Electronic Transactions on Numerical Analysis.
Volume 43, pp. 244–254, 2015.

ISSN 1068–9613.

ETNA
Kent State University
http://etna.math.kent.edu

ON COMPUTING MAXIMUM/MINIMUM SINGULAR VALUES OF A GENERALIZED TENSOR SUM*

ASUKA OHASHI† AND TOMOHIRO SOGABE‡

Abstract. We consider the efficient computation of the maximum/minimum singular values of a generalized tensor sum T. The computation is based on two approaches: first, the Lanczos bidiagonalization method is reconstructed over tensor space, which leads to a memory-efficient algorithm with a simple implementation, and second, a promising initial guess given in Tucker decomposition form is proposed. From the results of numerical experiments, we observe that our computation is useful for matrices being near symmetric, and it has the potential of becoming a method of choice for other cases if a suitable core tensor can be given.

Key words. generalized tensor sum, Lanczos bidiagonalization method, maximum and minimum singular values

AMS subject classifications. 65F10

1. Introduction.

We consider computing the maximum/minimum singular values of a generalized tensor sum

$$T := I_n \otimes I_m \otimes A + I_n \otimes B \otimes I_\ell + C \otimes I_m \otimes I_\ell \quad \in \mathbb{R}^{\ell mn \times \ell mn}, \tag{1.1}$$

where $A \in \mathbb{R}^{\ell \times \ell}$, $B \in \mathbb{R}^{m \times m}$, $C \in \mathbb{R}^{n \times n}$, I_m is the $m \times m$ identity matrix, the symbol "$\otimes$" denotes the Kronecker product, and the matrix T is assumed to be large, sparse, and nonsingular. Such matrices T arise in a finite-difference discretization of three-dimensional constant-coefficient partial differential equations, such as

$$\begin{aligned} \left[-\boldsymbol{a} \cdot \left(\nabla * \nabla\right) + \boldsymbol{b} \cdot \nabla + c\right] u(x,y,z) &= g(x,y,z) \quad && \text{in } \Omega, \\ u(x,y,z) &= 0 && \text{on } \partial\Omega, \end{aligned} \tag{1.2}$$

where $\Omega = (0,1) \times (0,1) \times (0,1)$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$, $c \in \mathbb{R}$, and the symbol "$*$" denotes the elementwise product. If $\boldsymbol{a} = (1,1,1)$, then $\boldsymbol{a} \cdot (\nabla * \nabla) = \Delta$. With regard to efficient numerical methods for linear systems of the form $T\boldsymbol{x} = \boldsymbol{f}$, see [3, 9].

The Lanczos bidiagonalization method is widely known as an efficient method to compute the maximum/minimum singular values of a large and sparse matrix. For its recent successful variants, see, e.g., [2, 6, 7, 10, 12], and for other successful methods, see, e.g., [5]. The Lanczos bidiagonalization method does not require T and T^{T} themselves but only the results of the matrix-vector multiplications $T\boldsymbol{v}$ and $T^{\mathrm{T}}\boldsymbol{v}$. Even though one stores, as usual, only the non-zero entries of T, the storage required grows cubically with n under the assumption that $\ell = m = n$ and, as it is often the case, that the number of non-zero entries of A, B, and C grows linearly with n.

In order to avoid large memory usage, we consider the Lanczos bidiagonalization method over tensor space. Advantages of this approach are a low memory requirement and a very simple implementation. In fact, the required memory for storing T grows only linearly under the above assumptions.

Using the tensor structure, we present a promising initial guess in order to improve the speed of convergence of the Lanczos bidiagonalization method over tensor space, which is a major contribution of this paper.

*Received October 31, 2013. Accepted July 29, 2015. Published online on October 16, 2015. Recommended by H. Sadok.

†Graduate School of Information Science and Technology, Aichi Prefectural University, 1522-3 Ibaragabasama, Nagakute-shi, Aichi, 480-1198, Japan (id141002@cis.aichi-pu.ac.jp).

‡Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan (sogabe@na.cse.nagoya-u.ac.jp).

ETNA
Kent State University
http://etna.math.kent.edu

This paper is organized as follows. In Section 2, the Lanczos bidiagonalization method is introduced, and an algorithm is presented. In Section 3, some basic operations on tensors are described. In Section 4, we consider the Lanczos bidiagonalization method over tensor space and propose a promising initial guess using the tensor structure of the matrix T. Numerical experiments and concluding remarks are given in Sections 5 and 6, respectively.

2. The Lanczos bidiagonalization method. The Lanczos bidiagonalization method, which is due to Golub and Kahan [4], is suitable for computing maximum/minimum singular values. In particular, the method is widely used for large and sparse matrices. It employs sequences of projections of a matrix onto judiciously chosen low-dimensional subspaces and computes the singular values of the obtained matrix. By means of the projections, computing these singular values is more efficient than for the original matrix since the obtained matrix is smaller and has a simpler structure.

For a matrix $M \in \mathbb{R}^{\ell \times n} (\ell \geq n)$, the Lanczos bidiagonalization method displayed in Algorithm 1 calculates a sequence of vectors $\boldsymbol{p}_i \in \mathbb{R}^n$ and $\boldsymbol{q}_i \in \mathbb{R}^\ell$ and scalars α_i and β_i, where $i = 1, 2, \ldots, k$. Here, k represents the number of bidiagonalization steps and is typically much smaller than either one of the matrix dimensions ℓ and n.

Algorithm 1 The Lanczos bidiagonalization method [4].

