SIAM J. Matrix Anal. Appl.
Vol. 26, No. 2, pp. 295–327

COMPUTATION OF THE CANONICAL DECOMPOSITION BY MEANS OF A SIMULTANEOUS GENERALIZED SCHUR DECOMPOSITION*

LIEVEN DE LATHAUWER[†], BART DE MOOR[‡], AND JOOS VANDEWALLE[‡]

Abstract. The canonical decomposition of higher-order tensors is a key tool in multilinear algebra. First we review the state of the art. Then we show that, under certain conditions, the problem can be rephrased as the simultaneous diagonalization, by equivalence or congruence, of a set of matrices. Necessary and sufficient conditions for the uniqueness of these simultaneous matrix decompositions are derived. In a next step, the problem can be translated into a simultaneous generalized Schur decomposition, with orthogonal unknowns [A.-J. van der Veen and A. Paulraj, *IEEE Trans. Signal Process.,* 44 (1996), pp. 1136–1155]. A first-order perturbation analysis of the simultaneous generalized Schur decomposition is carried out. We discuss some computational techniques (including a new Jacobi algorithm) and illustrate their behavior by means of a number of numerical experiments.

Key words. multilinear algebra, higher-order tensor, canonical decomposition, parallel factors analysis, generalized Schur decomposition

AMS subject classifications. 15A18, 15A69

DOI. 10.1137/S089547980139786X

1. Introduction.

An increasing number of signal processing problems involves the manipulation of quantities of which the elements are addressed by more than two indices. In the literature these higher-order equivalents of vectors (first order) and matrices (second order) are called higher-order tensors, multidimensional matrices, or multiway arrays. For a lot of applications involving higher-order tensors, the existing framework of vector and matrix algebra appears to be insufficient and/or inappropriate. The algebra of higher-order tensors is called multilinear algebra.

Rank-related issues in multilinear algebra are thoroughly different from their matrix counterparts. Let us first introduce some definitions. A rank-1 tensor is a tensor that consists of the outer product of a number of vectors. For an Nth-order tensor $\mathcal{A}$ and N vectors $U^{(1)}$, $U^{(2)}$, ..., $U^{(N)}$, this means that $a_{i_1 i_2 \ldots i_N} = u^{(1)}_{i_1} u^{(2)}_{i_2} \ldots u^{(N)}_{i_N}$ for all values of the indices, which will be concisely written as $\mathcal{A} = U^{(1)} \circ U^{(2)} \circ \cdots \circ U^{(N)}$. An n-mode vector of an $(I_1 \times I_2 \times \cdots \times I_N)$-tensor $\mathcal{A}$ is an I_n-dimensional vector obtained from $\mathcal{A}$ by varying the index i_n and keeping the other indices fixed. The n-rank of a higher-order tensor is the obvious generalization of the column (row) rank of matrices: it equals the dimension of the vector space spanned by the n-mode vec-

*Received by the editors November 12, 2001; accepted for publication (in revised form) by D. P. O'Leary, November 21, 2003; published electronically November 17, 2004. This research was supported by (1) the Flemish Government: (a) Research Council K.U.Leuven: GOA-MEFISTO-666, IDO, (b) the Fund for Scientific Research-Flanders (F.W.O.) projects G.0240.99, G.0115.01, G.0197.02, and G.0407.02, (c) the F.W.O. Research Communities ICCoS and ANMMM, (d) project BIL 98 with South Africa; (2) the Belgian State, Prime Minister's Office, Federal Office for Scientific, Technical and Cultural Affairs, Interuniversity Poles of Attraction Programme IUAP IV-02 and IUAP V-22.

http://www.siam.org/journals/simax/26-2/39786.html

[†]ETIS, UMR 8051, 6 avenue du Ponceau, BP 44, F 95014 Cergy-Pontoise Cedex, France (delathau@ensea.fr, http://www.etis.ensea.fr).

[‡]SCD-SISTA of the E.E. Dept. (ESAT) of the K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium (demoor@esat.kuleuven.ac.be, vdwalle@esat.kuleuven.ac.be, http://www.esat.kuleuven.ac.be/sista-cosic-docarch/).

tors. An important difference with the rank of matrices is that the different n-ranks of a higher-order tensor are not necessarily the same. The n-rank will be denoted as $\mathrm{rank_n}(\mathcal{A}) = R_n$. Even when all the n-ranks are the same, they can still be different from *the* rank of the tensor, denoted as $\mathrm{rank}(\mathcal{A}) = R$; $\mathcal{A}$ having rank R generally means that it can be decomposed in a sum of R, but not less than R, rank-1 terms; see, e.g., [34].

Example 1. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{111} = a_{112} = 1, \\ a_{221} = a_{222} = 2, \\ a_{211} = a_{121} = a_{212} = a_{122} = 0. \end{cases}$$

The 1-mode vectors are the columns of the matrix

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \end{pmatrix}.$$

Because of the symmetry, the set of 2-mode vectors is the same as the set of 1-mode vectors. The 3-mode vectors are the columns of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 2 \end{pmatrix}.$$

Hence, we have that $R_1 = R_2 = 2$ but $R_3 = 1$.

Example 2. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{211} = a_{121} = a_{112} = 1, \\ a_{111} = a_{222} = a_{122} = a_{212} = a_{221} = 0. \end{cases}$$

The 1-rank, 2-rank, and 3-rank are equal to 2. The rank, however, equals 3, since

$$\mathcal{A} = E_2 \circ E_1 \circ E_1 + E_1 \circ E_2 \circ E_1 + E_1 \circ E_1 \circ E_2,$$

in which

$$E_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

is a decomposition in a minimal linear combination of rank-1 tensors (a proof is given in [17]).

The scalar product $\langle \mathcal{A}, \mathcal{B} \rangle$ of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is defined in a straightforward way as $\langle \mathcal{A}, \mathcal{B} \rangle \overset{\mathrm{def}}{=} \sum_{i_1} \sum_{i_2} \ldots \sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$. The Frobenius-norm of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is then defined as $\|\mathcal{A}\| \overset{\mathrm{def}}{=} \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. Two tensors are called orthogonal when their scalar product is zero.

In [19] we discussed a possible multilinear generalization of the singular value decomposition (SVD). The different n-rank values can easily be read from this decomposition. In [20] we examined some techniques to compute the least-squares approximation of a given tensor by a tensor with prespecified n-ranks. On the other hand, in [19] we emphasized that the decomposition that was being studied, is not necessarily rank-revealing. This is a drawback of unitary (orthogonal) tensor decompositions in general. In this paper we will study the decomposition of a given tensor as a linear combination of a minimal number of possibly nonorthogonal, rank-1 terms. This type of decomposition is often called “canonical decomposition” (CANDECOMP) or “parallel factors” model (PARAFAC). It is a multilinear generalization of diagonalizing a matrix by an equivalence or congruence transformation. However, it has thoroughly different properties, e.g., as far as uniqueness is concerned.

Section 2 is a brief introduction to the subject, with a formal definition of the CANDECOMP-concept and an overview of the main current computational techniques. In this section we will also mark out the problem that we will consider in this paper (we will make some specific assumptions concerning the linear independence of the canonical components). In section 3 we discuss a preprocessing step that allows us to reduce the dimensionality of the problem. In section 4 we establish a computational link between the tensor decomposition and the simultaneous diagonalization of a set of matrices by equivalence or congruence; this problem might also be looked at as a simultaneous matrix eigenvalue decomposition (EVD). The fact that the CANDECOMP usually involves nonorthogonal factor matrices is numerically disadvantageous. By reformulating the problem as a simultaneous generalized Schur decomposition (SGSD), the unknowns are restricted to the manifold of orthogonal matrices in section 5. In section 6 we discuss the advantage of working via a simultaneous matrix decomposition as opposed to working via a single EVD; this section also contains a first-order perturbation analysis of the SGSD. Techniques for the actual computation of the SGSD are considered in section 7. In section 8 it is explained how the original CANDECOMP-components can be retrieved from the components of the SGSD. In section 9 the different techniques are illustrated by means of a number of numerical experiments.

This paper contains the following new contributions:

- In the literature one finds that, in theory, the CANDECOMP can be computed by means of a matrix EVD (under the uniqueness assumptions specified in section 2) [38, 43, 5, 42]. We show that one can actually interpret the tensor decomposition as a *simultaneous* matrix decomposition. The simultaneous matrix decomposition is numerically more robust than a single EVD.
- We show that the CANDECOMP can be reformulated as an *orthogonal* simultaneous matrix decomposition—the SGSD. The reformulation in terms of orthogonal unknowns allows for the application of typical numerical procedures that involve orthogonal matrices. The SGSD as such already appeared in [48]. The difference is that in this paper it is applied to unsymmetric, instead of symmetric, matrices. This generalization may raise some confusion. It might, for instance, be tempting to consider also a simultaneous lower triangularization, in addition to a simultaneous upper triangularization.
- We derive a Jacobi-algorithm for the computation of the SGSD. The formula for the determination of the rotation angle is an explicit solution for the case of rank-2 tensors.
- The way in which the canonical components are derived from the components of the SGSD is more general and more robust than the procedure proposed in [48].
- We derive necessary and sufficient conditions for the uniqueness of a number of simultaneous matrix decompositions: (1) simultaneous diagonalization by equivalence or congruence, (2) simultaneous EVD of nonsymmetric matrices, (3) simultaneous Schur decomposition (SSD).
- We conduct a first-order perturbation analysis of the SGSD.

Before starting with the next section, we add a comment on the notation that is used. To facilitate the distinction between scalars, vectors, matrices and higher-order tensors, the type of a given quantity will be reflected by its representation: scalars are denoted by lower-case letters (a, b, ...; α, β, ...), vectors are written as capitals (A, B, ...) (italic shaped), matrices correspond to bold-face capitals ($\mathbf{A}$, $\mathbf{B}$, ...) and tensors are written as calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...). This notation is consistently used for lower-order parts of a given structure. For instance, the entry with row index i and column index j in a matrix $\mathbf{A}$, i.e., $(\mathbf{A})_{ij}$, is symbolized by a_{ij} (also $(A)_i = a_i$ and $(\mathcal{A})_{i_1 i_2 \ldots i_N} = a_{i_1 i_2 \ldots i_N}$); furthermore, the ith column vector of a matrix $\mathbf{A}$ is denoted as A_i, i.e., $\mathbf{A} = [A_1 A_2 \ldots]$. To enhance the overall readability, we have made one exception to this rule: as we frequently use the characters i, j, r, and n in the meaning of indices (counters), I, J, R, and N will be reserved to denote the index upper bounds, unless stated otherwise.

2. The canonical decomposition.

The CANDECOMP or PARAFAC model is defined as follows.

DEFINITION 2.1 (CANDECOMP). *A canonical decomposition or parallel factors decomposition of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a decomposition of $\mathcal{A}$ as a linear combination of a minimal number of rank-*1 *terms*:

$$\mathcal{A} = \sum_{r}^{R} \lambda_r \, U_r^{(1)} \circ U_r^{(2)} \circ \cdots \circ U_r^{(N)}. \tag{2.1}$$

The decomposition is visualized for third-order tensors in Figure 2.1.

The terminology originates from psychometrics [10] and phonetics [26]. Later on, the decomposition model was also applied in chemometrics [1]. Recently, the decomposition drew the attention of researchers in signal processing [14, 16, 45, 46]. A good tutorial of the current state of the art in psychometrics and chemometrics is [3].

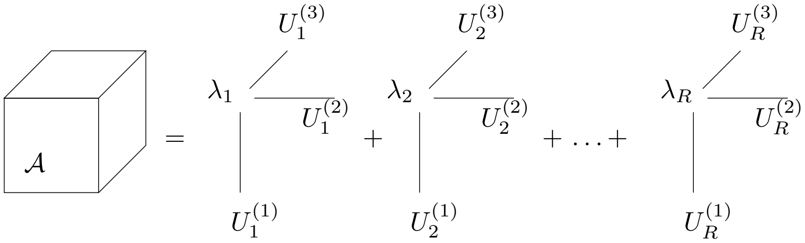

FIG. 2.1. *Visualization of the CANDECOMP for a third-order tensor.*

The decomposition can be considered as the tensorial generalization of the diagonalization of matrices by equivalence transformation (unsymmetric case) or by congruence transformation (symmetric case). However, its properties are thoroughly different from its second-order counterparts.

A first striking difference with the matrix case is that the rank of a real-valued tensor in the field of complex numbers is not necessarily equal to the rank of the same tensor in the field of real numbers [35]. Second, even if nonorthogonal rank-1 terms are allowed, the minimal number of terms is not bounded by $\min\{I_1, I_2, \ldots, I_N\}$ in general (cf. Example 2); it is usually larger and depends also on the tensor order. The determination of the maximal attainable rank value over the set of $(I_1 \times I_2 \times \cdots \times I_N)$-tensors is still an open problem in the literature. In [14] an overview of some partial results, obtained for super-symmetric tensors in the context of invariant theory, is given. (A real-valued tensor is called super-symmetric when it is invariant under arbitrary index permutations.) The paper includes a tensor-independent rank upper-bound, an algorithm to compute maximal generic ranks and a complete discussion of the case of super-symmetric $(2 \times 2 \times \cdots \times 2)$-tensors.

The uniqueness properties of the CANDECOMP are also very different from (and much more complicated than) their matrix equivalents. The theorems of [14] allow one to determine the dimensionality of the set of valid decompositions for generic super-symmetric tensors. The deepest result concerning uniqueness of the decomposition for third-order real-valued tensors is derived from a combinatorial algebraic perspective in [34]. The complex counterpart is concisely proved in [45]. The result is generalized to arbitrary tensor orders in [47]. In [6] complex fourth-order cumulant tensors are addressed. Here we will restrict ourselves to some remarks of a more general nature, that are of direct importance to this paper. From the CANDECOMP-definition it is clear that the decomposition is insensitive to

- a permutation of the rank-1 terms,
- a scaling of the vectors $U_r^{(n)}$, combined with the inverse scaling of the coefficients λ_r.

Apart from these trivial indeterminacies, uniqueness of the CANDECOMP has been established under mild conditions of linear independence (see further for a precise formulation of the conditions imposed in this paper). Contrarily, the decomposition of a matrix $\mathbf{A}$ in a sum of rank$(\mathbf{A})$ rank-1 terms is usually made unique by imposing stronger (e.g., orthogonality) constraints. In addition, for an essentially unique CANDECOMP the number of terms R can exceed $\min\{I_1, I_2, \ldots, I_N\}$.

Example 3. Consider the $(2 \times 2 \times 2)$-tensor $\mathcal{A}$ defined by

$$\begin{cases} a_{111} = a_{121} = -a_{212} = -a_{222} = 3, \\ a_{221} = a_{112} = -a_{211} = -a_{122} = 1. \end{cases}$$

The CANDECOMP of this tensor is given by

$$\mathcal{A} = X_1 \circ Y_1 \circ Z_1 + X_2 \circ Y_2 \circ Z_2, \tag{2.2}$$

in which

$$X_1 = Z_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad Z_1 = X_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad Y_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad Y_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Apart from the trivial indeterminacies described above, this decomposition is unique, as will become clear in section 4. The reason is that the matrices $\mathbf{X} = (X_1\, X_2)$, $\mathbf{Y} = (Y_1\, Y_2)$, and $\mathbf{Z} = (Z_1\, Z_2)$ are each nonsingular.

