Pacific Journal of Optimization
Vol. 17, No. 3, pp. 337–356, 2021

NUMERICAL METHODS FOR COMPUTING TWO KINDS OF THE BEST HANKEL TENSOR APPROXIMATIONS[*]

Chunmei Li, Zhuling Jiang and Xuefeng Duan[†]

Abstract: Hankel tensors and their approximation problems are of particular interest in the multidimensional seismic trace interpolator problem. In this paper, we investigate the numerical methods for two kinds of the best Hankel tensor approximation problems. Based on the Vandermonde decomposition of Hankel tensors, the Hankel tensor approximation problem with missing data is transformed into an unconstrained optimization problem, and then the BFGS method is used to solve it. For the Hankel tensor approximation problems with the interval constraint and box constraint, Dykstra's algorithm and its acceleration versions are designed to solve them. Numerical examples illustrate that these methods are feasible and effective.

Key words: *Hankel tensor approximation, BFGS method, Dykstra's algorithm, acceleration technique*

Mathematics Subject Classification: *16A69, 65F99*

1 Introduction

Throughout this paper, to distinguish scalars, vectors, matrices and higher-order tensors, scalars will be denoted by lower case Greek letters, e.g. α, β, vectors will be denoted by lowercase letters, e.g. v, w, matrices will be defined by uppercase letters, e.g. A, B and higher-order tensors will be denoted by calligraphic letters, e.g. $\mathcal{A}, \mathcal{B}$. Denote that $[n] := \{1, 2, \ldots, n\}$. A tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ is said to be a symmetric tensor if every entry $a_{i_1 i_2 \ldots i_m}$ is invariant under any index permutation. Let the vector $v = (v_0, v_1, \ldots, v_{(n-1)m})^T$. Define the symmetric tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ associated with v by

$$a_{i_1 i_2 \ldots i_m} = v_{i_1+i_2+\cdots+i_m-m}, \tag{1.1}$$

for $i_1, i_2, \ldots, i_m \in [n]$. Then $\mathcal{A}$ is called a Hankel tensor and v is called the generating vector of $\mathcal{A}$. We see that a sufficient and necessary condition for a symmetric tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m})$ to be a Hankel tensor is that whenever $i_1 + i_2 + \cdots + i_m = j_1 + j_2 + \cdots + j_m$ we have $a_{i_1 i_2 \ldots i_m} = a_{j_1 j_2 \ldots j_m}$. An interesting Theorem from [36, 25] is as follows.

An mth order $n-$dimensional Hankel tensor $\mathcal{T}$ can always be decomposed as

$$\mathcal{T} = \sum_{j=1}^{r} \lambda_j \mu_j \circ \mu_j \circ \cdots \circ \mu_j, \tag{1.2}$$

[*] The work was supported by the National Natural Science Foundation of China (No.11561015; 11761024), the Natural Science Foundation of Guangxi Province (No.2016GXNSFFA380009; 2017GXNSFBA198082; 2016GXNSFAA380074).

[†] Corresponding author.

where each λ_j is a scalar and $\mu_j = (1, a_j, a_j^2, \ldots, a_j^{n-1})^T$ for some number a_j. In this case, we have $r \leq (n-1)m + 1$.

The decomposition (1.2) is called the Vandermonde decomposition of the Hankel tensor $\mathcal{T}$. We call the minimum value r the Vandermonde rank of $\mathcal{T}$, and denote it by $Vrank(\mathcal{T})$ (see [36] for more details). We use $T_{m,n}$, $S_{m,n}$, $H_{m,n}$ to denote the set of mth order n-dimensional tensors, symmetric tensor and Hankel tensor, respectively. The symbol $\mathcal{A} \odot \mathcal{B}$ means that the entrywise product of the mth order $n-$dimensional tensors $\mathcal{A}$ and $\mathcal{B}$, that is, $\mathcal{A} \odot \mathcal{B} = (a_{i_1 i_2 \ldots i_m} b_{i_1 i_2 \ldots i_m})$. The symbol $vec(\cdot)$ stands for the vectorization operator of a matrix or tensor, and $vec^{-1}(\cdot)$ stands for its inverse operator. Set $\mathcal{U} = (u_{i_1 i_2, \ldots, i_m})$ and $\mathcal{V} = (v_{i_1 i_2, \ldots, i_m})$, $\mathcal{V} \geq \mathcal{U}$ means that $v_{i_1 i_2, \ldots, i_m} \geq u_{i_1 i_2, \ldots, i_m}$ for all possible $(i_1 i_2, \ldots, i_m)-$entries. We define the Frobenius norm of a tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_m}) \in T_{m,n}$ by

$$\|\mathcal{A}\| = \left(\sum_{i_1, i_2, \ldots, i_m = 1}^{n} a^2_{i_1 i_2 \ldots i_m} \right)^{\frac{1}{2}},$$

which is induced by the inner product

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \ldots, i_m = 1}^{n} a_{i_1 i_2 \ldots i_m} b_{i_1 i_2 \ldots i_m},$$

where $\mathcal{A}, \mathcal{B} \in T_{m,n}$.

In this paper, we consider the following Hankel tensor approximation problems.

Problem I. *Given an mth order n-dimensional tensor $\mathcal{A}$ with missing data, find an mth order n-dimensional Hankel tensor $\widehat{\mathcal{X}}$ with $Vrank(\widehat{\mathcal{X}}) \leq r$ such that*

$$\|W \odot (\mathcal{A} - \widehat{\mathcal{X}})\|^2 = \min_{\mathcal{X} \in H_{m,n}, \quad Vrank(\mathcal{X}) \leq r} \|W \odot (\mathcal{A} - \mathcal{X})\|^2, \tag{1.3}$$

where $W = (w_{i_1 i_2 \ldots i_m})$ is a $0-1$ tensor, that is, if the element $a_{i_1 i_2 \ldots i_m}$ of $\mathcal{A}$ is missing then $w_{i_1 i_2 \ldots i_m} = 0$, otherwise $w_{i_1 i_2 \ldots i_m} = 1$.

Problem II. *Given an mth order n-dimensional tensor $\mathcal{A}$, find an mth order n-dimensional Hankel tensor $\widehat{\mathcal{X}}$ with structure constraint $\digamma_1$ (or $\digamma_2$) such that*

$$\|\mathcal{A} - \widehat{\mathcal{X}}\|^2 = \min_{\mathcal{X} \in H_{m,n}, \mathcal{X} \in \digamma_i} \|\mathcal{A} - \mathcal{X}\|^2. \tag{1.4}$$

Here the structure constraints are

$$\digamma_1 = \{\mathcal{X} \in T_{m,n} | \ \ \mathcal{U} \leq \mathcal{X} \leq \mathcal{V}, \ \ \mathcal{U}, \mathcal{V} \in T_{m,n}\},$$

$$\digamma_2 = \{\mathcal{X} \in T_{m,n} | \ \ \|\mathcal{X}\| \leq \delta, \ \ \delta > 0\}.$$

Problem I arises in the multidimensional seismic trace interpolator problem (see [33,21] for more details), which can be stated as follows. Given a raw frequency slice S having two spatial dimensions with lengths s_1 and s_2, we can form a fourth-order Hankel tensor $\mathcal{A}$ by generating two tensor orders for every spatial dimension

$$\mathcal{A}(i, j, m, n) = S(i + j - 1, m + n - 1).$$

In four spatial dimensions we build an eighth-order tensor

$$\mathcal{A}(i, j, m, n, p, q, r, s) = S(i + j - 1, m + n - 1, p + q - 1, r + s - 1).$$

Of course, some elements of the tensor $\mathcal{A}$ will be unknown (and thus in need of interpolating) due to the missing traces. After that we must find a low rank Hankel tensor $\mathcal{X}$, which fits as closely as possible the known elements of $\mathcal{A}$. This process can be summarized as Problem I.

Problem II is of particular interests in the signal processing, such as in the multidimensional harmonic retrieval problems as follows (see [7,11,14,20,24,30,35]). Let us consider N samples of a time series $x_n, n \in \{1, 2, \ldots, N\}$ modeled as a finite sum of K exponentially damped complex sinusoids

$$x_n = \sum_{p=1}^{P} a_p e^{j\phi(p)} e^{(-d_p + jw_p)t_n},$$

where a_p are the amplitudes, ϕ_p the phases, d_p the damping factors and w_p the pulsations. $t_n = n\Delta t$ is the time lapse between the time origin and the sample x_n and Δt is the sampling time interval. These data may be arranged in a high-order Hankel tensor. In order to estimate a tensor data from noisy observations and increase the relative signal strength, we must use a Hankel tensor to approximate a data tensor, in particular, some structured conditions, such as interval constraint and box constraint, are added in the best Hankel tensor approximation problems, which lead to Problem II.

Recently, Hankel tensors have been investigated extensively and the research results mainly concentrated on the spectral characterizations ([25]), tensor decompositions ([7,8,10]) and their positivities (including positive definiteness, copositivity, and complete positivity [26-29,36]) and their applications ([11,13]).

