JOURNAL OF CHEMOMETRICS
J. Chemometrics 2005; **19**: 216–235
Published online 16 November 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.926

Mathematical improvements to maximum likelihood parallel factor analysis: theory and simulations

Lorenzo Vega-Montoto[1], Hong Gu[2] and Peter D. Wentzell[1]*

[1]Department of Chemistry, Trace Analysis Research Centre, Dalhousie University, Halifax, NS, Canada B3H 4J3
[2]Department of Mathematics, Statistics and Computing Science, Dalhousie University, Halifax, NS, Canada B3H 3J5

Received 15 February 2005; Accepted 12 July 2005

A number of simplified algorithms for carrying out *ma*ximum *l*ikelihood *para*llel *fac*tor analysis (MLPARAFAC) for three-way data affected by different error structures are described. The MLPARAFAC method was introduced to establish the theoretical basis to treat heteroscedastic and/or correlated noise affecting trilinear data. Unfortunately, the large size of the error covariance matrix employed in the general formulation of this algorithm prevents its application to solve standard three-way problems. The algorithms developed here are based on the principle of alternating least squares, but differ from the generalized MLPARAFAC algorithm in that they do not use equivalent alternatives of the objective function to estimate the loadings for the different modes. Instead, these simplified algorithms tackle the loss of symmetry of the PARAFAC model by using only one representation of the objective function to estimate the loadings of all of the modes. In addition, a compression step is introduced to allow the use of the generalized algorithm. Simulation studies carried out under a variety of measurement error conditions were used for statistical validation of the maximum likelihood properties of the algorithms and to assess the quality of the results and computation time. The simplified MLPARAFAC methods are also shown to produce more accurate results than PARAFAC under a variety of conditions. Copyright © 2005 John Wiley & Sons, Ltd.

KEYWORDS: MLPARAFAC; PARAFAC; weighted PARAFAC; maximum likelihood; three-way data; trilinear data; measurement errors; error covariance; parallel factor analysis; data compression

1. INTRODUCTION

Over the past three decades, the use of multivariate [1,2] and multiway [3–5] methods have driven a change in the analytical laboratory from a univariate and chemically selective paradigm into a multivariate/multiway and mathematically selective philosophy. Nonetheless, it was not until the 1990s that some researchers [6–11] started to consider in a consistent manner the nature of the noise corrupting these measurements in the context of multivariate analysis. The assumption of *iid*-normal (independent and identically distributed noise with a normal distribution) upon which univariate least squares methods [12] relied to provide optimal estimates was recognized as a limitation in the presence of other types of noise cases. The nature of the noise affecting multivariate measurements is strongly related to the nature of the experiment and the type of instrument employed [13], as well as different cosmetic manipulations [14,15] that make the noise deviate from the *iid* condition. Instrumental factors, such as spatial correlation in the detector sensors, detector response variation, source intensity instability, temperature fluctuations, and physical variation in the sample and in the positioning of the sample within the instrument are a few examples of the causes of the existence of correlated noise.

In 1994, Paatero [6] resurrected the idea of introducing some kind of weight information related to the uncertainty of the variables when the method positive matrix factorization (PMF) was introduced. Unfortunately, this weighting information was only related to the variance of these variables, correcting for the violation of the identical distribution of the noise, but their method still assumed that the errors were independent from channel to channel. A more complete alternative was available a few years later when Wentzell *et al.* [7] formulated maximum likelihood principal component analysis (MLPCA) which considered cases where the *iid* condition was completely violated due to the presence of heteroscedasticity and correlated noise. A principal innovation of this method was the use of the error covariance matrix (ECM), which is a more general way of describing the magnitude of the errors and the relationships among them. A few other closely related methods [8–10] have also been introduced to handle bilinear data in a maximum likelihood fashion, sometimes adding other constraints or information.

*Correspondence to: P. D. Wentzell, Department of Chemistry, Trace Analysis Research Centre, Dalhousie University, Halifax, NS, Canada B3H 4J3.
E-mail: Peter.Wentzell@Dal.Ca

Contract/grant sponsors: Natural Sciences and Engineering Research Council (NSERC) of Canada; Dow Chemical Company, Midland, MI, USA.

The application of this philosophy to multiway data lagged behind the bilinear case until recently. For trilinear data, the optimality of the least-squares solution obtained by PARAFAC was proven by Liu and Sidiropoulos [16] using simulated data that provided solutions that approached the Cramer–Rao lower bound, when the noise was *iid*. Therefore, it was perceived as useful to extend least squares approaches to cases of non-*iid* measurement noise. Recently Bro *et al.* [11] introduced a generic method called maximum likelihood via iterative least squares estimation (MILES), which worked as a iterative preprocessing tool to condition the data from a maximum likelihood perspective in order that least squares methods such as PCA and *par*allel *fac*tor (PARAFAC) analysis could optimally handle the estimation process. The method is based on a majorization strategy in which the original objective function is substituted by a simpler and equivalent objective function in each step of the estimation process. Unfortunately, the simplicity of this numerical implementation is hindered by the amount of computation time needed. Since the method runs the full least squares optimization in each step, the time needed to get an estimate is sometimes excessive. Another important drawback of this approach is that the physical problem becomes obscured by the efficient but unfamiliar numerical methodology.

More recently, a method called *m*aximum *l*ikelihood *parallel* *fac*tor analysis (MLPARAFAC) was introduced to the chemometrics literature [17]. The main difference with respect to MILES is that MLPARAFAC is a method based solely on an alternating least squares (ALS) optimization. The implementation is straightforward and runs faster since the noise information is introduced in each iteration rather than in each optimization step as it is in MILES. The method was designed to estimate the parameters of the well-known PARAFAC model from a maximum likelihood perspective in cases where different violations of the assumed *iid*-normal error condition exist. Four algorithms for carrying out MLPARAFAC based on an ALS framework were described in this work. The simplest of these was designed to work with cases where the measurement errors are non-uniform (heteroscedastic) but uncorrelated. The most general form of the algorithm can treat data with any type of error covariance structure. Two simplifications of the general algorithm were also presented which more efficiently handle more restricted error covariance structures. All of the algorithms were shown to produce maximum likelihood estimates through a comparison of the distribution of the objective function with the χ^2 distribution. It was also shown that the quality of the estimated loading vectors for MLPARAFAC is significantly better than for the PARAFAC models in cases where the error covariance matrix is known.

Although the original paper on MLPARAFAC outlined the theory for dealing with correlated error, demonstrated its validity through simulation, and introduced some exact simplifications based on mathematical properties of the matrices used in the estimation process, it was found that many important situations remained uncovered and they are the subject of this paper. This work will be divided into two parts: the first part will introduce, test, and apply the methodology to simulated data, while a companion paper will treat the application of MLPARAFAC to three experimental data sets. This paper will initially analyze the two simplifications introduced in the earlier work, since more interesting and useful simplifications can be found when those algorithmic alternatives are approached from a geometrical and computational point of view. This will lead us to the extension of one of these alternatives to more general cases where the noise structure along one order is less restricted and to cases where the error structure is correlated along two orders. Cases where the noise structure is correlated along more than two orders will need to be treated using the general algorithm, but since this is usually impractical from a computational point of view when the raw data are used, a compression approach will be introduced. After the algorithmic issues have been covered, a thorough analysis will be provided in order to go from these mathematically clear and well-defined cases to the more ''gray'' real cases. Also, some simulations will show the effects in the estimates when some cases with a considerable departure from the assumed structure are used with the corresponding simplification.

1.1. Notation

In this paper, scalars are indicated by italics and vectors by bold lower-case characters. Bold upper-case letters are used for two-way matrices and underlined bold upper-case letters for three-way data. The letters **A**, **B**, **C** and *I*, *J*, *K* are reserved for indicating the first, second, and third mode of three-way data and the dimensions of those modes, respectively. Also, the letter *P* is reserved to represent the number of factors used in the model. The terms mode, way, and order are used interchangeably, as well as the terms factors and components. When three-way arrays are unfolded to matrices, the following notation will be used. If $\underline{\mathbf{X}}$ $(I \times J \times K)$ is unfolded while retaining the first order to produce a $(I \times JK)$ matrix, this will be designated $\mathbf{X}_a$. In the same way, matrices $\mathbf{X}_b$ $(J \times IK)$ and $\mathbf{X}_c$ $(K \times IJ)$ will be used to represent unfolded matrices which retain the second and the third orders, respectively. In general, other matrices with subscripts *a*, *b*, and *c* represent unfolding while retaining the first, second, and third modes. The use of subscripts *i*, *j*, *k*, and *p* accompanying matrices and vectors refers to the use of the *i*, *j*, *k*, and *p*-th slice or row of the corresponding data array or matrix. An important exception to this notational rule is when subscripts *i*, *j*, *k*, and *p* accompany matrix **I** in which case it refers to the identity matrix of order represented by the subscript. The use of superscript '-T' accompanying square matrices indicates that the inverse of the transpose of the corresponding matrix is calculated. The symbol ''$\otimes$'' will be used primarily to indicate the Kronecker product, but will also be used to represent the tensor product in certain cases which will be clearly distinguished. The symbol ''$|\otimes|$'' will be used to indicate the Khatri–Rao product [3], which is a compact version of a column-wise Kronecker product.

2. THEORY

In the original paper introducing MLPARAFAC [17], it was noted that for many chemical applications, error covariance affects only one order, or at least the covariance in other orders can be neglected. This can, in certain cases, result in substantial simplification of the generalized algorithm. For the purpose of illustration, only the case where correlations exist along the rows will be considered, since correlations along other orders can be rendered equivalent through permutation of the original array or appropriate adjustment of equations introduced. For this case, three common situations can be distinguished: (1) the error covariance is different among all of the rows forming the array; (2) the error covariance is different among rows forming different slices but identical among the rows of the same slice; and (3) the error covariance is identical among the rows of all the slices. Simplifications for cases (2) and (3) were formulated based on mathematical identities and the more general scenario represented by case (1) was considered unsolvable by any simplification. Deeper scrutiny of these simplifications led the authors to realize the existence of more powerful and general simplifications for these cases. The next subsections will revisit these two simplifications from a geometrical and algorithmic point of view. One of these simplifications will be further extended to the case where error covariance is different among all of the rows forming the array and to the case where correlation is present along two modes.

