ELSEVIER

Chemometrics and Intelligent Laboratory Systems 36 (1997) 31–40

Chemometrics and intelligent laboratory systems

Relating two proposed methods for speedup of algorithms for fitting two- and three-way principal component and related multilinear models

Henk A.L. Kiers [a,*], Richard A. Harshman [b]

[a] *Department of Psychology (SPA), University of Groningen, Grote Kruisstraat 2 / 1, 9712 TS Groningen, The Netherlands*
[b] *Department of Psychology, University of Western Ontario, Social Science Centre, London, Ontario, Canada N6A 5C2*

Received 23 December 1994; accepted 4 November 1996

Abstract

Multilinear analysis methods such as component (and three-way component) analysis of very large data sets can become very computationally demanding and even infeasible unless some method is used to compress the data and/or speed up the algorithms. We discuss two previously proposed speedup methods. (a) Alsberg and Kvalheim have proposed use of data simplification along with some new analysis algorithms. We show that their procedures solve the same problem as (b) the more general approach proposed (in a different context) by Carroll, Pruzansky, and Kruskal. In the latter approach, a speed improvement is attained by applying *any* (three-mode) PCA algorithm to a small (three-way) array derived from the original data. Hence, it can employ the new algorithms by Alsberg and Kvalheim, but, as is shown in the present paper, it is easier and often more efficient to apply standard (three-mode) PCA algorithms to the small array. Finally, it is shown how the latter approach for speed improvement can also be used for other three-way models and analysis methods (e.g., PARAFAC/CANDECOMP and constrained three-mode PCA).

Keywords: Principal component analysis; Multilinear models; Two- and three-way principal component model

1. Introduction

In many chemical applications of three-way methods, the size of the three-way array is very large, and hence the procedures require a great deal of computation. Often, however, the data can be compressed considerably without losing significant information. In such cases, one can greatly reduce the amount of intermediate computation required by performing most of the analysis steps on a simplified or compressed version of the data array. For this purpose, Alsberg and Kvalheim [1,2] proposed the 'postponed basis matrix multiplication' (PBM) method. In their papers, they use the PBM method for speeding up the algorithms of principal components analysis (PCA) and (least squares) three-mode PCA when applied to very large data sets. The essential idea of the PBM method is to project the data on certain bases in such a way that the projected version of the data retains the essential information in the original data but has a simpler internal structure, with lower dimensionali-

[*] Corresponding author. Tel.: +31-50-636339; e-mail: h.a.l.kiers@ppsw.rug.nl.

PII S0169-7439(96)00074-3

ties of the spaces spanned by columns and or rows (and or fibers in the third way, if any) of the data array. They then show how this simplified data structure allows most computations to be performed on an array of values that is much smaller than the original data.

Suppose, for example, $\mathbf{X}$ denotes an $I \times J$ matrix which is too large to analyze conveniently by conventional PCA. The PBM approach is to project $\mathbf{X}$ on (a priori chosen) basis matrices $\mathbf{B}_1$ $(I \times I_c)$ and $\mathbf{B}_2$ $(J \times J_c)$, where, typically, $I_c \ll I$ and $J_c \ll J$. This projection yields a compressed $(I_c \times J_c)$ data matrix $\mathbf{C} = \mathbf{B}_1^+ \mathbf{X}_F \mathbf{B}_2^{+\mathrm{T}}$, where $\mathbf{B}_j^+ = (\mathbf{B}_j^{\mathrm{T}} \mathbf{B}_j)^{-1} \mathbf{B}_j^{\mathrm{T}}$, denoting the Moore–Penrose inverse of $\mathbf{B}_j$, $j = 1, 2$. The matrix $\mathbf{C}$ is used to construct a simplified approximation of $\mathbf{X}$, here called $\tilde{\mathbf{X}}$, as follows:

$$\tilde{\mathbf{X}} = \mathbf{B}_1 \mathbf{C}_F \mathbf{B}_2^{\mathrm{T}}. \tag{1}$$

To reduce computation, the PBM method uses the simplified $\tilde{\mathbf{X}}$ rather than the original $\mathbf{X}$. Of course, the choice of basis matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ must be done in such a way that $\tilde{\mathbf{X}}$ approximates the original data well, or even perfectly; for some useful choices of the basis matrices we refer the reader to Alsberg and Kvalheim [1,2]. Once the basis matrices and the matrix $\mathbf{C}$ are obtained, the PBM method uses the especially derived algorithms proposed by Alsberg and Kvalheim (from now on abbreviated as A&K) to perform a PCA in which the computations only involve the relatively small array $\mathbf{C}$.

When introducing their PBM approach for the two-way case, A&K ([1], p. 32) mention that a simple and tempting procedure would be to perform a PCA on $\mathbf{C}$ and obtain the actual solution by premultiplying the resulting matrices by the original basis matrices. They emphasize (p. 32) that this procedure produces the true scores and loadings only when the basis matrices are columnwise orthonormal, which they mention to almost never be the case (p. 33). This motivated them to develop a method ''which can be applied to rewrite algorithms to compensate for the distortion effects observed when using a coefficient matrix $\mathbf{C}$ instead of the original matrix $\mathbf{X}$'' (p. 33). However, they apparently overlooked a very simple way out of their problem: Rather than using their actual nonorthogonal basis matrices, one can just as well use columnwise orthonormal basis matrices that span the same spaces and perform a PCA on the ensuing $\mathbf{C}$ (as will be demonstrated in the present paper). In doing so, one ensures that the method does produce the true scores and loadings. Thus, rather than developing a new, and relatively complicated, approach, it suffices to simply replace the original basis matrices by orthonormal basis matrices for their column spaces.

In a subsequent article (Alsberg and Kvalheim [2]), the PBM method is generalized to PCA of three-way arrays (three-mode PCA). Specifically, suppose that $\mathbf{X}$ denotes an $I \times J \times K$ three-way array with frontal planes $\mathbf{X}_1, \ldots, \mathbf{X}_K$. The frontal planes are collected in the supermatrix $\mathbf{X}_F = (\mathbf{X}_1 | \ldots | \mathbf{X}_K)$. Then, the PBM method projects the array $\mathbf{X}$ on (a priori chosen) basis matrices $\mathbf{B}_1$ $(I \times I_c)$, $\mathbf{B}_2$ $(J \times J_c)$ and $\mathbf{B}_3$ $(K \times K_c)$ where, typically, $I_c \ll I$, $J_c \ll J$ and $K_c \ll K$. This projection yields a three-way array $\tilde{\mathbf{X}}$ and a $I_c \times J_c \times K_c$ regression weights array $\mathbf{C}$ (with frontal planes collected in $\tilde{\mathbf{X}}_F$ and $\mathbf{C}_F$, respectively) for which

$$\tilde{\mathbf{X}}_F = \mathbf{B}_1 \mathbf{C}_F \left(\mathbf{B}_3^{\mathrm{T}} \otimes \mathbf{B}_2^{\mathrm{T}} \right). \tag{2}$$