When the order $m = 2$, the Hankel tensor approximation problem reduces to the Hankel matrix approximation problem. Macinnes [23] developed a method for finding the best approximation of a matrix by a full rank Hankel matrix. The Hankel matrix approximation problem is transformed into a problem involving best approximation of a given vector by a second vector whose elements are constrained so that its inverse image can be algebraically illustrated by Hankel matrix. Based on the projection algorithm and Newton method, Al-Homidan [3] proposed a hybrid method for approximating the positive semidefinite Hankel matrix. Arnab, Pokala and Kumaresan [5] presented the algorithms for the least-squares approximation of Toeplitz and Hankel matrices from noise corrupted or ill-composed matrices. The similar structured matrix approximation problems, such as Toeplitz matrix approximation [1, 5, 17] and positive semidefinite matrix approximation [15,31], have also been studied, and some excellent numerical methods were developed to solve them. However, the research results of the high-order Hankel tensor approximation problems I and II are very few as far as we know. The biggest obstacle the characterization of the feasible sets of Problems I and II because they have highly structured constraints.

In this paper, we overcome these difficulties by the Vandermonde decomposition and projections. For the problem I, we first transform Problem I into a smooth unconstrained optimization problem by using the Vandermonde decomposition of Hankel tensor, then use the BFGS method with Armijo line search to solve it. For the problem II, we first compute the projections of a tensor onto the sets $H_{m,n}$, $\digamma_1$ and $\digamma_2$, and then design the Dykstra's algorithm and its acceleration versions to solve it. Numerical experiments illustrate that the new methods are feasible and effective.

This paper is organized as follows. In Section 2, we first use the Vandermonde decomposition of Hankel tensor to characterize the feasible set of Problem I and then use the BFGS method to solve it. In Section 3, we first compute the projections of a tensor onto the sets $H_{m,n}$, $\digamma_1$ and $\digamma_2$, and then design the Dykstra's algorithm and its acceleration versions to solve Problem II. Finally, some numerical examples are reported to verify the feasibility and effectiveness of these methods.

$\boxed{2}$ BFGS Method for Solving Problem I

In this section, we first transform Problem I into an unconstrained optimization problem, and then use the BFGS method with the Armijo line search to solve it.

Set

$$\Omega = \{\mathcal{X} \in T_{m,n} \mid \quad \mathcal{X} \in H_{m,n}, Vrank(\mathcal{X}) \leq r\},$$

then from (1.2) it follows that the feasible set Ω of Problem I can be characterized by

$$\mathcal{X} = \sum_{j=1}^{r} \lambda_j \mu_j \circ \mu_j \circ \ldots \mu_j,$$

where $\mu_j = (1, x_j, \ldots, x_j^{n-1})^T$. It is easy to verify that

$$\mathcal{X} = (x_{i_1 i_2 \ldots i_m}), \quad x_{i_1 i_2 \ldots i_m} = \sum_{j=1}^{r} \lambda_j x_j^{i_1+i_2+\cdots+i_m-m}.$$

Therefore Problem I can also be reformulated as an unconstrained optimization problem

$$\min_{\eta=(x_1,x_2,\ldots,x_r,\lambda_1,\lambda_2,\ldots,\lambda_r)\in R^{2r}} f(\eta), \tag{2.1}$$

where the objective function

$$\begin{aligned} f(\eta) &= \|W \odot (\mathcal{A} - \mathcal{X})\|^2 \\ &= \|W \odot (\mathcal{A} - \sum_{j=1}^{r} \lambda_j \mu_j \circ \mu_j \circ \ldots \mu_j)\|^2 \\ &= \sum_{i_1,i_2,\ldots,i_m=1}^{n} [w_{i_1 i_2 \ldots i_m}(a_{i_1 i_2 \ldots i_m} - \sum_{j=1}^{r} \lambda_j x_j^{i_1+i_2+\cdots+i_m-m})]^2. \end{aligned}$$

Now we begin to use the BFGS method with Armijo line search to solve (2.1). The important step to use this method is how to compute the gradient of the objective function $f(\eta)$. Set $p = i_1 + i_2 + \cdots + i_m - m$, then

$$\frac{\partial f}{\partial x_l} = 2 \sum_{i_1,i_2,\ldots,i_m=1}^{n} w_{i_1 i_2 \ldots i_m} \left(\sum_{j=1}^{s} \lambda_j x_j^p - a_{i_1 i_2 \ldots i_m} \right) (p \lambda_l x_l^{p-1}), \quad l = 1, 2, \ldots, r, \tag{2.2}$$

$$\frac{\partial f}{\partial \lambda_l} = 2 \sum_{i_1,i_2,\ldots,i_m=1}^{n} w_{i_1 i_2 \ldots i_m} \left(\sum_{j=1}^{s} \lambda_j x_j^p - a_{i_1 i_2 \ldots i_m} \right) (x_l^p), \quad l = 1, 2, \ldots, r. \tag{2.3}$$

Therefore the gradient

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_r}, \frac{\partial f}{\partial \lambda_1}, \frac{\partial f}{\partial \lambda_2}, \ldots, \frac{\partial f}{\partial \lambda_r} \right)$$

is given by (2.2) and (2.3).

The BFGS method with Armijo line search to solve (2.1) can be stated as follows.

Algorithm 2.1 (This algorithm attempts to solve (2.1))**.**
1. Given an initial value $\eta_0 \in R^{2r}$, the initial positive definite matrix B_0 and the tolerant error $\varepsilon > 0$. Set $\rho \in (0,1), \sigma \in (0, 0.5), k = 0$.
2. Compute $\nabla g_k = \nabla f(\eta_k)$. If $\| \nabla g_k \| \leq \varepsilon$, then stop and declare that η_k is a stationary point.
3. Determine d_k by solving the systems of linear equations $B_k d = -\nabla g_k$.
4. Determine the step length α_k by Armijo line search, that is, find the smallest nonnegative integer m_k such that

$$f(\eta_{k+1}) \leq f(\eta_k) + \sigma \rho^{m_k} \nabla g_k^T d_k.$$

5. Set $\alpha_k = \rho^{m_k}, \eta_{k+1} = \eta_k + \alpha_k d_k$.
6. Compute $s_k = \eta_{k+1} - \eta_k$, $y_k = \nabla g_{k+1} - \nabla g_k$.
7. Determine B_{k+1} by

$$B_{k+1} = \begin{cases} B_k, \quad if \;\; y_k^T s_k \leq 0; \\ B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \quad if \;\; y_k^T s_k > 0. \end{cases}$$

8. Let $k = k + 1$. Go to step 2.

The global convergence theorem for Algorithm 2.1 can be seen in Theorem 10.3.7 of Chen ([11]).

3 Dykstra's Algorithm and its Acceleration Versions for Solving Problem II

In this section, we use Dykstra's algorithm and its acceleration versions to compute the Hankel tensor approximation with structure constraints $\mathcal{F}_1$ and $\mathcal{F}_2$. Since the objective function of Problem II is a convex function and the feasible set of Problem II is a convex set, so the problem II always has a solution. We first introduce Dykstra's algorithm and its convergence theorem. We begin with a definition.

Definition 3.1 ([6])**.** Let M be a closed convex subset in a real Hilbert space H and u be a point in H, then the unique solution of the following minimization problem

$$\min_{x \in M} \|x - u\|$$

is called the projection of u onto M and denoted by $P_M(u)$.

In order to find the projection of a given point onto the intersection of a finite number of closed convex sets $C_1, C_2, \ldots, C_n$, Dykstra's algorithm was proposed in [6] which can be stated as follows. This algorithm can be also seen in [12,22].

Dykstra's Algorithm

```
1. Given the initial value $x_0$;
2. Set $x_n^{(0)} = x_0, \;\; I_i^{(0)} = 0, \;\; i = 1, 2, \ldots, n$.
3. For $k = 1, 2, 3, \ldots$
      $x_0^{(k)} = x_n^{(k-1)}$
          For $i = 1, 2, \ldots, n$
      $x_i^{(k)} = P_{C_i}(x_{i-1}^{(k)} - I_i^{(k-1)})$,
      $I_i^{(k)} = x_i^{(k)} - (x_{i-1}^{(k)} - I_i^{(k-1)})$.
          End
   End
```

The utility of Dykstra's algorithm is based on the following lemma.

Lemma 3.2 ([9], Theorem 2)**.** *Let $C_1, C_2, \dots, C_n$ be some closed convex subsets of a real Hilbert space H such that $C_1 \bigcap C_2 \bigcap \cdots \bigcap C_n \neq \varnothing$. For any $i = 1, 2, \dots, n$ and any $x_0 \in H$, then the sequences $\{x_i^{(k)}\}$ generated by Dykstra's algorithm converge to $P_{C_1 \bigcap C_1 \bigcap \cdots \bigcap C_n}(x_0)$, that is,*

$$x_i^{(k)} \to P_{C_1 \bigcap C_1 \bigcap \cdots \bigcap C_n}(x_0), \quad i = 1, 2, \dots, n, \quad k \to +\infty.$$

However, the rate of convergence of Dykstra's algorithm is at best linear and it can require a large number of iterations to converge within a given tolerance. In order to accelerate the convergence, the acceleration versions of Dykstra's algorithm were proposed (see [16,19] for more details). We first introduce the Anderson acceleration of the fixed point iteration. Anderson acceleration was proposed by Anderson [4] and developed by Toth and Kelley [32] and Walker and Ni [34]. Recently Higham [16] used this method to compute the nearest correlation matrix problem. Suppose we want to solve the equation $x = f(x)$ for some $g : R^n \to R^n$. Basic fixed point iteration for this problem is as follows

$$Given \;\; x_0, \quad x_{k+1} = f(x_k), \quad k = 0, 1, 2, \dots.$$

For this iteration, the Anderson acceleration is as follows. This algorithm starts with $x_0 \in R^n$ and an integer $m \geq 1$. The following notation is used: $mk = \min(m, k), \triangle x_i = x_{i+1} - x_i, X_k = [\triangle x_{k-mk} \dots \triangle x_{k-1}], f_i = f(x_i), \triangle f_i = f_{i+1} - f_i$ and $F_k = [\triangle f_{k-mk} \dots \triangle f_{k-1}]$.

