Fast Hankel Tensor-Vector Product and its Application to Exponential Data Fitting[†]

Weiyang Ding[1], Liqun Qi[2*], and Yimin Wei[3]

[1] *School of Mathematical Sciences, Fudan University, Shanghai, 200433, P. R. of China.*
[2]*Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.*
[3] *School of Mathematical Sciences and Shanghai Key Laboratory of Contemporary Applied Mathematics, Fudan University, Shanghai, 200433, P. R. of China.*

SUMMARY

This paper is contributed to a fast algorithm for Hankel tensor-vector products. First we explain the necessity of fast algorithms for Hankel and block Hankel tensor-vector products by sketching the algorithm for both one- and multi-dimensional exponential data fitting. For proposing the fast algorithm, we define and investigate a special class of Hankel tensors that can be diagonalized by the Fourier matrices, which is called anti-circulant tensors. Then we obtain a fast algorithm for Hankel tensor-vector products by embedding a Hankel tensor into a larger anti-circulant tensor. The computational complexity is about $\mathcal{O}(m^2 n \log mn)$ for a square Hankel tensor of order m and dimension n, and the numerical examples also show the efficiency of this scheme. Moreover, the block version for multi-level block Hankel tensors is discussed as well. Copyright © 2014 John Wiley & Sons, Ltd.

KEY WORDS: Hankel tensor; block Hankel tensor; anti-circulant tensor; fast tensor-vector product; fast Fourier transform; higher-order singular value decomposition; exponential data fitting

1. Introduction

Hankel structures arise frequently in signal processing (see [1]). Besides Hankel matrices, tensors with different Hankel structures are also applied. As far as we know, the term "Hankel tensor" was first introduced by Luque and Thibon [2]. And Badeau and Boyer [3] discussed the higher-order singular value decompositions (HOSVD) of some structured tensors including Hankel tensors in detail. Moreover, Papy et al. employed Hankel tensors and other Hankel-type tensors in exponential data fitting (see [4, 5, 6]). De Lathauwer [7] also concerned the "separation" of signals that can be modelled as sums of exponentials (or more generally, as exponential polynomials) by

---

Hankel tensor approaches. As to the properties of Hankel tensors, Qi [8] recently investigated the spectral properties of Hankel tensor largely via the generating function. Song and Qi [9] also proposed some spectral properties of Hilbert tensors, which are special Hankel tensors.

An mth-order tensor $\mathcal{H} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_m}$ is called a *Hankel tensor* if

$$\mathcal{H}_{i_1 i_2 \ldots i_m} = \phi(i_1 + i_2 + \cdots + i_m)$$

for all $i_k = 0, 1, \ldots, n_k - 1$ ($k = 1, 2, \ldots, m$). We call $\mathcal{H}$ a *square Hankel tensor* when $n_1 = n_2 = \cdots = n_m$. Note that the degree of freedom of a Hankel tensor is $d_{\mathcal{H}} := n_1 + n_2 + \cdots + n_m - m + 1$. Thus a vector $\mathbf{h}$ of length $d_{\mathcal{H}}$, which is called the *generating vector* of $\mathcal{H}$, defined by

$$h_k = \phi(k), \ k = 0, 1, \ldots, d_{\mathcal{H}} - 1$$

can completely determine this Hankel tensor $\mathcal{H}$ when the tensor size is fixed. Further when the entries of $\mathbf{h}$ can be written as

$$h_k = \int_{-\infty}^{+\infty} t^k f(t) \mathrm{d}t,$$

then we call $f(t)$ the *generating function* of $\mathcal{H}$. The generating function of a square Hankel tensor is essential for studying its eigenvalues, positive semi-definiteness, and copositiveness etc. (see [8]).

Fast algorithms for Hankel or Toeplitz matrix-vector products involving fast Fourier transformations (FFT) are well-known (see [10, 11, 12, 13]). However the topics on Hankel tensor computations are seldom discussed. We propose an analogous fast algorithm for Hankel tensor-vector products, which has its application to exponential data fitting. We first introduce Papy et al.'s algorithm for one-dimensional exponential data fitting briefly and extend it to multi-dimensional case in Section 2, in which the Hankel and block Hankel tensor-vector products are dominant steps in the sense of efficiency. Then in Section 3 we define the *anti-circulant tensor*, which can be diagonalized by the Fourier matrices, and study its properties carefully. In Section 4, we propose a fast algorithm for Hankel and block Hankel tensor-vector products by embedding them into a larger anti-circulant and block anti-circulant tensors, respectively. Finally we employ some numerical examples to show the efficiency of our scheme in Section 5.

2. Exponential Data Fitting

We begin with one of the sources of Hankel tensors, and see where we need fast Hankel tensor-vector products. Exponential data fitting is very important in many applications in scientific computing and engineering, which represents the signals as a sum of exponentially damped sinusoids. The computations and applications of exponential data fitting are generally studied, and the readers who are interested in these topics can refer to [14, 15, 16].

Papy et al. [5, 6] introduced a higher-order tensor approach into exponential data fitting by connecting it with the Vandermonde decomposition of a Hankel tensor. As stated in [5], their algorithm is a higher-order variant of the Hankel total least squares (HTLS) method. And HTLS is a modification of the famous ESPRIT algorithm [17, 18] by employing the total least squares (TLS, see [19]) instead of the least squares, which enhances the robustness since the TLS is a type of errors-in-variables regression. Furthermore, Papy et al. concluded from numerical experiments in their papers that the Hankel tensor approach can perform better for some difficult situations than the classical one based on Hankel matrices, although there is no exact theory on how to choose the optimal size of the Hankel tensor.

In order to understand the necessity of fast algorithms for Hankel and block Hankel tensors, we sketch Papy et al.'s algorithm in this section and simply extend it to multi-dimensional exponential data fitting.

2.1. The One-Dimensional Case

Assume that we get a one-dimensional noiseless signal with N complex samples $\{x_n\}_{n=0}^{N-1}$, and this signal is modelled as a sum of K exponentially damped complex sinusoids, i.e.,

$$x_n = \sum_{k=1}^{K} a_k \exp(\imath \varphi_k) \exp\big((-\alpha_k + \imath \omega_k)n\Delta t\big),$$

where $\imath = \sqrt{-1}$, Δt is the sampling interval, and the amplitudes a_k, the phases φ_k, the damping factors α_k, and the pulsations ω_k are the parameters of the model that are required to estimate. The signal can also be expressed as

$$x_n = \sum_{k=1}^{K} c_k z_k^n,$$

where $c_k = a_k \exp(\imath \varphi_k)$ and $z_k = \exp\big((-\alpha_k + \imath \omega_k)\Delta t\big)$. Here c_k is called the k-th complex amplitude including the phase and z_k is called the k-th pole of the signal. A part of the aim of exponential data fitting is to estimate the poles $\{z_k\}_{k=1}^{K}$ from the data $\{x_n\}_{n=0}^{N-1}$. After fixing the poles, we can obtain the complex amplitudes by solving a Vandermonde system.

Denote vector $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})^\top$. We first construct a Hankel tensor $\mathcal{H}$ of a fixed size $I_1 \times I_2 \times \cdots \times I_m$ with the generating vector $\mathbf{x}$, e.g., when $m = 3$ the Hankel tensor $\mathcal{H}$ is

$$\mathcal{H}_{:,:,1} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{I_2-2} & x_{I_2-1} \\ x_1 & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & x_{I_1+I_2-3} \\ x_{I_1-1} & x_{I_1} & \cdots & x_{I_1+I_2-3} & x_{I_1+I_2-2} \end{bmatrix},$$

$$\mathcal{H}_{:,:,2} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{I_2-1} & x_{I_2} \\ x_2 & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & x_{I_1+I_2-2} \\ x_{I_1} & x_{I_1+1} & \cdots & x_{I_1+I_2-2} & x_{I_1+I_2-1} \end{bmatrix}, \tag{1}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\mathcal{H}_{:,:,I_3} = \begin{bmatrix} x_{I_3-1} & x_{I_3} & \cdots & x_{I_2+I_3-3} & x_{I_2+I_3-2} \\ x_{I_3} & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & x_{I_1+I_2+I_3-4} \\ x_{I_1+I_3-2} & x_{I_1+I_3-1} & \cdots & x_{I_1+I_2+I_3-4} & x_{I_1+I_2+I_3-3} \end{bmatrix}.$$

The order m can be chosen arbitrarily and the size I_p of each dimension should be no less than K and satisfy $I_1 + I_2 + \cdots + I_m - m + 1 = N_1$. Papy et al. verified that the Vandermonde decomposition of $\mathcal{H}$ is

$$\mathcal{H} = \mathcal{C} \times_1 Z_1^\top \times_2 Z_2^\top \cdots \times_m Z_m^\top,$$

where $\mathcal{C}$ is a diagonal tensor with diagonal entries $\{c_k\}_{k=1}^{K}$, each Z_p is a Vandermonde matrix

$$Z_p^\top = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{I_p-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{I_p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_K & z_K^2 & \cdots & z_K^{I_p-1} \end{bmatrix},$$

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

Numer. Linear Algebra Appl. 2014; **00**:1–18

and $\mathcal{T} \times_p M$ denotes the mode-p product (see [20]) of $\mathcal{T} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_m}$ and $M \in \mathbb{C}^{n_p \times l_p}$ defined by

$$(\mathcal{T} \times_p M)_{i_1 \ldots i_{p-1} j_p i_{p+1} \ldots i_m} = \sum_{i_p=1}^{n_p} t_{i_1 \ldots i_{p-1} i_p i_{p+1} \ldots i_m} m_{i_p j_p}.$$

So the target will be attained if we obtain the Vandermonde decomposition of this Hankel tensor $\mathcal{H}$.

