Approximate Simultaneous Diagonalization of Matrices via Structured Low-Rank Approximation

Riku AKEMA(a), Student Member, Masao YAMAGISHI(b), Member, and Isao YAMADA(c), Fellow

SUMMARY Approximate Simultaneous Diagonalization (ASD) is a problem to find a common similarity transformation which approximately diagonalizes a given square-matrix tuple. Many data science problems have been reduced into ASD through ingenious modelling. For ASD, the so-called Jacobi-like methods have been extensively used. However, the methods have no guarantee to suppress the magnitude of off-diagonal entries of the transformed tuple even if the given tuple has a common exact diagonalizer, i.e., the given tuple is simultaneously diagonalizable. In this paper, to establish an alternative powerful strategy for ASD, we present a novel two-step strategy, called Approximate-Then-Diagonalize-Simultaneously (ATDS) algorithm. The ATDS algorithm decomposes ASD into (Step 1) finding a simultaneously diagonalizable tuple near the given one; and (Step 2) finding a common similarity transformation which diagonalizes exactly the tuple obtained in Step 1. The proposed approach to Step 1 is realized by solving a Structured Low-Rank Approximation (SLRA) with Cadzow’s algorithm. In Step 2, by exploiting the idea in the constructive proof regarding the conditions for the exact simultaneous diagonalizability, we obtain a common exact diagonalizer of the obtained tuple in Step 1 as a solution for the original ASD. Unlike the Jacobi-like methods, the ATDS algorithm has a guarantee to find a common exact diagonalizer if the given tuple happens to be simultaneously diagonalizable. Numerical experiments show that the ATDS algorithm achieves better performance than the Jacobi-like methods.

key words: Approximate Simultaneous Diagonalization (ASD), Joint EigenValue Decomposition (JEVD), Structured Low-Rank Approximation (SLRA), alternating projection algorithm

1. Introduction

One of the most important tasks in data sciences is to extract certain common features from given multiple data [1,2]. Many approaches [3,13] reduce such tasks into a problem to find a common similarity transformation which diagonalizes simultaneously a certain matrix tuple observed under the influence of noise. A matrix tuple is called in this paper a simultaneously diagonalizable tuple (see Definition 1) if it has a common similarity transformation which diagonalizes exactly all matrices in the tuple. Let $\mathcal{D}_{n,K}$ be the set of all simultaneously diagonalizable tuples in $\mathbb{C}^{n \times n} \times \cdots \times \mathbb{C}^{n \times n} =: \times_{k=1}^{K} \mathbb{C}^{n \times n}$ (see Definition 1 for the precise definition of $\mathcal{D}_{n,K}$ and Appendix A for the nonconvexity of $\mathcal{D}_{n,K}$). For a given $\mathbf{A} = (A_1, \ldots, A_K) \in \mathcal{D}_{n,K}$, the exact simultaneous diagonalization requires to find $\mathbf{S} \in \mathbb{C}^{n \times n}$ such that $S^{-1}A_kS (k = 1, \ldots, K)$ are diagonal. In this paper, for a general $\mathbf{A} \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$, we consider the following problem.

Problem 1 (Approximate Simultaneous Diagonalization: ASD). For a given $\mathbf{A} = (A_1, \ldots, A_K) \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$, approximate $\mathbf{A}$ with a certain $\tilde{\mathbf{A}} = (\tilde{A}_1, \ldots, \tilde{A}_K) \in \mathcal{D}_{n,K}$ and find $\tilde{\mathbf{S}} \in \mathbb{C}^{n \times n}$ such that $\tilde{S}^{-1}\tilde{A}_k\tilde{S} (k = 1, \ldots, K)$ are diagonal.

A conventional nonconvex optimization model for the ASD of $\mathbf{A} = (A_1, \ldots, A_K) \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$ is to find a minimizer of $f_{\lambda}(\mathbf{S}) := \sum_{k=1}^{K} \text{off}(S^{-1}A_kS)$, where $\text{off} : [x_{i,j}]_{1,1}^{n,n} \mapsto \sum_{1 \leq i \neq j \leq n} |x_{i,j}|^2$. For this optimization model, the so-called Jacobi-like methods have been used extensively [5,10,14,15]. Although such methods can be applied directly to general matrix tuples, they can update only few variables in an estimate of a minimizer of $f_{\lambda}$ with certain parameterized matrices, e.g., the Givens rotation matrices and the shearing matrices. More importantly, the Jacobi-like methods have no guarantee to suppress $f_{\lambda}$ even for $\mathbf{A} \in \mathcal{D}_{n,K}^{\text{abl}}$ (see Example 1 in Section 2.2 for its details). On the other hand, the idea found in the constructive proof for a classical relation between diagonalizability and commutativity (see Fact 2) regarding the exact simultaneous diagonalizability suggests that we can solve the exact simultaneous diagonalization for $\mathbf{A} \in \mathcal{D}_{n,K}$ algebraically (Note: such an exact simultaneous diagonalization along this idea is found as, e.g., Diagonalize-One-then-Diagonalize-Other (DODO) method in [16]; see Algorithm 1 in Section 2.2). This situation suggests the possibility to establish a new powerful strategy which can supersede the Jacobi-like methods for ASD if a simultaneously diagonalizable tuple can be found as a good approximation of $\mathbf{A} \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$.

In this paper, to establish such a computational scheme, we present a novel two-step strategy, named Approximate-Then-Diagonalize-Simultaneously (ATDS) algorithm, as a practical solution of the following problem.

Problem 2 (Two steps for approximate simultaneous diagonalization). For a given $\mathbf{A} \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$,

Step 1 (Approximation): approximate $\mathbf{A}$ with a certain $\tilde{\mathbf{A}} = (\tilde{A}_1, \ldots, \tilde{A}_K) \in \mathcal{D}_{n,K}^{\text{abl}}$, where $\tilde{\mathbf{A}} = \mathbf{A}$ must be

In most data science applications [1,13] of ASD, the observed matrix tuple $\mathbf{A} = (A_1, \ldots, A_K) \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$ is modeled as a slightly perturbed version of a certain $(A_1^*, \ldots, A_K^*) \in \mathcal{D}_{n,K}^{\text{abl}}$.
employed if \( A \in \mathcal{D}_{n,K}^{abl} \).  

**Step 2 (Simultaneous diagonalization):** find a common exact diagonalizer of \( A = (A_1, \ldots, A_K) \), i.e., find \( S \in \mathbb{C}^{m \times n} \) s.t. \( S^{-1} A_k S \) \((k = 1, \ldots, K)\) are diagonal.

Since Step 2 can be solved by the DODO method, we focus on Step 1. The proposed approach to Step 1 is designed based on the fact that a necessary condition for \( X \in \mathcal{D}_{n,K}^{abl} \) can be translated into a certain rank condition for a structured matrix defined with \( X \) (see Theorem 1). More precisely, we propose to relax Step 1 by Step 1’ with Structured Low-Rank Approximation (SLRA) [17] and then propose to solve Step 1’ with Cadzow’s algorithm [18]. Unlike the Jacobi-like methods, the proposed ATDS algorithm has a guarantee to find a common exact diagonalizer if \( A \) happens to satisfy \( A \in \mathcal{D}_{n,K}^{abl} \). Numerical experiments show that, compared with the Jacobi-like methods, the proposed ATDS algorithm achieves a better approximation to the desired common similarity transformation at the expense of reasonable computational time.

A preliminary version of this paper was presented at a conference [19].