On the other hand, consider the first "matrix slice" of $\mathcal{A}$ (cf. Figure 4.1):

$$\mathbf{A}_1 = \begin{pmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ -1 & 1 \end{pmatrix}.$$

Due to (2.2), we have that

$$\mathbf{A}_1 = X_1 Y_1^T + X_2 Y_2^T = \mathbf{X} \cdot \mathbf{Y}^T,$$

but this decomposition is not unique. As a matter of fact, one can write

$$\mathbf{A}_1 = (\mathbf{X}\mathbf{F}) \cdot (\mathbf{Y}\mathbf{F}^{-T})^T = \tilde{\mathbf{X}} \cdot \tilde{\mathbf{Y}}^T,$$

for any nonsingular (2×2) matrix $\mathbf{F}$. One way to make this decomposition essentially unique, is to claim that the columns of $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ are orthogonal. The solution is then given by the SVD of $\mathbf{A}_1$.

It is a common practice to look for the CANDECOMP-components by straightforward minimization of the quadratic cost function

$$f(\hat{\mathcal{A}}) = \|\mathcal{A} - \hat{\mathcal{A}}\|^2 \tag{2.3}$$

over all rank-R tensors $\hat{\mathcal{A}}$, which we will parametrize as

$$\hat{\mathcal{A}} = \sum_r^R \hat{\lambda}_r \, \hat{U}_r^{(1)} \circ \hat{U}_r^{(2)} \circ \cdots \circ \hat{U}_r^{(N)}. \tag{2.4}$$

It is possible to resort to an alternating least-squares (ALS) algorithm, in which the vector estimates are updated mode per mode [10]. The idea is as follows. Let us define

$$\hat{\mathbf{U}}^{(n)} \overset{\text{def}}{=} [\hat{U}_1^{(n)} \hat{U}_2^{(n)} \ldots \hat{U}_R^{(n)}],$$
$$\hat{\mathbf{\Lambda}} \overset{\text{def}}{=} \operatorname{diag}\{(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_R)\},$$

in which diag$\{\cdot\}$ is a diagonal matrix, containing the entries of its argument on the diagonal. If we now imagine that the matrices $\hat{\mathbf{U}}^{(m)}$, $m \neq n$, are fixed, then (2.3) is merely a quadratic expression in the components of the matrix $\hat{\mathbf{U}}^{(n)} \cdot \hat{\mathbf{\Lambda}}$; the estimation of these components is a classical linear least-squares problem. An ALS iteration consists of repeating this procedure for different mode numbers: in each step the estimate of one of the matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}$ is optimized, while the other matrix estimates are kept constant. Overflow and underflow can be avoided by normalizing the estimates of the columns $U_r^{(n)}$ $(1 \leqslant r \leqslant R; 1 \leqslant n \leqslant N)$ to unit-length.

For $R = 1$, the ALS algorithm can be interpreted as a generalization of the power method for the computation of the best rank-1 approximation of a matrix [20]. For $R > 1$, however, the canonical components can in principle not be obtained by means of a deflation algorithm. The reason is that the stationary points of the higher-order power iteration generally do not correspond to one of the terms in (2.4), and that the residue is in general not of rank $R - 1$ [32]. This even holds when the rank-1 terms are mutually orthogonal [33]. Only when each of the matrices $\{\mathbf{U}^{(n)}\}$ is column-wise orthonormal, the deflation approach will work, but in this special case, the components can be obtained by means of a matrix SVD [19].

Because the cost function is monotonically decreasing, one expects that the ALS algorithm converges to a (local) minimum of $f(\hat{\mathcal{A}})$. If the CANDECOMP-model is only approximately valid, the risk of finding a spurious local optimum can be diminished by repeating the optimization for a number of randomly chosen initial values. The decision on whether the global optimum has been found or not usually relies on heuristics. The process of iterating over different starting values can be time-consuming. In addition, if the directions of some of the n-mode vectors in the CANDECOMP-model $(1 \leqslant n \leqslant N)$ are close, then it seems unlikely that this configuration is found from a random start [14]. Some alternative initializations are discussed in [11]. The rank itself is usually determined by repeating the procedure for different values of R, and comparing the results. An alternative, also based on heuristics, is the evaluation of split-half experiments [27].

ALS iterations can be very slow. In addition, it is sometimes observed that the algorithm moves through a "swamp": the algorithm seems to converge, but then the convergence speed drastically decreases and remains small for several iteration steps, after which it may suddenly increase again. The nature of swamps and how they can be avoided forms a topic of ongoing research [41, 36]. To cope with the slow convergence, a number of acceleration methods have been proposed [26, 28, 31]. One could make use of a prediction technique, in which estimates of previous iteration steps are extrapolated to forecast new estimates [3].

In [40] a Gauss–Newton method is described, in which all the CANDECOMP-factors are updated simultaneously; in addition, the inherent indeterminacy of the decomposition has been fixed by adding a quadratic regularization constraint on the component entries.

On the other hand, setting the gradient of f to zero and solving the resulting set of equations, is computationally hard as well: a set of $R(I_1 + I_2 + \dots I_N) - R(N-1)$ polynomial equations of degree $2N-1$, in $R(I_1 + I_2 + \dots I_N) - R(N-1)$ independent unknowns, has to be solved (to determine this dimensionality, imagine that the indeterminacy has been overcome by incorporating the factor λ_r $(1 \leqslant r \leqslant R)$ in one of the vectors of the rth outer product, and by fixing one nonzero entry in the other vectors).

An interesting alternative procedure, which works under a number of assumptions among which the most restrictive is that $R \leqslant \min\{I_1, I_2\}$, has been proposed in [38]. Similar results have been proposed in [43, 5, 42]. It was explained that, if (2.1) is exactly valid, the decomposition can be found by a simple matrix EVD. When $\mathcal{A}$ is only known with limited accuracy, a least-squares matching of both sides of (2.1) can now be initialized with the EVD result. This technique forms the starting point for the developments in section 4.

Some promising computation schemes, at this moment only formulated in terms of (super-symmetric) cumulant tensors, have been developed as means to solve the problem of higher-order-only independent component analysis. In [7] Cardoso shows that under mild conditions the matrices in the intersection of the range of the cumulant tensor and the manifold of rank-1 matrices take the form of an outer product of a steering vector with itself; consequently MUSIC-like [44] algorithms are devised. In [6] the same author investigates the link between symmetry of the cumulant tensor and the rank-1 property of its components. The problem is subsequently reformulated in terms of a matrix EVD.

The decomposition of a dataset as a sum of rank-1 terms is sometimes called the *factor analysis* problem. With the decomposition, one aims at relating the different rank-1 terms to the different "physical mechanisms" that have contributed to the dataset. We repeat that factor analysis of matrices is, as such, essentially underdetermined. The extra conditions (maximal variance, orthonormality, etc.) that are usually imposed to guarantee uniqueness, are often physically irrelevant. In a wide range of parameters, this is not the case for the higher-order decomposition; the weaker conditions of linear independence to ensure uniqueness often have a physical meaning. This makes the CANDECOMP of higher-order tensors to an important signal processing tool.

In this paper, we will study the special but important case of an $(I_1 \times I_2 \times I_3)$-tensor $\mathcal{A}$ with rank $R \leqslant \min\{I_1, I_2\}$ and 3-rank $R_3 \geqslant 2$. (If $R_3 = 1$, then the different matrices obtained from $\mathcal{A}$ by fixing the index i_3 are proportional, and the CANDECOMP reduces to the diagonalization of one of these matrices by congruence or equivalence.) We assume that

(i) the set $\{U_r^{(1)}\}_{(1 \leqslant r \leqslant R)}$ is linearly independent (i.e., no vector can be written as a linear combination of the other vectors),

(ii) the set $\{U_r^{(2)}\}_{(1 \leqslant r \leqslant R)}$ is linearly independent,

(iii) the set $\{U_r^{(3)}\}_{(1 \leqslant r \leqslant R)}$ does not contain collinear vectors (i.e., no vector is a scalar multiple of an other vector).

Roughly speaking, we address the case in which the number of rank-1 terms is bounded by the second largest dimension of $\mathcal{A}$ (like in classical matrix decompositions). Conditions (i)–(iii) are generically satisfied, i.e., only in a set of Lebesgue measure zero they do not hold. In typical applications one has the prior knowledge that these assumptions are valid. Classical (not overcomplete) independent component analysis can be formulated in terms of this model [13, 49]. Conditions (i)–(iii) are required for the uniqueness of the solution. All the examples in the tutorial [3] belong to our class of interest. In chemometrical applications such as the ones described in [42], the conditions do not pose any problem. For instance, I_1 and I_2 correspond to the length of emission-excitation spectra and R is the number of chemical components.

If the rank of $\mathcal{A}$ is higher than $R_{\max} = \min\{I_1, I_2\}$, then our method will still try to fit a rank-$R_{\max}$ model to the data. Contrary to the matrix case, this does not simply correspond to discarding the rank-1 terms that have the smallest norm.

It can be verified that conditions (i)–(iii) are sufficient to make the CANDECOMP essentially unique [38] (see also sections 4 and 6). The exposition is restricted to real-valued third-order tensors for notational convenience. The generalization to higher tensor orders is straightforward. The method then applies to tensors of which the rank $R \leqslant \min\{I_1, I_2\}$ and at least one of the n-ranks R_n, for $n \geqslant 3$, satisfies $R_n \geqslant 2$. Conditions (i)–(iii) should be rephrased as the following:

(i) the set $\{U_r^{(1)}\}_{(1 \leqslant r \leqslant R)}$ is linearly independent,

(ii) the set $\{U_r^{(2)}\}_{(1 \leqslant r \leqslant R)}$ is linearly independent,

(iii) and at least one of the sets $\{U_r^{(n)}\}_{(1 \leqslant r \leqslant R)}$ for $n \geqslant 3$ does not contain collinear vectors.

Apart from section 7.2, the generalization to complex-valued tensors is also straightforward. An outline of the exposition is presented as Algorithm 1. In this algorithm we assume that a value of R is given or that the rank has been estimated as $\text{rank}_1(\mathcal{A}) = \text{rank}_2(\mathcal{A})$ (see next section).

3. Dimensionality reduction.

Under the assumptions specified in the previous section, we have that $R_1 = \text{rank}_1(\mathcal{A}) = R = \text{rank}_2(\mathcal{A}) = R_2$ and that $R_3 = \text{rank}_3(\mathcal{A}) = \text{rank}(\mathbf{U}^{(3)})$. To understand this, remark that (2.1) implies that the n-mode vectors of $\mathcal{A}$ are the columns of the matrix

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{\Lambda} \cdot (\mathbf{U}^{(m)} \odot \mathbf{U}^{(l)})^T,$$

in which $\odot$ is the Kathri–Rao or columnwise Kronecker product and (n, m, l) is an arbitrary permutation of $(1, 2, 3)$. Hence, conditions (i)–(iii) imply that the dimension of the n-mode vector space, which equals the rank of $\mathbf{A}_{(n)}$, is equal to $\text{rank}(\mathbf{U}^{(n)})$.

If $R < \max\{I_1, I_2\}$, or $R_3 < I_3$, then an a priori dimensionality reduction of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ to a tensor $\mathcal{B} \in \mathbb{R}^{R \times R \times R_3}$ decreases the computational load of the actual determination of the CANDECOMP (step 1 in Algorithm 1). Before starting the actual exposition, we briefly address this issue. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are related

```
Algorithm 1
CANDECOMP by SGSD
In: $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $R$.
Out: $\{U_r^{(1)}\}_{(1 \leqslant r \leqslant R)}$, $\{U_r^{(2)}\}_{(1 \leqslant r \leqslant R)}$, $\{U_r^{(3)}\}_{(1 \leqslant r \leqslant R)}$, $\{\lambda_r\}_{(1 \leqslant r \leqslant R)}$ such that $\mathcal{A} \simeq \sum_r^R \lambda_r \, U_r^{(1)} \circ U_r^{(2)} \circ U_r^{(3)}$.
    (1. Perform an initial best rank-$(R, R, R_3)$ approximation of $\mathcal{A}$: maximize $g(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}) = \|\mathcal{A} \times_1 \mathbf{X}^{(1)^T} \times_2 \mathbf{X}^{(2)^T} \times_3 \mathbf{X}^{(3)^T}\|^2$ over columnwise orthonormal $\mathbf{X}^{(1)} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{I_2 \times R}$ and $\mathbf{X}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$; $\max(g(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)})) = g(\mathbf{X}^{(1)}_{\max}, \mathbf{X}^{(2)}_{\max}, \mathbf{X}^{(3)}_{\max})$. $\mathcal{B} = \mathcal{A} \times_1 \mathbf{X}^{(1)^T}_{\max} \times_2 \mathbf{X}^{(2)^T}_{\max} \times_3 \mathbf{X}^{(3)^T}_{\max}$. Continue for $\mathcal{B}$ with steps 2, 3, 4, (5) below. $\hat{\mathcal{A}} = \hat{\mathcal{B}} \times_1 \mathbf{X}^{(1)}_{\max} \times_2 \mathbf{X}^{(2)}_{\max} \times_3 \mathbf{X}^{(3)}_{\max}$. (section 3.) (Perform step 5 for $\hat{\mathcal{A}}$.))
    2. Associate to $\mathcal{A}$ a linear mapping $f_{\mathcal{A}}$ from $\mathbb{R}^{I_3}$ to $\mathbb{R}^{I_1 \times I_2}$ (see (4.1)). Determine $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ such that the range of $f_{\mathcal{A}}$ is spanned by $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$.
    3. Compute orthogonal $\mathbf{Q}, \mathbf{Z}$ and (approximately) upper triangular $\{\mathbf{R}_k\}_{(1 \leqslant k \leqslant K)}$ from the SGSD of (5.1)–(5.3):
        - extended $QZ$-iteration (section 7.1, [48]), or
        - Jacobi-type iteration (section 7.2, [17, 18]).
    4. Compute $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ from $\{\mathbf{R}_k\}_{(1 \leqslant k \leqslant K)}$ and $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ (and $\mathbf{Q}$ , $\mathbf{Z}$). Compute $\mathbf{U}^{(3)}$ from $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and $\mathcal{A}$. (Detailed outline in section 8.)
    (5. Minimize $f(\hat{\mathcal{A}}) = \|\mathcal{A} - \hat{\mathcal{A}}\|^2$ (section 2).)
```

by

$$a_{i_1 i_2 i_3} = \sum_{r_1 r_2 r_3} x^{(1)}_{i_1 r_1} x^{(2)}_{i_2 r_2} x^{(3)}_{i_3 r_3} b_{r_1 r_2 r_3} \tag{3.1}$$

for all index values, where $\mathbf{X}^{(1)} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{I_2 \times R}$ and $\mathbf{X}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$, which we will write concisely as

$$\mathcal{A} = \mathcal{B} \times_1 \mathbf{X}^{(1)} \times_2 \mathbf{X}^{(2)} \times_3 \mathbf{X}^{(3)}. \tag{3.2}$$

If $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$ each have mutually orthonormal columns, then the optimal rank-R approximation $\hat{\mathcal{B}}$ of $\mathcal{B}$ and the optimal rank-R approximation $\hat{\mathcal{A}}$ of $\mathcal{A}$ are related in the same way:

$$\hat{\mathcal{A}} = \hat{\mathcal{B}} \times_1 \mathbf{X}^{(1)} \times_2 \mathbf{X}^{(2)} \times_3 \mathbf{X}^{(3)}, \tag{3.3}$$

since "n-mode multiplication" with the columnwise orthonormal matrices $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$ does not change the cost function f (2.3). If the CANDECOMP-model is exactly satisfied, then any orthonormal basis of the mode-1, mode-2, and mode-3 vectors of $\mathcal{A}$ gives a suitable $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\mathbf{X}^{(3)}$, respectively. In practice, however, $R = R_1 = R_2$ and R_3 will be estimated as the number of significant mode-1 / mode-2 and mode-3 singular values of $\mathcal{A}$ (see [19]). An optimal rank-(R, R, R_3) approximation of $\mathcal{A}$, before computing the optimal rank-R approximation, can then be realized. For techniques we refer to [20].