Here, $\mathbf{C}_F$ is obtained as $\mathbf{C}_F = \mathbf{B}_1^+ \mathbf{X}_F (\mathbf{B}_3^{+\mathrm{T}} \otimes \mathbf{B}_2^{+\mathrm{T}})$. The choice of the basis matrices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ must again be done such that $\tilde{\mathbf{X}}_F$ in Eq. (2), with $\mathbf{C}_F$ considerably smaller than $\mathbf{X}_F$, describes the original data (in $\mathbf{X}_F$) well, or even perfectly. As in the two-way case, A&K propose to use the projected array $\tilde{\mathbf{X}}$ rather than $\mathbf{X}$ for their analyses. Just as in the two-way case, however, it suffices to replace the basis matrices by columnwise orthonormal basis matrices, and apply three-mode PCA to the ensuing $\mathbf{C}$ (as will be shown in the present paper).

As will be shown in the present paper, the PBM approach as well as the above sketched alternative to the PBM approach are variants of a general procedure proposed by Carroll, Pruzansky and Kruskal (abbreviated as CPK henceforth) [3]. CPK describe a procedure for fitting a constrained version of the PARAFAC/CANDECOMP [4,5] model, which they call CANDELINC. They mention ([3], p. 10) how their method can also be used to fit three-mode PCA models where the component matrices are constrained, and they describe (p. 17) how the method can be used as an efficient way of obtaining the unconstrained (symmetric) PARAFAC solution (see also [6]).

In the present paper, we will describe the CPK approach, both for the two-way case and for the three-way case, and we will show that the above sketched alternative to the PBM approach is in fact a straightforward implementation of the CPK approach. Furthermore, it will be made clear that the PBM approach solves the same optimization problem as the CPK approach, however, employing a different, more complicated algorithm. In fact, it will be shown that we can use any (three-mode) PCA algorithm with the CPK approach, which obviates the need for new algorithms (like those proposed by A&K). Moreover, as will be shown, the three-mode PCA algorithm that A&K [2] took as their basis is not the most efficient one available for three-mode PCA. The present results allow us to replace their algorithm by a more efficient (three-mode) PCA algorithm for analyzing **C**. Finally, it will be shown that the CPK approach has more general applicability, in that it allows for similar speed improvements for other methods, like PARAFAC/CANDECOMP [3–5] and constrained three-mode PCA [7].

In Section 2, we will discuss the two-way case and relate the A&K method in [1] to the CPK approach. In Section 3 we discuss the three-way case and relate the A&K method in [2] to the CPK approach, as well as discuss some further possibilities of the CPK approach for three-way methods.

2. The CPK approach for PCA and its relation to the PBM approach

In the present section the CPK approach will be described for the two-way case. It will be shown that the algorithm proposed by A&K [1] is a variant of the CPK approach, giving exactly the same solution by a different algorithm.

The aim of PCA is to approximate an $I \times J$ matrix $\mathbf{X}$ by a matrix of low rank written as $\mathbf{TP}^{\mathrm{T}}$. Here $\mathbf{T}$ is an $I \times s$ matrix and $\mathbf{P}$ is a $J \times s$ loading matrix; following A&K [1], we constrain $\mathbf{P}$ such that $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}_s$. Thus, PCA can be considered as the minimization of

$$f(\mathbf{T}, \mathbf{P}) = \|\mathbf{X} - \mathbf{TP}^{\mathrm{T}}\|^2 \tag{3}$$

over $\mathbf{T}$ and $\mathbf{P}$ subject to $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}_s$. To identify the solution, PCA takes the matrices $\mathbf{T}$ and $\mathbf{P}$ orthogonal and orders the components with respect to the variance they explain. In the CPK approach, the matrices $\mathbf{T}$ and $\mathbf{P}$ are further constrained: They are constrained to be in the column spaces of certain 'external' matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ that are, for instance, obtained from previous research, or derived from theory; in the present context, the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ are to be chosen such that $\mathbf{B}_1\mathbf{CB}_2^{\mathrm{T}}$ describes $\mathbf{X}$ well (as in A&K [1]). Because of these constraints, we have $\mathbf{T} = \mathbf{B}_1\mathbf{U}$ and $\mathbf{P} = \mathbf{B}_2\mathbf{V}$ for certain (as yet unknown) matrices $\mathbf{U}$ and $\mathbf{V}$. Let $\mathbf{Q}_1$ and $\mathbf{Q}_2$ denote orthonormal bases for the column spaces of $\mathbf{B}_1$ and $\mathbf{B}_2$, respectively, and let $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ denote the matrices such that $\mathbf{T} = \mathbf{Q}_1\tilde{\mathbf{U}}$ and $\mathbf{P} = \mathbf{Q}_2\tilde{\mathbf{V}}$. Then, as shown by CPK [3], the problem of minimizing Eq. (3) over $\mathbf{T}$ and $\mathbf{P}$, subject to the above linear subspace constraints, is equivalent to the problem of minimizing

$$g(\tilde{\mathbf{U}}, \tilde{\mathbf{V}}) = \|\mathbf{Q}_1^{\mathrm{T}}\mathbf{XQ}_2 - \tilde{\mathbf{U}}\tilde{\mathbf{V}}^{\mathrm{T}}\|^2 \tag{4}$$

over $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$, and obtaining $\mathbf{T}$ as $\mathbf{Q}_1\tilde{\mathbf{U}}$ and $\mathbf{P}$ as $\mathbf{Q}_2\tilde{\mathbf{V}}$. The optimal $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$, denoted by $\tilde{\mathbf{U}}_s$ and $\tilde{\mathbf{V}}_s$ in the s-dimensional case, can be obtained by any PCA algorithm applied to $\tilde{\mathbf{C}} \equiv \mathbf{Q}_1^{\mathrm{T}}\mathbf{XQ}_2$, for instance by obtaining the SVD of $\tilde{\mathbf{C}}$. The resulting components matrix is $\mathbf{T}_s = \mathbf{Q}_1\tilde{\mathbf{U}}_s$ and the loading matrix is $\mathbf{P}_s = \mathbf{Q}_2\tilde{\mathbf{V}}_s$. Because the matrix $\tilde{\mathbf{C}}$ is (much) smaller than the original matrix, a considerable speed improvement is attained by using the procedure for constrained rather than unconstrained fitting of $\mathbf{X}$. Admittedly, this speed improvement is attained at the cost of imposing a restriction on the solution. However, as noted by CPK, for well chosen bases $\mathbf{B}_1$ and $\mathbf{B}_2$, the constrained solution will give a good approximation to the unconstrained solution.