**Notations (see also Table 1).** Let \( \mathbb{N} \), \( \mathbb{R} \), and \( \mathbb{C} \) denote the set of all nonnegative integers, the set of all real numbers, and the set of all complex numbers, respectively. For \( x \in \mathbb{C}^n \), \( |x| \) denotes the Euclidean norm of \( x \). The matrix \( I_n \) denotes the \( n \)-by-\( n \) identity matrix. For \( X \in \mathbb{C}^{m \times n} \), \( X^\top \), \( X^* \), \( X^{-1} \), \( \text{tr}(X) \), \( \|X\|_F \), \( \|X\|_\infty \), and \( \text{nul}(X) \) denote respectively the transpose, the conjugate transpose, the inverse, the inverse transpose, the trace, the Frobenius norm, the range space, and the nullspace of \( X \). The mapping \( \text{vec} : \mathbb{C}^{m \times n} \rightarrow \mathbb{C}^{mn} \) denotes vectorization by stacking the columns of a matrix and \( \text{vec}^{-1} \) its inverse. For \( X \in \mathbb{C}^n \), \( \text{diag}(x) \in \mathbb{C}^{m \times n} \) is a diagonal matrix whose diagonal entries are given by the components of \( x \). The Kronecker product of \( X = [x_{ij}] \in \mathbb{C}^{m \times n} \) and \( Y \in \mathbb{C}^{p \times q} \) is \( X \otimes Y \in \mathbb{C}^{mp \times nq} \) whose \((i,j)\)th blocks are \( [X \otimes Y]_{ij} = x_{ij} Y \) \((i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)\). The Kratri-Rao product of \( X = [x_{ij}] \in \mathbb{C}^{m \times n} \) and \( Y = [y_{ij}] \in \mathbb{C}^{n \times d} \) is \( X \circ Y := [x_{ij} \otimes y_{ij}] \in \mathbb{C}^{m \times nd} \). The direct sum of \( \tilde{X}_p \in \mathbb{C}^{m_p \times n_p} \) \((p = 1, \ldots, d)\) and \( \tilde{X}_d \in \mathbb{C}^{m_d \times n_d} \) is \( \oplus_{p=1}^d \tilde{X}_p \in \mathbb{C}^{m \times d_n} \). The block diagonal matrix \( \tilde{X}_1 \circ \tilde{X}_2 \circ \cdots \circ \tilde{X}_d \in \mathbb{C}^{m \times n} \) with the \( i \)-th diagonal block \( \tilde{X}_i \) \((i = 1, 2, \ldots, d)\), span\{\(X_1, \ldots, X_d\)\} \(\subset \mathbb{C}^{m \times n} \) is the set of all linear combinations of \(X_1, \ldots, X_d\). For \( X = (X_1, \ldots, X_K) \in \mathbb{C}^{K \times m \times n} \), we define \( \|X\|_F := \sqrt{\sum_{k=1}^K \|X_k\|_F^2} \).

### 2. Preliminaries

#### 2.1 Useful Facts on Diagonalizability

Recall that \( X \in \mathbb{C}^{m \times n} \) is said to be diagonalizable if there exists \( S \in \mathbb{C}^{m \times n} \) such that \( S^{-1} X S \) is diagonal. In the following, we use \( \mathcal{D}_{n,K}^{abl} \subset \mathbb{C}^{m \times n} \) to denote the set of all diagonalizable matrices. The simultaneous diagonalizability in Definition 1 below is a natural extension of the diagonalizability of a matrix and defined for a matrix tuple.

**Definition 1 (Simultaneously diagonalizable tuple) [20] (Definition 1.3.20).** A matrix tuple \( X = (X_1, \ldots, X_K) \in \mathbb{C}^{K \times m \times n} \) is said to be simultaneously diagonalizable if there exists a common \( S \in \mathbb{C}^{m \times n} \) s.t. \( S^{-1} X_k S \) \((k = 1, \ldots, K)\) are diagonal. In this paper, we use \( \mathcal{D}_{n,K}^{abl} \) to denote the set of all simultaneously diagonalizable tuples in \( \mathbb{C}^{K \times m \times n} \).

The exact simultaneous diagonalization of \( X \in \mathcal{D}_{n,K}^{abl} \) is said to be essentially unique [3] if its exact common diagonalizer \( S \in \mathbb{C}^{m \times n} \) is determined uniquely up to a permutation and a scaling of the column vectors of \( S \). The following is well-known as an equivalent condition for the essential uniqueness.

**Fact 1 (Necessary and sufficient condition for essential uniqueness) [3] (Theorem 6.1).** Suppose that \( S \in \mathbb{C}^{m \times n} \) is a common diagonalizer of \( X = (X_1, \ldots, X_K) \in \mathcal{D}_{n,K}^{abl} \) satisfying \( \Lambda_k := S^{-1} X_k S = \text{diag}(\lambda_1^{(k)}, \ldots, \lambda_n^{(k)}) \), where \( \lambda_1^{(k)}, \ldots, \lambda_n^{(k)} \in \mathbb{C} \) are all eigenvalues of \( X_k \) \((k = 1, \ldots, K)\) (Note: The order of eigenvalues \( \lambda_1^{(k)}, \ldots, \lambda_n^{(k)} \) is determined according to \( S \)). Then, the exact simultaneous diagonalization of \( X \) is essentially unique if and only if \( [\lambda_1^{(1)}, \ldots, \lambda_1^{(K)}; \lambda_2^{(1)}, \ldots, \lambda_2^{(K)}; \ldots; \lambda_n^{(1)}, \ldots, \lambda_n^{(K)}] \neq [\lambda_q^{(1)}, \ldots, \lambda_q^{(K)}; \lambda_{q+1}^{(1)}, \ldots, \lambda_{q+1}^{(K)}; \ldots; \lambda_n^{(1)}, \ldots, \lambda_n^{(K)}] \) for any \( p \neq q; p, q \in \{1, \ldots, n\} \).

As seen above, the simultaneous diagonalizability of a matrix tuple requires not merely the diagonalizability of every matrix but also the commutativity of all matrices therein.

**Fact 2 (Necessary and sufficient condition for \( \mathcal{D}_{n,K}^{abl} \) [20] Theorem 1.3.21).** Let \( \mathcal{E}_{n,K} \subset \mathbb{C}^{K \times m \times n} \) denote the set of all tuples \( (X_1, \ldots, X_K) \) such that \( X_k X_l = X_l X_k \), i.e., \( X_k \) and \( X_l \) commute, for any \( k, l \in \{1, \ldots, K\} \). Then,
\[
\mathcal{D}_{n,K}^{abl} = \mathcal{E}_{n,K} \cap (\mathcal{D}_{n,K}^{abl})^K.
\]

That is, a tuple is (exactly) simultaneously diagonalizable if and only if every pair \( X_k \) and \( X_l \) commutes and every \( X_k \) is diagonalizable.

**Remark 1.** The constructive proof for Fact 2 found in, e.g., [20] can be translated into a finite-step algorithm (the
DODO method \cite{16} for Step 2 in Problem \ref{problem} (see Section \ref{section}).

\subsection{Jacobi-like methods}

To see the basic idea of the Jacobi-like methods, let us explain the scheme of sh-rt \cite{14}, for $A = (A_1, \ldots, A_K) \in \times_{K=1}^{K=n} \mathbb{R}^{m \times n}$, which is known as the one of the extensions, of the Jacobi methods \cite{21} (see also, e.g., Section 8.4) for a symmetric-matrix diagonalization, to simultaneous diagonalization. Sh-rt uses the shearing matrix $H(p, q, \phi) = [h_{ij}] \in \mathbb{R}^{n \times n}$ and the Givens rotation matrix $G(p, q, \theta) = [g_{ij}] \in \mathbb{R}^{n \times n}$, which are expressed as

\begin{align*}
    h_{ij} &= \begin{cases}
        1 & \text{if } i < j,
        \cos \phi & \text{if } i = j,
        -\cos \phi & \text{if } i > j.
    \end{cases} \\
    g_{ij} &= \begin{cases}
        1 & \text{if } i < j,
        \sin \theta & \text{if } i = j,
        -\sin \theta & \text{if } i > j.
    \end{cases}
\end{align*}

Therefore, since $\tilde{A}_k = \tilde{B}_k = A_k$ (hence $n = |A|$), we get $\tan \phi^\ast = 0$, i.e., $H(1, 2, \phi^\ast) = I_2$. Moreover, since $\tilde{b}_{1,1}^\ast - \tilde{b}_{2,2}^\ast = \tilde{a}_{1,1}^\ast - \tilde{a}_{2,2}^\ast = 0$ (hence $n = 2$ in (3)), we get $\tan 4\theta^\ast = 0$, i.e., $G(1, 2, \theta^\ast) = I_2$. Therefore, since $\tilde{A}_k = \tilde{B}_k = A_k$ (hence $n = 1$), we get $f(\tilde{A}_1, \tilde{A}_2) = f(A_1, A_2)$.

\subsection{The DODO Method}

The key idea behind the DODO method (Algorithm \ref{algorithm}) is to reduce the exact simultaneous diagonalization of $A$ into exact simultaneous diagonalizations of tuples of smaller matrices. To see how Algorithm \ref{algorithm} works, let us demonstrate its procedures.