4. CANDECOMP and simultaneous EVD. Without loss of generality we assume that $I_1 = I_2 = R$ (if $I_1 > R$ or $I_2 > R$, we can always do a dimensionality reduction, as explained in the previous section). We start the derivation of our computation scheme with associating to $\mathcal{A}$ a linear transformation of the vector space $\mathbb{R}^{I_3}$ to the matrix space $\mathbb{R}^{I_1 \times I_2}$, in the following way:

$$\mathbf{V} = f_{\mathcal{A}}(W) = \mathcal{A} \times_3 W \qquad \Longleftrightarrow \qquad v_{i_1 i_2} = \sum_{i_3} a_{i_1 i_2 i_3} w_{i_3}, \tag{4.1}$$

for all index values. Substitution of (4.1) in (2.1) shows that the image of W can easily be expressed in terms of the CANDECOMP-components:

$$\mathbf{V} = \mathbf{U}^{(1)} \cdot \mathbf{D} \cdot \mathbf{U}^{(2)^T}, \tag{4.2}$$

in which we have used the following notations:

$$\mathbf{U}^{(n)} \overset{\text{def}}{=} [U_1^{(n)} U_2^{(n)} \dots U_{I_n}^{(n)}], \tag{4.3}$$

$$\mathbf{D} \overset{\text{def}}{=} \operatorname{diag}\{(\lambda_1, \lambda_2, \dots, \lambda_R)\} \cdot \operatorname{diag}\{\mathbf{U}^{(3)^T} W\}. \tag{4.4}$$

Any matrix in the range of the mapping $f_{\mathcal{A}}$ can be diagonalized by equivalence with the matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. (If $\mathcal{A}$ does not change under permutation of its first two indices, then any matrix in the range can be diagonalized by congruence with the matrix $\mathbf{U}^{(1)} = \mathbf{U}^{(2)}$.) If the range is spanned by the matrices $\mathbf{V}_1$, $\mathbf{V}_2$, …, $\mathbf{V}_K$, then we should solve the following simultaneous decomposition:

$$\mathbf{V}_1 = \mathbf{U}^{(1)} \cdot \mathbf{D}_1 \cdot \mathbf{U}^{(2)^T}, \tag{4.5}$$

$$\mathbf{V}_2 = \mathbf{U}^{(1)} \cdot \mathbf{D}_2 \cdot \mathbf{U}^{(2)^T}, \tag{4.6}$$

$$\vdots$$

$$\mathbf{V}_K = \mathbf{U}^{(1)} \cdot \mathbf{D}_K \cdot \mathbf{U}^{(2)^T}, \tag{4.7}$$

in which $\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K$ are diagonal. A possible choice of $\{\mathbf{V}_k\}_{(1 \leqslant k \leqslant K)}$ consists of the "matrix slices" $\{\mathbf{A}_i\}_{(1 \leqslant i \leqslant I_3)}$, obtained by fixing the index i_3 to i (see Figure 4.1); the corresponding vectors $\{W_i\}_{(1 \leqslant i \leqslant I_3)}$ are the canonical unit vectors. An other possible choice consists of the K dominant left singular matrices of the mapping in (4.1). In both cases, the cost function

$$\tilde{f}(\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \{\hat{\mathbf{D}}_k\}) = \sum_k \|\mathbf{V}_k - \hat{\mathbf{U}}^{(1)} \cdot \hat{\mathbf{D}}_k \cdot \hat{\mathbf{U}}^{(2)^T}\|^2$$

corresponds to the CANDECOMP cost function (2.3). The latter choice follows naturally from the analysis in section 3 [20].

For later use, we define

$$\tilde{\mathbf{U}}^{(3)} = \begin{pmatrix} (\mathbf{D}_1)_{11} & (\mathbf{D}_1)_{22} & \dots & (\mathbf{D}_1)_{RR} \\ (\mathbf{D}_2)_{11} & (\mathbf{D}_2)_{22} & \dots & (\mathbf{D}_2)_{RR} \\ \vdots & \vdots & & \vdots \\ (\mathbf{D}_K)_{11} & (\mathbf{D}_K)_{22} & \dots & (\mathbf{D}_K)_{RR} \end{pmatrix} \tag{4.8}$$

$$= [W_1 W_2 \dots W_K]^T \cdot \mathbf{U}^{(3)} \cdot \operatorname{diag}\{(\lambda_1, \lambda_2, \dots, \lambda_R)\}. \tag{4.9}$$

If the CANDECOMP-model is exactly satisfied, then its terms can be computed from two of the equations in (4.5)–(4.7). Let us assume that the matrix $\mathbf{V}_1$ has

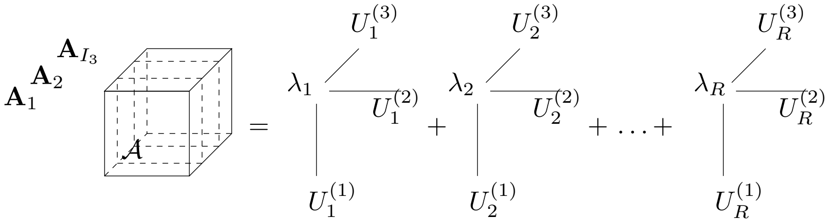

FIG. 4.1. *Definition of matrix slices for the computation of the CANDECOMP by simultaneous diagonalization.*

full rank (this is the case for a generic choice of W_1). Combination of the first two equations then leads to the following EVD:

$$\mathbf{V}_2 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}) \cdot {\mathbf{U}^{(1)}}^{-1}. \tag{4.10}$$

Remember that we assumed in section 2 that $\mathbf{U}^{(3)}$ does not contain collinear columns. As a consequence, the pair $((\mathbf{D}_1)_{ii}(\mathbf{D}_2)_{ii}) = \lambda_i(W_1^T U_i^{(3)} W_2^T U_i^{(3)})$ and $((\mathbf{D}_1)_{jj}(\mathbf{D}_2)_{jj}) = \lambda_j(W_1^T U_j^{(3)} W_2^T U_j^{(3)})$ is generically not proportional, for all $i \neq j$. Hence the diagonal elements of $\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}$ are mutually different and the EVD (4.10) reveals the columns of $\mathbf{U}^{(1)}$, up to an irrelevant scaling and/or permutation. Once $\mathbf{U}^{(1)}$ is known, $\mathbf{U}^{(2)}$ can be obtained, up to a scaling of its columns, as follows. From (4.5)–(4.7) we have

$$\mathbf{V}_1^T \cdot {\mathbf{U}^{(1)}}^{-T} = \mathbf{U}^{(2)} \cdot \mathbf{D}_1, \tag{4.11}$$

$$\mathbf{V}_2^T \cdot {\mathbf{U}^{(1)}}^{-T} = \mathbf{U}^{(2)} \cdot \mathbf{D}_2, \tag{4.12}$$

$$\vdots$$

$$\mathbf{V}_K^T \cdot {\mathbf{U}^{(1)}}^{-T} = \mathbf{U}^{(2)} \cdot \mathbf{D}_K. \tag{4.13}$$

Hence, if we denote the rth column of $\mathbf{V}_k^T \cdot {\mathbf{U}^{(1)}}^{-T}$ as B_{kr}, then $U_r^{(2)}$ can be estimated as the dominant left singular vector of $[B_{1r} B_{2r} \ldots B_{Kr}]$. Finally, the matrix $\mathbf{U}^{(3)} \cdot \text{diag}\{(\lambda_1, \lambda_2, \ldots, \lambda_R)\}$ is found by solving the CANDECOMP-model as a linear set of equations, for given matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. (Note that the assumptions that we have made for identifiability in section 2 indeed allow to obtain the CANDECOMP in an essentially unique way.) If the CANDECOMP-model is only approximately satisfied, then the estimates can be used to initialize an additional optimization algorithm for the minimization of cost function (2.3) (cf. step 5 in Algorithm 1). This EVD-approach is a variant of the techniques described in [38, 43, 5, 42].

It is intuitively clear, however, that it is preferable to exploit all the available information by taking into account all the equations in (4.5)–(4.7). This leads to a *simultaneous EVD*:

$$\mathbf{V}_2 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_2 \cdot \mathbf{D}_1^{-1}) \cdot {\mathbf{U}^{(1)}}^{-1}, \tag{4.14}$$

$$\mathbf{V}_3 \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_3 \cdot \mathbf{D}_1^{-1}) \cdot {\mathbf{U}^{(1)}}^{-1}, \tag{4.15}$$

$$\vdots$$

$$\mathbf{V}_K \cdot \mathbf{V}_1^{-1} = \mathbf{U}^{(1)} \cdot (\mathbf{D}_K \cdot \mathbf{D}_1^{-1}) \cdot {\mathbf{U}^{(1)}}^{-1}. \tag{4.16}$$

We will further discuss the advantages in section 6.

In this paper, we propose a reliable technique to deal with (4.5)–(4.7) simultaneously (steps 2–4 in Algorithm 1).

5. CANDECOMP and SGSD. The fact that the unknown matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are basically arbitrary nonsingular matrices, makes them hard to deal with in a proper numerical way. In this section, we will reformulate the problem in terms of orthogonal unknowns. Therefore, we can make an appeal to the technique established in [48], where the symmetric equivalent of (4.5)–(4.7) was encountered in the derivation of an analytical constant modulus algorithm.

Introducing a QR-factorization $\mathbf{U}^{(1)} = \mathbf{Q}^T \mathbf{R}'$ and an RQ-decomposition $\mathbf{U}^{(2)^T} = \mathbf{R}'' \mathbf{Z}^T$ leads to a set of matrix equations that we will call a *simultaneous generalized Schur decomposition* (a set of two of the equations below is called "Generalized Schur Decomposition" [24]):

$$\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z} = \mathbf{R}_1 = \mathbf{R}' \cdot \mathbf{D}_1 \cdot \mathbf{R}'', \tag{5.1}$$

$$\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z} = \mathbf{R}_2 = \mathbf{R}' \cdot \mathbf{D}_2 \cdot \mathbf{R}'', \tag{5.2}$$

$$\vdots$$

$$\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z} = \mathbf{R}_K = \mathbf{R}' \cdot \mathbf{D}_K \cdot \mathbf{R}'', \tag{5.3}$$

in which $\mathbf{Q}, \mathbf{Z} \in \mathbb{R}^{R \times R}$ are orthogonal and $\mathbf{R}', \mathbf{R}'', \mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K \in \mathbb{R}^{R \times R}$ are upper triangular. If the CANDECOMP model is exactly satisfied, the new problem consists of the determination of $\mathbf{Q}$ and $\mathbf{Z}$ such that $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ are each upper triangular. In practice, this is only possible in an approximate sense. For instance, one could maximize the function g, given by

$$g(\mathbf{Q}, \mathbf{Z}) = \|\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z}\|^2_{UF} + \|\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z}\|^2_{UF} + \cdots + \|\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z}\|^2_{UF}, \tag{5.4}$$

in which $\| \cdot \|_{UF}$ denotes the Frobenius-norm of the upper triangular part of a matrix. So we will determine $\mathbf{Q}$ and $\mathbf{Z}$ as the orthogonal matrices that make $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ simultaneously as upper triangular as possible. Equivalently, one may minimize

$$h(\mathbf{Q}, \mathbf{Z}) = \|\mathbf{Q} \cdot \mathbf{V}_1 \cdot \mathbf{Z}\|^2_{LFs} + \|\mathbf{Q} \cdot \mathbf{V}_2 \cdot \mathbf{Z}\|^2_{LFs} + \cdots + \|\mathbf{Q} \cdot \mathbf{V}_K \cdot \mathbf{Z}\|^2_{LFs} \tag{5.5}$$

$$= \sum_k \|\mathbf{V}_k\|^2 - g(\mathbf{Q}, \mathbf{Z}), \tag{5.6}$$

in which $\| \cdot \|_{LFs}$ denotes the Frobenius-norm of the strictly lower triangular part of a matrix. The decomposition is visualized in Figure 5.1.

In section 7 we will discuss two algorithms for the computation of the SGSD. In section 8 we will explain how $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be calculated once $\mathbf{Q}$ and $\mathbf{Z}$ have been estimated.

Remark 4. At first sight the unsymmetric case allows for the derivation of an additional set of equations if we substitute a QL-factorization $\mathbf{U}^{(1)} = \tilde{\mathbf{Q}}^T \mathbf{L}'$ and an LQ-decomposition $\mathbf{U}^{(2)^T} = \mathbf{L}'' \tilde{\mathbf{Z}}^T$ in (4.5)–(4.7) ($\mathbf{L}'$ and $\mathbf{L}''$ are lower triangular). This leads to a simultaneous lower triangularization of the matrices $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$. Both approaches are in fact equivalent because they simply correspond to a different permutation of the columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$, which cannot be determined in advance. Since the aim of the algorithms that will be discussed in section 7 is only to find matrices $\mathbf{Q}$ and $\mathbf{Z}$ that correspond to an arbitrary column permutation (not necessarily the one that happens to globally minimize the cost function h in the presence of noise), both formulations may in practice lead to results that are close but not exactly equal.

Remark 5. In [49] an alternative scheme, in which one directly works with the components of (4.5)–(4.7), instead of going via a SGSD, was formulated for the symmetric case, i.e., $\mathbf{U}^{(1)} = \mathbf{U}^{(2)} = \mathbf{U}$. Before continuing with the actual exposition, let

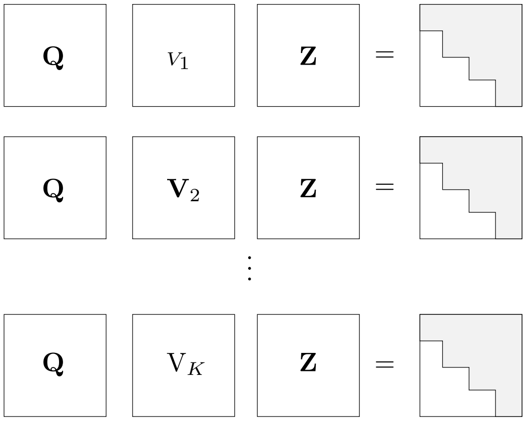

FIG. 5.1. *Visualization of a SGSD.*

us briefly address this approach. It is an ALS strategy, with the particular problem that for two of the modes the components are equal. The technique is called the "AC–DC" algorithm, standing for "alternating columns–diagonal centers". Let us associate with (4.5)–(4.7) the following weighted cost function:

$$c(\mathbf{U}, \mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K) = \sum_{k=1}^{K} w_k \|\mathbf{V}_k - \mathbf{U} \cdot \mathbf{D}_k \cdot \mathbf{U}^T\|^2. \tag{5.7}$$

Note that for $w_k = 1$ $(1 \leqslant k \leqslant K)$ and $\{\mathbf{V}_k\}_{(1\leqslant k\leqslant K)}$ equal to the matrix slices $\{\mathbf{A}_i\}_{(1\leqslant i\leqslant I_3)}$ defined in Figure 4.1, this cost function corresponds to the obvious CANDECOMP cost (2.3). In the technique one alternates between updates of $\{\mathbf{D}_k\}_{(1\leqslant k\leqslant K)}$, given $\mathbf{U}$ (DC-phase) and updates of $\mathbf{U}$, given $\{\mathbf{D}_k\}_{(1\leqslant k\leqslant K)}$ (AC-phase). It is clear that a DC-step amounts to a linear least-squares problem. In [49] it is shown that the conditional update of a column of $\mathbf{U}$ amounts to the best rank-1 approximation of a symmetric $(I \times I)$-matrix $(I = I_1 = I_2)$. An AC-phase then consists of one, or more, updates of the different columns of $\mathbf{U}$.