Choose an initial vector $\boldsymbol{p}_1 \in \mathbb{R}^n$ such that $\|\boldsymbol{p}_1\|_2 = 1$.
$\boldsymbol{q}_1 := M\boldsymbol{p}_1$;
$\alpha_1 := \|\boldsymbol{q}_1\|_2$;
$\boldsymbol{q}_1 := \boldsymbol{q}_1/\alpha_1$;
for $i = 1, 2, \ldots, k$ **do**
  $\boldsymbol{r}_i := M^{\mathrm{T}}\boldsymbol{q}_i - \alpha_i \boldsymbol{p}_i$;
  $\beta_i := \|\boldsymbol{r}_i\|_2$;
  $\boldsymbol{p}_{i+1} := \boldsymbol{r}_i/\beta_i$;
  $\boldsymbol{q}_{i+1} := M\boldsymbol{p}_{i+1} - \beta_i \boldsymbol{q}_i$;
  $\alpha_{i+1} := \|\boldsymbol{q}_{i+1}\|_2$;
  $\boldsymbol{q}_{i+1} := \boldsymbol{q}_{i+1}/\alpha_{i+1}$;
end for

After k steps, Algorithm 1 yields the following decompositions:

$$\begin{aligned} MP_k &= Q_k D_k, \\ M^{\mathrm{T}} Q_k &= P_k D_k^{\mathrm{T}} + \boldsymbol{r}_k \boldsymbol{e}_k^{\mathrm{T}}, \end{aligned} \tag{2.1}$$

where the vectors $\boldsymbol{e}_k$ and $\boldsymbol{r}_k$ denote the k-th canonical and the k-th residual vector in Algorithm 1, respectively, and the matrices D_k, P_k, and Q_k are given as

$$D_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{k-1} & \beta_{k-1} \\ & & & & \alpha_k \end{bmatrix} \in \mathbb{R}^{k \times k}, \tag{2.2}$$

$$P_k = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_k) \in \mathbb{R}^{n \times k}, \quad Q_k = (\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k) \in \mathbb{R}^{\ell \times k}.$$

Here, P_k and Q_k are column orthogonal matrices, i.e., $P_k^{\mathrm{T}} P_k = Q_k^{\mathrm{T}} Q_k = I_k$.

Now, the singular triplets of the matrices M and D are explained. Let $\sigma_1^{(M)}, \sigma_2^{(M)}, \ldots, \sigma_n^{(M)}$ be the singular values of M such that $\sigma_1^{(M)} \geq \sigma_2^{(M)} \geq \cdots \geq \sigma_n^{(M)}$. Moreover, let $\boldsymbol{u}_i^{(M)} \in \mathbb{R}^\ell$

ETNA
Kent State University
http://etna.math.kent.edu

and $\boldsymbol{v}_i^{(M)} \in \mathbb{R}^n$, where $i = 1, 2, \ldots, n$, be the left and right singular vectors corresponding to the singular values $\sigma_i^{(M)}$, respectively. Then, $\{\sigma_i^{(M)}, \boldsymbol{u}_i^{(M)}, \boldsymbol{v}_i^{(M)}\}$ is referred to as a singular triplet of M, and the relations between M and its singular triplets are given as

$$M\boldsymbol{v}_i^{(M)} = \sigma_i^{(M)}\boldsymbol{u}_i^{(M)}, \quad M^{\mathrm{T}}\boldsymbol{u}_i^{(M)} = \sigma_i^{(M)}\boldsymbol{v}_i^{(M)},$$

where $i = 1, 2, \ldots, n$.

Similarly, with regard to D_k in (2.2), let $\{\sigma_i^{(D_k)}, \boldsymbol{u}_i^{(D_k)}, \boldsymbol{v}_i^{(D_k)}\}$ be the singular triplets of D_k. Then, the relations between D_k and its singular triplets are

$$D_k\boldsymbol{v}_i^{(D_k)} = \sigma_i^{(D_k)}\boldsymbol{u}_i^{(D_k)}, \quad D_k^{\mathrm{T}}\boldsymbol{u}_i^{(D_k)} = \sigma_i^{(D_k)}\boldsymbol{v}_i^{(D_k)}, \tag{2.3}$$

where $i = 1, 2, \ldots, k$. Moreover, $\{\tilde{\sigma}_i^{(M)}, \tilde{\boldsymbol{u}}_i^{(M)}, \tilde{\boldsymbol{v}}_i^{(M)}\}$ denotes the approximate singular triplet of M. They are determined from the singular triplet of D_k as follows:

$$\tilde{\sigma}_i^{(M)} := \sigma_i^{(D_k)}, \quad \tilde{\boldsymbol{u}}_i^{(M)} := Q_k\boldsymbol{u}_i^{(D_k)}, \quad \tilde{\boldsymbol{v}}_i^{(M)} := P_k\boldsymbol{v}_i^{(D_k)}. \tag{2.4}$$

Then, it follows from (2.1), (2.3), and (2.4) that

$$\begin{aligned} M\tilde{\boldsymbol{v}}_i^{(M)} &= \tilde{\sigma}_i^{(M)}\tilde{\boldsymbol{u}}_i^{(M)}, \\ M^{\mathrm{T}}\tilde{\boldsymbol{u}}_i^{(M)} &= \tilde{\sigma}_i^{(M)}\tilde{\boldsymbol{v}}_i^{(M)} + \boldsymbol{r}_k\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{u}_i^{(D_k)}, \end{aligned} \tag{2.5}$$

where $i = 1, 2, \ldots, k$. Equations (2.5) imply that the approximate singular triplet $\{\tilde{\sigma}_i^{(M)}, \tilde{\boldsymbol{u}}_i^{(M)}, \tilde{\boldsymbol{v}}_i^{(M)}\}$ is acceptable for the singular triplet $\{\sigma_i^{(M)}, \boldsymbol{u}_i^{(M)}, \boldsymbol{v}_i^{(M)}\}$ if the value of $\|\boldsymbol{r}_k\|_2|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{u}_i^{(D_k)}|$ is sufficiently small.