2.1. Correlation along one order

2.1.1. Case 1A

Imagine a trilinear data set such that the error correlation can be expected to affect only one order, which we will assume to be the second order. In addition, in certain cases where this assumption applies, it may be possible to make the additional assumption that the error covariance matrix is the same for each row in all the slices of data. Given that the observed data, $\underline{\mathbf{X}}$, can be considered the sum of the true data, $\underline{\mathbf{X}}^{\mathrm{o}}$, and an array of measurement errors, $\underline{\mathbf{E}}$, this can be mathematically represented using any of the following three equations:

$$\begin{aligned}\underline{\mathbf{X}} &= \underline{\mathbf{X}}^{\mathrm{o}} + \underline{\mathbf{E}}\\ \mathbf{X}_k &= \mathbf{X}_k^{\mathrm{o}} + \mathbf{E}\\ \mathbf{x}_{i:k} &= \mathbf{x}_{i:k}^{\mathrm{o}} + \mathbf{e}\end{aligned} \tag{1}$$

The trilinear data can be equivalently represented as a three-way array of elements, a slice-by-slice representation, or a vector representation, respectively. As mentioned above, all these representations are equivalent but only the last representation allows a clear representation of the characteristics of the noise, which follows a normal distribution around zero and with variance/covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{e} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$. Since the errors are correlated, $\boldsymbol{\Sigma}$ cannot be expressed as a multiple of the identity matrix. This case is conceptually similar to the case treated by Brown *et al.* [15] in which bilinear data corrupted by drift noise were accommodated by applying an optimally designed filter. Therefore, we can consider our problem as a similar preprocessing problem in which each frontal slice k of data is multiplied by a filter as shown in Equation 2:

$$^{\mathrm{F}}\mathbf{X}_k = \mathbf{X}_k\mathbf{F} \tag{2}$$

$\mathbf{F}$ is an optimal filter matrix that will be applied to the data, and thus to the individual error vectors in each slice as shown in Equation 3:

$$^{\mathrm{F}}\mathbf{X}_k = (\mathbf{X}_k^{\mathrm{o}} + \mathbf{E})\mathbf{F} = \mathbf{X}_k^{\mathrm{o}}\mathbf{F} + \mathbf{E}\mathbf{F} \tag{3}$$

The error covariance matrix after filtering can be expressed as:

$$^{\mathrm{F}}\boldsymbol{\Sigma} = E(\mathbf{F}^{\mathrm{T}}\mathbf{e}^{\mathrm{T}}\mathbf{e}\mathbf{F}) \tag{4}$$

Since the filter matrices are constant, they can be extracted from the expectation operator $E(\bullet)$ to obtain:

$$^{\mathrm{F}}\boldsymbol{\Sigma} = \mathbf{F}^{\mathrm{T}}E(\mathbf{e}^{\mathrm{T}}\mathbf{e})\mathbf{F} = \mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{F} \tag{5}$$

$\mathbf{F}$ is an optimally constructed filter in the sense that it will rotate and scale the data yielding a new noise data, $\mathbf{EF}$, which follows a normal distribution around zero with variance/covariance matrix equal to a multiple of the identity matrix, $^{\mathrm{F}}\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. Therefore, dropping the proportionality constant (which can be viewed simply as a scaling factor) and substituting this equality into Equation 5 yields:

$$\mathbf{I} = \mathbf{F}^{\mathrm{T}}\boldsymbol{\Sigma}\mathbf{F} \tag{6}$$

The filter matrix $\mathbf{F}$, which solves Equation 6, can be readily obtained considering the estimation process as an extended eigenproblem in which matrix $\boldsymbol{\Sigma}$ is initially rotated to yield a diagonal matrix that then goes through a scaling process producing the identity matrix. This linear transformation can only be executed when $\mathbf{F}$ is defined as the product of the eigenvectors of $\mathbf{F}$, $\mathbf{U}$, multiplied by the inverse of the diagonal matrix $\mathbf{S}$ formed by the square root of corresponding eigenvalues of $\mathbf{F}$ as shown in Equation 7.

$$\begin{aligned}\mathbf{F} &= \mathbf{U}^{*}\mathbf{S}^{-1}\\ \boldsymbol{\Sigma} &= \mathbf{U}\mathbf{S}^{2}\mathbf{U}^{\mathrm{T}} = \mathbf{U}\mathbf{S}\mathbf{S}\mathbf{U}^{\mathrm{T}}\end{aligned} \tag{7}$$

It is worth noting that, even though the term filter has been used thus far, this optimal filter will not have the typical form of a least squares polynomial filter such as the usual symmetric/antisymmetric band diagonal Savitzky–Golay filters [18]. In fact, it will not technically be a filter since no noise reduction is carried out. It can better be understood as a ''modulator'' which transforms the original signal corrupted by non-*iid* noise to a signal corrupted by *iid*-noise. This transformation affects not only the noise but also the imbedded true signal that is the aim of the estimation process. Fortunately, this transformation will not affect the trilinear structure, since all the slices are going to be rotated and scaled equivalently, as is evident from Equation 3. Additionally, uniqueness, which is one of the most appealing characteristics of trilinear data, will be preserved since the inverse transformation exists and can be easily applied to the estimated loadings describing the order along which the noise is correlated. This is mathematically represented by Equation 8:

$$\begin{aligned}^{\mathrm{F}}\hat{\mathbf{X}}_a &= \hat{\mathbf{A}}\mathbf{I}_a(\hat{\mathbf{C}}^{\mathrm{T}} \otimes {}^{\mathrm{F}}\hat{\mathbf{B}}^{\mathrm{T}})\\ \hat{\mathbf{B}} &= \mathbf{F}^{-1}({}^{\mathrm{F}}\hat{\mathbf{B}})\end{aligned} \tag{8}$$

The advantages of this approach with respect to the previous approach formulated in Table IV of Reference [17] to treat this type of data optimally are twofold. First, it will not be necessary to calculate the inverse of $\boldsymbol{\Psi}_a(\boldsymbol{\Psi}_a = \mathbf{I}_K \otimes \boldsymbol{\Sigma})$ in order to estimate the parameters of the model since the error structure information is reflected in the data and not in the projection of the data. This is rather convenient, since $\boldsymbol{\Sigma}$ can be rank deficient for a variety of reasons. Second, the estimation procedure will be carried out using the standard PARAFAC algorithm, which is more stable and less computationally involved than the algorithm in Table IV of Reference [17].

2.1.2. Case 1B

In addition to the simplest case treated above, Figure 1 represents a few other cases where the complexity of the error structure increases gradually up to the most complex case where the errors affecting all the elements of the multi-way data are related. Case B represented in Figure 1 takes the simplest error structure one step further to the case where noise is still correlated along one dimension but the structure and/or magnitude of it changes from slice to slice. The first reasonable approach to treat such a case might be to use the previously described strategy, utilizing in each case a filter matrix derived from the error covariance matrix obtained for each individual slice as shown in Equation 9:

$$^{\mathrm{F}}\mathbf{X}_k = \mathbf{X}_k\mathbf{F}_k \qquad (9)$$

Equation 10 shows that the reasoning holds from a noise treatment perspective, since the local filtering will produce a diagonal matrix because the filter matrices are going to rotate and scale the original error covariance matrix for each slice in order to fulfill the *iid* condition.

$$^{\mathrm{F}}\boldsymbol{\Sigma}_k = \mathbf{F}_k^{\mathrm{T}}E(\mathbf{e}_k^{\mathrm{T}}\mathbf{e}_k)\mathbf{F}_k = \mathbf{F}_k^{\mathrm{T}}\boldsymbol{\Sigma}_k\mathbf{F}_k \qquad (10)$$

However, when this strategy is thoroughly explored via Equation (11), it is clear that the ''cleaning effect'' produced over the noise has a negative collateral effect over the part of the data related to the chemical information since the trilinearity is destroyed due to a different rotation of the data in each slice.

$$^{\mathrm{F}}\mathbf{X}_k = (\mathbf{X}_k^{\mathrm{o}} + \mathbf{E}_k)\mathbf{F}_k = \mathbf{X}_k^{\mathrm{o}}\mathbf{F}_k + \mathbf{E}_k\mathbf{F}_k \qquad (11)$$

A solution based on a mathematical simplification of the full error covariance matrix was introduced in Reference [17]. The approach used to obtain this simplification was based on the idea of finding a simpler representation of the error covariance matrix to express the normal equations to estimate the loading for each mode. A relatively concise and computationally efficient formulation was found for the estimation of the loading for the modes A and C, but the

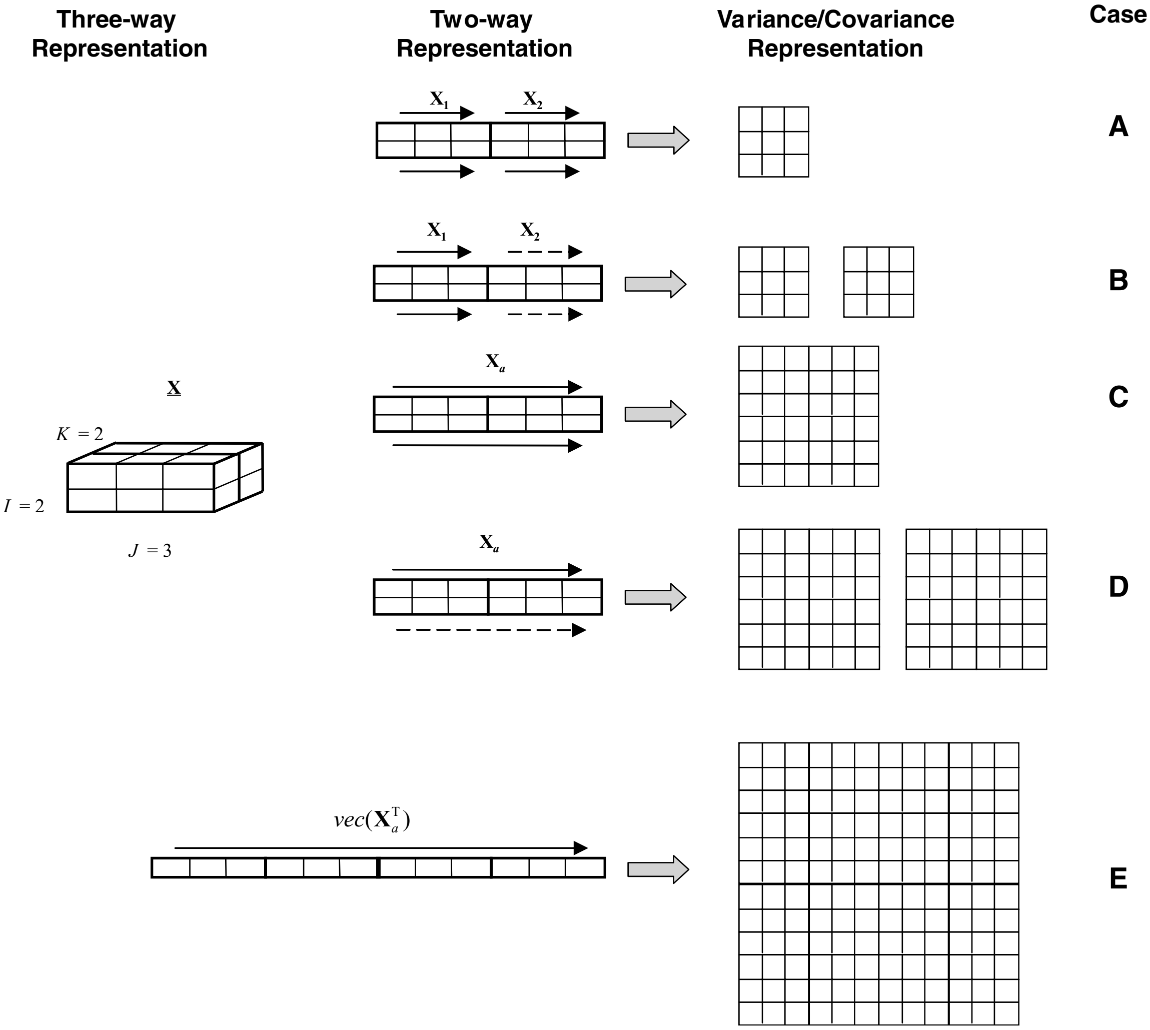

Figure 1. Illustration of the possible scenarios in which correlated errors might pervade a three-way array and the corresponding representations of the structure of the error covariance matrix to describe all the sources of variation. Arrows indicate which elements of the unfolded or vectorized three-way array have correlated errors. Different arrows represent different error structures.

equation for the estimation of B was still a function of the full error covariance matrix for this particular mode, as can be seen in Equation (12):

$$\operatorname{vec}\left(\hat{\mathbf{B}}^{\mathrm{T}}\right) = \left(\mathbf{V}_b^{\mathrm{T}}\boldsymbol{\Omega}_b^{-1}\mathbf{V}_b\right)^{-1}\mathbf{V}_b^{\mathrm{T}}\boldsymbol{\Omega}_b^{-1}\operatorname{vec}\left(\mathbf{X}_b^{\mathrm{T}}\right) \tag{12}$$

Equivalently to the notation in reference 17, $\mathbf{V}_b$ is a $JP \times IJK$ matrix with $\mathbf{Z}_b^{\mathrm{T}} = (\mathbf{C} \otimes \mathbf{A})^{\mathrm{T}}$ repeating along the diagonal. The matrix $\boldsymbol{\Omega}_b$ is the full error covariance matrix for $\operatorname{vec}(\mathbf{X}_b^{\mathrm{T}})$, providing information about the error covariance among all the measurements. The presence of $\boldsymbol{\Omega}_b$ in this equation makes this simplification practically useless since its dimensions in a practical application will make the storage and manipulation for this equation prohibitive.

