we first consider a linear system of equations that is equivalent to (3), and then derive lower and upper bounds for the condition number of the matrix of this linear system.

It is well-known that (2) is equivalent to the linear system of equations

$$\tilde{\mathcal{A}} x = b,$$

with $x = \operatorname{vec}(\mathscr{X})$, $b = \operatorname{vec}(\mathscr{D})$, and

$$\tilde{\mathcal{A}} = \sum_{j=1}^{N} I^{(I_N)} \otimes \cdots \otimes I^{(I_{j+1})} \otimes A^{(j)} \otimes I^{(I_{j-1})} \otimes \cdots \otimes I^{(I_1)}.$$

Moreover, we have (see [19])

$$\mathscr{Y} = \mathscr{X} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)} \quad \Leftrightarrow \quad Y_{(1)} = A^{(1)} X_{(n)} (A^{(N)} \otimes \cdots \otimes A^{(2)})^T.$$

As a result, it follows that (3) corresponds to the linear system of equations

$$\mathcal{A}x := \left(I - A^{(N)} \otimes \cdots \otimes A^{(2)} \otimes A^{(1)}\right) \mathrm{vec}(\mathscr{X}) = \mathrm{vec}(\mathscr{F}).$$

We use the tensor norm

$$\|\mathscr{X}\| = \|\mathrm{vec}(\mathscr{X})\|_2 .$$

Therefore, the sensitivity analyses of (2) and (3) are closely related to deriving bounds for the condition number of the matrices $\tilde{\mathcal{A}}$ and $\mathcal{A}$. For linear systems of equations $\mathcal{A}x = b$ and $\mathcal{A}(x + \Delta x) = b + \Delta b$ with a non-singular matrix, it is well known that

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \mathrm{cond}(\mathcal{A}) \frac{\|\Delta b\|_2}{\|b\|_2}.$$

Moreover, if $\left\|\mathcal{A}^{-1}\right\|_2 \|\Delta\mathcal{A}\|_2 < 1$, then

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \frac{\mathrm{cond}(\mathcal{A})}{1 - \mathrm{cond}(\mathcal{A}) \frac{\|\Delta\mathcal{A}\|_2}{\|\mathcal{A}\|_2}} \left\{ \frac{\|\Delta\mathcal{A}\|_2}{\|\mathcal{A}\|_2} + \frac{\|\Delta b\|_2}{\|b\|_2} \right\};$$

see, e.g., [13] for further details on perturbation analysis for linear systems of equations.

Lower and upper bounds for $\tilde{\mathcal{A}}$ have been derived in [24] under suitable conditions. Therefore, we limit our discussion to the tensor $\mathcal{A}$, which we will assume to be invertible. It is shown in [28] that

$$\mathrm{cond}(\mathcal{A}) \geq \frac{\max_{\lambda_{i_k} \in \sigma(A^{(k)}), k=1,2,\ldots,N} \left|1 - \lambda_{i_1}\lambda_{i_2} \ldots \lambda_{i_N}\right|}{\min_{\lambda_{i_k} \in \sigma(A^{(k)}), k=1,2,\ldots,N} \left|1 - \lambda_{i_1}\lambda_{i_2} \ldots \lambda_{i_N}\right|}$$

and

$$\mathrm{cond}(\mathcal{A}) \leq \frac{1 + \prod_{i=1}^N \|A^{(i)}\|_2}{1 - \prod_{i=1}^N \|A^{(i)}\|_2}, \tag{4}$$

where the latter bound requires the inequality $\|A^{(N)} \otimes \cdots \otimes A^{(2)} \otimes A^{(1)}\|_2 < 1$ to hold. The following proposition presents an alternative upper bound.

Proposition 2 *Assume that* $\prod_{i=1}^N \sigma_{\min}(A^{(i)}) > 1$. *Then*

$$\mathrm{cond}(\mathcal{A}) \leq \left(\frac{\prod_{i=1}^N \sigma_{\min}(A^{(i)})}{\prod_{i=1}^N \sigma_{\min}(A^{(i)}) - 1} \right) \left(1 + \prod_{i=1}^N \|A^{(i)}\|_2 \right).$$

Proof Define $\mathcal{F} = A^{(N)} \otimes \cdots \otimes A^{(1)}$ and let $\rho(M)$ denote the spectral radius of the matrix M. Then

$$\begin{aligned}\|\mathcal{A}\|_2 &\leq 1 + \|\mathcal{F}\|_2 = 1 + \sqrt{\rho\left(\mathcal{F}\mathcal{F}^T\right)} \\ &= 1 + \prod_{i=1}^{N} \sigma_{\max}\left(A^{(i)}\right) \\ &= 1 + \prod\nolimits_{i=1}^{N} \|A^{(i)}\|_2. \end{aligned} \tag{5}$$

Since $(I - \mathcal{F})^{-1} = -(I - \mathcal{F}^{-1})^{-1}\mathcal{F}^{-1}$ and

$$\mathcal{F}^{-1} = \left(A^{(N)}\right)^{-1} \otimes \cdots \otimes \left(A^{(1)}\right)^{-1},$$

we obtain

$$\|\mathcal{F}^{-1}\|_2 = \prod\nolimits_{i=1}^{N} \| \left(A^{(i)}\right)^{-1} \|_2 = \left(\prod\nolimits_{i=1}^{N} \sigma_{\min}\left(A^{(i)}\right)\right)^{-1} < 1$$

and

$$\|(I - \mathcal{F})^{-1}\|_2 \leq \|(I - \mathcal{F}^{-1})^{-1}\|_2 \|\mathcal{F}^{-1}\|_2 \leq \|(I - \mathcal{F}^{-1})^{-1}\|_2 \leq \frac{1}{1 - \|\mathcal{F}^{-1}\|_2},$$

which shows the proposition. □

Remark 1 We note that the assumption in Proposition 2 differs from the one used in [28]. Because of the importance of determining upper bounds in perturbation analysis, we report the upper bounds provided by (4) and Proposition 2 for two matrices $\mathcal{A}$. The bounds and the exact condition numbers are plotted in Fig. 1. We used the MATLAB function "cond(full($\mathcal{A}$))." This allowed us to calculate the condition number of $\mathcal{A}$ for small n only due to lack of computer memory.[1] When the matrix $\mathcal{A}$ is large and sparse, we can compute an estimate of the condition number with MATLAB function "condest($\mathcal{A}$)."

Case I: We let the matrices $A^{(i)}$, $i = 1, 2, 3$, be ill-conditioned "prolate" Toeplitz matrices. This kind of Toeplitz matrix can be generated with the MATLAB command $A = \text{gallery}\,('\text{prolate}', n, w)$, which returns the n-by-n prolate Toeplitz matrix with parameter w. We set $w = 0.11$ for $A^{(1)}$, $w = 0.12$ for $A^{(2)}$, and $w = 0.13$ for $A^{(3)}$. Then $\|A^{(3)} \otimes A^{(2)} \otimes A^{(1)}\| < 1$. Notice that $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are full matrices. We therefore do not report "condest($\mathcal{A}$)" for this case.

[1] All computations for this section were carried out on a 64-bit 2.50-GHz core i5 processor with 8.00-GB RAM using MATLAB version 9.4 (R2018a).

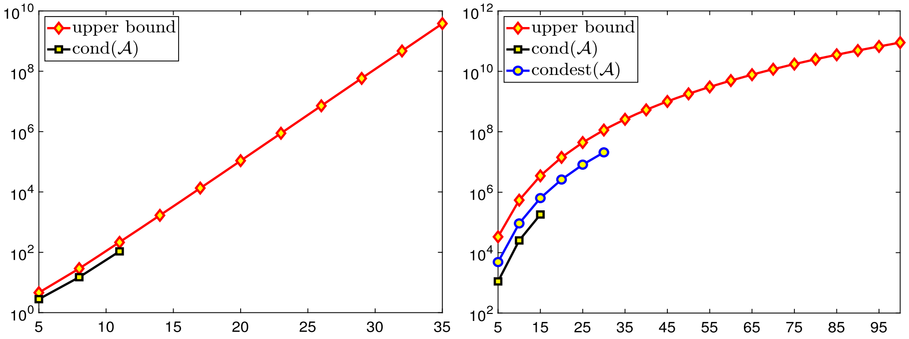


Fig. 1 Computed bounds versus the exact condition numbers for different values of n; Case I (left) and Case II (right)

Case II: For $i = 1, 2, 3$, consider the matrices

$$A^{(i)} = \frac{\nu}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} + \frac{c_i}{4h}\begin{bmatrix} 3 & -5 & 1 & & \\ 1 & 3 & -5 & 1 & \\ & \ddots & \ddots & \ddots & 1 \\ & & 1 & 3 & -5 \\ & & & 1 & 3 \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad (6)$$

that are the sum of a symmetric tridiagonal matrix and a banded upper Hessenberg Toeplitz matrix with $\nu = 0.1$, $c_1 = 1$, $c_2 = 2$, $c_3 = 3$, and $h = 1/(n+1)$. It can be verified that $\prod_{i=1}^{3} \sigma_{\min}(A^{(i)}) > 1$.

We next derive new bounds for cond($\mathcal{A}$). This requires the following two propositions.

Proposition 3 *Let $A^{(i)} \in \mathbb{R}^{n_i\times n_i}$ and $x_i \in \mathbb{R}^{n_i}$ for $i = 1, 2, \ldots, \ell$. Then*

$$\left(\bigotimes_{i=1}^{\ell} x_i\right)^T \mathcal{H}\left(A^{(1)} \otimes A^{(2)} \otimes \cdots \otimes A^{(\ell)}\right) \bigotimes_{i=1}^{\ell} x_i = \prod_{i=1}^{\ell} x_i^T \mathcal{H}(A^{(i)}) x_i. \qquad (7)$$

Proof We show the assertion by induction. Let $\ell = 2$. Using the fact that $x_i^T \mathcal{S}(A^{(i)}) x_i = 0$ for $i = 1, 2$, we obtain (7) from the following equality (see [29]):

$$\mathcal{H}(A^{(1)} \otimes A^{(2)}) = \mathcal{H}(A^{(1)}) \otimes \mathcal{H}(A^{(2)}) + \mathcal{S}(A^{(1)}) \otimes \mathcal{S}(A^{(2)}).$$

Now assume that (7) holds for $\ell = k$. Let $\ell = k + 1$ and define

$$\mathcal{Y}_k = \bigotimes_{i=2}^{(k+1)} x_i, \qquad \mathcal{Y}_{k+1} = x_1 \otimes \mathcal{Y}_k, \qquad \mathcal{A}_k = A^{(2)} \otimes \cdots \otimes A^{(k+1)}.$$