3. Some basic operations on tensors. This section provides a brief explanation of tensors. For further details, see, e.g., [1, 8, 11].

A tensor is a multidimensional array. In particular, a first-order tensor is a vector, a second-order tensor is a matrix, and a third-order tensor, which is mainly used in this paper, has three indices. Third-order tensors are denoted by $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{P}}$, $\boldsymbol{\mathcal{Q}}$, $\boldsymbol{\mathcal{R}}$, and $\boldsymbol{\mathcal{S}}$. An element (i, j, k) of a third-order tensor $\boldsymbol{\mathcal{X}}$ is denoted by x_{ijk}. When the size of a tensor $\boldsymbol{\mathcal{X}}$ is $I \times J \times K$, the ranges of i, j, and k are $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, and $k = 1, 2, \ldots, K$, respectively.

We describe the definitions of some basic operations on tensors. Let x_{ijk} and y_{ijk} be elements of the tensors $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I\times J\times K}$. Then, addition is defined by elementwise summation of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$:

$$(\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{Y}})_{ijk} := x_{ijk} + y_{ijk},$$

scalar tensor multiplication is defined by the product of the scalar λ and each element of $\boldsymbol{\mathcal{X}}$:

$$(\lambda\boldsymbol{\mathcal{X}})_{ijk} := \lambda x_{ijk},$$

and the dot product is defined by the summation of elementwise products of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$:

$$(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}) := \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}(\boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{Y}})_{ijk},$$

where the symbol "$*$" denotes the elementwise product. Then, the norm is defined as

$$\|\boldsymbol{\mathcal{X}}\| := \sqrt{(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}})}.$$

ETNA
Kent State University
http://etna.math.kent.edu

Let us define some tensor multiplications: an n-mode product, which is denoted by the symbol "$\times_n$", is a products of a matrix M and a tensor $\boldsymbol{\mathcal{X}}$. The n-mode product for a third-order tensor has three different types.

The 1-mode product of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ and $M \in \mathbb{R}^{P\times I}$ is defined as

$$(\boldsymbol{\mathcal{X}} \times_1 M)_{pjk} := \sum_{i=1}^{I} x_{ijk} m_{pi},$$

the 2-mode product of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ and $M \in \mathbb{R}^{P\times J}$ is defined as

$$(\boldsymbol{\mathcal{X}} \times_2 M)_{ipk} := \sum_{j=1}^{J} x_{ijk} m_{pj},$$

and the 3-mode product of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ and $M \in \mathbb{R}^{P\times K}$ is defined as

$$(\boldsymbol{\mathcal{X}} \times_3 M)_{ijp} := \sum_{k=1}^{K} x_{ijk} m_{pk},$$

where $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, and $p = 1, 2, \ldots, P$.

Finally, the operator "vec" vectorizes a tensor by combining all column vectors of the tensor into one long vector:

$$\text{vec}: \quad \mathbb{R}^{I\times J\times K} \to \mathbb{R}^{IJK},$$

and the operator "vec^{-1}" reshapes a tensor from one long vector:

$$\text{vec}^{-1}: \quad \mathbb{R}^{IJK} \to \mathbb{R}^{I\times J\times K}.$$

We will see that the vec^{-1}-operator plays an important role in reconstructing the Lanczos bidiagonalization method over tensor space.

4. The Lanczos bidiagonalization method over tensor space with a promising initial guess.

4.1. The Lanczos bidiagonalization method over tensor space. If the Lanczos bidiagonalization method is applied to the generalized tensor sum T in (1.1), the following matrix-vector multiplications are required:

$$\begin{aligned} T\boldsymbol{p} &= (I_n \otimes I_m \otimes A + I_n \otimes B \otimes I_\ell + C \otimes I_m \otimes I_\ell)\,\boldsymbol{p}, \\ T^{\mathrm{T}}\boldsymbol{p} &= \left(I_n \otimes I_m \otimes A^{\mathrm{T}} + I_n \otimes B^{\mathrm{T}} \otimes I_\ell + C^{\mathrm{T}} \otimes I_m \otimes I_\ell\right)\boldsymbol{p}, \end{aligned} \tag{4.1}$$

where $\boldsymbol{p} \in \mathbb{R}^{\ell mn}$. In an implementation, however, computing the multiplications (4.1) becomes complicated since it requires the non-zero structure of a large matrix T. Here, the relations between the vec-operator and the Kronecker product are represented by

$$\begin{aligned} (I_n \otimes I_m \otimes A)\,\text{vec}(\boldsymbol{\mathcal{P}}) &= \text{vec}(\boldsymbol{\mathcal{P}} \times_1 A), \\ (I_n \otimes B \otimes I_\ell)\,\text{vec}(\boldsymbol{\mathcal{P}}) &= \text{vec}(\boldsymbol{\mathcal{P}} \times_2 B), \\ (C \otimes I_m \otimes I_\ell)\,\text{vec}(\boldsymbol{\mathcal{P}}) &= \text{vec}(\boldsymbol{\mathcal{P}} \times_3 C), \end{aligned}$$

where $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{\ell\times m\times n}$ is such that $\text{vec}(\boldsymbol{\mathcal{P}}) = \boldsymbol{p}$. Using these relations, the multiplications (4.1) can be described by

$$\begin{aligned} T\boldsymbol{p} &= \text{vec}\left(\boldsymbol{\mathcal{P}} \times_1 A + \boldsymbol{\mathcal{P}} \times_2 B + \boldsymbol{\mathcal{P}} \times_3 C\right), \\ T^{\mathrm{T}}\boldsymbol{p} &= \text{vec}\left(\boldsymbol{\mathcal{P}} \times_1 A^{\mathrm{T}} + \boldsymbol{\mathcal{P}} \times_2 B^{\mathrm{T}} + \boldsymbol{\mathcal{P}} \times_3 C^{\mathrm{T}}\right). \end{aligned} \tag{4.2}$$

ETNA
Kent State University
http://etna.math.kent.edu

Then, an implementation based on (4.2) only requires the non-zero structures of the matrices A, B, and C being much smaller than of T, and thus it is simplified.