The lack of success of this approach can be attributed to the well-established strategy in standard PARAFAC in which the different estimation sub-steps are formulated using the same objective function expressed differently for each mode. This strategy is used because, due to the symmetry of the PARAFAC model, the implementation is not only efficient but also extremely simple, making the normal equations very similar from one mode to the other. However, when the characteristics of the noise are taken into account, this symmetry is lost, making it necessary to express the problem as the general problem, since the existence of a simplified version of the error covariance matrix in the given space is not possible or extremely difficult to find. Therefore, in this paper, a new approach is introduced in which the data are initially arranged in order to have the major source of correlated noise along the mode B, followed by the second major source of correlation along mode C leaving mode A as the mode not affected by correlated noise. After the data are arranged, the estimation equations are obtained by expressing all the sub-steps as minimization problems of the objective function written to preserve mode A alone. It is worth noting in advance that this alternative is laborious, since the equations will no longer be simple bilinear representations in which the mode to be determined is represented independently and the other two modes are represented as a composite mode, but as a more complex set of equations in which the modes are going to be interrelated most of the time.

We will start by showing the estimation of the normal equation for case B represented in Figure 1. For this particular case, we will be able to see how the equation obtained for mode A is exactly the same as the equation shown in Reference [17] as proof that this strategy is equivalent to the standard strategy used in the past. Also, our goal will be accomplished by formulating a tractable equation for mode B, making Equation (12) unnecessary. Even though the estimation of the loading for mode C was not particularly complex, the new strategy will provide a set of equations that is less demanding from a storage point of view. We start by defining the objective function as Equation (13):

$$f = \sum_{k=1}^{K} \operatorname{trace}\left[(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}\right] \tag{13}$$

In this equation, $\mathbf{X}_k$ represents the kth slice of the three-way array $\underline{\mathbf{X}}$, $\mathbf{A}$ and $\mathbf{B}$ are matrices of dimensions $I \times P$ and $J \times P$ representing the loading vectors for mode A and B respectively, $\mathbf{D}_k$ is a $P \times P$ diagonal matrix with the kth row of the $K \times P$ matrix $\mathbf{C}$ along the diagonal and $\boldsymbol{\Psi}_k^{-1}$ is the inverse of the error covariance matrix that describes the noise affecting all the rows of the kth slice of the three-way array $\underline{\mathbf{X}}$. The implementation of an alternating least squares algorithm for the estimation of mode A loadings assumes B and C are known and then Equation (13) is minimized with respect to each element forming $\mathbf{A}$. Before proceeding with the derivation, it will be convenient to express Equation (13) as the quadratic form shown in Equation (14) where $\mathbf{M}_k = \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}$:

$$f = \sum_{k=1}^{K} \operatorname{trace}(\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\mathbf{M}_k^{\mathrm{T}}) \tag{14}$$

Equation (15) shows the derivation:

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{A}_{ip}} &= \frac{\partial f}{\partial \mathbf{M}_k}\left(\frac{\partial \mathbf{M}_k}{\partial \mathbf{A}_{ip}}\right)^{\mathrm{T}} = \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial \mathbf{M}_k^{\mathrm{T}}}{\partial \mathbf{A}_{ip}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial\left(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\right)^{\mathrm{T}}}{\partial \mathbf{A}_{ip}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\left(-\mathbf{E}_{ip}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\left(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\right)\boldsymbol{\Psi}_k^{-1}\left(-\mathbf{E}_{ip}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(-2\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}} + 2\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right) \\
&= -2\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right) \\
&\quad + 2\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right)
\end{aligned} \tag{15}$$

Equation (15) represents the first derivative of the objective function with respect to the elements of $\mathbf{A}$. The matrix $\mathbf{E}_{ip}$ is an elementary $I \times P$ matrix with all of its elements equal to zero with the exception to the element located in the position $i \times p$, which is equal to 1. This equation will be equal to zero for the optimum value of $\mathbf{A}_{ip}$ given $\mathbf{B}$ and $\mathbf{C}$. In order to calculate this value of $\mathbf{A}_{ip}$ Equation (15) is transformed as follows:

$$\begin{aligned}
&\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right) = \sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right) \\
&\sum_{k=1}^{K} \operatorname{vec}\left(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}}\right)^{\mathrm{T}}\operatorname{vec}\left(\mathbf{E}_{pi}\right) \\
&= \sum_{k=1}^{K} \operatorname{vec}\left(\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{T}}\operatorname{vec}\left(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\mathbf{E}_{ip}^{\mathrm{T}}\right) \\
&\sum_{k=1}^{K} \operatorname{vec}\left(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}}\right)^{\mathrm{T}}\operatorname{vec}\left(\mathbf{E}_{pi}\right) \\
&= \sum_{k=1}^{K} \operatorname{vec}\left(\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbf{I}_I \otimes \mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\right)\operatorname{vec}\left(\mathbf{E}_{pi}\right) \\
&\operatorname{vec}\left(\sum_{k=1}^{K}\left(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}}\right)\right)^{\mathrm{T}}\operatorname{vec}\left(\mathbf{E}_{pi}\right) \\
&= \operatorname{vec}\left(\mathbf{A}^{\mathrm{T}}\right)^{\mathrm{T}}\sum_{k=1}^{K}\left(\mathbf{I}_I \otimes \mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\right)\operatorname{vec}\left(\mathbf{E}_{pi}\right)
\end{aligned} \tag{16}$$

Equation (16) is one of the *IP* equations necessary to estimate the loadings of **A**. The rest of the equations are obtained as the right and left parts of this equation are multiplied by the different vectorized $\mathbf{E}_{pi}$ matrices. Since this term is completely independent in both sides of the equation, the process can be carried out in a straightforward manner using a matrix **E** formed as [vec($\mathbf{E}_{11}$) vec($\mathbf{E}_{21}$) ... vec($\mathbf{E}_{IP}$)]. A closer examination of this matrix reveals that **E** is the identity matrix of order *IP*, making the multiplication theoretically sound but numerically unnecessary and providing Equation (17) to estimate the loadings of mode A:

$$\mathrm{vec}(\mathbf{A}^{\mathrm{T}}) = \left(\sum_{k=1}^{K}(\mathbf{I}_I \otimes \mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k)\right)^{-1} \mathrm{vec}\left(\sum_{k=1}^{K}(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}})\right) \tag{17}$$

Taking into consideration the properties of the vec operator and the Kronecker product, Equation (17) can be transformed to:

$$\mathbf{A} = \sum_{k=1}^{K}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\left(\sum_{k=1}^{K}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{D}_k\right)^{-1} \tag{18}$$

For this scenario, Equation (18) is a more compact and computationally efficient representation of the equivalent Equation (30) in Reference [17] and reproduced here as Equation (19).

$$\mathbf{A} = \mathbf{X}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}(\mathbf{Z}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}})^{-1} \tag{19}$$

The summations over *k* found in Equation (18) can be eliminated by using the unfolded representation of $\underline{\mathbf{X}}$ retaining mode A ($\mathbf{X}_a$) and by expressing $\boldsymbol{\Psi}_a$ as the block diagonal error covariance matrix with the individual error covariance matrices for each slice along the diagonal and expressing the projection space by $\mathbf{Z}_a = \mathbf{I}_a(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}}$.

For convenience, the mathematical procedure to derive the estimation equations for the rest of the loadings in this and the rest of the different scenarios are provided in the Appendix. The results of these derivations are given below.

$$\mathrm{vec}(\mathbf{B}) = \left(\sum_{k=1}^{K}(\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{D}_k \otimes \boldsymbol{\Psi}_k^{-1})\right)^{-1} \mathrm{vec}\left(\sum_{k=1}^{K}(\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1})\right) \tag{20}$$

$$\mathbf{c}_k = \left[\left(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B} \otimes \mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathrm{vec}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\right)\right]^{\mathrm{T}}\mathbf{E} \tag{21}$$

It is interesting to note how both equations are composed of the two key parts of a standard weighted least squares estimator: a projection matrix spanning the space where the best approximation of the noiseless signal is located and a vector representing a weighted projection of the data onto the space where the signal is located. The awkward form of these two components is a consequence of the manner in which these equations were obtained, as anticipated at the beginning of this section. Two important details have to be mentioned for the expression to obtain loadings for mode C: vector $\mathbf{c}_k^{\mathrm{T}}$ represents the *k*th row of the $K \times P$ matrix **C** and the matrix **E** in this case is a matrix formed as [vec($\mathbf{E}_{11}$) vec($\mathbf{E}_{22}$) ... vec($\mathbf{E}_{PP}$)] and is used to choose the necessary elements for the estimation of $\mathbf{c}_k$, since the optimization was originally designed to have this vector along the main diagonal of $\mathbf{D}_k$.

2.2. Correlation along two orders

2.2.1. *Case 1C*

Figure 1C represents cases where the error structure becomes more complex by affecting elements of the data set located in two different modes. Such types of scenarios are not unusual, for example in kinetic studies where the course of the reaction is followed spectroscopically giving rise to errors that are correlated in both the time and wavelength modes, while the other mode may be composed of samples with different compositions of the reactants that are run independently of one another. For this case, we will consider that the three-way data, $\underline{\mathbf{X}}$, will be unfolded preserving the samples of different compositions in mode A, while modes B and C will be combined in one composite mode formed by the spectral information and the time information for each sample. The objective function in this case can be expressed as shown in Equation (22).

$$\begin{aligned} f &= \mathrm{trace}[(\mathbf{X}_a - \mathbf{A}\mathbf{I}_a(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a - \mathbf{A}\mathbf{I}_a(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}})^{\mathrm{T}}] \\ &= \mathrm{trace}[(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}})^{\mathrm{T}}] \end{aligned} \tag{22}$$

As mentioned in the notation section, the variables with the ''*a*'' subscript such as $\mathbf{X}_a$ and $\mathbf{I}_a$ represent the three-way arrays $\underline{\mathbf{X}}$ and $\underline{\mathbf{I}}$ unfolded preserving mode A independently. Array $\underline{\mathbf{I}}$ is $P \times P \times P$ with all the elements equal to zero but those on the superdiagonal, which are equal to unity. A small modification was made in the second expression in Equation (22) to make it more compact by expressing $\tilde{\mathbf{A}}$ as the product of **A** by $\mathbf{I}_a$. Equation (22) will be used only to obtain the loadings for modes B and C, since the loadings for mode A can be obtained by Equation (19). It is important to anticipate that the expressions obtained are not going to have the visual clarity to be interpreted as Equation (19) due to the manner in which they were obtained. The expression for the estimation of the loadings **B** for this noise characteristic is shown in Equation (23):

$$\begin{aligned} \mathrm{vec}(\mathbf{B}^{\mathrm{T}}) = &\left(\sum_{m=1}^{K}\sum_{n=1}^{K}(\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes \mathbf{L}_{mn})\right)^{-1} \\ &\times \left(\sum_{m=1}^{K}\sum_{n=1}^{K}(\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes \mathbf{R}_{mn})\right)\mathrm{vec}(\mathbf{I}_J) \end{aligned} \tag{23}$$

Equations (24) and (25) show the expressions to calculate matrices **R** and **L**, respectively.

$$\mathbf{L} = (\mathbf{C} \otimes \mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C} \otimes \mathbf{I}_p)^{\mathrm{T}} \tag{24}$$

$$\mathbf{R} = (\mathbf{C} \otimes \mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a \tag{25}$$