Then

$$\begin{aligned}\mathcal{Y}_{k+1}^T\mathcal{H}(A^{(1)}\otimes A^{(2)}\otimes\cdots\otimes A^{(k+1)})\mathcal{Y}_{k+1} &= (x_1\otimes\mathcal{Y}_k)^T\mathcal{H}(A^{(1)}\otimes\mathcal{A}_k)(x_1\otimes\mathcal{Y}_k)\\ &= (x_1^T\mathcal{H}(A^{(1)})x_1)\times(\mathcal{Y}_k^T\mathcal{H}(\mathcal{A}_k)\mathcal{Y}_k).\end{aligned}$$

The proposition now follows from the induction hypothesis. □

Proposition 4 *Let $\mathcal{A} = I - A^{(N)}\otimes\cdots\otimes A^{(2)}\otimes A^{(1)}$. Then*

$$\lambda_{\max}(\mathcal{A}\mathcal{A}^T) \geq 1 + \prod\nolimits_{i=1}^{N}\sigma_{\max}^2(A^{(i)}) - 2\prod\nolimits_{i=1}^{N} y_i^T\mathcal{H}(A^{(i)})y_i$$

and

$$\lambda_{\min}(\mathcal{A}\mathcal{A}^T) \leq 1 + \prod\nolimits_{i=1}^{N}\sigma_{\min}^2(A^{(i)}) - 2\prod\nolimits_{i=1}^{N} z_i^T\mathcal{H}(A^{(i)})z_i,$$

where the y_i and z_i are unit eigenvectors such that, for $i = 1, 2, \ldots, N$,

$$A^{(i)}(A^{(i)})^T z_i = \sigma_{\min}^2(A^{(i)})z_i \quad \textit{and} \quad A^{(i)}(A^{(i)})^T y_i = \sigma_{\max}^2(A^{(i)})y_i.$$

Proof It is easy to verify that

$$\begin{aligned}\mathcal{A}\mathcal{A}^T &= (I - A_N\otimes\cdots\otimes A_1)(I - A_N^T\otimes\cdots\otimes A_1^T)\\ &= I + A_NA_N^T\otimes\cdots\otimes A_1A_1^T - 2\mathcal{H}(A_N\otimes\cdots\otimes A_1).\end{aligned}\tag{8}$$

Let $\mathcal{Y} = (y_N\otimes\cdots\otimes y_1)$ and $\mathcal{Z} = (z_N\otimes\cdots\otimes z_1)$. Then it follows from Proposition 3 that

$$\mathcal{Y}^T\mathcal{A}\mathcal{A}^T\mathcal{Y} = 1 + \prod\nolimits_{i=1}^{N}\sigma_{\max}^2(A^{(i)}) - 2\prod\nolimits_{i=1}^{N} y_i^T\mathcal{H}(A^{(i)})y_i$$

and

$$\mathcal{Z}^T\mathcal{A}\mathcal{A}^T\mathcal{Z} = 1 + \prod\nolimits_{i=1}^{N}\sigma_{\min}^2(A^{(i)}) - 2\prod\nolimits_{i=1}^{N} z_i^T\mathcal{H}(A^{(i)})z_i.$$

This shows the desired result. □

Remark 2 If the matrices $A^{(i)}$, for $i = 1, 2, \ldots, N$, are positive definite, then

$$\lambda_{\min}(\mathcal{A}\mathcal{A}^T) \leq 1 + \prod\nolimits_{i=1}^{N}\sigma_{\min}^2(A^{(i)}).$$

We note that the matrices $A^{(i)}$ are not required to be symmetric. Positive definiteness of the matrix $A^{(i)}$, $i = 1, 2, \ldots, N$, implies that $\mathcal{H}(A^{(i)})$ is symmetric positive definite. Furthermore, if

$$\prod\nolimits_{i=1}^{N}\sigma_{\max}^2(A^{(i)}) \geq 2\prod\nolimits_{i=1}^{N}\lambda_{\max}(\mathcal{H}(A^{(i)})),$$

then the following upper bound follows from Proposition 4,

$$\begin{aligned}\mathrm{cond}(\mathcal{A}) &\geq \frac{\sqrt{1 + \prod_{i=1}^{N}\sigma_{\max}^2(A^{(i)}) - 2\prod_{i=1}^{N}\lambda_{\max}(\mathcal{H}(A^{(i)}))}}{\sqrt{1 + \prod_{i=1}^{N}\sigma_{\min}^2(A^{(i)})}}\\ &\geq \frac{1}{\sqrt{1 + \prod_{i=1}^{N}\sigma_{\min}^2(A^{(i)})}}.\end{aligned}$$

Under additional assumptions, we can derive an alternative upper bound for the condition number. To this end, we need the following result, which is a consequence of Weyl's Theorem [14, Theorem 4.3.1].

Proposition 5 *Let the matrices $A, B \in \mathbb{R}^{n\times n}$ be symmetric. Then*

$$\begin{aligned}\lambda_{\max}(A+B) &\le \lambda_{\max}(A)+\lambda_{\max}(B),\\ \lambda_{\min}(A+B) &\ge \lambda_{\min}(A)+\lambda_{\min}(B).\end{aligned}$$

Remark 3 Let $\mathcal{F} = A^{(N)} \otimes \cdots \otimes A^{(1)}$ and $\lambda \in \sigma(\mathcal{H}(\mathcal{F}))$. Let $\mathcal{E}_N$ denote the set of non-negative even numbers less than or equal to N. Then

$$\lambda_{max}(\mathcal{H}(\mathcal{F})) \le \sum_{r\in\mathcal{E}_N} \frac{N!}{r!\,(N-r)!} M_S^r M_H^{N-r} \le (M_S+M_H)^N,$$

where

$$M_S = \max_{i=1,2,\ldots,N} \|\mathcal{S}(A^{(i)})\|_2 \quad \text{and} \quad M_H = \max_{i=1,2,\ldots,N} \|\mathcal{H}(A^{(i)})\|_2.$$

The result can be shown by considering the symmetric part of $\mathcal{F}$. For simplicity, let $N=3$. Then

$$\begin{aligned}\mathcal{H}(\mathcal{F}) = {} & \mathcal{H}(A^{(3)})\otimes\mathcal{H}(A^{(2)})\otimes\mathcal{H}(A^{(1)}) + \mathcal{H}(A^{(3)})\otimes\mathcal{S}(A^{(2)})\otimes\mathcal{S}(A^{(1)})\\ & +\mathcal{S}(A^{(3)})\otimes\mathcal{H}(A^{(2)})\otimes\mathcal{S}(A^{(1)}) + \mathcal{S}(A^{(3)})\otimes\mathcal{S}(A^{(2)})\otimes\mathcal{H}(A^{(1)}).\end{aligned}$$

Using Proposition 5, we have

$$\begin{aligned}\lambda_{max}(\mathcal{H}(\mathcal{F})) &\le \prod_{i=1}^{3} \|\mathcal{H}(A^{(i)})\|_2 + \|\mathcal{H}(A^{(3)})\|_2\|\mathcal{S}(A^{(2)})\|_2\|\mathcal{S}(A^{(1)})\|_2\\ &\quad +\|\mathcal{S}(A^{(3)})\|_2\|\mathcal{H}(A^{(2)})\|_2\|\mathcal{S}(A^{(1)})\|_2\\ &\quad +\|\mathcal{S}(A^{(3)})\|_2\|\mathcal{S}(A^{(2)})\|_2\|\mathcal{H}(A^{(1)})\|_2\\ &\le M_H^3 + 3M_H M_S^2 \le (M_H+M_S)^3.\end{aligned}$$

It follows from the above discussions that if $1+\prod_{i=1}^{N}\sigma_{\min}^2(A^{(i)}) - 2(M_S+M_H)^N > 0$, then we can derive an upper bound for $\|\mathcal{A}^{-1}\|_2$ in the following manner: We obtain from (8) that

$$\begin{aligned}\lambda_{\min}(\mathcal{A}\mathcal{A}^T) &\ge 1+\prod_{i=1}^{N}\sigma_{\min}^2(A^{(i)}) - 2\lambda_{max}(\mathcal{H}(\mathcal{F}))\\ &\ge 1+\prod_{i=1}^{N}\sigma_{\min}^2(A^{(i)}) - 2(M_S+M_H)^N.\end{aligned}$$

Therefore,

$$\|\mathcal{A}^{-1}\|_2 \leq \frac{1}{\sqrt{1 + \prod_{i=1}^{N} \sigma_{\min}^2(A^{(i)}) - 2(M_S + M_H)^N}}. \tag{9}$$

Combining the inequalities (5) and (9) yields

$$\operatorname{cond}(\mathcal{A}) \leq \frac{1 + \prod_{i=1}^{N} \sigma_{\max}(A^{(i)})}{\sqrt{1 + \prod_{i=1}^{N} \sigma_{\min}^2(A^{(i)}) - 2(M_S + M_H)^N}}. \tag{10}$$

To illustrate the bound (10), we let $A^{(1)} = A^{(2)} = A^{(3)} = \tilde{A}$, where the matrix $\tilde{A} \in \mathbb{R}^{n \times n}$ is defined by

$$\tilde{A} = M + 2rL + \frac{100}{(n+1)^2} I$$

with $M = \operatorname{tridiag}(-1, 2, -1)$ and $L = \operatorname{tridiag}(0.5, 0, -0.5)$. We note that the matrix $\tilde{A}$ is taken from [29–31]. The condition

$$1 + \prod_{i=1}^{3} \sigma_{\min}^2(A^{(i)}) - 2(M_S + M_H)^3 > 0$$

holds for suitable choices of r and even values of n. Figure 2 displays graphs for the exact condition number $\operatorname{cond}(\mathcal{A})$ and the bound (10). The computations are carried out on the same computer as for Fig. 1. In particular, the function $\operatorname{cond}(\cdot)$ can be evaluated for fairly small values of n, only.

We conclude this section by considering the situation when all the matrices $A^{(i)}$ are diagonalizable.

