A RECURSIVE EIGENSPACE COMPUTATION FOR THE CANONICAL POLYADIC DECOMPOSITION

ERIC EVERT†, MICHEIL VANDECAPPELLE†, AND LIEVEN DE LATHAUWER†

Abstract. The canonical polyadic decomposition (CPD) is a compact decomposition which expresses a tensor as a sum of its rank-1 components. A common step in the computation of a CPD is computing a generalized eigenvalue decomposition (GEVD) of the tensor. A GEVD provides an algebraic approximation of the CPD which can then be used as an initialization in optimization routines.

While in the noiseless setting GEVD exactly recovers the CPD, it has recently been shown that pencil-based computations such as GEVD are not stable. In this article we present an algebraic method for approximation of a CPD which greatly improves on the accuracy of GEVD. Our method is still fundamentally pencil-based; however, rather than using a single pencil and computing all of its generalized eigenvectors, we use many different pencils and in each pencil compute generalized eigenspaces corresponding to sufficiently well-separated generalized eigenvalues. The resulting “generalized eigenspace decomposition” is significantly more robust to noise than the classical GEVD.

Accuracy of the generalized eigenspace decomposition is examined both empirically and theoretically. In particular, we provide a deterministic perturbation theoretic bound which is predictive of error in the computed factorization.

Key words. tensors, canonical polyadic decomposition, generalized eigenvalue decomposition, multilinear algebra, multidimensional arrays

AMS subject classifications. Primary: 15A69, 15A22. Secondary: 15A42.

1. Introduction. Tensors, or multiindexed arrays, play an important role in fields such as machine learning and signal processing [4, 32]. These higher-order generalizations of matrices allow for preservation of higher-order structure present in data and low-rank decompositions of tensors allow for compression of data and recovery of underlying information [32, 4, 5]. One of the most popular decompositions for tensors is the canonical polyadic decomposition (CPD) which expresses a tensor as a sum of rank one tensors. An important feature of the CPD is that, with mild assumptions [10, 23, 33], the CPD of a low-rank tensor is unique. Furthermore, for a low-rank tensor with a unique CPD, the CPD can often be found algebraically. Such an algebraic solution can typically be obtained with limited computation time, hence is often used as an initialization for optimization based methods when the tensor is noisy.

One of the most popular algorithms for algebraic computation of a CPD is the generalized eigenvalue decomposition (GEVD) [11, 12, 31, 25]. The key idea behind the classical GEVD is that a factor matrix of a tensor may be obtained by computing the generalized eigenvectors of any subpencil of the tensor. However, it has recently been shown that pencil-based algorithms such as GEVD are not stable [1]. That
is, the condition number for computing a generalized eigenvalue decomposition of a 
subpencil can be arbitrarily larger than the condition number [2] for computing the 
CPD of the tensor. In this article we present an extension of the GEVD algorithm 
which significantly improves the accuracy of algebraic computation of the CPD. This 
increase in accuracy is obtained by using many subpencils of the tensor to compute 
the CPD. In each subpencil we only compute generalized eigenvectors and eigenspaces 
which correspond to sufficiently well-separated generalized eigenvalues.

The ability to accurately compute generalized eigenvectors of a matrix pencil 
is heavily dependent on the separation between the generalized eigenvalues and the 
generalized eigenvectors of the pencil [35, 19, 27]. In the case that the generalized 
eigenvalues and eigenvectors are well-separated, a GEVD may be accurately computed. 
However, when either a pair of generalized eigenvalues or generalized eigenvectors is 
near parallel, the accuracy of the computed generalized eigenvectors deteriorates. As 
such, the GEVD algorithm performs well when there is a subpencil for the tensor in 
which all generalized eigenvalues are well-separated; however, GEVD runs into chal-

lenges if one is unable to find a subpencil in which all generalized eigenvalues are 
well-separated.

When computing a GEVD of a rank $R$ tensor, one computes the generalized 
eigenvalues and eigenvectors of a pair of $R \times R$ matrices. The generalized eigenvalues 
of this matrix pair can be naturally interpreted as a collection of $R$ points on the 
unit circle [35], so it is not difficult to see that the generalized eigenvalues cannot be 
well-separated when the tensor rank $R$ is sufficiently large.

We propose an alternative approach in which we exploit the fact that a tensor 
has many subpencils, and that a factor matrix of the tensor may be computed by 
combining information from many different subpencils. Using the language of the 
joint generalized eigenvectors (JGE vectors) [13], the primary goal of the GEVD 
algorithm is in fact to compute the JGE vectors of the tensor.

Let $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. Suppose the tensor $T \in \mathbb{K}^{R \times R \times K}$ is a rank $R$ tensor with 
JGE vectors $\{x_r\}_{r=1}^R$. Then for every subpencil of $T$, the generalized eigenspaces are 
of the form $\text{span}(\{x_{r_1}, \ldots, x_{r_k}\})$ for some subcollection of JGE vectors of $T$. Here 
$\{r_1, \ldots, r_k\}$ is a subcollection of indices from $\{1, \ldots, R\}$ with cardinality $k$. Intuitively 
it follows that one may compute the JGE vectors of $T$ by computing generalized 
eigenspaces for many different subpencils of $T$.

An important feature of this approach is that the accuracy of computing a par-
ticular generalized eigenspace does not depend on the separation of the generalized 
eigenvalues which correspond to that generalized eigenspace. Rather the accuracy 
depends on the separation of the set of generalized eigenvalues corresonding to the 
generalized eigenspace of interest from the other generalized eigenvalues of the pencil 
as well as the separation between the generalized eigenspace itself and the other gener-

alized eigenvectors of the pencil [34]. Thus by only computing generalized eigenspaces 
which correspond to collections of generalized eigenvalues that are well-separated from 
the other generalized eigenvalues of the pencil, one source of inaccuracy for the gen-

eralized eigenvalue decomposition, i.e. inaccuracy introduced by projecting a tensor 
to a matrix pencil [1], is significantly reduced.

1.1. Notation and definitions. Let $\mathbb{K}$ denote either $\mathbb{R}$ or $\mathbb{C}$. We denote scalars, 
vectors, matrices, and tensors by lower case \(a\), bold lower case \(\mathbf{a}\), bold upper case 
\(\mathbf{A}\), and calligraphic script \(\mathcal{A}\), respectively. Additionally, fraktur font \(\mathfrak{A}\) is used 
to denote a subspace of $\mathbb{K}^I$. For a matrix $\mathbf{A} \in \mathbb{K}^{I \times J}$, we let $\mathbf{A}^T$ denote the transpose 
of $\mathbf{A}$ while $\mathbf{A}^H$ denotes the conjugate transpose of $\mathbf{A}$. If $\mathbf{A}$ is invertible, then we let
\( A^{-T} \) denote the inverse of \( A^{T} \). A tensor is a multiindexed array with entries in \( \mathbb{K} \). In this article we restrict our attention to tensors of order three. In other words, the tensors we consider are elements of \( \mathbb{K}^{I_1 \times I_2 \times I_3} \).

Given nonzero vectors \( a \in \mathbb{K}^{I_1}, b \in \mathbb{K}^{I_2}, c \in \mathbb{K}^{I_3} \), let
\[
    a \circ b \circ c \in \mathbb{K}^{I_1 \times I_2 \times I_3}
\]
denote the \( I_1 \times I_2 \times I_3 \) tensor with \( i, j, k \) entry equal to \( a_ib_jc_k \). A tensor of this form is called a rank one tensor. The minimal integer \( R \) such that \( \mathcal{M} \in \mathbb{K}^{I_1 \times I_2 \times I_3} \) may be written as
\[
    \mathcal{M} = \sum_{r=1}^{R} a_r \circ b_r \circ c_r
\]
where the vectors \( a_r, b_r, c_r \) have entries in \( \mathbb{K} \) is called the \( \mathbb{K} \)-rank of the tensor \( \mathcal{M} \), and a decomposition of this form is called a canonical polyadic decomposition (CPD) of \( \mathcal{M} \). Compactlty we write \( \mathcal{M} = [A, B, C] \) where the matrices \( A, B, C \) have \( a_r, b_r, c_r \) as their \( r \)th columns, respectively. The matrices \( A, B, C \) are called the factor matrices for \( \mathcal{M} \).

Given a tensor \( \mathcal{M} \in \mathbb{K}^{I_1 \times I_2 \times I_3} \), let \( \mathbf{M}_{[1:3,2]}^{[1:3,2]} \) denote an \( I_1 \times I_2 I_3 \) matrix with columns \( \{m(i, k, \ell)\}_{i, k, \ell} \). Here, the columns of \( \mathbf{M}_{[1:3,2]}^{[1:3,2]} \) are ordered so that the second index of \( m(i, k, \ell) \) increments faster than the third. The matrices \( \mathbf{M}_{[2:3,1]}^{[2:3,1]} \) and \( \mathbf{M}_{[3:2,1]}^{[3:2,1]} \) are defined similarly. A matrix of the form \( \mathbf{M}_{[i:k, j]}^{[i:k, j]} \) is called a mode-\( i \) unfolding of \( \mathcal{M} \).

Having defined the mode-\( i \) unfoldings of \( \mathcal{M} \), we define
\[
    R_i(\mathcal{M}) := \text{rank}(\mathbf{M}_{[i:k, j]}^{[i:k, j]}) \quad \text{for } i = 1, 2, 3.
\]
The integer triple \( (R_1(\mathcal{M}), R_2(\mathcal{M}), R_3(\mathcal{M})) \) is called the multilinear rank of \( \mathcal{M} \).

For tensors \( \mathcal{M} \) occurring in applications, \( \mathcal{M} \) is often thought of as a noisy measurement of some signal tensor of interest. That is, one can decompose \( \mathcal{M} = \mathcal{T} + \mathcal{N} \) where \( \mathcal{T} \) is the true signal tensor of interest and \( \mathcal{N} \) is measurement error. Most commonly the signal tensor \( \mathcal{T} \) has low multilinear rank. That is, one has \( R_i(\mathcal{T}) << I_i \) for each \( i \). In this case, under an appropriate unitary change of bases one has \( t_{i,j,k} = 0 \) if \( i > R_1(\mathcal{T}) \) or \( j > R_2(\mathcal{T}) \) or \( k > R_3(\mathcal{T}) \). In this choice of bases it is clear that \( \mathcal{T} \) may be compressed to a tensor of size \( R_1(\mathcal{T}) \times R_2(\mathcal{T}) \times R_3(\mathcal{T}) \) without loss of information. Such a compression is called an orthogonal compression of \( \mathcal{T} \) to its multilinear rank, and a popular choice of orthogonal compression is the multilinear singular value decomposition (MLSVD). See [7] for details.

Compression of a noisy measurement \( \mathcal{M} = \mathcal{T} + \mathcal{N} \) to the multilinear rank of \( \mathcal{T} \) is a common step in computation of a CPD which both approximates the (core) of the signal tensor \( \mathcal{T} \) and reduces computational complexity. However, such a compression has very few implications for the methods discussed in this article. For this reason we often restrict to considering tensors \( \mathcal{Y} \) with the property \( R_i(\mathcal{Y}) = I_i \) for \( i = 1, 2, 3 \). For clarity, here \( I_i \) refers to the \( i \)th dimension of the compressed tensor \( \mathcal{Y} \). Intuitively, the symbol \( \mathcal{Y} \) is used to denote a tensor which can be thought of as a compression of \( \mathcal{M} = \mathcal{T} + \mathcal{N} \) to the multilinear rank of \( \mathcal{T} \). In particular, we may write \( \mathcal{Y} = \mathcal{T} + \mathcal{N} \) where \( \mathcal{T} \) is an orthogonal compression \( \mathcal{T} \) to its multilinear rank and \( \mathcal{N} \) is noise.