Anderson acceleration algorithm

```
$x_1 = x_0 + f(x_0)$
For $k = 1, 2, 3, \dots$
    $mk = \min(m, k)$;
    Determine $\gamma_k = (\gamma_{k-mk}^{(k)}, \dots, \gamma_{k-1}^{(k)})^T$ that solves $\min_{\gamma \in R^{mk}} \|f_k - F_k\gamma\|_2^2$ by using QR factorization;
    $\bar{x}_k = x_k - \sum_{i=k-mk}^{k-1} \gamma_i^k \triangle x_i = x_k - X_k \gamma^{(k)}$;
    $\bar{f}_k = f_k - \sum_{i=k-mk}^{k-1} \gamma_i^k \triangle f_i = f_k - F_k \gamma^{(k)}$;
    $x_{k+1} = \bar{x} + \bar{f}$.
End
```

Now we begin to solve Problem II with structure constraints $\digamma_1$ and $\digamma_2$. It is easy to verify that Problem II is equivalent to finding the projections $P_{\digamma_i \bigcap H_{m,n}}(\mathcal{A})$ of $\mathcal{A}$ onto the intersection set $\digamma_i \bigcap H_{m,n}$. Now we will use Dykstra's algorithm and its acceleration versions to solve Problem II. The key problems to realize these algorithms are how to compute the projections $P_{H_{m,n}}(\mathcal{Z})$ and $P_{\digamma_i}(\mathcal{Z}), i = 1, 2$ of a given tensor $\mathcal{Z}$ onto the sets $H_{m,n}$ and $\digamma_i$, respectively. Such problems are perfectly solved as follows.

We can see that $H_{m,n}$ is a subspace of $S_{m,n}$, and its dimension is $(n-1)m + 1$. Let $\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_{(n-1)m}$ be the orthogonal basis of $H_{m,n}$, that is, $\langle \mathcal{G}_i, \mathcal{G}_j \rangle = 0, 0 \leq i \neq j \leq (n-1)m$. Then we have the following results.

Theorem 3.3. *For a given mth order n-dimensional tensor $\mathcal{Z}$, then the projection $P_{H_{m,n}}(\mathcal{Z})$ of $\mathcal{Z}$ onto $H_{m,n}$ is*

$$P_{H_{m,n}}(\mathcal{Z}) = \sum_{k=0}^{(n-1)m} x_k \mathcal{G}_k,$$

where

$$x_k = \frac{\langle \mathcal{Z}, \mathcal{G}_k \rangle}{\langle \mathcal{G}_k, \mathcal{G}_k \rangle}, \quad k = 0, 1, \dots, (n-1)m.$$

Proof. Since $\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_{(n-1)m}$ be the orthogonal basis of $H_{m,n}$, then for arbitrary $\mathcal{X} \in H_{m,n} \subset S_{m,n}$, there exists real numbers $x_0, x_1, \ldots, x_{(n-1)m}$, such that $\mathcal{X} = \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k$. Therefore,

$$\begin{array}{rcl} f(\mathcal{X}) & = & \|\mathcal{Z} - \mathcal{X}\|^2 \\ & = & \|\mathcal{Z} - \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k\|^2 \\ & = & \langle \mathcal{Z} - \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k, \mathcal{Z} - \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k \rangle \\ & = & \langle \mathcal{Z}, \mathcal{Z} \rangle - 2\langle \mathcal{Z}, \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k \rangle + \langle \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k, \sum\limits_{k=0}^{(n-1)m} x_k \mathcal{G}_k \rangle \\ & = & \|\mathcal{Z}\|^2 - 2 \sum\limits_{k=0}^{(n-1)m} x_k \langle \mathcal{Z}, \mathcal{G}_k \rangle + \sum\limits_{k=0}^{(n-1)m} x_k^2 \langle \mathcal{G}_k, \mathcal{G}_k \rangle. \end{array}$$

Set

$$\frac{\partial f}{\partial x_i} = 2x_k \langle \mathcal{G}_k, \mathcal{G}_k \rangle - 2\langle \mathcal{Z}, \mathcal{G}_k \rangle = 0,$$

then we derive the stationary points of f are as follows

$$x_k = \frac{\langle \mathcal{Z}, \mathcal{G}_k \rangle}{\langle \mathcal{G}_k, \mathcal{G}_k \rangle}, \quad k = 0, 1, \ldots, (n-1)m.$$

Noting that

$$\frac{\partial^2 f}{\partial x_i^2} = 2\langle \mathcal{G}_k, \mathcal{G}_k \rangle = 2\|\mathcal{G}_k\|^2 > 0,$$

then by Definition 3.1 we obtain that the projection $P_{H_{m,n}}(\mathcal{Z})$ is

$$P_{H_{m,n}}(\mathcal{Z}) = \sum_{k=0}^{(n-1)m} x_k \mathcal{G}_k,$$

where

$$x_k = \frac{\langle \mathcal{Z}, \mathcal{G}_k \rangle}{\langle \mathcal{G}_k, \mathcal{G}_k \rangle}, \quad k = 0, 1, \ldots, (n-1)m.$$

□

Proof. Set the mth order n-dimensional tensors $\mathcal{V} = (v_{i_1 i_2, \ldots, i_m})$ and $\mathcal{U} = (u_{i_1 i_2, \ldots, i_m})$. Suppose that the set $\mathcal{F}_1$ be nonempty. For a given mth order n-dimensional tensor $\mathcal{Z} = (z_{i_1 i_2, \ldots, i_m})$, then the $(i_1 i_2, \ldots, i_m)$-entry of the projection $P_{\mathcal{F}_1}(\mathcal{Z})$ is as follows

$$(P_{\mathcal{F}_1}(\mathcal{Z}))_{i_1 i_2, \ldots, i_m} = \begin{cases} z_{i_1 i_2, \ldots, i_m}, & if \;\; u_{i_1 i_2, \ldots, i_m} \leq z_{i_1 i_2, \ldots, i_m} \leq v_{i_1 i_2, \ldots, i_m}, \\ v_{i_1 i_2, \ldots, i_m}, & if \;\; z_{i_1 i_2, \ldots, i_m} > v_{i_1 i_2, \ldots, i_m}, \\ u_{i_1 i_2, \ldots, i_m}, & if \;\; z_{i_1 i_2, \ldots, i_m} < u_{i_1 i_2, \ldots, i_m}. \end{cases}$$

□

Proof. By Definition 3.1 we obtain that the projection $P_{\mathcal{F}_1}(\mathcal{Z})$ is the unique solution of the following minimization problem

$$\min_{\mathcal{X} \in \mathcal{F}_1} \|\mathcal{Z} - \mathcal{X}\|^2 = \sum_{i_1, i_2, \ldots, i_m = 1}^{n} (z_{i_1 i_2, \ldots, i_m} - x_{i_1 i_2, \ldots, i_m})^2.$$

This problem attains its minimum if and only if

$$x_{i_1 i_2,\ldots,i_m} = \left\{ \begin{array}{ll} z_{i_1 i_2,\ldots,i_m}, & if \quad u_{i_1 i_2,\ldots,i_m} \leq z_{i_1 i_2,\ldots,i_m} \leq v_{i_1 i_2,\ldots,i_m}, \\ v_{i_1 i_2,\ldots,i_m}, & if \quad z_{i_1 i_2,\ldots,i_m} > v_{i_1 i_2,\ldots,i_m}, \\ u_{i_1 i_2,\ldots,i_m}, & if \quad z_{i_1 i_2,\ldots,i_m} < u_{i_1 i_2,\ldots,i_m}, \end{array} \right.$$

which is the projection $P_{\mathcal{F}_1}(\mathcal{Z})$ of $\mathcal{Z}$ onto the set $\mathcal{F}_1$. □

Theorem 3.4. *For a given mth order n-dimensional tensor* $\mathcal{Z} = (z_{i_1 i_2,\ldots,i_m})$, *then the projection* $P_{\mathcal{F}_2}(\mathcal{Z})$ *of* $\mathcal{Z}$ *onto* $\mathcal{F}_2$ *is as follows*

$$P_{\mathcal{F}_2}(\mathcal{Z}) = \left\{ \begin{array}{ll} \mathcal{Z}, & if \quad \|\mathcal{Z}\| \leq \delta, \\ \frac{\delta}{\|\mathcal{Z}\|}\mathcal{Z}, & if \quad \|\mathcal{Z}\| > \delta. \end{array} \right.$$