(i) Choose $l \in \{1, \ldots, K\}$ s.t. $A_l \in \mathbb{D}^\text{abl}$ is not diagonal, arbitrarily. Diagonalize $A_l$ with $S_0 \in \mathbb{C}^{m \times n}$ as

\begin{equation}
    S_0^{-1} A_l S_0 = \Lambda_l^{(1)} I_{n_1} \oplus \cdots \oplus \Lambda_d^{(1)} I_{n_d},
\end{equation}

where $\Lambda_l^{(1)} \in \mathbb{C}^{n_1 \times n_1}$ are distinct and $\max\{n_1, \ldots, n_d\} < n$ with $n_1 + \cdots + n_d = n$ holds because $A_l$ is not a constant multiple of $I_n$ (Note: if $n_1 = \cdots = n_d = 1$, i.e., $d = n$ holds, $S_0$ is a common exact diagonalizer for $A_l$, which is verified essentially in the following procedure (ii)).

(ii) Compute $S_0^{-1} A_k S_0$ ($k = 1, \ldots, K; k \neq l$) which can be expressed as $S_0^{-1} A_k S_0 = \tilde{A}_1^{(k)} \oplus \cdots \oplus \tilde{A}_d^{(k)}$ with some $\tilde{A}_d^{(k)} \in \mathbb{C}^{n_1 \times n_1}$ for any $p = 1, \ldots, d$ (Note: this is verified by applying Fact \ref{fact}c) to the commutativity\footnote{Such $S_0$ can be computed by permuting column vectors of any $S$ satisfying $S^{-1} A_l S$ is diagonal.} of $S_0^{-1} A_k S_0$ and $S_0^{-1} A_l S_0$ in \ref{fact}).

Example 1 (A weakness of a Jacobi-like method: sh-rt). Suppose that $A_1 := [a_{i,j}] \in \mathbb{R}^{2 \times 2}$ and $A_2 = c_1I_2 + c_1A_1 \in \mathbb{R}^{2 \times 2}$, where $a_{1,1} = a_{2,2}$, $|a_{1,2}| \neq |a_{2,1}|$, $a_{1,2}, a_{2,1} > 0$, and $c_0, c_1 \neq 0$. Since the discriminant of the characteristic polynomial of $A_1$ is positive, $A_1$, and then $A_2$, have distinct real eigenvalues and therefore they are diagonalizable. By commutativity of $A_1$ and $A_2$, moreover, we see $A_1, A_2 \in \mathbb{D}^\text{abl}$ (see Fact \ref{fact}).

\begin{algorithm}

\textbf{Algorithm 1} The DODO method \cite{16}

\begin{algorithmic}

\Function{DODO}{$(A_1, \ldots, A_K), n$}
\If{$A_k$ ($k = 1, \ldots, K$) are diagonal}
\Return{$I_n$}
\Else
\State Diagonalize $A_l$ ($l \in \{1, \ldots, K\}$) as $S_0^{-1} A_l S_0$ of the form \ref{equation},
\State $S_0^{-1} A_k S_0 = \Lambda_1^{(k)} \oplus \cdots \oplus \Lambda_d^{(k)}$ ($k = 1, \ldots, K$)
\For{$p = 1$ to $d$}
\State $S_p = DODO((\Lambda_1^{(p)}, \ldots, \Lambda_d^{(p)}), n_p)$
\EndFor
\State Return $S = S_0(S_1 \oplus \cdots \oplus S_d)$
\EndIf
\EndFunction

\end{algorithmic}

\end{algorithm}

\begin{table}
\centering
\caption{Notations on the sets used in this paper.}
\begin{tabular}{|c|c|}
\hline
$\mathbb{N}$ & the nonnegative integers \\
$\mathbb{R}$ & the real numbers \\
$\mathbb{R}^{m \times n}$ & $m$-by-$n$ real matrices \\
$\mathbb{C}$ & the complex numbers \\
$\mathbb{C}^{m \times n}$ & $m$-by-$n$ complex matrices \\
$\mathbb{D}^\text{abl}$ & diagonalizable matrices in $\mathbb{C}^{m \times n}$ \\
$\mathbb{D}^\text{abl}, K$ & commuting tuples in $\times_{K=1}^{K=n} \mathbb{C}^{m \times n}$ \\
$\mathbb{D}^\text{abl}, n, K$ & simultaneously diagonalizable tuples in $\times_{K=1}^{K=n} \mathbb{C}^{m \times n}$ \\
\hline
\end{tabular}
\end{table}
(iii) Construct $\hat{A}_p := (\hat{A}_p^{(1)}, \ldots, \hat{A}_p^{(K)}) \in \bigotimes_{k=1}^K C^{n_p \times n_p}$ with $\hat{A}_p^{(i)} \sim A_p^{(i)} I_{n_p}$ (for $p = 1, \ldots, d$). Then, we obtain $\hat{A}_p \in D_{n,K}$ for all $p = 1, \ldots, d$ (Note on Commutativity: for any pair $(k_1, k_2) \in \{1, \ldots, K\}^2$, the commutativity of $S_0^{-1} A_k S_0 = \hat{A}_p^{(k_1)} \cdots \hat{A}_p^{(k_d)}$ and $S_0^{-1} A_k S_0 = \hat{A}_p^{(k_1)} \cdots \hat{A}_p^{(k_d)}$ (see (ii) for these structures) implies commutativity of $\hat{A}_p^{(k_1)}$ and $\hat{A}_p^{(k_2)}$ ($p = 1, \ldots, K$); on Diagonalizability: the diagonalizability of $S_0^{-1} A_k S_0$ (guaranteed by Fact 2) implies diagonalizability of $\hat{A}_p^{(k)}$ ($p = 1, \ldots, d$) (see Fact 4).

(iv) For each $\hat{A}_p$ ($p = 1, \ldots, K$), repeat (i-iii), where many $K$-tuples of smaller matrices may appear in the process, until all $\hat{A}_p^{(k)}$ ($k = 1, \ldots, K; p = 1, \ldots, d$) become diagonal.

(v) Compute a common exact diagonalizer for $A$ as $S := S_0 (S_1 \otimes \cdots \otimes S_d)$ (guaranteed by Fact 2) is a common exact diagonalizer of $\hat{A}_p$, for each $p = 1, \ldots, d$, constructed with a diagonalizer in (i) regarding $\hat{A}_p$.

3. Approximate-Then-Diagonalize-Simultaneously Algorithm

3.1 Simultaneous Diagonalizability Condition in terms of the Kronecker Sums

It is not hard to see that $X \in C^{n \times n}$ and $Y \in C^{m \times m}$ commute if and only if vec$(Y) \in$ nullspace$(I_n \otimes X - X^T \otimes I_n)$, where $I_n \otimes X - X^T \otimes I_n$ is called the Kronecker sum of $X$ and $-X^T$. This simple fact motivates us to introduce a linear mapping

$$
\Xi : \bigotimes_{k=1}^K C^{n \times n} \rightarrow C^{K n \times n^2}, \quad X := (X_1, X_2, \ldots, X_K) \mapsto \Xi(X) := \left[ \begin{array}{c} I_n \otimes X_1 - X_1^T \otimes I_n \\ I_n \otimes X_2 - X_2^T \otimes I_n \\ \vdots \\ I_n \otimes X_K - X_K^T \otimes I_n \end{array} \right] \in C^{K n \times n^2}. 
$$

(5)

Moreover, for $\hat{X} \in \Xi(\bigotimes_{k=1}^K C^{n \times n}) := \{ \Xi(Y) \in C^{K n \times n^2} \mid Y \in \bigotimes_{k=1}^K C^{n \times n} \}$, we introduce an affine subspace $\Xi^{-1}(\hat{X}) := \{ Y \in \bigotimes_{k=1}^K C^{n \times n} \mid \Xi(Y) = \hat{X} \} \subset \bigotimes_{k=1}^K C^{n \times n}$.

**Theorem 1** (Characterizations of $C_{n,K}$ and $D_{n,K}$ with $\Xi$). Let $X = (X_1, \ldots, X_K) \in \bigotimes_{k=1}^K C^{n \times n}$.

(a) $X \in C_{n,K} \Leftrightarrow (\forall k \in \{1, \ldots, K\}) \Xi(X) \text{vec}(X_k) = 0$.

(b) $X \in D_{n,K} \Leftrightarrow (\exists S \in C^{n \times n})$ nullspace$(\Xi(X)) \supset$ range$(S^T \otimes S)$.