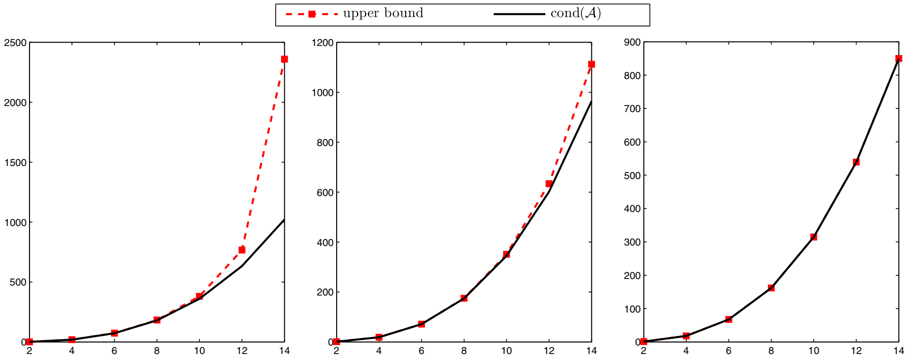


Fig. 2 Computed bounds by (10) versus the exact condition numbers for different values of n; $r = 70$ (left), $r = 100$ (center), and $r = 500$ (right)

Remark 4 Let the matrices $A^{(i)}$ be diagonalizable, i.e., there are non-singular matrices S_i and diagonal matrices D_i such that $A^{(i)} = S_i D_i S_i^{-1}$ for $i = 1, 2, \ldots, N$. Introduce

$$\mathcal{A} = I - A^{(N)} \otimes \cdots \otimes A^{(1)} \quad \text{and} \quad \mathcal{S} = S_N \otimes \cdots \otimes S_1.$$

Then $\mathcal{A} = \mathcal{S}(I - D_N \otimes \cdots \otimes D_1)\mathcal{S}^{-1}$. Hence, if $1 \notin \sigma(A^{(N)} \otimes \cdots \otimes A^{(1)})$, then

$$\mathcal{A}^{-1} = \mathcal{S}(I - D_N \otimes \cdots \otimes D_1)^{-1}\mathcal{S}^{-1}.$$

As a result, we get

$$\|\mathcal{A}^{-1}\|_2 \le \prod_{i=1}^{N} \|S_i^{-1}\|_2 \|S_i\|_2 M_{1,D} = \prod_{i=1}^{N} \operatorname{cond}(S_i) M_{1,D},$$

$$\|\mathcal{A}\|_2 \le \prod_{i=1}^{N} \|S_i^{-1}\|_2 \|S_i\|_2 M_{2,D} = \prod_{i=1}^{N} \operatorname{cond}(S_i) M_{2,D},$$

where

$$M_{1,D} = \max\left\{ \frac{1}{|1 - \lambda_{\min}(D_N \otimes \cdots \otimes D_1)|}, \frac{1}{|1 - \lambda_{\max}(D_N \otimes \cdots \otimes D_1)|} \right\},$$

$$M_{2,D} = \max\left\{|1-\lambda_{\min}(D_N \otimes \cdots \otimes D_1)|, |1-\lambda_{\max}(D_N \otimes \cdots \otimes D_1)|\right\} \le 1 + \prod_{i=1}^{N} \|D_i\|_2.$$

We obtain the inequality

$$\operatorname{cond}(\mathcal{A}) \le \prod_{i=1}^{N} (\operatorname{cond}(S_i))^2 \, M_{1,D} M_{2,D}.$$

Let $\prod_{i=1}^{N} \|D_i^{-1}\|_2 < 1$. Then analogously to the proof of Proposition 2, we have

$$\operatorname{cond}(\mathcal{A}) \le \prod_{i=1}^{N} (\operatorname{cond}(S_i))^2 \, \frac{M_{2,D}}{1 - \prod_{i=1}^{N} \|D_i^{-1}\|_2}.$$

If $\prod_{i=1}^{N} \|D_i\|_2 < 1$, then

$$\operatorname{cond}(\mathcal{A}) \le \prod_{i=1}^{N} (\operatorname{cond}(S_i))^2 \, \frac{M_{2,D}}{1 - \prod_{i=1}^{N} \|D_i\|_2} \le \prod_{i=1}^{N} (\operatorname{cond}(S_i))^2 \, \frac{2}{1 - \prod_{i=1}^{N} \|D_i\|_2}.$$

Finally, we note that if the matrices D_i, $i = 1, 2, \ldots, N$, are all positive definite, then

$$\lambda_{\min}(D_N \otimes \cdots \otimes D_1) = \prod_{i=1}^{N} \lambda_{\min}(D_i) \quad \text{and} \quad \lambda_{\max}(D_N \otimes \cdots \otimes D_1) = \prod_{i=1}^{N} \lambda_{\max}(D_i).$$

3 The tensor form of GKB and Tikhonov regularization

We first describe the implementation of the Golub–Kahan bidiagonalization (GKB) process in the tensor framework. Subsequently, we discuss an application of the GKB process to Tikhonov regularization. For notational simplicity, we introduce the two linear operators $\tilde{\mathcal{M}}, \mathcal{M} : \mathbb{R}^{I_1\times I_2\times\cdots\times I_N} \to \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ defined by

$$\begin{aligned}\tilde{\mathcal{M}}(\mathscr{X}) &:= \mathscr{X} \times_1 A^{(1)} + \mathscr{X} \times_2 A^{(2)} + \cdots + \mathscr{X} \times_N A^{(N)},\\ \mathcal{M}(\mathscr{X}) &:= \mathscr{X} - \mathscr{X} \times_1 A^{(1)} \times_2 A^{(2)} \ldots \times_N A^{(N)}.\end{aligned}$$

The adjoint operators of $\tilde{\mathcal{M}}$ and $\mathcal{M}$ are given by

$$\begin{aligned}\tilde{\mathcal{M}}^*(\mathscr{Y}) &:= \mathscr{Y} \times_1 (A^{(1)})^T + \mathscr{Y} \times_2 (A^{(2)})^T + \cdots + \mathscr{Y} \times_N (A^{(N)})^T,\\ \mathcal{M}^*(\mathscr{Y}) &:= \mathscr{Y} - \mathscr{Y} \times_1 (A^{(1)})^T \times_2 (A^{(2)})^T \ldots \times_N (A^{(N)})^T,\end{aligned}$$

for $\mathscr{Y} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. The tensor equations (2) and (3) can be expressed as

$$\begin{aligned}\tilde{\mathcal{M}}(\mathscr{X}) &= \mathscr{D},\\ \mathcal{M}(\mathscr{X}) &= \mathscr{F}.\end{aligned} \tag{11}$$

We remark that the results and methods of this section also can be applied to other linear operators from $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ to $\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. For notational convenience, we discuss in the sequel results and methods for (11).

Consider for the moment the linear system of equations $Ax = b$ with a nonsingular matrix $A \in \mathbb{R}^{n\times n}$. Application of k steps of the GKB process to A with initial vector b produces the decompositions

$$AU_k = V_{k+1}\bar{T}_k, \qquad A^T V_k = U_k T_k^T, \tag{12}$$

where the matrices $V_{k+1} \in \mathbb{R}^{n\times(k+1)}$ and $U_k \in \mathbb{R}^{n\times k}$ have orthonormal columns, the matrix V_k is made up of the first k columns of V_{k+1}, the first column of V_{k+1} is $b/\|b\|_2$, the matrix $\bar{T}_k \in \mathbb{R}^{(k+1)\times k}$ is lower bidiagonal with all diagonal and subdiagonal entries positive, and T_k is the leading $k \times k$ submatrix of $\bar{T}_k$. We assume that k is small enough so that the decompositions (12) with the stated properties exist. This is the generic situation. Otherwise, the GKB process is said to break down. In the latter event, the computations simplify. We will not dwell on the handling of breakdowns. Thorough discussions on the GKB process can be found in [13, 25].

It is natural to extend the GKB process to tensor equations. Algorithm 1 describes the application of the GKB process to (11). We refer to the process so defined as the GKB based on tensor format (GKB_BTF) process.

Algorithm 1 The GKB_BTF process associated with the linear operator $\mathcal{M}$.

```
1  **Input:** Linear operator $\mathcal{M}$, the right-hand side $\mathscr{F}$, number of steps $k$.
2  Set $\beta_1 = \|\mathscr{F}\|$, $\mathscr{V}_1 = \frac{1}{\beta_1}\mathscr{F}$, and $\mathscr{U}_0 = 0$.
3  begin
4      for $j = 1, 2, \ldots, k$ do
5          $\mathscr{U} = \mathcal{M}^*(\mathscr{V}_j) - \beta_j \mathscr{U}_{j-1}$;
6          $\alpha_j = \|\mathscr{U}\|$;
7          if $\alpha_j = 0$ then
8              Stop
9          end
10         $\mathscr{U}_j = \mathscr{U}/\alpha_j$;
11         $\mathscr{V} = \mathcal{M}(\mathscr{U}_j) - \alpha_j \mathscr{V}_j$;
12         $\beta_{j+1} = \|\mathscr{V}\|$;
13         if $\beta_{j+1} = 0$ then
14             Stop
15         end
16         $\mathscr{V}_{j+1} = \mathscr{V}/\beta_{j+1}$ ;
17     end
18 end
```

Assume that the first k steps of Algorithm 1 can be carried out without breakdown, i.e., without any coefficients α_j and β_j vanishing. The analogue of the lower bidiagonal matrix $\bar{T}_k \in \mathbb{R}^{(k+1)\times k}$ in (12), which we also refer to as $\bar{T}_k$, has the diagonal entries $\alpha_1, \alpha_2, \ldots, \alpha_k$. They are computed in line 6 of Algorithm 1. The subdiagonal elements $\beta_2, \beta_3, \ldots, \beta_{k+1}$ of $\bar{T}_k$ are computed in line 12 of the algorithm. We can express the matrix $\bar{T}_k$ in the form

$$\bar{T}_k = \begin{pmatrix} T_k \\ \beta_{k+1} e_k^T \end{pmatrix}.$$

Theorem 1 *Let $\tilde{\mathscr{V}}_k$, $\tilde{\mathscr{U}}_k$, $\tilde{\mathscr{W}}_k$, and $\tilde{\mathscr{W}}_k^*$ be $(N+1)$-mode tensors with frontal slices $\mathscr{V}_j$, $\mathscr{U}_j$, $\mathscr{W}_j := \mathcal{M}(\mathscr{U}_j)$, and $\mathscr{W}_j^* := \mathcal{M}^*(\mathscr{V}_j)$, respectively, for $j = 1, 2, \ldots, k$, computed by Algorithm 1. Then*

$$\tilde{\mathscr{W}}_k = \tilde{\mathscr{V}}_k \times_{(N+1)} T_k^T + \beta_{k+1} \mathscr{Z} \times_{(N+1)} E_k = \tilde{\mathscr{V}}_{k+1} \times_{(N+1)} \bar{T}_k^T, \tag{13}$$

$$\tilde{\mathscr{W}}_k^* = \tilde{\mathscr{U}}_k \times_{(N+1)} T_k, \tag{14}$$

where $\mathscr{Z}$ is an $(N+1)$-mode tensor with k column tensors $0, \ldots, 0, \mathscr{V}_{k+1}$. The last column of the matrix $E_k = [0, \ldots, 0, e_k] \in \mathbb{R}^{k\times k}$ is the last column of the identity matrix of order k.