Proof. Set $f(\mathcal{X}) = \|\mathcal{Z} - \mathcal{X}\|$. By Definition 3.1 we know that the projection $P_{\mathcal{F}_2}(Z)$ is the unique solution of the minimization problem $\min\limits_{\mathcal{X}\in\mathcal{F}_2} f(\mathcal{X})$. It is easy to see that if $\|\mathcal{Z}\| < \delta$, then the unique solution of $\min\limits_{\mathcal{X}\in\mathcal{F}_2} f(\mathcal{X})$ is Z. Now we consider the case $\|\mathcal{Z}\| \geq \delta$. For arbitrary $\mathcal{X} \in \mathcal{F}_2$ with $\|\mathcal{Z}\| \geq \delta$, then

$$\begin{array}{rcl} \|\mathcal{Z}\| & = & \|\mathcal{Z} - \mathcal{X} + \mathcal{X}\| \\ & \leq & \|\mathcal{Z} - \mathcal{X}\| + \|\mathcal{X}\|, \end{array}$$

which implies that

$$\begin{array}{rcl} \|\mathcal{Z} - \mathcal{X}\| & \geq & \|\mathcal{Z}\| - \|\mathcal{X}\| \\ & \geq & \|\mathcal{Z}\| - \delta. \end{array}$$

This inequality shows that for arbitrary $\mathcal{X} \in \mathcal{F}_2$ with $\|\mathcal{Z}\| \geq \delta$, we have

$$\begin{array}{rcl} f(\mathcal{X}) & = & \|\mathcal{Z} - \mathcal{X}\| \\ & \geq & \|\mathcal{Z}\| - \delta \\ & = & \frac{1}{\|\mathcal{Z}\|}(\|\mathcal{Z}\| - \delta)\|\mathcal{Z}\| \\ & = & \frac{1}{\|\mathcal{Z}\|}\|\|\mathcal{Z}\|\mathcal{Z} - \delta\mathcal{Z}\| \\ & = & \|\mathcal{Z} - \frac{\delta}{\|\mathcal{Z}\|}\mathcal{Z}\| \\ & = & f(\frac{\delta}{\|\mathcal{Z}\|}\mathcal{Z}), \end{array}$$

which shows that when $\mathcal{X} = \frac{\delta}{\|\mathcal{Z}\|}\mathcal{Z}$, $f(\mathcal{X})$ attains its minimum. Therefore the projection $P_{\mathcal{F}_2}(\mathcal{Z})$ of $\mathcal{Z}$ onto $\mathcal{F}_2$ is

$$P_{\mathcal{F}_2}(\mathcal{Z}) = \left\{ \begin{array}{ll} \mathcal{Z}, & if \quad \|\mathcal{Z}\| \leq \delta, \\ \frac{\delta}{\|\mathcal{Z}\|}\mathcal{Z}, & if \quad \|\mathcal{Z}\| > \delta. \end{array} \right.$$

□

By Dykstra's algorithm and combining Theorems 3.3-3.4, we derive a new algorithm to solve Problem II as follows.

Algorithm 3.5 (Dykstra's algorithm for solving Problem II)**.**

```
1. Set the initial value $\mathcal{X}_0 = \mathcal{A}$;
2. Set $\mathcal{X}_2^{(0)} = X_0,\ \ \mathcal{I}_1^{(0)} = 0,\ \ \mathcal{I}_2^{(0)} = 0$;
3. For $k = 1, 2, 3, \ldots$
       $\mathcal{X}_0^{(k)} = \mathcal{X}_2^{(k-1)}$,
    $\mathcal{X}_1^{(k)} = P_{H_{m,n}}(\mathcal{X}_0^{(k)} - \mathcal{I}_1^{(k-1)})$,
    $\mathcal{I}_1^{(k)} = \mathcal{X}_1^{(k)} - (\mathcal{X}_0^{(k)} - \mathcal{I}_1^{(k-1)})$,
    $\mathcal{X}_2^{(k)} = P_{\digamma_i}(\mathcal{X}_1^{(k)} - \mathcal{I}_2^{(k-1)})$,
    $\mathcal{I}_2^{(k)} = \mathcal{X}_2^{(k)} - (\mathcal{X}_1^{(k)} - \mathcal{I}_2^{(k-1)})$,
End
```

By Lemma 3.2, we get the convergence theorem for Algorithm 3.1.

Theorem 3.6. *If the set $\digamma_i \bigcap H_{m,n}$ is nonempty, then the sequences $\{\mathcal{X}_1^{(k)}\}$ and $\{\mathcal{X}_2^{(k)}\}$ generated by Algorithm 3.1 converge to the projection $P_{\digamma_i \bigcap H_{m,n}}(\mathcal{A})$, which is the unique solution of Problem II.*

In order to accelerate the convergence of Dykstra's algorithm, the Anderson acceleration algorithm was used in Algorithm 3.1, that is, we will design Anderson acceleration of Dykstra's algorithm to solve Problem II. Set $\Delta\mathcal{S}^{(0)} = 0, \mathcal{X}_2^{(0)} = \mathcal{A}$ and define the function

$$[\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \Delta\mathcal{S}^{(k)}] = g(\mathcal{X}_2^{(k-1)}, \Delta\mathcal{S}^{(k-1)}), \tag{3.2}$$

where

$$\begin{aligned}
&\mathcal{R}^{(k)} = \mathcal{X}_2^{(k-1)} - \Delta\mathcal{S}^{(k-1)},\\
&\mathcal{X}_1^{(k)} = P_{H_{m,n}}(\mathcal{R}^{(k)}),\\
&\Delta\mathcal{S}^{(k)} = \mathcal{X}_1^{(k)} - \mathcal{R}^{(k)},\\
&\mathcal{X}_2^{(k)} = P_{F_i}(\mathcal{X}_1^{(k)}).
\end{aligned}$$

Then Algorithm 3.1 can be recast as a fixed point method (see [16] for more details)

$$[\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \triangle\mathcal{S}^{(k)}] = g(\mathcal{X}_2^{(k-1)}, \Delta\mathcal{S}^{(k-1)}), \quad k = 0, 1, 2, \ldots.$$

Combining **Anderson acceleration algorithm**, the Anderson acceleration of Dykstra's algorithm to solve Problem II is as follows.

Algorithm 3.7 (Anderson acceleration of Dykstra's algorithm for solving Problem II)**.**
1. Run Anderson acceleration algorithm on the function $f : R^{2n^m} \to R^{2n^m}$ given by $f(z) = vec(\tilde{g}(\mathcal{Z}) - \mathcal{Z})$, where $z_k = vec(\mathcal{Z}_k), \mathcal{Z}_k = (\mathcal{X}_2^{(k)}, \Delta\mathcal{S}^{(k)}) \in R^{n \times 2n \times \cdots \times n}$ and $[\mathcal{X}_1^{(k)}, \tilde{g}(\mathcal{Z}_k)] = g(\mathcal{Z}_k)$ for the function g defined by (3.2). Terminate the iteration when the stopping criteria attains. Denote the result by $\mathcal{X}_*$.
2. Output the solution $\widehat{\mathcal{X}} = vec^{-1}(\mathcal{X}_*)$.

It is worthy to point out that, unlike Algorithm 3.1, Algorithm 3.7 is not guaranteed to converge, since there are no suitable convergence results for Anderson acceleration. Whether convergence can be proved under reasonable assumptions is an open question.

We begin to discuss the other acceleration scheme for Dykstra's Algorithm (see [19] for more details). For that we need to consider an auxiliary sequence in the product space $H \times H$ that will be denoted as H^2. For each $k \geq 0$, let's define

$$\hat{\mathcal{X}}_k = (\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}) \in H^2,$$

where $\mathcal{X}_1^{(k)}$ and $\mathcal{X}_2^{(k)}$ are defined in Dykstra's Algorithm. The idea to accelerate the convergence of Dykstra's Algorithm is building another sequence in H^2 which also converges to $P_{C_1 \bigcap C_1 \bigcap \cdots \bigcap C_n}(\mathcal{X}_0)$. Based on this idea, Lopez and Raydan [19] designed an acceleration scheme of Dykstra's algorithm is as follows

$$Z_k = \tag{3.3}$$
$$\begin{cases} \frac{1}{2}(\mathcal{X}_2^{(k+1)} + \mathcal{X}_1^{(k+1)}), & if \ \mid \|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\| + < \mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}, \mathcal{X}_1^{(k+1)} - X_2^{(k+1)} > \mid < \varepsilon, \\ \mathcal{X}_2^{(k)} + \alpha_k(\mathcal{X}_2^{(k+1)} - \mathcal{X}_2^{(k)}), & if \ \mid \|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\| + < \mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}, \mathcal{X}_1^{(k+1)} - X_2^{(k+1)} > \mid \geq \varepsilon, \end{cases}$$

where $\varepsilon > 0$ is a small given safeguard fixed real number and

$$\alpha_k = \frac{\|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\|^2}{\|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\|^2 + < \mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}, \mathcal{X}_1^{(k+1)} - X_2^{(k+1)} >}.$$

By making use of (3.3), the second accelerating version of Dykstra's algorithm is designed to solve problem II as follows