It should be noted that in order to obtain Equation (23), a number of manipulations of the different matrices involved in the estimation process are performed as shown in the Appendix. The most remarkable manipulation the reader must be aware of in order to understand Equation (23) is the partitioning of the $JK \times JK$ inverse error covariance matrix $\boldsymbol{\Psi}^{-1}$, the $KP \times KP$ matrix **L**, and the $KP \times KJ$ matrix **R** into three $K \times K$ super-matrices composed of the corresponding $J \times J$, $P \times P$, and $P \times J$ matrices. A graphical representation is presented in Equation (A19) in the Appendix. It is clear from the equation that subscripts *m* and *n* indicate the use of different partitioned pieces of $\boldsymbol{\Psi}^{-1}$, **L**, and **R**. Although Equation (23) does not resemble the traditional representation of a weighted least squares estimator, a closer look will actually indicate, as before, that it is formed by the key pieces of this type of estimator: a projection matrix spanning the space where the vector to be estimated resides (the term within the inverse operator) and a weighted image of the signal in the same space (the term following the inverse operator). The equation to estimate the loadings for mode C are equivalently obtained and will have a similar structure as can be seen in Equation (26):

$$\mathrm{vec}(\mathbf{C}^{\mathrm{T}}) = \left(\sum_{m=1}^{P}\sum_{n=1}^{J}\left(\mathbf{T}_{mn}\otimes\mathbf{S}_{nm}^{\mathrm{T}}\right)\right)^{-1} \times \left(\sum_{m=1}^{J}\sum_{n=1}^{J}\left(\mathbf{L}_{mn}^{\mathrm{T}}\otimes\mathbf{R}_{nm}\right)\right)\mathrm{vec}(\mathbf{I}_K) \tag{26}$$

Equations (27)–(30) show the necessary expressions to calculate the matrices involved in Equation (26). As before, matrices $\mathbf{K}_{JK}$ and $\mathbf{K}_{KJ}$ are $JK \times KK$ commutation matrices.

$$\mathbf{T} = \left(\mathbf{B}^{\mathrm{T}}\otimes\mathbf{I}_K\right)\mathbf{K}_{JK}\mathbf{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{27}$$

$$\mathbf{S} = (\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}} \tag{28}$$

$$\mathbf{L} = \mathbf{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{29}$$

$$\mathbf{R} = (\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a \tag{30}$$

Again, the subscripts m and n indicate the use of different partitioned pieces of the full matrices previously shown. In all cases, we have tried to produce the most compact representation for the expression used to calculate the estimates, but it is possible that further simplifications have been unnoticed by the authors. Also, some of these expressions will be computationally implemented in a more efficient way than the one used here, which was preferred for its notational simplicity.

2.2.2. *Case 1D*

Figure 1D represents chemical scenarios that are very similar to the previous case. The complexity of the system is taken a step further by considering that the noise propagates in a correlated fashion along two modes but the structure of this correlated noise changes from sample to sample independently. This type of situation is not uncommon when spectroscopic techniques such as NIR spectroscopy are used due to path length variations. Mathematically, the trilinear errors-in-variable model best suited to describe these data can be obtained by minimizing Equation (31):

$$f = \sum_{i=1}^{I}\left({}^{i}\mathbf{x}_a - {}^{i}\tilde{\mathbf{a}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}}\right){}^{i}\mathbf{\Psi}_a^{-1}\left({}^{i}\mathbf{x}_a - {}^{i}\tilde{\mathbf{a}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}}\right)^{\mathrm{T}} \tag{31}$$

This objective function yields expressions for the estimates that are very similar to the previous case, but in this particular case the estimates are obtained in a row by row fashion for mode A, and as a summation over the I objects in mode A for modes B and C, as can be seen in Equations (32)–(34):

$${}^{i}\mathbf{a} = {}^{i}\mathbf{x}_a{}^{i}\mathbf{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}\left(\mathbf{Z}_a{}^{i}\mathbf{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}\right)^{-1} \tag{32}$$

$$\mathrm{vec}(\mathbf{B}^{\mathrm{T}}) = \left(\sum_{i=1}^{I}\sum_{m=1}^{K}\sum_{n=1}^{K}\left({}^{i}\mathbf{\Psi}_{nm}^{-\mathrm{T}}\otimes{}^{i}\mathbf{L}_{mn}\right)\right)^{-1} \times \left(\sum_{i=1}^{I}\sum_{m=1}^{K}\sum_{n=1}^{K}\left({}^{i}\mathbf{\Psi}_{nm}^{-\mathrm{T}}\otimes{}^{i}\mathbf{R}_{mn}\right)\right)\mathrm{vec}(\mathbf{I}_J) \tag{33}$$

$$\mathrm{vec}(\mathbf{C}^{\mathrm{T}}) = \left(\sum_{i=1}^{I}\sum_{m=1}^{P}\sum_{n=1}^{J}\left({}^{i}\mathbf{T}_{mn}\otimes{}^{i}\mathbf{S}_{nm}^{\mathrm{T}}\right)\right)^{-1} \times \left(\sum_{i=1}^{I}\sum_{m=1}^{J}\sum_{n=1}^{J}\left({}^{i}\mathbf{L}_{mn}^{\mathrm{T}}\otimes{}^{i}\mathbf{R}_{nm}\right)\right)\mathrm{vec}(\mathbf{I}_K) \tag{34}$$

The estimation of the loadings for mode B and C will use the same equations shown before, but in all cases the matrix $\tilde{\mathbf{A}}$ will be replaced by the corresponding row vector ${}^{i}\tilde{\mathbf{a}}$ and the $I \times JK$ matrix $\mathbf{X}_a$ will be replaced by the row vector ${}^{i}\mathbf{x}_a$. It is important to emphasize that a set of I error covariance matrices of dimensions $JK \times JK$ will be used by this method making this alternative very expensive from a storage and computational point of view.

Thus far in Section 2, a number of different simplified scenarios have been examined, ranging from the simplest, where the error covariance matrix can be fully represented by a $J \times J$ matrix, to the most complex case, where it is necessary to consider I different $JK \times JK$ error covariance matrices. From the estimation equations, it is evident that the computational effort and the storage space increase as the complexity of the error structure characterizing the noise affecting the data grows. Therefore, the main advantage of using a simpler alternative will be the reduction of time needed to estimate the loadings for each mode. On the other hand, some scenarios will show the merit of using the more complex alternatives in order to provide the maximum likelihood estimation for each mode. The situation in which practioners will have to compromise to estimate the best possible errors-in-variables model using the minimum amount of time will depend on the characteristics of the data at hand and will be difficult to assess on an *a priori* basis. In the experimental section of this paper, a number of simulated data sets are used to validate the statistical properties of these algorithms and also to show the advantages of using one algorithm over the other in terms of time, computational power, and quality of the results.

2.3. Correlation along three orders

In the previous sections, the expressions for a number of simplified algorithms were derived for a variety of scenarios characterized by error covariance matrices of different complexity. However, there are going to be cases where none of these simplifications will provide the best solution, making it necessary to use the full algorithm presented in Reference [17]. As noted in that work, the full algorithm is not a viable alternative except when the dimensions of each order are unrealistically small. This is also the case with some of the simplifications discussed here (e.g., case 1D) for which the amount of storage space is prohibitive from a practical point of view. In these cases, some compression methods, taking advantage of different intrinsic levels of structure present within the data, will be introduced to tackle the situation. This section provides the theoretical basis of this approach and describes the implementation in the context of the model using the full error covariance matrix, although it is important to note that this can also be applied to some of the simplified models previously discussed.

2.3.1. Compression

Compression is a natural concept for two-way and multiway data since both types of data can model deterministic relationships among variables, especially in cases where a high degree of collinearity and multilinearity exist. These types of data can be represented by a smaller number of variables. Using this smaller set of variables, the data can be described within experimental error as a P-dimensional hyperplane. In this case, P is called the chemical rank or pseudorank of the data set in order to distinguish it from the mathematical rank. In general, the chemical rank is typically related to the number of underlying chemical factors or chemical components present in the mixture. For multiway data, the theoretical basis of the idea was initially introduced by Carroll *et al.* [20] in 1980, stating the optimality theorem of the Canonical Decomposition with Linear Constraints (CANDELINC) model, which ensures that the compressed array preserves the original variation maximally when a set of orthogonal bases, usually Tucker3 factors, are used to project the original array onto the space spanned by them. In 1981, Appellof and Davidson [21] provided the first application of trilinear decomposition to chemistry using both simulated and real LC/emission/excitation measurements by compressing the original data. They used the scores provided by the principal component decomposition of the unfolded data in each mode as compression bases. Later, Alsberg and Kvalheim published a number of papers [22,23] proposing a method called postponed basis matrix multiplication (PBM) using B-spline basis sets for the compression of high-dimensional arrays. A comparative study done by Kiers and Harshman [24] proved that PBM is equivalent to the more general approach based on the CANDELINC model. They also stressed that there is no need for special algorithms in the CANDELINC approach, showing it was only necessary to compress the array using a selected set of optimal bases, to use any existing multiway algorithm on the compressed array, and to decompress the result by post-multiplying the solution with the bases. The latest additions to the arsenal of compression basis sets have been a variety of wavelet families of basis sets used not only as a compression method but also as smoothing and denoising alternatives [25]. It is worth noting that these positive side effects commonly attributed to the compression using wavelets are not completely an intrinsic characteristic of the basis set, but a consequence of the projection step involved in the compression procedure.

From a structural point of view, the possibility of using different basis sets such as Tucker3 factors, PCA factors, B-splines, and wavelets is a consequence of the different levels of underlying structure present in the chemical part of any multiway data. The type of data encountered in chemistry is normally collinear (well suited for B-splines and wavelets), bilinear (ideally treated by PCA) and, in many cases, trilinear (where Tucker3 basis sets are the perfect option). This idea will be clearly demonstrated from a mathematical point of view throughout the theoretical development of an example shown next.

An $I \times J \times K$ array $\underline{\mathbf{X}}$ is given, such that matrices $\mathbf{U}$ $(I \times D)$, $\mathbf{V}$ $(J \times E)$, and $\mathbf{Z}$ $(K \times F)$, representing orthogonal basis for the systematic variation in the first, second, and third mode respectively, are considered known. Dimensions D, E, and F are the pseudo-rank (i.e., the rank of the subspace spanning the systematic variation when noise is not present [26]) for each mode. It is important to clarify that matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$ as well as ranks D, E, and F must be estimated beforehand, but in this case, for the sake of illustration, will be considered known. The standard estimation of the PARAFAC model can be expressed via Equation (35):

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \left\| \mathbf{X}_a - \mathbf{A}\mathbf{I}_a(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \tag{35}$$

The CANDELINC optimality theorem expresses the existence of three matrices $\mathbf{\Delta}$, $\mathbf{\Theta}$, and $\mathbf{\Phi}$ of orders $(D \times P)$, $(E \times P)$, and $(F \times P)$ that are related to $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ through a bilinear relationship with $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$ as shown in Equation (36):

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\mathbf{\Delta} \\ \mathbf{B} &= \mathbf{V}\mathbf{\Theta} \\ \mathbf{C} &= \mathbf{Z}\mathbf{\Phi} \end{aligned} \tag{36}$$

From a geometric point of view, this is equivalent to saying that each mode is linearly constrained to sub-spaces $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$. Therefore, if the minimization problem represented by Equation (35) is to be solved subject to the constraints expressed by Equation (36), it is only necessary to estimate the much smaller matrices $\mathbf{\Delta}$, $\mathbf{\Theta}$, and $\mathbf{\Phi}$ using the smaller array $\underline{\mathbf{Y}}$ of order $D \times E \times F$ obtained after the projection. Mathematically, this is carried out by projecting $\underline{\mathbf{X}}$ onto the space spanned by $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$ as shown in Equation (37):

$$\hat{\mathbf{X}}_a = \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{X}_a(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} \otimes \mathbf{V}\mathbf{V}^{\mathrm{T}}) \tag{37}$$

Using Equation (37), array $\underline{\mathbf{Y}}$ can be defined as:

$$\mathbf{Y}_a = \mathbf{U}^{\mathrm{T}}\mathbf{X}_a(\mathbf{Z} \otimes \mathbf{V}) \tag{38}$$

Equation (38) coincides with the expression used to calculate the core matrix for the Tucker3 model [27] when matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Z}$ represent the respective modes for this model. This is a clear mathematical proof to demonstrate the earlier statement indicating Tucker3 loadings as the perfect basis set for compression of multiway data. As mentioned before, array $\mathbf{Y}$ can be used to estimate $\mathbf{\Delta}$, $\mathbf{\Theta}$, and $\mathbf{\Phi}$ and, using the expressions depicted in Equation (36), the loadings in the original space can be calculated as the standard estimation problem depicted in Equation (35), which is reduced to the one represented in Equation (39):

$$\min_{\mathbf{\Delta},\mathbf{\Theta},\mathbf{\Phi}} \left\| \mathbf{Y}_a - \mathbf{\Delta}\mathbf{I}_a(\mathbf{\Phi} \otimes \mathbf{\Theta})^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \tag{39}$$

Thus far, it has been demonstrated why Tucker3 provides the best basis set for compression. In addition to the method of choice for the compression basis set, another key piece of information is the dimensions for corresponding basis set. In general, compression will provide an approximate solution, although it has been reported in the Reference [28] that, in situations where only one mode is high dimensional, an exact compression can be obtained by compressing this mode with a basis set of dimension equal to the product of dimensions of the other smaller two orders. In reality, exact compression can be considered the exception instead of the rule. No formal theory exists to choose the number of components for each Tucker3 loading. A rule of thumb is to use at least five more components than the number of components expected for the system, since the main objective is to speed up the algorithm and therefore only information related to the chemical structure is needed.