Proof From lines 11 and 16 of Algorithm 1, we have

$$\mathcal{M}(\mathscr{U}_{j-1}) = \alpha_{j-1}\mathscr{V}_{j-1} + \beta_j \mathscr{V}_j. \tag{15}$$

Note that the $(j-1)$st frontal slice of (13) is given by

$$(\tilde{\mathscr{V}}_{k+1} \times_{(N+1)} \bar{T}_k^T)_{:,\ldots,:(j-1)} = \sum_{\ell=1}^{k+1} \mathscr{V}_\ell (\bar{T}_k)_{\ell,j-1} = \alpha_{j-1}\mathscr{V}_{j-1} + \beta_j \mathscr{V}_j .$$

Equation (13) now follows from (15) and the above relation.

To show (14), we first note that lines 2, 5, and 10 of Algorithm 1 yield

$$\mathcal{M}^*(\mathscr{V}_j) = \beta_j \mathscr{U}_{j-1} + \alpha_j \mathscr{U}_j, \quad j = 1, 2, \ldots ,$$

where $\mathscr{U}_0$ is defined to be zero. Equation (14) now follows by comparing the above equation and the jth frontal slice of the right-hand side of (14). □

We turn to the situation when the operator $\mathcal{M}$ in (11) is severely ill-conditioned and the right-hand side tensor $\mathscr{F}$ is contaminated by error. Let $\hat{\mathscr{F}}$ denote the unknown error-free tensor associated with $\mathscr{F}$, and assume that $\hat{\mathscr{F}}$ is in the range of $\mathcal{M}$. We would like to determine the solution of minimal norm, $\hat{\mathscr{X}}$, of

$$\mathcal{M}(\mathscr{X}) = \hat{\mathscr{F}}.$$

Straightforward solution of (11) may not give a meaningful approximation of $\hat{\mathscr{X}}$ due to a large propagated error in the solution of (11) stemming from the error in $\mathscr{F}$. A common way to address this difficulty is to replace (11) by a nearby problem, whose solution is less sensitive to the error in $\mathscr{F}$. This replacement is known as regularization. One of the most popular regularization methods is due to Tikhonov. This regularization method replaces the solution of (11) by the minimization problem

$$\min_{\mathscr{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}} \left\{ \|\mathcal{M}(\mathscr{X}) - \mathscr{F}\|^2 + \mu \|\mathscr{X}\|^2 \right\}. \tag{16}$$

The parameter $\mu > 0$ is referred to as a regularization parameter. Its purpose is to balance the influence of the first term (the fidelity term) and the second term (the regularization term).

Let $\mathscr{X}_{k,\mu_k} = \tilde{\mathscr{U}}_k \bar{\times}_{(N+1)} y_{k,\mu_k}$ be an approximate solution of (16), where $\tilde{\mathscr{V}}_k$ is defined as above. We obtain from (13), by using Lemma 1 and Proposition 1, that

$$\begin{aligned} \left\|\mathcal{M}(\mathscr{X}_{k,\mu_k}) - \mathscr{F}\right\| &= \left\| \tilde{\mathscr{V}}_{k+1} \times_{(N+1)} \bar{T}_k^T \bar{\times}_{(N+1)} y_{k,\mu_k} - \mathscr{F} \right\| \\ &= \left\| \tilde{\mathscr{V}}_{k+1} \bar{\times}_{(N+1)} \bar{T}_k y_{k,\mu_k} - \mathscr{F} \right\| \\ &= \left\| \tilde{\mathscr{V}}_{k+1} \boxtimes^{(N+1)} (\tilde{\mathscr{V}}_{k+1} \bar{\times}_{(N+1)} \bar{T}_k y_{k,\mu_k} - \mathscr{F}) \right\|_2 \\ &= \left\| (\tilde{\mathscr{V}}_{k+1} \boxtimes^{(N+1)} \tilde{\mathscr{V}}_{k+1}) \bar{T}_k y_{k,\mu_k} - \tilde{\mathscr{V}}_{k+1} \boxtimes^{(N+1)} \mathscr{F} \right\|_2 \\ &= \left\| \bar{T}_k y_{k,\mu_k} - \beta_1 e_1 \right\|_2 . \end{aligned} \tag{17}$$

This shows that (16) is equivalent to the following low-dimensional minimization problem

$$\min_{y \in \mathbb{R}^k} \left\{ \|\bar{T}_k y - \beta_1 e_1\|_2^2 + \mu \|y\|_2^2 \right\} = \min_{y \in \mathbb{R}^k} \left\| \begin{pmatrix} \bar{T}_k \\ \sqrt{\mu} I \end{pmatrix} y - \beta_1 e_1 \right\|_2^2 . \tag{18}$$

The minimization problem on the right-hand side can be solved in only $\mathcal{O}(k)$ arithmetic floating point operations for each value of $\mu > 0$; see Eldén [11] for details.

We turn to the choice of the regularization parameter and assume that an upper bound $\epsilon > 0$ for the norm of the error in the right-hand side $\mathscr{F}$ is explicitly known. Then the discrepancy principle can be applied to determine the regularization parameter μ. The discrepancy principle prescribes that $\mu > 0$ be chosen so that

$$\left\|\mathcal{M}(\mathscr{X}_{k,\mu_k}) - \mathscr{F}\right\| = \eta\epsilon \tag{19}$$

for some parameter $\eta > 1$ that is independent of ϵ. This is a non-linear equation for $\mu > 0$; see, e.g., Engl et al. [12] for a discussion on the discrepancy principle. Of course, other techniques for determining a suitable value of μ also can be applied; see, e.g., Kindermann and Raik [17, 18] for discussions.

It is not advisable to use the normal equations associated with the right-hand side of (18) in computations. However, the normal equations are convenient to apply when deriving expressions for determining a value of $\mu > 0$ that approximately satisfies (19). Let $y_{k,\mu}$ denote the solution of (18). Using the normal equations associated with the right-hand side of (18), $y_{k,\mu}$ can be expressed as

$$y_{k,\mu} = \beta_1(\bar{T}_k^T\bar{T}_k + \mu I)^{-1}\bar{T}_k^T e_1. \tag{20}$$

Consequently,

$$\begin{aligned}\left\|\bar{T}_k y_{k,\mu} - \beta_1 e_1\right\|_2^2 &= \left\|\beta_1\bar{T}_k(\bar{T}_k^T\bar{T}_k + \mu I)^{-1}\bar{T}_k^T e_1 - \beta_1 e_1\right\|_2^2\\ &= \left\|(\bar{T}_k(\bar{T}_k^T\bar{T}_k + \mu I)^{-1}\bar{T}_k^T - I)\beta_1 e_1\right\|_2^2\\ &= \left\|(\mu^{-1}\bar{T}_k\bar{T}_k^T + I)^{-1}\beta_1 e_1\right\|_2^2\\ &= \beta_1^2 e_1^T(\mu^{-1}\bar{T}_k\bar{T}_k^T + I)^{-2}e_1.\end{aligned}$$

Introduce the functions

$$\begin{aligned}\psi_k(\mu) &= \beta_1^2 e_1^T(\mu^{-1}\bar{T}_k\bar{T}_k^T + I)^{-2}e_1,\\ \phi_k(\mu) &= \beta_1^2 e_1^T(\mu^{-1}T_kT_k^T + I)^{-2}e_1.\end{aligned} \tag{21}$$

Proposition 6 *Let $\eta > 1$ and $\epsilon > 0$ be constants, and let the function $\phi_k(\mu)$ be defined by (21). If $\mu > 0$ satisfies*

$$\epsilon^2 \leq \phi_k(\mu) \leq \eta^2\epsilon^2, \tag{22}$$

then the associated solution $y_{k,\mu}$ of (18) is such that

$$\epsilon \leq \left\|T_k y_{k,\mu} - \beta_1 e_1\right\|_2 \leq \eta\epsilon,$$

and $\mathscr{X}_{k,\mu} = \tilde{\mathscr{V}}_k \bar{\times}_{(N+1)} y_{k,\mu}$ fulfills

$$\epsilon \leq \left\|\mathcal{M}(\mathscr{X}_{k,\mu}) - \mathscr{F}\right\| \leq \eta\epsilon. \tag{23}$$

Moreover,

$$\psi_k(\mu) = \left\|\mathcal{M}(\mathscr{X}_{k,\mu}) - \mathscr{F}\right\|^2. \tag{24}$$

Proof It can be shown that $\phi_k(\mu) \leq \psi_k(\mu)$ for $\mu \geq 0$. A proof based on interpreting $\phi_k(\mu)$ as a Gauss quadrature rule and $\psi_k(\mu)$ as a Gauss–Radau quadrature rule with a fixed node at the origin is provided in [8] in the context of solving large linear systems of equations with a severely ill-conditioned matrix and an error-contaminated right-hand side. Equation (24) follows from (17). □

The following result is easy to show. A proof can be found in [8].

Proposition 7 *Let $\phi_k(\mu)$ be defined by (21). Then the function $\mu \to \phi_k(1/\mu)$ is strictly decreasing and convex for $\mu > 0$. Moreover,*

$$\lim_{\mu \to \infty} \phi_k(1/\mu) = \beta_1^2.$$

In particular, Newton's method applied to compute the solution μ_k of the equation

$$\phi_k(1/\mu) = \eta^2 \epsilon^2 \tag{25}$$

with an initial approximate solution $\mu_0 \geq 0$ to the left of the solution converges monotonically and quadratically. For instance, one may choose $\mu_0 = 0$ when the function $\mu \to \phi_k(1/\mu)$ and its derivative are suitably defined at $\mu = 0$.

It follows from Proposition 7 that the use of the Newton method to solve (25) is easy to implement, because the method does not have to be safe-guarded when starting with $\mu_0 = 0$. This is discussed in [8]. However, a cubically convergent zero-finder described in [26] and applied in [7, 26] requires fewer iterations and less CPU-time.

The most expensive part of the computations with Algorithm 1 is the evaluation of $\mathcal{M}^*(\mathscr{V}_j)$ and $\mathcal{M}(\mathscr{U}_j)$ in lines 5 and 11 of the algorithm. With the aim of keeping the computational effort required by Algorithm 1 as small as possible, we would like to choose the number of steps, k, of the algorithm small, but large enough to be able to satisfy (23). To achieve this, we proceed as follows: Carry out a few steps $k > 0$ with Algorithm 1, say $k = 2$, and compute the solution $\mu_k > 0$ of $\phi_k(1/\mu) = \epsilon^2$. If $\psi_k(1/\mu_k) \leq \eta^2 \epsilon^2$, then (23) holds for

$$\mathscr{X}_{k,\mu_k} = \tilde{\mathscr{U}}_k \bar{\times}_{(N+1)} y_{k,\mu_k}, \tag{26}$$

where y_{k,μ_k} is defined by (20) with $\mu = \mu_k$. If, instead, $\psi_k(1/\mu_k) > \eta^2 \epsilon^2$, then we increase k by one, i.e., we set $k = k + 1$ and carry out one more step with Algorithm 1. We increase the number of steps until (23) holds. Typically, only a few steps of Algorithm 1 are required to satisfy (23). The required number of evaluations of the expressions $\mathcal{M}^*(\mathscr{V}_j)$ and $\mathcal{M}(\mathscr{U}_j)$ typically is fairly small. This is illustrated in Section 4. Algorithm 2 summarizes the computations required for Tikhonov regularization based on the GKB_BTF process.