It is not difficult to show that the \( \mathbb{K} \)-rank of a tensor \( \mathcal{Y} \) occurring in applications, one often has (up to a permutation of indices) \( \text{rank}_\mathbb{K}(\mathcal{T}) = R_1(\mathcal{T}) = R_2(\mathcal{T}) \geq R_3(\mathcal{T}) \). In light of this, the
tensors $\mathcal{Y} = \mathcal{T} + \mathcal{N}$ we consider are taken to have size $R \times R \times K$ where $K \leq R$ and the signal portion $\mathcal{T}$ of $\mathcal{Y}$ is assumed to have $\mathbb{K}$-rank $R$. To summarize, throughout the article we let $\mathcal{M}$ denote an arbitrary tensor of size $I_1 \times I_2 \times I_3$, we let $\mathcal{Y}$ denote a full multilinear rank tensor of size $R \times R \times K$, and we let $\mathcal{T}$ denote a $R \times R \times K$ tensor with $\mathbb{K}$-rank equal to $R$ and full multilinear rank.

Define the mode-1 product between a tensor $\mathcal{M} \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ and a matrix $\mathbf{U} \in \mathbb{K}^{J_1 \times I_2 \times I_3}$ to be the tensor $\mathcal{M} \cdot_1 \mathbf{U} \in \mathbb{K}^{J_1 \times I_2 \times I_3}$ with mode-1 fibers given by

$$ (\mathcal{M} \cdot_1 \mathbf{U})(; k, \ell) = \mathbf{U} \cdot \mathbf{m}(; k, \ell) \quad \text{for all } k, \ell. $$

The mode-2 and mode-3 products are defined analogously. Say a tensor $\mathcal{Y} \in \mathbb{K}^{R \times R \times K}$ is slice mix invertible if there is a vector $\mathbf{v} \in \mathbb{K}^R$ such that the matrix $\mathcal{Y} \cdot_3 \mathbf{v}^T$ is invertible. That is, $\mathcal{Y}$ is slice mix invertible if there exists a linear combination of the matrices $\mathcal{Y}(; :, k)$ for $k = 1, \ldots, K$ which is invertible. We call the matrices $\mathcal{Y}(; :, k)$ the frontal slices of $\mathcal{Y}$ and use $\mathbf{Y}_k$ to denote $\mathcal{Y}(; :, k)$ for $k = 1, \ldots, K$.

Let $\mathcal{Y} \in \mathbb{K}^{R \times R \times K}$. An implicit subpencil of $\mathcal{Y}$ is a tensor of the form $\mathcal{Y} \cdot_3 \mathbf{v} \in \mathbb{K}^{R \times R \times 2}$ where $\mathbf{v} \in \mathbb{K}^2$ has rank 2. More generally, call a tensor of the form $\mathcal{Y} \cdot_3 \mathbf{v} \in \mathbb{K}^{R \times R \times L}$ where $\mathbf{v} \in \mathbb{K}^L$ has full row rank an implicit subtensor of $\mathcal{Y}$. For emphasis we note that implicit subpencils and implicit subtensors differ from the usual notions in that they can be revealed by modal multiplication in the third mode of the tensor. Throughout the article, implicit subtensors and subpencils of a given tensor are denoted using the calligraphic letter $\mathcal{S}$. In the reminder of the article we drop the distinction “implicit”, as all subpencils and subtensors we consider are implicit.

### 1.2. Joint generalized eigenvalues and eigenvectors.

Given a tensor $\mathcal{T} \in \mathbb{K}^{R \times R \times K}$, we say a nonzero vector $\mathbf{z} \in \mathbb{K}^R$ is a joint generalized eigenvector (JGE vector) of $\mathcal{T}$ if there exists a $\lambda \in \mathbb{K}^K$ and a nonzero vector $\mathbf{q} \in \mathbb{K}^R$ such that

$$ T_k \mathbf{z} = \lambda_k \mathbf{q} $$

for all $k = 1, \ldots, K$. Note that assuming $\mathbf{q} \neq \mathbf{0}$ guarantees that $\lambda_k = 0$ if and only if $T_k \mathbf{z} = \mathbf{0}$. If $(\lambda, \mathbf{z})$ is a solution to (1.2), then the one-dimensional subspace $\mathcal{L} := \text{span}(\lambda) \subset \mathbb{K}^K$ is a joint generalized eigenvalue (JGE value) of $\mathcal{T}$ and $(\mathcal{L}, \mathbf{z})$ is a joint generalized eigenpair (JGE pair) of $\mathcal{T}$. Throughout the article we make frequent use of the following theorem [13, Theorem 2.3 (1)] which explains a connection between JGE vectors and tensor rank.

**Theorem 1.1.** Let $\mathcal{T}' \in \mathbb{K}^{I_1 \times I_2 \times I_3}$ and assume $\mathcal{T}'$ has multilinear rank $(R, R, K)$. Let $\mathcal{T} \in \mathbb{K}^{R \times R \times K}$ be an orthogonal compression of $\mathcal{T}'$. Then $\mathcal{T}'$ has $\mathbb{K}$-rank $R$ if and only if the JGE vectors of $\mathcal{T}$ span $\mathbb{K}^R$. Furthermore, if $\mathcal{T} = [\mathbf{A}, \mathbf{B}, \mathbf{C}]$ has $\mathbb{K}$-rank $R$, then the JGE values of $\mathcal{T}$ are equal to the spans of columns of the factor matrix $\mathbf{C}$, and the columns of $\mathbf{B}^{-1}$ are JGE vectors of $\mathcal{T}$.

Given a subspace $\mathcal{Z} \subset \mathbb{K}^R$ of dimension $\ell$ and a tensor $\mathcal{T} \in \mathbb{K}^{R \times R \times K}$, say $\mathcal{Z}$ is a joint generalized eigenspace of $\mathcal{T}$ if there is a subspace $\mathcal{W} \subset \mathbb{K}^R$ of dimension less than or equal to $\ell$ such that

$$ T_k \mathcal{Z} \subseteq \mathcal{W} \quad \text{for all } k = 1, \ldots, K. $$

---

1. Note that one always has $\text{rank}_{\mathbb{K}}(\mathcal{T}') \geq R_3(\mathcal{T}')$ hence the case $K > R$ does not occur after orthogonal compression.

2. If $K = 2$, then joint generalized eigenvectors and joint generalized eigenvalues are exactly equal to classical generalized eigenvalues and generalized eigenvectors (in the sense of [35]).
Here $T_3 \mathbf{z}$ denotes the subspace $\{ T_3 \mathbf{z} : \mathbf{z} \in \mathcal{Z} \} \subset \mathbb{K}^R$. In the case that $K = 2$ we simply call $\mathcal{Z}$ a generalized eigenspace.\(^3\) Given a JGE value $\mathcal{L} \subset \mathbb{K}^K$ of a tensor $\mathcal{T} \in \mathbb{K}^{R \times R \times K}$, let $\mathcal{Z}_2 \subset \mathbb{K}^R$ be the subspace of joint generalized eigenvectors corresponding to $\mathcal{L}$. One may easily show that $\mathcal{Z}_2$ is a joint generalized eigenspace of $\mathcal{T}$.

### 1.2.1. Generalized eigenspaces of subtensors. The main goal of this article is to improve the accuracy of algebraic (pencil-based) methods for computing a CPD. Key observations which allow us to accomplish this are that generalized subspaces may often be more stably computed than individual generalized eigenvectors and that the joint generalized eigenvectors of a tensor may be recovered by computing generalized eigenspaces for sufficiently many subtensors. The following proposition helps formalize this idea.

**Proposition 1.2.** Let $\mathcal{T} \in \mathbb{K}^{R \times R \times K}$ be a tensor of $\mathbb{K}$-rank $R$ and let $\mathbf{V} \in \mathbb{K}^{K \times L}$ be a matrix with full row rank. Set $\mathcal{S} = \mathcal{T} \cdot V^T$, assume $\mathcal{S}$ is slice mix invertible, and let $\mathcal{L}_S$ be a JGE value of $\mathcal{S}$. There exist JGE values $\mathcal{L}_1, \ldots, \mathcal{L}_J$ of $\mathcal{T}$ such that

$$3_{\mathcal{L}_S} = \text{span}(\{3_{\mathcal{L}_j}\}_{j=1}^J).$$

**Proof.** First note that assuming $\mathcal{S}$ is slice mix invertible implies $\mathcal{T}$ is slice mix invertible. Now, following the proof of [13, Proposition 5.1], one finds that each joint generalized eigenvalue of $\mathcal{S}$ has the form $\mathbf{V}^T \mathcal{L}$ for some JGE value $\mathcal{L}$ of $\mathcal{T}$. Note that one must have $\mathbf{V}^T \mathcal{L} = 0$ as otherwise Theorem 1.1 would imply that $\mathcal{S}$ has rank less than $R$, contradicting the assumption that $\mathcal{S}$ is slice mix invertible. Given some JGE value $\mathcal{L}_S$ of $\mathcal{S}$, let $\mathcal{L}_1, \ldots, \mathcal{L}_J$ be those joint generalized eigenvalues of $\mathcal{T}$ such that $\mathbf{V}^T \mathcal{L}_j = \mathcal{L}_S$. It is straightforward to check that

$$\text{span}(\{3_{\mathcal{L}_j}\}_{j=1}^J) \subset 3_{\mathcal{L}_S}. \tag{1.3}$$

To prove the reverse containment, use the assumption that $\mathcal{S}$ is slice mix invertible and the fact that $\mathcal{T}$ has rank $R$ to conclude $\mathcal{S}$ has rank $R$. Letting $\{\mathcal{L}_{S,1}, \ldots, \mathcal{L}_{S,J}\}$ be the set of JGE values of $\mathcal{S}$, it follows from Theorem 1.1 together with [13, Lemma 2.2] that the collection of spaces $\{\mathcal{Z}_{S,j}\}_{j=1}^J$ spans $\mathbb{K}^R$ and also that these spaces pairwise have trivial intersection. Again using Theorem 1.1 shows that the JGE vectors of $\mathcal{T}$ span $\mathbb{K}^R$. Thus, since each JGE vector of $\mathcal{T}$ lies in some $3_{\mathcal{L}_{S,j}}$, the containment in equation (1.3) must in fact be an equality. \(\square\)

Proposition 1.2 has the implication that one may, for example, compute joint generalized eigenvectors of $\mathcal{T}$ by computing intersections of generalized eigenspaces of various different subpencils of $\mathcal{T}$. In this approach one must compute enough generalized eigenspaces so that all individual generalized eigenvectors may be recovered by taking intersections of generalized eigenspaces.

### 1.2.2. The generalized eigenvalue decomposition for CPD computation. Suppose $\mathcal{T} = [\mathbf{A}, \mathbf{B}, \mathbf{C}] \in \mathbb{R}^{R \times R \times K}$ has rank $R$ and is slice mix invertible, and let $\mathbf{V} \in K \times 2$ be a generic matrix. Additionally set $\mathcal{S} = \mathcal{T} \cdot V^T = [\mathbf{A}, \mathbf{B}, \mathbf{V}^T \cdot \mathbf{C}]$. Since $\mathbf{V}$ is generic, the tensor $\mathcal{S}$ has rank $R$ and is slice mix invertible. It follows, e.g. from Theorem 1.1, that the factor matrix $\mathbf{B}$ of $\mathcal{T}$ can be computed

\(^3\)Generalized eigenspaces are sometimes called deflating subspaces by other authors.
by computing a generalized eigenvalue decomposition of $\mathcal{S}$. Having solved for $B$, one may recover $A$ and $C$ by solving for $C \odot A$ in

$$T_{[2,3,1]} = B(C \odot A)^T,$$

then computing rank-1 decompositions of the columns of $C \odot A$.

The key difficulty in this algorithm that we aim to address is that the step which maps $T$ to $\mathcal{S} = T \cdot 3 \, V^T$ is not stable \cite{1}. Intuitively, this is related to the fact that the separation between the columns of $V^T \, C$ is expected to be much smaller than the separation between the columns of $C$. Indeed when one projects a collection of vectors in $\mathbb{R}^K$ to $\mathbb{R}^2$, the angle between those vectors cannot increase and is expected to decrease.\footnote{In fact, if one projects a collection of vectors from $\mathbb{R}^K$ to $\mathbb{R}^2$, then the angle between some pair of vectors in the collection must decrease unless the collection is contained in a two-dimensional subspace of $\mathbb{R}^K$.}

The original GEVD algorithm was communicated in 1970 by Harshman who credits the algorithm to Jennrich \cite{20}. The GEVD algorithm has since been rediscovered many times, e.g., see \cite{31, 25, 15}. We note that aside from GEVD, there are several popular semi-algebraic methods for computing the CPD of a tensor, e.g., see \cite{6, 28, 30}.