The PBM method is related to the CPK approach as follows: The PBM method also relies on the choice of useful subspaces $\mathbf{B}_1$ and $\mathbf{B}_2$, and on the assumption that $\mathbf{T} = \mathbf{B}_1\mathbf{U}$ and $\mathbf{P} = \mathbf{B}_2\mathbf{V}$ for certain (as yet unknown) matrices $\mathbf{U}$ and $\mathbf{V}$. Obviously, in practice, this assumption will often be satisfied only approximately, hence, in practice this assumption represents a (mild) constraint on the solution. In contrast to the CPK method, the PBM method was not described as a PCA on a small sized matrix, but as PCA on a matrix $\tilde{\mathbf{X}}$ (of the same size as $\mathbf{X}$) which results after projecting $\mathbf{X}$ on the bases $\mathbf{B}_1$ and $\mathbf{B}_2$, according to $\tilde{\mathbf{X}} = \mathbf{B}_1\mathbf{CB}_2^{\mathrm{T}}$ (Eq. (1)), with $\mathbf{C} = \mathbf{B}_1^{+}\mathbf{XB}_2^{+\mathrm{T}}$. The PBM method is based on replacing $\mathbf{X}$ by its approximation $\tilde{\mathbf{X}} = \mathbf{B}_1\mathbf{C}\mathbf{B}_2^{\mathrm{T}}$, and thus minimizing

$$f(\mathbf{P},\mathbf{T}) = \|\mathbf{B}_1\mathbf{C}\mathbf{B}_2^{\mathrm{T}} - \mathbf{P}\mathbf{T}^{\mathrm{T}}\|^2. \tag{5}$$

The algorithm A & K use for minimizing Eq. (5) is an ordinary (iterative) PCA algorithm applied to $\mathbf{B}_1\mathbf{C}\mathbf{B}_2^{\mathrm{T}}$, in which, however, the order of the computations is modified. This modification is based on the assumption that $\mathbf{T}$ and $\mathbf{P}$ are in the column spaces of $\mathbf{B}_1$ and $\mathbf{B}_2$ (see [1], p. 33), respectively. Using this assumption, A & K showed that in their PCA algorithm premultiplications with the large matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ could be postponed until the very end of the iterative algorithm, and that thus a considerable speed improvement can be obtained. The PBM method uses this postponed basis multiplication. It will now be shown that, implicitly, the PBM method also performs a PCA of a *small* matrix, and that it, in fact, solves the same optimization problem as the CPK approach does, however, using a different algorithm.

Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be replaced by any matrix products $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$ and $\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2$, with $\mathbf{Q}_1$ and $\mathbf{Q}_2$ columnwise orthonormal. For this purpose, one could use, for example, the singular value decompositions of $\mathbf{B}_1$ and $\mathbf{B}_2$ (and hence $\mathbf{R}_1$ and $\mathbf{R}_2$ are columnwise orthogonal matrices), or the QR decompositions (see [8], p. 211 ff) of $\mathbf{B}_1$ and $\mathbf{B}_2$, in which case $\mathbf{R}_1$ and $\mathbf{R}_2$ are upper triangular. Upon substituting $\mathbf{C} = \mathbf{B}_1^{+}\mathbf{X}\mathbf{B}_2^{+\mathrm{T}} = (\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1)^{-1}\mathbf{B}_1^{\mathrm{T}}\mathbf{X}\mathbf{B}_2(\mathbf{B}_2^{\mathrm{T}}\mathbf{B}_2)^{-1}$, $\mathbf{T} = \mathbf{B}_1\mathbf{U}$ and $\mathbf{P} = \mathbf{B}_2\mathbf{V}$ in Eq. (5), and replacing $\mathbf{B}_1$ and $\mathbf{B}_2$ by their (QR) decompositions, we obtain

$$\begin{aligned} f(\mathbf{U},\mathbf{V}) &= \|\mathbf{B}_1(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_1)^{-1}\mathbf{B}_1^{\mathrm{T}}\mathbf{X}\mathbf{B}_2(\mathbf{B}_2^{\mathrm{T}}\mathbf{B}_2)^{-1}\mathbf{B}_2^{\mathrm{T}} \\ &\quad - \mathbf{B}_1\mathbf{U}\mathbf{V}^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\|^2 = \|\mathbf{Q}_1\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2\mathbf{Q}_2^{\mathrm{T}} \\ &\quad - \mathbf{Q}_1\mathbf{R}_1\mathbf{U}\mathbf{V}^{\mathrm{T}}\mathbf{R}_2^{\mathrm{T}}\mathbf{Q}_2^{\mathrm{T}}\|^2 = \|\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2 \\ &\quad - \mathbf{R}_1\mathbf{U}\mathbf{V}^{\mathrm{T}}\mathbf{R}_2^{\mathrm{T}}\|^2 \end{aligned} \tag{6}$$

(compare [9]), where it is used that $\|\mathbf{Q}_1\mathbf{E}\mathbf{Q}_2^{\mathrm{T}}\|^2 = \mathrm{tr}(\mathbf{Q}_1\mathbf{E}\mathbf{Q}_2^{\mathrm{T}}\mathbf{Q}_2\mathbf{E}^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}) = \mathrm{tr}(\mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_1\mathbf{E}\mathbf{Q}_2^{\mathrm{T}}\mathbf{Q}_2\mathbf{E}^{\mathrm{T}}) = \mathrm{tr}(\mathbf{E}^{\mathrm{T}}\mathbf{E}) = \|\mathbf{E}\|^2$ for any $I_c \times J_c$ matrix $\mathbf{E}$. Because $\mathbf{R}_1$ and $\mathbf{R}_2$ are nonsingular, the problem of minimizing Eq. (6) is equivalent to minimizing

$$f(\tilde{\mathbf{T}},\tilde{\mathbf{P}}) = \|\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2 - \tilde{\mathbf{T}}\tilde{\mathbf{P}}^{\mathrm{T}}\|^2, \tag{7}$$

where $\tilde{\mathbf{T}} \equiv \mathbf{R}_1\mathbf{U}$ and $\tilde{\mathbf{P}} \equiv \mathbf{R}_2\mathbf{V}$, over arbitrary $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{P}}$, which is equivalent to minimizing Eq. (4) in the CPK approach. The problem of minimizing Eq. (7) comes down to finding the first s principal components of $\mathbf{C} = \mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2$. When the s-dimensional solution for $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{P}}$ is given by $\tilde{\mathbf{T}}_s$ and $\tilde{\mathbf{P}}_s$, the s-dimensional solutions for the matrices $\mathbf{T}$ and $\mathbf{P}$ are given by $\mathbf{T}_s \equiv \mathbf{B}_1\mathbf{U}_s = \mathbf{B}_1\mathbf{R}_1^{-1}\tilde{\mathbf{T}}_s = \mathbf{Q}_1\tilde{\mathbf{T}}_s$ and $\mathbf{P}_s \equiv \mathbf{B}_2\mathbf{V}_s = \mathbf{B}_2\mathbf{R}_2^{-1}\tilde{\mathbf{P}}_s = \mathbf{Q}_2\tilde{\mathbf{P}}_s$, which is equivalent to the solution obtained by the CPK approach. Hence, the PBM method solves the same minimization problem as does the CPK approach, using a somewhat indirect algorithm (instead of applying a standard PCA procedure to $\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2$).