(c) $X \in D_{n,K} \Rightarrow$

$$
\text{rank}(\Xi(X)) \leq n^2 - n, \quad \text{rank}(\Xi(X)) = n^2 - n \Leftrightarrow \text{exact simultaneous diagonalization of } X \text{ is essentially unique}.
$$

(d) If at least one $X_i$ has $n$ distinct eigenvalues, $X \in T_{n,K} \Leftrightarrow \text{rank}(\Xi(X)) = n^2 - n$.

(e) Let $\mathcal{E}_{n-n} := \{ \hat{X} \in C^{K n \times n^2} \mid \text{rank}(\hat{X}) \leq n^2 - n \}$. Suppose $\hat{X} \in \Xi(\bigotimes_{k=1}^K C^{n \times n}) \cap \mathcal{E}_{n-n}$ and $\Xi(Y) = \hat{X}$ for some $Y := (Y_1, \ldots, Y_K) \in \bigotimes_{k=1}^K C^{n \times n}$, where at least one $Y_1$ has $n$ distinct eigenvalues and is diagonalizable as $S_1 Y_1 S$ with $S \in C^{n \times n}$. Then, $Y \in T_{n,K}$ and $S_1 Y_1 S = (1, \ldots, K)$ are diagonal.

(The proof of Theorem 1 is given in Appendix D.)

**Remark 2** (On Theorem 1).

(a) Theorem 1(c) implies $\Xi^{-1}(\Xi(\bigotimes_{k=1}^K C^{n \times n}) \cap \mathcal{E}_{n-n}) := \{ X \in \bigotimes_{k=1}^K C^{n \times n} \mid \text{rank}(\Xi(X)) \leq n^2 - n \} \supset D_{n,K}$.

(b) Theorem 1(e) is a sufficient condition for $X \in D_{n,K}$ in terms of the range of $\Xi$. Although this condition is similar to (d), the condition in (e) directly motivates us to solve a Structured Low-Rank Approximation (SLRA) below (see Problem 2) for finding a simultaneously diagonalizable tuple near $A$.

For $\hat{X} \in \Xi(\bigotimes_{k=1}^K C^{n \times n})$, the projection onto $\Xi^{-1}(\hat{X})$ can be computed as in Proposition 1.

**Proposition 1** (On projection onto $\Xi^{-1}(\hat{X})$). Let $\hat{X} \in \Xi(\bigotimes_{k=1}^K C^{n \times n})$. Choose $X^\circ := (X_1^\circ, \ldots, X_K^\circ) \in \Xi^{-1}(\hat{X})$ arbitrarily. Then,

(a) $\Xi^{-1}(\hat{X}) = X^\circ + (\text{span}(I_n))^2$;

(b) the projection of $X = (X_1, \ldots, X_K) \in \bigotimes_{k=1}^K C^{n \times n}$ onto $\Xi^{-1}(\hat{X})$, i.e., $P_{\Xi^{-1}(\hat{X})} : (X_1, \ldots, X_K) \mapsto (Z_1, \ldots, Z_K) := \arg\min_{(Y_1, \ldots, Y_K) \in \Xi^{-1}(\hat{X})} \sum_{k=1}^K \| X_k - Y_k \|_F^2$,

(c) for $X = (X_1, \ldots, X_K)$ and $P_{\Xi^{-1}(\hat{X})} (X) = (Z_1, \ldots, Z_K)$, $\sum_{k=1}^K \| X_k - Z_k \|_F^2$ is given by $Z_k = X_k^\circ + (\text{tr}(X_k - X_k^\circ)/n) I_n$ ($k = 1, \ldots, K$);

(The proof of Proposition 1 is given in Appendix E.) Finally, by using Theorem 1(a), Proposition 1(b), and 1(c), we propose to relax Step 1 in Problem 2 by the following Step 1’ (see Remark 2(a)).

**Step 1’ (Approximation with a structured low-rankness):** approximate $A$ with a certain $\tilde{A} \in \Xi^{-1}(\Xi(\bigotimes_{k=1}^K C^{n \times n}) \cap \mathcal{E}_{n-n})$, where $\tilde{A} = A$ must be employed if $A \in T_{n,K}$.

Proposition 1 suggests that Step 1’ can be decomposed further into the following Step 1’a and Step 1’b.

**Problem 2** (Two steps for ATDS algorithm with a Structured Low-Rank Approximation: SLRA). For a given $A \in \bigotimes_{k=1}^K C^{n \times n}$,

Step 1’a (A structured low-rank approximation):
approximate $\Xi(A)$ with a certain $\hat{A}_s \in \Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n}) \cap \mathbb{Q}_{n^2-n^2}$ where $\hat{A}_s = \Xi(A)$ must be employed if $\Xi(A) \in \Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n}) \cap \mathbb{Q}_{n^2-n^2}$.

Step 1'b (Projection onto an affine subspace):
compute $\hat{A} = P_{\Xi^{-1}(\hat{A}_s)}(A)$ (see Proposition 1).

Step 2 (Simultaneous diagonalization): find a common exact diagonalizer of $\hat{A} = (A_1, \ldots, A_K)$, i.e., find $\hat{S} \in \mathbb{C}^{n \times n}$ s.t. $\hat{S}^{-1} \hat{A}_k \hat{S}$ ($k = 1, \ldots, K$) are diagonal.

Remark 3 (On the proposed approach).

(a) Proposition 1(c) ensures that $\hat{A} = (\hat{A}_1, \ldots, \hat{A}_K) := P_{\Xi^{-1}(\hat{A}_s)}(A)$ is expected to be close to $A$ if $\hat{A}_s$ is obtained in Step 1'a as a good approximation of $\Xi(A)$. From Theorem 1(e), moreover, $\hat{A}$ is guaranteed to be simultaneously diagonalizable if $\hat{A}$ has $n$ distinct eigenvalues for any $l \in \{1, \ldots, K\}$.

(b) The proposed approach exploits an algebraic property, i.e., $\mathbb{D}_{n,K}^{abl} = \mathbb{C}_{n,K} \cap (\mathbb{D}^{abl})^K$, of a simultaneously diagonalizable tuple. Using this algebraic property aims to achieve a denoising effect in Step 1'. The effectiveness, of using the commutativity condition, for denoising in ASD was suggested in [11] but only for $K = 2$.

3.2 Approximate Simultaneous Diagonalization Algorithm by Cadzow’s Algorithm

We have already shown how to solve Step 1'b in Proposition 1(b)5. To realize Step 1'a, we propose to use Cadzow’s algorithm [18] also known as alternating projection algorithm, below:

\[
\begin{align*}
\hat{A}(0) &:= \Xi(A); \\
\hat{A}(t+1) &:= P_{\Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n})} \circ P_{\mathbb{Q}_{n^2-n^2}}(\hat{A}(t))
\end{align*}
\]  \hspace{1cm} (6)

($t = 0, 1, \ldots$), where $\circ$ denotes a composition of mappings and, for any $\hat{X} \in \mathbb{C}^{K \times n \times n}$, we define

\[
\begin{align*}
P_{\mathbb{Q}_{n^2-n^2}}(\hat{X}) &:= \underset{\hat{Y} \in \mathbb{Q}_{n^2-n^2}}{\text{argmin}} \|\hat{X} - \hat{Y}\|_F; \\
P_{\Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n})}(\hat{X}) &:= \underset{\hat{Y} \in \Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n})}{\text{argmin}} \|\hat{X} - \hat{Y}\|_F.
\end{align*}
\]  \hspace{1cm} (7)

Proposition 2 (Monotone approximation property of alternating projection). Let $\{\hat{A}(t)\}_{t=0}^{\infty}$ be the sequence generated by (6). Then, the sequence $\{\hat{A}(t)\}_{t=0}^{\infty}$ defined by

\[
\hat{A}(t) := P_{\Xi^{-1}(\hat{A}(t))}(A) \in \times_{k=1}^{K} \mathbb{C}^{n \times n} \text{ satisfies } \Xi(\hat{A}(t)) = \hat{A}(t) \in \Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n}) \ (t = 0, 1, \ldots) \text{ and }
\]

\[
\|\Xi(\hat{A}(t+1)) - P_{\mathbb{Q}_{n^2-n^2}}(\Xi(\hat{A}(t+1)))\|_F \leq \|\Xi(\hat{A}(t)) - P_{\mathbb{Q}_{n^2-n^2}}(\Xi(\hat{A}(t)))\|_F \ (t = 0, 1, \ldots).
\]

(The proof of Proposition 2 is given in Appendix F). Finally, we propose the ATDS algorithm with SLRA (Algorithm 2), as a practical solution to Problem 3, where we also propose additionally to use “Pseudo Common Diagonalizer (PCD)” (see function PCD in Algorithm 2 under Assumption 1 for $P_{\Xi^{-1}(\hat{A}_{tot})}(A)$ (see Remark 2(c)) if $A \in \mathbb{D}_{n,K}^{abl}$ is not achieved by Step 1'b. Note that, if $A \in \mathbb{D}_{n,K}^{abl}$, Algorithm 2 guarantees to satisfy $\hat{A} = A$ (a requirement in Step 1 of Problem 2) and to find the common exact diagonalizer of $A$ with the DODO method (see Algorithm 1) unlike the Jacobi-like methods (see Example 1 in Section 2.2).