In [5] the Vandermonde matrices are estimated by applying the TLS to the factor matrices in the higher-order singular value decomposition (HOSVD, see [21, 20]) of the best rank-$(R, R, \ldots, R)$ approximation (see [22, 20]) of $\mathcal{H}$. Here if K is known, then $R = K$. Otherwise, when K is unknown, R should be chosen to be much larger than a guess of K. Therefore, computing the HOSVD of the best low rank approximation of a Hankel tensor is a main part in exponential data fitting.

De Lathauwer et al. [22] proposed an effective algorithm called higher-order orthogonal iterations (HOOI) for this purpose. There are other algorithms with faster convergence such as [23, 24] proposed, and one can refer to [20] for more details. Nevertheless, HOOI is still very popular, because it is so simple and effective in applications. Thus Papy et al. chose HOOI in [5].

The original HOOI algorithm for general tensors is displayed as follows, and the result $\mathcal{S} \times_1 U_1^\top \times_2 U_2^\top \cdots \times_m U_m^\top$ is the best rank-$(R_1, R_2, \ldots, R_m)$ approximation of $\mathcal{A}$.

Algorithm 2.1. *HOOI for the best rank-$(R_1, R_2, \ldots, R_m)$ approximation of $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_m}$.*
  Initialize $U_p \in \mathbb{C}^{I_p \times R_p}$ $(p = 1, 2, \ldots, m)$ by the HOSVD of $\mathcal{A}$
  Repeat
      for $p = 1 : m$
          $U_p \leftarrow R_p$ leading left singular vectors of
                  $\mathrm{Unfold}_p(\mathcal{A} \times_1 \bar{U}_1 \cdots \widehat{\times_p \bar{U}_p} \cdots \times_m \bar{U}_m)$
      end
  Until convergence
  $\mathcal{S} = \mathcal{A} \times_1 \bar{U}_1 \times_2 \bar{U}_2 \cdots \times_m \bar{U}_m.$

Here $\mathrm{Unfold}_p(\cdot)$ denotes the mode-p unfolding of a tensor (see [19]), and $\mathcal{A} \times_1 \bar{U}_1 \cdots \widehat{\times_p \bar{U}_p} \cdots \times_m \bar{U}_m$ means that we skip the p-th item. There are plenty of tensor-matrix products in the above algorithm, which can be complemented by tensor-vector products. For instance, the tensor-matrix product

$$(\mathcal{A} \times_2 \bar{U}_2 \cdots \times_m \bar{U}_m)_{:,i_2,\ldots,i_m} = \mathcal{A} \times_2 (\bar{U}_2)_{:,i_2} \cdots \times_m (\bar{U}_m)_{:,i_m},$$

and others are the same. Therefore, if all the Hankel tensor-vector products can be computed by a fast algorithm, then the efficiency must be highly raised when we invoke the HOOI algorithm in exponential data fitting.

Papy et al. also studied the multi-channel case and the decimative case of exponential data fitting in [5, 6] using higher-order approaches. The tensors arise from these cases are not exactly Hankel tensors, but have some Hankel structures. A tensor is called *partially Hankel tensor*, if the lower-order subtensors are all Hankel tensors when some indexes are fixed. For instance, the tensor $\mathcal{H}$ from the multi-channel or decimative exponential data fitting is 3rd-order, and $\mathcal{H}(:,:,k)$ are Hankel tensors for all k, so we call $\mathcal{H}$ a 3rd-order $(1, 2)$-Hankel tensor. The HOOI algorithm will be applied to some partially Hankel tensors. Hence the fast tensor-vector products for partially Hankel tensors are also required to discuss.

2.2. The Multi-Dimensional Case

Papy et al.'s method can be extended to multi-dimensional exponential data fitting as well, which involves the block tensors. Similarly to block matrices, a *block tensor* is understood as a tensor

whose entries are also tensors. The size of each block is called the *level*-1 *size* of the block tensor, and the size of the block-entry tensor is called the *level*-2 *size*. Furthermore, a *level-d block tensor* can be regarded as a tensor whose entries are level-$(d-1)$ block tensor.

We take the 2-dimensional exponential data fitting (see [25, 26]) as an example to illustrate our block tensor approach. Assume that there is a 2D noiseless signal with $N_1 \times N_2$ complex samples $\{x_{n_1 n_2}\}_{\substack{n_1=0,1,\ldots,N_1-1 \\ n_2=0,1,\ldots,N_2-1}}$ which is modeled as a sum of K exponential items

$$x_{n_1 n_2} = \sum_{k=1}^{K} a_k \exp(\iota\varphi_k) \exp\left((-\alpha_{1,k} + \iota\omega_{1,k})n_1\Delta t_1 + (-\alpha_{2,k} + \iota\omega_{2,k})n_2\Delta t_2\right),$$

where the meanings of parameters are the same as those of 1D signals. Also, this 2D signal can be rewritten into a compact form

$$x_{n_1 n_2} = \sum_{k=1}^{K} c_k z_{1,k}^{n_1} z_{2,k}^{n_2}.$$

Our aim is still to estimate the poles $\{z_{1,k}\}_{k=1}^{K}$ and $\{z_{2,k}\}_{k=1}^{K}$ of the signal from the samples. We shall see shortly that the extended Papy et al.'s algorithm can also be regarded as a modified version of the 2D ESPRIT method [25].

Denote matrix $X = (x_{n_1 n_2})_{N_1 \times N_2}$. Then we map the data X into a block Hankel tensor with Hankel blocks (BHHB tensor) $\mathcal{H}$ of level-1 size $I_1 \times I_2 \times \cdots \times I_m$ and level-2 size $J_1 \times J_2 \times \cdots \times J_m$. The size I_p and J_p of each dimension should be no less than K and satisfy that $I_1 + I_2 + \cdots + I_m - m + 1 = N_1$ and $J_1 + J_2 + \cdots + J_m - m + 1 = N_2$. First we construct Hankel tensors $\mathcal{H}_j$ of size $I_1 \times I_2 \times \cdots \times I_m$ with the generating vectors $X(:,j)$ for $j = 0, 1, \ldots, N_2 - 1$ as shown in (1). Then, in block sense, we construct block Hankel tensors $\mathcal{H}$ of size $J_1 \times J_2 \times \cdots \times J_m$ with the block generating vectors $[\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{N_2-1}]^\top$, e.g., when $m = 3$ the slices in block sense of BHHB tensor $\mathcal{H}$ are

$$\mathcal{H}_{:,:,1}^{(b)} = \begin{bmatrix} \mathcal{H}_0 & \mathcal{H}_1 & \cdots & \mathcal{H}_{J_2-2} & \mathcal{H}_{J_2-1} \\ \mathcal{H}_1 & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \mathcal{H}_{J_1+J_2-3} \\ \mathcal{H}_{J_1-1} & \mathcal{H}_{J_1} & \cdots & \mathcal{H}_{J_1+J_2-3} & \mathcal{H}_{J_1+J_2-2} \end{bmatrix},$$

$$\mathcal{H}_{:,:,2}^{(b)} = \begin{bmatrix} \mathcal{H}_1 & \mathcal{H}_2 & \cdots & \mathcal{H}_{J_2-1} & \mathcal{H}_{J_2} \\ \mathcal{H}_2 & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \mathcal{H}_{J_1+J_2-2} \\ \mathcal{H}_{J_1} & \mathcal{H}_{J_1+1} & \cdots & \mathcal{H}_{J_1+J_2-2} & \mathcal{H}_{J_1+J_2-1} \end{bmatrix}, \tag{2}$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\mathcal{H}_{:,:,J_3}^{(b)} = \begin{bmatrix} \mathcal{H}_{J_3-1} & \mathcal{H}_{J_3} & \cdots & \mathcal{H}_{J_2+J_3-3} & \mathcal{H}_{J_2+J_3-2} \\ \mathcal{H}_{J_3} & \iddots & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \mathcal{H}_{J_1+J_2+J_3-4} \\ \mathcal{H}_{J_1+J_3-2} & \mathcal{H}_{J_1+J_3-1} & \cdots & \mathcal{H}_{J_1+J_2+J_3-4} & \mathcal{H}_{J_1+J_2+J_3-3} \end{bmatrix}.$$

Then the BHHB tensor $\mathcal{H}$ has the level-2 Vandermonde decomposition

$$\mathcal{H} = \mathcal{C} \times_1 (Z_{2,1} \underset{\mathrm{KR}}{\otimes} Z_{1,1})^\top \times_2 (Z_{2,2} \underset{\mathrm{KR}}{\otimes} Z_{1,2})^\top \cdots \times_m (Z_{2,m} \underset{\mathrm{KR}}{\otimes} Z_{1,m})^\top,$$

where $\mathcal{C}$ is a diagonal tensor with diagonal entries $\{c_k\}_{k=1}^{K}$, each $Z_{1,p}$ or $Z_{2,p}$ is a Vandermonde

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

Numer. Linear Algebra Appl. 2014; **00**:1–18

matrix

$$
Z_{1,p}^\top = \begin{bmatrix} 1 & z_{1,1} & z_{1,1}^2 & \cdots & z_{1,1}^{I_p-1} \\ 1 & z_{1,2} & z_{1,2}^2 & \cdots & z_{1,2}^{I_p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{1,K} & z_{1,K}^2 & \cdots & z_{1,K}^{I_p-1} \end{bmatrix}, \; Z_{2,p}^\top = \begin{bmatrix} 1 & z_{2,1} & z_{2,1}^2 & \cdots & z_{2,1}^{J_p-1} \\ 1 & z_{2,2} & z_{2,2}^2 & \cdots & z_{2,2}^{J_p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{2,K} & z_{2,K}^2 & \cdots & z_{2,K}^{J_p-1} \end{bmatrix},
$$

and the notation "$\underset{\mathrm{KR}}{\otimes}$" denotes the *Khatri-Rao product* (see [19, Chapter 12.3]) of two matrices with the same column sizes, i.e.,

$$
[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \underset{\mathrm{KR}}{\otimes} [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n] = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_2 \otimes \mathbf{b}_2, \ldots, \mathbf{a}_n \otimes \mathbf{b}_n].
$$

So the target will be attained, if we obtain the level-2 Vandermonde decomposition of this BHHB tensor $\mathcal{H}$.