Having seen that the PBM method and the CPK approach solve the same PCA problem, one may still wonder if the CPK solution fulfills the same conditions for identification of the parameters as the PBM method (i.e., having $\mathbf{T}_s$ columnwise orthogonal, and $\mathbf{P}_s^{\mathrm{T}}\mathbf{P}_s = \mathbf{I}_s$, see A & K [1], p. 35). In the CPK approach, the components matrix is given by $\mathbf{T}_s = \mathbf{Q}_1\tilde{\mathbf{U}}_s$ and the loading matrix by $\mathbf{P}_s = \mathbf{Q}_2\tilde{\mathbf{V}}_s$, where $\tilde{\mathbf{U}}_s$ and $\tilde{\mathbf{V}}_s$ minimize Eq. (4). When these matrices are obtained by an SVD, or a similar method which finds $\tilde{\mathbf{U}}_s$ as the first s principal components of $\tilde{\mathbf{C}}$, we have $\tilde{\mathbf{U}}_s^{\mathrm{T}}\tilde{\mathbf{U}}_s$ is diagonal and $\tilde{\mathbf{V}}_s^{\mathrm{T}}\tilde{\mathbf{V}}_s = \mathbf{I}_s$. As a result, $\mathbf{T}_s^{\mathrm{T}}\mathbf{T}_s = \tilde{\mathbf{U}}_s^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_1\tilde{\mathbf{U}}_s = \tilde{\mathbf{U}}_s^{\mathrm{T}}\tilde{\mathbf{U}}_s$ is diagonal, and $\mathbf{P}_s^{\mathrm{T}}\mathbf{P}_s = \tilde{\mathbf{V}}_s^{\mathrm{T}}\mathbf{Q}_2^{\mathrm{T}}\mathbf{Q}_2\tilde{\mathbf{V}}_s = \tilde{\mathbf{V}}_s^{\mathrm{T}}\tilde{\mathbf{V}}_s = \mathbf{I}_s$. Hence the CPK component matrix is columnwise orthogonal and the associated loading matrix is columnwise orthonormal, just as in the PBM method. In fact, it can be verified that $\mathbf{T}_s$ contains the first s principal components of $\tilde{\mathbf{X}}$, as follows. Because $\tilde{\mathbf{U}}_s$ contains the first s principal components of $\tilde{\mathbf{C}}$, we have $\tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\mathrm{T}}\tilde{\mathbf{U}}_s = \tilde{\mathbf{U}}_s\Lambda$ for a diagonal matrix Λ with the first s eigenvalues of $\tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\mathrm{T}}$. It follows that $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T}_s = \mathbf{Q}_1\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2\mathbf{Q}_2^{\mathrm{T}}\mathbf{Q}_2\mathbf{Q}_2^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Q}_1\mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_1\tilde{\mathbf{U}}_s = \mathbf{Q}_1\tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\mathrm{T}}\tilde{\mathbf{U}}_s = \mathbf{Q}_1\tilde{\mathbf{U}}_s\Lambda = \mathbf{T}_s\Lambda$, which shows that $\mathbf{T}_s$ contains the principal components of $\tilde{\mathbf{X}}$. Thus it has been shown that the CPK approach, when based on principal components of $\tilde{\mathbf{C}}$ has the same properties as the PBM method.

The CPK approach can be summarized as follows:

Step 1. Compute the decompositions $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$ and $\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2$.

Step 2. Compute $\tilde{\mathbf{C}} = \mathbf{Q}_1^{\mathrm{T}}\mathbf{X}\mathbf{Q}_2$.

Step 3. Obtain $\tilde{\mathbf{U}}_s$ and $\tilde{\mathbf{V}}_s$ by applying a PCA algorithm to $\tilde{\mathbf{C}}$.

Step 4. Compute the PCA solution of $\mathbf{X}$ as $\mathbf{T}_s = \mathbf{Q}_1\tilde{\mathbf{U}}_s$ and $\mathbf{P}_s = \mathbf{Q}_2\tilde{\mathbf{V}}_s$.

It should be noted that the decompositions in Step 1 can be obtained by any procedure that gives an orthonormal basis for $\mathbf{B}_1$ and $\mathbf{B}_2$. If $\mathbf{Q}_\mathrm{R}$ decompositions are used, one can compute these indirectly by computing $\mathbf{R}_1$ and $\mathbf{R}_2$ from the Cholesky decompositions $\mathbf{B}_1^\mathrm{T}\mathbf{B}_1 = \mathbf{R}_1^\mathrm{T}\mathbf{R}_1$ and $\mathbf{B}_2^\mathrm{T}\mathbf{B}_2 = \mathbf{R}_2^\mathrm{T}\mathbf{R}_2$, and computing $\mathbf{Q}_1 = \mathbf{B}_1\mathbf{R}_1^{-1}$ and $\mathbf{Q}_2 = \mathbf{B}_2\mathbf{R}_2^{-1}$. Step 3 can be performed by using the power algorithm successively, or by using any other algorithm for computing the first s eigenvectors of a matrix (as done by, e.g., the Bauer–Rutishauser algorithm, see [10]), or even by computing a complete eigendecomposition of $\tilde{\mathbf{C}}^\mathrm{T}\tilde{\mathbf{C}}$ or $\tilde{\mathbf{C}}\tilde{\mathbf{C}}^\mathrm{T}$, depending on which is smaller.