6. Single vs. simultaneous decomposition and perturbation analysis.

Before introducing some algorithms for the computation of the SGSD, we will discuss in this section some advantages of the simultaneous decomposition approach over the computation of a single EVD (cf. [38, 43, 5, 42]). In this context, we will also provide a first-order perturbation analysis of the SGSD.

6.1. Uniqueness.

First, let us reconsider (4.14)–(4.16). One could solve these EVDs separately, and retain the solution that leads to the best CANDECOMP-estimate. However, it is safer from a numerical point of view to solve (4.14)–(4.16) simultaneously, in some optimal sense, especially when the perturbation of the matrices $\{\mathbf{V}_k\}_{(1\leqslant k\leqslant K)}$ (with respect to their ideal values in an exact CANDECOMP) may have caused eigenvalues to cross each other. This is illustrated in the next example; a symmetric version of the example can be found in [4].

Example 6. Consider the following matrix pair:

$$\mathbf{M}_1 = \begin{pmatrix} 1-\epsilon & 0 & 0 & 0 \\ 0 & 1+\epsilon & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 3 \end{pmatrix}, \quad \mathbf{M}_2 = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1-\epsilon & 0 \\ 0 & 0 & 0 & 1+\epsilon \end{pmatrix},$$

in which $\epsilon \in \mathbb{R}$ is small. For $\epsilon = 0$, the two matrices have a common eigenmatrix:

$$\mathbf{E} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

If $\epsilon \neq 0$, $\mathbf{E}$ still nearly diagonalizes $\mathbf{V}_1$ and $\mathbf{V}_2$:

$$\mathbf{M}_1 \cdot \mathbf{E} = \mathbf{E} \cdot \mathrm{diag}\{[1\ 1\ 2\ 3]\} + O(\epsilon), \qquad \mathbf{M}_2 \cdot \mathbf{E} = \mathbf{E} \cdot \mathrm{diag}\{[2\ 3\ 1\ 1]\} + O(\epsilon).$$

On the other hand, for $\epsilon \neq 0$, the distinct eigenmatrices $\mathbf{E}_1$ and $\mathbf{E}_2$, of $\mathbf{V}_1$ and $\mathbf{V}_2$, respectively, are not suitable for diagonalization of the other matrix:

$$\mathbf{M}_1 \cdot \mathbf{E}_2 = \mathbf{E}_2 \cdot \mathrm{diag}\{[1\ 1\ 2\ 3]\} + O(1), \qquad \mathbf{M}_2 \cdot \mathbf{E}_1 = \mathbf{E}_1 \cdot \mathrm{diag}\{[2\ 3\ 1\ 1]\} + O(1).$$

For a simultaneous EVD we have the following uniqueness theorem.

THEOREM 6.1. *For given matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_L \in \mathbb{R}^{R\times R}$, the simultaneous decomposition*

$$\mathbf{M}_1 = \mathbf{U} \cdot \mathbf{D}_1 \cdot \mathbf{U}^{-1}, \tag{6.1}$$

$$\mathbf{M}_2 = \mathbf{U} \cdot \mathbf{D}_2 \cdot \mathbf{U}^{-1}, \tag{6.2}$$

$$\vdots$$

$$\mathbf{M}_L = \mathbf{U} \cdot \mathbf{D}_L \cdot \mathbf{U}^{-1}, \tag{6.3}$$

with $\mathbf{U} \in \mathbb{R}^{R\times R}$ nonsingular and $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_L \in \mathbb{R}^{R\times R}$ diagonal, is unique up to a permutation and a scaling of the columns of $\mathbf{U}$ if and only if all the columns of the matrix

$$\mathbf{D} = \begin{pmatrix} (\mathbf{D}_1)_{11} & (\mathbf{D}_1)_{22} & \ldots & (\mathbf{D}_1)_{RR} \\ (\mathbf{D}_2)_{11} & (\mathbf{D}_2)_{22} & \ldots & (\mathbf{D}_2)_{RR} \\ & & \vdots & \\ (\mathbf{D}_L)_{11} & (\mathbf{D}_L)_{22} & \ldots & (\mathbf{D}_L)_{RR} \end{pmatrix}$$

are distinct.

Proof. Consider $Y = \mathbf{D}^T \cdot X$, for $X \in \mathbb{R}^L$. The ith and jth entry of Y are distinct if X is not perpendicular to $D_i - D_j$. Because $D_i \neq D_j$, the kernel of $D_i^T - D_j^T$ is a subspace of dimension $L-1$. Let $\mathbb{K}$ be the union of the kernels for all $i \neq j$ and let $\tilde{X} \in \mathbb{R}^L \setminus \mathbb{K}$. The EVD of $\sum_l \tilde{x}_l \mathbf{M}_l$ is given by

$$\sum_l \tilde{x}_l \mathbf{M}_l = \mathbf{U} \cdot \left(\sum_l \tilde{x}_l \mathbf{D}_l \right) \cdot \mathbf{U}^{-1} = \mathbf{U} \cdot \mathrm{diag}\{\mathbf{D}^T \cdot \tilde{X}\} \cdot \mathbf{U}^{-1}.$$

Because all eigenvalues are distinct, the eigenmatrix $\mathbf{U}$ is unique up to a permutation and a scaling of its columns. On the other hand, if columns of $\mathbf{D}$ are equal, it is not possible to discriminate between different eigenvectors in the corresponding eigenspace. □

The equivalent for unitary diagonalization is given in [2].

Because of the link between (4.5)–(4.7) and (4.14)–(4.16), the CANDECOMP is essentially unique when $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are nonsingular and $\mathbf{U}^{(3)}$ does not contain collinear columns, as put forward in section 2.

Theorem 6.1 shows that a simultaneous EVD is much more robust than a single EVD. It is well known that, when eigenvalues are close, the eigenvectors in a single EVD may be strongly affected by small perturbations [30]. The reason is that for coinciding eigenvalues only the corresponding eigenspace is defined; different directions in this subspace will emerge as eigenvectors for different infinitesimal perturbations. When this happens for one or more of the matrices in a simultaneous EVD, the other matrices may still allow to identify the actual eigenvectors. We may conclude that, under the conditions of section 2, the CANDECOMP is likely to be stable.

Different permutations of the canonical components will correspond to entirely different matrices $\mathbf{Q}$ and $\mathbf{Z}$ in the SGSD (5.1)–(5.3). However, these in turn lead to different matrices $\mathbf{R}$ and $\mathbf{R}''$ such that, eventually, $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are still subject to the same indeterminacies. In other words, the uniqueness condition has not been weakened by formulating the problem in terms of orthogonal unknowns $\mathbf{Q}$, $\mathbf{Z}$.

It is worth mentioning that, for arbitrary matrices $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K$ (not satisfying our CANDECOMP model), the uniqueness conditions of a S(G)SD are much more severe. In general, only one sequence of (generalized) Schur vectors is possible. For convenience, we will illustrate this only for the SSD (which, in our application, would arise from substitution of the QR-factorization of $\mathbf{U}^{(1)}$ in (4.14)–(4.16)). We have the following theorem.

Theorem 6.2. *Let the matrices* $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_L \in \mathbb{R}^{R\times R}$ *satisfy the SSD*

$$\mathbf{M}_1 = \mathbf{Q}\cdot\mathbf{R}_1\cdot\mathbf{Q}^T, \tag{6.4}$$

$$\mathbf{M}_2 = \mathbf{Q}\cdot\mathbf{R}_2\cdot\mathbf{Q}^T, \tag{6.5}$$

$$\vdots$$

$$\mathbf{M}_L = \mathbf{Q}\cdot\mathbf{R}_L\cdot\mathbf{Q}^T, \tag{6.6}$$

with $\mathbf{Q} = [Q_1 Q_2 \ldots Q_R] \in \mathbb{R}^{R\times R}$ *orthogonal and* $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_L \in \mathbb{R}^{R\times R}$ *upper triangular. An equivalent simultaneous decomposition, in terms of* $\tilde{\mathbf{Q}}$ *and* $\{\tilde{\mathbf{R}}_l\}_{1\leqslant l\leqslant L}$, *in which the diagonal of* $(\tilde{\mathbf{R}}_l)$ *subsequently contains* $(\mathbf{R}_l)_{11}$, $(\mathbf{R}_l)_{22}$, $\ldots$, $(\mathbf{R}_l)_{I-1,I-1}$, $(\mathbf{R}_l)_{JJ}$, $(\mathbf{R}_l)_{I+1,I+1}$, $\ldots$,$(\mathbf{R}_l)_{J-1,J-1}$, $(\mathbf{R}_l)_{II}$, $(\mathbf{R}_l)_{J+1,J+1}$, $\ldots$, $(\mathbf{R}_l)_{KK}$ $(1\leqslant l\leqslant L)$, *exists if and only if the following matrix is rank deficient*:

$$\begin{pmatrix} \mathbf{M}_1 - (\mathbf{R}_1)_{JJ}\mathbf{I} & [Q_1\ldots Q_{I-1}] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{M}_2 - (\mathbf{R}_2)_{JJ}\mathbf{I} & \mathbf{0} & [Q_1\ldots Q_{I-1}] & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_L - (\mathbf{R}_L)_{JJ}\mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & [Q_1\ldots Q_{I-1}] \end{pmatrix}. \tag{6.7}$$

Proof. Let us first answer the simple question of which diagonal entry could be permuted to position $(1,1)$. There is a common eigenvector, other than Q_1, if and only if there exists a $J > 1$ such that all the equations

$$(\mathbf{M}_l - (\mathbf{R}_l)_{JJ}\mathbf{I})X = 0, \qquad 1\leqslant l\leqslant L,$$

have a common solution. This is the condition specified by the theorem for $I = 1$. One can verify that the upper triangular structure can be maintained for new matrices $\tilde{\mathbf{R}}_1$, $\tilde{\mathbf{R}}_2$, …, $\tilde{\mathbf{R}}_L$ and $\tilde{\mathbf{Q}}$ when the entries at position (J, J) are permuted to position $(1, 1)$ and the old entries at positions $(1, 1)$, $(2, 2)$, …, $(J - 1, J - 1)$ are shifted one place down on the diagonal. (The strictly upper diagonal entries of rows 1 to J have to be recomputed.)

In general, the entries at position (J, J) can be brought in Ith position if and only if there exists a vector $X \neq 0$ and scalars b_{li}, $1 \leqslant l \leqslant L$, $1 \leqslant i \leqslant I - 1$, such that

$$(\mathbf{M}_l - (\mathbf{R}_l)_{JJ}\,\mathbf{I})X = \sum_{i=1}^{I-1} b_{li}\, Q_i, \qquad 1 \leqslant l \leqslant L.$$

This is a set of homogeneous linear equations of which the unknowns are the coefficients of X and the scalars $\{b_{li}\}$. The coefficient matrix is given by (6.7). □

Moreover, for noisy data, different permutations of the canonical components will lead to matrices $\mathbf{Q}$, $\mathbf{Z}$ that yield different values of the cost function h defined in (5.6).

6.2. First-order perturbation analysis. To increase our understanding of the stability of the SGSD, let us now conduct a first-order perturbation analysis.

THEOREM 6.3. *Consider the function* $g(\mathbf{Q}, \mathbf{Z})$ *in* (5.4) *and let the matrices* $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$ *be defined by* (5.1)–(5.3)*. The gradients of* g*, with respect to* $\mathbf{Q}$ *and* $\mathbf{Z}$*, over the manifold of orthogonal matrices, are given by*

$$\nabla_{\mathbf{Q}} g = 2\,\mathrm{skew}\left(\sum_k \mathrm{upp}(\mathbf{R}_k)\mathbf{R}_k^T\right) \cdot \mathbf{Q}, \tag{6.8}$$

$$\nabla_{\mathbf{Z}} g = 2\,\mathbf{Z} \cdot \mathrm{skew}\left(\sum_k \mathbf{R}_k^T \mathrm{upp}(\mathbf{R}_k)\right), \tag{6.9}$$

in which skew$(\cdot)$ *is the skew-symmetric and* upp$(\cdot)$ *the upper triangular part of a matrix.*

Proof. We will prove this result by resorting to the framework established in [15, 22]. The gradient of g with respect to $\mathbf{Q}$ can be determined by assuming that $\mathbf{Q}$ has a velocity $\dot{\mathbf{Q}}$ on the manifold of orthogonal matrices and expressing the evolution of g:

$$\dot{g} = \langle \nabla_{\mathbf{Q}}\, g, \dot{\mathbf{Q}} \rangle \tag{6.10}$$

(see, e.g., [15, p. 48]; the formula corresponds to a chain rule for the derivation).

First we express the function g as

$$g(\mathbf{Q}, \mathbf{Z}) = \sum_{k=1}^{K} \langle \mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}, \mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \rangle.$$

Assuming that $\mathbf{Q}$ is time dependent, the derivative with respect to the time coordinate is given by (taking into account that upp$(\cdot)$ is a linear operation)

$$\begin{aligned}\dot{g} &= \sum_{k=1}^{K} [\langle \dot{\mathbf{Q}} \cdot \mathbf{V}_k \cdot \mathbf{Z}, \mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \rangle + \langle \mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}, \mathrm{upp}(\dot{\mathbf{Q}} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \rangle] \\ &= 2 \sum_{k=1}^{K} \langle \dot{\mathbf{Q}} \cdot \mathbf{V}_k \cdot \mathbf{Z}, \mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \rangle.\end{aligned}$$

With a property of the scalar product, we obtain

$$\dot{g} = 2\sum_{k=1}^{K} \langle \dot{\mathbf{Q}}, \mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \rangle.$$

The right term is proportional to the gradient of g over $\mathbb{R}^{R \times R}$. To ensure that $\mathbf{Q}$ stays on the manifold of orthogonal matrices, we claim additionally that

$$\dot{\mathbf{Q}} = \mathbf{\Omega} \cdot \mathbf{Q},$$

in which $\mathbf{\Omega} \in \mathbb{R}^{R \times R}$ is skew-symmetric [22, p. 307]. Now the inner product can be written in the form of (6.10):

$$\begin{aligned}\dot{g} &= 2\sum_{k=1}^{K} \langle \mathbf{\Omega}, \mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T \rangle \\ &= \left\langle \mathbf{\Omega}, 2\sum_{k=1}^{K} \mathrm{skew}\{\mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T\} \right\rangle \\ &= \left\langle \mathbf{\Omega} \cdot \mathbf{Q}, 2\sum_{k=1}^{K} \mathrm{skew}\{\mathrm{upp}(\mathbf{Q} \cdot \mathbf{V}_k \cdot \mathbf{Z}) \cdot \mathbf{Z}^T \cdot \mathbf{V}_k^T \cdot \mathbf{Q}^T\} \cdot \mathbf{Q} \right\rangle,\end{aligned}$$

which proves (6.8). The gradient with respect to $\mathbf{Z}$ can be found in an analogous way. □