We can use HOOI as well to compute the best rank-$(K, K, \ldots, K)$ approximation of the BHHB tensor $\mathcal{H}$

$$
\mathcal{H} = \mathcal{S} \times_1 U_1^\top \times_2 U_2^\top \cdots \times_m U_m^\top,
$$

where $\mathcal{S} \in \mathbb{C}^{K \times K \times \cdots \times K}$ is the core tensor and $U_p \in \mathbb{C}^{(I_p J_p) \times K}$ has orthogonal columns. Then U_p and $Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p}$ have the common column space, that is, there is a nonsingular matrix T such that

$$
Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p} = U_p T.
$$

Denote

$$
\begin{aligned}
A^{1\uparrow} &= \left[A_{0:I-2,:}^\top, A_{I,2I-2,:}^\top, \ldots, A_{(J-1)I:JI-2,:}^\top \right]^\top, \\
A^{1\downarrow} &= \left[A_{1:I-1,:}^\top, A_{I+1,2I-1,:}^\top, \ldots, A_{(J-1)I+1:JI-1,:}^\top \right]^\top, \\
A^{2\uparrow} &= A_{0:(J-1)I-1,:}, \\
A^{2\downarrow} &= A_{I:JI-1,:},
\end{aligned}
$$

for matrix $A \in \mathbb{C}^{(IJ) \times K}$. Then it is easy to verify that

$$
(Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p})^{1\uparrow} D_1 = (Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p})^{1\downarrow}, \quad (Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p})^{2\uparrow} D_2 = (Z_{2,p} \underset{\mathrm{KR}}{\otimes} Z_{1,p})^{2\downarrow},
$$

where D_1 is a diagonal matrix with diagonal entries $\{z_{1,k}\}_{k=1}^K$ and D_2 is a diagonal matrix with diagonal entries $\{z_{2,k}\}_{k=1}^K$. Then we have

$$
U_p^{1\uparrow}(T D_1 T^{-1}) = U_p^{1\downarrow}, \quad U_p^{2\uparrow}(T D_2 T^{-1}) = U_p^{2\downarrow}.
$$

Therefore, if two matrices W_1 and W_2 satisfy that

$$
U_p^{1\uparrow} W_1 = U_p^{1\downarrow}, \quad U_p^{2\uparrow} W_2 = U_p^{2\downarrow},
$$

then W_1 and W_2 share the same eigenvalues with D_1 and D_2, respectively. Equivalently, the eigenvalues of W_1 and W_2 are exactly the poles of the first and second dimension, respectively. Furthermore, we also choose the TLS as in [5] for solving the above two equations, since the noise on both sides should be taken into consideration.

Unlike to the 2D ESPRIT method, we obtain the poles of both dimensions by introducing only one BHHB tensor rather than constructing two related BHHB matrices. Hence the matrices W_1 and W_2 have the same eigenvectors. This information is useful for finding the assignment of the poles, that is, the eigenvalues of W_1 and W_2 with a same eigenvector is assigned into a pair.

Recall that Algorithm 2.1 is also called for BHHB tensors in 2D exponential data fitting. Thus the fast algorithm for BHHB tensor-vector products is also essential for this situation. Moreover, when we deal with the exponential data fitting problems of higher dimensions, i.e., 3D, 4D, etc., higher-level block Hankel tensors will be naturally involved. Therefore, it is also required to derive a unified fast algorithm for higher-level block Hankel tensor-vector products.

3. Anti-Circulant Tensors

The fast algorithm for Hankel tensor-vector products is based on a class of special Hankel tensors called *anti-circulant tensor*. Thus we introduce and investigate the anti-circulant tensors at first.

Circulant matrix [27] is famous, which is a special class of Toeplitz matrices [10, 12]. The first column entries of a circulant matrix shift down when moving right, as shown in the following 3-by-3 example

$$\begin{bmatrix} c_0 & c_2 & c_1 \\ c_1 & c_0 & c_2 \\ c_2 & c_1 & c_0 \end{bmatrix}.$$

If the first column entries of a matrix shift up when moving right, as shown in the following

$$\begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_0 \\ c_2 & c_0 & c_1 \end{bmatrix},$$

then it is a special Hankel matrix, which is called an *anti-circulant matrix*, or a *left circulant matrix*, or a *retrocirculant matrix* (see [27, Chapter 5]). Naturally, we generalize the anti-circulant matrix to the tensor case. A square Hankel tensor $\mathcal{C}$ of order m and dimension n is called an *anti-circulant tensor*, if its generating vector $\mathbf{h}$ satisfies that

$$h_k = h_l, \quad \text{if } k \equiv l \,(\mathrm{mod}\, n).$$

Thus the generating vector is periodic and displayed as

$$\mathbf{h} = (\underbrace{h_0, h_1, \ldots, h_{n-1}}_{\mathbf{c}^\top}, \underbrace{h_n, h_{n+1}, \ldots, h_{2n-1}}_{\mathbf{c}^\top}, \ldots, \underbrace{h_{(m-1)n}, \ldots, h_{m(n-1)}}_{\mathbf{c}(0:n-m)^\top})^\top.$$

Since the vector $\mathbf{c}$, which is exactly the "first" column $\mathcal{C}(:, 0, \cdots, 0)$, contains all the information about $\mathcal{C}$ and is more compact than the generating vector, we call it the *compressed generating vector* of the anti-circulant tensor. For instance, a $3 \times 3 \times 3$ anti-circulant tensor $\mathcal{C}$ is unfolded by mode-1 into

$$\mathrm{Unfold}_1(\mathcal{C}) = \left[\begin{array}{ccc|ccc|ccc} c_0 & c_1 & c_2 & c_1 & c_2 & c_0 & c_2 & c_0 & c_1 \\ c_1 & c_2 & c_0 & c_2 & c_0 & c_1 & c_0 & c_1 & c_2 \\ c_2 & c_0 & c_1 & c_0 & c_1 & c_2 & c_1 & c_2 & c_0 \end{array} \right],$$

and its compressed generating vector is $\mathbf{c} = [c_0, c_1, c_2]^\top$. Note that the degree of freedom of an anti-circulant tensor is always n no matter how large its order m will be.

3.1. Diagonalization

One of the essential properties of circulant matrices is that every circulant matrix can be diagonalized by the Fourier matrix (see [27]), where the Fourier matrix of size n is defined as $F_n = \left(\exp(-\frac{2\pi i}{n} jk) \right)_{j,k=0,1,\ldots,n-1}$. Actually, the Fourier matrix is exactly the Vandermonde matrix for the roots of unity, and it is also a unitary matrix up to the normalization factor

$$F_n F_n^* = F_n^* F_n = n I_n,$$

where I_n is the identity matrix of $n \times n$ and F_n^* is the conjugate transpose of F_n. We will show that anti-circulant tensors also have a similar property, which brings much convenience for both analysis and computations.

In order to describe this property, we recall the definition of mode-p tensor-matrix product first. It should be pointed out that the tensor-matrix products in this paper are slightly different with some standard notations (see [21, 20]) just for easy use and simple descriptions. In the standard notation system, two indices in "M_{ij}" should be exchanged. There are some basic properties of the tensor-matrix products

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

Numer. Linear Algebra Appl. 2014; **00**:1–18

1. $\mathcal{A} \times_p M_p \times_q M_q = \mathcal{A} \times_q M_q \times_p M_p$, if $p \neq q$,
2. $\mathcal{A} \times_p M_{p_1} \times_p M_{p_2} = \mathcal{A} \times_p (M_{p_1} M_{p_2})$,
3. $\mathcal{A} \times_p M_{p_1} + \mathcal{A} \times_p M_{p_2} = \mathcal{A} \times_p (M_{p_1} + M_{p_2})$,
4. $\mathcal{A}_1 \times_p M + \mathcal{A}_2 \times_p M = (\mathcal{A}_1 + \mathcal{A}_2) \times_p M$.