We now consider the Lanczos bidiagonalization method over tensor space. Applying the vec^{-1}-operator to (4.2) yields

$$\begin{aligned}\mathrm{vec}^{-1}(T\boldsymbol{p}) &= \boldsymbol{\mathcal{P}} \times_1 A + \boldsymbol{\mathcal{P}} \times_2 B + \boldsymbol{\mathcal{P}} \times_3 C,\\ \mathrm{vec}^{-1}\left(T^{\mathrm{T}}\boldsymbol{p}\right) &= \boldsymbol{\mathcal{P}} \times_1 A^{\mathrm{T}} + \boldsymbol{\mathcal{P}} \times_2 B^{\mathrm{T}} + \boldsymbol{\mathcal{P}} \times_3 C^{\mathrm{T}}.\end{aligned}$$

Then, the Lanczos bidiagonalization method over tensor space for T is obtained and summarized in Algorithm 2.

Algorithm 2 The Lanczos bidiagonalization method over tensor space.

Choose an initial tensor $\boldsymbol{\mathcal{P}}_1 \in \mathbb{R}^{\ell\times m\times n}$ such that $\|\boldsymbol{\mathcal{P}}_1\| = 1$.
$\boldsymbol{\mathcal{Q}}_1 := \boldsymbol{\mathcal{P}}_1 \times_1 A + \boldsymbol{\mathcal{P}}_1 \times_2 B + \boldsymbol{\mathcal{P}}_1 \times_3 C$;
$\alpha_1 := \|\boldsymbol{\mathcal{Q}}_1\|$;
$\boldsymbol{\mathcal{Q}}_1 := \boldsymbol{\mathcal{Q}}_1/\alpha_1$;
for $i = 1, 2, \ldots, k$ **do**
  $\boldsymbol{\mathcal{R}}_i := \boldsymbol{\mathcal{Q}}_i \times_1 A^{\mathrm{T}} + \boldsymbol{\mathcal{Q}}_i \times_2 B^{\mathrm{T}} + \boldsymbol{\mathcal{Q}}_i \times_3 C^{\mathrm{T}} - \alpha_i \boldsymbol{\mathcal{P}}_i$;
  $\beta_i := \|\boldsymbol{\mathcal{R}}_i\|$;
  $\boldsymbol{\mathcal{P}}_{i+1} := \boldsymbol{\mathcal{R}}_i/\beta_i$;
  $\boldsymbol{\mathcal{Q}}_{i+1} := \boldsymbol{\mathcal{P}}_{i+1} \times_1 A + \boldsymbol{\mathcal{P}}_{i+1} \times_2 B + \boldsymbol{\mathcal{P}}_{i+1} \times_3 C - \beta_i \boldsymbol{\mathcal{Q}}_i$;
  $\alpha_{i+1} := \|\boldsymbol{\mathcal{Q}}_{i+1}\|$;
  $\boldsymbol{\mathcal{Q}}_{i+1} := \boldsymbol{\mathcal{Q}}_{i+1}/\alpha_{i+1}$;
end for

The maximum/minimum singular values of T are computed by a singular value decomposition of the matrix D_k in (2.2), whose entries α_i and β_i are obtained from Algorithm 2. The convergence of Algorithm 2 can be monitored by

$$\|\boldsymbol{\mathcal{R}}_k\| \left| \boldsymbol{e}_k^{\mathrm{T}} \boldsymbol{u}_i^{(D_k)} \right|, \tag{4.3}$$

where $\boldsymbol{u}_i^{(D_k)}$ is computed by the singular value decomposition for D_k in (2.2).

4.2. A promising initial guess. We consider utilizing the eigenvectors of the small matrices A, B, and C for determining a promising initial guess for Algorithm 2. We propose the initial guess $\boldsymbol{\mathcal{P}}_1$ that is given in Tucker decomposition form:

$$\boldsymbol{\mathcal{P}}_1 := \boldsymbol{\mathcal{S}} \times_1 P_A \times_2 P_B \times_3 P_C, \tag{4.4}$$

where $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{2\times 2\times 2}$ is the core tensor such that $\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} s_{ijk} = 1$ and $s_{ijk} \geq 0$, $P_A = [\boldsymbol{x}_{i_{\mathrm{M}}}^{(A)}, \boldsymbol{x}_{i_{\mathrm{m}}}^{(A)}]$, $P_B = [\boldsymbol{x}_{j_{\mathrm{M}}}^{(B)}, \boldsymbol{x}_{j_{\mathrm{m}}}^{(B)}]$, and $P_C = [\boldsymbol{x}_{k_{\mathrm{M}}}^{(C)}, \boldsymbol{x}_{k_{\mathrm{m}}}^{(C)}]$. The rest of this section defines the vectors $\boldsymbol{x}_{i_{\mathrm{M}}}^{(A)}$, $\boldsymbol{x}_{j_{\mathrm{M}}}^{(B)}$, $\boldsymbol{x}_{k_{\mathrm{M}}}^{(C)}$, $\boldsymbol{x}_{i_{\mathrm{m}}}^{(A)}$, $\boldsymbol{x}_{j_{\mathrm{m}}}^{(B)}$, and $\boldsymbol{x}_{k_{\mathrm{m}}}^{(C)}$.