THEOREM 6.4. *Consider a first-order perturbation of the matrices in the SGSD* (5.1)–(5.3): $\mathbf{V}_k(\epsilon) = \mathbf{V}_k(0) + \epsilon\,\mathbf{B}_k$ $(1 \leqslant k \leqslant K)$. *As a first-order approximation, the maximum of* $g(\mathbf{Q}, \mathbf{Z})$ *is then obtained for*

$$\begin{aligned}\mathbf{Q}(\epsilon) &= (\mathbf{I} + \epsilon\,\mathbf{\Lambda} + o(\epsilon)) \cdot \mathbf{Q}(0), \\ \mathbf{Z}(\epsilon) &= \mathbf{Z}(0) \cdot (\mathbf{I} + \epsilon\,\mathbf{\Omega} + o(\epsilon)),\end{aligned}$$

in which $\mathbf{\Lambda}, \mathbf{\Omega} \in \mathbb{R}^{R \times R}$ *are skew-symmetric matrices that satisfy the following set of linear equations*:

$$\sum_k \mathrm{lows}(\mathbf{R}_k\mathbf{\Omega} + \mathbf{E}_k + \mathbf{\Lambda}\mathbf{R}_k) \cdot \mathbf{R}_k^T = \mathbf{0}, \tag{6.11}$$

$$\sum_k \mathbf{R}_k^T \cdot \mathrm{lows}(\mathbf{R}_k\mathbf{\Omega} + \mathbf{E}_k + \mathbf{\Lambda}\mathbf{R}_k) = \mathbf{0}, \tag{6.12}$$

where lows$(\cdot)$ *is the strictly lower triangular part of a matrix and*

$$\mathbf{E}_k = \mathbf{Q}(0) \cdot \mathbf{B}_k \cdot \mathbf{Z}(0), \qquad 1 \leqslant k \leqslant K.$$

Proof. Again, we will work in the framework of [15, 22]. Let us start from (5.1)–(5.3). If the matrices $\mathbf{A}_k$ have a velocity $\dot{\mathbf{A}}_k = \mathbf{B}_k$, then $\mathbf{Q}$ evolves in such a way that the identity $\nabla_{\mathbf{Q}} g \equiv 0$ holds. Taking the form of the gradient (6.8) into account, we should have that

$$\mathrm{skew}\left(\sum_k \mathrm{upp}(\mathbf{R}_k)\mathbf{R}_k^T\right) \equiv 0. \tag{6.13}$$

Taking the derivative with respect to the time coordinate yields

$$\operatorname{skew}\left(\sum_k \operatorname{upp}(\dot{\mathbf{Q}} \cdot \mathbf{A}_k \cdot \mathbf{Z} + \mathbf{Q} \cdot \dot{\mathbf{A}}_k \cdot \mathbf{Z} + \mathbf{Q} \cdot \mathbf{A}_k \cdot \dot{\mathbf{Z}})\mathbf{R}_k^T \right.$$
$$\left. + \operatorname{upp}(\mathbf{R}_k)(\dot{\mathbf{Z}}^T \cdot \mathbf{A}_k^T \cdot \mathbf{Q}^T + \mathbf{Z}^T \cdot \dot{\mathbf{A}}_k^T \cdot \mathbf{Q}^T + \mathbf{Z}^T \cdot \mathbf{A}_k^T \cdot \dot{\mathbf{Q}}^T)\right) = 0.$$

To ensure that $\mathbf{Q}$ and $\mathbf{Z}$ stay on the manifold of orthogonal matrices, we claim that

$$\dot{\mathbf{Q}} = \boldsymbol{\Omega} \cdot \mathbf{Q},$$
$$\dot{\mathbf{Z}} = \mathbf{Z} \cdot \boldsymbol{\Lambda},$$

in which $\boldsymbol{\Omega}, \boldsymbol{\Lambda} \in \mathbb{R}^{R \times R}$ are skew-symmetric. If (5.1)–(5.3) are exactly satisfied, then $\operatorname{upp}(\mathbf{R}_k) = \mathbf{R}_k$. Substitution of $\mathbf{E}_k = \mathbf{Q} \cdot \mathbf{B}_k \cdot \mathbf{Z}$ then yields

$$\operatorname{skew}\left(\sum_k \mathbf{R}_k \cdot \boldsymbol{\Omega} \cdot \mathbf{R}_k^T - \operatorname{upp}(\mathbf{R}_k \cdot \boldsymbol{\Omega}) \cdot \mathbf{R}_k^T + \mathbf{E}_k \cdot \mathbf{R}_k^T - \operatorname{upp}(\mathbf{E}_k) \cdot \mathbf{R}_k^T \right.$$
$$\left. + \boldsymbol{\Lambda} \cdot \mathbf{R}_k \cdot \mathbf{R}_k^T - \operatorname{upp}(\boldsymbol{\Lambda} \cdot \mathbf{R}_k) \cdot \mathbf{R}_k^T \right) = 0$$

or

$$\operatorname{skew}\left(\sum_k \operatorname{lows}(\mathbf{R}_k \boldsymbol{\Omega} + \mathbf{E}_k + \boldsymbol{\Lambda} \mathbf{R}_k)\mathbf{R}_k^T\right) = 0.$$

We may drop "skew" because its argument is strictly lower triangular. Equation (6.12) is obtained by starting from the identity $\nabla_{\mathbf{Z}} g \equiv 0$. □

Remark 7. For matrices $\mathbf{A}_k$ that do not allow for an exact upper triangularization, the derivation can be taken over provided that $\operatorname{upp}(\mathbf{R}_k)$ is not simplified to $\mathbf{R}_k$.

Remark 8. Note that the expressions derived in this section may be used to develop routines for the computation of the SGSD by means of an optimization over the (product of two) manifold(s) of orthogonal matrices. We refer to [22].

By the summation in (6.11) and (6.12) the perturbation is to some extent "averaged" over the different matrices $\mathbf{A}_k$. When components of $\mathbf{Q}$ and $\mathbf{Z}$ are ill conditioned for a subset of $\{\mathbf{A}_k\}$, this may be compensated by the other matrices.

7. Algorithms for the SGSD.

7.1. Extended QZ-iteration.

For the actual computation of the SGSD, an extended QZ-iteration was proposed in [48]. One alternates between updates of $\mathbf{Q}$ and $\mathbf{Z}$ in such a way that the cost function h in (5.6) is approximately optimized. In each step, the estimate of $\mathbf{Q}$ (given $\mathbf{Z}$, or vice-versa) is obtained as a product of matrices $\mathbf{H}_1\mathbf{H}_2 \ldots \mathbf{H}_{R-1}$, that form the equivalent of Householder matrices for the computation of a simple QR-decomposition [24]. For instance, as far as $\mathbf{Q}$ is concerned, $\mathbf{H}_1$ maximally reduces (in least-squares sense) the below-diagonal norm of the first columns of the instantaneous estimates of $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K$. After multiplication with $\mathbf{H}_1$, $\mathbf{H}_2$ minimizes the below-diagonal norm of the second columns, without further affecting the first rows, and so on. $\mathbf{H}_1$ is determined through an SVD of an $(R \times K)$-matrix (actually only the left singular vector corresponding to the largest singular value, and its orthogonal complement, have to be computed), the determination of $\mathbf{H}_2$ involves an SVD of an $((R-1) \times K)$-matrix, and so on.

Because of the high computational cost, it makes sense to initialize the algorithm with matrices $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$ defined by two of the equations (5.1)–(5.3). If these two joint decompositions are well conditioned, then $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$ may be close to the optimum; if not, then the extended QZ-iteration may involve more work than just a fine tuning of a good initialization.

The resulting scheme is observed to find a good estimate of the global optimum in a limited number of steps, if the CANDECOMP-model is exactly satisfied. However, even moderate perturbations can cause the algorithm to end up in good estimates of the theoretical matrices $\mathbf{Q}$ and $\mathbf{Z}$ that do not globally minimize the cost function. It is also possible that at some point in the iteration (e.g., initially, or after approximate convergence), the algorithm starts to increase the value of h. The reason for this behavior is that the way in which $\mathbf{Q}$ and $\mathbf{Z}$ are computed does not imply monotonic convergence in terms of h: for instance, it is possible that the matrix $\mathbf{H}_1$ increases the Frobenius-norm of the part of columns 2 to $R-1$ below the diagonal. Nevertheless, these aspects do not seem to pose major problems in practice: over several hundreds of simulations, we have only once obtained a meaningless result.

7.2. Jacobi iteration. In [17, 18] we derived a Jacobi-type algorithm for the computation of the SGSD. Here, $\mathbf{Q}$ and $\mathbf{Z}$ are found as a sequence of elementary Jacobi-rotation matrices. In a step (i, j), $\mathbf{Q}$ and $\mathbf{Z}$ are multiplied by elementary rotation matrices, affecting rows and columns i and j. These rotation matrices are such that they maximize the function g in (5.4). It turns out that the determination of a Jacobi-rotation pair basically amounts to rooting a polynomial of degree 8. One sweeps over all the possible pairs (i, j), and then iterates over such sweeps.

The iteration can be initialized with matrices $\mathbf{Q}^{(0)}$ and $\mathbf{Z}^{(0)}$, obtained from the generalized Schur decomposition corresponding to two of the equations in (5.1)–(5.3) [24]. Assume that at iteration step $l+1$, the estimates $\mathbf{Q}^{(l)}$, $\mathbf{Z}^{(l)}$, and $\mathbf{R}_1^{(l)}, \ldots, \mathbf{R}_K^{(l)}$ are available. Let $\mathbf{G}_{ij} \in \mathbb{R}^{R \times R}$ represent an elementary Givens rotation matrix that affects rows i and j, i.e., $\mathbf{G}_{ij}$ equals the identity matrix, except for the entries

$$(\mathbf{G}_{ij})_{ii} = (\mathbf{G}_{ij})_{jj} = \cos\alpha,$$
$$(\mathbf{G}_{ij})_{ji} = -(\mathbf{G}_{ij})_{ij} = \sin\alpha,$$

in which α is the rotation angle (assume that $j > i$). An update of $\mathbf{Q}^{(l)}$ takes the form of $\mathbf{Q}^{(l+1)} = \mathbf{G}_{ij} \cdot \mathbf{Q}^{(l)}$. Similarly, an update of $\mathbf{Z}^{(l)}$ takes the form of $\mathbf{Z}^{(l+1)} = \mathbf{Z}^{(l)} \cdot {\mathbf{G}'_{ij}}^T$, where the Givens rotation matrix $\mathbf{G}'_{ij}$ is defined in the same way as $\mathbf{G}_{ij}$, in terms of an angle β. At the same time $\mathbf{R}_1^{(l)}, \mathbf{R}_2^{(l)}, \ldots, \mathbf{R}_K^{(l)}$ are updated as $\mathbf{R}_1^{(l+1)} = \mathbf{G}_{ij} \cdot \mathbf{R}_1^{(l)} \cdot {\mathbf{G}'_{ij}}^T$, $\mathbf{R}_2^{(l+1)} = \mathbf{G}_{ij} \cdot \mathbf{R}_2^{(l)} \cdot {\mathbf{G}'_{ij}}^T, \ldots, \mathbf{R}_K^{(l+1)} = \mathbf{G}_{ij} \cdot \mathbf{R_K}^{(l)} \cdot {\mathbf{G}'_{ij}}^T$.

At iteration step l, the maximization of the function g in (5.4) is equivalent to the minimization of

$$h(\alpha, \beta) = \sum_{k=1}^{K} \left[(\mathbf{R}_k^{(l+1)})_{ji}^2 + \sum_{r=i+1}^{j-1} ((\mathbf{R}_k^{(l+1)})_{ri}^2 + (\mathbf{R}_k^{(l+1)})_{jr}^2) \right] \tag{7.1}$$

(the other entries do not affect the norm of the strictly lower diagonal parts). The function h is given in explicit form by

$$h(\alpha, \beta) = \sum_{k=1}^{K} \sum_{n=1}^{5} h_{kn}(\alpha, \beta), \tag{7.2}$$

in which

$$\begin{aligned} h_{k1}(\alpha,\beta) &= \sin^2\alpha \\ &\times[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ii}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{ij}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij}], \end{aligned} \tag{7.3}$$

$$\begin{aligned} h_{k2}(\alpha,\beta) &= 2\sin\alpha\cos\alpha\,\Big\{\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji} + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj} \\ &\qquad - \sin\beta\cos\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}]\Big\}, \end{aligned} \tag{7.4}$$

$$\begin{aligned} h_{k3}(\alpha,\beta) &= \cos^2\alpha \\ &\times[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ji}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{jj}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj}], \end{aligned} \tag{7.5}$$

$$\begin{aligned} h_{k4}(\alpha,\beta) &= (\sin^2\alpha + \cos^2\alpha) \\ &\times \sum_{r=i+1}^{j-1}[\cos^2\beta\,(\mathbf{R}_k^{(l)})_{ri}^2 + \sin^2\beta\,(\mathbf{R}_k^{(l)})_{rj}^2 - 2\sin\beta\cos\beta\,(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{rj}], \end{aligned} \tag{7.6}$$

$$\begin{aligned} h_{k5}(\alpha,\beta) &= (\sin^2\beta + \cos^2\beta) \\ &\times \sum_{r=i+1}^{j-1}[\cos^2\alpha\,(\mathbf{R}_k^{(l)})_{jr}^2 + \sin^2\alpha\,(\mathbf{R}_k^{(l)})_{ir}^2 + 2\sin\alpha\cos\alpha\,(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}]. \end{aligned} \tag{7.7}$$

Setting the partial derivatives of h, with respect to α and β, equal to zero, leads to a set of biquadratic equations in $\tan\alpha$ and $\tan\beta$:

$$b_1(\beta)\tan^2\alpha + b_2(\beta)\tan\alpha - b_1(\beta) = 0, \tag{7.8}$$

$$b_3(\beta)\tan^2\alpha + b_4(\beta)\tan\alpha + b_5(\beta) = 0, \tag{7.9}$$

in which $b_n(\beta) = \sum_{k=1}^{K} b_{kn}(\beta)$, with

$$\begin{aligned} b_{k1}(\beta) &= \tan^2\beta\left\{(\mathbf{R}_k^{(l)})_{ij}^2 - (\mathbf{R}_k^{(l)})_{jj}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{ir}^2 - (\mathbf{R}_k^{(l)})_{jr}^2]\right\} \\ &\quad +2\tan\beta\,[(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj} - (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij}] \\ &\quad +\left\{(\mathbf{R}_k^{(l)})_{ii}^2 - (\mathbf{R}_k^{(l)})_{ji}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{ir}^2 - (\mathbf{R}_k^{(l)})_{jr}^2]\right\}, \end{aligned} \tag{7.10}$$

$$\begin{aligned} b_{k2}(\beta) &= \tan^2\beta\left[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}\right] \\ &\quad -\tan\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}] \\ &\quad +\left[(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ir}(\mathbf{R}_k^{(l)})_{jr}\right], \end{aligned} \tag{7.11}$$

$$\begin{aligned} b_{k3}(\beta) &= (\tan^2\beta - 1)\left[(\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ij} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{jr}\right] \\ &\quad +\tan\beta\left\{(\mathbf{R}_k^{(l)})_{ij}^2 - (\mathbf{R}_k^{(l)})_{ii}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{rj}^2 - (\mathbf{R}_k^{(l)})_{ri}^2]\right\}, \end{aligned} \tag{7.12}$$

$$b_{k4}(\beta) = (\tan^2\beta - 1)\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{ji} + (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{jj}]$$

$$+2\tan\beta\,[(\mathbf{R}_k^{(l)})_{ij}(\mathbf{R}_k^{(l)})_{jj} - (\mathbf{R}_k^{(l)})_{ii}(\mathbf{R}_k^{(l)})_{ji}], \tag{7.13}$$

$$b_{k5}(\beta) = (\tan^2\beta - 1)\left[(\mathbf{R}_k^{(l)})_{ji}(\mathbf{R}_k^{(l)})_{jj} + \sum_{r=i+1}^{j-1}(\mathbf{R}_k^{(l)})_{ri}(\mathbf{R}_k^{(l)})_{rj}\right]$$

$$+\tan\beta\left\{(\mathbf{R}_k^{(l)})_{jj}^2 - (\mathbf{R}_k^{(l)})_{ji}^2 + \sum_{r=i+1}^{j-1}[(\mathbf{R}_k^{(l)})_{rj}^2 - (\mathbf{R}_k^{(l)})_{ri}^2]\right\}. \tag{7.14}$$

The global minimum of $h(\alpha,\beta)$ can be determined by computing the various solutions of (7.8)–(7.9) and selecting the one corresponding to the smallest value in (7.2).