Particularly, when $\mathcal{A}$ is a matrix, the mode-1 and mode-2 products can be written as

$$\mathcal{A} \times_1 M_1 \times_2 M_2 = M_1^\top \mathcal{A} M_2.$$

Notice that $M_1^\top \mathcal{A} M_2$ is totally different with $M_1^* \mathcal{A} M_2$! ($M_1^\top$ is the transpose of M_1.) We will also adopt some notations from Qi's paper [28, 29] that

$$\mathcal{A}\mathbf{x}^{m-1} = \mathcal{A} \qquad \times_2 \mathbf{x} \cdots \times_m \mathbf{x},$$
$$\mathcal{A}\mathbf{x}^m = \mathcal{A} \times_1 \mathbf{x} \times_2 \mathbf{x} \cdots \times_m \mathbf{x}.$$

We are now ready to state our main result about anti-circulant tensors.

Theorem 3.1. *A square tensor of order m and dimension n is an anti-circulant tensor if and only if it can be diagonalized by the Fourier matrix of size n, that is,*

$$\mathcal{C} = \mathcal{D}F_n^m := \mathcal{D} \times_1 F_n \times_2 F_n \cdots \times_m F_n,$$

where $\mathcal{D}$ is a diagonal tensor and $\mathrm{diag}(\mathcal{D}) = \mathrm{ifft}(\mathbf{c})$. Here "ifft" is a Matlab-type symbol, an abbreviation of inverse fast Fourier transformation.

Proof. It is direct to verify that a tensor that can be expressed as $\mathcal{D}F_n^m$ is anti-circulant. Thus we only need to prove that every anti-circulant tensor can be written into this form. And this can be done constructively.

First, assume that an anti-circulant tensor $\mathcal{C}$ could be written into $\mathcal{D}F_n^m$. Then how do we obtain the diagonal entries of $\mathcal{D}$ from $\mathcal{C}$? Since

$$\mathrm{diag}(\mathcal{D}) = \mathcal{D}\mathbf{1}^{m-1} = \frac{1}{n^m}\big(\mathcal{C}(F_n^*)^m\big)\mathbf{1}^{m-1} = \frac{1}{n^m}\bar{F}_n\big(\mathcal{C}(F_n^*\mathbf{1})^{m-1}\big)$$
$$= \frac{1}{n}\bar{F}_n(\mathcal{C}\mathbf{e}_0^{m-1}) = \frac{1}{n}\bar{F}_n\mathbf{c},$$

where $\mathbf{1} = [1, 1, \ldots, 1]^\top$, $\mathbf{e}_0 = [1, 0, \ldots, 0]^\top$, $\bar{F}_n$ is the conjugate of F_n, and $\mathbf{c}$ is the compressed generating vector of $\mathcal{C}$, then the diagonal entries of $\mathcal{D}$ can be computed by an inverse fast Fourier transformation (IFFT)

$$\mathrm{diag}(\mathcal{D}) = \mathrm{ifft}(\mathbf{c}).$$

Finally, it is enough to check that $\mathcal{C} = \mathcal{D}F^m$ with $\mathrm{diag}(\mathcal{D}) = \mathrm{ifft}(\mathbf{c})$ directly. Therefore, every anti-circulant tensor is diagonalized by the Fourier matrix of proper size. $\square$

From the expression $\mathcal{C} = \mathcal{D}F_n^m$, we have a corollary about the spectra of anti-circulant tensors. The definitions of tensor Z-eigenvalues and H-eigenvalues follow the ones in [28, 29].

Corollary 3.2. *An anti-circulant tensor $\mathcal{C}$ of order m and dimension n with the compressed generating vector $\mathbf{c}$ has a Z-eigenvector/H-eigenvector $\frac{1}{\sqrt{n}}\mathbf{1}$, and the corresponding Z-eigenvalue is $n^{\frac{m-2}{2}}\mathbf{1}^\top\mathbf{c}$ and the corresponding H-eigenvalue is $n^{m-2}\mathbf{1}^\top\mathbf{c}$. When n is even, it has another Z-eigenvector $\frac{1}{\sqrt{n}}\widetilde{\mathbf{1}}$, where $\widetilde{\mathbf{1}} = [1, -1, \ldots, 1, -1]^\top$, and the corresponding Z-eigenvalue is $n^{\frac{m-2}{2}}\widetilde{\mathbf{1}}^\top\mathbf{c}$; Moreover, this is also an H-eigenvector if m is even, and the corresponding H-eigenvalue is $n^{m-2}\widetilde{\mathbf{1}}^\top\mathbf{c}$.*

Proof. It is easy to check that

$$\mathcal{C}\mathbf{1}^{m-1} = F_n^\top\big(\mathcal{D}(F_n\mathbf{1})^{m-1}\big) = n^{m-1}F_n^\top(\mathcal{D}\mathbf{e}_0^{m-1}) = n^{m-1}\mathcal{D}_{1,1,\ldots,1} \cdot F_n^\top\mathbf{e}_0$$
$$= n^{m-2}(\mathbf{e}_0^\top\bar{F}_n\mathbf{c})\mathbf{1} = (n^{m-2}\mathbf{1}^\top\mathbf{c})\mathbf{1}.$$

The proof of the rest part is similar, so we omit it. $\square$

3.2. Singular Values

Lim [30] defined the tensor singular values as

$$
\begin{cases}
\mathcal{A} \times_2 \mathbf{u}_2 \times_3 \mathbf{u}_3 \cdots \times_m \mathbf{u}_m = \varphi_{p_1}(\mathbf{u}_1) \cdot \sigma, \\
\mathcal{A} \times_1 \mathbf{u}_1 \times_3 \mathbf{u}_3 \cdots \times_m \mathbf{u}_m = \varphi_{p_2}(\mathbf{u}_2) \cdot \sigma, \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\
\mathcal{A} \times_1 \mathbf{u}_1 \times_2 \mathbf{u}_2 \cdots \times_{m-1} \mathbf{u}_{m-1} = \varphi_{p_m}(\mathbf{u}_m) \cdot \sigma,
\end{cases}
$$

where $\sigma \geq 0$ and $\mathbf{u}_l^\top \varphi_{p_l}(\mathbf{u}_l) = \|\mathbf{u}_l\|_{p_l}^{p_l} = 1$ for $l = 1, 2, \ldots, m$. When $p_1 = p_2 = \cdots = p_m = 2$, $\varphi_2(\mathbf{u}) = \bar{\mathbf{u}}$ and the singular values are unitarily invariant.

Consider the singular values of anti-circulant tensors. Let $\mathcal{C} = \mathcal{D}F^m$ be an anti-circulant tensor. There exists a permutation matrix P such that the diagonal entries of $\mathcal{D}P^m$ are arranged in descending order by their absolute values, then $\mathcal{C} = (\mathcal{D}P^m)\big((FP)^\top\big)^m$. Denote Λ is a diagonal matrix satisfying that $\Lambda_{kk}^m = \mathrm{sgn}\big((\mathcal{D}P^m)_{kk}\big)$, where $\mathrm{sgn}(\cdot)$ denotes the signum function, i.e.,

$$
\mathrm{sgn}(\xi) = \begin{cases} \xi/|\xi|, & \xi \neq 0, \\ 0, & \xi = 0. \end{cases}
$$

Hence it is easy to understand that tensor $\mathcal{C}$ can be rewritten into $\mathcal{C} = \widetilde{\mathcal{D}}(V^\top)^m$, where $\widetilde{\mathcal{D}} = |\mathcal{D}P^m|$ is a nonnegative diagonal tensor with ordered diagonal entries and $V = FP\Lambda$ is a unitary matrix. If $\{\sigma; \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\}$ is a singular value and the corresponding singular vectors of $\mathcal{C}$, then

$$
\big\{\sigma; V^\top \mathbf{u}_1, V^\top \mathbf{u}_2, \ldots, V^\top \mathbf{u}_m\big\}
$$

is a singular value and the associated singular vectors of $\widetilde{\mathcal{D}}$, and vice versa. Therefore, we need only to find the singular values and singular vectors of a diagonal tensor $\widetilde{\mathcal{D}}$. Let $d_1 \geq d_2 \geq \cdots \geq d_n \geq 0$ be the diagonal entries of $\widetilde{\mathcal{D}}$ and $\mathbf{w}_l = V^\top \mathbf{u}_l$ for $l = 1, 2, \ldots, m$. Then the above equations become into

$$
\begin{cases}
d_k(\mathbf{w}_2)_k(\mathbf{w}_3)_k \ldots (\mathbf{w}_m)_k = (\bar{\mathbf{w}}_1)_k \cdot \sigma, \\
d_k(\mathbf{w}_1)_k(\mathbf{w}_3)_k \ldots (\mathbf{w}_m)_k = (\bar{\mathbf{w}}_2)_k \cdot \sigma, \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\
d_k(\mathbf{w}_1)_k(\mathbf{w}_2)_k \ldots (\mathbf{w}_{m-1})_k = (\bar{\mathbf{w}}_m)_k \cdot \sigma,
\end{cases} \quad k = 1, 2, \ldots, n.
$$

From the above equations we have for $k = 1, 2, \ldots, n$,

$$
d_k(\mathbf{w}_1)_k(\mathbf{w}_2)_k \ldots (\mathbf{w}_m)_k = |(\mathbf{w}_1)_k|^2 \cdot \sigma = |(\mathbf{w}_2)_k|^2 \cdot \sigma = \cdots = |(\mathbf{w}_m)_k|^2 \cdot \sigma.
$$