Up to this point, the theory and most important equations for the compression and estimation of multiway data to be treated with the standard PARAFAC and other multiway models such as PARAFAC2 and PARATUCK2 have been introduced. However, when this philosophy is to be extended for cases where a maximum likelihood method such as MLPAPAFAC is to be used, a few other equations must be introduced. These new equations will lead us to issues related to the selection, calculation, and number of basis sets needed for this approach.

Even though compression can be applied to any of the simplified scenarios, we will treat here the case where the full error covariance matrix must be used. The expression used to compress the full error covariance matrix is a direct extension of the projection expression shown in Equation (40) in a vectorized form:

$$\Xi_a = (\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})^{\mathrm{T}}\boldsymbol{\Omega}_a(\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V}) \tag{40}$$

Equation (40) will convert the original $IJK \times IJK$ full error covariance matrix describing the noise structure present in the original array $\underline{\mathbf{X}}$ in a compressed $DEF \times DEF$ full error covariance matrix describing the noise in the compressed array $\underline{\mathbf{Y}}$. Although Equation (40) represents the theoretical expression to compress the error covariance matrix, it does not solve the size problem associated with it. In order to solve this problem, the compression step must be carried out on the original data and the compressed arrays used to calculate the compressed error covariance matrix. These alternatives are equivalent, as can be seen in Equation (41), where Equation (40) is used as starting point in a backward transformation.

$$\begin{aligned}\Xi_a &= (\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})^{\mathrm{T}}\boldsymbol{\Omega}_a(\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})\\ &= (\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})^{\mathrm{T}}E\Big(\mathrm{vec}(\mathbf{E}_a^{\mathrm{T}})\mathrm{vec}(\mathbf{E}_a^{\mathrm{T}})^{\mathrm{T}}\Big)(\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})\\ &= E\Big((\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})^{\mathrm{T}}\mathrm{vec}(\mathbf{E}_a^{\mathrm{T}})\mathrm{vec}(\mathbf{E}_a^{\mathrm{T}})^{\mathrm{T}}(\mathbf{U}\otimes\mathbf{Z}\otimes\mathbf{V})\Big)\\ &= E\left(\mathrm{vec}\Big\{\big[\mathbf{U}^{\mathrm{T}}\mathbf{E}_a(\mathbf{Z}\otimes\mathbf{V})\big]^{\mathrm{T}}\Big\}\mathrm{vec}\Big\{\big[\mathbf{U}^{\mathrm{T}}\mathbf{E}_a(\mathbf{Z}\otimes\mathbf{V}\big]^{\mathrm{T}}\Big\}^{\mathrm{T}}\right)\\ &= E\Big(\mathrm{vec}(\mathbf{N}_a^{\mathrm{T}})\mathrm{vec}(\mathbf{N}_a^{\mathrm{T}})^{\mathrm{T}}\Big)\end{aligned} \tag{41}$$

It is important to differentiate in Equation (41), the expression $E(\bullet)$, which represents the expectation value of the expression in parenthesis from expression $\mathbf{E}_a$, which represents the noise array $\underline{\mathbf{E}}$ unfolded as an $I \times JK$ matrix. Expression (41) represents the symmetric outer product of the multiplication of the unfolded error array and the compression basis set in vector form. This can be transformed to the following matrix expression to be further explored:

$$\begin{aligned}\mathbf{N}_a &= \mathbf{U}^{\mathrm{T}}\mathbf{E}_a(\mathbf{Z}\otimes\mathbf{V})\\ &= \mathbf{U}^{\mathrm{T}}\big(\mathbf{X}_a - \mathbf{X}_a^{\mathrm{o}}\big)(\mathbf{Z}\otimes\mathbf{V})\\ &= \mathbf{U}^{\mathrm{T}}\mathbf{X}_a(\mathbf{Z}\otimes\mathbf{V}) - \mathbf{U}^{\mathrm{T}}\mathbf{X}_a^{\mathrm{o}}(\mathbf{Z}\otimes\mathbf{V})\end{aligned} \tag{42}$$

Here, $\mathbf{X}_a$ and $\mathbf{X}_a^{\mathrm{o}}$ are the unfolded forms of the measured data array and the error-free data array, respectively. Equation (42) shows that a successful estimation of the noise in the compressed space can be obtained if the compression basis sets are chosen to optimally compress the chemical part represented by $\mathbf{X}_a^{\mathrm{o}}$. Two detrimental effects can be foreseen if the chosen basis set does not span the space of $\mathbf{X}_a^{\mathrm{o}}$ properly. The first is related to the loss of meaningful chemical information during the projection step and it is common to PARAFAC and MLPARAFAC. The second is a direct consequence of the first one and related to the inclusion of chemical variability in the error covariance matrix as if it were noise. Clearly, the second detrimental effect will only affect MLPARAFAC since PARAFAC does not use any noise information. In order to prevent these effects when compression is used with MLPARAFAC, it is necessary to retain as much variation as possible. This alternative is not advisable for PARAFAC, since including a large amount of variation can increase the uncertainty of the estimates, but in the case of MLPARAFAC there is no danger of this, since this meaningless variation (noise) will be down-weighted via the error covariance matrix during the estimation process.

It is well known that, in practice, $\mathbf{X}_a^{\mathrm{o}}$ is not generally available; hence, in the absence of *a priori* knowledge, the error-free data array is replaced by its best unbiased estimate (considering the normal assumption), which is the average array $\bar{\mathbf{X}}_a$ calculated by obtaining replicates of the measurements. For practical applications, Equation (42) becomes Equation (43):

$$\mathbf{N}_a = \mathbf{U}^{\mathrm{T}}\mathbf{X}_a(\mathbf{Z}\otimes\mathbf{V}) - \mathbf{U}^{\mathrm{T}}\bar{\mathbf{X}}_a(\mathbf{Z}\otimes\mathbf{V}) = \mathbf{Y}_a - \bar{\mathbf{Y}}_a \tag{43}$$

Equation (43) also unveils another important practical issue regarding the selection of the compression basis sets, indicating that the optimal basis set will be obtained as a Tucker3 decomposition of the mean array $\underline{\mathbf{X}}$. The compressed error covariance matrix Ξ_a will be calculated using a set of R replicates as shown in Equation (44):

$$\Xi_a \approx \frac{1}{(R-1)}\sum_{r=1}^{R}\left[\mathrm{vec}\Big\{\big(\mathbf{Y}_a^r - \bar{\mathbf{Y}}_a\big)^{\mathrm{T}}\Big\}\mathrm{vec}\Big\{\big(\mathbf{Y}_a^r - \bar{\mathbf{Y}}_a\big)^{\mathrm{T}}\Big\}^{\mathrm{T}}\right] \tag{44}$$

Based on the theoretical expressions derived in this section, a sequence of steps to prepare the data for the most general MLPARAFAC algorithm is shown in Table I.

It is important to note that, although this strategy was explained for the compression of all three orders, it can also be applied to the compression of one or two orders in a very straightforward manner. For example, if only mode A is compressed, Equation (38) will become Equation (45), since in that case $\mathbf{Z}$ and $\mathbf{V}$ will be the identity matrix of orders J and K, respectively and $(\mathbf{Z}\otimes\mathbf{V}) = \mathbf{I}_{JK}$.

$$\mathbf{Y}_a = \mathbf{U}^{\mathrm{T}}\mathbf{X}_a \tag{45}$$

This result is equivalent and symmetric for all the orders. Therefore, if an order different from A is to be compressed, the data will be unfolded, keeping the desired order unmodified, and multiplied by the optimal base describing this

Table I. Algorithm for the MLPARAFAC algorithm using compression.

1. Given R replicates of an $I \times J \times K$ cube of data $\underline{\mathbf{X}}$. The algorithm starts by calculating a Tucker-3 model for the average cube of data, $\bar{\underline{\mathbf{X}}}$

$$[\mathbf{U}, \mathbf{V}, \mathbf{Z}, \bar{\underline{\mathbf{Y}}}] = \text{tucker3}(\bar{\underline{\mathbf{X}}}, P) \tag{T1}$$

2. For each replicate, unfold $\underline{\mathbf{X}}^r$, retain the first order and regress $\mathbf{X}_a^r$ onto the subspace spanned by $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{Y}$ in order to calculate $\mathbf{Y}_a^r$ for each replicate as shown in Equation T2:

$$\mathbf{Y}_a^r = \mathbf{U}^{\mathrm{T}}\mathbf{X}_a^r(\mathbf{Z} \otimes \mathbf{V}) \tag{T2}$$

Using all the $\mathbf{Y}_a^r$, estimate the error covariance matrix in the compressed subspace, represented by Ξ_a in Equation T3.

$$\Xi_a \approx \frac{1}{(R-1)} \sum_{r=1}^{R} \left[\mathrm{vec}\left\{ (\mathbf{Y}_a^r - \bar{\mathbf{Y}}_a)^{\mathrm{T}} \right\} \mathrm{vec}\left\{ (\mathbf{Y}_a^r - \bar{\mathbf{Y}}_a)^{\mathrm{T}} \right\}^{\mathrm{T}} \right] \tag{T3}$$

3. Submit Ξ_a and each $\mathbf{Y}_a^r$ to the MLPARAFAC algorithm previously introduced until convergence is achieved.

$$[\boldsymbol{\Delta}^r, \boldsymbol{\Theta}^r, \boldsymbol{\Phi}^r] = MLPARAFAC(\mathbf{Y}^r, \Xi_a, P) \tag{T4}$$

4. Using the following relationships, the uncompressed MLPARAFAC loadings can be obtained.

$$\begin{aligned} \mathbf{A}^r &= \mathbf{U}\boldsymbol{\Delta}^r \\ \mathbf{B}^r &= \mathbf{V}\boldsymbol{\Theta}^r \\ \mathbf{C}^r &= \mathbf{Z}\boldsymbol{\Phi}^r \end{aligned} \tag{T5}$$

order. Equivalently, if more than one order needs to be compressed, this methodology can be individually repeated for both orders, including a folding and unfolding intermediate step between the multiplication by each basis set.

In the experimental part of this paper, a number of simulated data sets will be used to test the performance of the compression approach under different conditions, such as the level of noise and the amount of structure in the chemical data. Also, a comparative study between Tucker3 and PCA loadings will be carried out to confirm the theoretical results.

3. EXPERIMENTAL

3.1. Data Sets

Since the objective of this work is to introduce the theoretical basis and test the statistical properties and performance of a number of simplified alternatives of the MLPARAFAC algorithm, all of the data sets employed in this work were simulated so that the rank and error structure could be known with confidence. Experimental results will be presented in a companion paper to examine the performance of the algorithm for real experimental systems. Although a wide range of simulations were carried out, the results from only six data sets are presented here to support the main conclusions. In all cases, the data sets were relatively small, since the studies generally involved statistical validation requiring numerous runs.