For the solution of the set of biquadratic equations, let us first consider the special case where (7.8) is linear in $\tan\alpha$: $b_1(\beta) = 0$. The only ways in which a root β_0 of b_1 can lead to a solution of (7.8)–(7.9) are (a) $\tan\alpha = 0$ and additionally $b_5(\beta_0) = 0$ and (b) α is a solution of (7.9), for $\beta = \beta_0$, which additionally satisfies $b_2(\beta_0) = 0$.

Now let us investigate the general case, i.e., $b_1(\beta) \neq 0$. Substitution of the square roots of (7.8) in (7.9) (considered as quadratic expressions in the unknown $\tan\alpha$) then leads to the following polynomial of degree 8 in $\tan\beta$:

$$b_1^2(\beta)b_3^2(\beta) + b_1^2(\beta)b_5^2(\beta) - b_1(\beta)b_2(\beta)b_4(\beta)b_5(\beta) + 2b_1^2(\beta)b_3(\beta)b_5(\beta)$$

$$+ b_2^2(\beta)b_3(\beta)b_5(\beta) - b_1^2(\beta)b_4^2(\beta) + b_1(\beta)b_2(\beta)b_3(\beta)b_4(\beta) = 0. \tag{7.15}$$

For the roots of this polynomial, the corresponding value of $\tan\alpha$ that gives a solution to (7.8)–(7.9), can be found from

$$(b_2(\beta)b_3(\beta) - b_1(\beta)b_4(\beta))\tan\alpha - b_1(\beta)(b_3(\beta) + b_5(\beta)) = 0. \tag{7.16}$$

The computational cost is in line with results obtained for other simultaneous matrix decompositions. A Jacobi-rotation for a simultaneous real symmetric EVD can be computed by rooting a polynomial of degree 2 [8, 9]. For an SSD ($\mathbf{Q} = \mathbf{Z}$), polynomials are of degree 4 [25].

The Jacobi-result is an explicit solution for the CANDECOMP of rank-2 tensors. Apart from this result, a Jacobi-sweep is more expensive than an extended QZ-step if not $\min\{R, K\} \gg 8$.

If the simultaneous equivalence transformation of (4.5)–(4.7) is not exactly satisfied, different permutations of the CANDECOMP components may cause the corresponding orthogonal factors $\mathbf{Q}$ and $\mathbf{Z}$ to yield values of the function g that are somewhat different. There is no guarantee that the Jacobi-algorithm will converge to the solution with that specific column ordering that leads to the global optimum. Apart from the reordering of columns, there is no formal evidence that the two-sided Jacobi-algorithm cannot get stuck in a local optimum; local or global convergence is still an open problem for the computation of other simultaneous matrix decompositions as well [4, 8, 9, 12, 23, 48]. We have not observed convergence to a local optimum in any of our simulations for the unsymmetric CANDECOMP-problem. For the case where $\mathbf{U}^{(1)} = \mathbf{U}^{(2)}$, a meaningless result has been obtained for one out of hundreds of simulations. In this odd case, the problem could be overcome by reinitializing the algorithm.

8. Estimation of the canonical components from the components of the SGSD. In this section we will explain how the matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be estimated, once $\mathbf{Q}$ and $\mathbf{Z}$ are known. How $\mathbf{U}^{(3)}$ may subsequently be estimated was explained in section 4. This corresponds to step 4 in Algorithm 1. Computation of the SGSD is in general only equivalent to least-squares fitting of the CANDECOMP-model if that model is exactly valid. The estimates obtained so far may then be used to initialize an additional optimization algorithm for the minimization of cost function (2.3), as also mentioned in section 4 (step 5 in Algorithm 1).

In [48] a procedure has been proposed that works under the assumption that the columns of $\mathbf{U}^{(3)}$ are linearly independent (and sufficiently well conditioned). Hence this technique can be used only when $K \geqslant R$. The solution is obtained via the computation of the pseudoinverse of a $(K \times R)$ matrix and the estimation of the best rank-1 approximation of R $(R \times R)$ matrices.

We will derive a new technique that works under the assumptions established in section 2. This technique is also computationally less demanding. It essentially requires solving $R(R-1)/2$ overdetermined sets of K linear equations in 2 unknowns.

We will estimate $\mathbf{R}'$ and $\mathbf{R}''$ from (5.1)–(5.3) and then combine them with $\mathbf{Q}$ and $\mathbf{Z}$ to obtain $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$. If we assume that the main diagonals of $\mathbf{R}'$ and $\mathbf{R}''$ contain only entries equal to 1 (we can make this assumption because the columns of $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ can be determined only up to a scaling factor), then $\mathbf{D}_k = \text{diag}\{\mathbf{R}_k\}$ $(1 \leqslant k \leqslant K)$, in which diag$\{\cdot\}$ now denotes the diagonal part of a matrix. The strictly upper diagonal elements of $\mathbf{R}'$ and $\mathbf{R}''$ can be estimated by subsequently solving in a least-squares sense the equations related to the entries of $\{\mathbf{R}_k\}_{(1 \leqslant k \leqslant K)}$ at positions $(R-1, R)$, $(R-2, R-1)$, $(R-2, R)$, …, $(1, 2)$, $(1, 3)$, …, $(1, R)$ in (5.1)–(5.3) with respect to the unknowns $r'_{R-1,R}$ and $r''_{R-1,R}$, $r'_{R-2,R-1}$ and $r''_{R-2,R-1}$, $r'_{R-2,R}$, and $r''_{R-2,R}$, …, $r'_{1,2}$ and $r''_{1,2}$, $r'_{1,3}$ and $r''_{1,3}$, …, $r'_{1,R}$ and $r''_{1,R}$, respectively. For instance, with the entries at position $(R-1, R)$ corresponds the equation

$$\begin{pmatrix} (\mathbf{R}_1)_{R,R} & (\mathbf{R}_1)_{R-1,R-1} \\ (\mathbf{R}_2)_{R,R} & (\mathbf{R}_2)_{R-1,R-1} \\ \vdots & \vdots \\ (\mathbf{R}_K)_{R,R} & (\mathbf{R}_K)_{R-1,R-1} \end{pmatrix} \begin{pmatrix} r'_{R-1,R} \\ r''_{R-1,R} \end{pmatrix} = \begin{pmatrix} (\mathbf{R}_1)_{R-1,R} \\ (\mathbf{R}_2)_{R-1,R} \\ \vdots \\ (\mathbf{R}_K)_{R-1,R} \end{pmatrix}.$$

Note that, according to the third working assumption made in section 2, the columns of the matrix on the left-hand side of this equation should be linearly independent.

For the computation of $\mathbf{U}^{(3)}$, remark that (4.5)–(4.7) correspond to a CANDECOMP of a tensor $\mathcal{V} \in \mathbb{R}^{R \times R \times K}$, with entries $v_{ijk} = (\mathbf{V}_k)_{ij}$, of which the component matrices are $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$ and the matrix $\tilde{\mathbf{U}}^{(3)}$ defined in (4.8). Let $\mathbf{V}_{(R^2 \times K)} \in \mathbb{R}^{R^2 \times K}$, with entries

$$(\mathbf{V}_{(R^2 \times K)})_{(i-1)R+j,k} = (\mathbf{V}_k)_{ij},$$

be a matrix representation of $\mathcal{V}$. (4.5)–(4.7) can be reformulated as

$$\mathbf{V}_{(R^2 \times K)} = (\mathbf{U}^{(1)} \odot \mathbf{U}^{(2)}) \cdot \tilde{\mathbf{U}}^{(3)^T}. \tag{8.1}$$

$\tilde{\mathbf{U}}^{(3)}$ can be computed from this (possibly overdetermined) set of linear equations. Finally, $\mathbf{U}^{(3)}$ follows from (4.9).

To conclude, let us give an outline of the computation of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$ from the results of the SGSD (5.1)–(5.3). This scheme details step 4 in Algorithm 1.

```
4.1 Computation of $\mathbf{R}'$ and $\mathbf{R}''$.
    Set diag$\{\mathbf{R}'\}$ = diag$\{\mathbf{R}''\} = \mathbf{I}$.
    for $i = R-1, R-2, \ldots, 1$
```

```
      for $j = i+1, i+2, \ldots, R$
          $\begin{pmatrix} (\mathbf{R}_1)_{jj} & (\mathbf{R}_1)_{ii} \\ (\mathbf{R}_2)_{jj} & (\mathbf{R}_2)_{ii} \\ \vdots & \vdots \\ (\mathbf{R}_K)_{jj} & (\mathbf{R}_K)_{ii} \end{pmatrix} \begin{pmatrix} r'_{ij} \\ r''_{ij} \end{pmatrix} = \begin{pmatrix} (\mathbf{R}_1)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip}\,(\mathbf{R}_1)_{pp}\,r''_{pj} \\ (\mathbf{R}_2)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip}\,(\mathbf{R}_2)_{pp}\,r''_{pj} \\ \vdots \\ (\mathbf{R}_K)_{ij} - \sum_{p=i+1}^{j-1} r'_{ip}\,(\mathbf{R}_K)_{pp}\,r''_{pj} \end{pmatrix}$
      end
   end
4.2 $\mathbf{U}^{(1)} = \mathbf{Q}^T \cdot \mathbf{R}'$. $\mathbf{U}^{(2)} = \mathbf{Z} \cdot (\mathbf{R}'')^T$.
4.3 Compute $\tilde{\mathbf{U}}^{(3)}$ from (8.1). Compute $\mathbf{U}^{(3)}$, modulo a scaling of its columns,
    from (4.9).
```

9. Numerical experiments. In this section we illustrate the performance of the algorithms proposed in this paper by means of a number of numerical experiments. These experiments are helpful to understand and evaluate the different methods, given that a rigorous mathematical analysis of their convergence properties often proves to be extremely tough (as is witnessed by the fact that only very few related results are available [4, 50]).

In a first series of experiments we will compare the accuracy of both techniques presented in section 7 and check whether an additional direct optimization of the cost function f, defined in (2.3), is needed (step 5 in Algorithm 1). We will also show that the extended QZ-iteration is not simply based on the minimization of cost function h, defined in (5.6).

Tensors $\mathcal{A} \in \mathbb{R}^{3\times 3\times 3}$, of which the canonical components will afterwards be estimated, are generated in the following way:

$$\mathcal{A} = \tilde{\mathcal{A}}/\|\tilde{\mathcal{A}}\| + \sigma_N \tilde{\mathcal{N}}/\|\tilde{\mathcal{N}}\|, \tag{9.1}$$

in which $\tilde{\mathcal{A}}$ exactly satisfies the CANDECOMP-model:

$$\tilde{\mathcal{A}} = U_1^{(1)} \circ U_1^{(2)} \circ U_1^{(3)} + U_2^{(1)} \circ U_2^{(2)} \circ U_2^{(3)} + U_3^{(1)} \circ U_3^{(2)} \circ U_3^{(3)}. \tag{9.2}$$

The components in (9.1)–(9.2) are generated as follows. First consider the (3×3)-matrices $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$, defined by (4.3). The entries of 3 (3×3)-matrices are randomly taken from a uniform distribution on the interval $[0, 1)$. $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ are derived from two of these matrices by replacing their singular values by 3, 2, 1, while keeping the singular vectors. $\mathbf{U}^{(1)}$ is generated in the same way but three different sets of singular values will be considered: 3, 2, 1; 30, 15, 1; 100, 50, 1. The entries of $\tilde{\mathcal{N}}$ are drawn from a zero-mean unit-variance Gaussian distribution. For each particular choice of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$, and $\tilde{\mathcal{N}}$, the scalar σ_N is varied between $1e-3$ and 1.

For each of the sets of singular values of $\mathbf{U}^{(1)}$, 50 independent samples of $\tilde{\mathcal{A}}$ are realized; for each of them 7 logarithmically equidistant values of σ_N are considered. In each Monte Carlo simulation the following algorithms are run: (a) the Jacobi-algorithm, discussed in section 7.2; (b) a least-squares matching of both sides of (2.3), for which the `leastsq` command of the Optimization Toolbox 1.0 of MATLAB 4.2 has been used, initialized with the result of (a); and (c) the extended QZ-iteration, described in section 7.1. The algorithm (a) is terminated if a full sweep no longer allows the reduction of the cost function $h(\mathbf{Q}, \mathbf{Z})$ with at least 0.01%. The same termination criterion is used for a Q-step followed by a Z-step in the extended QZ-iteration. For the least-squares matching (b) a minimal precision of $1e-5$ for the optimal values of the cost function f, defined in (2.3), and the corresponding components is presumed; the MATLAB routine maximally performs 2100 iteration steps.

To evaluate the accuracy of the different algorithms we will consider the quality of the estimate $\hat{\mathbf{U}}^{(1)}$ of $\mathbf{U}^{(1)}$; at this point the columns of $\mathbf{U}^{(1)}$ are normalized to unit-length. In Figure 9.1 the error is plotted as a function of the noise level σ_N. For a given noise level and a given algorithm, this error measure has been computed as the average, over the different Monte Carlo simulations, of the Frobenius-norm $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$; the ordering of the columns of $\hat{\mathbf{U}}^{(1)}$ that corresponds to the ordering of the columns of $\mathbf{U}^{(1)}$, has been determined as the ordering that minimizes the error; we also scale the columns of $\hat{\mathbf{U}}^{(1)}$ in the optimal way. Algorithms (a), (b), and (c) correspond to solid, dotted, and dash-dot curves, respectively. The upper, middle, and lower curves correspond to a condition number of $\mathbf{U}^{(1)}$, equal to 100, 30 and 3, respectively.

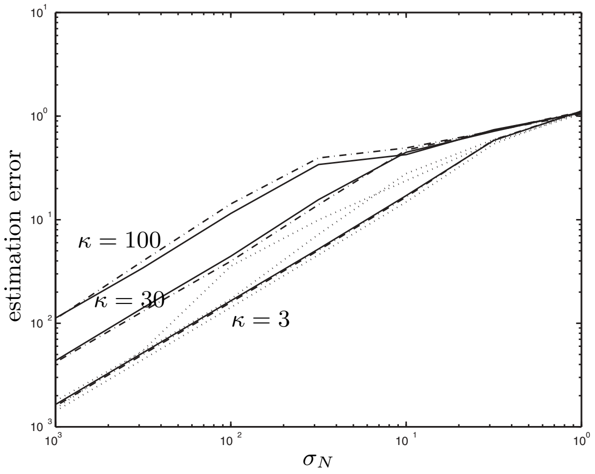

FIG. 9.1. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$*, as a function of the noise level* σ_N*, for the Jacobi-algorithm (solid), with additional least-squares matching (dotted) and the extended* QZ*-algorithm (dashdot). The upper, middle, and lower curves correspond to a condition number* κ *of* $\mathbf{U}^{(1)}$*, equal to* 100*,* 30*, and* 3*, respectively.*

Figure 9.1 displays the expected performance degradation as the noise level and/or the condition number of $\mathbf{U}^{(1)}$ increases. The number of simulations is high enough to give a good picture: the variance of the error, divided by its squared value ranges from at least $4e-6$ ($\sigma_N = 1e-3$) to typically $2.5e-2$ ($\sigma_N = 1$). We notice that on the average, the accuracy of methods (a) and (c) is comparable. The figure also shows that an additional least-squares matching routine generally improved the accuracy, but that the marginal improvement became smaller as the CANDECOMP factors were better conditioned. For well-conditioned problems, no direct optimization of f is needed.