Let $\{\lambda_i^{(A)}, \boldsymbol{x}_i^{(A)}\}$, $\{\lambda_j^{(B)}, \boldsymbol{x}_j^{(B)}\}$, and $\{\lambda_k^{(C)}, \boldsymbol{x}_k^{(C)}\}$ be eigenpairs of the matrices A, B, and C, respectively. Then, $\boldsymbol{x}_{i_{\mathrm{M}}}^{(A)}$, $\boldsymbol{x}_{j_{\mathrm{M}}}^{(B)}$, and $\boldsymbol{x}_{k_{\mathrm{M}}}^{(C)}$ are the eigenvectors corresponding to the eigenvalues $\lambda_{i_{\mathrm{M}}}^{(A)}$, $\lambda_{j_{\mathrm{M}}}^{(B)}$, and $\lambda_{k_{\mathrm{M}}}^{(C)}$ of A, B, and C, where

$$(i_{\mathrm{M}}, j_{\mathrm{M}}, k_{\mathrm{M}}) = \operatorname*{argmax}_{(i,j,k)} \left\{ \left| \lambda_i^{(A)} + \lambda_j^{(B)} + \lambda_k^{(C)} \right| \right\}.$$

ETNA
Kent State University
http://etna.math.kent.edu

Similarly, $\boldsymbol{x}_{i_\mathrm{m}}^{(A)}$, $\boldsymbol{x}_{j_\mathrm{m}}^{(B)}$, and $\boldsymbol{x}_{k_\mathrm{m}}^{(C)}$ are the eigenvectors corresponding to the eigenvalues $\lambda_{i_\mathrm{m}}^{(A)}$, $\lambda_{j_\mathrm{m}}^{(B)}$, and $\lambda_{k_\mathrm{m}}^{(C)}$ of A, B, and C, where

$$(i_\mathrm{m}, j_\mathrm{m}, k_\mathrm{m}) = \operatorname*{argmin}_{(i,j,k)} \left\{ \left| \lambda_i^{(A)} + \lambda_j^{(B)} + \lambda_k^{(C)} \right| \right\}.$$

Here, we note that the eigenvector corresponding to the maximum absolute eigenvalue of T is given by $\boldsymbol{x}_{k_\mathrm{M}}^{(C)} \otimes \boldsymbol{x}_{j_\mathrm{M}}^{(B)} \otimes \boldsymbol{x}_{i_\mathrm{M}}^{(A)}$ and that the eigenvector corresponding to the minimum absolute eigenvalue of T is given by $\boldsymbol{x}_{k_\mathrm{m}}^{(C)} \otimes \boldsymbol{x}_{j_\mathrm{m}}^{(B)} \otimes \boldsymbol{x}_{i_\mathrm{m}}^{(A)}$.

5. Numerical examples. In this section, we report the results of numerical experiments using test matrices given below. All computations were carried out using MATLAB version R2011b on a HP Z800 workstation with two 2.66 GHz Xeon processors and 24GB of memory running under a Windows 7 operating system.

The maximum/minimum singular values were computed by Algorithm 2 with random initial guesses and with the proposed initial guess (4.4). From (4.3), the stopping criteria we used were $\|\boldsymbol{\mathcal{R}}_k\| |\boldsymbol{e}_k^\mathrm{T} \boldsymbol{u}_\mathrm{M}^{(D_k)}| < 10^{-10}$ for the maximum singular value $\sigma_\mathrm{M}^{(D_k)}$ and $\|\boldsymbol{\mathcal{R}}_k\| |\boldsymbol{e}_k^\mathrm{T} \boldsymbol{u}_\mathrm{m}^{(D_k)}| < 10^{-10}$ for the minimum singular value $\sigma_\mathrm{m}^{(D_k)}$. Algorithm 2 was stopped when both criteria were satisfied.

5.1. Test matrices. The test matrices T arise from a 7-point central difference discretization of the PDE (1.2) over an $(n+1) \times (n+1) \times (n+1)$ grid, and they are written as a generalized tensor sum of the form

$$T = I_n \otimes I_n \otimes A + I_n \otimes B \otimes I_n + C \otimes I_n \otimes I_n \in \mathbb{R}^{n^3 \times n^3},$$

where $A, B, C \in \mathbb{R}^{n \times n}$. To be specific, the matrices A, B, and C are given by

$$A = \frac{1}{h^2} a_1 M_1 + \frac{1}{2h} b_1 M_2 + c I_n, \tag{5.1}$$

$$B = \frac{1}{h^2} a_2 M_1 + \frac{1}{2h} b_2 M_2, \tag{5.2}$$

$$C = \frac{1}{h^2} a_3 M_1 + \frac{1}{2h} b_3 M_2, \tag{5.3}$$

where a_i and b_i $(i = 1, 2, 3)$ correspond to the i-th elements of $\boldsymbol{a}$ and $\boldsymbol{b}$ in (1.2), respectively, and h, M_1, and M_2 are given as

$$h = \frac{1}{n+1},$$

$$M_1 = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n \times n}, \tag{5.4}$$

$$M_2 = \begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{5.5}$$

As can be seen from (5.1)–(5.5), the matrix T has high symmetry when $\|\boldsymbol{a}\|_2$ is much larger than $\|\boldsymbol{b}\|_2$ and low symmetry else.

ETNA
Kent State University
http://etna.math.kent.edu

5.2. Initial guesses used in the numerical examples. In our numerical experiments, we set $\boldsymbol{\mathcal{S}}$ in (4.4) to be a diagonal tensor, i.e., $s_{ijk} = 0$ except for $i = j = k = 1$ and $i = j = k = 2$. Then, the proposed initial guess (4.4) is represented by the following convex combination of rank-one tensors:

$$\boldsymbol{\mathcal{P}}_1 = s_{111}\left(\boldsymbol{x}_{i_{\mathrm{M}}}^{(A)} \circ \boldsymbol{x}_{j_{\mathrm{M}}}^{(B)} \circ \boldsymbol{x}_{k_{\mathrm{M}}}^{(C)}\right) + s_{222}\left(\boldsymbol{x}_{i_{\mathrm{m}}}^{(A)} \circ \boldsymbol{x}_{j_{\mathrm{m}}}^{(B)} \circ \boldsymbol{x}_{k_{\mathrm{m}}}^{(C)}\right), \tag{5.6}$$

where the symbol "$\circ$" denotes the outer product and $s_{111} + s_{222} = 1$ with $s_{111}, s_{222} \geq 0$.