Data Sets 1–5 share the same noise-free structure. This structure was a rank-three data set of dimensions $12 \times 15 \times 6$ used to test the statistical characteristics and the performance of the different algorithms introduced. The loadings for mode A were represented by a 12×3 matrix drawn from a uniform distribution of random numbers from zero to three ($U(0,3)$). Similarly, $\mathbf{B}$ was a 15×3 matrix from $U(0,2)$, and $\mathbf{C}$ was a 6×3 matrix from $U(0,5)$. The error-free data were generated using the well-known PARAFAC model, yielding the 12×90 matrix of error-free data, unfolded to maintain the A mode. Each data set is used to generate 100 replicates obtained by adding this noise-free structure to different realizations of the following error structures.

The matrix of measurement errors for Data Set 1 was a 12×90 matrix with a very simple structure. The simplest noise structure studied in this paper (Case 1A) was imposed on this data set. Initially, six different 12×15 matrices of normally distributed random numbers drawn from $N(0,0.1)$ were generated. These matrices were individually treated with a 7-point moving average filter along each row in order to produce error covariance. At the boundaries of the error matrix, the filter was wrapped around the opposite side in order to eliminate edge effects. Since these error matrices were individually treated with the same filter, this approach produced correlation among the measurements in one mode, and it is identical for all the slices (Case 1A). Although this approach is not particularly realistic, it represents a general case for which the covariance structure could be easily predicted. Finally, the error-free data were added to the noise matrix in order to generate the data set.

The matrix of measurement errors for Data Set 2 was a created in a very similar fashion to the matrix of measurement errors for Data Set 1. The only difference is that each of the six different 12×15 matrices of normally distributed random numbers drawn from $N(0,0.1)$ were individually multiplied by a different filter matrix. The filter matrices were constructed from moving average filters (wrapped around the opposite side in order to eliminate edge effects) of dimensions 3, 5, 7, 7, 9, 5. Since these error matrices were individually treated with the same filter, this approach produced correlation among the measurements in one mode, and different from slice to slice (Case 1B).

The noise matrix of Data Set 3 was created to introduce correlated noise in two orders. Initially, a 12×90 matrix of normally distributed random numbers drawn from $N(0,0.1)$ was generated. This matrix was treated with a 67-point moving average filter along each row in order to produce error covariance. Since the error matrix was unfolded to maintain mode A, this approach produced the same row correlation among the measurements in the other two other modes (Case 1C).

The noise matrix of Data Set 4 was constructed in a similar way to the noise matrix of Data Set 3. However, different-size moving average filters were used along each row in order to produce error covariance among the measurements in the two other modes but with a different structure for each row. Twelve different moving average filter matrices with sizes in the range between 53 and 77 points were used (Case 1D).

The matrix of measurement errors for Data Set 5 was created to have the most complex noise structure studied in this paper (Case 1E). Initially, a 12×90 matrix of normally distributed random numbers drawn from $N(0,0.1)$ was generated. This matrix was vectorized by stacking the transposed rows on top of each other producing a 1080×1 vector that was multiplied by a 1080 × 1080 filter matrix. This filter matrix was formed by accommodating eight 135 × 135 filter matrices of a 127-point moving average filter along the diagonal to produce error covariance. As before, the boundaries of the filter matrices were wrapped around the opposite side in order to eliminate edge effects. Considering that dimension of the filter matrices was 135 × 135 this approach produced correlation among the measurements in three modes. Again, the error-free data were added to the noise matrix in order to generate the data set.

The matrix of measurement errors for Data Set 6 was also created to represent the most complex in case 1E but with a more heterogeneous structure. The noise structure was constructed in a similar fashion to Data Set 5, but in this case eight different 135 × 135 filter matrices of 101-, 133-, 109-, 131-, 119-, 121-, 127-, and 97-point moving average filters along the diagonal were used to produce error covariance. As before, the boundaries of the filter matrices were wrapped around the opposite side in order to eliminate edge effects. Considering that dimension of the filter matrices was 135 × 135 and each individual filter matrix was created with a different number of points, this approach produced correlation with a very heterogeneous structure among the measurements in three modes.

Data Sets 7–10 were rank-three data sets of dimensions 32 × 128 × 8 and were used to test the compression approach for different conditions of noise and data structure. In a generic way, the data sets are generated to contain the same broad spectral characteristics commonly observed in fluorescence excitation/emission matrices. The pure components for modes A and B were generated by adding Gaussian peaks of random means and standard deviations. The position of the center of each peak is a random number drawn between one and the largest channel number. The width of each peak is also drawn from a uniform distribution with a range between 10 and 40 ($U(10, 40)$). The spectra were normalized to unit length in all cases. The information about the intensity for each component is carried in mode C, in which the pure component concentrations are represented by an 8 × 3 matrix drawn from a uniform distribution of random numbers from 0 to 30 (U(0,30)). Two different issues affecting the compression were investigated with these data sets: the amount of chemical information contained in the data and the level of noise affecting the data. Data Sets 7 and 8 were constructed with unimodal components for modes A and B. Data Sets 9 and 10 were constructed using components obtained by adding five Gaussian peaks for each component. All the data sets were constructed using the same error structure. In all cases, an error structure equivalent to Data Set 1 was used to compare the results obtained after compression with results obtained without any compression using an algorithm which is optimal but not computationally involved. For this case, a 61-point moving average filter was used for each row. The error structure is the same in each case but the signal-to-noise ratio (SNR) is varied to test the performance of compression with respect to the noise. Data Sets 7 and 9 have a SNR = 1000 and Data Sets 8 and 10 have a SNR = 250. The SNR values reported here represent the best case scenario, since they are calculated as the ratio between the maximum peak for the most concentrated sample and the value for the noise defined as three times the standard deviation. Therefore, there will be parts of these data sets with poorer SNR. All the data sets utilize 25 replicates calculated by adding the respective noise-free data and a different realization of the noise structure described for each data set.

3.2. Computational aspects

All calculations performed in this work were carried out on a Sun Ultra 60 workstation with 2 × 300 MHz processors and 512 MB of RAM and a 3.2 GHz Pentium-IV PC with 1 GB of RAM. All programs were written in-house using Matlab 6.0 (The MathWorks, Inc., Natick, MA) with the exception of the PARAFAC and TUCKER3 functions that were run using the N-Way Toolbox [29].

4. RESULTS AND DISCUSSION

In this section, the estimation equations for each method will be validated using Data Sets 1–5 in order to cover different possible scenarios. In addition to the validation discussion, some general conclusions will be drawn about the merits of using the different algorithms based on the quality of the results and computational efforts invested to get them. Data Sets 7–10 will be used to compare the quality of the results for the compression approach with standard PARAFAC using different scenarios (noise level, amount of structural information, and different basis sets) which have a very simple error structure in order to use the simplification developed for Case 1A as a benchmark value.

4.1. Statistical validation

In order to validate the various proposed algorithms, it was necessary to verify that they yield the maximum likelihood solution. This can be accomplished by exploiting the statistical characteristics of S^2 values for the correct model. This methodology has been explained elsewhere [8,17] but it will be briefly reproduced here for the sake of completeness. Operationally, this is done by analyzing replicate data sets, each with the same matrix of error-free data and the same error structure, but with different realizations of the measurement error each time. If the distribution of S^2 values for these replicates follows a χ^2 distribution with the appropriate degrees of freedom [17,30], it can then be concluded that the algorithm is finding the maximum likelihood solution. Probability plots are used in this work to make this comparison. Initially, the replicate data sets (normally 100 replicates) are analyzed and the S^2 values are stored. Then, the S^2 values are sorted from the smallest to the largest and assigned a cumulative probability according to their position in the list; this is called the observed probability. For instance, the third element in the list would be assigned an observed probability of $2/n$, where n is the number of replicates. The expected probability is then calculated using the χ^2 distribution. The cumulative probability density function for χ^2 can be calculated using the incomplete gamma function as shown in Equation (46):

$$P(S^2|\nu) = \Gamma_{\text{inc}}\left(\frac{S^2}{2}, \frac{\nu}{2}\right) \tag{46}$$

where ν is the number of degrees of freedom [17]. If the two distributions are the same, a plot of the observed probabilities versus the expected probabilities should yield a straight line with a slope of unity. If the model is insufficient to account for the systematic variance, either because the form of the model is incorrect or the parameters are suboptimal, then the points of the plot will lie above the ideal line. If the model accounts for an excessive amount of variance, (i.e., the estimated rank is too high and measurement variance is modeled), the points will lie below the ideal line.

Figure 2 shows the probability plots obtained when all of the algorithms introduced in this work; in addition, the general MLPARAFAC algorithm (without compression) and PARAFAC are used to estimate Data Sets 1–5. The general MLPARAFAC was included as a benchmark since it can accommodate any covariance structure. Figure 2 shows a perfect trend, starting with all the methods but PARAFAC providing optimal models and ending with only the general MLPARAFAC algorithm providing an optimal model. As the complexity of the error structure increases, the methods designed to handle simpler error structures join the PARAFAC method, indicating the suboptimality of their estimates. Even though this trend was theoretically expected since each data set was constructed mimicking the error structure and therefore the objective function used to derive the estimation equation for each method, the results show from a numerical point of view the correctness of the estimation expressions for each case and how all of these methods are different simplified instances of a general class of method. It is important to emphasize that this particular methodology is very sensitive to suboptimal solutions; therefore, it should not be used to compare the quality of different solutions.

4.2. Model quality and performance

The preceding section dealt with the statistical validation of the maximum likelihood estimation process, but nothing has been said about the quality of the estimates obtained using these new algorithms. Although it has been previously demonstrated [17] that MLPARAFAC estimates are closer to the true underlying factors than the PARAFAC estimates, two reasonable questions are still not answered: (1) How do the MLPARAFAC estimates from different simplifications behave as the complexity of the error structure increases? and (2) what is the computational price paid for the increment on complexity. Both questions will be answered using simulated data. The computational workload and the quality of the data will be assessed using the average time needed for convergence and loading vector angles, respectively. Both magnitudes will be calculated using 100 replicates. In order to put this comparison into context, the value for each