In Figure 9.2 we have plotted the mean value of the cost function h in (5.6) for the algorithms (a) and (c). The figure shows that the extended QZ-iteration indeed does not minimize cost function h; this effect is more outspoken as the condition number of $\mathbf{U}^{(1)}$ is larger. On the other hand, it is clear from the discussion in section 7.1 that, in the absence of noise, the theoretical solution is a stationary point of the extended QZ-algorithm; in Figure 9.1 we see that the algorithm was still reliable in the presence

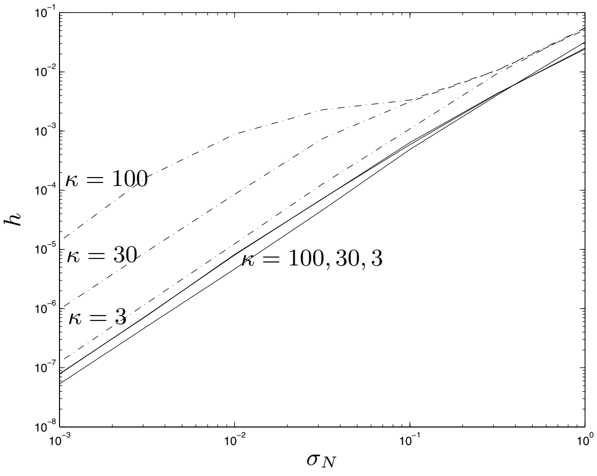

FIG. 9.2. *The mean value of h, defined in* (5.6), *as a function of the noise level σ_N, for the Jacobi-algorithm (solid) and the extended QZ-algorithm (dashdot). The upper, middle, and lower curves correspond to a condition number κ of $\mathbf{U}^{(1)}$, equal to* 100, 30, *and* 3, *respectively.*

of noise.

In a second series of experiments we illustrate the convergence behavior of methods (a) and (c). For each of the sets of singular values of $\mathbf{U}^{(1)}$ (3, 2, 1; 30, 15, 1; 100, 50, 1), 100 independent samples of $\mathcal{A}$ are realized as before, with $\sigma_N = 0$. The different algorithms are now terminated if the instantaneous value of h has been reduced below $1e - 14$.

In Figure 9.3 we have plotted the average evolution of the value of h as a function of the iteration step l (for the scenarios with condition number $\kappa = 3, 30, 100$) and for algorithm (a) (only with condition number $\kappa = 3$, as will be motivated immediately). For these curves, the convergence speed is quasi-linear. The curves for the extended QZ-iteration have only been marginally affected by the chosen value of κ. On the other hand, it makes less sense to plot an average curve for the Jacobi-method in the cases where $\kappa = 30$ or 100, as the results can be strongly data-dependent. Namely, the convergence is still good in most cases, but for some particular instances of $\mathcal{A}$, the algorithm is observed to move through a swamp: apparently, like ALS iterations, Jacobi-iterations can be affected by swamps, although for well-conditioned problems they seem to form a minor issue. The extended QZ-algorithm appears to be less vulnerable, as the typical swamp behavior has only been observed for one instance of $\mathcal{A}$ ($\kappa = 100$). Rather than plotting the remaining mean convergence curves, we show in a histogram how many iterations were needed to terminate the algorithm in the different Monte Carlo simulations, for the various set-ups (see Figure 9.4, in which we have taken as a convention that experiments in which more than 100 iterations were needed, are added to the final histogram bin). Concerning Figures 9.3 and 9.4, we finally remark that for 5 instances of $\mathcal{A}$, the extended QZ-algorithm was observed to start by increasing the value of h to some extent, before actual convergence.

With respect to Figure 9.4 we conclude that for good condition numbers, the Jacobi-algorithm requires less iterations than the extended QZ-iteration. However, when the condition number increases, the risk increases that the Jacobi-algorithm requires a higher number of iterations. In this respect, we should also keep in mind

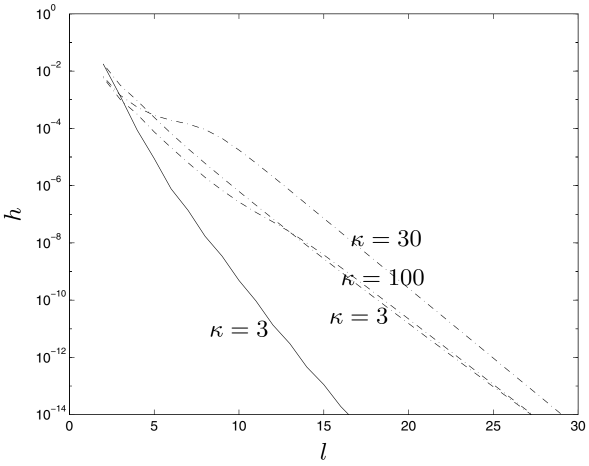

FIG. 9.3. *The evolution of h, defined in* (5.6)*, as a function of the iteration step l, for the Jacobi-algorithm (solid) and the extended QZ-algorithm (dashdot). The parameter κ is the condition number of* $\mathbf{U}^{(1)}$.

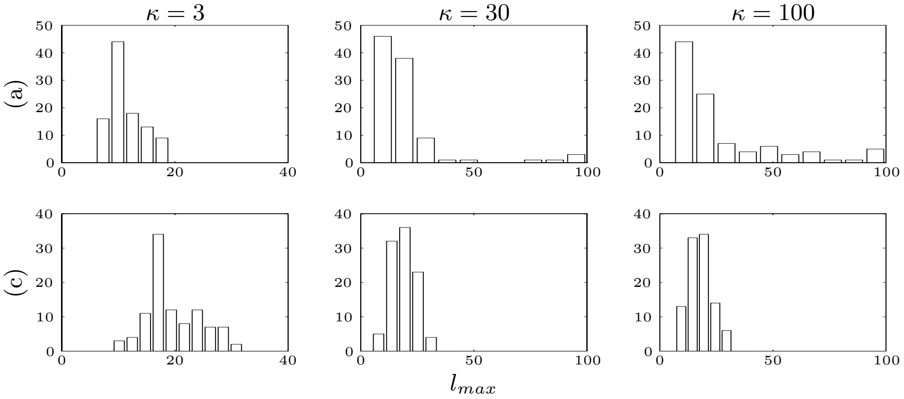

FIG. 9.4. *Histogram of* 100 *Monte Carlo simulations, showing the number of iterations required to reduce the value of h (see* (5.6)*) below* 1e−14*, for methods* (a) *(Jacobi-algorithm) and* (c)*(extended QZ-algorithm), and a condition number κ of* $\mathbf{U}^{(1)}$*, equal to* 3*,* 30*, or* 100*. Experiments in which more than* 100 *iterations were needed have been added to the final histogram bin.*

that for small tensors the computational complexity of a Jacobi-iteration step is higher than that of an extended QZ-iteration step; for larger tensor sizes, the extended QZ-iteration steps are more complex (as is clear from the discussion in section 7).

Figure 9.5 is an example of an ALS iteration moving through a "swamp." After 5 iterations the convergence speed becomes almost equal to zero, and after 70 iterations it starts to increase again. It is clear that tolerances have to be set very tight in order to reach the global optimum. This type of convergence is not uncommon. The figure was obtained for a $(2 \times 2 \times 4)$ tensor of the form (9.1), with the condition numbers of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{U}^{(3)}$ equal to 2 and $\sigma_N = 1e - 2$. Note that in this case (7.15) provides an explicit expression for the solution.

In the following experiment we will compare the performance of the simultaneous generalized Schur approach with other techniques. In each of 50 Monte Carlo runs, a tensor $\mathcal{A} \in \mathbb{R}^{2\times 2\times 10}$ of the form (9.1) is generated. The singular values of $\mathbf{U}^{(2)}$ are taken equal to $2, 1$. For $\mathbf{U}^{(1)}$ three different sets of singular values are considered:

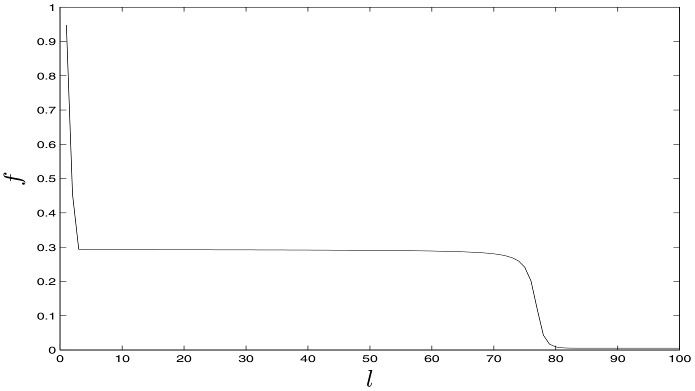

FIG. 9.5. *Example of a "swamp"-type convergence curve for ALS iterations. f is the cost function defined in* (2.3) *and l the iteration step.*

2,1; 10,1; 100,1. The entries of $\mathbf{U}^{(3)}$ are generated as $u_{ij}^{(3)} = 1 + g_{ij}/50$, in which g_{ij} is drawn from a Gaussian distribution with unit variance. For each particular choice of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, $\mathbf{U}^{(3)}$, and $\tilde{\mathcal{N}}$, the scalar σ_N is varied between $1e-4$ and $1e-2$. In this way, σ_N ranges from a level where the eigenvalues in (4.10) are subject only to a small perturbation to a level where there is a certain risk that these eigenvalues have crossed each other.

In Figure 9.6 we compare the mean value of $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$ obtained with a SGSD to the one obtained from the EVD of the matrix $\mathbf{V}_2 \cdot \mathbf{V}_1^{-1}$ (cf. [38, 43, 5, 42]). It is clear that the SGSD is more accurate than a single EVD, because it takes all the matrices $\mathbf{V}_k$ into account. However, the technique is more sensitive to the condition number of $\mathbf{U}^{(1)}$. In the case of an ill-conditioned matrix $\mathbf{U}^{(1)}$, the performance may considerably degrade when the noise level is high; as such, this effect cannot be examined by means of the first-order perturbation analysis in section 6. Note that the EVD may yield complex eigenvalues and eigenvectors for low signal-to-noise ratios.

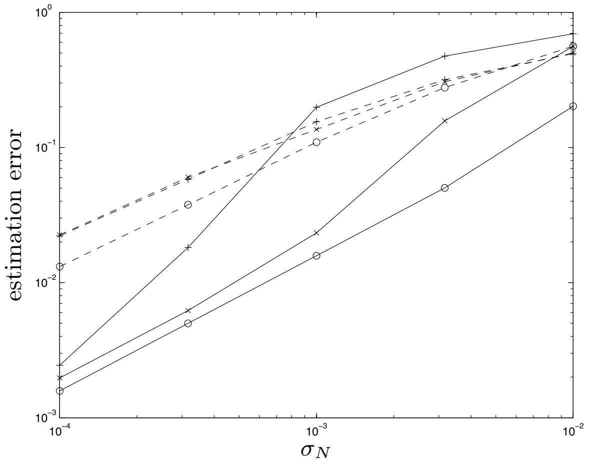

FIG. 9.6. *The mean value of $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, as a function of the noise level σ_N, for the SGSD (solid) and for a single EVD (dashed). The condition number κ of $\mathbf{U}^{(1)}$ is equal to* 2 *(○),* 10 *(×), or* 100 *(+).*

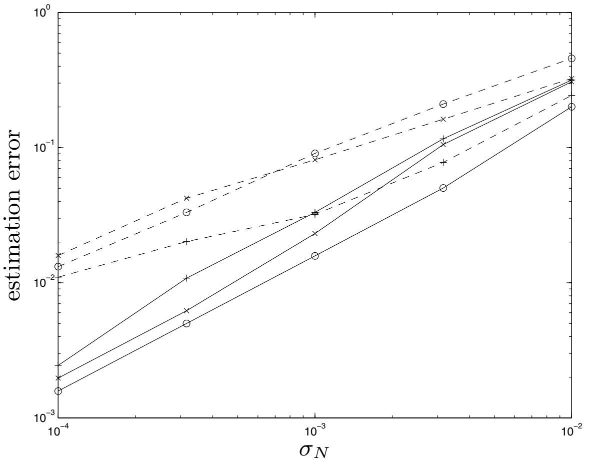

FIG. 9.7. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$, *as a function of the noise level* σ_N, *for the SGSD, followed by an ALS iteration (solid) and for an EVD followed by an ALS iteration (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(*$\circ$*),* 10 *(*$\times$*), or* 100 *(*$+$*).*

In Figure 9.7 we display the accuracy obtained when the results of Figure 9.6 are used to initialize an ALS routine. The iteration was terminated when

$$\left\| \begin{pmatrix} \hat{\mathbf{U}}^{(1)}_{k+1} \\ \hat{\mathbf{U}}^{(2)}_{k+1} \\ \hat{\mathbf{U}}^{(3)}_{k+1} \end{pmatrix} - \begin{pmatrix} \hat{\mathbf{U}}^{(1)}_{k} \\ \hat{\mathbf{U}}^{(2)}_{k} \\ \hat{\mathbf{U}}^{(3)}_{k} \end{pmatrix} \right\| < 1e-4,$$

in which $\hat{\mathbf{U}}^{(i)}_k$ is the estimate of $\mathbf{U}^{(i)}$ at iteration step k. We see that, even after an ALS iteration, the EVD approach remains less accurate than the simultaneous generalized Schur approach. In additional simulations we observed that this is less the case when $\mathbf{U}^{(3)}$ is better conditioned.

In Figure 9.8 we put the result obtained by the SGSD and the enhanced result obtained by an extra ALS iteration next to each other. It turns out that the performance degradation that is linked to a bad condition number of $\mathbf{U}^{(1)}$ (as mentioned in the discussion of Figure 9.6), can be mitigated by an additional ALS iteration. If there is no such problem, then an extra ALS iteration is not required.

In Figure 9.9 we compare the ALS-enhanced SGSD result to the best result obtained by ALS iteration, starting from 10 random initializations. Remarkably enough, ALS gives better results when the condition number of $\mathbf{U}^{(1)}$ increases. The SGSD turns out to be more accurate than the direct ALS approach. In additional simulations we observed that the difference in performance decreases when $\mathbf{U}^{(3)}$ is better conditioned.