1.3. Guide to the reader. Section 2 explains the generalized eigenspace decomposition (GESD) algorithm for computation of a canonical polyadic decomposition. In Section 3, we give a bound that is predictive for the maximal noise level that does not inhibit accurate decomposition. In particular, we give a deterministic bound using perturbation theory for JGE values. Sections 4 and 5 present numerical experiments for the GESD algorithm. In Section 4 we compare the performance (in terms of timing and accuracy) of GESD to the classical GEVD algorithm. In Section 5 we compare the accuracy of the GESD algorithm to the bound computed in Section 3.

2. The GESD algorithm. We now give a formal description of the GESD algorithm for computation of a canonical polyadic decomposition. The main step of the algorithm is to use generalized eigenspaces to recursively split a given tensor into subtensors of reduced rank until the subtensors all have rank one (up to noise). The CPD of each (noisy) rank one tensor may be computed by a best rank one approximation, and the CPD for the full tensor is obtained by combining the decompositions of these rank one tensors.

Let $T$ be an $R \times R \times K$ slice mix invertible tensor of rank $R$ with CPD $T = [A, B, C]$. Let $V \in \mathbb{R}^{K \times 2}$ be a column-wise orthonormal matrix\footnote{Column-wise orthonormal matrices are used for numerical reasons.} of size $K \times 2$ and set $\mathcal{S} = T \cdot 3 \, V^T$. The GESD algorithm begins by computing a collection of generalized eigenspaces $\mathcal{Z}_1, \ldots, \mathcal{Z}_L$ for the generalized eigenvalue problem

\begin{equation}
S_i z = \lambda_i q \text{ for } i = 1, 2
\end{equation}

such that $\text{span}(\{\mathcal{Z}_i\}_{i=1}^L) = \mathbb{R}^K$. In each step the GESD algorithm computes as many well-separated eigenspaces as possible; however, to simplify exposition we will take $L = 2$.

Say the generalized eigenvalues of $\mathcal{S}$ are split into two distinct groups of size $j_1$ and $j_2$ where $j_1 + j_2 = R$. The corresponding eigenspaces may then for instance be
computed with a QZ decomposition\(^6\) of the matrix pencil \((S_1, S_2)\) [3]. In particular, the QZ decomposition computes unitary matrices \(Q, Z \in \mathbb{C}^R\) such that

\[
S_1 = QX_1Z^H \quad \text{and} \quad S_2 = QX_2Z^H
\]

where \(X_1\) and \(X_2\) are (quasi\(^7\)) upper triangular matrices with the classical generalized eigenvalues of the pencil on the diagonal.

If the decomposition is ordered so that the \(j_1\) generalized eigenvalues in the first cluster appear first on the diagonal of \(X_1\) and \(X_2\), then the first \(j_1\) columns of \(Z\) give us a matrix \(Z_1\) whose columns form a basis for the corresponding generalized eigenspace of \(S\). Similarly, we obtain a matrix \(Z_2\) whose columns form a basis for the eigenspace corresponding to the next \(j_2\) generalized eigenvalues by reordering the terms in the QZ decomposition so that these generalized eigenvalues appear first on the diagonal of \(X_1\) and \(X_2\).

Using Proposition 1.2 together with Theorem 1.1 shows that the matrices \(Z_1\) and \(Z_2\) that we have obtained are of the form \(Z_i = B_i^{-T}E_i\) where the \(E_i\) have the property that the \(j\)th row of \(E_1\) is nonzero if and only if the \(j\)th row of \(E_2\) is a zero row, and vice versa. Define the tensors \(\mathcal{T}^1\) and \(\mathcal{T}^2\) by

\[
\mathcal{T}^i = \mathcal{T} \cdot Z_i^T = [A, Z_i^T B, C] = [A, E_i^T, C]
\]

and for \(i = 1, 2\) set \(R_i = \text{rank}(\mathcal{T}^i)\). Since the \(E_i^T\) have zero columns corresponding to the zero rows of \(E_i\), we have that \(R_1 + R_2 = \text{rank}(\mathcal{T})\). Moreover, suppose \(\mathcal{T}\) has CPD \([A_1, E_1, C_1]\) for \(i = 1, 2\). Here the matrix \(E_i \in \mathbb{K}^{R_i \times R_i}\) is the matrix obtained by removing the zero columns of \(E_i^T\). Then up to permutation and scaling, one has \(A = (A_1, A_2)\). Similarly, the factor matrix \(C\) of \(\mathcal{T}\) has the form \(C = (C_1, C_2)\) up to scaling and the same permutation as \(A\).

This procedure partitions the rank one terms of \(\mathcal{T}\) into two distinct collections so that one collection gives a decomposition for the tensor \(\mathcal{T}_1\) while the other collection gives a decomposition for \(\mathcal{T}_2\). The number of rank one terms in each collection, i.e., the rank of \(\mathcal{T}_1\) and of \(\mathcal{T}_2\), is determined by the number of generalized eigenvalues in each cluster.

The above methodology may be recursively applied to each \(\mathcal{T}^i\) to further reduce their ranks until one obtains a collection of tensors all having rank one (up to noise). The columns of the factor matrices \(A\) and \(C\) may then be approximated by computing rank one approximations for each of these (approximately) rank one tensors, see Section 2.1 below for details.

Once we have computed the factor matrices \(A\) and \(C\), we may compute \(B\) by solving

\[
T_{[2,3,1]} = B(C \odot A)^T.
\]

We emphasize that the GESD algorithm is guaranteed to recover a CPD of \(\mathcal{T}\) in the noiseless setting under the assumption that \(\mathcal{T}\) is an \(R \times R \times K\) slice mix invertible tensor of rank \(R\). Note that the CPD of such a tensor is not necessarily unique. In particular, if \(\mathcal{T} = [A, B, C]\) meets this assumption but the factor matrix \(C\) has a repeated column, i.e. if \(\mathcal{T}\) has a repeated joint generalized eigenvalue, then GESD can successfully recover a CPD of \(\mathcal{T}\) even though \(\mathcal{T}\) does not have a unique CPD.

\(^6\)The QZ decomposition is the generalization of the Schur decomposition to matrix pencils, e.g., see [35, Chapter VI, Theorem 1.9].

\(^7\)When working over \(\mathbb{R}\), if \(S\) has complex generalized eigenvectors, then since \(Q\) and \(Z\) have real entries, there will be \(2 \times 2\) block matrices on the diagonal of \(X_1\) and \(X_2\). E.g., see [22, Theorem 1.2].
2.1. Rank one terms. Consider the case where $Z_1 \in \mathbb{K}^{R \times 1}$ where $Z_1$ spans an eigenspace of dimension one. In this case $Z_1$ must have the form

$$Z_1 = \beta B^{-T} e_\ell^T$$

where $e_\ell$ is a standard basis vector and $\beta \in \mathbb{K}$ is some constant. Thus we have

$$T \cdot_2 Z_1^T = [A, \beta e_\ell, C] = [a_\ell, \beta, c_\ell].$$

In this case we may compute $a_\ell$ and $c_\ell$ by computing a rank one approximation of the matrix $T \cdot_2 Z_1^T$. In practice one may verify that such a tensor has rank one (up to noise) by computing the multilinear singular values of $T \cdot_2 Z_1^T$.

2.2. Choice of slices. We observe that the average performance of the GESD algorithm is improved by first considering a pencil formed by taking the first two frontal slices of the core of the MLSVD of the current subtensor on each splitting step. Intuitively, this is related to the fact that this matrix pencil explains more of the overall sum of squared entries than any other matrix pencil which can be formed from the frontal slices of the tensor.

We emphasize that this strategy is not always successful. Indeed, there are many settings where the generalized eigenvalues obtained by considering the first two slices of the core of an MLSVD are poorly separated. When this occurs it is necessary to consider other matrix pencils. See Section 4.4 for an example. In the case that a successful split is not obtained by considering slices of the MLSVD, we move on to considering random linear combinations of frontal slices.

Note that an MLSVD is always computed for each subtensor to address other potential issues (e.g., repeated factor matrix columns). Furthermore computing MLSVDs of the subtensors is cheap relative to other steps of the algorithm. Thus first considering slices of the MLSVDs has little impact on overall computational cost and on average significantly improves accuracy.

Pseudo code for the GESD algorithm is found in Algorithm 1 and Algorithm 2.

2.3. Handling of complex eigenvectors when working over $\mathbb{R}$. Our explanation of the GESD algorithm above focuses on the noiseless setting. An important aspect of this setting is that the joint generalized eigenvectors of a slice mix invertible real rank $R$ tensor $T \in \mathbb{R}^{R \times R \times K}$ are necessarily real valued [13], and, as a consequence, the subpencils of such a tensor have real valued generalized eigenvectors. In the noisy setting, however, it is possible that a subpencil of a perturbation $Y = T + \mathcal{N}$ has complex valued generalized eigenvectors.

For example, suppose $T \in \mathbb{R}^{R \times R \times K}$ is a slice mix invertible tensor of real rank $R$, and suppose $\mathcal{Z}$ is a JGE space of dimension $2$. Let $Z \in \mathbb{K}^{R \times 2}$ be a matrix whose columns form a basis for $\mathcal{Z}$ and let $S$ be an MLSVD of the tensor $Y \cdot_2 Z^T$. Then $S$ is a $2 \times 2 \times 2$ tensor. In the noiseless setting (i.e., if $Y = T$), if $S$ is slice mix invertible, then $S$ necessarily has a real rank 2 and both generalized eigenvectors of $S$ are real valued. However, with sufficient noise, it is possible that $S$ has real rank 3. This corresponds to the case where the generalized eigenvectors of $S$ are complex valued.

If $S$ has rank 3, then $S$ does not have a best real rank 2 approximation [8], so $S$ cannot be further decomposed and the true real rank $R$ decomposition of the original tensor cannot be computed by the GESD algorithm at this point. However, the factor matrix $A$ (resp. $C$) in the CPD for the original tensor will have two columns which
Algorithm 1: GESD recursive step

1 function cpd_gesd (M, R, Random)
   Input : Tensor M, rank R, boolean Random
   Output: [A, B, C]
2   K = min(size(T, 3), R)
3   [V1, V2, V3, Y] = mlsvd(M, [R, R, K]);  % Compute MLSVD of T / Compress
4   if R = 1 then
5       Acore = y111; Bcore = 1; Ccore = 1;  % Trivial case
6   else
7       [Z1, ..., Z_N] = qz_split(Y, Random);  % Split JGE spaces into N separated subgroups (Algorithm 2)
8       for i = 1 : N do
9           Y_i = Y · Z_i^T;  % Reduce tensor to Nth subgroup
10          R_i = size(Z_i, 2)  % Specify rank for Nth subgroup
11          [A_i, B_i, C_i] = cpd_gesd(Y_i, R_i);  % Recursion on smaller tensor and smaller rank
12       end
13       Acore = [A_1, ..., A_N]; Ccore = [C_1, ..., C_N];  % Build A and C from subblocks
14       Bcore = Y_{[2,3,1]} / (C ∘ A)^T;  % Find B by solving linear system
15   end
16   A = V1Acore; B = V2Bcore; C = V3Ccore;  % Undo the MLSVD / Expand

(approximately) lie in the mode-1 (resp. mode-3) subspace of the tensor Y · Z^T.

At this point one may accept that these particular columns of A and C cannot be computed and replace them with bases for the two-dimensional subspaces to which they belong. Alternatively, one could use the results from the GESD algorithm (with the bases for these two-dimensional subspaces used as those two columns for A and C) as an initialization in an optimization routine for computing the full CPD. While it is unlikely that these basis vectors are the true columns of A or C, the optimization routine will at least be initialized with these columns in the correct two-dimensional subspace which is a significant improvement over a random initialization.

3. A deterministic bound for distinct eigenvalue clusters. Our main goal in this section is to provide a deterministic bound for the existence of a subpencil of the tensor of interest which has two distinct clusters of generalized eigenvalues which cannot have coalesced, a notion we now explain.