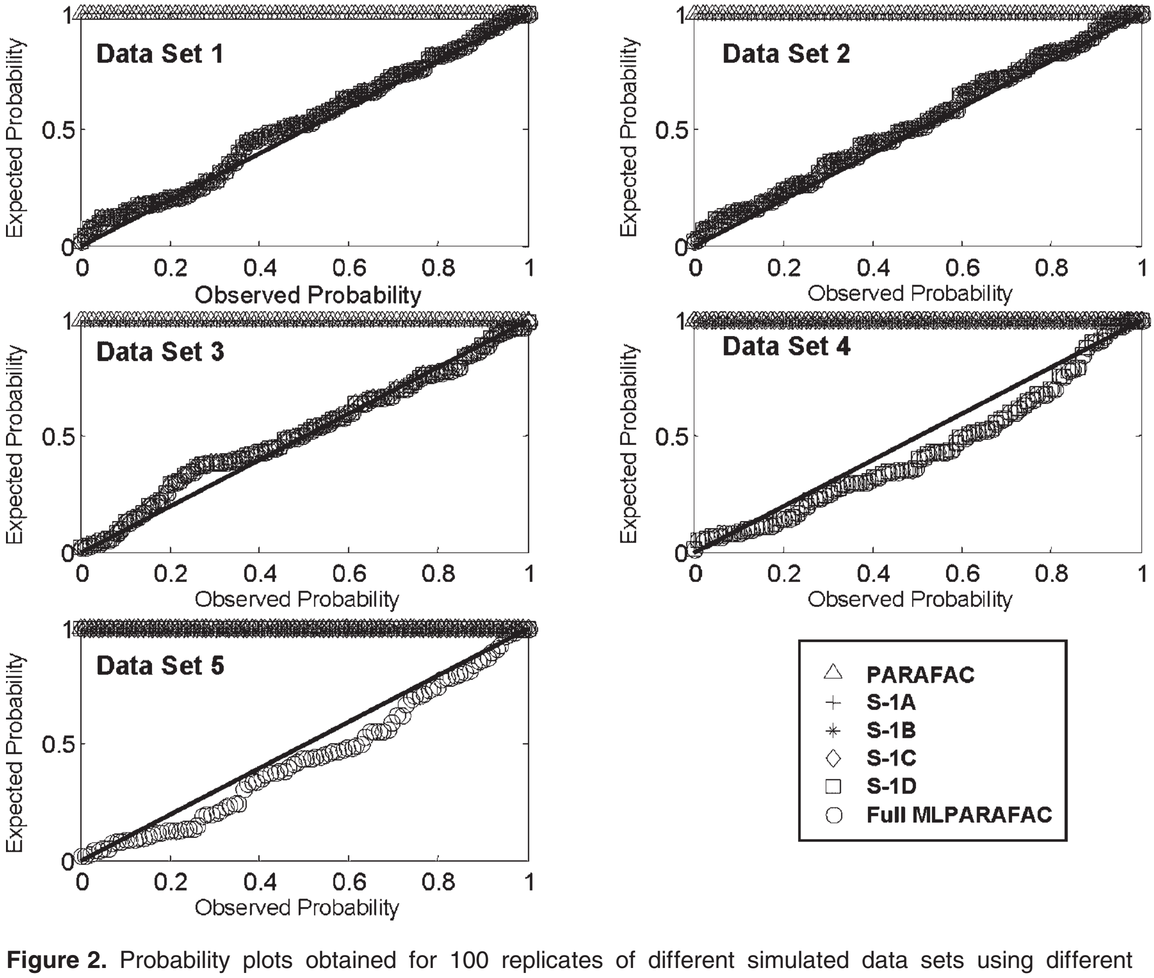

Figure 2. Probability plots obtained for 100 replicates of different simulated data sets using different algorithms such as parallel factor analysis (PARAFAC) (△); simplifications 1A (+), 1B (∗), 1C (◇), 1D (□); and full maximum likelihood parallel factor analysis (MLPARAFAC) (○). The solid line with unity slope indicates ideal behavior for maximum likelihood estimation.

method relative to the value for the PARAFAC model will be used.

As mentioned above, the quality of the estimates will be measured as the closeness of estimates to the true factors using vector angles as a figure of merit. This figure of merit is the angular difference between the true loading vectors and the estimated loading vectors in each mode. For example, the vector angle between two loading vectors in mode A is given by:

$$\theta_p^a = \cos^{-1}\left(\frac{\hat{\mathbf{a}}_p^{\mathrm{T}}\mathbf{a}_p}{\|\hat{\mathbf{a}}_p\|\|\mathbf{a}_p\|}\right) \tag{47}$$

where $\mathbf{a}_p$ and $\hat{\mathbf{a}}_p$ are the true and estimated values for the pth loading vector of $\mathbf{A}$. Analogous equations can be used for the other orders. Smaller angles mean a greater similarity, so by comparing the vector angles obtained by the different simplifications of MLPARAFAC with those of PARAFAC, the agreement with the true vector can be assessed. An alternative measure is the correlation coefficient of the vectors, which is simply the term in parentheses, but since this approaches unity with small differences, it is less sensitive. The quality of the estimates as well as the computation time will be compared in a relative fashion with respect to the corresponding values for the PARAFAC model. Equations (48) and (49) represent the expressions to calculate the relative average angle (RA) and the relative computation time (RCT):

$$\mathrm{RQ} = \frac{\bar{\theta}_{\mathrm{X}}}{\bar{\theta}_{\mathrm{PAR}}} \tag{48}$$

$$\mathrm{RCT} = \frac{\bar{t}_{\mathrm{X}}}{\bar{t}_{\mathrm{PAR}}} \tag{49}$$

Two completely different scenarios will be explored in order to have a broad view of the problem, since the degree to which these results will be extendable to a given application depends on the nature of the application and the characteristics of the noise. Data Sets 1 and 5 will be used since they represent very different scenarios in which clear comparisons can be made and conclusions drawn. The validation results showed that all of the simplifications provided optimal estimates for Data Set 1; therefore, this is a good scenario to test the computational advantages of using simpler algorithms over more complex algorithms when the data merit the simplification. Data Set 6 has a more complex error structure and these simplifications are also used to treat it.

Figures 3 and 4 show the results for the comparison in terms of quality and performance, respectively, when different simplifications are used. As expected, all the methods but PARAFAC provided the same results for Data Set 1 in terms of quality in Figure 3, since the error structure used was the simplest case. However, when the time employed to reach the convergence is taken into account (Figure 4), it is possible to appreciate the advantages of using simplified algorithms when the data at hand merit the use of a simplification.

In Figure 3, the relative average angle for Data Set 5 exhibits a nice trend, showing an improvement of the quality of the results as the complexity of the algorithms used increases. PARAFAC and general MLPARAFAC are located at the two extremes, corresponding to the methods providing the worst and best estimates. Again, a positive correlation between the complexity of the algorithm and the time needed to obtain the best possible solution is observed, indicating in this case that a better solution will require the use of more computational effort. These results were expected, since the common wisdom tends to assume that the application of more complex algorithms (which in turn translates into error covariance matrices that are bigger

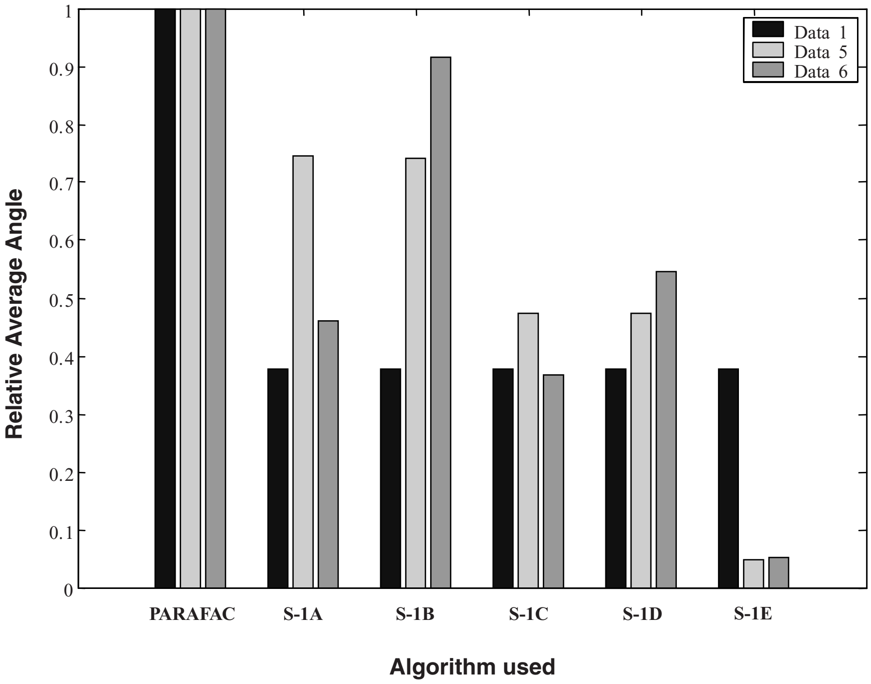

Figure 3. Comparison of the improvements in the quality of the estimates obtained for different MLPARAFAC algorithms for three characteristic data sets. The quality is measured using the relative average vector angle with respect to PARAFAC and the results are based on 100 replicates.

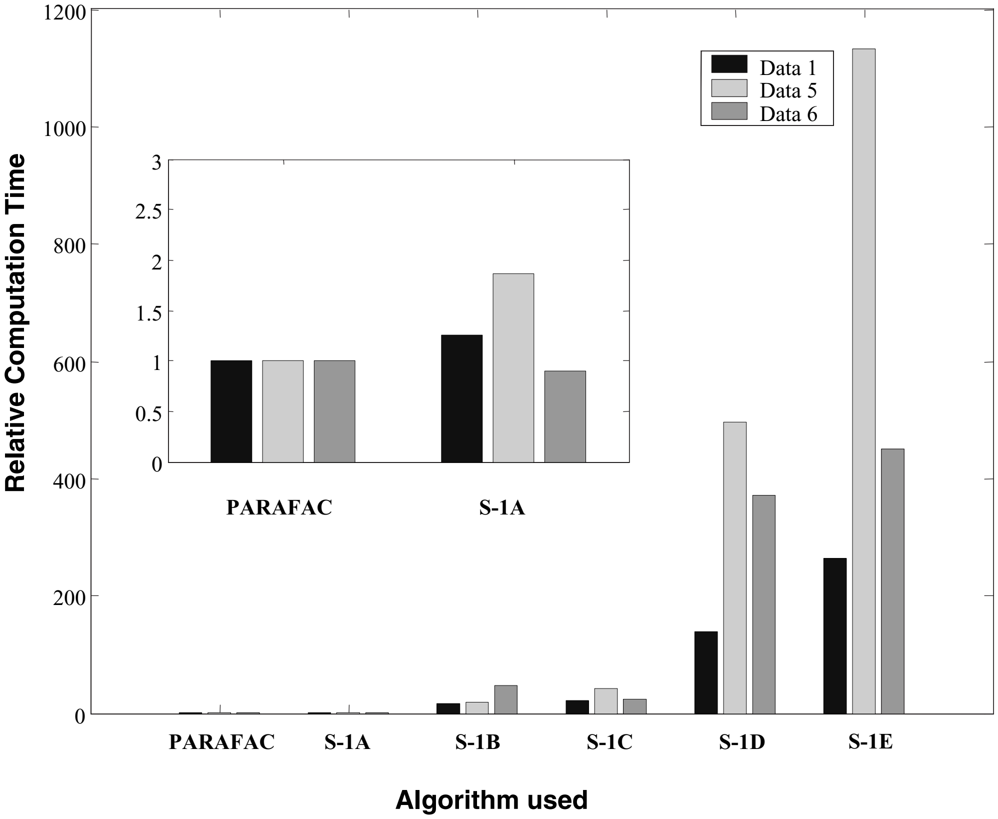

Figure 4. Comparison of the time utilized by different MLPARAFAC algorithms for three characteristic data sets. The performance is measured using relative time with respect to PARAFAC and the results are based on 100 replicates.

and richer in information) will provide estimates of a better quality. Even though the error structures in real applications tend to be simple in general [13], the authors believe that in this particular case, the perfect monotonic trend in quality was mainly the result of an oversimplified (i.e., very symmetric) error structure. To avoid a misleading conclusion in this regard, this issue was further explored using a Data Set 6, which has a similar but less symmetric error structure.

The results for Data Set 6 are quite surprising. For Data Set 5, a trend showing a monotonic improvement in the quality of the results with the complexity of the algorithm used was observed, but the simulations for Data Set 6 show a very different scenario. PARAFAC and general MLPARAFAC were the only methods that coincide with the expected trend results. The remainder of the simplifications did not provide a clear trend in quality. For instance, the simplifications assuming that the errors are correlated along one order and are the same everywhere (Case 1B) gave estimates that are as good as the ones provided by the methodology assuming errors with the same structure affecting two orders (Case 1D). Another striking inconsistency evident from Figure 3 is that the quality of the results for the simplifications representing case 1B and 1D were worse than the quality of the results for Cases 1A and 1C, respectively. The mathematical theory behind these expressions makes Cases 1A and 1C subsets of the more general implementations representing Cases 1B and 1D, respectively, when the models are properly used. Therefore, all these inconsistent results clearly illustrate the importance of a thorough characterization of the error structure, since the applications of an incorrect model can significantly degrade the quality of the result. It is important to note that the comparison of these methodologies from a computational point of view is meaningless for Data Set 6, since all of them produced a variety of sub-optimal models.

In reality, data commonly found in chemistry will have a behavior closer to the scenario illustrated by the simulations using Data Set 5. Probably, the error structure will not be exactly equivalent to the error covariance matrix used to derive the expression for a particular simplification, but it will not depart to the extent that Data Set 6 did to make the simplifications useless. However, it is important to fully characterize the error structure in order to apply the most suitable algorithm given the data set at hand in order to avoid erratic results such as the ones shown for Data Set 6. Unfortunately, due to the length and scope of this paper, only exact mathematical simplifications were shown, but in a companion paper to this work, a number of important guidelines will be introduced and used with different experimental data sets in order to cover more gray scenarios.