As seen in Section 4.2, the vectors $\boldsymbol{x}_{i_{\mathrm{M}}}^{(A)}$, $\boldsymbol{x}_{j_{\mathrm{M}}}^{(B)}$, $\boldsymbol{x}_{k_{\mathrm{M}}}^{(C)}$, $\boldsymbol{x}_{i_{\mathrm{m}}}^{(A)}$, $\boldsymbol{x}_{j_{\mathrm{m}}}^{(B)}$, and $\boldsymbol{x}_{k_{\mathrm{m}}}^{(C)}$ are determined by specific eigenvectors of the matrices A, B, and C. Since these matrices are tridiagonal Toeplitz matrices, it is widely known that the eigenvalues and eigenvectors are given in analytical form as follows: let D be a tridiagonal Toeplitz matrix

$$D = \begin{bmatrix} d_1 & d_3 & & & \\ d_2 & d_1 & d_3 & & \\ & \ddots & \ddots & \ddots & \\ & & d_2 & d_1 & d_3 \\ & & & d_2 & d_1 \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $d_2 \neq 0$ and $d_3 \neq 0$. Then, the eigenvalues $\lambda_i^{(D)}$ are computed by

$$\lambda_i^{(D)} = d_1 + 2d\cos\left(\frac{i\pi}{n+1}\right), \quad i = 1, 2, \ldots, n,$$

where $d = \operatorname{sign}(d_2)\sqrt{d_2 d_3}$, and the corresponding eigenvectors $\boldsymbol{x}_i^{(D)}$ are given by

$$\boldsymbol{x}_i^{(D)} = \sqrt{\frac{2}{n+1}} \begin{bmatrix} \left(\frac{d_3}{d_2}\right)^{-\frac{0}{2}} \sin\left(\frac{i\pi}{n+1}\right) \\ \left(\frac{d_3}{d_2}\right)^{-\frac{1}{2}} \sin\left(\frac{2i\pi}{n+1}\right) \\ \vdots \\ \left(\frac{d_3}{d_2}\right)^{-\frac{n-1}{2}} \sin\left(\frac{ni\pi}{n+1}\right) \end{bmatrix}, \quad i = 1, 2, \ldots, n.$$

5.3. Numerical results. First, we use the matrix T with the parameters $\boldsymbol{a} = (1, 1, 1)$, $\boldsymbol{b} = (1, 1, 1)$, and $c = 1$ in (5.1)–(5.3).

The convergence history of Algorithm 2 with the proposed initial guess (5.6) using $s_{111} = s_{222} = 0.5$ is displayed in Figure 5.1 with the number of iterations required by Algorithm 2 on the horizontal axis and the $\log_{10}$ of the residual norms on the vertical axis. Here, the i-th residual norms are computed by $\|\boldsymbol{\mathcal{R}}_i\||\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{u}_{\mathrm{M}}^{(D_i)}|$ for the maximum singular value and $\|\boldsymbol{\mathcal{R}}_i\||\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{u}_{\mathrm{m}}^{(D_i)}|$ for the minimum singular value, and the size of the tensor in Algorithm 2 is $20 \times 20 \times 20$.

As illustrated in Figure 5.1, Algorithm 2 with the proposed initial guess required 68 iterations for the maximum singular value and 460 iterations for the minimum singular value. From Figure 5.1, we observe a smooth convergence behavior and faster convergence for the maximum singular value.

Next, using the matrix T with the same parameters as for Figure 5.1, the variation in the number of iterations is displayed in Figure 5.2, where the dependence of the number

ETNA
Kent State University
http://etna.math.kent.edu

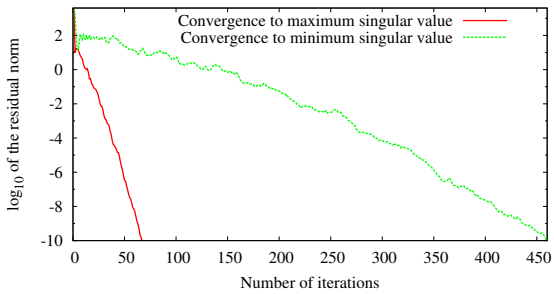
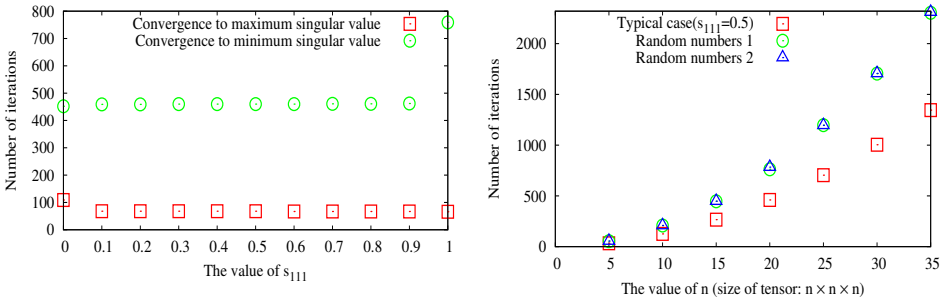


FIG. 5.1. *The convergence history for Algorithm 2 with the proposed initial guess.*

of iterations on the value s_{111} in the proposed initial guess is given in Figure 5.2(a), and the dependence on the tensor size is given in Figure 5.2(b). For comparison, the numbers of iterations required by Algorithm 2 with initial guesses being random numbers are also displayed in Figure 5.2(b). In Figure 5.2(a), the horizontal axis denotes the value of s_{111}, varying from 0 to 1 incrementally with a stepsize of 0.1, and the vertical axis denotes the number of iterations required by Algorithm 2 with the proposed initial guess. Here, the value of s_{222} is computed by $s_{222} = 1 - s_{111}$, and the matrix T used for Figure 5.2(a) is obtained from the discretization of the PDE (1.2) over a $21 \times 21 \times 21$ grid. On the other hand, in Figure 5.2(b), the horizontal axis denotes the value of n, where the size of the tensor is $n \times n \times n$ ($n = 5, 10, \ldots, 35$), and the vertical axis denotes the number of iterations required by Algorithm 2 with the proposed initial guess using $s_{111} = s_{222} = 0.5$. Here, this initial guess is referred to as the typical case.