Suppose T ∈ K^{R×R×2} is a matrix pencil with distinct generalized eigenvalues. Then for sufficiently small perturbations N, the generalized eigenvalues and generalized eigenvectors of T + N are continuous in N. However, if the perturbation N is large enough so that the matrix pencil T + N has a repeated generalized eigenvalue, i.e., so that at least two generalized eigenvalues of T + N have coalesced, then the path of corresponding generalized eigenvectors cannot be unambiguously continued.

\(^8\)When working over C there is no natural way to sort the generalized eigenvalues. As a consequence one must compute the chordal distance between all pairs of generalized eigenvalues in order to cluster the generalized eigenvalues into well-separated collections.
**Algorithm 2:** Splitting eigenspaces (over $\mathbb{R}$)

```
function qz_split ($\mathcal{Y}$, Random)
Input : MLSVD core tensor $\mathcal{Y}$, boolean Random
Output: $N$ matrices $Z_n$ spanning JGE spaces

1. $R = \text{size}(\mathcal{Y}, 1)$
2. $N = 0$

while $N < 2$ do

3. if Random then
4. $S = \mathcal{Y} \cdot \text{qr}([\text{randn}(R, 2)]^T)$; % Construct random pencil
5. else
6. Set indices $k_1$ and $k_2$ of next set of MLSVD slices
7. $S = \mathcal{Y}(:,[k_1,k_2])$; % Take MLSVD slices as pencil
8. end
9. $[X_1, X_2, Q, Z] = \text{qz}(S::1, S::2)$; % Compute QZ decomposition of pencil
10. Order eigenvalues $x_{2,ii}/x_{1,ii}$
11. Compute chordal gaps between adjacent pairs $(x_{1,ii}, x_{2,ii})$
12. Set $N$ to be the number of chordal gaps above threshold

end

13. Split eigenvalues into $N$ clusters according to gaps above threshold

14. for $n = 1 : N$ do
15. Define binary vector $i_n$ indicating the $n$th cluster.
16. $[X_{1,n}, X_{2,n}, Q_n, Z_n] = \text{ordqz}(X_1, X_2, Q, Z, i_n)$;
17. % Modify QZ decomposition such that $n$th cluster is at top
18. $Z_n = (Z_n)_{:,1:n(i_n)}$ % Extract JGE space of $n$th cluster

end
```

as after the generalized eigenvalues separate there is ambiguity in which generalized eigenvalue corresponds to which generalized eigenvector.

Although it is no longer possible to unambiguously continue the paths of generalized eigenvectors themselves after generalized eigenvalues have coalesced, the path of the corresponding generalized eigenspace is still continuous and unambiguous in $\mathcal{N}$ so long as the underlying generalized eigenvalues do not coalesce with other generalized eigenvalues. Thus guaranteeing the existence of two distinct clusters of generalized eigenvalues which cannot have coalesced guarantees that $\mathcal{T} + \mathcal{N}$ has at least two distinct generalized eigenspaces which are still continuous in the perturbation $\mathcal{N}$. These two generalized eigenspaces are still closely related to the original generalized eigenspaces of $\mathcal{T}$. This idea is formalized for matrix pencils in [9]. Also see [21] for the matrix case.

We note that the bound presented in this section can not only guarantee the existence of clusters of generalized eigenvalues which have not coalesced, but can also guarantee a minimum separation between the clusters of generalized eigenvalues. As previously mentioned, the separation between clusters of generalized eigenvalues heavily impacts the accuracy of computing corresponding generalized eigenspaces, so in this way our bounds have significant implications for the accuracy of a given step of the GESD algorithm.

When discussing our deterministic bound we restrict to the real setting. That is,
in this section $\mathcal{T} \in \mathbb{R}^{R \times R \times K}$ is a slice mix invertible real tensor which has real rank $R$.

We now introduce some notation. Given a matrix $M$, let $\sigma_{\min}(M)$ denote the smallest singular value of $M$. Define the Grassmannian $\text{Gr}(1, \mathbb{K}^2)$ to be the collection of subspaces of $\mathbb{K}^2$ of dimension 1. Note that any element of $\text{Gr}(1, \mathbb{R}^2)$ may be uniquely represented in the form $\text{span}((\cos(\theta), \sin(\theta)))$ for some $\theta \in [0, \pi)$. Thus given a subspace $\mathcal{L} \in \text{Gr}(1, \mathbb{R}^2)$ we define

$$\theta(\mathcal{L}) \in [0, \pi) \quad \text{by} \quad \mathcal{L} = \text{span}\left((\cos(\theta(\mathcal{L})), \sin(\theta(\mathcal{L})))\right).$$

Additionally, for a nonzero vector $\lambda \in \mathbb{R}^2$, define

$$\theta(\lambda) = \theta(\text{span}(\lambda)).$$

The metric we use for the Grassmannian is the chordal metric. Given two subspaces $\mathcal{L}_1, \mathcal{L}_2 \in \text{Gr}(1, \mathbb{K}^2)$ define the chordal metric between $\mathcal{L}_1$ and $\mathcal{L}_2$, denoted $\chi(\mathcal{L}_1, \mathcal{L}_2)$ to be the sine of the (principal) angle between $\mathcal{L}_1$ and $\mathcal{L}_2$. It is straightforward to check that $\text{Gr}(1, \mathbb{K}^2)$ is a path connected metric space under the chordal metric. We extend the definition of $\chi$ to $\mathbb{K}^2 \setminus \{0\}$ by identifying nonzero vectors in $\mathbb{K}^2$ with their spans. That is, given $\lambda_1, \lambda_2 \in \mathbb{K}^2 \setminus \{0\}$ we have

$$\chi(\lambda_1, \lambda_2) := \chi(\text{span}(\lambda_1), \text{span}(\lambda_2)).$$

See [38] for further discussion of the chordal metric and distance in Grassmanians.

Lastly, for a tensor $T \in K^{R \times R \times K}$, define the spectral norm of $T$, denoted $\|T\|_{\text{sp}}$ by

$$\|T\|_{\text{sp}} := \max_{\|x\|_2 = \|y\|_2 = \|z\|_2 = 1} |\langle T \cdot_1 x^T \cdot_2 y^T \cdot_3 z^T \rangle|$$

where the vectors $x, y, z$ are of compatible dimension. The spectral norm of $T$ is equal to the Frobenius norm of a best rank 1 approximation of $T$. See [16, 17, 18, 29] for further discussion.

Our first step towards a bound for an $R \times R \times K$ tensor to have a subpencil in which generalized eigenvalues have not coalesced is to give a bound which guarantees that a matrix pencil (i.e., an $R \times R \times 2$ tensor) has generalized eigenvalues which have not coalesced.

**Theorem 3.1.** Let $T$ and $Y$ be tensors of size $R \times R \times 2$. Assume that $T$ is slice mix invertible and has real rank $R$ with CPD $T = [A, B, C]$, where the columns of $C$ are normalized to have unit norm.

Let $\{\text{span}(\lambda_1), \ldots, \text{span}(\lambda_R)\}$ be the generalized eigenvalues of $T$ where the indices are ordered so that

$$0 \leq \theta(\lambda_r) \leq \theta(\lambda_{r+1}) < \pi \quad \text{for} \quad r = 1, \ldots, R - 1,$$

and assume at least $J$ of the $\lambda_r$ are distinct. Define $\lambda_{R+1} = \lambda_1$ and let $\delta$ be the $J$th largest element of the set

$$\left\{ \frac{\chi(\lambda_r, \lambda_{r+1})}{2} \right\}_{r=1}^R.$$

Set $\epsilon_1 = \sigma_{\min}(A)\sigma_{\min}(B)\delta$ and let $\epsilon_2$ be the distance in the spectral norm from $T$ to the nearest pencil which is not slice mix invertible. If

$$\|T - Y\|_{\text{sp}} < \min\{\epsilon_1, \epsilon_2\},$$

(3.1)
then the tensor \( \mathcal{Y} \) has at least \( J \) distinct (possibly complex valued) generalized eigenvalues.

Furthermore, if equation (3.1) is satisfied and one sets \( \| T - \mathcal{Y} \|_{sp} = \epsilon \), then the generalized eigenvalues of \( \mathcal{Y} \) may be divided into \( J \) nonempty sets \( \{ \mathcal{L}^j \}_{j=1}^J \) such that

\[
\min_{j \neq \ell} \chi(\mathcal{L}^j, \mathcal{L}^\ell) \geq 2\delta - \frac{2\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} > 0.
\]

Remark 3.2. See the upcoming Lemma 3.6 for discussion of how to compute a lower bound for the quantity \( \epsilon_2 \) appearing in Theorem 3.1. In the case \( J = R \) one always has \( \epsilon_1 \leq \epsilon_2 \), see [13, Proposition 4.4].

The proof of Theorem 3.1 follows a simple idea. Using perturbation theoretic results for a matrix pencil, one may show that there are at least \( J \) distinct connected regions in \( Gr(1, \mathbb{C}^2) \) such that each region contains a generalized eigenvalue of \( \mathcal{Y} \) and such that all generalized eigenvalues of \( \mathcal{Y} \) are contained in the union of these disjoint regions. Roughly speaking, the “real parts” of these regions lie between each of the \( J \) largest chordal gaps between the ordered generalized eigenvalues of \( T \). To formalize this idea we introduce the notion of a Bauer–Fike ball and a Bauer–Fike region around a given generalized eigenvalue of \( T \).

Let the tensor \( T = [A, B, C] \) and the generalized eigenvalues \( \{ \text{span}(\lambda_r) \}_{r=1}^R \) be as in the statement of Theorem 3.1. In particular, \( T \) is slice mix invertible with \( \mathbb{R} \)-rank \( R \) and the columns of \( C \) have unit norm. We define the (real or complex) Bauer–Fike ball, for the parameter \( \epsilon \) around a given generalized eigenvalue \( \text{span}(\lambda_r) \) of \( T \), denoted \( BF_{\epsilon}^{K,0}(\lambda_r) \subset Gr(1, \mathbb{R}^2) \), by

\[
BF_{\epsilon}^{K,0}(\lambda_r) = \left\{ \text{span}(\gamma) \in Gr(1, \mathbb{R}^2) : \gamma \in \mathbb{R}^2/\{0\} \text{ and } \chi(\gamma, \lambda_r) \leq \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \right\}.
\]

Having defined Bauer–Fike balls\\(^9\\)\(^{,}\\), we define the Bauer–Fike region \( BF_{\epsilon}^{K}(\lambda_r) \) to be the connected component of the union of Bauer–Fike balls for \( T \) which contains \( \text{span}(\lambda_r) \). That is, we set

\[
\mathcal{D}^K = \bigcup_{r=1}^R BF_{\epsilon}^{K,0}(\lambda_r) \subset Gr(1, \mathbb{R}^2),
\]

and let \( \mathcal{D}^K = \bigcup_{r=1}^{R'} \mathcal{D}_r^K \) be a partitioning of \( \mathcal{D}^K \) into \( R' \leq R \) disjoint connected components (in the topology induced by the chordal metric). The Bauer–Fike region \( BF_{\epsilon}^{K}(\lambda_r) \) is the unique connected component of \( \mathcal{D}^K \) which contains \( \text{span}(\lambda_r) \).