An interesting special case of the above algorithm is where $\mathbf{B}_1$ and $\mathbf{B}_2$ are columnwise orthonormal. Then $\mathbf{B}_1 = \mathbf{Q}_1$ and $\mathbf{B}_2 = \mathbf{Q}_2$, and the CPK approach reduces to the first approach described by A&K [1] in their formulas Eqs. (2)–(6). Above, it has been mentioned that the PBM algorithm by A&K [1] uses a somewhat indirect algorithm for minimizing Eq. (7). Specifically, their algorithm is based on successively computing the different columns of the matrices $\mathbf{U}_s$ and $\mathbf{V}_s$ (which are used to define $\mathbf{T}_s = \mathbf{B}_1\mathbf{U}_s$ and $\mathbf{P}_s = \mathbf{B}_2\mathbf{V}_s$) as follows:

```
Step 1′. Compute $\mathbf{C} = \mathbf{B}_1^+\mathbf{X}\mathbf{B}_2^{+\mathrm{T}}$.
Step 2′. Compute $\boldsymbol{\Gamma}_1 = \mathbf{B}_1^\mathrm{T}\mathbf{B}_1$ and $\boldsymbol{\Gamma}_2 = \mathbf{B}_2^\mathrm{T}\mathbf{B}_2$.
For $r = 1, \ldots, s$
Step 3a′. Initialize $\mathbf{u}_r^0$
Step 3b′. $\mathbf{v}_r^0 := \mathbf{C}^\mathrm{T}\boldsymbol{\Gamma}_1\mathbf{u}_r^0$
Step 3c′. $\mathbf{v}_r^1 := \mathbf{v}_r^0/(\mathbf{v}_r^{0\mathrm{T}}\boldsymbol{\Gamma}_2\mathbf{v}_r^0)^{1/2}$
Step 3d′. $\mathbf{u}_r^1 := \mathbf{C}\boldsymbol{\Gamma}_2\mathbf{v}_r^1$
If $\|\mathbf{u}_r^0 - \mathbf{u}_r^1\| > \epsilon$, for a small value $\epsilon$, go back to Step 3b′.
Step 3e′. $\mathbf{C} := \mathbf{C} - \mathbf{u}_r^1\mathbf{v}_r^{1\mathrm{T}}$.
Step 4′. Compute the PCA solution of $\mathbf{X}$ as $\mathbf{T}_s = \mathbf{B}_1\mathbf{U}_s$ and $\mathbf{P}_s = \mathbf{B}_2\mathbf{V}_s$.
```

To facilitate comparison with the CPK approach, we describe the CPK approach employing a variant of the power algorithm for obtaining the principal components (in $\tilde{\mathbf{U}}_s$) and loadings (in $\tilde{\mathbf{V}}_s$) of $\tilde{\mathbf{C}}$.

```
Step 1. Compute the (QR) decompositions $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$ and $\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2$.
Step 2. Compute $\tilde{\mathbf{C}} = \mathbf{Q}_1^\mathrm{T}\mathbf{X}\mathbf{Q}_2$.
For $r = 1, \ldots, s$
Step 3a. Initialize $\tilde{\mathbf{u}}_r^0$
Step 3b. $\tilde{\mathbf{v}}_r^0 := \tilde{\mathbf{C}}^\mathrm{T}\tilde{\mathbf{u}}_r^0$
Step 3c. $\tilde{\mathbf{v}}_r^1 := \tilde{\mathbf{v}}_r^0/(\tilde{\mathbf{v}}_r^{0\mathrm{T}}\tilde{\mathbf{v}}_r^0)^{1/2}$
Step 3d. $\tilde{\mathbf{u}}_r^1 := \tilde{\mathbf{C}}\tilde{\mathbf{v}}_r^1$
If $\|\tilde{\mathbf{u}}_r^0 - \tilde{\mathbf{u}}_r^1\| > \epsilon$, for a small value $\epsilon$, go back to Step 2.
Step 3e. $\tilde{\mathbf{C}} := \tilde{\mathbf{C}} - \tilde{\mathbf{u}}_r^1\tilde{\mathbf{v}}_r^{1\mathrm{T}}$.
Step 4. Compute the PCA solution of $\mathbf{X}$ as $\mathbf{T}_s = \mathbf{Q}_1\tilde{\mathbf{U}}_s$ and $\mathbf{P}_s = \mathbf{Q}_2\tilde{\mathbf{V}}_s$.
```

The above CPK algorithm may seem more time consuming then A&K's procedure, because it requires two (QR) decompositions (in Step 1). However, in the A&K procedure, Step 1′ requires the computation of $\mathbf{C} = \mathbf{B}_1^+\mathbf{X}\mathbf{B}_2^{+\mathrm{T}}$, which is efficiently and reliably done by first computing the (QR) decompositions of $\mathbf{B}_1$ and $\mathbf{B}_2$ and then computing $\mathbf{C}$ as $\mathbf{C} = \mathbf{R}_1^{-1}\mathbf{Q}_1^\mathrm{T}\mathbf{X}\mathbf{Q}_2(\mathbf{R}_2^{-1})^\mathrm{T}$, so, in that case, Step 1′ of the A&K algorithm actually requires more time than our Steps 1 and 2 together. As far as the iterative part is concerned, it can be seen that each iteration of the CPK algorithm (Step 3) is more efficient than that of A&K, since it requires fewer matrix multiplications. Finally, Step 4 in the CPK algorithm and Step 4′ in the A&K algorithm require the same amount of flops. Hence, it can be concluded that the CPK approach using a power algorithm is faster than the A&K algorithm, assuming that the algorithms need the same numbers of iterations in Step 3. In fact, it can be shown that the iterations in Steps 3a′, 3b′, 3c′ and 3d′ are related to Steps 3a, 3b, 3c and 3d in a simple way as follows. If the initializations in the two algorithms are chosen such that $\tilde{\mathbf{u}}_r^0 = \mathbf{R}_1\mathbf{u}_r^0$, then, using $\tilde{\mathbf{C}} = \mathbf{Q}_1^\mathrm{T}\mathbf{X}\mathbf{Q}_2 = \mathbf{R}_1\mathbf{R}_1^{-1}\mathbf{Q}_1^\mathrm{T}\mathbf{X}\mathbf{Q}_2(\mathbf{R}_2^{-1})^\mathrm{T}\mathbf{R}_2^\mathrm{T} = \mathbf{R}_1\mathbf{C}\mathbf{R}_2^\mathrm{T}$, we have

$$
\begin{aligned}
\tilde{\mathbf{u}}_r^1 &= \tilde{\mathbf{C}}\tilde{\mathbf{C}}^\mathrm{T}\tilde{\mathbf{u}}_r^0/\left(\tilde{\mathbf{u}}_r^{0\mathrm{T}}\tilde{\mathbf{C}}\tilde{\mathbf{C}}^\mathrm{T}\tilde{\mathbf{u}}_r^0\right)^{1/2}\\
&= \mathbf{R}_1\mathbf{C}\mathbf{R}_2^\mathrm{T}\mathbf{R}_2\mathbf{C}^\mathrm{T}\mathbf{R}_1^\mathrm{T}\mathbf{R}_1\mathbf{u}_r^0\\
&\quad /\left(\mathbf{u}_r^{0\mathrm{T}}\mathbf{R}_1^\mathrm{T}\mathbf{R}_1\mathbf{C}\mathbf{R}_2^\mathrm{T}\mathbf{R}_2\mathbf{C}^\mathrm{T}\mathbf{R}_1^\mathrm{T}\mathbf{R}_1\mathbf{u}_r^0\right)^{1/2}\\
&= \mathbf{R}_1\mathbf{C}\boldsymbol{\Gamma}_2\mathbf{C}^\mathrm{T}\boldsymbol{\Gamma}_1\mathbf{u}_r^0\\
&\quad /\left(\mathbf{u}_r^{0\mathrm{T}}\boldsymbol{\Gamma}_1\mathbf{C}\boldsymbol{\Gamma}_2\mathbf{C}^\mathrm{T}\boldsymbol{\Gamma}_1\mathbf{u}_r^0\right)^{1/2}\\
&= \mathbf{R}_1\mathbf{C}\boldsymbol{\Gamma}_2\mathbf{v}_r^0/\left(\mathbf{v}_r^{0\mathrm{T}}\boldsymbol{\Gamma}_2\mathbf{v}_r^0\right)^{1/2}\\
&= \mathbf{R}_1\mathbf{u}_r^1. \qquad (8)
\end{aligned}
$$