Proposition 3 (On Algorithm 2 in the case of $A \in \mathbb{D}_{n,K}^{abl}$). Suppose that $A = (A_1, \ldots, A_K) \in \times_{k=1}^{K} \mathbb{C}^{n \times n}$ happens to satisfy $\hat{A} \in \mathbb{D}_{n,K}^{abl}$. Then, Algorithm 2 finds a common exact diagonalizer $\hat{S} \in \mathbb{C}^{n \times n}$ s.t. $\hat{S}^{-1} \hat{A}_k \hat{S}$ ($k = 1, \ldots, K$) are diagonal.

Proof. Recall that an initial guess $\hat{A}(0) = \Xi(A)$ with $A \in \mathbb{D}_{n,K}^{abl}$ satisfies $\hat{A}(0) \in \Xi(\times_{k=1}^{K} \mathbb{C}^{n \times n}) \cap \mathbb{Q}_{n^2-n^2}$ (see Remark 2(a)), which implies that $\|\hat{A}(0) - P_{\mathbb{Q}_{n^2-n^2}}(\hat{A}(0))\|_F = \|O\|_F = 0$. Therefore, we get $\hat{A}_s := \hat{A}(0) = \Xi(A)$ by Cadzow’s step and then $\hat{A} := P_{\Xi^{-1}(\hat{A}_s)}(A) = A \in \mathbb{D}_{n,K}^{abl}$. Consequently, $\hat{S} \in \mathbb{C}^{n \times n}$ found by the DODO method makes $\hat{S}^{-1} \hat{A}_k \hat{S}$ ($k = 1, \ldots, K$) diagonal. \qed

Remark 4 (On Algorithm 2).

(a) The initial guess $\hat{A}(0) = \Xi(A)$ can be computed without multiplications because each $n$-by-$n$ block $\hat{A}_{i,j} \in \mathbb{C}^{n \times n}$ ($i, j = 1, \ldots, n$) of $I_n \otimes A_k - A_k^i \otimes I_n$ ($k = 1, \ldots, K$) is given by $\hat{A}_{i,j} = A_k - a_{i,k}^j I_n$ if $i = j$; otherwise $\hat{A}_{i,j} = -a_{j,k}^i I_n$, where $A_k := [a_{i,k}^j]_{i,j=1}$. 

(b) The projection $P_{\mathbb{Q}_{n^2-n^2}}(\hat{X})$ in (7) can be computed with the truncated Singular Value Decomposition (SVD) of $\hat{X}$ (see the Schmidt approximation theorem in, e.g., [23] Theorem 3). The SVD of $\hat{A}(t) \in \mathbb{C}^{K \times n \times n}$ ($t = 0, 1, \ldots$) is certainly dominant in the computation time of Algorithm 2 although efficient SVD algorithms for a large matrix have been studied extensively (see [24] and the references therein). We also remark that $P_{\mathbb{Q}_{n^2-n^2}}(\hat{X})$ in (7) is determined uniquely except in a very special case.
Algorithm 2 ATDS algorithm with SLRA

Input: \( A \in X_n^{K_1 \times \mathbb{C}^{n \times n}} \) (Set \( \eta > 0 \) and \( t_{\text{max}} \in \mathbb{N} \) for stopping condition.)

Output: a common exact diagonalizer \( \hat{S} \in \mathbb{C}^{n \times n} \) of \((A \ast \hat{A}) \in \mathcal{T}_{n,K}^\text{ab}

```latex
\begin{align*}
\hat{A}(0) &= \mathbb{E}(A); t = 0 \\
\text{while} & \left[ \left| \hat{A}(t) - \mathbb{P}_{n,K}^{\text{ab}}(\hat{A}(t)) \right| \right] \geq \eta \text{ and } t < t_{\text{max}} \text{ do} \\
\hat{A}(t + 1) &= \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}} \circ \mathbb{P}_{n,K}^{\text{ab}}(\hat{A}(t)) \\
t &= t + 1 \\
\text{end while}
\end{align*}
```

if \( \mathbb{P}_{n,K}^{\text{ab}}(\hat{A}(t)) \in \mathcal{T}_{n,K}^\text{ab} \) then \( \hat{A} = \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}} \circ \mathbb{P}_{n,K}^{\text{ab}}(\hat{A}(t)) \) \( \Rightarrow \) can be judged with Fact 2
else
\( \hat{S} = \mathcal{DODO}(\hat{A}, n) \)
\( \Rightarrow \) see Algorithm 1 for DODO.
\( \hat{S} = \mathcal{PCD}(\mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\hat{A}(t))) \)
\( \Rightarrow \) see below for PCD.
\( \hat{A} = \mathbb{P}_{n,K}^{\text{ab}}(\hat{S}) \).
end if

Assumption 1: at least one of \( X_k \in \mathbb{C}^{n \times n} \) \((k = 1, \ldots, K)\) is diagonalizable.

function PCD(\( X_1, \ldots, X_K \)) \( \Rightarrow \) PCD: “Pseudo Common Diagonalizer”
\( \mathfrak{D} \coloneqq \{ k \in \{1, \ldots, K\} \mid X_k \text{ is diagonalizable.} \}

Compute \( \hat{S}_k \in \mathbb{C}^{n \times n} \) \((k \in \mathfrak{D})\) s.t. \( \hat{S}_k X_k \hat{S}_k \text{ is diagonal.} \)

Choose \( \eta \in \arg\min_{\eta} \| A - \mathbb{P}_{n,K}^{\text{ab}}(\hat{S}) \|_F \).

return \( \hat{S}_k \in \mathbb{C}^{n \times n} \)

end function

The projection \( \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\{\hat{X}_k^T, \ldots, \hat{X}_K^T\}^T) \coloneqq [\hat{Z}_1^T, \ldots, \hat{Z}_K^T]^T \) \((\hat{X}_k, \hat{Z}_k \in \mathbb{C}^{n \times n}; k = 1, \ldots, K)\) in (7) can be computed by assigning, to \( \hat{Z}_k \), the orthogonal projection of \( \hat{X}_k \) onto \( M \coloneqq \text{span}(\hat{E}(i,j)) \((i,j) = \{i,j \mid (i,j) \in \{1, \ldots, n\}^2 \setminus \{(n,n)\}\)\), where \( \hat{E}(i,j) = e_{[i,j]^T} \in \mathbb{C}^{n \times n} \) is given by \( e_{[i,j]^T} = 1 \) if \( (p,q) = (i,j) \); otherwise 0.
Moreover, the orthogonal projections \( \mathbb{P}_M(\hat{X}_k) \) can be computed efficiently by using the sparsity of \( \hat{E}(i,j) \((i,j) = 1, \ldots, n) \) and \( \mathbb{P}_M(\hat{X}_k) = \mathbb{P}(\hat{E}(i,i) || \hat{X}_k) + \sum_{1 \leq s \neq t \leq n} \mathbb{P}(\hat{E}(s,t)) || \hat{X}_k) \).

(d) Alternating projection algorithm used in Step 1a is a powerful tool to solve feasibility problems. Even for nonconvex feasibility problems, the algorithm has a guarantee to converge locally \([25,26]\) to a point in the intersection and has been used extensively for finding a point, near the initial guess, in the intersection, e.g., phase retrieval \([27]\).