For use in the proof of the upcoming Proposition 3.4, we also provide the following equivalent definition of a Bauer–Fike region. For each \( N = 0, 1, 2, \ldots \) define

\[
BF_{\epsilon}^{K,N+1}(\lambda_r) = \bigcup_{BF_{\epsilon}^{K,N}(\lambda_r) \cap BF_{\epsilon}^{K,0}(\lambda_r) \neq \emptyset} BF_{\epsilon}^{K,0}(\lambda_r).
\]

That is, \( BF_{\epsilon}^{K,N+1}(\lambda_r) \) is the union over \( \ell \) of Bauer–Fike balls \( BF_{\epsilon}^{K,0}(\lambda_r) \) such that \( BF_{\epsilon}^{K,N}(\lambda_r) \) has nontrivial intersection with \( BF_{\epsilon}^{K,0}(\lambda_r) \). The Bauer–Fike region \( BF_{\epsilon}^{K}(\lambda_r) \) is defined by

\[
BF_{\epsilon}^{K}(\lambda_r) = \bigcup_{N=1}^{\infty} BF_{\epsilon}^{K,N}(\lambda_r).
\]

\\(^9\)\(^{For clarity we emphasize that a Bauer–Fike ball for the parameter \( \epsilon \) around a generalized eigenvalue \( \mathcal{L} \) of \( T \) is simply the closed neighborhood in \( Gr(1, \mathbb{R}^2) \) of radius \( \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \) centered at \( \mathcal{L} \). We introduce this notation to streamline the proof of the upcoming Proposition 3.4, and to clearly illustrate the connection between these sets.\)
It is not difficult to show that there is an integer \( N \) such that

\[
BF^N_\epsilon (\lambda_r) = BF^N_\epsilon (\lambda_r).
\]

In particular one can take \( N = R \).

The complex Bauer–Fike regions for a given tensor \( T \) are useful in that they give us control over the locations of the generalized eigenvalues of a perturbation of \( T \). Assume \( \epsilon \) is small enough so that \( \|T - Y\|_{\text{sp}} \leq \epsilon \) implies that \( Y \) is slice mix invertible. Then following the argument in the proof of [13, Theorem 3.5] shows that the number of generalized eigenvalues of \( Y \) contained by each complex Bauer–Fike region for the parameter \( \epsilon \) is equal to the number of overlapping complex Bauer–Fike balls which make up that complex Bauer–Fike region. That is, for a fixed index \( r \), if there are exactly \( m \) indices \( \ell \) (counting the case \( r = \ell \)) such that

\[
BF^c_\epsilon (\lambda_r) \cap BF^{c,0}_\epsilon (\lambda_\ell) \neq \emptyset,
\]

and if \( \|T - Y\|_{\text{sp}} \leq \epsilon \), then the complex Bauer–Fike region \( BF^c_\epsilon (\lambda_r) \) contains exactly \( m \) of the generalized eigenvalues of the tensor \( Y \) counting algebraic multiplicity.

Our goal in the proof of Theorem 3.1 will be to show that \( T \) has at least \( J \) distinct complex Bauer–Fike regions for the parameter \( \epsilon \). Following the above discussion shows that if \( T \) and \( Y \) satisfy the assumptions of Theorem 3.1, then \( Y \) has at least \( J \) distinct clusters of generalized eigenvalues which cannot have coalesced in a perturbation from \( T \) to \( Y \). To show that \( T \) has \( J \) distinct Bauer–Fike regions, we will make use of the fact that Bauer–Fike regions are connected subsets of \( \text{Gr}(1, \mathbb{K}^2) \).

**Lemma 3.3.** Let \( T \in \mathbb{R}^{R \times R \times 2} \) be a slice mix invertible \( R \)-rank \( R \) tensor with generalized eigenvalues \( \Sigma = \{\lambda_r\}_{r=1}^R \). Then for each \( r \), the Bauer–Fike region \( BF^c_\epsilon (\lambda_r) \subset \text{Gr}(1, \mathbb{K}^2) \) is path connected in the topology on \( \text{Gr}(1, \mathbb{K}^2) \) given by the chordal metric.

Another technical part of our proof will be to restrict from considering complex Bauer–Fike regions to considering real Bauer–Fike regions. We emphasize that the generalized eigenvalues of \( Y \) can only be guaranteed to lie in the union of the complex Bauer–Fike regions of \( T \), thus it is necessary to use complex Bauer–Fike regions. However, complex Bauer–Fike regions are more difficult to work with due to a lack of any notion of ordering in \( \text{Gr}(1, \mathbb{C}^2) \), hence our desire to reduce to working over \( \mathbb{R} \).

The following proposition makes this reduction possible.

**Proposition 3.4.** Let \( T \in \mathbb{R}^{R \times R \times 2} \) be a slice mix invertible tensor having real rank \( R \). Let \( \{\text{span}(\lambda_r)\}_{r=1}^R \) be the set of generalized eigenvalues of \( T \) with \( \lambda_r \in \mathbb{R}^2 \) for each \( r = 1, \ldots, R \). Then for any \( r_1, r_2 \) one has

\[
BF^{c,0}_\epsilon (\lambda_{r_1}) \cap BF^{c,0}_\epsilon (\lambda_{r_2}) = \emptyset \quad \text{if and only if} \quad BF^{R,0}_\epsilon (\lambda_{r_1}) \cap BF^{R,0}_\epsilon (\lambda_{r_2}) = \emptyset.
\]

Additionally,

\[
BF^c_\epsilon (\lambda_{r_1}) = BF^c_\epsilon (\lambda_{r_2}) \quad \text{if and only if} \quad BF^R_\epsilon (\lambda_{r_1}) = BF^R_\epsilon (\lambda_{r_2}).
\]

It follows that the number of distinct complex Bauer–Fike regions for \( T \) is exactly equal to the number of distinct real Bauer–Fike regions for \( T \).

**Proof.** Let \( T \) and \( \lambda_{r_1} \) and \( \lambda_{r_2} \) be as in the statement of the lemma. It is clear that \( BF^{c,0}_\epsilon (\lambda_{r_1}) \cap BF^{c,0}_\epsilon (\lambda_{r_2}) = \emptyset \) implies \( BF^{R,0}_\epsilon (\lambda_{r_1}) \cap BF^{R,0}_\epsilon (\lambda_{r_2}) = \emptyset \). On the other hand, using the fact that a vector \( \zeta \) which minimizes the chordal distance to a set of real vectors \( \{\lambda_1, \lambda_2\} \) may be obtained by taking \( \zeta \) to be a real vector that bisects the angle
to show that if \( BF_{\epsilon}^{C,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{C,0}(\lambda_{r_2}) \neq \emptyset \), then \( BF_{\epsilon}^{R,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{R,0}(\lambda_{r_2}) \neq \emptyset \). In particular, if \( BF_{\epsilon}^{C,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{C,0}(\lambda_{r_2}) \neq \emptyset \), then \( \text{span}(\zeta) \in BF_{\epsilon}^{C,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{C,0}(\lambda_{r_2}) \) hence \( \text{span}(\zeta) \in BF_{\epsilon}^{R,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{R,0}(\lambda_{r_2}) \).

Thus we have \( BF_{\epsilon}^{C,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{C,0}(\lambda_{r_2}) = \emptyset \) if and only if \( BF_{\epsilon}^{R,0}(\lambda_{r_1}) \cap BF_{\epsilon}^{R,0}(\lambda_{r_2}) = \emptyset \).

Using equation (3.3) together with the above, one may show by induction that for all \( N \in \mathbb{N} \) and all \( r_1, r_2 \in \{1, \ldots, R\} \) one has

\[
BF_{\epsilon}^{C,0}(\lambda_{r_2}) \subset BF_{\epsilon}^{C,N}(\lambda_{r_1}) \quad \text{if and only if} \quad BF_{\epsilon}^{R,0}(\lambda_{r_2}) \subset BF_{\epsilon}^{R,N}(\lambda_{r_1}).
\]

Furthermore, it is straightforward to check that

\[
BF_{\epsilon}^{K,0}(\lambda_{r_1}) = BF_{\epsilon}^{K}(\lambda_{r_2})
\]

if and only if there is an integer \( N \) such that

\[
BF_{\epsilon}^{K,0}(\lambda_{r_2}) \subset BF_{\epsilon}^{K,N}(\lambda_{r_1}).
\]

Combining this fact with equation (3.4) completes the proof.

**3.1. Proof of Theorem 3.1.** We have now laid the necessary groundwork to give the proof of Theorem 3.1. The proof is technical, but, as previously mentioned, the main idea behind the proof is simple. Before giving technical details, we give an overview of the role played by Bauer–Fike regions.

As previously discussed, each distinct Bauer–Fike region for \( T \) contains a generalized eigenvalue of \( Y \) and all the generalized eigenvalues of \( Y \) are contained in some Bauer–Fike region of \( T \). This fact is proved in the tensor spectral norm by [13, Theorem 4.3]; however, in a different norm (which upper bounds the tensor spectral norm) this fact is classical, e.g. see [35, Chapter VI, Theorem 2.7]. By using this fact together with Proposition 3.4 to show that \( Y \) has at least \( J \) distinct generalized eigenvalues, it is sufficient to show that \( T \) has \( J \) distinct real Bauer–Fike regions.

To show that \( T \) has at least \( J \) distinct real Bauer–Fike regions, we divide \( \text{Gr}(1, \mathbb{R}^2) \) into \( J \) distinct regions such that each region fully contains a real Bauer–Fike region of \( T \). The desired division of \( \text{Gr}(1, \mathbb{R}^2) \) into distinct regions is accomplished by using subspaces of \( \mathbb{R}^2 \) to cut in half each of the \( J \) largest gaps between the ordered generalized eigenvalues of \( T \).

Our choice of parameter for our Bauer–Fike balls will guarantee that these cuts cannot be contained in any Bauer–Fike ball of \( T \), hence, these cuts cannot be contained in any Bauer–Fike region. The fact that a real Bauer–Fike region lying between two adjacent cuts is distinct from a real Bauer–Fike region lying between another pair of adjacent cuts follows from the path connectedness of the real Bauer–Fike regions. Details follow below.

**Proof.** Let \( T = [A, B, C] \) and \( J \) and \( \epsilon \) be as in the statement of Theorem 3.1 and assume \( J \geq 2 \). We first show that \( T \) has \( J \) distinct complex Bauer–Fike regions. Using Proposition 3.4, it is sufficient to show that \( T \) has \( J \) distinct real Bauer–Fike regions.

Now, from the ordering of our eigenvalues, there must exist \( J \) distinct indices \( 1 \leq r_1 < \cdots < r_J \leq R \) such that for each \( r_j \neq R \) there is a unit vector \( \eta_j \in \mathbb{R}^2 \) satisfying

\[
\theta(\lambda_{r_j}) < \theta(\eta_j) < \theta(\lambda_{r_j+1})
\]
and
\[ \chi(\eta_j, \lambda_{r_j}) \geq \delta \quad \text{and} \quad \chi(\eta_j, \lambda_{r_j+1}) \geq \delta. \]
Using \( \epsilon_1 = \sigma_{\min}(A)\sigma_{\min}(B)\delta \) and \( \epsilon < \epsilon_1 \) gives
\[ \chi(\eta_k, \lambda_{r_k}) > \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \quad \text{and} \quad \chi(\eta_k, \lambda_{r_k+1}) > \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)}. \]
If the case \( r_j = R \) occurs, then we may instead choose a unit vector \( \eta_J \) such that
\[ \chi(\eta_J, \lambda_R) > \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \quad \text{and} \quad \chi(\eta_J, \lambda_1) > \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \]
and such that either
\[ \theta(\lambda_R) < \theta(\eta_J) \quad \text{or} \quad \theta(\eta_J) < \theta(\lambda_1). \]
To simplify exposition, we assume that \( \theta(\lambda_R) < \theta(\eta_J). \)

Note that \( \bigcup_{j=1}^J \text{span}(\eta_j) \) divides \( \text{Gr}(1, \mathbb{R}^2) \) into \( J \) distinct open connected components. In particular, there exist \( J \) disjoint open subsets \( \mathcal{G}_1, \ldots, \mathcal{G}_J \) of \( \text{Gr}(1, \mathbb{R}^2) \) such that
\[ \bigcup_{j=1}^J \mathcal{G}_j = \text{Gr}(1, \mathbb{R}^2) \setminus (\bigcup_{j=1}^J \text{span}(\eta_j)). \]

Using the ordering of our generalized eigenvalues together with the definition of the \( \eta_j \), a straightforward check shows that \( \text{span}(\eta_j) \) cannot be contained in a real Bauer–Fike region \( BF_{\epsilon_r}(\lambda_r) \) for any \( r = 1, \ldots, R. \) In particular we have
\[ \bigcup_{r=1}^R BF_{\epsilon_r}(\lambda_r) \subset (\bigcup_{j=1}^J \mathcal{G}_j). \]
(3.5) Recalling that each Bauer–Fike region is connected and that \( \mathcal{G}_j \) are open and disjoint sets, equation (3.5) then implies that if there are indices \( r \) and \( j \) such that
\[ BF_{\epsilon_r}(\lambda_r) \cap \mathcal{G}_j \neq \emptyset, \quad \text{then} \quad BF_{\epsilon_r}(\lambda_r) \subseteq \mathcal{G}_j. \]
(3.6) By construction we have
\[ \mathcal{G}_j \cap \left( \bigcup_{r=1}^R BF_{\epsilon_r}(\lambda_r) \right) \neq \emptyset \quad \text{for each} \quad j = 1, \ldots, J. \]
Combining this with equation (3.6) shows that each \( \mathcal{G}_j \) contains some real Bauer–Fike region of \( T. \) In particular, \( T \) has at least \( J \) distinct real Bauer–Fike regions. An application of Proposition 3.4 then shows that \( T \) has at least \( J \) distinct complex Bauer–Fike regions.