Hence, after any set of complete iterations, we have $\tilde{\mathbf{u}}_r = \mathbf{R}_1\mathbf{u}_r$, so the two algorithms have exactly the same convergence properties. In practice, the one or the other may stop earlier, depending on the initialization and on whether $\|\mathbf{u}_r^0 - \mathbf{u}_r^1\|$ is larger than or smaller than $\|\tilde{\mathbf{u}}_r^0 - \tilde{\mathbf{u}}_r^1\| = \|\mathbf{R}_1\mathbf{u}_r^0 - \mathbf{R}_1\mathbf{u}_r^1\|$, which depends on the data.

Similarly, the advantage of using other algorithms than the power algorithm will depend on the data, as well as on the implementation at hand. For instance, in case one uses MATLAB (as do A&K), the built-in procedure for computing an SVD will usually be much faster than any iterative power like procedure. Therefore, in such cases the CPK procedure will be most valuable, because it can use this built-in routine, whereas the A&K procedure cannot.

Another remark on the efficiency of the algorithms is in order here. A&K mention that, in case the basis matrices are to some extent sparse, their algorithm can fruitfully use this sparseness. In the CPK approach, using the QR decomposition will at least partly eliminate this advantage. However, in that case one should use the Cholesky decompositions of the small matrices $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ to obtain $\mathbf{R}_1$ and $\mathbf{R}_2$, and replace the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ by the (unevaluated) products $\mathbf{B}_1\mathbf{R}_1^{-1}$ and $\mathbf{B}_2\mathbf{R}_2^{-1}$, which do involve the sparse matrices $\mathbf{B}_1$ and $\mathbf{B}_2$. Thus, sparse technology is still usable.

In the present section, we have described the CPK approach for two-way PCA and compared it to the A&K approach. In the next section, we will describe the CPK approach for the analysis of three-way data.

3. The CPK approach for the analysis of three-way data

As observed by A&K [1], the PBM approach can easily be generalized to the N-way case. A&K [2] considered the three-way case in detail, and applied their PBM method to obtain an adjusted algorithm for three-mode PCA. In the present section, the CPK approach to three-way data is described, and it is demonstrated that the A&K procedure for three-mode PCA solves the same optimization problem as does the CPK approach. It is emphasized again that the advantage of the CPK approach is that it can use any algorithm for three-mode PCA, whereas the A&K procedure requires adjusting existing algorithms. Moreover, the CPK approach can be used for other three-way methods as well, among which PARAFAC/CANDECOMP for which it was developed originally. Finally, it is shown that some existing short-cut algorithms for data with one large mode, can be replaced fruitfully by the CPK approach as well.

Let the basis matrices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, for the three respective modes, and their (QR) decompositions $\mathbf{B}_j = \mathbf{Q}_j\mathbf{R}_j$ ($j = 1, 2, 3$) be given. Then, the CPK approach comes down to applying the three-way method at hand to the array with frontal planes collected in $\tilde{\mathbf{C}}_F = \mathbf{Q}_1^{\mathrm{T}}\mathbf{X}_F(\mathbf{Q}_3 \otimes \mathbf{Q}_2)$, and computing the component matrices $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ from those obtained for $\tilde{\mathbf{C}}_F$ (denoted as $\tilde{\mathbf{E}}$, $\tilde{\mathbf{F}}$, and $\tilde{\mathbf{G}}$) as $\mathbf{E} = \mathbf{Q}_1\tilde{\mathbf{E}}$, $\mathbf{F} = \mathbf{Q}_2\tilde{\mathbf{F}}$, and $\mathbf{G} = \mathbf{Q}_3\tilde{\mathbf{G}}$. In case of three-mode PCA, the CPK approach thus minimizes

$$g\left(\tilde{\mathbf{E}}, \tilde{\mathbf{F}}, \tilde{\mathbf{G}}, \mathbf{H}_F\right) = \|\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}_F(\mathbf{Q}_3 \otimes \mathbf{Q}_2) - \tilde{\mathbf{E}}\mathbf{H}_F(\tilde{\mathbf{G}}^{\mathrm{T}} \otimes \tilde{\mathbf{F}}^{\mathrm{T}})\|^2 \qquad (9)$$

where the component matrices $\tilde{\mathbf{E}}$ ($I_c \times P$), $\tilde{\mathbf{F}}$ ($J_c \times Q$) and $\tilde{\mathbf{G}}$ ($K_c \times R$) are usually taken columnwise orthonormal. As shown by CPK [3], this procedure is equivalent to minimizing the original three-mode PCA loss function

$$g(\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}_F) = \|\mathbf{X}_F - \mathbf{E}\mathbf{H}_F(\mathbf{G}^{\mathrm{T}} \otimes \mathbf{F}^{\mathrm{T}})\|^2, \quad (10)$$

subject to the constraints that $\mathbf{E} = \mathbf{Q}_1\tilde{\mathbf{E}}$, $\mathbf{F} = \mathbf{Q}_2\tilde{\mathbf{F}}$, and $\mathbf{G} = \mathbf{Q}_3\tilde{\mathbf{G}}$. This procedure will give a good approximation of the unconstrained three-mode PCA solution if the bases $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{Q}_3$ are well chosen. CPK give a suggestion for this choice (also see [6]). Moreover, because $\mathbf{Q}_j$ ($j = 1, 2, 3$) is columnwise orthonormal, $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ are columnwise orthonormal as soon as $\tilde{\mathbf{E}}$, $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ are. Thus, the CPK results have the same property as the ordinary three-mode PCA solution. The CPK approach can be summarized as follows:

Step 1. Compute (QR) decompositions $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$, $\mathbf{B}_2 = \mathbf{Q}_2\mathbf{R}_2$ and $\mathbf{B}_3 = \mathbf{Q}_3\mathbf{R}_3$.

Step 2. Compute $\tilde{\mathbf{C}}_F = \mathbf{Q}_1^{\mathrm{T}}\mathbf{X}_F(\mathbf{Q}_3 \otimes \mathbf{Q}_2)$.