Then $|\mathbf{w}_1| = |\mathbf{w}_2| = \cdots = |\mathbf{w}_m| := \mathbf{q} = [q_1, q_1, \ldots, q_n]^\top$ when $\sigma \neq 0$. Denote $K = \{k : q_k \neq 0\}$. Then $d_k q_k^{m-2} = \sigma$. Since $\mathbf{q}$ is normalized, we have $d_k > 0$ and $\sum_{k \in K} (\sigma/d_k)^{\frac{2}{m-2}} = 1$. Thus the singular value is

$$
\sigma = \Big(\sum_{k \in K} d_k^{-\frac{2}{m-2}} \Big)^{-\frac{m-2}{2}},
$$

and the singular vectors are determined by

$$
q_k = \begin{cases} (\sigma/d_k)^{\frac{1}{m-2}}, & k \in K, \\ 0, & \text{otherwise}, \end{cases} \quad \text{and } \mathrm{sgn}(\mathbf{w}_1)_k \mathrm{sgn}(\mathbf{w}_2)_k \ldots \mathrm{sgn}(\mathbf{w}_m)_k = 1.
$$

Therefore, if $d_1 \geq \cdots \geq d_r > d_{r+1} = \cdots = d_n = 0$, then the anti-circulant tensor $\mathcal{C}$ has at most $2^r - 1$ nonzero singular values when $m > 2$, since the index set K can be chosen as an arbitrary subset of $\{1, 2, \ldots, n\}$. As to the zero singular value, the situation is a little more complicated. It is directly verified that the above equations hold for some k if there are two of $\{(\mathbf{w}_1)_k, (\mathbf{w}_2)_k, \ldots, (\mathbf{w}_m)_k\}$ equal to zero. Furthermore, for $k = r+1, r+2, \ldots, n$, the k-th entries of $\mathbf{w}_l$'s can also be chosen such that

$$
\mathrm{sgn}(\mathbf{w}_1)_k \mathrm{sgn}(\mathbf{w}_2)_k \ldots \mathrm{sgn}(\mathbf{w}_m)_k = 1.
$$

One can easily prove that the largest singular value of a nonnegative diagonal tensor is

$$
\begin{aligned}
d_1 &= \max \sigma(\widetilde{\mathcal{D}}) \\
&= \max \left\{ |\widetilde{\mathcal{D}} \times_1 \mathbf{w}_1 \cdots \times_m \mathbf{w}_m| : \|\mathbf{w}_1\|_2 = \cdots = \|\mathbf{w}_m\|_2 = 1 \right\}.
\end{aligned}
$$

So we also have that the largest singular value of an anti-circulant tensor

$$
\begin{aligned}
d_1 &= \max \sigma(\mathcal{C}) \\
&= \max \left\{ |\mathcal{C} \times_1 \mathbf{u}_1 \cdots \times_m \mathbf{u}_m| : \|\mathbf{u}_1\|_2 = \cdots = \|\mathbf{u}_m\|_2 = 1 \right\},
\end{aligned}
$$

and the maximum value can be attained when $\mathbf{u}_1 = \mathbf{u}_2 = \cdots = \mathbf{u}_m = \bar{V}\mathbf{e}_0$. Recall the definition of the tensor Z-eigenvalues (see [28])

$$
\begin{cases}
\mathcal{C}\mathbf{x}^{m-1} = \lambda\mathbf{x}, \\
\mathbf{x}^\top \mathbf{x} = 1,
\end{cases}
$$

where $\mathbf{x} \in \mathbb{R}^n$, then $\lambda = \mathcal{C}\mathbf{x}^m$. So the maximum absolute value of an anti-circulant tensor's Z-eigenvalues is bounded by the largest singular value, that is,

$$
\rho_Z(\mathcal{C}) := \{|\lambda| : \lambda \text{ is a Z-eigenvalue of } \mathcal{C}\} \le d_1.
$$

Particularly, when the anti-circulant tensor $\mathcal{C}$ is further nonnegative, which is equally that its compressed generating vector $\mathbf{c}$ is nonnegative, it can be verified that

$$
\mathrm{ifft}(\mathbf{c})_1 = \max_k |\mathrm{ifft}(\mathbf{c})_k|
$$

where $\mathrm{ifft}(\mathbf{c})_k$ denotes the k-th entry of $\mathrm{ifft}(\mathbf{c})$. So the singular vectors corresponding to the largest singular value are $\mathbf{u}_1 = \mathbf{u}_2 = \cdots = \mathbf{u}_m = \frac{1}{\sqrt{n}}\mathbf{1}$. Note that $\frac{1}{\sqrt{n}}\mathbf{1}$ is also a Z-eigenvector of $\mathcal{C}$ (see Corollary 3.2). Therefore, the Z-spectral radius of a nonnegative anti-circulant tensor is exactly its largest singular value.

3.3. Block Tensors

Block structures arise in a variety of applications in scientific computing and engineering (see [31, 1]). We have utilized the block tensors to multi-dimensional data fitting in Section 2.2.

If a block tensor can be regarded as a Hankel tensor or an anti-circulant tensor with tensor entries, then we call it a *block Hankel tensor* or a *block anti-circulant tensor*, respectively. Moreover, its generating vector $\mathbf{h}^{(b)}$ or compressed generating vector $\mathbf{c}^{(b)}$ in block sense is called the *block generating vector* or *block compressed generating vector*, respectively. For instance, the block-entry vector $[\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{N_2-1}]^\top$ is the block generating vector of $\mathcal{H}$ in Section 2.2. Recall the definition of Kronecker product [19]

$$
A \otimes B = \begin{bmatrix}
A_{11}B & A_{12}B & \cdots & A_{1q}B \\
A_{21}B & A_{22}B & \cdots & A_{2q}B \\
\vdots & \vdots & \ddots & \vdots \\
A_{p1}B & A_{p2}B & \cdots & A_{pq}B
\end{bmatrix},
$$

where A and B are two matrices of arbitrary sizes. Then it can be proved following Theorem 3.1 that a block anti-circulant tensor $\mathcal{C}$ can be block-diagonalized by $F_N \otimes I$, that is,

$$
\mathcal{C} = \mathcal{D}^{(b)}(F_N \otimes I)^m,
$$

where $\mathcal{D}^{(b)}$ is a block diagonal tensor with diagonal blocks $\mathbf{c}^{(b)} \times_1 \left(\frac{1}{N}\bar{F}_N \otimes I\right)$ and $\bar{F}_N$ is the conjugate of F_N.

Furthermore, when the blocks of a block Hankel tensor are also Hankel tensors, we call it a *block Hankel tensor with Hankel blocks*, or BHHB tensor for short. Then its block generating vector can be reduced to a matrix, which is called the *generating matrix H of a BHHB tensor*

$$H = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{N_1 + \cdots + N_m - m}] \in \mathbb{C}^{(n_1 + n_2 + \cdots + n_m - m + 1) \times (N_1 + N_2 + \cdots + N_m - m + 1)},$$

where $\mathbf{h}_k$ is the generating vector of the k-th Hankel block in $\mathbf{h}^{(b)}$. For instance, the data matrix X is exactly the generating matrix of the BHHB tensor $\mathcal{H}$ in Section 2.2. Similarly, when the blocks of a block anti-circulant tensor are also anti-circulant tensors, we call it a *block anti-circulant tensor with anti-circulant blocks*, or BAAB tensor for short. Its *compressed generating matrix C* is defined by

$$C = [\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{N-1}] \in \mathbb{C}^{n \times N},$$

where $\mathbf{c}_k$ is the compressed generating vector of the k-th anti-circulant block in the block compressed generating vector $\mathbf{c}^{(b)}$. We can also verify that a BAAB tensor $\mathcal{C}$ can be diagonalized by $F_N \otimes F_n$, that is,

$$\mathcal{C} = \mathcal{D}(F_N \otimes F_n)^m,$$

where $\mathcal{D}$ is a diagonal tensor with diagonal $\mathrm{diag}(\mathcal{D}) = \frac{1}{nN}\mathrm{vec}(\bar{F}_n C \bar{F}_N)$, which can be computed by 2D inverse fast Fourier transformation (IFFT2). Here $\mathrm{vec}(\cdot)$ denotes the vectorization operator (see [19]).

We can even define higher-level block Hankel tensors. For instance, a block Hankel tensor with BHHB blocks is called a *level-3 block Hankel tensor*, and it is easily understood that a level-3 block Hankel tensor has the generating tensor of order 3. Generally, a block Hankel or anti-circulant tensor with level-(k-1) block Hankel or anti-circulant blocks is called a *level-k block Hankel or anti-circulant tensor*, respectively. Furthermore, a level-k block anti-circulant tensor $\mathcal{C}$ can be diagonalized by $F_{n^{(k)}} \otimes F_{n^{(k-1)}} \otimes \cdots \otimes F_{n^{(1)}}$, that is,

$$\mathcal{C} = \mathcal{D}(F_{n^{(k)}} \otimes F_{n^{(k-1)}} \otimes \cdots \otimes F_{n^{(1)}})^m,$$

where $\mathcal{D}$ is a diagonal tensor with diagonal that can be computed by multi-dimensional inverse fast Fourier transformation.