Algorithm 3.8 (Acceleration Dykstra's algorithm for solving Problem II)**.**

```
1. Given the initial value $\mathcal{X}_0, 0 < tol < 1, 0 < \varepsilon < 1$, set $k = 1$ and $error = 1$.
2. $\mathcal{X}_2^{(0)} = \mathcal{X}_0, \mathcal{I}_1^{(0)} = 0, \mathcal{I}_2^{(0)} = 0$.
3.      $\mathcal{X}_0^{(1)} = \mathcal{X}_2^{(0)}$,
     $\mathcal{X}_1^{(1)} = P_{H_{m,n}}(\mathcal{X}_0^{(1)} - \mathcal{I}_1^{(0)})$,
     $\mathcal{I}_1^{(1)} = \mathcal{X}_1^{(1)} - (\mathcal{X}_0^{(1)} - \mathcal{I}_1^{(0)})$,
     $\mathcal{X}_2^{(1)} = P_{\mathcal{F}_i}(\mathcal{X}_1^{(1)} - \mathcal{I}_2^{(0)})$,
     $\mathcal{I}_2^{(1)} = \mathcal{X}_2^{(1)} - (\mathcal{X}_1^{(1)} - \mathcal{I}_2^{(0)})$.

4. Set $\mathcal{Z}_0 = \mathcal{X}_2^{(1)}$.
5. While $error > tol$
     $\mathcal{X}_0^{(k+1)} = \mathcal{X}_2^{(k)}$,
     $\mathcal{X}_1^{(k+1)} = P_{H_{m,n}}(\mathcal{X}_0^{(k+1)} - \mathcal{I}_1^{(k)})$,
     $\mathcal{I}_1^{(k+1)} = \mathcal{X}_1^{(k+1)} - (\mathcal{X}_0^{(k+1)} - \mathcal{I}_1^{(k)})$,
     $\mathcal{X}_2^{(k+1)} = P_{\mathcal{F}_i}(\mathcal{X}_1^{(k+1)} - \mathcal{I}_2^{(k)})$,
     $\mathcal{I}_2^{(k+1)} = \mathcal{X}_2^{(k+1)} - (\mathcal{X}_1^{(k+1)} - \mathcal{I}_2^{(k)})$,
     if $\left| \|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\| + < \mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}, \mathcal{X}_1^{(k+1)} - X_2^{(k+1)} > \right| \geq \varepsilon$, then compute
     $\alpha_k = \frac{\|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\|^2}{\|\mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}\|^2 + < \mathcal{X}_2^{(k)} - \mathcal{X}_1^{(k)}, \mathcal{X}_1^{(k+1)} - X_2^{(k+1)} >}$,
     $\mathcal{Z}_k = \mathcal{X}_2^{(k)} + \alpha_k(\mathcal{X}_2^{(k+1)} - \mathcal{X}_2^{(k)})$,
     else
     $\mathcal{Z}_k = \frac{1}{2}(\mathcal{X}_2^{(k+1)} + \mathcal{X}_1^{(k+1)})$,
     end if
     $error = \|\mathcal{Z}_k - \mathcal{Z}_0\|$, $\mathcal{Z}_0 = \mathcal{Z}_k$, and $k \leftarrow k + 1$.
     end while
```

By Theorem 3.6 of Lopez and Raydan [19], we get the convergence theorem of Algorithm 3.8.

Theorem 3.9. *The sequence $\{Z_k\}$ generated by Algorithm 3.8 converges to the projection $P_{\mathcal{F}_i \bigcap H_{m,n}}(\mathcal{A})$, which is the unique solution of Problem II.*

4 Numerical Experiments

In this section, we present some numerical examples to illustrate that Algorithms 2.1 and 3.1, 3.7, 3.8 are feasible and effective to solve Problems I and II, respectively. All tests are performed by using MATLAB R2013a on a PC with an Pentium(R) Dual-core CPU at 2.8GHz.

4.1 Algorithm 2.1 for solving Problem I

In this subsection, we first present a simple example to illustrate that Algorithm 2.1 is feasible to solve Problem I, and then compare Algorithm 2.1 with the nonlinear conjugate gradient method (denoted by **NCG method**), the steepest descent method (denoted by **SD method**) (see [18]) and the alternating least squares method (denoted by **ALS method**) in Trickett-Burroughs-Milton [33]. For these algorithms, the stopping criterion is that either the iteration step has reached the upper limit 1000 or the norm of the gradient

$$Err_1(\eta_k) = \|\nabla f(\eta_k)\|^2 = \nabla f(\eta_k)^T \nabla f(\eta_k) \leq 1 \times 10^{-6}.$$

Example 4.1. In the seismic trace interpolation problem (see [33]), the raw frequency slice

$$S = \begin{pmatrix} 0.02 & ? & 0.99 \\ ? & 0.99 & 0.03 \\ 0.99 & 0.03 & 0.02 \end{pmatrix},$$

with two spatial dimensions with lengths $s_1 = 3$ and $s_2 = 3$. Here the symbol "?" means that the data missed. By the formula

$$\mathcal{A}(i, j, m, n) = S(i + j - 1, m + n - 1),$$

we can form a 4th order 2-dimensional tensor $\mathcal{A}$, i.e.

$$\mathcal{A}(:, :, 1, 1) = \begin{pmatrix} 0.02 & ? \\ ? & 0.09 \end{pmatrix}, \mathcal{A}(:, :, 2, 1) = \begin{pmatrix} ? & 0.99 \\ 0.99 & 0.03 \end{pmatrix},$$

$$\mathcal{A}(:, :, 1, 2) = \begin{pmatrix} ? & 0.99 \\ 0.99 & 0.03 \end{pmatrix}, \mathcal{A}(:, :, 2, 2) = \begin{pmatrix} 0.99 & 0.03 \\ 0.03 & 0.02 \end{pmatrix},$$

We consider Problem I with $r = 2$, $\mathcal{A}$ and

$$\mathcal{W}(:, :, 1, 1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathcal{W}(:, :, 2, 1) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

$$\mathcal{W}(:, :, 1, 2) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \mathcal{W}(:, :, 2, 2) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

We use Algorithm 2.1 to solve this problem with $B_0 = I$ and the initial values

$$\eta_0 = [0.6787, 0.7577, 0.7431, 0.3922]^T.$$

After 15 iterations, we get the solution $\hat{\mathcal{X}}$ of problem I is as follows

$$\hat{\mathcal{X}}(:, :, 1, 1) = \begin{pmatrix} 1.7025 & 0.6310 \\ 0.6310 & 0.2513 \end{pmatrix}, \hat{\mathcal{X}}(:, :, 2, 1) = \begin{pmatrix} 06310 & 0.2513 \\ 0.2513 & 0.1060 \end{pmatrix},$$

$$\hat{\mathcal{X}}(:,:,1,2)=\begin{pmatrix}06310 & 0.2513\\ 0.2513 & 0.1060\end{pmatrix},\hat{\mathcal{X}}(:,:,2,2)=\begin{pmatrix}0.2513 & 0.1060\\ 0.1060 & 0.0467\end{pmatrix},$$

and the norm of the gradient $Err_1(\eta_{15}) = 7.9288 \times 10^{-7}$.

Example 4.1 shows that Algorithm 2.1 is feasible to solve Problem I.

Example 4.2. In the seismic trace interpolation problem (see [33]), the raw frequency slice

$$S=\begin{pmatrix}0.76 & 0.17 & 0.25 & ? & 0.92\\ 0.17 & 0.25 & ? & 0.92 & 0.01\\ 0.25 & ? & 0.92 & 0.01 & 0.72\\ ? & 0.92 & 0.01 & 0.72 & 0.30\\ 0.92 & 0.01 & 0.72 & 0.30 & 0.71\end{pmatrix},$$

with two spatial dimensions with lengths $s_1 = 5$ and $s_2 = 5$. Here the symbol "?" means that the data missed. By the formula

$$\mathcal{A}(i,j,m,n)=S(i+j-1,m+n-1),$$

we can form a 4th order 3-dimensional tensor $\mathcal{A}$, i.e.

$$\mathcal{A}(:,:,1,1)=\begin{pmatrix}0.76 & 0.17 & 0.25\\ 0.17 & 0.25 & ?\\ 0.25 & ? & 0.92\end{pmatrix},\mathcal{A}(:,:,2,1)=\begin{pmatrix}0.17 & 0.25 & ?\\ 0.25 & ? & 0.92\\ ? & 0.92 & 0.01\end{pmatrix},$$

$$\mathcal{A}(:,:,3,1)=\begin{pmatrix}0.25 & ? & 0.92\\ ? & 0.92 & 0.01\\ 0.92 & 0.01 & 0.72\end{pmatrix},\mathcal{A}(:,:,1,2)=\begin{pmatrix}0.17 & 0.25 & ?\\ 0.25 & ? & 0.92\\ ? & 0.92 & 0.01\end{pmatrix},$$

$$\mathcal{A}(:,:,2,2)=\begin{pmatrix}0.25 & ? & 0.92\\ ? & 0.92 & 0.01\\ 0.92 & 0.01 & 0.72\end{pmatrix},\mathcal{A}(:,:,3,2)=\begin{pmatrix}? & 0.92 & 0.01\\ 0.92 & 0.01 & 0.72\\ 0.01 & 0.72 & 0.3\end{pmatrix},$$

$$\mathcal{A}(:,:,1,3)=\begin{pmatrix}0.25 & ? & 0.92\\ ? & 0.92 & 0.01\\ 0.92 & 0.01 & 0.72\end{pmatrix},\mathcal{A}(:,:,2,3)=\begin{pmatrix}? & 0.92 & 0.01\\ 0.92 & 0.01 & 0.72\\ 0.01 & 0.72 & 0.3\end{pmatrix},$$

$$\mathcal{A}(:,:,3,3)=\begin{pmatrix}0.92 & 0.01 & 0.72\\ 0.01 & 0.72 & 0.3\\ 0.72 & 0.3 & 0.71\end{pmatrix}.$$