Now let \( Y \in \mathbb{R}^{R \times R \times 2} \) be a tensor which satisfies
\[ \|T - Y\|_{sp} \leq \epsilon < \min\{\epsilon_1, \epsilon_2\}. \]
Following the argument used to prove [13, Theorem 2.8] shows that for each \( j = 1, \ldots, J \) the region \( BF_{\epsilon_r}(\lambda_r) \) must contain at least one generalized eigenvalue of \( Y \) and that all the generalized eigenvalues of \( Y \) are contained in some Bauer–Fike region. Therefore, since \( T \) has at least \( J \) distinct Bauer–Fike regions, the tensor \( Y \) has at least \( J \) distinct generalized eigenvalues.

It remains to construct the nonempty sets \{\( L^j \)\}_{j=1}^J \ of generalized eigenvalues of \( Y \) which satisfy equation (3.2). To this end, let \{\( \text{span}(\lambda_r) \)\}_{r=1}^R \ be the set of generalized eigenvalues of \( Y \) and for each \( j = 1, \ldots, J \) define \( L^j \) by
\[ L^j = \left\{ \text{span}(\hat{\lambda}_r) : \hat{\lambda}_r \in \{\hat{\lambda}_r\}_{r=1}^R \cap G_j \right\}. \]
In words, \( \mathcal{G}^j \) is the set of generalized eigenvalues of \( Y \) which are contained in \( \mathcal{G}_j \).

Now suppose \( \lambda_m \) and \( \lambda_n \) are generalized eigenvalues of \( Y \) such that \( \lambda_m \in \mathcal{G}_i \) and \( \lambda_n \in \mathcal{G}_\ell \) for some indices \( i \neq \ell \). Then there must exist generalized eigenvalues \( \tilde{\lambda}_m \) and \( \tilde{\lambda}_n \) of \( \mathcal{T} \) such that \( \lambda_m \in \mathcal{G}_i \) and \( \lambda_n \in \mathcal{G}_\ell \) and
\[
\chi(\lambda_m, \tilde{\lambda}_m) \leq \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)} \quad \text{and} \quad \chi(\lambda_n, \tilde{\lambda}_n) \leq \frac{\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)}.
\]
Furthermore, from the definition of the \( \mathcal{G}_j \) we must have
\[
\chi(\lambda_m, \lambda_n) \geq 2\delta.
\]
Using the triangle inequality shows
\[
\chi(\lambda_m, \tilde{\lambda}_n) \geq \chi(\lambda_m, \lambda_n) - \chi(\lambda_m, \tilde{\lambda}_m) - \chi(\lambda_n, \tilde{\lambda}_n) \geq 2\delta - \frac{2\epsilon}{\sigma_{\min}(A)\sigma_{\min}(B)}
\]
from which inequality (3.2) follows.

3.2. Distinct eigenvalue clusters for subpencils of a tensor. We now use Theorem 3.1 to give the radius of a Frobenius norm ball around a given tensor in which every tensor is guaranteed to have a subpencil with at least \( J \) distinct generalized eigenvalues.

Corollary 3.5. Let \( \mathcal{T} \in \mathbb{R}^{R \times R \times K} \) be a slice mix invertible tensor of real rank \( R \). Let \( U \in \mathbb{R}^{K \times K} \) be any orthogonal matrix. Set \( S = \mathcal{T} :_3 U \). For each \( k = 1, \ldots, [K/2] \), let \( \epsilon_k \geq 0 \) be the bound obtained from Theorem 3.1 which guarantees that matrix pencils in a ball of Frobenius radius \( \epsilon \) around the subpencil
\[
(S_{2k-1}, S_{2k})
\]
have at least \( J \) distinct generalized eigenvalues. Here we take \( \epsilon_k = 0 \) if the corresponding subpencil is not slice mix invertible. Set \( \epsilon = \|\epsilon_1, \ldots, \epsilon_{[K/2]}\|_2 \). Then every tensor in an open ball of Frobenius radius \( \epsilon \) around \( \mathcal{T} \) has a subpencil which has at least \( J \) distinct generalized eigenvalues.

Proof. Let \( Y \in \mathbb{R}^{R \times R \times K} \) be a tensor such that \( \|T - Y\|_F < \epsilon \) and set \( P = Y :_3 U \). Then we have \( \|S - P\|_F < \epsilon \). It follows that there must be some index \( k \in \{1, \ldots, [K/2]\} \) such that
\[
\| (S_{2k-1}, S_{2k}) - (P_{2k-1}, P_{2k}) \|_{sp} \leq \| (S_{2k-1}, S_{2k}) - (P_{2k-1}, P_{2k}) \|_F < \epsilon_k,
\]
hence Theorem 3.1 may be applied to the subpencil \( (P_{2k-1}, P_{2k}) \) of \( P \). In the above, the norm \( \| (M_1, M_2) \|_{sp} \) is the tensor spectral norm of the tensor \( M \in \mathbb{R}^{R \times R \times 2} \) whose frontal slices are \( M_1 \) and \( M_2 \). \( \quad \) \( \square \)

Theorem 3.1 and Corollary 3.5 are similar to results appearing in [13]; however, the results appearing here are more general and their proofs require additional consideration.

3.3. Computation of the bounds appearing in Theorem 3.1 and Corollary 3.5. We now explain how to compute the bounds presented in this section. We begin by showing how to compute a lower bound for the quantity \( \epsilon_2 \) in Theorem 3.1.

Lemma 3.6. Let \( \mathcal{T} \in \mathbb{R}^{R \times R \times K} \) and let \( \mathcal{S} \) be a tensor which is not slice mix invertible. Then
\[
\|T - S\|_{sp} \geq \max_{v \in \mathbb{R}^K} \sigma_{\min}(T :_3 v^T).
\]

Proof. Let \( v_\ast \in \mathbb{R}^K \) be a unit vector which maximizes the right hand side of equation (3.7). Since \( S \) is not slice mix invertible, the matrix \( S^{\cdot 3} v_T^T \in \mathbb{R}^{R \times R} \) has a nontrivial null space. It follows that there is a unit vector \( y_\ast \in \mathbb{R}^R \) such that \( S^{\cdot 2} y_T^T \cdot 3 v_T^T = 0 \). We then have
\[
\|T - S\|_{sp} \geq \|(T - S) \cdot 2 y_T^T \cdot 3 v_T^T\|_2 = \|(T \cdot 3 v_T^T) y_\ast\|_2 \geq \sigma_{\min}(T \cdot 3 v_T^T)
\]

as claimed.

While the quantity on the right hand side of inequality (3.7) is itself nontrivial to compute, one can for example use the lower bound
\[
\sigma_{\min}(T \cdot 3 v_T^T) \geq \max\{\sigma_{\min}(T_1), \ldots, \sigma_{\min}(T_k)\}
\]

We use this lower bound for \( \epsilon_2 \) in the numerical experiment presented in the upcoming Figure 8 and for almost all pencils observe that the quantity \( \epsilon_1 \) of Theorem 3.1 is less than or equal to this lower bound, hence the computed bound is rarely affected in our experiments by using this lower bound.

To compute the quantity \( \epsilon_1 \), one need only compute a CPD \( T = [A, B, C] \) of \( T \) where the columns of \( C \) are scaled to have unit norm. As shown in Theorem 3.1, the generalized eigenvalues of \( T \) are then equal to the spans of the columns of the factor matrix \( C \). A subtlety is that the quantity \( \sigma_{\min}(A)\sigma_{\min}(B) \) depends on the scaling of the columns of \( A \) and \( B \). We observe good performance if we scale these factor matrices so that for all \( r = 1, \ldots, R \) the norm of \( r \)-th column of \( A \) is equal to that of the \( r \)-th column of \( B \), and this is how we scale the factors in our experiments. Locally optimal scaling can be computed using an ALS approach for example. See the discussion in [13, Section 4.4.1] for more details.

For a given unitary \( U \), the bound presented in Corollary 3.5 may be computed by applying the preceding discussion to the appropriate subpencils of \( T^{\cdot 3} U \). Computing a unitary which gives the optimal bound in Corollary 3.5 is a nontrivial problem which is related to the travelling salesman problem. See the discussion in [13, Section 5.1].

In practice we randomly generate unitaries \( U_1, \ldots, U_L \in \mathbb{R}^{K \times K} \) for some chosen parameter \( L \), then keep the maximum bound obtained by applying Corollary 3.5 to the tensors \( T^{\cdot 3} U_1, \ldots, T^{\cdot 3} U_L \). The random unitaries we consider have the form \( Q \cdot \text{diag}(\text{sign}(R)) \) where \( Q \) and \( R \) are obtained from the QR decomposition applied to randomly generated matrices with i.i.d. Gaussian entries. Here \( \text{sign}(M) \) applies the sign operator to each entry of the matrix \( M \), while \( \text{diag}(M) \) is the diagonal matrix whose diagonal entries are equal to those of \( M \). Note that generating unitaries in this method is equivalent to randomly sampling unitaries from the Haar (uniform) measure on the unitary group [26].

4. GESD vs. GEVD accuracy and timing. We now compare both the accuracy and timing of GESD to GEVD where the GEVD solution is computed using the first two slices of the MLSVD of the tensor\(^\text{10}\). To this end, we generate low-rank tensors and add noise with varying signal to noise ratios (SNRs). We then compute the CPD with both methods and compare the computation time and factor matrix error (\text{cpderr}) of the results. For a low-rank tensor generated from the factor matrices \( A, B \) and \( C \), the factor matrix error of its estimated CPD \( \left[ \hat{A}, \hat{B}, \hat{C} \right] \) is defined as
\[
\max\left( \frac{\|A - \hat{A}\|_F}{\|A\|_F}, \frac{\|B - \hat{B}\|_F}{\|B\|_F}, \frac{\|C - \hat{C}\|_F}{\|C\|_F} \right),
\]
where columns of \( \hat{A}, \hat{B} \) and \( \hat{C} \) have been optimally
\(^{10}\)Similar to GESD, the average performance of GEVD benefits by using the first two slices of the MLSVD. However, this strategy is not always successful, see Section 4.4.
permuted and scaled to match the columns of $A, B$ and $C$, respectively. The SNR of the tensor $T$ and the noise $N$ in dB is defined as $20 \log_{10} \left( \frac{\|T\|_F}{\|N\|_F} \right)$. For each SNR, the median error and computation time are computed over 50 trials unless differently specified. The computation time is the time required to obtain the factor matrices $\hat{A}$, $\hat{B}$ and $\hat{C}$ starting from the noisy tensor $T + N$. All experiments in this section and Section 5 are performed on a machine with an Intel Core i7-6820HQ CPU at 2.70 GHz and 16GB of RAM using MATLAB R2020b and Tensorlab 3.0 [37].

We start with two easier problems where the tensor dimensions are considerably larger than the tensor rank. For a more challenging case, we then move on to tensors with factor matrices that have correlated columns. Next, we examine the accuracy and speed of the GESD method compared to a state-of-the-art optimization method.