(a) *Number of iteration versus the value of* s_{111}. (b) *Number of iteration versus the tensor size.*

FIG. 5.2. *The variation in the number of iterations for the matrix* T *with* $\boldsymbol{a} = \boldsymbol{b} = (1, 1, 1)$, $c = 1$.

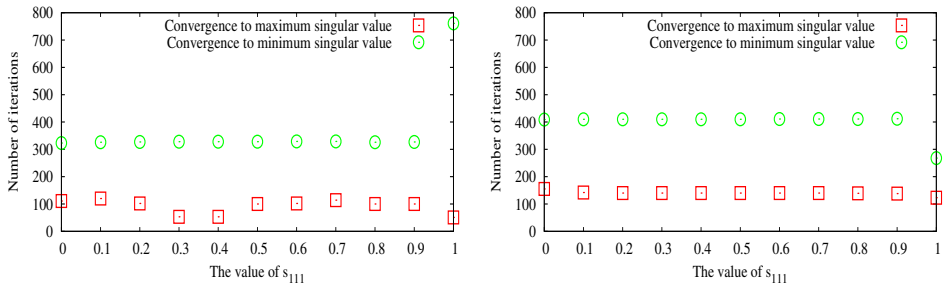
In Figure 5.2(a), the number of iterations hardly depends on the value of s_{111}, but there is a big difference in this number between the cases $s_{111} = 0.9$ and $s_{111} = 1$. We therefore ran Algorithm 2 with the proposed initial guess using the values $s_{111} = 0.90, 0.91, \ldots, 0.99$. As a result, we confirm that the numbers of iterations are almost the same as those for the cases $s_{111} = 0, 0.1, \ldots, 0.9$. From these results, we find almost no dependency of the number of iterations on the choice of the parameter s_{111}, and this implies the robustness of the proposed initial guess. It seems that the high number of iterations required for the case

ETNA
Kent State University
http://etna.math.kent.edu

$s_{111} = 1$ is due to that the given initial guess only has a very small component of the singular vector corresponding to the minimum singular value. In fact, for a symmetric matrix, such a choice means that the proposed initial guess includes no component of the singular vector corresponding to the minimum singular value. In Figure 5.2(b) we observe that Algorithm 2 in the typical case requires fewer iterations than with an initial guess of random numbers, and the gap grows as n increases.

In what follows, we use matrices T with higher or lower symmetry than for the matrices used in Figures 5.1 and 5.2. A matrix T with higher symmetry is created from the parameters $\boldsymbol{a} = (100, 100, 100)$, $\boldsymbol{b} = (1, 1, 1)$, and $c = 1$, and a matrix T with lower symmetry from the parameters $\boldsymbol{a} = (1, 1, 1)$, $\boldsymbol{b} = (100, 100, 100)$, and $c = 1$.

The variation in the number of iterations by the value of s_{111} in the proposed initial guess is presented in Figure 5.3. Here, the matrices T arise from the discretization of the PDE (1.2) with the above parameters over a $21 \times 21 \times 21$ grid.



(a) T *with higher symmetry.* (b) T *with lower symmetry.*

FIG. 5.3. *The variation in the number of iterations versus the value of* s_{111}.

In Figure 5.3, the variation in the number of iterations for the matrices with high and low symmetry showed similar tendencies as in Figure 5.2(a). Furthermore, the variation in the number of iterations required by Algorithm 2 with the proposed initial guess using the values $s_{111} = 0.90, 0.91, \ldots, 0.99$ in Figure 5.3 has the same behavior as that in Figure 5.2(a). For the low-symmetry case, the choice $s_{111} = 1$ was optimal unlike the other cases.

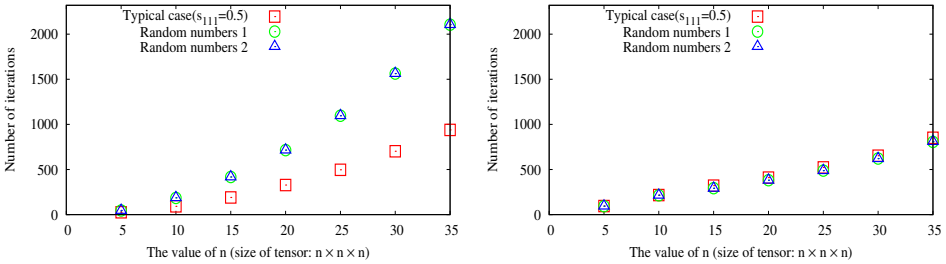
In the following example, the variation in the number of iterations versus the tensor size is displayed in Figure 5.4. For comparison, we ran Algorithm 2 with several initial guesses: random numbers and the proposed one with the typical case $s_{111} = 0.5$.

According to Figure 5.4(a), Algorithm 2 in the typical case required fewer iterations than for the initial guesses using random numbers when T had higher symmetry. On the other hand, Figure 5.4(b) indicates that Algorithm 2 in the typical case required as many iterations as for a random initial guess when T had lower symmetry. From Figures 5.2(b) and 5.4, we observe that the initial guess using $0 \leq s_{111} < 1$ improves the speed of convergence of Algorithm 2 except for the case where T has lower symmetry.