We consider Problem I with $r = 1$, $\mathcal{A}$ and

$$\mathcal{W}(:,:,1,1)=\begin{pmatrix}1&1&1\\1&1&0\\1&0&1\end{pmatrix},\mathcal{W}(:,:,2,1)=\begin{pmatrix}1&1&0\\1&0&1\\0&1&1\end{pmatrix},\mathcal{W}(:,:,3,1)=\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix},$$

$$\mathcal{W}(:,:,1,2)=\begin{pmatrix}1&1&0\\1&0&1\\0&1&1\end{pmatrix},\mathcal{W}(:,:,2,2)=\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix},\mathcal{W}(:,:,3,2)=\begin{pmatrix}0&1&1\\1&1&1\\1&1&1\end{pmatrix},$$

$$\mathcal{W}(:,:,1,3)=\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix},\mathcal{W}(:,:,2,3)=\begin{pmatrix}0&1&1\\1&1&1\\1&1&1\end{pmatrix},\mathcal{W}(:,:,3,3)=\begin{pmatrix}1&1&1\\1&1&1\\1&1&1\end{pmatrix}.$$

We use Algorithm 2.1, NCG method and SD method to solve this problem with initial value $\eta_0 = [0.1678, 0.755]^T$, and get solution $\hat{\mathcal{X}}$ to Problem 1.1 as follows

$$\hat{\mathcal{X}}(:,:,1,1) = \begin{pmatrix} 1.0656 & 0.3959 & 0.1471 \\ 0.3959 & 0.1471 & 0.0547 \\ 0.1471 & 0.0547 & 0.0203 \end{pmatrix}, \hat{\mathcal{X}}(:,:,2,1) = \begin{pmatrix} 0.3959 & 0.1471 & 0.0547 \\ 0.1471 & 0.0547 & 0.0203 \\ 0.0547 & 0.0203 & 0.0075 \end{pmatrix},$$

$$\hat{\mathcal{X}}(:,:,3,1) = \begin{pmatrix} 0.1471 & 0.0547 & 0.0203 \\ 0.0547 & 0.0203 & 0.0075 \\ 0.0203 & 0.0075 & 0.0028 \end{pmatrix}, \hat{\mathcal{X}}(:,:,1,2) = \begin{pmatrix} 0.3959 & 0.1471 & 0.0547 \\ 0.1471 & 0.0547 & 0.0203 \\ 0.0547 & 0.0203 & 0.0075 \end{pmatrix},$$

$$\hat{\mathcal{X}}(:,:,2,2) = \begin{pmatrix} 0.1471 & 0.0547 & 0.0203 \\ 0.0547 & 0.0203 & 0.0075 \\ 0.0203 & 0.0075 & 0.0028 \end{pmatrix}, \hat{\mathcal{X}}(:,:,3,2) = \begin{pmatrix} 0.0547 & 0.0203 & 0.0075 \\ 0.0203 & 0.0075 & 0.0028 \\ 0.0075 & 0.0028 & 0.0010 \end{pmatrix},$$

$$\hat{\mathcal{X}}(:,:,1,3) = \begin{pmatrix} 0.1471 & 0.0547 & 0.0203 \\ 0.0547 & 0.0203 & 0.0075 \\ 0.0203 & 0.0075 & 0.0028 \end{pmatrix}, \hat{\mathcal{X}}(:,:,2,3) = \begin{pmatrix} 0.0547 & 0.0203 & 0.0075 \\ 0.0203 & 0.0075 & 0.0028 \\ 0.0075 & 0.0028 & 0.0010 \end{pmatrix},$$

$$\hat{\mathcal{X}}(:,:,3,3) = \begin{pmatrix} 0.0203 & 0.0075 & 0.0028 \\ 0.0075 & 0.0028 & 0.0010 \\ 0.0028 & 0.0010 & 0.0004 \end{pmatrix}.$$

The convergence curves of $\log_{10}(Err_1(\eta_k))$ and the objective function $\|\mathcal{A} - \mathcal{X}_k\|^2$ are given Figure 1.

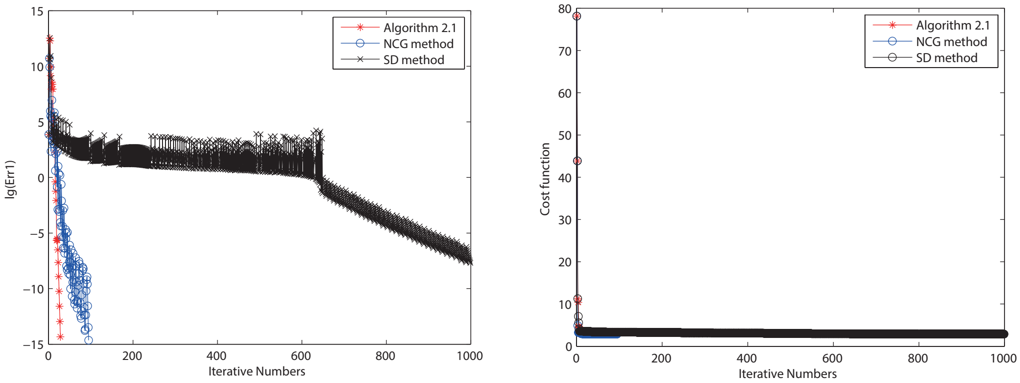

Figure 1: Convergence curves of $\log_{10}(Err_1(\eta_k))$ and the cost function $\|\mathcal{A} - \mathcal{X}_k\|^2$.

Example 4.3. Choose a 25-dimensional random vector v with missing data, and then construct a 4th order 8-dimensional Hankel tensor $\mathcal{A}$ by making use of (1.1). We can use the following way to form a 4th order 8-dimensional 0-1 tensor $\mathcal{W} = (w_{i_1 i_2, theorem, i_m})$. If the element $a_{i_1 i_2,\ldots,i_m}$ of $\mathcal{A}$ is unknown then $w_{i_1 i_2,\ldots,i_m} = 0$, otherwise $w_{i_1 i_2,\ldots,i_m} = 1$. Consider Problem I with $s = 1$, $\mathcal{A}$ and $\mathcal{W}$. We use Algorithm 2.1, NCG method and SD method to solve problem I with the same initial values, which is generated by MATLAB function rand(·). The convergence curves of $\log_{10}(Err_1(\eta_k))$ and the cost function $\|\mathcal{A} - \mathcal{X}_k\|^2$ are given in Figure 2.

From Figures 1 and 2 we can see that Algorithm 2.1 and NCG method both work very effectively for Problem I, while the convergence rate of Algorithm 2.1 is slight faster than NCG method.

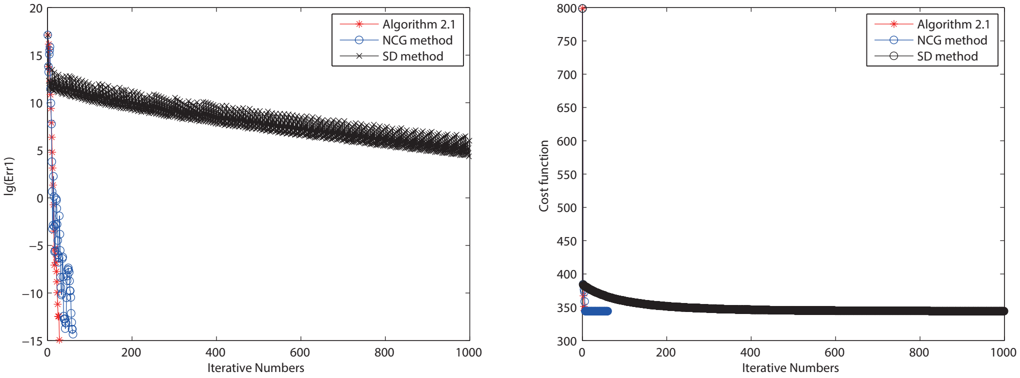

Figure 2: Convergence curves of $\log_{10}(Err_1(\eta_k))$ and the cost function $\|\mathcal{A} - \mathcal{X}_k\|^2$.

Example 4.4. Choose a $(n-1)m-$dimensional random vector v with missing data, and then construct a m order $n-$dimensional Hankel tensor $\mathcal{A}$ by making use of (1.1). We can use the following way to form a m order $n-$dimensional $0-1$ tensor $\mathcal{W} = (w_{i_1 i_2 \cdots m})$. If the element $a_{i_1 i_2 \ldots m}$ of $\mathcal{A}$ is unknown, then $w_{i_1 i_2 \ldots i_m} = 0$, otherwise $w_{i_1 i_2 \ldots i_m} = 1$. Consider Problem I with $s = \frac{n}{s}$, $\mathcal{A}$ and $\mathcal{W}$. We use Algorithm 2.1 and the ALS method in Trickett-Burroughs-Milton [33] to solve this problem with the same initial value. The experiment results are reported in Table 1.

Table 1: Comparative results of Example 4.4 for different values m and n.

$m = 11, n = 20$	IT	$CPU(s)$	Err_1	$Costfunction$
Algorithm 2.1	25	12.95	1.46×10^{-7}	27.74
ALS method	41	27.63	2.54×10^{-7}	27.71
$m = 21, n = 30$	IT	$CPU(s)$	Err_1	$Costfunction$
Algorithm 2.1	47	51.54	2.23×10^{-7}	321.36
ALS method	78	95.32	5.58×10^{-7}	321.30
$m = 31, n = 40$	IT	$CPU(s)$	Err_1	$Costfunction$
Algorithm 2.1	142	177.29	3.52×10^{-7}	713.13
ALS method	289	394.82	2.86×10^{-7}	713.09

From Table 1 we can see that Algorithm 2.1 has a faster convergence rate than the ALS method in Trickett-Burroughs-Milton [33].