Algorithm 2 The GKB_BTF-Tikhonov method for the solution of (16).

```
1 Input: Linear operator $\mathcal{M}$, right-hand side $\mathscr{F}$, bound $\epsilon > 0$ for the error in $\mathscr{F}$.
2 Choose $\eta > 1$ and set $k = 2$;
3 Set $\mathscr{V}_1 = \mathscr{F}/\|\mathscr{F}\|$;
4 Determine the orthonormal bases $\{\mathscr{U}_j\}_{j=1}^{k}$ and $\{\mathscr{V}_j\}_{j=1}^{k+1}$ of tensors, and the
  bidiagonal matrices $T_k$ and $\bar{T}_k$ with Algorithm 1;
5 Determine the regularization parameter $\mu_k$ such that (22) holds for $\mu = \mu_k$.
  This may require increasing $k$. In this case, let $k = k + 1$ and go to line 4;
6 Determine $y_{k,\mu_k}$ by solving (18) and compute $\mathscr{X}_{k,\mu_k}$ from (26);
```

This section focused on the solution of equation (11). However, the solution method described can be applied to the solution of more general tensor (1).

4 Numerical examples

This section shows a few numerical experiments that illustrate the performance of the method described in Section 3. We limit ourselves to the case $N = 3$ in (2) and (3). For notational simplicity, we write (2) and (3) in the form $\mathcal{L}(\mathscr{X}) = \mathscr{C}$. The right-hand side tensor is in all test problems contaminated by an error tensor $\mathscr{E}$ with normally distributed random entries with zero mean. The entries are scaled to yield a specified noise level

$$\nu := \frac{\|\mathscr{E}\|}{\|\mathscr{C}\|}.$$

All computations were carried out using the Tensor Toolbox [1] in MATLAB version R2018b with an Intel Core i7-4770K CPU @ 3.50-GHz processor and 24-GB RAM.

We report the relative errors

$$e_k := \frac{\|\mathscr{X}_{\mu_k,k} - \hat{\mathscr{X}}\|}{\|\hat{\mathscr{X}}\|},$$

where $\hat{\mathscr{X}}$ denotes the desired solution of the problem with error-free right-hand side tensor $\hat{\mathscr{C}}$ associated with $\mathscr{C}$, and $\mathscr{X}_{\mu_k,k}$ denotes the kth computed approximation determined by the algorithms.

In the computations for Tables 1, 5, and 7, the iterations were terminated as soon as an approximate solution $\mathscr{X}_{\mu_k,k}$ was found such that the discrepancy principle

$$\left\|\mathcal{L}(\mathscr{X}_{\mu_k,k}) - \mathscr{C}\right\| \le \eta\varepsilon, \tag{27}$$

was satisfied, where $\eta = 1.01$ is a user-chosen constant and ε is the norm of error in $\mathscr{C}$, i.e., $\varepsilon = \|\mathscr{E}\|$. Our numerical results illustrate that the performance of the algorithms is not very sensitive to the choice of $\eta(\ge 1)$; we illustrate the convergence behavior of the algorithms for several values of η in Example 5. We remark that the left-hand side of (27) can be computed inexpensively by using (17) with $\mathcal{M}$ and $\mathscr{F}$ replaced by $\mathcal{L}$ and $\mathscr{C}$, respectively. We compare Algorithm 2 of the present paper to methods that apply the Hessenberg and flexible Hessenberg processes based on tensor format to reduce the given large problem to smaller ones. These methods are used together with Tikhonov regularization and are described in [24]. The discrepancy principle is used to determine the regularization parameter. We refer to the method that uses the Hessenberg process based on tensor format together with Tikhonov regularization as the HT_BTF method; when the Hessenberg process based on tensor format is replaced by the flexible Hessenberg process based on tensor format, the resulting method is referred to as the FHT_BTF method.

When the coefficient matrices are dense and not very large, the FHT_BTF method outperforms the other methods in our comparison. However, for large and sparse coefficient matrices, FHT_BTF requires more CPU time than Algorithm 2. For large problems, the FHT_BTF method requires many iterations to satisfy the stopping criterion (27). We therefore for the results reported in Tables 2, 3, 4, 6,8, 9, and 10 used the alternative stopping criterion

$$\frac{\|\mathscr{X}_{\mu_k,k}-\mathscr{X}_{\mu_{k-1},k-1}\|}{\|\mathscr{X}_{\mu_{k-1},k-1}\|}\leq\tau \tag{28}$$

for a user-specified value of the parameter $\tau > 0$. Moreover, at most 300 iterations were allowed. In the FHT_BTF method, we used two steps of the stabilized biconjugate gradient method based on tensor format (BiCGSTAB_BTF) [9] as inner iteration; see [24] for further details. Choosing a smaller value of τ results in that a larger number of iteration is required to satisfy (28). We illustrate the performance of Algorithm 2 for several values of τ in Example 5.

We report the number of iterations and the CPU-time (in seconds) required by the methods in our comparison to compute approximate solutions that satisfy the specified stopping criteria. Section 4.1 discusses the solution of severely ill-conditioned problems of the form (2) and Section 4.2 considers severely ill-conditioned problems of the form (3). The blurring matrices used in Section 4.1 can be expressed as

$$I\otimes I\times A^{(1)}+I\otimes A^{(2)}\otimes I+A^{(3)}\otimes I\otimes I,$$

while the blurring matrices applied in Section 4.2 can be written as

$$I-A^{(3)}\otimes A^{(2)}\otimes A^{(1)},$$

where the $A^{(\ell)}$ are a Gaussian Toeplitz matrix $A=[a_{ij}]$ given by

$$a_{ij}=\begin{cases}\dfrac{1}{\sigma\sqrt{2\pi}}\exp\left(-\dfrac{(i-j)^2}{2\sigma^2}\right), & |i-j|\leq r,\\ 0, & \text{otherwise},\end{cases} \tag{29}$$

or a Toeplitz matrix with entries

$$a_{ij}=\begin{cases}\dfrac{1}{2r-1}, & |i-j|\leq r,\\ 0, & \text{otherwise}.\end{cases} \tag{30}$$

We further present some experiments for Sylvester and Stein tensor equations with the coefficient matrices given in Case II of Remark 1 at the end of each subsection.

Blurring matrices of type (29) and (30) have been used in the literature to test iterative schemes for image deblurring; see [4–6, 15].

4.1 Experimental results for severely ill-conditioned Sylvester tensor equations

We consider (2) with coefficient matrices that are dense and very ill-conditioned. This kind of equation arises from the discretization of a fully three-dimensional microscale dual-phase lag problem by a mixed-collocation finite difference method; see [21–23] for details.

Example 1 Consider (2) with the matrices $A^{(\ell)} = [a_{ij}] \in \mathbb{R}^{n\times n}$ for $\ell = 1, 2, 3$ defined by

$$a_{ij} = \begin{cases} -2\left(\frac{\pi}{L}\right)^2 \frac{(-1)^{i+j}}{\sin^2\left[\frac{1}{2}\left(\frac{2\pi\xi_j}{L} - x_i\right)\right]}, & i \neq j \\ -\left(\frac{\pi}{L}\right)^2\left(\frac{n^2+2}{3}\right), & i = j, \end{cases}$$

where $x_i = \frac{2\pi(i-1)}{n}$, $\xi_j = \frac{(j-1)L}{n}$, $i, j = 1, 2, \ldots, n$, and $L = 300$. When n is odd, the coefficient matrices $A^{(\ell)}$ are well-conditioned and the problem can be solved successfully with a block iterative method; see [3]. However, when n is even, the coefficient matrices are very ill-conditioned. This is illustrated in [24, Example 5.4]. The error-free right-hand side of (2) is constructed so that $\hat{\mathscr{X}} = \text{randn}(n, n, n)$ is the exact solution, i.e., $\hat{\mathscr{X}}$ has normally distributed random entries with mean zero and variance one. Table 1 shows the numerical results obtained. Computed approximate solutions and the exact solution are displayed in Fig. 3.

Table 1 Comparison results for Example 1 with respect to stopping criterion (27)

Grid	cond($A^{(i)}$)	Level of noise (ν)	Method	Iter(k)	e_k	CPU-time (sec)
$100 \times 100 \times 100$	$1.25 \cdot 10^{16}$	0.01	Algorithm 2	39	$1.11 \cdot 10^{-1}$	2.31
			HT_BTF	11	$7.51 \cdot 10^{-2}$	3.62
			FHT_BTF	5	$6.25 \cdot 10^{-2}$	0.98
		0.001	Algorithm 2	134	$4.48 \cdot 10^{-2}$	7.53
			HT_BTF	24	$2.57 \cdot 10^{-2}$	34.84
			FHT_BTF	8	$2.33 \cdot 10^{-2}$	2.35
$150 \times 150 \times 150$	$4.67 \cdot 10^{16}$	0.01	Algorithm 2	37	$1.18 \cdot 10^{-1}$	7.03
			HT_BTF	11	$7.40 \cdot 10^{-2}$	12.32
			FHT_BTF	5	$6.33 \cdot 10^{-2}$	3.34
		0.001	Algorithm 2	178	$4.02 \cdot 10^{-2}$	36.89
			HT_BTF	21	$3.21 \cdot 10^{-2}$	72.95
			FHT_BTF	8	$2.61 \cdot 10^{-2}$	8.15
$180 \times 180 \times 180$	$3.28 \cdot 10^{16}$	0.01	Algorithm 2	36	$1.19 \cdot 10^{-1}$	11.13
			HT_BTF	11	$7.55 \cdot 10^{-2}$	21.45
			FHT_BTF	5	$6.13 \cdot 10^{-2}$	5.66
		0.001	Algorithm 2	154	$4.18 \cdot 10^{-2}$	58.64
			HT_BTF	22	$2.89 \cdot 10^{-2}$	134.51
			FHT_BTF	7	$2.91 \cdot 10^{-2}$	11.65

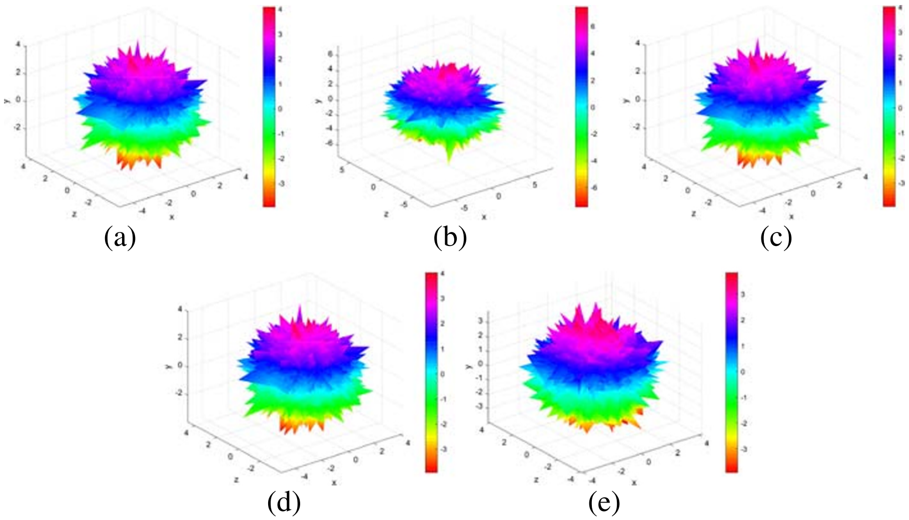


Fig. 3 **a** Exact solution on grid $180 \times 180 \times 180$, **b** noisy data with noise level $\nu = 0.01$, **c** restored solution by HT_BTF, **d** FHT_BTF, and **e** Algorithm 2

Table 1 shows the FHT_BTF method to perform better than the other methods. This is typical for problems with dense coefficient matrices.