(e) Suppose \( S \in \mathbb{C}^{n \times n} \) is invertible. Since \( \mathbb{D}_{n,K}^{\text{ab}}(S) \coloneqq \{X \in \mathbb{D}_{n,K}^{\text{ab}} \mid X \text{ has a common diagonalizer} S\} \) is a subspace of \( X_n^{K_1 \times \mathbb{C}^{n \times n}} \), the orthogonal projection of \( A \) onto \( \mathbb{D}_{n,K}^{\text{ab}}(S) \), i.e., \( \mathbb{P}_{n,K}^{\text{ab}}(S)(A) \coloneqq (\hat{Z}_1, \ldots, \hat{Z}_K) \in \mathbb{D}_{n,K}^{\text{ab}}(S) \), is well-defined and can be computed as

\( Z_k \coloneqq \mathbb{E}^{-1}(\mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\text{vec}(\text{vec}(A_k)))) \( (k = 1, \ldots, K) \),

where \( \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\text{vec}(A_k)) \) are the orthogonal projections onto \( \text{range}(S^{-1} \odot S) \subset \mathbb{C}^{n \times n} \). Even for the cases of \( \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\text{vec}(A_k)) \in \mathbb{D}_{n,K}^{\text{ab}} \), in Algorithm 2, we can obtain \( \hat{A} = \mathbb{P}_{n,K}^{\text{ab}}(S)(\hat{S}) \in \mathbb{D}_{n,K}^{\text{ab}} \) via the function PCD, if at least one matrix in the tuple \( \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\text{vec}(A_k)) \) is diagonalizable.

4. Numerical Experiments

To see the numerical performance of Algorithm 2 in comparison to the two Jacobi-like methods (sh-rt \([13]\) and JDTM \([5]\)), under several conditions (e.g., noise levels, condition numbers of an ideal common diagonalizer), we conduct numerical experiments for a perturbed version \( A = (A_1, \ldots, A_K) \in X_n^{K_1 \times \mathbb{C}^{n \times n}} \) \((n = 5, K = 20) \) of \( \mathcal{A}^* = (A_1^*, \ldots, A_K^*) \coloneqq (S^* X^*_k (S^*)^{-1})^T_k \) \((k = 1, \ldots, K) \), with \( A_k \coloneqq A_k^* + \sigma N_k \( (k = 1, \ldots, K) \), where the diagonal entries of diagonal matrices \( A_k^* \in \mathbb{C}^{n \times n} \) and all the entries of \( N_k \in \mathbb{C}^{n \times n} \) are drawn from the standard normal distribution \( \mathcal{N}(0,1) \) and \( \sigma > 0 \) is used to define the Signal to Noise Ratio (SNR). To conduct numerical experiments for \( S^* \in \mathbb{C}^{n \times n} \) of various condition numbers, such as, we design \( \mathcal{A}^* \) by replacing singular values, of a matrix whose entries are drawn from the standard normal distribution \( \mathcal{N}(0,1) \), with \( \sigma_1(S^*) = (k - 1)(n-i)/{(n-1)} + 1 + i = 1, \ldots, n \) implying thus \( \sigma_1(S^*)/\sigma_n(S^*) = \kappa \). For Algorithm 2, we use, as the rth estimates, \( \hat{S}(t) = \mathcal{DODO}(\mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\hat{A}(t))) \( A \)(n) \) if \( \mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\hat{A}(t))) \( A \)(n) \) \( \text{otherwise} \) \( \hat{S}(t) = \mathcal{PCD}(\mathbb{P}_{X_n^{K_1 \times \mathbb{C}^{n \times n}}}(\hat{A}(t))) \) (see function PCD in Algorithm 2).

\footnote{This is verified by the well-known identity vec(\text{Adiag}(d)C) = (C^T \odot A)\text{vec}(A) \in \mathbb{C}^{n \times n} \text{ (see, e.g., [28]) which is also used in the proof in Appendix D.}}
We conducted numerical experiments on Intel Core i7-40 refresh, $\text{Algorithm 2} \text{JDTM}$.\cite{5} and (ii) $\|S^* - S_{\text{ILH}}(t)\|_F^2 / \|S^*\|_F^2 \leq 10^{-2}$, $10^{-3}$, or $10^{-4}$, and rate of successful trials (from left to right) in the case of SNR 50 dB and $\kappa = 50$. Note that we use "." if the rate of successful trials is smaller than 10%.

| Algorithm | Iteration | SNR [dB] | Rate (%) |
|-----------|-----------|----------|----------|
| sh-rt (4) | 4.208 ms  | 52.69    | 75%      |
| JDTM \cite{5} | 1.552 ms  | 18.85    | 100%     |
| Algorithm 2 | 4.004 ms  | 1.610    | 100%     |

Fig. 2 Comparison of ASD algorithms for $S^*$ of condition number $\kappa = 50$.

Table 3 Computation time, number of iterations, until $\|S^* - S_{\text{ILH}}(t)\|_F^2 / \|S^*\|_F^2 \leq 10^{-2}$, $10^{-3}$, or $10^{-4}$, and rate of successful trials (from left to right) in the case of SNR 50 dB and $\kappa = 5$.

| Algorithm | Iteration | SNR [dB] | Rate (%) |
|-----------|-----------|----------|----------|
| sh-rt (4) | 2.139 ms  | 21.27    | 100%     |
| JDTM \cite{5} | 1.277 ms  | 14.09    | 100%     |
| Algorithm 2 | 3.208 ms  | 1.000    | 100%     |

stands for the column-wise permutation applied to achieve the best approximation to $S^*$ after the column-wise normalization of $S(t)$, and (ii) $\tilde{\Lambda}_I(t)$ stands for the diagonal matrix after applying the corresponding permutation for $S_{\text{ILH}}(t)$ to diagonal entries of $\tilde{\Lambda}_I(t)$.

We conducted numerical experiments on Intel Core i7-8559U running at 2.7 GHz with 4 cores and 32 GB of main memory. By using Matlab, we implemented all the ASD algorithms by ourselves. We measured the computation times of all the ASD algorithms by Matlab’s tic/toc functions. We compared the computation time and the number of iterations, of all the algorithms, until $\|S^* - S_{\text{ILH}}(t)\|_F^2 / \|S^*\|_F^2 \leq 10^{-2}$, $10^{-3}$, and $10^{-4}$ ($t = 0, 1, \ldots$) as shown in Table 2 and 3.

Fig. 1 Transition of approximation errors in the case of SNR 50 dB and $\kappa = 50$.

Since it is reported in $\cite{5}$ that the Jacobi-like methods tend to suffer from the cases where the input tuple is a perturbed version of a simultaneously diagonalizable tuple with a common diagonalizer of large condition number,
we first compared all the ASD algorithms for \( \kappa = 50 \). Figure 1 depicts the transition of the mean values, of the relative squared errors of \( \hat{A}(t), \tilde{S}(t) \) and \( \tilde{(\Lambda_1(t), \ldots, \Lambda_k(t))} \) after proper column-wise normalization/permutation for \( S(t) \) and \( \tilde{(\Lambda_1(t))}^K_{k=1} \) over 100 trials in the case of SNR 50 dB. Figure 1 illustrates that, compared with the Jacobi-like methods, Algorithm 2 takes around 5\% of the computation time and 1\% of the numbers of iterations for successful trials over 100 trials. This result shows that Algorithm 2 takes around 3 times longer computation time than JDTM but its \( \tilde{S}(t_{\text{end}}) \) achieves the prescribed conditions even for the trials where the Jacobi-like methods fail to achieve the prescribed conditions. Figure 2 depicts the mean values, of the relative squared errors of \( \hat{A}(t_{\text{end}}), \tilde{S}(t_{\text{end}}), \) and \( \tilde{(\Lambda_k(t_{\text{end}}))}^K_{k=1} \) after proper column-wise normalization/permutation for \( S(t) \) and \( \tilde{(\Lambda_k(t))}^K_{k=1} \) over 100 trials in the cases of SNR from 0 dB to 50 dB. Figure 2 illustrates Algorithm 2 outperforms the Jacobi-like methods in the sense of achieving approximation errors at \( t_{\text{end}} \) especially when SNR is higher than 10 dB.

We also compared all the algorithms in the case of \( \kappa = 5 \). All the values in Table 3 and Figure 3 are calculated by the same way as for \( \kappa = 50 \). Figure 3 and Table 3 show that Algorithm 2 takes around 3 times longer computation time than JDTM but can outperform the Jacobi-like methods in the sense of achieving approximation errors at \( t_{\text{end}} \) in SNR from 5 dB to 50 dB. From these experiments, we see that Algorithm 2 is robust against wider range of condition numbers of \( S^* \) than the Jacobi-like methods.