4.2 Algorithms 3.5, 3.7, 3.8 for solving Problem II

In this subsection, we present some numerical examples to illustrate that Algorithms 3.5, 3.7, 3.8 are feasible to solve Problems II, and the comparison of these algorithms are also given. For these algorithms, the stopping criterion is that either the iteration step has reached the upper limit 1000 or the iterative error

$$Err_2(\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}) = \|\mathcal{X}_1^{(k)} - \mathcal{X}_2^{(k)}\| \leq 1 \times 10^{-9}.$$

Example 4.5. Consider Problem II with the tensors

$$\mathcal{A} = \left(\begin{array}{ccc|ccc|ccc} 2 & 5 & 6 & 9 & 2 & 7 & 0 & 0 & 2 \\ 8 & 9 & 4 & 9 & 9 & 5 & 5 & 9 & 4 \\ 4 & 4 & 2 & 5 & 2 & 8 & 4 & 10 & 7 \end{array}\right),$$

$$\mathcal{U} = \left(\begin{array}{ccc|ccc|ccc} -0.55 & -1.05 & 0.15 & -0.95 & 0.45 & 0.55 & 0.25 & -0.15 & -0.55 \\ -0.75 & -0.35 & -0.05 & 0.35 & 0.05 & -0.95 & -0.15 & -0.25 & -0.7 \\ 0.35 & 0.25 & -0.45 & 0.35 & -0.65 & -0.7 & -0.15 & -0.4 & -0.45 \end{array}\right)$$

and

$$\mathcal{V} = \left(\begin{array}{ccc|ccc|ccc} 1.75 & 1.05 & 1.85 & 0.95 & 1.55 & 1.05 & 1.75 & 1.75 & 1.15 \\ 0.75 & 2.35 & 1.65 & 1.65 & 1.55 & 1.55 & 1.75 & 0.85 & 0.7 \\ 1.65 & 1.35 & 1.05 & 1.25 & 1.25 & 0.7 & 0.75 & 0.4 & 1.45 \end{array}\right).$$

We use Algorithms 3.5, 3.7, 3.8 to solve Problem II with the initial values

$$\mathcal{X}_0 = \left(\begin{array}{ccc|ccc|ccc} 0.3 & 0.7 & 0.8 & 0.8 & 0.7 & 0.3 & 0.7 & 0.3 & 0.7 \\ 0.9 & 0.1 & 0.1 & 0.3 & 0.7 & 0.7 & 0.2 & 1 & 0.8 \\ 0.2 & 0.3 & 0.3 & 0.6 & 0.7 & 0.8 & 0.3 & 0.8 & 0.4 \end{array}\right).$$

We get the solution of Problem II as follows

$$\hat{\mathcal{X}} = \left(\begin{array}{ccc|ccc|ccc} 0.3 & 0.75 & 0.4667 & 0.75 & 0.4667 & 0.55 & 0.4667 & 0.55 & 0.6167 \\ 0.75 & 0.4667 & 0.55 & 0.4667 & 0.55 & 0.6167 & 0.55 & 0.6167 & 0.4 \\ 0.4667 & 0.55 & 0.6167 & 0.55 & 0.6167 & 0.4 & 0.6167 & 0.4 & 0.4 \end{array}\right).$$

The convergence curves of $\log_{10}(Err_2)$ is given in Figure 3

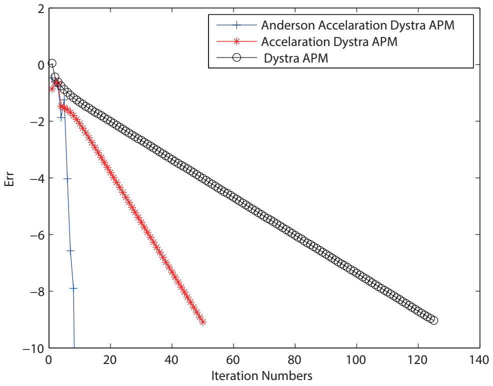

Figure 3: Convergence curves of $\log_{10}(Err_2)$.

Example 4.6. Consider Problem II with the structure constraint $\mathcal{F}_2$. Set the tensor

$$\mathcal{A} = \left(\begin{array}{ccc|ccc|ccc} 9 & 5 & 10 & 8 & 3 & 5 & 1 & 4 & 10 \\ 5 & 8 & 3 & 8 & 8 & 8 & 7 & 3 & 6 \\ 8 & 3 & 1 & 9 & 5 & 10 & 2 & 8 & 7 \end{array}\right)$$

and $\delta = 1$. We use Algorithms 3.5, 3.7, 3.8 to solve Problem II with the initial value

$$\mathcal{X}_0 = \left(\begin{array}{ccc|ccc|ccc} 0.9 & 0.4 & 0.4 & 0.5 & 0.3 & 0.3 & 0.1 & 0.6 & 0.2 \\ 0.7 & 0.6 & 0.4 & 0.9 & 0.6 & 0.6 & 0.4 & 0.5 & 0.6 \\ 0.6 & 0.4 & 0.5 & 0.2 & 0.5 & 0.9 & 0.1 & 0.8 & 0.5 \end{array}\right).$$

We get the solution of Problem II as follows

$$\hat{\mathcal{X}} = \left(\begin{array}{ccc|ccc|ccc} 0.3345 & 0.1982 & 0.1796 & 0.1982 & 0.1796 & 0.1540 & 0.1796 & 0.1540 & 0.1487 \\ 0.1982 & 0.1796 & 0.1540 & 0.1796 & 0.1540 & 0.1487 & 0.1540 & 0.1487 & 0.2849 \\ 0.1796 & 0.1540 & 0.1487 & 0.1540 & 0.1487 & 0.2849 & 0.1487 & 0.2849 & 0.1858 \end{array}\right).$$

The convergence curves of $\log_{10}(Err_2)$ are given in Figure 4.

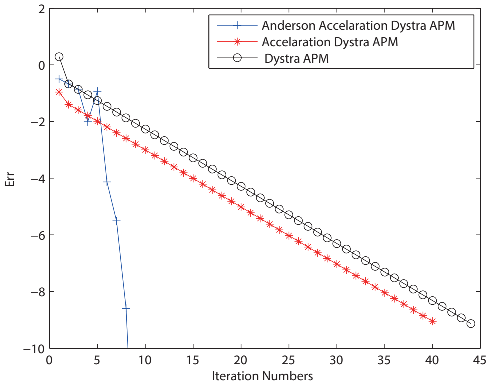

Figure 4: Convergence curves of $\log_{10}(Err_2)$.

Example 4.7. Consider Problem II with the structure constraint $\digamma_1$. The tensors $\mathcal{A} \in T_{3,n}$ generated by MATLAB function rand$(\cdot)$, and

$$\mathcal{U} = \mathcal{E} - u\mathcal{F} - v\mathcal{Z}, \mathcal{V} = \mathcal{E} + u\mathcal{F} + v\mathcal{Z}.$$

Here, $\mathcal{E}$ and $\mathcal{F}$ are Hankel tensors with following elements,

$$\mathcal{E}_{i_1 i_2 i_3} = a_{i_1+i_2+i_3-2},$$

$$\mathcal{E}_{i_1 i_2 i_3} = b_{i_1+i_2+i_3-2},$$

where $a \in R^{3n-2}$ and $b \in R^{3n-2}$ generated by MATLAB function rand$(\cdot)$. The tensor $\mathcal{Z} \in T_{3,n}$ generated by MATLAB function rand$(\cdot)$, and $u, v > 0$ are scalars. We use Algorithms 3.5, 3.7 to solve this problem and use the same initial tensor $\mathcal{X}_0$, which generated by MATLAB function rand$(\cdot)$. Under the stopping criterion, the experiment results including the iteration step (denoted by "IT"), CPU time (denoted by "$CPU(s)$"), iterative error (denoted by "Err_2") and the cost function (denoted by "$Costfunction$") are reported in table 2.

Example 4.8. Consider Problem II with the structure constraint $\digamma_2$. Here the tensors $\mathcal{A} \in T_{3,n}$ is generated by MATLAB function rand$(\cdot)$ and δ can be seen in Table 2. We use Algorithms 3.5, 3.7, 3.8 to solve this problem with the same initial tensor $\mathcal{X}_0$. Under the stopping criterion, the experiment results are reported in Table 3.

Several comments can be made on Figures 3, 4 and Tables 2, 3.

(1) From Figures 3, 4 and Tables 2, 3 we can see that Algorithms 3.7 and 3.8 work very effectively for Problem I, while the performance of Algorithm 3.7 is better than that of Algorithm 3.8 in term of iteration steps and computing time.

(2) In Tables 2 and 3, the symbol $'-'$ means that the iteration step k has reached the upper limit 1000, but it did not derive a solution. Compared with Algorithms 3.7 and 3.8, Algorithm 3.5 is relatively less efficient, especially when the problem size is large. The reason is that the convergence rate of Algorithm 3.5 is very slow.

Table 2: Comparative results of Example 4.6 for different values n, u and v.