We next turn to an image restoration problem, in which the error-free right-hand side in (2) is constructed so that the exact solution is a hyperspectral image. Here, the matrices $A^{(i)}$, $i = 1, 2, 3$, are sparse and we will see that Algorithm 2 performs the best.

Example 2 We consider the situation when the exact solution of (2) is a tensor of order $1019 \times 1337 \times 33$ that represents a hyperspectral image of a natural scene.[2] The coefficient matrices $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are defined by (30) with suitable dimensions and with $r = 2$ for $A^{(1)}$ and $A^{(2)}$, and $r = 3$ for $A^{(3)}$. This gives $\text{cond}(A^{(1)}) = 5.26 \cdot 10^{16}$, $\text{cond}(A^{(2)}) = 1.75 \cdot 10^{17}$, and $\text{cond}(A^{(3)}) = 4.75 \cdot 10^{16}$. Thus, all the coefficient matrices are numerically singular.

As mentioned above, the (F)HT_BTF methods can not be efficiently used with the stopping criterion (27). Therefore, we used the stopping criterion (28) for all algorithms. The results are reported in Table 2. Algorithm 2 can be seen to perform better than the HT_BTF and FHT_BTF methods. Table 7 illustrates that the computational effort increases as the error in the tensor $\mathscr{C}$ decreases. Here, Algorithm 2 was terminated as soon as (27) was satisfied. The contaminated and restored images are displayed in Figs. 4 and 5.

[2] http://personalpages.manchester.ac.uk/staff/d.h.foster

Table 2 Results for Example 2 using the stopping criterion (28) with $\tau = 2 \cdot 10^{-2}$

Noise level ν	Method	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	4	$3.85 \cdot 10^{-2}$	7.45
	HT_BTF	4	$6.77 \cdot 10^{-2}$	27.01
	FHT_BTF	2	$6.28 \cdot 10^{-2}$	26.81
0.001	Algorithm 2	4	$3.57 \cdot 10^{-2}$	7.47
	HT_BTF	4	$4.40 \cdot 10^{-2}$	26.67
	FHT_BTF	2	$3.85 \cdot 10^{-2}$	25.30

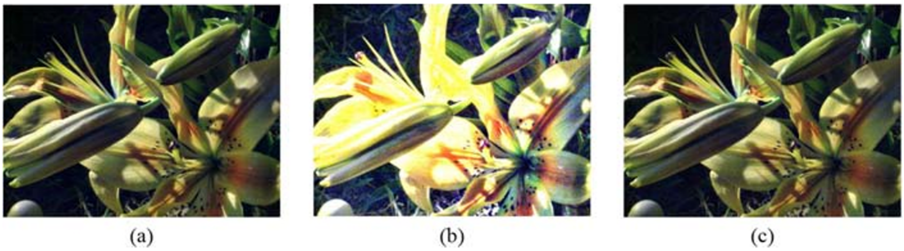


Fig. 4 Example 2. **a** Exact image, **b** blurred and noisy image with noise level $\nu = 0.01$, and **c** restored image by Algorithm 2 using the stopping criterion (27)

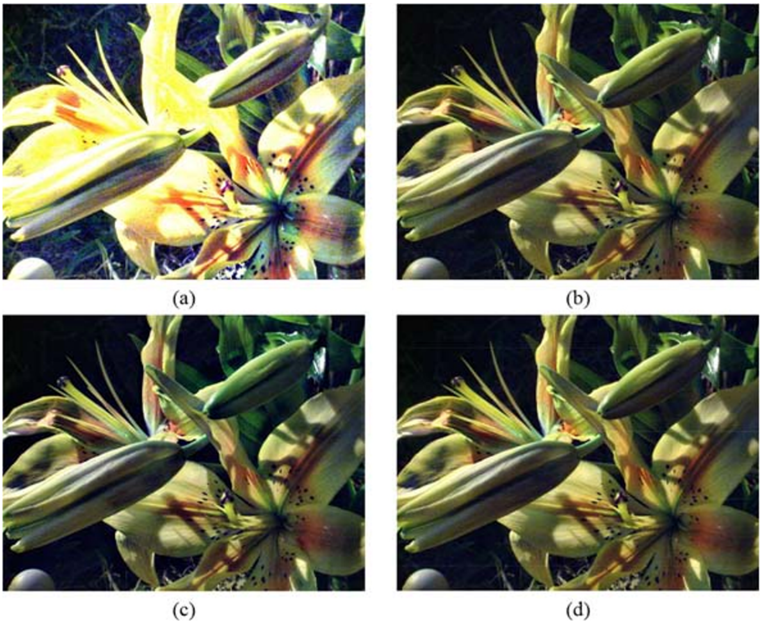


Fig. 5 Example 2. **a** blurred and noisy image with noise level $\nu = 0.01$, restored images by **b** HT_BTF, **c** FHT_BTF, and **d** Algorithm 2 using the stopping criterion (28)

Example 3 Consider the Sylvester tensor equation (2) whose coefficient matrices $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are defined by (30) with $r = 30$ for $A^{(1)}$, $r = 20$ for $A^{(2)}$, and $r = 20$ for $A^{(3)}$. We examine the performance of algorithms for the following cases:

Case I Let the exact solution of (2) be hyperspectral image of order $1019 \times 1337 \times 33$ in the above example. Here, we have $\mathrm{cond}(A^{(1)}) = 1.66 \cdot 10^{18}$, $\mathrm{cond}(A^{(2)}) = 4.13 \cdot 10^{19}$, and $\mathrm{cond}(A^{(3)}) = 5.59 \cdot 10^{18}$.

Case II Let $\hat{\mathscr{X}} = \mathrm{randn}(1000, 500, 100)$ be the exact solution of (2), i.e., $A^{(1)} \in \mathbb{R}^{1000\times 1000}$, $A^{(2)} \in \mathbb{R}^{500\times 500}$ and $A^{(3)} \in \mathbb{R}^{100\times 100}$ for which $\mathrm{cond}(A^{(1)}) = 1.74 \cdot 10^{18}$, $\mathrm{cond}(A^{(2)}) = 8.07 \cdot 10^{17}$, and $\mathrm{cond}(A^{(3)}) = 3.66 \cdot 10^{18}$.

Results for these cases are reported in Table 3. The table shows Algorithm 2 to converge faster for Case I. However, the HT_BTF method outperforms the other approaches for the noise level 0.01 for Case II. We remark that the performance of the methods when applied to the Stein tensor equation is different when increasing r in the coefficient matrices; see Example 7 for more details.

We turn to results for the Sylvester tensor equation with the coefficient matrices given in Case II of Remark 1. This equation arises from the discretisation of a three-dimensional convection-diffusion equation on a uniform grid using a standard finite difference for the diffusion term and a second order convergent scheme (Fromm's scheme) for the convection term with mesh size $h = 1/(n + 1)$; see [2, 3]. This problem was examined in [3] for $n \times n \times n$ grids with $n \leq 110$, for which the corresponding matrix $\mathcal{A}$ is not severely ill-conditioned. However, the condition number increases with the value of n.

Example 4 Consider the Sylvester tensor equation for $N = 3$ with the coefficient matrices $A^{(\ell)}$ for $\ell = 1, 2, 3$ given in the second case of Remark 1. Table 4 shows that Algorithm 2 is an efficient solver. When the noise level is small, FHT_BTF requires more CPU-time than HT_BTF and produces slightly more accurate approximate solutions.

Table 3 Results for Example 3 using the stopping criterion (28) with $\tau = 2 \cdot 10^{-2}$

Noise level ν	Mehtod	Case I			Case II		
		Iter(k)	e_k	CPU-time (sec)	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	7	$1.02 \cdot 10^{-1}$	41.09	41	$7.02 \cdot 10^{-1}$	439.29
	HT_BTF	8	$9.08 \cdot 10^{-2}$	123.69	10	$9.03 \cdot 10^{-1}$	221.93
	FHT_BTF	7	$1.18 \cdot 10^{-1}$	217.22	7	$8.65 \cdot 10^{-1}$	245.47
0.001	Algorithm 2	8	$8.22 \cdot 10^{-2}$	49.77	42	$6.95 \cdot 10^{-1}$	452.10
	HT_BTF	8	$8.67 \cdot 10^{-2}$	123.64	14	$8.82 \cdot 10^{-1}$	498.97
	FHT_BTF	5	$6.93 \cdot 10^{-2}$	145.75	7	$8.63 \cdot 10^{-1}$	245.85

Table 4 Results for Example 4 using stopping criterion (28)

Grid		$200 \times 200 \times 200$			$350 \times 350 \times 350$		
		$(\tau = 1 \cdot 10^{-3})$			$(\tau = 2 \cdot 10^{-3})$		
Noise level ν	Method	Iter(k)	e_k	CPU-time (sec)	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	67	$4.36 \cdot 10^{-2}$	82.99	44	$5.03 \cdot 10^{-2}$	221.85
	HT_BTF	22	$3.25 \cdot 10^{-2}$	264.85	17	$3.30 \cdot 10^{-2}$	647.16
	FHT_BTF	20	$3.21 \cdot 10^{-2}$	205.10	15	$3.28 \cdot 10^{-2}$	547.81
0.001	Algorithm 2	65	$3.88 \cdot 10^{-2}$	79.20	44	$4.85 \cdot 10^{-2}$	222.13
	HT_BTF	18	$2.16 \cdot 10^{-2}$	142.22	17	$2.19 \cdot 10^{-2}$	614.08
	FHT_BTF	19	$1.60 \cdot 10^{-2}$	187.02	16	$1.79 \cdot 10^{-2}$	643.74

4.2 Experimental results for severely ill-conditioned Stein tensor equations

In this subsection, we consider the solution of three severely ill-conditioned problems of the form (3). For the first two examples, error-free right-hand sides are constructed so that the exact solutions are color images. The iterations with the algorithms were terminated with the stopping criteria (27) or (28). We conclude this subsection by reporting the results for Stein tensor equations with the coefficient matrices given in Case II of Remark 1.