In Figure 9.10 we plot the total CPU time, over 50 Monte Carlo runs and 10 random initializations per run, required by the ALS routine. Analysis of the data showed that, for a given value of κ and σ_N, not more than 2 out of 3 initializations led to an estimation error $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$ that was more than twice its minimal value over all runs and initializations. This ratio depended little on the particular value of κ and σ_N. This means that we actually did not have to start from 10 random initializations; 5 initializations would have been sufficient and the computational cost can be divided by two. Nevertheless, this remains much more expensive than a SGSD

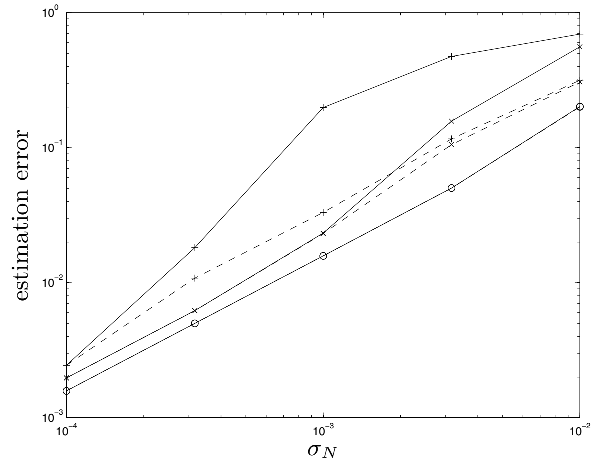

FIG. 9.8. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$*, as a function of the noise level* σ_N*, obtained by the SGSD (solid), and by a subsequent ALS iteration (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(○),* 10 *(×), or* 100 *(+).*

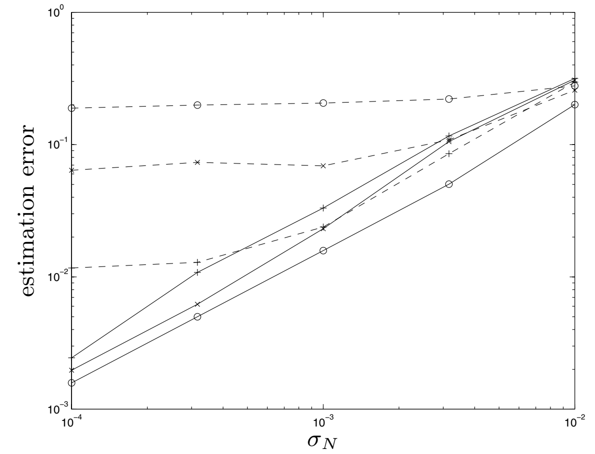

FIG. 9.9. *The mean value of* $\|\hat{\mathbf{U}}^{(1)} - \mathbf{U}^{(1)}\|$*, as a function of the noise level* σ_N*, for the ALS-enhanced SGSD (solid) and direct ALS starting from* 10 *random initializations (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(○),* 10 *(×), or* 100 *(+).*

(overall CPU time approximately 1 s, independent of κ and σ_N) or a simple EVD (CPU time in the order of magnitude of $1e - 2$ s) (the latter merely consists of MATLAB's function `eig` applied to a (2×2) matrix).

Figure 9.11 shows the CPU time required by the ALS iteration that was initialized by means of the SGSD or the EVD. Fewer computations were needed for the SGSD. The figure also shows that each of these two special initializations led to fewer ALS iterations than an average random start.

Whenever in this section we have used ALS iterations for the optimization of cost function f, we have also tried the general-purpose Levenberg–Marquardt algorithm [39] (we used the command `lsqnonlin` of the Optimization Toolbox 2.0 of MATLAB 5.3). In the last series of experiments, Levenberg–Marquardt gave consistently much less accurate results than ALS, even when the tolerance on the value of f was set as

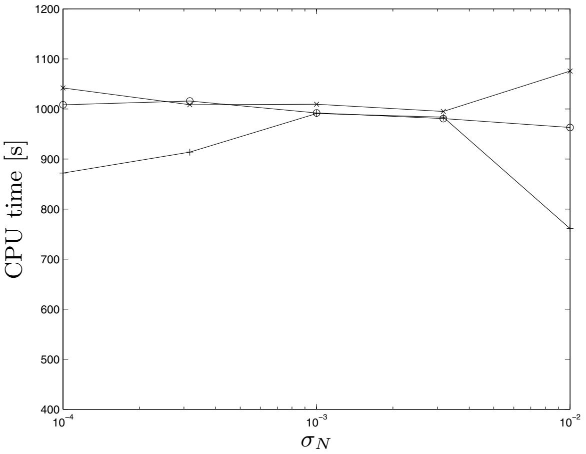

FIG. 9.10. *Total CPU time, over* 50 *Monte Carlo runs and* 10 *random initializations per run, required by the ALS routine. The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(○),* 10 *(×), or* 100 *(+).*

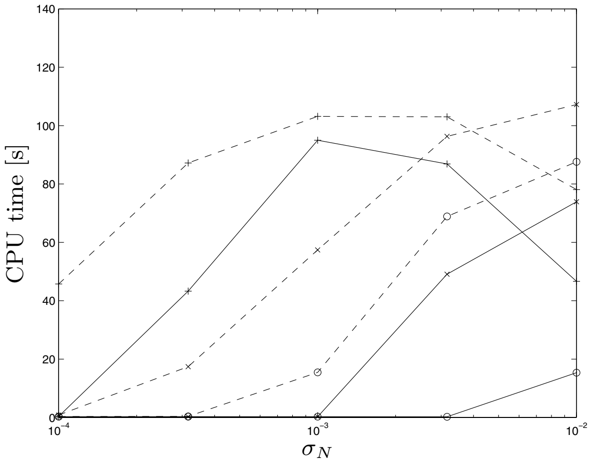

FIG. 9.11. *Total CPU time, over* 50 *Monte Carlo runs, required by the ALS iteration following a SGSD (solid) or a single EVD (dashed). The condition number* κ *of* $\mathbf{U}^{(1)}$ *is equal to* 2 *(○),* 10 *(×), or* 100 *(+).*

sharp as $1e{-}10$. For well-conditioned problems, the accuracy of Levenberg–Marquardt and ALS may be comparable. However, in this case, Levenberg–Marquardt is typically an order of magnitude more expensive than ALS.

Finally, we have applied Algorithm 1 to a real-life dataset. It concerns a real-valued $(5 \times 10 \times 13)$-tensor representing the displacements of 13 points of the tongue of 5 test persons while pronouncing 10 vowels. A detailed description of these data and their analysis by means of a CANDECOMP can be found in [29]. The dataset can be downloaded from [21].

First, we observed that the two dominant 1-mode, 2-mode, and 3-mode singular values [19] explain 94.5%, 95.4%, and 96.0%, respectively, of the "energy" in the dataset. Therefore we performed a dimensionality reduction by calculating the best rank-$(2, 2, 2)$ approximation of the data tensor before starting the actual CANDECOMP computations, as explained in section 3. The approximation was obtained by means of a higher-order orthogonal iteration, initialized with the truncated HOSVD [20]. The stop criterion consisted of checking if the adjustment of each of the component matrices in an iteration step was below $1e-4$ (Frobenius-norm). The algorithm converged in 5 steps. The approximation contained 92.6% of the energy.

Next, we looked for the least-squares approximation of the $(2{\times}2{\times}2)$-core tensor by a sum of two rank-1 components. We resorted to the Jacobi-technique of section 7.2, in which the solution was found by rooting a polynomial of degree 8. The error of the fit was in the order of the numerical accuracy of MATLAB. Backtransformation to the original dimensionality by multiplication with the best rank-$(2,2,2)$ components yielded the best rank-2 approximation of the original dataset. Further enhancement by an additional minimization of cost function (2.3) was not possible (step 5 in Algorithm 1); otherwise, the rank-$(2,2,2)$ approximation would not have been optimal or the core tensor would not have been rank-2.

The cosine of the angle in $\mathbb{R}^{5\times 10\times 13}$ between the original data tensor and its rank-2 approximation was equal to 0.962, which was even slightly better than the result of [29] (0.956); the latter result had been obtained by repeating ALS iterations for different rank estimates and different starting values, and cross-examining the results. On a SUN Ultra 2 Sparc and using MATLAB 4.2c, our computations took $0.2+0.04s$ of CPU-time, which was a drastic improvement [37].

10. Conclusion. In this paper we have investigated the computation of the CANDECOMP, under the assumptions made in section 2. Currently, the calculation of the factors mostly takes the form of an ALS descent algorithm, possibly initialized with an estimate obtained by a matrix EVD. For well-conditioned problems ALS iterations are reliable. However, for some ill-conditioned problems the results are less satisfactory. In this paper the CANDECOMP is computed via a simultaneous diagonalization, by equivalence or congruence, of a set of matrices. Since we take all the available information into account, this is numerically more reliable than the calculation of a single EVD. Diagonalization by a simultaneous congruence transformation was encountered as well in the derivation of an analytical constant modulus algorithm [48], where it was translated into a SGSD and subsequently solved by means of an extended QZ-iteration scheme. In this paper, we have also proposed a Jacobi-type algorithm. In this context we have derived the explicit solution for the case of rank-2 tensors. The behavior of the different algorithms was illustrated and their performance compared by means of some numerical experiments. In this paper we have also studied necessary and sufficient conditions for the uniqueness of some simultaneous matrix decompositions; in addition, we have performed a first-order perturbation analysis of the SGSD.

Acknowledgment. The authors wish to thank Dr. J. Dehaene (K.U.Leuven) for explaining the basic principles underlying the derivation in section 6.2.

REFERENCES

[1] C.J. Appellof and E.R. Davidson, *Strategies for analyzing data from video fluoromatric monitoring of liquid chromatographic effluents*, Analytical Chemistry, 53 (1981), pp. 2053–2056.

[2] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, *A blind source separation technique using second-order statistics*, IEEE Trans. Signal Process., 45 (1997), pp. 434–444.

[3] R. Bro, *PARAFAC. Tutorial & applications*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 149–171.

[4] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, *Numerical methods for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 927–949.

[5] D. Burdick, X. Tu, L. McGown, and D. Millican, *Resolution of multicomponent fluorescent mixtures by analysis of the excitation-emission-frequency array*, J. Chemometrics, 4 (1990), pp. 15–28.

[6] J.-F. Cardoso, *Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Toronto, Canada, 1991, pp. 3109–3112.

[7] J.-F. Cardoso, *Iterative techniques for blind source separation using only fourth-order cumulants*, in Signal Processing VI: Theories and Applications, Proc. EUSIPCO-92, Brussels, Belgium, 1992, pp. 739–742.

[8] J.-F. Cardoso and A. Souloumiac, *Blind beamforming for non-Gaussian signals*, IEE Proc.-F, 140 (1994), pp. 362–370.

[9] J.-F. Cardoso and A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 161–164.

[10] J. Carroll and J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 9 (1970), pp. 267–283.

[11] J. Carroll, G. Desoete, and S. Pruzansky, *An evaluation of five algorithms for generating an initial configuration for SINDSCAL*, J. Classification, 6 (1989), pp. 105–119.

[12] M.T. Chu, *A continuous Jacobi-like approach to the simultaneous reduction of real matrices*, Linear Algebra Appl., 147 (1991), pp. 75–96.

[13] P. Comon, *Independent component analysis, a new concept?* Signal Process., 36 (1994), pp. 287–314.

[14] P. Comon and B. Mourrain, *Decomposition of quantics in sums of powers of linear forms*, Signal Process., 53 (1996), pp. 93–108.

[15] J. Dehaene, *Continuous-Time Matrix Algorithms, Systolic Algorithms and Adaptive Neural Networks*, Ph.D. Thesis, E.E. Dept. (ESAT), K.U.Leuven, Belgium, 1995.

[16] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Independent component analysis based on higher-order statistics only*, in Proceedings of the IEEE Signal Processing Workshop on Statistical Signal and Array Processing, Corfu, Greece, 1996, pp. 356–359.

[17] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K.U. Leuven, E.E. Dept. (ESAT), Belgium, 1997.

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Jacobi-algorithm for simultaneous generalized Schur decomposition in higher-order-only ICA*, in Proceedings of the IEEE Benelux Signal Processing Chapter Signal Processing Symposium (SPS'98), Leuven, Belgium, 1998, pp. 67–70.

[19] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[20] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[21] B. De Moor, Ed., *Database for the Identification of Systems (DAISY)*, E.E. Dept., ESAT/SCD, K.U.Leuven, Belgium, http://www.esat.kuleuven.ac.be/sista/daisy/.

[22] A. Edelman, T.A. Arias, and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[23] B. Flury, *Common Principal Components & Related Multivariate Models*, John Wiley & Sons, New York, 1988.

[24] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[25] M. Haardt, K. Hüper, J. Moore, and J. Nossek, *Simultaneous Schur Decomposition of several matrices to achieve automatic pairing in multidimensional harmonic retrieval problems*, in Signal Processing VIII: Theories and Applications, Proceedings of EUSIPCO-96, Trieste, Italy, 1996, Vol. 1, pp. 531–534.

[26] R. Harshman, *Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[27] R.A. Harshman and M.E. Lundy, *The PARAFAC model for three-way factor analysis and multidimensional scaling*, in Research Methods for Multimode Data Analysis, H.G. Law, C.W. Snyder, J.A. Hattie, and R.P. McDonald, eds., Praeger, NY, 1984, pp. 122–215.

[28] R.A. Harshman and M.E. Lundy, *PARAFAC: Parallel factor analysis*, Comput. Statist. Data

Anal., 18 (1994), pp. 39–72.

[29] R. Harshman, P. Ladefoged, and L. Goldstein, *Factor analysis of tongue shapes*, J. Acoust. Soc. Am., 62 (1977), pp. 693–707.

[30] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.

[31] H. Kiers and W. Krijnen, *An efficient algorithm for PARAFAC of three-way data with large numbers of observational units*, Psychometrika, 56 (1991), pp. 147–152.

[32] E. Kofidis and P.A. Regalia, *On the best rank-*1 *approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 863–884.

[33] T. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.

[34] 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.

[35] J.B. Kruskal, *Rank, decomposition, and uniqueness for* 3*-way and N-way arrays*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 7–18.

[36] J.B. Kruskal, R.A. Harshman, and M.E. Lundy, *How* 3*-MFA data can cause degenerate PARAFAC solutions, among other relationships*, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 115–122.

[37] P. Ladefoged, *personal communication*, Linguistics Dept., UCLA, Los Angeles, CA, 1996.

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

[39] J.J. More, *The Levenberg-Marquardt algorithm: implementation and theory*, in Numerical Analysis, Lecture Notes in Math. 630, G.A. Watson, ed., Springer-Verlag, Berlin, 1977, pp. 105–116.

[40] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 223–242.

[41] W.S. Rayens and B.C. Mitchell, *Two-factor degeneracies and a stabilization of PARAFAC*, Chemom. Intell. Lab. Syst., 38 (1997), pp. 173–181.

[42] E. Sanchez and B.R. Kowalski, *Tensorial resolution: A direct trilinear decomposition*, J. Chemometrics, 4 (1990), pp. 29–45.

[43] R. Sands and F. Young, *Component models for three-way data: An alternating least squares algorithm with optimal scaling features*, Psychometrika, 45 (1980), pp. 39–67.

[44] R. Schmidt, *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, Ph.D. thesis, Stanford University, 1981.

[45] N. Sidiropoulos, G. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810–823.

[46] N. Sidiropoulos, R. Bro, and G. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Trans. Signal Process., 48 (2000), pp. 2377–2388.

[47] N. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of N-way arrays*, J. Chemometrics, 14 (2000), pp. 229–239.

[48] A.-J. van der Veen and A. Paulraj, *An analytical constant modulus algorithm*, IEEE Trans. Signal Process., 44 (1996), pp. 1136–1155.

[49] A. Yeredor, *Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation*, IEEE Trans. Signal Process., 50 (2002), pp. 1545–1553.

[50] T. Zhang and G.H. Golub, *Rank-one approximation to high order tensors*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 534–550.