Step 3. Obtain the three-mode PCA solution for $\tilde{\mathbf{C}}$, denoted as $\tilde{\mathbf{E}}_P$, $\tilde{\mathbf{F}}_Q$, $\tilde{\mathbf{G}}_R$ and $\mathbf{H}_{PQR}$ (where the indices P, Q, and R denote the respective dimensionalities of the solution).

Step 4. Obtain the three-mode PCA solution of $\mathbf{X}$ as $\mathbf{E}_P = \mathbf{Q}_1\tilde{\mathbf{E}}_P$, $\mathbf{F}_Q = \mathbf{Q}_2\tilde{\mathbf{F}}_Q$, $\mathbf{G}_R = \mathbf{Q}_3\tilde{\mathbf{G}}_R$ and $\mathbf{H}_{PQR}$.

The PBM method for three-way methods also relies on the idea of choosing good basis matrices, and, in fact, some suggestions for these have been made

[2]. The main difference between the PBM method and the CPK approach is in the way of obtaining estimates for **E**, **F** and **G**. The PBM method is based on a three-mode PCA algorithm for analyzing the array $\tilde{\mathbf{X}}$ of which the frontal planes are collected in $\tilde{\mathbf{X}}_F = \mathbf{B}_1\mathbf{C}_F(\mathbf{B}_3^{\mathrm{T}} \otimes \mathbf{B}_2^{\mathrm{T}})$. The matrices **E**, **F** and **G** are assumed to be in the column spaces of $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, respectively. The algorithm by A&K differs from the standard algorithm only in that all multiplications with the basis matrices are postponed until after the iterative part of the algorithm. Hence, their method minimizes

$$g(\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}_F) = \|\mathbf{B}_1\mathbf{C}_F(\mathbf{B}_3^{\mathrm{T}} \otimes \mathbf{B}_2^{\mathrm{T}}) - \mathbf{E}\mathbf{H}_F(\mathbf{G}^{\mathrm{T}} \otimes \mathbf{F}^{\mathrm{T}})\|^2, \qquad (11)$$

subject to $\mathbf{E} = \mathbf{B}_1\mathbf{U}$, $\mathbf{F} = \mathbf{B}_2\mathbf{V}$ and $\mathbf{G} = \mathbf{B}_3\mathbf{W}$ for certain matrices **U**, **V** and **W**. We can rewrite Eq. (11) as

$$g(\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{H}_F) = \|\mathbf{B}_1\mathbf{C}_F(\mathbf{B}_3^{\mathrm{T}} \otimes \mathbf{B}_2^{\mathrm{T}}) - \mathbf{B}_1\mathbf{U}\mathbf{H}_F(\mathbf{W}^{\mathrm{T}}\mathbf{B}_3^{\mathrm{T}} \otimes \mathbf{V}^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}})\|^2, \qquad (12)$$

which is minimized over **U**, **V**, **W** and $\mathbf{H}_F$. Replacing $\mathbf{C}_F$ by $\mathbf{B}_1^{+}\mathbf{X}_F(\mathbf{B}_3^{+\mathrm{T}} \otimes \mathbf{B}_2^{+\mathrm{T}})$, and $\mathbf{B}_j$ $(j = 1, 2, 3)$ by its decomposition, we find

$$\begin{aligned} &g(\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{H}_F) \\ &= \|\mathbf{Q}_1\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}_F(\mathbf{Q}_3\mathbf{Q}_3^{\mathrm{T}} \otimes \mathbf{Q}_2\mathbf{Q}_2^{\mathrm{T}}) - \mathbf{Q}_1\mathbf{R}_1\mathbf{U}\mathbf{H}_F(\mathbf{W}^{\mathrm{T}}\mathbf{R}_3^{\mathrm{T}}\mathbf{Q}_3^{\mathrm{T}} \otimes \mathbf{V}^{\mathrm{T}}\mathbf{R}_2^{\mathrm{T}}\mathbf{Q}_2^{\mathrm{T}})\|^2 \\ &= \|\mathbf{Q}_1^{\mathrm{T}}\mathbf{X}_F(\mathbf{Q}_3 \otimes \mathbf{Q}_2) - \mathbf{R}_1\mathbf{U}\mathbf{H}_F(\mathbf{W}^{\mathrm{T}}\mathbf{R}_3^{\mathrm{T}} \otimes \mathbf{V}^{\mathrm{T}}\mathbf{R}_2^{\mathrm{T}})\|^2, \qquad (13) \end{aligned}$$

where the second step is analogous to the third step in Eq. (6). Clearly, Eq. (13) is equivalent to Eq. (9) with $\tilde{\mathbf{E}} \equiv \mathbf{R}_1\mathbf{U}$, $\tilde{\mathbf{F}} \equiv \mathbf{R}_2\mathbf{V}$, and $\tilde{\mathbf{G}} \equiv \mathbf{R}_3\mathbf{W}$. Because $\mathbf{R}_j$ $(j = 1, 2, 3)$ is nonsingular, minimization of Eq. (13) is equivalent to minimization of Eq. (9). It follows that the PBM method, indirectly, minimizes the same loss function as the CPK approach does. Hence, the PBM method can be considered as a variant of the CPK approach, with a special choice for the algorithm to minimize Eq. (9).

The main advantage of the CPK approach over A&K's [2] procedure is, again, that it does not rely on a new algorithm. In Step 3 we can use any algorithm, for instance, the efficient algorithm proposed by Kroonenberg et al. [11]. Since this does not rely on repeated eigendecompositions, it can be expected to be considerably faster than the one modified by A&K. A more important advantage is that a user who has a program for three-mode PCA is now able to analyze very large arrays (after projecting them on useful bases) without having to resort to an alternative program. To show how one may proceed, an example of this simple procedure (in MATLAB) is described in Appendix A.