We note that as a tuning parameter in GESD one may adjust the chordal distance between generalized eigenvalues which is sufficiently large for splitting between the corresponding generalized eigenspaces. To a point, the accuracy of GESD may be improved by increasing the requested chordal distance; however, increasing this chordal distance can decrease the speed of GESD. One could also choose to restrict the number of generalized eigenvalue clusters which GESD is allowed to separate in at each step. The optimal splitting strategy in terms of both timing and accuracy is a topic of further investigation. Furthermore, one could additionally consider the separation between computed generalized eigenspaces as a parameter for deciding if a particular split is acceptable. Unless otherwise stated, in the experiments in this section we use a splitting threshold of 0.2 chordal distance.

4.1. Tensor dimension exceeds tensor rank. The following experiments resemble typical problems in signal processing, where the tensor dimensions are often orders of magnitude larger than the tensor rank. For the first experiment, the factor matrix entries of a $100 \times 100 \times 100$-tensor of rank 10 are independently sampled from the standard normal distribution. The results are shown in Figure 1. In this setting, the GESD algorithm is marginally more accurate than GEVD, but the computational cost of GESD is about three times that of GEVD. While GESD is more computationally expensive, the first step in this setting is to compress the starting tensor. The cost of this compression dominates the other parts of the computation, thus using GESD over GEVD leads to about a 25 percent increase in the total run time of the algorithm.

For the second experiment, the factor matrices are independently sampled from the uniform distribution on $[0, 1]$ instead of the standard normal distribution. This setting is more challenging than the first since the columns of the factor matrices are less well-separated. In Figure 2, we can see the accuracies and computation times of both methods. Compared to the case with normally distributed factor matrix entries, a higher SNR is needed to obtain a reasonable solution, but from an SNR of 0 dB upwards, the error of the GESD method is more than an order of magnitude smaller than the error for the GEVD method. That is, using GESD leads to a gain of over 20 dB in accuracy. As in the previous experiment, the total cost of compression together with GESD is only slightly higher than compression with GEVD since compression is

---

11 An in-depth discussion of the sensitivity of generalized eigenspace computations to the separation between the corresponding generalized eigenspaces is given in [9]. While the bounds discussed therein take into account both eigenvalue and eigenspace conditioning, they are expensive to compute and lead to major increases in runtime if used in GESD. This is of course undesirable if the GESD algorithm is used to quickly obtain a good initialization for an optimization algorithm. [14] further explores this topic in the context of joint generalized eigenspaces.
the most computationally expensive step.

4.2. Factor matrices with correlated columns. We now consider the more challenging setting where the tensor of interest has rank equal to dimensions. That is, the tensors we consider in this section have rank $R$ and size $R \times R \times R$. This case is significantly more difficult than the $I_1 \times I_2 \times I_3$ rank $R$ case with $I_1, I_2, I_3 \gg R$ which generally occurs in practice and is discussed in Section 4.1. Recall that the first step in both GESD and GEVD is to compute a MLSVD of the tensor of interest. In the $I_1, I_2, I_3 \gg R$ case, computing this MLSVD and reducing to an $R \times R \times R$ tensor typically significantly increases the SNR of the problem thereby reducing the difficulty of the problem. In addition, the compression generally takes much more time than computing the CPD of the small core tensor using GEVD or GESD.

In Figure 3 we consider the particularly difficult case where the columns of the factor matrices are highly correlated. More concretely, we consider the case where the first column of $A$ is sampled from the uniform distribution on $[0, 1]$ as before and each successive column of $A$ is randomly sampled so that it forms a $10^\circ$ angle with $a_1$. The other factor matrices are sampled analogously. Results for this case are shown in Figure 3. We note that, due to the difficulty of this setting, very high SNR is required before either GESD or GEVD is able to compute a meaningful solution.

For the problems in Figure 3, we see that GESD is significantly more accurate than GEVD. In fact, GESD is able to compute meaningful solutions starting at about an order of magnitude lower SNR than GEVD. This demonstrates that the handling of pencil specific eigenvalue clusters is better in GESD than in GEVD due to the fact that GESD is able to compute an eigenspace for the cluster, but delays computation of eigenvectors to a pencil in which the previously clustered eigenvalues are better separated. As before, the GESD algorithm takes about three times as long as the GEVD algorithm.

In Figure 4, we investigate the speed and accuracy of both algorithms as a function of tensor rank. Here we run GESD and GEVD on noiseless tensors of rank $R$ and size $R \times R \times R$ where $R$ ranges from 50 to 350 and where the factor matrices are independently sampled from the uniform distribution on $[0, 1]$. For all values of $R$ we see that GESD is roughly two orders of magnitude more accurate than GEVD, but that GESD takes about three times as long to run as GEVD.

4.3. Comparison with optimization. As the GEVD algorithm can quickly compute a reasonably good fit for the CPD, its solution is often used as an initialization for an optimization algorithm. By providing a good initialization, the number of iterations of the more accurate yet more computationally expensive optimization algorithm can ideally be lowered significantly. In the next experiment, we compare the error and computation time of a nonlinear least-squares (NLS) algorithm [36] for three different initialization strategies: starting from the GEVD solution, the GESD solution or a random initial point. The optimization algorithm is run until either the relative cost function change or the relative factor matrix change drops under $10^{-8}$. The time to compute the GEVD or GESD solutions is included in the runtime of the optimization algorithm.

In Figure 5, median values are shown over 20 trials for a $10 \times 10 \times 10$-tensor of rank 10 with factor matrices independently sampled from the uniform distribution on $[0, 1]$. We can see that the NLS algorithm generally converges to a good solution for all three initialization strategies. The GEVD solution is notably less accurate than either a GESD or optimization based solution. In the right plot of Figure 5, it can be seen that the GESD method is about an order of magnitude faster than the
optimization algorithm. Also, the computation time of the optimization algorithm is significantly reduced by starting from the GEVD or GESD initialization, even when their computation time is included. For all SNRs, computation of a GESD or GEVD initialization followed by optimization is equally fast.

In Figure 6, median values are shown over 20 trials for a $10 \times 10 \times 10$-tensor of rank 10 where the columns of the factor matrices are sampled with a fixed $10^\circ$ angle between them. For this more challenging problem, we can see from the left plot of Figure 6 that decent algebraic initializations are only achieved at higher SNR values compared to the uncorrelated case. Additionally we see that the SNR required for GESD to compute reasonably accurate initializations is significantly lower than that for GEVD. In this experiment, random initialization of the optimization algorithm does not lead to a meaningful solution at all, which again highlights the importance of algebraic initialization. As before, optimization with a GESD initialization and optimization with a GEVD initialization typically take the same amount of time.

In Figure 7 we consider an even more difficult setting. Here median values are shown over 100 trials for a $20 \times 20 \times 20$-tensor of rank 20 where the columns of the factor matrices are sampled with a fixed $8^\circ$ angle between them. Due to the difficulty of this setting, meaningful solutions can only be computed with very high SNR. In the left plot of Figure 7 we see both that GESD computes a more accurate algebraic initialization than GEVD and, moreover, that using the algebraic initialization from GESD for optimization leads to a more accurate CPD than using the algebraic initialization from GEVD. In addition we observe that GESD+optimization is faster than GEVD+optimization. As in the previous figure, optimization with random initialization does not lead to a meaningful solution. We emphasize that in Figure 7, optimization following either GESD or GEVD ultimately stops due to insufficient advancement of the objective function. This illustrates that, in this problem, using GESD based initialization indeed leads to better local minima than GEVD based initialization.

These experiments illustrate that using GESD as an algebraic initialization compared to GEVD does not cause any loss in terms of time or accuracy. In fact, GESD+optimization can be both more accurate and faster than GEVD+optimization. Furthermore see we that GESD based initializations are significantly more accurate than GEVD based initializations. Thus using GESD over GEVD to initialize gives more reliable initializations without impacting total computation time.

4.4. GESD and GEVD vs. noiseless problems. We now illustrate an additional advantage of GESD compared to GEVD. Namely there is no fixed strategy for pencil selection in GEVD which is able to solve all problems meeting the assumptions of the algorithm even in the noiseless setting. As an example, let $\mathcal{T}$ be the (real) rank 3 tensor with frontal slices

$$
\mathbf{T}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}, \quad \mathbf{T}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/5 \\ 0 & 1/5 & 0 \end{pmatrix}.
$$

As previously discussed, a popular strategy for pencil selection in GEVD is to use the first two slices of a core for the MLSVD of $\mathcal{T}$. The tensor $\mathcal{S}$ with frontal slices

$$
\mathbf{S}_1 \approx -\begin{pmatrix} 1.39 & 0 & 0 \\ 0 & 1.11 & 0 \\ 0 & 0 & 1.11 \end{pmatrix}, \quad \mathbf{S}_2 \approx \begin{pmatrix} -0.25 & 0 & 0 \\ 0 & 0.16 & 0 \\ 0 & 0 & 0.16 \end{pmatrix}, \quad \mathbf{S}_3 \approx -\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1/5 \\ 0 & 1/5 & 0 \end{pmatrix}
$$
is a core for the MLSVD of $\mathcal{T}$. If one uses the first two slices of $\mathcal{S}$ to compute the CPD of $\mathcal{T}$ via GEVD, then one must compute the eigenvectors of the matrix

$$S_1^{-1}S_2 \approx \begin{pmatrix} 0.181 & 0 & 0 \\ 0 & 0.143 & 0 \\ 0 & 0 & 0.143 \end{pmatrix}.$$ 

Since span$\{e_2, e_3\}$ is an eigenspace of dimension 2 for this matrix corresponding to the eigenvalue $-0.143$, there is an ambiguity in which eigenvectors should be used. Tensorlab’s GEVD implementation, for example, assumes $e_2$ and $e_3$ are the correct eigenvectors to take in this subspace and proceeds from there. Instead considering the matrix pencil $(S_1, S_3)$ shows that $e_2 + e_3$ and $e_2 - e_3$ are the correct eigenvectors to take from the subspace span$\{e_2, e_3\}$.

To make matters worse for GEVD, if one proceeds under the assumption that $e_1$, $e_2$, and $e_3$ are the correct eigenvectors to take, then the GEVD based solution one arrives at is a local optima for the optimization based approach for computing a CPD. Thus using this GEVD based initialization followed by optimization cannot
For factor matrices with correlated columns, the GESD method performs significantly better than the GEVD method with respect to the factor accuracy. Median (over 50 trials) cpderr (left) and computation time (right) for rank 10 tensors of dimensions $10 \times 10 \times 10$ with factor matrices with correlated columns ($10^\circ$ angles). Because of the difficulty of the problem only very high SNR values lead to a decent solution. The computation time of the GESD method is about three times that of the GEVD method.

The relative time difference between the GESD and GEVD methods remains about the same as tensor rank increases. Median (over 20 trials) cpderr and computation time for exact rank $R$ tensors of dimensions $R \times R \times R$ with factor matrices sampled from the uniform distribution on $[0,1]$. The GESD method obtains factors with a higher accuracy than the GEVD method for all tensor dimensions. In this experiment we use a splitting threshold of $5/R$.

yield the true rank 3 decomposition of $T$.\textsuperscript{12}

GESD, on the other hand, is easily able to compute the decomposition of this tensor. Although GESD still uses the pencil $(S_1, S_2)$ on its first iteration, GESD recognizes that the eigenspace span$\{e_2, e_3\}$ cannot be decomposed with this pencil and delays the decomposition to a later pencil.

Examined in detail, from the pencil $(S_1, S_2)$, GESD computes

\[
Z_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad Z_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]

and decomposes $S = S \cdot_2 Z_1^T + S \cdot_2 Z_2^T$. The tensor $S \cdot_2 Z_1^T$ has rank-1 and GESD extracts one rank-1 component of $T$ from this tensor. $S \cdot_2 Z_2^T$ is a rank-2 tensor of size $3 \times 2 \times 3$. Letting $A$ denote the core of a $2 \times 2 \times 2$ MLSVD of $S \cdot_2 Z_2^T$, one has

\[
A_1 = \begin{pmatrix} 0 & 1.12 \\ 1.12 & 0 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1/5 & 0 \\ 0 & 1/5 \end{pmatrix}.
\]

\textsuperscript{12}The GEVD based decomposition has approximately 0.516 factor matrix error when compared to the correct decomposition.
Fig. 5. Initializing an optimization algorithm with the GESD or GEVD solution is faster than starting from a random initialization. Median (over 20 trials) \( \text{cpderr} \) (left) and computation time (right) for rank 10 tensors of dimensions \( 10 \times 10 \times 10 \) with factor matrices sampled from the uniform distribution on \([0, 1]\). All initializations lead to the same solution. Optimization after initializing with the more accurate GESD solution is equally fast as optimization after initializing with the GEVD solution.