4. Fast Hankel Tensor-Vector Product

General tensor-vector products without structures are very expensive for the high order and the large size that a tensor could be of. For a square tensor $\mathcal{A}$ of order m and dimension n, the computational complexity of a tensor-vector product $\mathcal{A}\mathbf{x}^{m-1}$ or $\mathcal{A}\mathbf{x}^m$ is $\mathcal{O}(n^m)$. However, since Hankel tensors and anti-circulant tensors have very low degrees of freedom, it can be expected that there is a much faster algorithm for Hankel tensor-vector products. We focus on the following two types of tensor-vector products

$$\mathbf{y} = \mathcal{A} \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m \text{ and } \alpha = \mathcal{A} \times_1 \mathbf{x}_1 \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m,$$

which will be extremely useful to applications.

The fast algorithm for anti-circulant tensor-vector products is easy to derive from Theorem 3.1. Let $\mathcal{C} = \mathcal{D}F_n^m$ be an anti-circulant tensor of order m and dimension n with the compressed generating vector $\mathbf{c}$. Then for vectors $\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_m \in \mathbb{C}^n$, we have

$$\mathbf{y} = \mathcal{C} \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m = F_n(\mathcal{D} \times_2 F_n\mathbf{x}_2 \cdots \times_m F_n\mathbf{x}_m).$$

Recall that $\mathrm{diag}(\mathcal{D}) = \mathrm{ifft}(\mathbf{c})$ and $F_n\mathbf{v} = \mathrm{fft}(\mathbf{v})$, where "fft" is a Matlab-type symbol, an abbreviation of fast Fourier transformation. So the fast procedure for computing the vector $\mathbf{y}$ is

$$\mathbf{y} = \mathrm{fft}\big(\mathrm{ifft}(\mathbf{c}).*\mathrm{fft}(\mathbf{x}_2).*\cdots.*\mathrm{fft}(\mathbf{x}_m)\big),$$

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

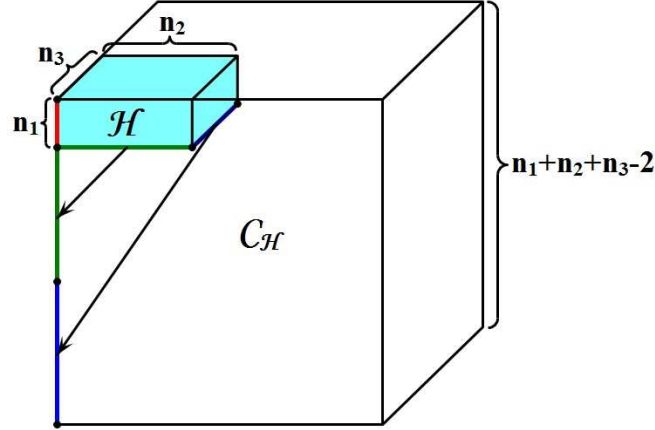*Numer. Linear Algebra Appl.* 2014; **00**:1–18

Figure 1. Embed a Hankel tensor into an anti-circulant tensor.

where $\mathbf{u} . * \mathbf{v}$ multiplies two vectors element-by-element. Similarly, for vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m \in \mathbb{C}^n$, we have

$$\alpha = \mathcal{C} \times_1 \mathbf{x}_1 \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m = \mathcal{D} \times_1 F_n \mathbf{x}_1 \times_2 F_n \mathbf{x}_2 \cdots \times_m F_n \mathbf{x}_m,$$

and the fast procedure for computing the scalar α is

$$\alpha = \text{ifft}(\mathbf{c})^\top \big(\text{fft}(\mathbf{x}_1). * \text{fft}(\mathbf{x}_2). * \cdots . * \text{fft}(\mathbf{x}_m)\big).$$

Since the computational complexity of either FFT or IFFT is $\mathcal{O}(n \log n)$, both two types of anti-circulant tensor-vector products can be obtained with complexity $\mathcal{O}\big((m + 1)n \log n\big)$, which is much faster than the product of a general n-by-n matrix with a vector.

For deriving the fast algorithm for Hankel tensor-vector products, we embed a Hankel tensor into a larger anti-circulant tensor. Let $\mathcal{H} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_m}$ be a Hankel tensor with the generating vector $\mathbf{h}$. Denote $\mathcal{C}_\mathcal{H}$ as the anti-circulant tensor of order m and dimension $d_\mathcal{H} = n_1 + n_2 + \cdots + n_m - m + 1$ with the compressed generating vector $\mathbf{h}$. Then we will find out that $\mathcal{H}$ is in the "upper left frontal" corner of $\mathcal{C}_\mathcal{H}$ as shown in Figure 1. Hence, we have

$$\mathcal{C}_\mathcal{H} \times_2 \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{0} \end{bmatrix} \cdots \times_m \begin{bmatrix} \mathbf{x}_m \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathcal{H} \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m \\ \dagger \end{bmatrix},$$

$$\mathcal{C}_\mathcal{H} \times_1 \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{0} \end{bmatrix} \cdots \times_m \begin{bmatrix} \mathbf{x}_m \\ \mathbf{0} \end{bmatrix} = \mathcal{H} \times_1 \mathbf{x}_1 \cdots \times_m \mathbf{x}_m,$$

so that the Hankel tensor-vector products can be realized by multiplying a larger anti-circulant tensor by some augmented vectors. Therefore, the fast procedure for computing $\mathbf{y} = \mathcal{H} \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m$ is

$$\begin{cases} \widetilde{\mathbf{x}}_p = [\mathbf{x}_p^\top, \underbrace{0, 0, \ldots, 0}_{d_\mathcal{H} - n_p}]^\top, \ p = 2, 3, \ldots, m, \\ \widetilde{\mathbf{y}} = \text{fft}\big(\text{ifft}(\mathbf{h}). * \text{fft}(\widetilde{\mathbf{x}}_2). * \cdots . * \text{fft}(\widetilde{\mathbf{x}}_m)\big), \\ \mathbf{y} = \widetilde{\mathbf{y}}(0 : n_1 - 1), \end{cases}$$

and the fast procedure for computing $\alpha = \mathcal{H} \times_1 \mathbf{x}_1 \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m$ is

$$\begin{cases} \widetilde{\mathbf{x}}_p = [\mathbf{x}_p^\top, \underbrace{0, 0, \ldots, 0}_{d_\mathcal{H} - n_p}]^\top, \ p = 1, 2, \ldots, m, \\ \alpha = \text{ifft}(\mathbf{h})^\top \big(\text{fft}(\widetilde{\mathbf{x}}_1). * \text{fft}(\widetilde{\mathbf{x}}_2). * \cdots . * \text{fft}(\widetilde{\mathbf{x}}_m)\big). \end{cases}$$

Moreover, the computational complexity is $\mathcal{O}\big((m+1)d_{\mathcal{H}}\log d_{\mathcal{H}}\big)$. When the Hankel tensor is a square tensor, the complexity is at the level $\mathcal{O}(m^2 n\log mn)$, which is much smaller than the complexity $\mathcal{O}(n^m)$ of non-structured products.

Apart from the low computational complexity, our algorithm for Hankel tensor-vector products has two advantages. One is that this scheme is compact, that is, there is no redundant element in the procedure. It is not required to form the Hankel tensor explicitly. Just the generating vector is needed. Another advantage is that our algorithm treats the tensor as an ensemble instead of multiplying the tensor by vectors mode by mode.

For BAAB and BHHB cases, we also have fast algorithms for the tensor-vector products. Let $\mathcal{C}$ be a BAAB tensor of order m with the compressed generating matrix $C\in\mathbb{C}^{n\times N}$. Since $\mathcal{C}$ can be diagonalized by $F_N\otimes F_n$, i.e.,

$$\mathcal{C}=\mathcal{D}(F_N\otimes F_n)^m,$$

we have for vectors $\mathbf{x}_2,\mathbf{x}_3,\ldots,\mathbf{x}_m\in\mathbb{C}^{nN}$

$$\mathbf{y}=\mathcal{C}\times_2\mathbf{x}_2\cdots\times_m\mathbf{x}_m=(F_N\otimes F_n)\big(\mathcal{D}\times_2(F_N\otimes F_n)\mathbf{x}_2\cdots\times_m(F_N\otimes F_n)\mathbf{x}_m\big).$$

Recall the vectorization operator and its inverse operator

$$\text{vec}(A)=[A_{:,0}^\top,A_{:,1}^\top,\ldots,A_{:,N-1}^\top]^\top\in\mathbb{C}^{nN},$$

$$\text{vec}_{n,N}^{-1}(\mathbf{v})=[\mathbf{v}_{0:n-1},\mathbf{v}_{n:2n-1},\ldots,\mathbf{v}_{(N-1)n:Nn-1}]\in\mathbb{C}^{n\times N},$$

for matrix $A\in\mathbb{C}^{n\times N}$ and vector $\mathbf{v}\in\mathbb{C}^{nN}$, and the relation holds

$$(B\otimes A)\mathbf{v}=\text{vec}\big(A\cdot\text{vec}_{n,N}^{-1}(\mathbf{v})\cdot B^\top\big).$$