As can be observed in Figure 5.4(b), the typical case shows no advantage over the random initial guess for the low-symmetry matrix. On the other hand, for some cases the proposed initial guess could still become a method of choice, for instance, for the case $s_{111} = 1$ displayed in Figure 5.5.

It is likely, though it requiring further investigation, that the result in Figure 5.5 indicates a potential for improvement of the proposed initial guess (4.4) even for the low-symmetry case.

ETNA
Kent State University
http://etna.math.kent.edu



(a) T *with higher symmetry.* (b) T *with lower symmetry.*

FIG. 5.4. *The variation in the number of iterations versus the tensor size.*

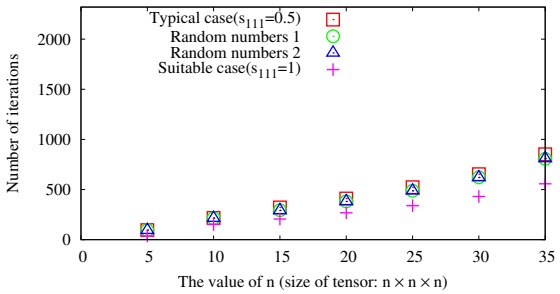


FIG. 5.5. *The variation in the number of iterations required by Algorithm 2 in the suitable case versus the tensor size when T has lower symmetry.*

In fact, we only used a diagonal tensor as initial guess, which is a subtensor of the core tensor. For low-symmetry matrices, an experimental investigation of the optimal choice of a full core tensor will be considered in future work.

6. Concluding remarks. In this paper, first, we derived the Lanczos bidiagonalization method over tensor space from the conventional Lanczos bidiagonalization method using the vec^{-1}-operator in order to compute the maximum/minimum singular values of a generalized tensor sum T. The resulting method achieved a low memory requirement and a very simple implementation since it only required the non-zero structure of the matrices A, B, and C.

Next, we proposed an initial guess given in Tucker decomposition form using eigenvectors corresponding to the maximum/minimum eigenvalues of T. Computing the eigenvectors of T was easy since the eigenpairs of T were obtained from the eigenpairs of A, B, and C.

Finally, from the results of the numerical experiments, we showed that the maximum/minimum singular values of T were successfully computed by the Lanczos bidiagonalization method over tensor space with some of the proposed initial guesses. We see that the proposed initial guesses improved the speed of convergence of the Lanczos bidiagonalization method over tensor space for the high-symmetry case and that it could become a method of choice for other cases if a suitable core tensor can be found.

Future work is devoted to an experimental investigations using the full core tensor in the proposed initial guess in order to choose an optimal initial guess for low-symmetry matrices. If the generalized tensor sum (1.1) is sufficiently close to a symmetric matrix, our initial guess

ETNA
Kent State University
http://etna.math.kent.edu

works very well, but in general, restarting techniques are important for a further improvement of the speed of convergence to the minimum singular value. In this case, restarting techniques should be combined not in vector- but in tensor space. Thus, constructing a general framework in tensor space and combining Algorithm 2 with successful restarting techniques, e.g., [2, 6, 7] are topics of future work. With regards to other methods, the presented approach may be applied to other successful variants of the Lanczos bidiagonalization method, e.g., [10] and to Jacobi-Davidson-type singular value decomposition methods, e.g., [5].

Acknowledgments. This work has been supported in part by JSPS KAKENHI (Grant No. 26286088). We wish to express our gratitude to Dr. D. Savostyanov, University of Southampton, for constructive comments at the NASCA2013 conference. We are grateful to Dr. T. S. Usuda and Dr. H. Yoshioka of Aichi Prefectural University for their support and encouragement. We would like to thank the anonymous referees for informing us of reference [11] and many useful comments that enhanced the quality of the manuscript.

REFERENCES

[1] B. W. BADER AND T. G. KOLDA, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653.
[2] J. BAGLAMA AND L. REICHEL, *An implicitly restarted block Lanczos bidiagonalization method using Leja shifts*, BIT Numer. Math., 53 (2013), pp. 285–310.
[3] J. BALLANI AND L. GRASEDYCK, *A projection method to solve linear systems in tensor format*, Numer. Linear Algebra Appl., 20 (2013), pp. 27–43.
[4] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.
[5] M. E. HOCHSTENBACH, *A Jacobi-Davidson type method for the generalized singular value problem*, Linear Algebra Appl., 431 (2009), pp. 471–487.
[6] Z. JIA AND D. NIU, *A refined harmonic Lanczos bidiagonalization method and an implicitly restarted algorithm for computing the smallest singular triplets of large matrices*, SIAM J. Sci. Comput., 32 (2010), pp. 714–744.
[7] E. KOKIOPOULOU, C. BEKAS, AND E. GALLOPOULOS, *Computing smallest singular triplets with implicitly restarted Lanczos bidiagonalization*, Appl. Numer. Math., 49 (2004), pp. 39–61.
[8] T. G. KOLDA AND B. W. BADER, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[9] D. KRESSNER AND C. TOBLER, *Krylov subspace methods for linear systems with tensor product structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1688–1714.
[10] D. NIU AND X. YUAN, *A harmonic Lanczos bidiagonalization method for computing interior singular triplets of large matrices*, Appl. Math. Comput., 218 (2012), pp. 7459–7467.
[11] B. SAVAS AND L. ELDEN, *Krylov-type methods for tensor computations I*, Linear Algebra Appl., 438 (2013), pp. 891–918.
[12] M. STOLL, *A Krylov-Schur approach to the truncated SVD*, Linear Algebra Appl., 436 (2012), pp. 2795–2806.