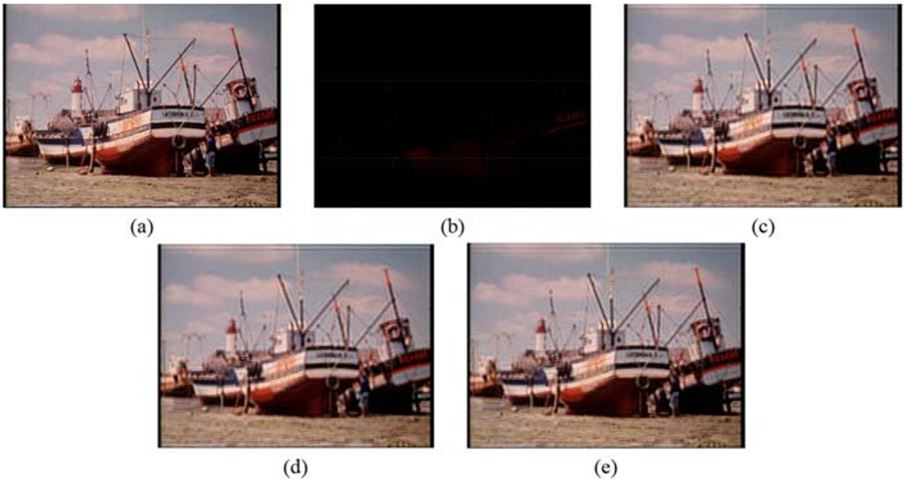
Example 5 The "exact" image[3] is represented by a $576 \times 787 \times 3$ tensor and is displayed in Fig. 6a. The coefficient matrices of (3) are $A^{(1)}$, which is defined by (29), and $A^{(2)}$ and $A^{(3)}$, which are given by (30), and have suitable dimensions. We set $r = 7, \sigma = 2$ for $A^{(1)}$, and $r = 2$ for $A^{(2)}$ and $A^{(3)}$. Then $\text{cond}(A^{(1)}) = 1.79 \cdot 10^6$, $\text{cond}(A^{(2)}) = 4.05 \cdot 10^{17}$, and $\text{cond}(A^{(3)}) = 6.45 \cdot 10^{49}$. We found that when using the stopping criterion (28), the performance of Algorithm 2 is not very sensitive to small changes in $\eta (> 1)$ and τ; see Fig. 7 for details.

Example 6 Let the exact solution of (3) be of order $1019 \times 1337 \times 33$; it represents the hyperspectral image shown in Fig. 8. The coefficient matrices $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ of suitable dimensions are defined by (30) with $r = 12$ for $A^{(1)}$, $r = 2$ for $A^{(2)}$, and $r = 6$ for $A^{(3)}$. Then $\text{cond}(A^{(1)}) = 2.05 \cdot 10^{18}$, $\text{cond}(A^{(2)}) = 1.75 \cdot 10^{17}$, and $\text{cond}(A^{(3)}) = 2.44 \cdot 10^{17}$.

Tables 5, 6, 7, and 8 show results for Examples 5 and 6. Algorithm 2 can be seen to be superior to the other methods examined. The exact, contaminated, and restored images are shown in Figs. 6, 8, and 9.

Similarly to Example 3, we consider coefficient matrices (30) with larger values of r. Differently from Sylvester tensor equations, all algorithms perform better when

[3] The image is available at https://www.hlevkin.com/TestImages/Boats.ppm


(a) (b) (c)

(d) (e)

Fig. 6 Example 5. **a** Exact image, **b** blurred and noisy image with noise level $\nu = 0.01$, restored image by **c** HT_BTF, **d** FHT_BTF, and **e** Algorithm 2 using the stopping criterion (27)

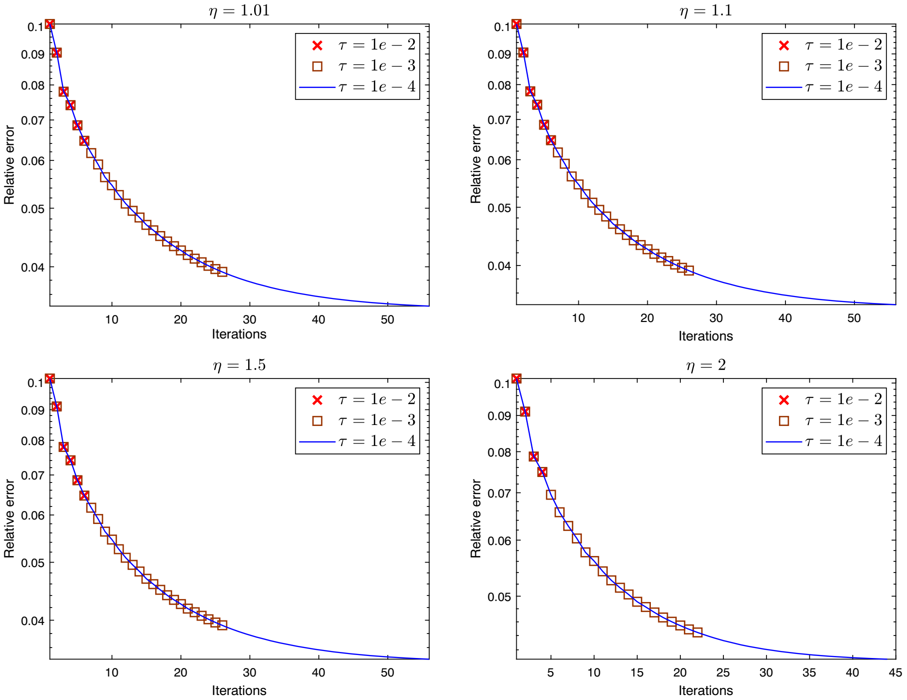


Fig. 7 Convergence history of Algorithm 2 for Example 5: Relative error versus iteration numbers with respect to different η and τ for noise level 0.01

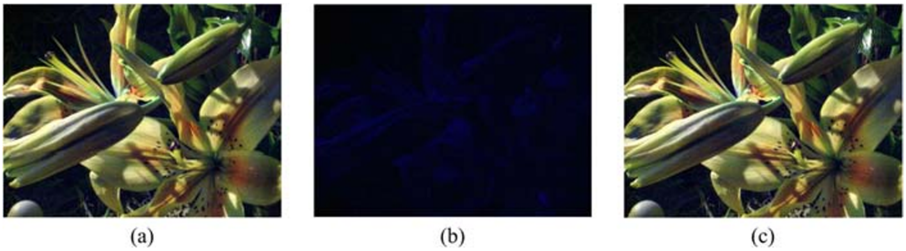


Fig. 8 Example 6. **a** Exact image, **b** blurred and noisy image with noise level $\nu = 0.01$, and **c** restored image by Algorithm 2 using the stopping criterion (27)

Table 5 Results for Example 5 using the stopping criterion (27)

Level of noise (ν)	Method	Iter(k)	e_k	CPU-times(sec)
0.01	Algorithm 2	13	$5.31 \cdot 10^{-2}$	0.80
	HT_BTF	12	$6.46 \cdot 10^{-2}$	8.71
	FHT_BTF	5	$6.58 \cdot 10^{-2}$	2.25
0.001	Algorithm 2	52	$2.63 \cdot 10^{-2}$	3.78
	HT_BTF	13	$5.97 \cdot 10^{-2}$	10.64
	FHT_BTF	6	$6.41 \cdot 10^{-2}$	3.02

Table 6 Results for Example 5 using the stopping criterion (28) with $\tau = 2 \cdot 10^{-2}$

Noise level ν	Method	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	6	$7.40 \cdot 10^{-2}$	0.41
	HT_BTF	9	$7.88 \cdot 10^{-2}$	4.09
	FHT_BTF	6	$6.54 \cdot 10^{-2}$	2.86
0.001	Algorithm 2	6	$7.32 \cdot 10^{-2}$	0.38
	HT_BTF	9	$7.84 \cdot 10^{-2}$	4.14
	FHT_BTF	6	$6.42 \cdot 10^{-2}$	2.87

Table 7 Results for Algorithm 2 with the stopping criterion (27)

	Example 2			Example 6		
Noise level ν	Iter(k)	e_k	CPU-time (sec)	Iter(k)	e_k	CPU-time (sec)
0.01	6	$3.54 \cdot 10^{-2}$	14.31	18	$7.98 \cdot 10^{-2}$	58.76
0.001	20	$1.72 \cdot 10^{-2}$	57.03	31	$5.62 \cdot 10^{-2}$	96.14

Table 8 Results for Example 6 using the stopping criterion (28) with $\tau = 3 \cdot 10^{-2}$

Noise level ν	Method	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	6	$1.39 \cdot 10^{-1}$	17.73
	HT_BTF	6	$2.61 \cdot 10^{-1}$	54.83
	FHT_BTF	4	$1.40 \cdot 10^{-1}$	57.04
0.001	Algorithm 2	6	$1.37 \cdot 10^{-1}$	17.74
	HT_BTF	6	$2.44 \cdot 10^{-1}$	54.94
	FHT_BTF	4	$1.38 \cdot 10^{-1}$	57.96

increasing the value of r. For the Stein tensor equation, we note that Algorithm 2 can be competitive with the (F)HT_BTF method.

Example 7 Consider the Stein tensor (3) with the matrices $A^{(\ell)}$ given by (30) for $\ell = 1, 2, 3$. Let $r = 40$ for $A^{(1)}$, $r = 50$ for $A^{(2)}$, and $r = 30$ for $A^{(3)}$. Table 9 reports results for the following two cases:

Case I Let the exact solution of (3) be the hyperspectral image of order $1019 \times 1337 \times 33$ mentioned above. We have $\text{cond}(A^{(1)}) = 1.18 \cdot 10^{18}$, $\text{cond}(A^{(2)}) = 4.87 \cdot 10^{18}$, and $\text{cond}(A^{(3)}) = 3.12 \cdot 10^{114}$.