So $(F_N\otimes F_n)\mathbf{x}_p=\text{vec}\big(F_n\cdot\text{vec}_{n,N}^{-1}(\mathbf{x}_p)\cdot F_N\big)$ can be computed by 2D fast Fourier transformation (FFT2). Then the fast procedure for computing $\mathbf{y}=\mathcal{C}\times_2\mathbf{x}_2\cdots\times_m\mathbf{x}_m$ is

$$\begin{cases} X_p=\text{vec}_{n,N}^{-1}(\mathbf{x}_p),\ p=2,3,\ldots,m,\\ Y=\text{fft2}\big(\text{ifft2}(C).*\text{fft2}(X_2).*\cdots.*\text{fft2}(X_m)\big),\\ \mathbf{y}=\text{vec}(Y), \end{cases}$$

and the fast procedure for computing $\alpha=\mathcal{C}\times_1\mathbf{x}_1\times_2\mathbf{x}_2\cdots\times_m\mathbf{x}_m$ is

$$\begin{cases} X_p=\text{vec}_{n,N}^{-1}(\mathbf{x}_p),\ p=1,2,\ldots,m,\\ \alpha=\big\langle\text{ifft2}(C),\text{fft2}(X_1).*\text{fft2}(X_2).*\cdots.*\text{fft2}(X_m)\big\rangle, \end{cases}$$

where $\langle A,B\rangle$ denotes

$$\langle A,B\rangle=\sum_{j,k}A_{jk}B_{jk}.$$

For a BHHB tensor $\mathcal{H}$ with the generating matrix H, we do the embedding twice. First we embed each Hankel block into a larger anti-circulant block, and then we embed the block Hankel tensor with anti-circulant blocks into a BAAB tensor $\mathcal{C}_{\mathcal{H}}$ in block sense. Notice that the compressed generating matrix of $\mathcal{C}_{\mathcal{H}}$ is exactly the generating matrix of $\mathcal{H}$. Hence we have the fast procedure for computing $\mathbf{y}=\mathcal{H}\times_2\mathbf{x}_2\cdots\times_m\mathbf{x}_m$

$$\begin{cases} \widetilde{X}_p=\underbrace{\begin{bmatrix}\text{vec}_{n_p,N_p}^{-1}(\mathbf{x}_p) & \text{O}\\ \text{O} & \text{O}\end{bmatrix}}_{N_1+N_2+\cdots+N_m-m+1}\Big\}{\scriptstyle n_1+n_2+\cdots+n_m-m+1},\ p=2,3,\ldots,m,\\ \widetilde{Y}=\text{fft2}\big(\text{ifft2}(H).*\text{fft2}(\widetilde{X}_2).*\cdots.*\text{fft2}(\widetilde{X}_m)\big),\\ \mathbf{y}=\text{vec}\big(\widetilde{Y}(0:n_1-1,0:N_1-1)\big). \end{cases}$$

Sometimes in applications there is no need to do the vectorization in the last line, and we just keep it as a matrix for later use. We also have the fast procedure for computing $\alpha = \mathcal{H} \times_1 \mathbf{x}_1 \times_2 \mathbf{x}_2 \cdots \times_m \mathbf{x}_m$

$$\begin{cases} \widetilde{X}_p = \underbrace{\begin{bmatrix} \mathrm{vec}_{n_p,N_p}^{-1}(\mathbf{x}_p) & \mathrm{O} \\ \mathrm{O} & \mathrm{O} \end{bmatrix}}_{N_1+N_2+\cdots+N_m-m+1} \Big\} n_1+n_2+\cdots+n_m-m+1, \ p = 1, 2, \ldots, m, \\ \alpha = \big\langle \mathrm{ifft2}(H), \mathrm{fft2}(\widetilde{X}_1).*\mathrm{fft2}(\widetilde{X}_2).*\cdots.*\mathrm{fft2}(\widetilde{X}_m) \big\rangle. \end{cases}$$

Similarly, we can also derive the fast algorithms for higher-level block Hankel tensor-vector products using the multi-dimensional FFT.

5. Numerical Examples

In this section, we will verify the effect of our fast algorithms for Hankel and block Hankel tensor-vector products by several numerical examples.

We first construct

- 3rd-order square Hankel tensors of size $n \times n \times n$ ($n = 10, 20, \ldots, 100$), and
- 3rd-order square BHHB tensors of level-1 size $n_1 \times n_1 \times n_1$ and level-2 size $n_2 \times n_2 \times n_2$ ($n_1, n_2 = 5, 6, \ldots, 12$).

Then compute the tensor-vector products $\mathcal{H} \times_2 \mathbf{x}_2 \times_3 \mathbf{x}_3$ using both our fast algorithm and the non-structured algorithm directly based on the definition. The average running times of 1000 products are shown in Figure 2. From the results, we can see that the running time of our algorithm increases far more slowly than that of the non-structured algorithm just as the theoretical analysis. Moreover, the difference in running times is not only the low computational complexity, but also the absence of forming the Hankel or BHHB tensors explicitly in our algorithm.

Next, we shall apply our algorithm to the problems from exponential data fitting in order to show its efficiency, several of which are borrowed from [5, 6]. We do the experiments for both the one-dimensional case and the two-dimensional case:

- A one-dimensional signal is modelled as

$$x_n = \exp\big((-0.01 + 2\pi \imath 0.20)n\big) + \exp\big((-0.02 + 2\pi \imath 0.22)n\big) + e_n,$$

  where e_n is a complex white Gaussian noise (WGN).
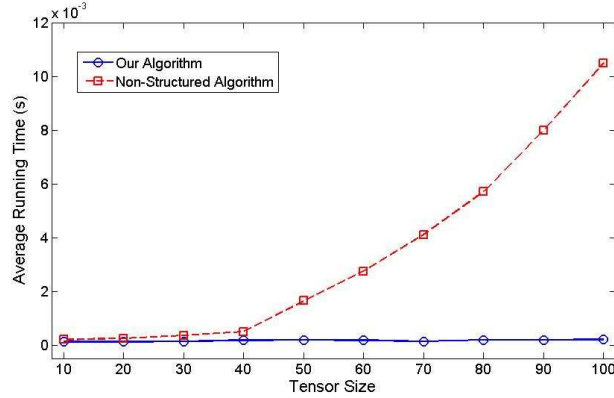- A two-dimensional signal is modelled as

$$\begin{aligned} x_{n_1 n_2} = {} & \exp\big((-0.01 + 2\pi \imath 0.20)n_1\big) \cdot \exp\big((-0.02 + 2\pi \imath 0.18)n_2\big) \\ & + \exp\big((-0.02 + 2\pi \imath 0.22)n_1\big) \cdot \exp\big((-0.01 - 2\pi \imath 0.20)n_2\big) + e_{n_1 n_2}, \end{aligned}$$

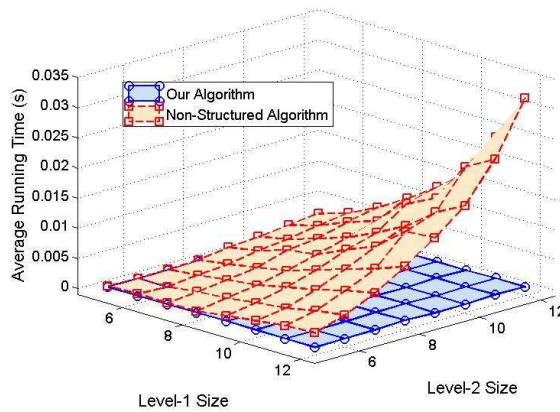 where $e_{n_1 n_2}$ is a two-dimensional complex white Gaussian noise.

The 3rd-order approach is accepted for both cases. We test the running times of the rank-$(2, 2, 2)$ approximation since these signals both have 2-peaks. Moreover, we will illustrate the HOSVDs of these Hankel and BHHB tensors, which shows that Papy et al.'s algorithm and our extended multi-dimensional version can also work when the number of peaks is unknown.

Figure 3 shows the comparison of these two algorithms' speeds. It provides a similar trend with the one in Figure 2, since the tensor-vector product plays a dominant role in the HOOI procedure. Therefore, when the speed of tensor-vector products is largely increased by exploiting the Hankel or block Hankel structure, we can handel much larger problems than before.

Then we fix the size of the Hankel and BHHB tensors. The Hankel tensor for 1D exponential data fitting is of size $15 \times 15 \times 15$, and the BHHB tensor for 2D exponential data fitting is of

Copyright © 2014 John Wiley & Sons, Ltd.
Prepared using nlaauth.cls

*Numer. Linear Algebra Appl.* 2014; **00**:1–18

(a) Hankel tensors

(b) BHHB tensors

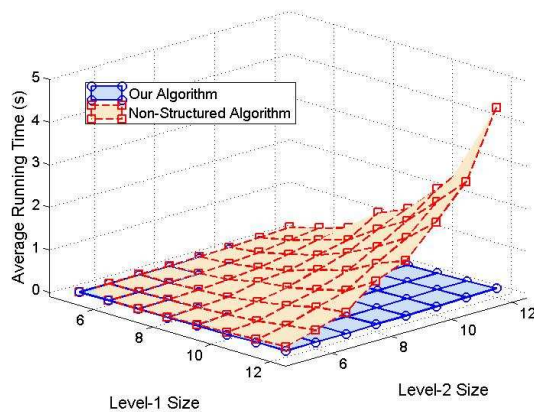Figure 2. The average running time of tensor-vector products.