Fig. 6. For factor matrices with correlated columns, the GESD solution is more accurate than the GEVD solution. An optimization algorithm initialized with the GESD solution is faster than when initialized with the GEVD solution. Median (over 20 trials) \( \text{cpderr} \) (left) and computation time (right) for rank 10 tensors of dimensions \( 10 \times 10 \times 10 \) with correlated factor matrix columns (\( 10^\circ \) angles). Initializing the optimization algorithm randomly does not lead to a good solution.

Fig. 7. For sufficiently difficult problems, initialization with GESD leads to both a more accurate solution and lower computation time of the optimization algorithm compared to initialization with GEVD. Median (over 100 trials) \( \text{cpderr} \) (left) and computation time (right) for rank-20 tensors of dimensions \( 20 \times 20 \times 20 \) with correlated factor matrix columns (\( 8^\circ \) angles). A splitting threshold of 0.02 was used in this experiment.
The generalized eigenvalues of the matrix pencil \((A_1, A_2)\) are \(\text{span}((1.12, 1/5))\) and \(\text{span}((-1.12, 1/5))\) with corresponding generalized eigenvectors \(e_1 + e_2\) and \(e_1 - e_2\), respectively. Since this matrix pencil has distinct generalized eigenvalues, GESD successfully extracts the remaining rank-1 components of \(T\) from the pencil.

5. Accuracy of GESD versus Corollary 3.5. In this section we compare the deterministic bound computed in Section 3 for the existence of two distinct eigenvalue clusters against the accuracy of the first step of our GESD algorithm. We note that the bound computed from the original tensor only provides information about the first step of our algorithm. For this reason, we omit a direct comparison of this bound to the accuracy of the GESD algorithm.

Let \(T \in \mathbb{R}^{R \times R \times K}\) be a real rank \(R\) tensor with CPD \([A, B, C]\), and let \(N\) be a noise tensor. As previously discussed, given \(T + N\) as an input, the goal of the first step of the GESD algorithm is to robustly decompose \(\mathbb{R}^R\) into (at least) two disjoint JGE spaces of \(T\).

Suppose \(Z_1\) and \(Z_2\) are matrices such that the range of each \(Z_i\) is a JGE space of \(T\) and such that the matrix \((Z_1, Z_2)\) is invertible. Recall that the \(Z_i\) have the form \(Z_i = B^{-T}E_i\) where \(E_1\) and \(E_2\) are matrices which for all \(j = 1, \ldots, R\) satisfy that the \(j\)th row of \(E_1\) is nonzero if and only if the \(j\)th row of \(E_2\) is a zero row. Using this fact we may evaluate the robustness of the first step of our algorithm in two ways.

Let \(\tilde{Z}_1\) and \(\tilde{Z}_2\) be the matrices computed by the first step of our algorithm. Then one should have that the ranges of

\[
B^T\tilde{Z}_1 = \tilde{E}_1 \quad \text{and} \quad B^T\tilde{Z}_2 = \tilde{E}_2,
\]

are (approximately) orthogonal. Additionally, there should exist disjoint index sets \(J_1, J_2 \subset \{1, \ldots, R\}\) such that \(J_1 \cup J_2 = \{1, \ldots, R\}\) and such that

\[
\text{ran} \; \tilde{Z}_1 \approx \text{span}([b_j]_{j \in J_1}) \quad \text{and} \quad \text{ran} \; \tilde{Z}_2 \approx \text{span}([b_j]_{j \in J_2})
\]

where the \(b_j\) are columns of the matrix \((B^{-T})\). That is, the principal (largest) angle between the range of \(\tilde{Z}_i\) and \(\text{span}([b_j]_{j \in J_i})\) should be approximately zero.

In Figure 8, we plot the median (over 50 trials) of the aforementioned angles as a function of the SNR of \(T\) to \(N\) where \(N\) is randomly generated Gaussian noise. That is, for many choices of the SNR \(s\), we randomly generate noise tensors \(N\) such that the SNR of \(T + N\) is equal to \(s\). In each trial we compute subspaces \(\tilde{Z}_1\) and \(\tilde{Z}_2\) for the tensor \(T + N\) using the GESD algorithm. We check accuracy by plotting the median over the trials of the angles described above. As before, \(T\) has been generated by independently sampling all entries of the factor matrices \(A, B,\) and \(C\) uniformly on \([0, 1]\) and forming the associated low-rank tensor.

The first image of Figure 8 shows the \(R \times R \times R\) rank \(R\) case with \(R = 4\), while the case \(R = 8\) is displayed in the second image. In addition to plotting the aforementioned angles, the bound from Corollary 3.5 for existence of two distinct eigenvalue clusters is plotted. The deterministic bound is computed by taking the largest bound found using either five or fifty different unitary matrices and the subpencil which gives the best bound is the subpencil used to compute generalized eigenspaces in the GESD algorithm.

We find that errors in GESD are well predicted by this bound. For SNR above the deterministic bound, the first step of GESD is highly accurate, while small errors start to appear when SNR decreases below the deterministic bound. For all SNR values, the inaccuracy in the orthogonality of the \(\tilde{E}_i\) and the inaccuracy in the angle between the \(\tilde{Z}_i\) and the columns of \(B^{-T}\) are observed to have similar magnitude.
6. Conclusion. We have presented an algebraic generalized eigenspace based algorithm for CPD computation which uses multiple subpencils of a given tensor to compute a decomposition. A key idea in the GESD algorithm is that one may combine information from many subpencils of a tensor to compute its CPD, thus allowing one to only perform trustworthy computations in each pencil. The performance of GESD has been compared to that of the popular GEVD algorithm. We find that in all situations, GESD is at least as accurate as GEVD, and that GESD is typically significantly more accurate than GEVD. We see that GESD particularly shines in difficult problems where the factor matrices of a tensor are highly correlated.

While the GESD algorithm typically takes about three times as long to run on a core tensor as GEVD, in practical settings which involve compression steps and optimization steps, a CPD computation using GESD for algebraic initialization of optimization is at least as fast as a CPD computation using GEVD for algebraic initialization. We observe that GESD based initializations are often significantly more accurate than GEVD based initializations and that they can lead to more accurate and reliable optimization. We also illustrate that no single fixed pencil selection strategy for GEVD can solve all given noiseless CPD problems even when the initial tensor
meets the assumptions of the GEVD algorithm. Since GESD intelligently determines which computations can be reliably performed with a given subpencil, GESD does not suffer from this issue.

In addition to experimental analysis, we provide a theoretical analysis of the GESD algorithm. Here we compute a deterministic bound for the existence of a subpencil which has two distinct clusters of generalized eigenvalues which have not coalesced. This bound may be interpreted as a theoretical guarantee that a next step in the GESD algorithm is able to compute generalized eigenspaces which are meaningfully related to the generalized eigenspaces of the original noiseless tensor.

Currently we assume the existence of reasonably sized eigenvalue gaps which may be used to compute at least two distinct generalized eigenspaces. While this assumption is reasonable for many tensors of low-rank, this assumption may fail to hold for particularly ill-conditioned tensors or when rank becomes large. Indeed, for generic tensors, the expected generalized eigenvalue gap becomes arbitrarily small as tensor rank and dimensions increase. Thus for tensors of sufficiently high rank, one should not expect to find a suitable gap. Resolving this is a direction for further research.

Acknowledgements. The authors thank Alexander Wagner, Ph.D for discussions which helped improve exposition in the article. We also thank the anonymous referees for their helpful comments.

REFERENCES

[1] C. Beltrán, P. Breiding, and N. Vannieuwenhoven, Pencil-based algorithms for tensor rank decomposition are not stable, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 739–773.
[2] P. Breiding and N. Vannieuwenhoven, The condition number of join decompositions, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 287–309.
[3] D. Camps, Pole Swapping Methods for the Eigenvalue Problem—Rational QR Algorithms, PhD thesis, KU Leuven, 2019.
[4] A. Cichoki, D. P. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. F. Caiafa, and A.-H. Phan, Tensor decompositions for signal processing applications: From two-way to multiway component analysis, IEEE Signal Process. Mag., 32 (2015), pp. 145–163.
[5] P. Comon and C. Jutten, Handbook of blind source separation: Independent component analysis and applications, Academic press, 2009.
[6] L. De Lathauwer, A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.
[7] L. De Lathauwer, B. De Moor, and J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.
[8] V. de Silva and L. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM J. Matrix Anal. & Appl., 30 (2008), pp. 1084–1127.
[9] J. Demmel and B. Kågström, Computing stable eigendecompositions of matrix pencils, Linear Algebra and its Applications, 88–89 (1987), pp. 139 – 186.
[10] I. Domanov and L. De Lathauwer, On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part I: Basic results and uniqueness of one factor matrix, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 855–875.
[11] I. Domanov and L. De Lathauwer, Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 636–660.
[12] I. Domanov and L. De Lathauwer, Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm, Linear Algebra and its Applications, 513 (2017), pp. 342–375.
[13] E. Evert and L. De Lathauwer, Guarantees for existence of a best canonical polyadic approximation of a noisy low-rank tensor, Technical Report 19–71, ESAT-STADIUS, KU Leuven, Leuven, Belgium, (2021). (Accepted for publication in SIAM J. Matrix Anal. Appl.).
[14] E. Evert and L. De Lathauwer, Perturbation theory for joint generalized eigenspaces, Tech.
RECURSIVE EIGENSPACE DECOMPOSITION FOR CPD

Report 20-82, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2021.

[15] N. M. Faber, L. M. C. Buydens, and G. Kateman, Generalized rank annihilation method. I: Derivation of eigenvalue problems, J. Chemometrics, 8 (1994), pp. 147–154.

[16] S. Friedland and L.-H. Lim, Nuclear norm of higher-order tensors, Math. Comp., 87 (2018), pp. 1255–1281.

[17] S. Friedland, V. Mehrmann, R. Pajarola, and S. Suter, On best rank one approximation of tensors, Numerical Linear Algebra with Applications, 20 (2013), pp. 942–955.

[18] S. Friedland and G. Ottaviani, The number of singular vector tuples and uniqueness of best rank-one approximation of tensors, Foundations of Computational Mathematics, 14 (2014), pp. 1209–1242.

[19] G. Golub and C. V. Loan, Matrix Computations, The Johns Hopkins University Press, 1996.

[20] R. A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[21] T. Kato, Perturbation Theory for Linear Operators, Springer, 1995.

[22] D. Kressner, Numerical Methods for General and Structured Eigenvalue Problems, Springer-Verlag Berlin Heidelberg, 2005.

[23] J. B. Kruskal, Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, Linear Algebra Appl., 18 (1977), pp. 95–138.

[24] J. Landsberg, Tensors: Geometry and applications, American Mathematical Society, 2012.

[25] S. E. Leurgans, R. T. Ross, and R. B. Abell, A decomposition for three-way arrays, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.

[26] F. Mezzadri, How to generate random matrices from the classical compact groups, Not. Am. Math. Soc, 54 (2007), pp. 592 – 604.

[27] Y. Nakatsukasa, Algorithms and Perturbation Theory for Matrix Eigenvalue Problems and the Singular Value Decomposition, PhD thesis, University of California at Davis, 2011.

[28] A. Phan, P. Tichavský, and A. Cichocki, Tensor deflation for CANDECOMP/PARAFAC—Part I: Alternating subspace update algorithm, IEEE Trans. Signal Process., 63 (2015), pp. 5924–5938.

[29] L. Qi, S. Hu, and Y. Xu, Spectral norm and nuclear norm of a third order tensor, Journal of Industrial & Management Optimization, (2021).

[30] K. Ye and L.-H. Lim, Schubert varieties and distances between subspaces of different dimensions, SIAM Journal on Matrix Analysis and Applications, 37 (2016), pp. 1176–1197.