$n = 10, u = 9, v = 1$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	26	2.4901	4.7399×10^{-13}	9.2569
Algorithm 3.8	453	67.2828	3.5422×10^{-10}	9.2569
Algorithm 3.5	482	81.6221	9.7847×10^{-10}	9.2569
$n = 50, u = 15, v = 1$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	88	131.21	6.8628×10^{-10}	189.28
Algorithm 3.8	461	709.36	1.8913×10^{-10}	189.28
Algorithm 3.5	489	973.21	1.1512×10^{-10}	189.28
$n = 100, u = 15, v = 1$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	112	16492	8.4990×10^{-10}	298.74
Algorithm 3.8	569	129970	9.9536×10^{-10}	298.74
Algorithm 3.5	–	–	–	–

Table 3: Comparative results of Example 4.8 for different values n and δ.

$n = 10, \delta = 8$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	9	1.3572	2.0244×10^{-12}	11.7646
Algorithm 3.8	17	3.0125	5.0406×10^{-10}	11.7646
Algorithm 3.5	34	5.1869	9.0246×10^{-10}	11.7646
$n = 50, \delta = 80$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	9	319.8291	1.3819×10^{-12}	140.4808
Algorithm 3.8	40	1509.8	7.6405×10^{-10}	140.4808
Algorithm 3.5	43	1533.5	5.5299×10^{-10}	140.4808
$n = 100, \delta = 100$	IT	$CPU(s)$	Err_2	$Costfunction$
Algorithm 3.7	9	2075.8	5.1599×10^{-10}	5699.1
Algorithm 3.8	103	28225	8.5377×10^{-10}	5699.1
Algorithm 3.5	–	–	–	–

$\boxed{5}$ Conclusion

In this paper, we consider two kinds of the best Hankel tensor approximations, which arise in the multidimensional seismic trace interpolator and the signal processing and the asset portfolio. One is the Hankel tensor approximation with missing data, and the other is the Hankel tensor approximation with the box constraint or the norm constraint. For the Hankel tensor approximation problem with missing data, the feasible set is characterized by the Vandermonde decomposition of Hankel tensor, and then the original problem is transformed into a smooth unconstrained optimization problem. The BFGS method with Armijo linear search is used to solve the equivalent problem. For the the Hankel tensor approximation problem with the box constraint or the norm constraint, we use Dykstra's algorithm to solve it. However, the main drawback of this algorithm is its slow convergence. So two acceleration versions of this algorithm are used in this paper. Some numerical examples show that these new algorithms are feasible and effective to solve Problems I and II.

References

[1] A.N. Akansu and M.U. Torun, Toeplitz approximation to empirical correlation matrix of asset returns: A signal processing perspective, *IEEE J. Select. Top. Signal Pro.* 6 (2012) 319–326.

[2] S. Al-Homidan, SQP algorithms for solving Toeplitz matrix approximation problem, *Numer. Linear Algebra Appl.* 9 (2002) 619–627.

[3] S. Al-Homidan, Hybrid methods for approximating Hankel matrix, *Numer Algor.* 32 (2003) 57–66.

[4] D.G. Anderson, Iterative procedures for nonlinear integral equations, *J. Assoc. Comput. Math.* 12 (1965) 547–560.

[5] K. Arnab, S. Pokala and R. Kumaresan, Toeplitz and Hankel matrix approximation using structured approach, in: *IEEE Inter. Confer. Acoustics, Speech and Signal Processing*, Berlin, 1998.

[6] H.H. Bauschke and J.M. Borwein, Dykstra's alternating projection algorithm for two sets, *J. Approx. Theory.* 79 (1994) 418–443.

[7] R. Badeau and R. Boyer, Fast multilinear singular value decomposition for structured tensors exponential data fitting using multilinear algebra: the single-channel and multichannel case, *SIAM J. Matrix Anal. Appl.* 30 (2008) 1008–1021.

[8] M. Boizard, R. Boyer, G. Favierz and P. Larzabaly, Fast multilinear singular value decomposition for higher-order Hankel tensors, in: *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop*, vol. 10, 2014, pp. 437–440.

[9] J.P. Boyle and R.L. Dykstra, A method for finding projections onto the intersections of convex sets in Hilbert spaces, *Lecture notes in Statistics.* 37 (1986) 28–47.

[10] P. Comon, G. Golub, L.H. Lim and B. Mourrain, Symmetric tensors and symmetric tensor rank, *SIAM J. Matrix Anal. Appl.* 30 (2008) 1254–1279.

[11] B.L. Chen, *Theories and Algorithms of Numerical Optimization*, Beijing, Tsinghua University Press, 2003, pp. 361–368.

[12] W. Ding, L.Qi and Y. Wei, Fast Hankel tensor-vector products and application to exponential data fitting, *Numer. Linear Algebra Appl.* 22 (2015) 814–832.

[13] R. Escalante and M. Raydan, Dykstra's algorithm for a constrained least-squares matrix problem, Numer. *Linear Algebra Appl.* 3 (1996) 459–471.

[14] M. Fazel, T.K. Pong, D. Sun and P. Tseng, Hankel matrix rank minimization with applications in system identification and realization, *SIAM J. Matrix Anal. Appl.* 34 (2013) 946–977.

[15] M. Haardt, F. Roemer and G. Del Galdo, Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems, *IEEE Trans. on Proc. Sig. Proc.* 56 (2008) 3198–3213.

[16] N.J. Higham, Computing a nearness symmetric positive semidefinite matrix, *Linear Algebra Appl.* 103 (1988) 103–118.

[17] N.J. Higham and N. Strabi, Anderson acceleration of the alternating projections method for computing the nearest correlation matrix, *Comput. Optim. Appl.* 72 (2016) 1021–1042.

[18] I.S. Iohvidov, *Toeplitz and Hankel Matrices and Forms*, Birkhauser, Boston, 1987, pp. 65–76.

[19] W. Lopez and M. Raydan, An acceleration scheme for Dykstra's algorithm, *Comput. Optim. Appl.* 63 (2016) 29–44.

[20] J.F. Li, W. Li and R. Huang, An efficient method for solving a matrix least squares problem over a matrix inequality constraint, *Comput. Optim. Appl.* 63 (2016) 393–423.

[21] N. Kreimer and M.D. Sacchi, Tensor completion via nuclear norm minimization for 5D seismic data reconstruction, *Geophysics.* 78 (2013) 273–284.

[22] P. M. Morillas, Dykstra's algorithm with strategies for projecting onto certain polyhedral cones, *Appl. Math. Comput.* 167 (2005) 635–649.

[23] G.S. Macinnes, The solution to a structured matrix approximation problem using Grassman coordinates, *SIAM J. Matrix Anal. Appl.* 26 (1999) 446–453.

[24] J.M. Papy, L. De Lathauwer and S. Van Huffel, Exponential data fitting using multilinear algebra: the single-channel and multi-channel case, *Numerical Linear Algebra Appl.* 12 (2005) 809–826.

[25] L.Q. Qi, Hankel tensors: associated Hankel matrices and Vandermonde decomposition, *Commun. Math. Sci.* 13 (2015) 113–125.

[26] L.Q. Qi, Symmetric nonnegative tensors and copositive tensors, *Commun. Math. Sci.* 439 (2013) 228–238.

[27] L.Q. Qi, C. Xu and Y. Xu, Nonnegative tensor factorization, completely positive tensors and an hierar-chically elimination algorithm, *SIAM J. Matrix Anal. Appl.* 35 (2014) 1227–1241.

[28] Y. Song and L.Q. Qi, Necessary and sufficient conditions for copositive tensors, *Linear Multilinear Algebra.* 63 (2015) 120–131

[29] Y. Song and L.Q. Qi, Properties of some classes of structured tensors. *J. Optim. Theory Appl.* 165 (2015) 854–873.

[30] P. Stoica and R. L. Moses, *Spectral Analysis of Signals*, Upper Saddle River, Prentice Hall, 2005, pp. 25–79.

[31] M. Tanaka and K. Nakata, Positive definite matrix approximation with condition number constraint, *Optim Lett.* 8 (2014) 939–947.

[32] A. Toth and C.T. Kelley, Convergence analysis for Anderson acceleration, *SIAM J. Numer. Anal.* 53 (2015) 805–819.

[33] S. Trickett, K. Burroughs and A. Milton, Interpolation using Hankel tensor completion, *Ratio.* 16 (2013) 3634–3638.

[34] H.F. Walker and P. Ni, Anderson acceleration for fixed-point iterations, *SIAM J. Numer. Anal.* 49 (2015) 1715–1735.

[35] H. Markowitz, Portfolio selection, *J. Finance* 7 (1952) 77–91.

[36] C.Q. Xu, Hankel tensors, Vandermonde tensors and their positivities, *Linear Algebra Appl.* 491 (2016) 56–72.

Manuscript received 24 December 2020
revised 15 June 2021
accepted for publication 18 June 2021

Chunmei Li
College of Mathematics and Computational Science
Guangxi Colleges and Universities
Key Laboratory of Data Analysis and Computation
Guilin University of Electronic Technology, Guilin 541004, P.R. China
E-mail address: lengyue123@126.com

Zhuling Jiang
School of Mathematics and Statistics, Xi'an Jiaotong University
Xi'an 710049, P.R. China
E-mail address: 543274306@qq.com

Xuefeng Duan
College of Mathematics and Computational Science
Guangxi Colleges and Universities
Key Laboratory of Data Analysis and Computation
Guilin University of Electronic Technology, Guilin 541004, P.R. China
E-mail address: guidian520@126.com