level-1 size $5 \times 5 \times 5$ and level-2 size $6 \times 6 \times 6$. Assume that we do not know the number of peaks. Then we compute the HOSVD of the best rank-$(10, 10, 10)$ approximation

$$\mathcal{H} \approx \mathcal{S} \times_1 U_1^\top \times_2 U_2^\top \times_3 U_3^\top,$$

where the core tensor $\mathcal{S}$ is of size $10 \times 10 \times 10$. Figure 4 displays the Frobenius norm of $\mathcal{S}(k, :, :)$ for $k = 1, 2, \ldots, 10$. We can see that the first two of them are apparently larger than the others. (The others should be zero when the signal is noiseless, but here we add a noise at 10^{-4}-level.) Thus we can directly conclude that the number of peaks is two. Furthermore, our fast algorithm enables us to accept a much wild guess rather than to be anxious for the running time.
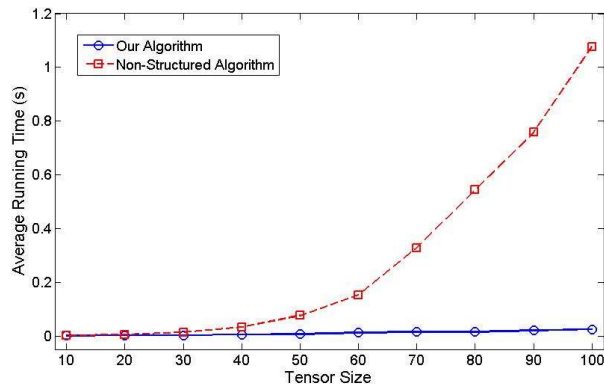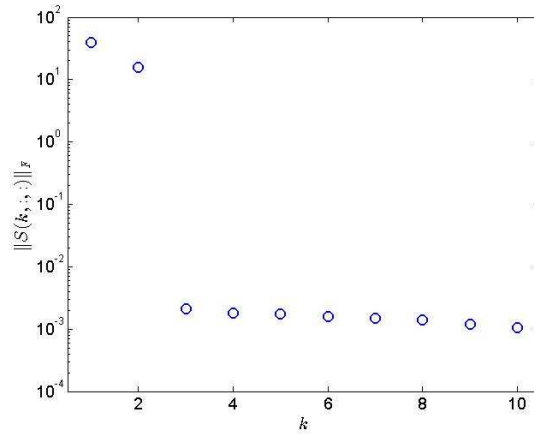
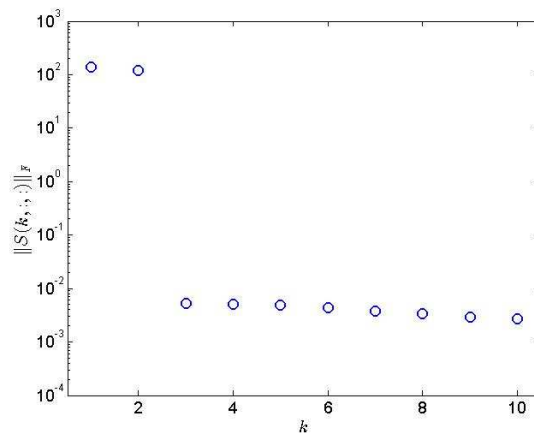(a) Hankel tensors

(b) BHHB tensors

Figure 3. The average running time of HOOI.

6. Conclusions

We propose a fast algorithm for Hankel tensor-vector products, which reduces the computational complexity from $\mathcal{O}(n^m)$ to $\mathcal{O}(m^2 n \log mn)$ comparing with the algorithm without exploit the Hankel structure. This fast algorithm is derived by embedding the Hankel tensor into a larger anti-circulant tensor, which can be diagonalized by the Fourier matrices. The fast algorithm for higher-level block Hankel tensors is also described. Furthermore, the fast Hankel and BHHB tensor-vector products can largely accelerate the Papy et al.'s algorithm for one-dimensional exponential data fitting and our generalized algorithm for multi-dimensional case, respectively. It should be pointed out that our algorithm can also analogously be applied to higher-dimensional case although we only introduce the one- and two-dimensional cases for examples. The numerical experiments show the efficiency and effectiveness of our algorithms. Finally, this fast scheme

(a) Hankel tensors

(b) BHHB tensors

Figure 4. The Frobenius norms of slices of the core tensor.

should be introduced into every algorithm that involves Hankel or higher-level block Hankel tensor-vector products to improve its performance.

REFERENCES

1. Olshevsky V. *Structured Matrices in Mathematics, Computer Science, and Engineering. II.* Contemporary Mathematics, vol. 281. American Mathematical Society, Providence, RI, 2001.
2. Luque JG, Thibon JY. Hankel hyperdeterminants and Selberg integrals. *Journal of Physics A: Mathematical and General* 2003; **36**(19):5267–5292.
3. Badeau R, Boyer R. Fast multilinear singular value decomposition for structured tensors. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**(3):1008–1021.
4. Boyer R, De Lathauwer L, Abed-Meraim K. Higher-order tensor-based method for delayed exponential fitting. *IEEE Transactions on Signal Processing* 2007; **55**(6):2795–2809.
5. Papy JM, De Lathauwer L, Van Huffel S. Exponential data fitting using multilinear algebra: The single-channel and multi-channel case. *Numerical Linear Algebra with Applications* 2005; **12**(8):809–826.
6. Papy JM, De Lathauwer L, Van Huffel S. Exponential data fitting using multilinear algebra: The decimative case. *Journal of Chemometrics* 2009; **23**(7-8):341–351.
7. De Lathauwer L. Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, \ldots, 1)$ terms. *SIAM Journal on Matrix Analysis and Applications* 2011; **32**(4):1451–1474.
8. Qi L. Hankel tensors: Associated Hankel matrices and Vandermonde decomposition. *Communications in Mathematical Sciences* 2015; **13**:113–125.
9. Song Y, Qi L. Infinite and finite dimensional Hilbert tensors. *Linear Algebra and its Applications* 2014; **451**:1–14.
10. Chan R, Jin XQ. *An Introduction to Iterative Toeplitz Solvers.* SIAM, 2007.
11. Luk FT, Qiao S. A fast singular value algorithm for Hankel matrices. *Contemporary Mathematics*, vol. 323, Edited by Vadim Olshevsky, American Mathematical Society, Providence, RI; SIAM, Philadelphia, 2003; 169–178.
12. Ng MK. *Iterative Methods for Toeplitz Systems.* Oxford University Press, 2004.
13. Xu W, Qiao S. A fast symmetric SVD algorithm for square Hankel matrices. *Linear Algebra and its Applications* 2008; **428**(2):550–563.
14. Pereyra V, Scherer G. *Exponential Data Fitting and its Applications.* Bentham Science Publishers, 2010.
15. Potts D, Tasche M. Parameter estimation for multivariate exponential sums. *Electronic Transactions on Numerical Analysis* 2013; **40**:204–224.
16. Potts D, Tasche M. Parameter estimation for nonincreasing exponential sums by prony-like methods. *Linear Algebra and its Applications* 2013; **439**(4):1024–1039.
17. Roy R, Kailath T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech and Signal Processing* 1989; **37**(7):984–995.
18. Kailath T, Roy III RH. ESPRIT–estimation of signal parameters via rotational invariance techniques. *Optical Engineering* 1990; **29**(4):296–313.
19. Golub GH, Van Loan CF. *Matrix Computations.* 4th edn., The Johns Hopkins University Press, 2013.
20. Kolda TG, Bader BW. Tensor decompositions and applications. *SIAM Review* 2009; **51**(3):455–500.
21. De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications* 2000; **21**(4):1253–1278.
22. De Lathauwer L, De Moor B, Vandewalle J. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications* 2000; **21**(4):1324–1342.
23. De Lathauwer L, Hoegaerts L. Rayleigh quotient iteration for the computation of the best rank-$(R_1, R_2, \ldots, R_N)$ approximation in multilinear algebra. *Technical Report*, SCD-SISTA 04-003.
24. Eldén L, Savas B. A Newton-Grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor. *SIAM Journal on Matrix Analysis and Applications* 2009; **31**(2):248–271.
25. Rouquette S, Najim M. Estimation of frequencies and damping factors by two-dimensional ESPRIT type methods. *IEEE Transactions on Signal Processing* 2001; **49**(1):237–245.
26. Wang Y, Chen JW, Liu Z. Comments on "Estimation of frequencies and damping factors by two-dimensional ESPRIT type methods". *IEEE Transactions on Signal Processing* 2005; **53**(8):3348–3349.
27. Davis PJ. *Circulant Matrices.* John Wiley & Sons: New York, 1979.
28. Qi L. Eigenvalues of a real supersymmetric tensor. *Journal of Symbolic Computation* 2005; **40**(6):1302–1324.
29. Qi L. Eigenvalues and invariants of tensors. *Journal of Mathematical Analysis and Applications* 2007; **325**(2):1363–1377.
30. Lim LH. Singular values and eigenvalues of tensors: a variational approach. *2005 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, IEEE, 2005; 129–132.
31. Jin XQ. *Developments and Applications of Block Toeplitz Iterative Solvers.* Science Press, Beijing and Kluwer Academic Publishers, Dordrecht, 2002.
32. Qiao S, Liu G, Xu W. Block Lanczos tridiagonalization of complex symmetric matrices. *Advanced Signal Processing Algorithms, Architectures, and Implementations XV, Proceedings of the SPIE*, vol. 5910, SPIE, 2005; 285–295.
33. Browne K, Qiao S, Wei Y. A Lanczos bidiagonalization algorithm for Hankel matrices. *Linear Algebra and its Applications* 2009; **430**(5):1531–1543.
34. Chen Z, Qi L. Circulant tensors with applications to spectral hypergraph theory and stochastic process. *arXiv preprint arXiv:1312.2752* 2014; .