Recently, a different efficient algorithm for three-mode PCA has been proposed by Kiers et al. [12]. In fact, this method is very similar in spirit to the A&K procedure. It also uses a projection step and a modification of an existing algorithm. The main difference with the A&K procedure is that the latter is used in situations where the data array is *approximated* by a projected version, whereas in Kiers et al.'s approach, the projection is performed only if it describes the original data *perfectly*. Such a projection results in an array **C** of smaller size than **X** as soon as the product of two of the three orders is smaller than the order of the third mode, for instance, when $JK < I$. In such cases, the array **X** can be projected on the matrix $\mathbf{B}_1 = \mathbf{X}_F$ of order $I \times JK$ to obtain an array **C** of order $JK \times J \times K$. Like A&K, Kiers et al. developed a new algorithm for analyzing the projected data. As seen in the present paper, however, existing algorithms can be used as well. In fact, in the CPK framework, the present projection procedure is very simple indeed: The projection of **X** on $\mathbf{B}_1 = \mathbf{X}_F$ comes down to premultiplying the frontal planes of **X** by $\mathbf{Q}^{\mathrm{T}}$, from the (QR) decomposition $\mathbf{X}_F = (\mathbf{X}_1|\ldots|\mathbf{X}_K) = \mathbf{QR}$. Then $\tilde{\mathbf{C}}_F = \mathbf{Q}^{\mathrm{T}}\mathbf{X}_F = \mathbf{Q}^{\mathrm{T}}\mathbf{QR} = \mathbf{R}$. Hence, in this case we merely have to compute the Cholesky decomposition of $\mathbf{X}_F^{\mathrm{T}}\mathbf{X}_F$ to obtain **R** and to consider this as our matrix $\tilde{\mathbf{C}}_F$; upon restoring this as a three-way array **C**, we can apply any three-mode PCA algorithm to this to obtain the solution for **F**, **G** and **H** directly, and that of **E** indirectly as $\mathbf{E} = \mathbf{Q}\tilde{\mathbf{E}}$. Clearly, to obtain **F**, **G** and **H**, it suffices to have the matrix with inner products of the columns of $\mathbf{X}_F$, from which **R** can be computed. When I is large indeed, it is conceivable that there is no substantive interest in **E**. However, in case there is, computation of the component matrix **E** merely requires one final multiplication involving the large matrix **Q**.

Three-mode PCA is not the only method that yields components for which it is reasonable to assume that they approximately lie in the column spaces of the matrices $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ suggested by CPK or A&K. For instance, the same assumption may be reasonable for PARAFAC/CANDECOMP [3–5] and Kiers' [7] core constrained three-mode PCA. In fact, the former (PARAFAC/CANDECOMP) was the method for which the CPK approach was originally developed [3]. In both cases, the procedure can be applied completely analogously. This even holds in case where missing data are handled by missing data imputation, provided that the missing data occur in complete columns of the matrix $(\mathbf{X}_1|...|\mathbf{X}_K)$.

For PARAFAC/CANDECOMP, a short-cut procedure similar to the one by Kiers et al. [12], was proposed by Kiers and Krijnen [13] for efficiently analyzing data with $I > JK$. Their procedure is, as the procedure in [12], based on the derivation of a new algorithm. However, just as for three-mode PCA, a simpler algorithm can be obtained by following the CPK approach and using the existing PARAFAC/CANDECOMP algorithm. The ensuing method consists of applying PARAFAC/CANDECOMP to the $JK \times J \times K$ array $\tilde{\mathbf{C}}$ of which the frontal planes are taken from $\mathbf{R}$ from the (QR) decomposition of $\mathbf{X}_F$ (see above). Similar short-cuts can be devised for other methods for which it can be specified exactly in which column spaces the optimal components lie. When the components are unconstrained, or constrained to be columnwise orthonormal, the existing algorithms can be applied directly to the array $\tilde{\mathbf{C}}$.

4. Discussion

In the present paper, the PBM methods by A&K have been shown to be variants of the CPK approach. It has been seen that the latter approach is more flexible in that it obviates the need to adjust existing algorithms. This is an advantage in itself, because it allows a user to analyze large data arrays by means of existing programs without additional programming effort (except for the projection steps). The projection steps become particularly simple when one uses columnwise orthonormal basis matrices. In fact, in that case, the simple procedure sketched by A&K ([1], p. 32) as a procedure that almost never gives true scores and loadings, is equivalent to the CPK approach (and does give true scores and loadings).

The fact that the A&K algorithm can be replaced by any algorithm for the (three-way) method at hand is useful not only because it allows the user to use existing programs. It also allows the user to choose the most efficient algorithms available for the (three-way) method under consideration.

Acknowledgements

This research has been made possible by a fellowship from the Royal Netherlands Academy of Arts and Sciences to the first author. The second author was assisted by a grant from the Natural Sciences and Engineering Research Council of Canada. The authors are obliged to Jos ten Berge for helpful comments on an earlier version.

Appendix A. Example MATLAB program for using a standard three-way method for the analysis of very large data

```
% Procedure for analyzing a very large three-way array X by the method "threeway".
% Here threeway stands for a matlab function threeway.m performing a three-way analysis,
% e.g., three-mode PCA, PARAFAC.
%
% "qqrr" refers to a function that modifies the results of the matlab qr.m
% routine by deleting all zero rows in R and the corresponding columns in Q.
%
% input:   X: I×JK matrix with frontal planes of data array next to each other
%          I,J,K: order of the array
%          B1, B2, B3: externally obtained basis matrices
% output: A,B,C: component matrices for the three modes.


       Iorig=I;Jorig=J;Korig=K;                 % (original sizes)
       [Iorig,I]=size(B1);
       [Jorig,J]=size(B2);
       [Korig,K]=size(B3);
       Xorig=X;                                 % save original data
       [Q1,R1]=qqrr(B1);                        % qqrr computes a QR decompostion
       [Q2,R2]=qqrr(B2);
       [Q3,R3]=qqrr(B3);
       X=Q1'*X*kron(Q3,Q2);
       [A,B,C,...]=threeway(X,I,J,K)
       A=Q1*A;
       B=Q2*B;
       C=Q3*C;
       X=Xorig;I=Iorig;J=Jorig;K=Korig;        % restore original values
```

References

[1] B.K. Alsberg and O.M. Kvalheim, Chemom. Intell. Lab. Syst. 24 (1994) 31–42.

[2] B.K. Alsberg and O.M. Kvalheim, Chemom. Intell. Lab. Syst. 24 (1994) 43–54.

[3] J.D. Carroll, S. Pruzansky and J.B. Kruskal, Psychometrika 45 (1980) 3–24.

[4] J.D. Carroll and J.-J. Chang, Psychometrika 35 (1970) 283–319.

[5] R.A. Harshman, UCLA Working Papers in Phonetics 16 (1970) 1–84.

[6] J.D. Carroll and S. Pruzanksy, Use of LINCINDS as a rational starting configuration for INDSCAL, unpublished memorandum (Bell Telephone Laboratories, 1979).

[7] H.A.L. Kiers, Stat. Appl. 4 (1992) 659–667.

[8] G.H. Golub and C.F. Van Loan, Matrix Computations, 2nd Ed. (The Johns Hopkins University Press, Baltimore, 1989).

[9] Y. Takane and T. Shibayama, Psychometrika 56 (1991) 97–120.

[10] P.M. Kroonenberg and J. De Leeuw, Psychometrika 45 (1980) 69–97.

[11] P.M. Kroonenberg, J.M.F. Ten Berge, P. Brouwer and H.A.L. Kiers, Comput. Stat. Q. 5 (1989) 81–87.

[12] H.A.L. Kiers, P.M. Kroonenberg and J.M.F. Ten Berge, Psychometrika 57 (1992) 415–422.

[13] H.A.L. Kiers and W.P. Krijnen, Psychometrika 56 (1991) 147–152.