5. Concluding Remarks

In this paper, for the approximate simultaneous diagonalization of \( A \in \mathbb{C}^{K \times n \times n} \), we newly presented the Approximate-Then-Diagonalize-Simultaneously (ATDS) algorithm by solving a certain Structured Low-Rank Approximation (SLRA). For \( A \in \mathbb{T}^{n \times K} \), the proposed ATDS algorithm has a guarantee to find its common exact diagonalizer unlike the Jacobi-like methods. Numerical experiments show that, at the expense of reasonable computational time, the proposed ATDS algorithm achieves better approximations to the desired information in Problem 2 than the Jacobi-like methods for almost all SNR as well as both for small and large condition numbers of \( S^* \).

The reduction of the computational cost for SVD of \( \hat{A}(t) \in \mathbb{C}^{K \times n^2 \times n^2} \) (\( t = 0, 1, \ldots \)) in Algorithm 2 is under study. We are also studying applications of the proposed ATDS algorithm to certain signal processing problems. These will be reported elsewhere.

References

[1] I.T. Jolliffe, Principal Component Analysis, 2nd ed., Springer-Verlag, 2002.
[2] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[3] L. De Lathauwer, B. De Moor, and J. Vandewalle, “Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition,” SIAM J. Matrix Anal. Appl., vol.26, no.2, pp.295–327, 2004.
[4] F. Roemer and M. Haardt, “A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (עבד),” Signal Processing, vol.93, no.9, pp.2722–2738, 2013.
[5] X. Luciani and L. Albera, “Canonical Polyadic Decomposition based on joint eigenvalue decomposition,” Chemom. Intell. Lab. Syst., vol.132, pp.152–167, 2014.
[6] R. André, X. Luciani, and E. Moreau, “Joint EigenValue Decomposition Algorithms Based on First-Order Taylor Expansion,” IEEE Trans. Signal Process., vol.68, pp.1716–1727, 2020.
[7] J.F. Cardoso and A. Souloumiac, “Blind beamforming for non-Gaussian signals,” IEE Proc. F Radar Signal Process., vol.140, no.6, pp.362–370, 1993.
[8] L. Albera, A. Ferréol, P. Chevalier, and P. Comon, “ICAR: A Tool for Blind Source Separation Using Fourth-Order Statistics Only,” IEEE Trans. Signal Process., vol.53, no.10, pp.3633–3643, 2005.
[9] L. De Lathauwer and J. Casteing, “Blind identification of underdetermined mixtures by simultaneous matrix diagonalization,” IEEE Trans. Signal Process., vol.56, no.3, pp.1096–1105, 2008.
[10] X. Luciani and L. Albera, “Joint Eigenvalue Decomposition of Non-Defective Matrices Based on the LU Factorization With Application to ICA,” IEEE Trans. Signal Process., vol.63, no.17, pp.4594–4608, 2015.
[11] A.J. van der Veen, P.B. Ober, and E.F. Deprettere, “Azimuth and
Appendix A: Nonconvexity of $\mathcal{D}^{abl}_{2,2}$

The following simple example shows the nonconvexity of $\mathcal{D}^{abl}_{2,2}$. Let $A_1 := \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $A_2 := 2A_1$, $B_1 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and $B_2 := 2B_1$. Since $A_1, A_2, B_1$ and $B_2$ are normal, these are respectively diagonalizable [20, Theorem 2.5.3]. Moreover, by $A_1 A_2 = A_2 A_1$ and $B_1 B_2 = B_2 B_1$, we see $A := (A_1, A_2), B := (B_1, B_2) \in \mathcal{D}^{abl}_{2,2}$ (see Fact 2 in Section 2.1).

Below, we will show that $A/2 + B/2 \notin \mathcal{D}^{abl}_{2,2}$. Since every eigenvector of $(A_1 + B_1)/2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is given by $v = [0 \in \mathbb{C} \setminus \{0\}]$, the eigenspace of $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ has dimension 1, which implies $(A_1 + B_1)/2$ is not diagonalizable and hence $A/2 + B/2 \notin \mathcal{D}^{abl}_{2,2}$.

Appendix B: Useful Facts on Commutativity and Diagonizability

Fact 3 (On commutativity).

(a) For a given $X \in \mathbb{C}^{n \times n}$, the set of all matrices which commute with $X$ is a subspace of $\mathbb{C}^{n \times n}$ with dimension at least $n$; the dimension is equal to $n$ if and only if each eigenvalue of $X$ has geometric multiplicity 1, i.e., $X$ is nonderogatory [30, Corollary 4.4.15].

(b) For a given $X \in \mathbb{C}^{n \times n}$, $X$ is nonderogatory if and only if every matrix $Y \in \mathbb{C}^{n \times n}$ which commutes with $X$ can be expressed as $Y = c_0 I + c_1 X + c_2 X^2 + \cdots + c_{n-1} X^{n-1}$ for some $c_0, c_1, \ldots, c_{n-1} \in \mathbb{C}$, i.e., $Y$ is a polynomial in $X$ [30, Corollary 4.4.18].

(c) Let $\lambda_1, \ldots, \lambda_d \in \mathbb{C}$ be distinct and let $\Lambda = \lambda_1 I_{n,1} + \cdots + \lambda_d I_{n,d} \in \mathbb{C}^{n \times n}$. Then, for $X = [X_{i,j}]_{d \times d} \in \mathbb{C}^{n \times n}$, the operator $\Lambda X = X \Lambda \Longleftrightarrow (\forall i, j \in \{1, \ldots, d\}) \lambda_i X_{i,j} = \lambda_j X_{i,j}$, and $(\forall i \neq j) \lambda_i X_{i,j} = \lambda_j X_{i,j} \Leftrightarrow (\forall i \neq j) X_{i,j} = 0$.

Fact 4 (On diagonalizability of block diagonal matrix).

Suppose $X = X_1 \oplus \cdots \oplus X_l \in \mathbb{C}^{n \times n}$ for some $X_p \in \mathbb{C}^{n_p \times n_p}$ ($n_1 + \cdots + n_l = n$). Then $X \in \mathcal{D}^{abl}_{n} \Leftrightarrow (\forall p \in \{1, \ldots, l\}) X_p \in \mathcal{D}^{abl}_{n_p}$.

Appendix C: Structured Low-Rank Approximation

Structured Low-Rank Approximation (SLRA) [17] is a problem, for a given matrix $A \in \mathbb{C}^{m \times n}$, a given integer $r \in (0, \min(m, n))$, and a given affine subspace $\Omega \subset \mathbb{C}^{m \times n}$, to find a minimizer, in all matrices of rank at most $r$, of $\|A - X\|_F^2$. SLRA has many applications in signal processing (see e.g., [17] and references therein). For SLRA, Cadzow’s algorithm [18] also known as alternating projection algorithm has been used extensively.
while some methods to find its local minimizer \[31,32\] or its global one \[33\] also have been proposed. It is reported that, in practice, Cadzow’s algorithm finds a structured low-rank matrix close to a given one \[19\].

### Appendix D: Proof of Theorem 1

(a) This follows from the expression of the condition \(X_lX_l^T - X_lX_l = 0\) \((k, l = 1, \ldots, K)\) in vector form.

(b) (Proof of "⇒") Let \(X_k = SL_kS^{-1}\) be such that \(L_k \in \mathbb{C}^{n \times n}\) \((k = 1, \ldots, K)\) are diagonal. By using identities: \((A \otimes B)(C \otimes D) = AC \otimes BD\) \((A, B, C, D \in \mathbb{C}^{n \times n})\) \[30\], Lemma 4.2.10 and \((A \otimes B)^{-1} = A^{-1} \otimes B^{-1}\) \((A, B \in \mathbb{C}^{n \times n})\) \[30\], Corollary 4.2.11, we get \(I_n \otimes X_k - X_k^T \otimes I_n = (S^{-T} \otimes S)(I_n \otimes L_k - L_k \otimes I_n)(S^{-T} \otimes S)^{-1}\). Since \((I_n \otimes L_k - L_k \otimes I_n)\) is diagonal and the \((i - j)m + \jth diagonal entry is 0 for every \(j = 1, \ldots, n\), we see \((∀K \in \{1, \ldots, K\})\) nullspace\((I_n \otimes X_k - X_k^T \otimes I_n) \supseteq \text{range}(S^{-T} \otimes S) \Leftarrow \text{nullspace}(\Xi(X^T)) \supseteq \text{range}(S^{-T} \otimes S).