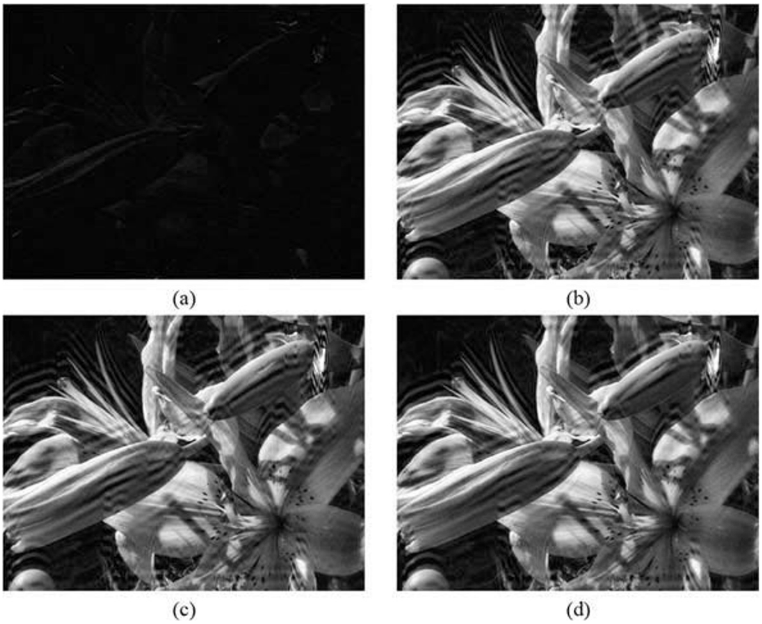


Fig. 9 Example 6. **a** Blurred and noisy image, restored images by **b** HT_BTF, **c** FHT_BTF, and **d** Algorithm 2 using the stopping criterion (28) with $\tau = 3 \cdot 10^{-2}$

Table 9 Results for Example 7 using the stopping criterion (28) with $\tau = 3 \cdot 10^{-2}$

Noise level ν	Mehtod	Case I			Case II		
		Iter(k)	e_k	CPU-time (sec)	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	6	$2.02 \cdot 10^{-2}$	43.31	4	$1.34 \cdot 10^{-2}$	24.12
	HT_BTF	6	$7.40 \cdot 10^{-3}$	75.35	3	$1.04 \cdot 10^{-2}$	28.74
	FHT_BTF	3	$7.30 \cdot 10^{-3}$	103.68	3	$1.01 \cdot 10^{-2}$	115.28
0.001	Algorithm 2	6	$1.06 \cdot 10^{-2}$	43.35	4	$1.50 \cdot 10^{-3}$	24.21
	HT_BTF	6	$1.20 \cdot 10^{-3}$	75.61	3	$1.30 \cdot 10^{-3}$	28.65
	FHT_BTF	3	$7.32 \cdot 10^{-4}$	102.92	3	$1.00 \cdot 10^{-3}$	115.34

Case II Let $\hat{\mathscr{X}} = \text{randn}(1000, 500, 100)$ be the exact solution of (3); i.e., $A^{(1)} \in \mathbb{R}^{1000\times 1000}$, $A^{(2)} \in \mathbb{R}^{500\times 500}$, and $A^{(3)} \in \mathbb{R}^{100\times 100}$. We have $\text{cond}(A^{(1)}) = 2.48 \cdot 10^{19}$, $\text{cond}(A^{(2)}) = 6.70 \cdot 10^{17}$, and $\text{cond}(A^{(3)}) = 5.70 \cdot 10^{18}$.

The results reported in Table 9 show Algorithm 2 to perform better than (F)HT_BTF for larger values of r.

We conclude this subsection by reporting results for a Stein tensor equation, whose coefficient matrices are given by (6).

Example 8 Let $\hat{\mathscr{X}} = \text{randn}(n, n, n)$ be the exact solution of equation (3) and let the coefficient matrices $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ be defined by (6). We observe that the (F)HT_BT methods perform less well when increasing the problem size. Therefore, we used a slightly larger value of τ for $n = 200$. Table 10 shows that HT_BT is superior to Algorithm 2 for $n = 120$. When $n = 200$, Algorithm 2 outperforms (F)HT_BT.

Table 10 Results for Example 8 using the stopping criterion (28)

Noise level ν	Mehtod	$n = 120$ ($\tau = 2 \cdot 10^{-3}$)			$n = 200$ ($\tau = 1 \cdot 10^{-2}$)		
		Iter(k)	e_k	CPU-time (sec)	Iter(k)	e_k	CPU-time (sec)
0.01	Algorithm 2	212	$5.10 \cdot 10^{-1}$	128.95	49	$6.21 \cdot 10^{-1}$	47.93
	HT_BTF	14	$5.70 \cdot 10^{-1}$	16.14	22	$5.74 \cdot 10^{-1}$	242.70
	FHT_BTF	17	$6.01 \cdot 10^{-1}$	30.97	24	$6.31 \cdot 10^{-1}$	336.09
0.001	Algorithm 2	192	$5.09 \cdot 10^{-1}$	109.50	49	$5.93 \cdot 10^{-1}$	47.86
	HT_BTF	16	$5.48 \cdot 10^{-1}$	22.94	21	$5.88 \cdot 10^{-1}$	213.60
	FHT_BTF	20	$3.75 \cdot 10^{-1}$	46.99	12	$4.08 \cdot 10^{-1}$	61.33

5 Conclusions

This paper first presents some results on the conditioning of the Stein tensor equation. Then it introduces the Golub–Kahan bidiagonalization process with application to solving severely ill-conditioned linear tensor equations, such as Sylvester and Stein tensor equations. The iterative scheme also can be applied to the solution of general linear tensor equations with an operator over $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_k}$. We provide new theoretical results and present some numerical examples with applications to high-dimensional PDEs and color image restoration to illustrate the applicability and effectiveness of the proposed iterative method.

Acknowledgments The authors would like to thank the anonymous referees for their suggestions and comments.

Funding information Research by LR was supported in part by NSF grants DMS-1720259 and DMS-1729509.

References

1. Bader, B.W., Kolda, T.G.: MATLAB tensor toolbox version 2.5. http://www.sandia.gov/tgkolda/TensorToolbox
2. Ballani, J., Grasedyck, L.: A projection method to solve linear systems in tensor format. Numer. Linear Algebra Appl. **20**, 27–43 (2013)
3. Beik, F.P.A., Movahed, F.S., Ahmadi-Asl, S.: On the Krylov subspace methods based on tensor format for positive definite Sylvester tensor equations. Numer. Linear Algebra Appl. **23**, 444–466 (2016)
4. Bentbib, A.H., El Guide, M., Jbilou, K., Reichel, L.: A global Lanczos method for image restoration. J. Comput. Appl. Math. **300**, 233–244 (2016)
5. Bentbib, A.H., El Guide, M., Jbilou, K., Reichel, L.: Global Golub–Kahan bidiagonalization applied to large discrete ill-posed problems. J. Comput. Appl. Math. **322**, 46–56 (2017)
6. Bouhamidi, A., Jbilou, K., Reichel, L., Sadok, H.: A generalized global Arnoldi method for ill-posed matrix equations. J. Comput. Appl. Math. **236**, 2078–2089 (2012)
7. Buccini, A., Pasha, M., Reichel, L.: Generalized singular value decomposition with iterated Tikhonov regularization. J. Comput. Appl. Math. **373**, 112276 (2020)
8. Calvetti, D., Reichel, L.: Tikhonov regularization of large linear problems. BIT **43**, 263–283 (2003)
9. Chen, Z., Lu, L.Z.: A projection method and Kronecker product preconditioner for solving Sylvester tensor equations. Sci. China Math. **55**, 1281–1292 (2012)
10. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I.: Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons, Chichester (2009)
11. Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT **17**, 134–145 (1977)
12. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems. Kluwer, Dordrecht (1996)
13. Golub, G.H., Van Loan, C.F.: Matrix computations. The Johns Hopkins University Press, Batimore (1996)
14. Horn, R.A., Johnson, C.R.: Matrix analysis. Cambridge University Press, Cambridge (1985)
15. Huang, G., Reichel, L., Yin, F.: On the choice of subspace for large-scale Tikhonov regularization problems in general form. Numer. Algorithms **81**, 33–55 (2019)
16. Huang, B., Xie, Y., Ma, C.: Krylov subspace methods to solve a class of tensor equations via the Einstein product. Numer. Linear Algebra Appl. **26**, e2254 (2019)
17. Kindermann, S.: Convergence analysis of minimization-based noise level-freeparameter choice rules for linear ill-posed problems. Electron. Trans. Numer. Anal. **38**, 233–257 (2011)

18. Kindermann, S., Raik, K.: A simplified L-curve method as error estimator. Electron. Trans. Numer. Anal. **53**, 217–238 (2020)
19. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2009)
20. Liang, L., Zheng, B.: Sensitivity analysis of the Lyapunov tensor equation. Linear Multilinear Algebra **67**, 555–572 (2019)
21. Malek, A., Bojdi, Z.K., Golbarg, P.N.N.: Solving fully three-dimensional microscale dual phase lag problem using mixed-collocation finite difference discretization. J. Heat Transf. **134**, 094504 (2012)
22. Malek, A., Masuleh, S.H.M.: Mixed collocation-finite difference method for 3D microscopic heat transport problems. J. Comput. Appl. Math. **217**, 137–147 (2008)
23. Masuleh, S.H.M., Phillips, T.N.: Viscoelastic flow in an undulating tube using spectral methods. Comput. Fluids **33**, 1075–1095 (2004)
24. Najafi-Kalyani, M., Beik, F.P.A., Jbilou, K.: On global iterative schemes based on Hessenberg process for (ill-posed) Sylvester tensor equations. J. Comput. Appl. Math. **373**, 112216 (2020)
25. Paige, C.C., Saunders, M.A.: LSQR: An algorithm for sparse linear equations and sparse least squares problems. ACM Trans. Math. Softw. **8**, 43–71 (1982)
26. Reichel, L., Shyshkov, A.: A new zero-finder for Tikhonov regularization. BIT **48**, 627–643 (2008)
27. Sun, Y.S., Jing, M., Li, B.W.: Chebyshev collocation spectral method for three-dimensional transient coupled radiative-conductive heat transfer. J. Heat Transf. **134**, 092701–092707 (2012)
28. Xu, X., Wang, Q.-W.: Extending BiCG and BiCR methods to solve the Stein tensor equation. Comput. Math. Appl. **77**, 3117–3127 (2019)
29. Zak, M.K., Toutounian, F.: Nested splitting conjugate gradient method for matrix equation $AXB = C$ and preconditioning. Comput. Math. Appl. **66**, 269–278 (2013)
30. Zak, M.K., Toutounian, F.: Nested splitting CG-like iterative method for solving the continuous Sylvester equation and preconditioning. Adv. Comput. Math. **40**, 865–880 (2014)
31. Zak, M.K., Toutounian, F.: An iterative method for solving the continuous Sylvester equation by emphasizing on the skew-Hermitian parts of the coefficient matrices. Internat. J. Comput. Math. **94**, 633–649 (2017)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.