(Proof of "⇐") Note first that any \(y \in \text{range}(S^{-T} \otimes S)\) can be expressed with \(λ := [λ_i] \in \mathbb{C}^n\) as \(y = (S^{-T} \otimes S)λ\) and that \(Y := \text{vec}^{-1}(y) = \text{Sdiag}(λ_1, \ldots, λ_n)S^{-1}\), where we used the well-known identity \(\text{vec}(\text{Sdiag}(λ)) = (C^T \otimes A)λ\) \((A, C \in \mathbb{C}^{n \times n}, \text{vec} \in \mathbb{C}^n\) (see, e.g., \[28\], Property 1). By \(y \in \text{nullspace}(\Xi(X^T))\) and (a), we see \(X_kY = X_k\) \((k = 1, \ldots, K)\). Choose specifically \(λ\) such that all entries \(λ_i\) \((i = 1, \ldots, n)\) are distinct in order to ensure \(Y\) is nonderogatory. In this case, Fact \[4,10\] ensures that \(X_k\) \((k = 1, \ldots, K)\) are polynomials in \(Y\), say \(X_k = P_k(Y)\), and therefore, \(X_k = \text{Sdiag}(P_k(λ_1), \ldots, P_k(λ_n))S^{-1}\), which implies \(X \in \mathcal{A}_{n,K}\).

(c) Let \(X_k = SL_kS^{-1}\) be such that \(L_k \in \mathbb{C}^{n \times n}\) \((k = 1, \ldots, K)\) are diagonal. Since \(\text{dim}(\text{range}(S^{-T} \otimes S)) = n\), we see from (b) that \(\text{dim}(\text{nullspace}(\Xi(X^T))) \geq n \Leftrightarrow \text{rank}(\Xi(X)) \leq n^2 - n\). Moreover, \(\Xi(X) = (I_k \otimes (S^{-T} \otimes S))(\text{diag}(L_1, L_2, \ldots, L_K)(S^{-T} \otimes S)^{-1})\), we see \(\text{rank}(\Xi(X)) = \text{rank}(\Xi(L_1, \ldots, L_K))\). Fact \[3\] ensures the simultaneous diagonalization of \(X\) is essentially unique if and only if \(\text{rank}(\Xi(A))\) has just zero vectors as the column vectors, i.e., \(\text{rank}(\Xi(L_1, \ldots, L_K)) = n^2 - n\).

(d) (Proof of "⇒") It follows from Fact \[4,10\] and (c).

(Proof of "⇐") Suppose that \(X_1\) has \(n\) distinct eigenvalues. Therefore, there exist \(S \in \mathbb{C}^{n \times n}\) and a diagonal matrix \(Λ_1 \in \mathbb{C}^{n \times n}\) such that \(X_1 = SL_1S^{-1}\). By Fact \[4,10\] \(\text{dim}(\text{nullspace}(I_n \otimes X_1 - X_1^T \otimes I_n)) = n\) and hence \(\text{nullspace}(\Xi(X)) = \text{nullspace}(I_n \otimes X_1 - X_1^T \otimes I_n)\). For any \(k = 2, \ldots, K\), moreover, \(\text{nullspace}(I_n \otimes X_k - X_k^T \otimes I_n) \supseteq \text{nullspace}(\Xi(X)) = \text{nullspace}(I_n \otimes X_1 - X_1^T \otimes I_n) \supseteq \text{nullspace}(X_1), \) which ensures the commutativity of \(X_1\) and \(X_k\). By using Fact \[4,10\] we see that each \(X_k\) is a certain polynomial in \(X_1\). Therefore, \(S^{-1}X_kS\) \((k = 2, \ldots, K)\) are diagonal.

(e) From (d), it is sufficient to show \(\text{rank}(\Xi(Y)) = n^2 - n\). Since \(Y_1\) has \(n\) distinct eigenvalues, Fact \[4,10\] and \(\Xi(Y) \in \mathcal{V}_{n^2 - n}\) ensure \(n^2 - n = \text{rank}(I_n \otimes Y_1 - Y_1^T \otimes I_n) \leq \text{rank}(\Xi(Y)) \leq n^2 - n\).

### Appendix E: Proof of Proposition 1

(a) It is clear that \(X \in (\text{span}(I_n))^K \equiv \Xi(X) = O\).

(Proof of "⇒") Since, for any \(X \in X^o + (\text{span}(I_n))^K\), \(\Xi(X) = \Xi(X^o) = \hat{X}\), we hold, see \(X \in \Xi^{-1}(\hat{X})\).

(Proof of "⇐") For any \(X \in \Xi^{-1}(\hat{X})\), we have \(\Xi(X - X^o) = O \iff X - X^o \in (\text{span}(I_n))^K\), where \(X^o \in \Xi^{-1}(\hat{X})\). Therefore \(X \in X^o + (\text{span}(I_n))^K\).

(b) This follows from (a) and equalities

\[
\min_{(Y_1, \ldots, Y_K) \in \Xi^{-1}(\hat{X})} \sum_{k=1}^K \|X_k - Y_k - \hat{X_k}I_n\|^2_F = \min_{\alpha_1, \ldots, \alpha_K \in \mathbb{C}} \sum_{k=1}^K \|X_k - \hat{X_k} - \alpha_k I_n\|^2_F
\]

\[= \sum_{k=1}^K \|X_k - \hat{X_k} - P_{\text{span}(I_n)}(X_k - \hat{X_k})\|^2_F.
\]

(c) For \(X = [x_{i,j}] \in \mathbb{C}^{n \times n}\) and \(Y = [y_{i,j}] \in \mathbb{C}^{n \times n}\),

\[
\|I_n \otimes X - Y \otimes I_n - I_n \otimes Y - Y^T \otimes I_n\|^2_F
\]

\[= \sum_{i=1}^n \|X_{i,i} - Y_{i,i} - Y_{i,i}I_n\|^2_F
\]

\[+ n \sum_{p,q=1,p \neq q}^n |x_{p,q} - y_{p,q}|^2
\]

\[= n\|X - Y\|^2_F + n \sum_{i=1}^n |x_{i,i} - y_{i,i}|^2
\]

\[= 2\|X - Y\|^2_F + 2\|X - Y\|^2_F.
\]

Hence \(\Xi(X) - \hat{X}\|^2_F = \|\Xi(X) - \Xi(Z_1, \ldots, Z_K)\|^2_F = \sum_{k=1}^K 2\|X_k - Z_k\|^2_F - 2\|X_k - Z_k\|^2_F.\) Since (b) ensures \(\text{tr}(X_k - Z_k) = 0\) \((k = 1, \ldots, K)\), we get \(\sum_{k=1}^K \|X_k - Z_k\|^2_F = \|\Xi(X) - \hat{X}\|^2_F / 2n\).

### Appendix F: Proof of Proposition 2

This follows from \(\|\Xi(\hat{A}(t + 1)) - P_{g_{n^2 - n}}(\Xi(\hat{A}(t)))\|^2_F \geq \|\Xi(\hat{A}(t)) - P_{g_{n^2 - n}}(\Xi(\hat{A}(t)))\|^2_F\).
\[ \| \Xi(\tilde{A}(t+1)) - P_{u \times n} (\Xi(\tilde{A}(t+1))) \|_F \geq \tau. \]

\[ \begin{align*}
Riku Akema & \quad \text{received the B.E. degree in computer science and M.E. degree in information and communications engineering from the Tokyo Institute of Technology in 2016 and 2018, respectively. Currently, he is a Ph.D. student in the Department of Information and Communications Engineering, Tokyo Institute of Technology. His current research interests are in signal processing and multi-way data analytics.} \\
Masao Yamagishi & \quad \text{received the B.E. degree in computer science and the M.E. and Ph.D. degrees in communications and integrated systems from the Tokyo Institute of Technology, Tokyo, Japan, in 2007, 2008, and 2012, respectively. Currently, he is an assistant professor with the Department of Information and Communications Engineering, Tokyo Institute of Technology.} \\
Isao Yamada & \quad \text{received the B.E. degree in computer science from the University of Tsukuba in 1985 and the M.E. and the Ph.D. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, in 1987 and 1990. Currently, he is a professor with the Department of Information and Communications Engineering, Tokyo Institute of Technology. He has been the IEICE Fellow and IEEE Fellow since 2015.} 
\end{align*} \]