Finally, it is important to emphasize that, although only the results for three data sets were shown, many different data sets with the same characteristics of Data Sets 1, 5, and 6 were used to ensure the generality of the conclusions drawn.

4.3. Compression results

In the results shown in the previous section, general MLPARAFAC always provided the best solution, provoking the question: why not use general MLPARAFAC for every case? There are two reasons for this. The first is that general MLPARAFAC usually takes more time to produce the estimates, as already demonstrated. The second reason is that the previous results used general MLPARAFAC for a very small data set. For a more typical size data set, general MLPARAFAC cannot be applied directly due to storage and memory limitations. In order to overcome these limitations, a compression strategy was formulated. This section will show that, even though compressed MLPARAFAC will not give exactly the same results as general MLPARAFAC, the solutions will generally be superior to the PARAFAC solution. Figure 5 shows the comparative results for different cases (Data Sets 7, 8, 9, and 10) and compression basis sets with respect to PARAFAC and general MLPARAFAC. In general, these results clearly demonstrate that improved estimates of loadings with respect to PARAFAC can be obtained from the general algorithm when information about the measurement error structure is compressed and is incorporated into the modeling process in the correct way. As already noted, the extent to which these improvements will be significant for a given application depends on the nature of the chemical data and the level and structure of the noise affecting the measurements. As can be seen from Figure 5, when the amount of information related to the chemical data increases, a larger number of factors are needed to yield better estimates using the compressed data. For Data Sets 7 and 8, which are constructed by unimodal components, six factors are enough to produce good results while for Data Sets 9 and 10, 10 components are necessary to produce similar results. It is important to note, that Tucker3 and Tucker1 (PCA) basis sets produces very similar results in all cases, at least to the extent of these simulations. The different noise levels produce an equivalent worsening of all methods, indicating that this does not play an important role in the compression strategy. In addition to the PARAFAC, general MLPARAFAC, Tucker1-MLPARAFAC, and Tucker3-MLPARAFAC, Tucker3-PARAFAC is also included to dissect the improved results with respect to PARAFAC in its two most important contributions: the effects of compression and the use of the error information in the estimation process. It can be observed in all cases that although the compression step by itself produced some improvement in the results, the use of compression and weighting yield much better estimates. It is also important to comment about the worsening of the estimates as the number of components increases, shown as a trend in all cases when the compressed data are treated with standard PARAFAC. This situation does not occur when MLPARAFAC is used due to its capacity to down-weight noisy regions as anticipated in the theory section.

In reality, the difference between the quality of the estimates of compressed MLPARAFAC and PARAFAC will not be as large as the differences encountered in the simulation studies, since the results presented here were obtained assuming an absolute knowledge of the measurement error covariance matrix, while in practice this is typically estimated on the basis of replicate measurements and hence may be less reliable. Therefore, the benefits of including measurement error information must be weighed against the detrimental effects of including poor-quality information. In many cases, it will be more advisable to use one of the previous simplifications because, in those situations, the advantages gained by pooling

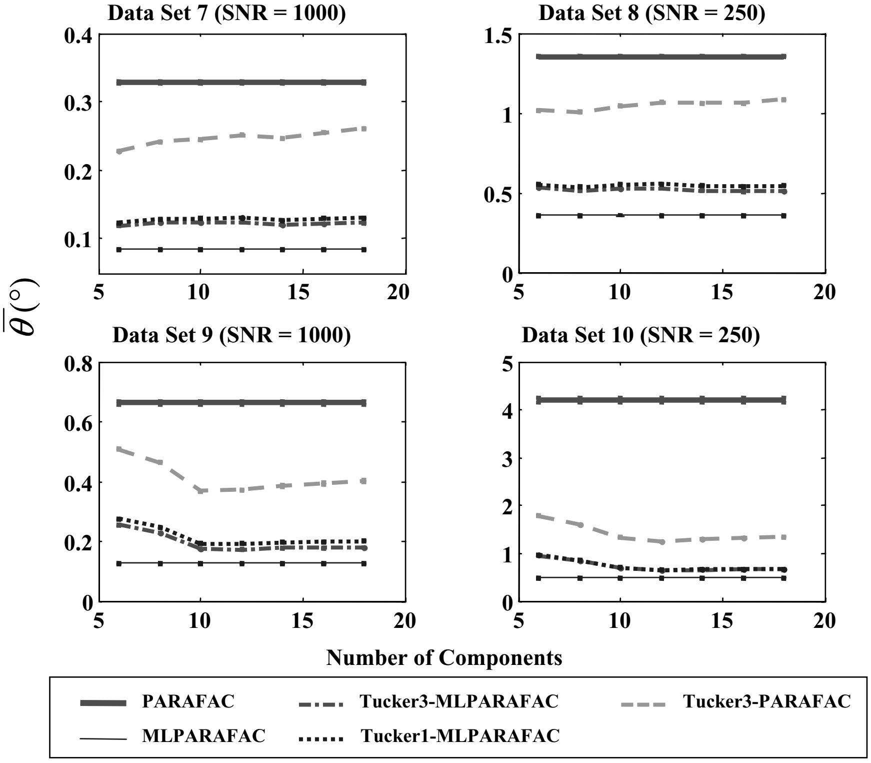

Figure 5. Comparison of the quality, in terms of average vector angle, of the estimates obtained for four different data sets when PARAFAC and general MLPARAFAC are employed on the original data and on compressed data. Tucker1 (PCA) and Tucker3 loadings were used as compression basis sets.

error covariance estimates may outweigh the benefits of using the full error covariance matrix.

5. CONCLUSIONS

In this work, the standard practice of expressing the estimation process by minimizing the different formulations of the same objective function was discarded since it does not take into account the loss of symmetry caused by the introduction of error information. A new approach, in which the same objective function is used to estimate the loadings for all the modes, was introduced due to the benefits of locating the noise information in one or two modes as a simple representation and using it equivalently to obtain the estimation equations for each mode.

Four algorithms for carrying out simplified variations of general MLPARAFAC when the data at hand are corrupted by correlated noise affecting one or two orders have been described in this work by using the new approach. Also, a compression step was included prior to the use of general MLPARAFAC for cases where the noise structure is affecting three modes and the volume of data precludes the use of general MLPARAFAC on the raw data.

All of the algorithms were shown to produce maximum likelihood estimates through a comparison of the distribution of the objective function with the χ^2 distribution. It was also shown that the use of simplified algorithms when the data at hand merit the simplification is beneficial from a computational point of view. When the error structure was properly used, the quality of the estimates was the same for all the methods designed to handle this error structure. Two simulated scenarios where the error structure assumed departs from the actual error structure were studied to illustrate the importance of a thorough characterization of the error structure.

The merits of using compressed MLPARAFAC over PARAFAC were studied in different scenarios. Also, no significant differences were found between Tucker3 and Tucker1 basis sets, at least for the data used in the simulation studies.

Although the principles of general MLPARAFAC and a number of simplifications have been established here, a number of more practical aspects related to its application on experimental data remain to be examined. These include issues related to the characterization of the error structure and the application of the different simplifications. These subjects will be the focus of a companion paper.

Acknowledgements

The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Dow Chemical Company (Midland, MI, USA).

REFERENCES

1. Martens H, Næs T. *Multivariate Calibration*. John Wiley & Sons: New York, 1989.
2. de Juan A, Tauler R. Chemometrics applied to unravel multicomponent processes and mixtures. Revisiting latest trends in multivariate resolution. *Anal. Chim. Acta* 2003; **500**: 195.
3. Bro R. PARAFAC. Tutorial and applications. *Chemom. Intell. Lab. Syst.* 1997; **38**: 149.
4. Kiers HAL, ten Berge JMF, Bro R. PARAFAC2—part I. A direct fitting algorithm for the PARAFAC2 model. *J. Chemom.* 1999; **13**: 275.
5. Bro R. Multi-way calibration. Multi-linear PLS. *J. Chemom.* 1996; **10**: 47.
6. Paatero P. A weighted non-negative least squares algorithm for three-way ''PARAFAC'' factor analysis. *Chemom. Intell. Lab. Syst.* 1997; **38**: 223.
7. Wentzell PD, Andrews DT, Hamilton DC, Faber K, Kowalski BR. Maximum likelihood principal component analysis. *J. Chemom.* 1997; **11**: 339.
8. Van Huffel S, Vandewalle J. *The Total Least Squares Problem: Computational Aspect and Analysis*. SIAM: Philadelphia, 1991.
9. Kiers HAL. Weighted least squares fitting using ordinary least squares algorithms. *Psychometrika* 1997; **62**: 251.
10. Martens H, Hoy M, Wise BM, Bro R, Brockhoff PB. Prewhitening of data covariance-weighted pre-processing. *J. Chemom.* 2003; **17**: 153.
11. Bro R, Sidiropoulos ND, Smilde AK. Maximum likelihood fitting using simple least squares algorithms. *J. Chemom.* 2002; **16**: 183.
12. Weisberg S. *Applied Linear Regression*. Wiley: Chichester, 1985.
13. Leger M, Vega-Montoto L, Wentzell PD. Methods for systematic investigation of measurement error covariance matrices. *Chemom. Intell. Lab. Syst.* 2005; **77**: 181.
14. Brown CD, Wentzell PD. Hazard of digital smoothing filters as a preprocessing tool in multivariate calibration. *J. Chemom.* 1999; **13**: 133.
15. Brown CD, Vega-Montoto L, Wentzell PD. Derivative preprocessing and optimal corrections for baseline drift in multivariate calibration. *Appl. Spectrosc.* 2000; **54**: 1055.
16. Liu X, Sidiropoulos N. Cramér–Rao lower bound for low-rank decomposition of multidimensional arrays. *IEEE Trans. Signal Processing* 2001; **49**: 2074.
17. Vega-Montoto L, Wentzell PD. Maximum likelihood parallel factor analysis (MLPARAFAC). *J. Chemom.* 2003; **17**: 237.
18. Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares. *Anal. Chem.* 1964; **36**: 1627.
19. Magnus JR, Neudecker H. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley: Chichester, 1988.
20. Carroll JD, Pruzansky S, Kruskal JB. Candelinc: A general approach to multidimensional analysis of many-ways arrays with linear constraints on parameters, *Psychometrika* 1980; **45**: 3.
21. Apellof CJ, Davison ER. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. *Anal. Chem.* 1981; **53**: 2053.
22. Alsberg BK, Kvalheim OM. Speed improvement of multivariate algorithms by the method of postponed basis matrix multiplications. Part I. Principal component analysis. *Chemom. Intell. Lab. Syst.* 1994; **24**: 31.
23. Alsberg BK, Kvalheim OM. Speed improvement of multivariate algorithms by the method of postponed basis matrix multiplications. Part II. Three-mode principal component analysis. *Chemom. Intell. Lab. Syst.* 1994; **24**: 43.
24. Kiers HAL, Harshman RA. Relating two proposed methods for speed-up of algorithms for fitting two- and three-way principal component and related multilinear methods. *Chemom. Intell. Lab. Syst.* 1997; **39**: 31.
25. Leger MN, Wentzell PD. Maximum likelihood principal component regression on wavelet-compressed data. *Appl. Spectrosc.* 2004; **58**: 855.

26. Tomiši V, Simeon V. Assessment of the effective rank of a (co)variance matrix: a nonparametric goodness-of-fit test. *J. Chemom.* 1993; **7**: 381.
27. Andersson CA, Bro R. Improving the speed of multi-way algorithms. Part I: Tucker3. *Chemom. Intell. Lab. Syst.* 1998; **42**: 93.
28. Kiers HAL, Krijnen WP. An efficient algorithm for PARAFAC of three-way data with large numbers of observation units. *Psychometrika* 1991; **56**: 147.
29. Andersson CA, Bro R. The N-way toolbox for MATLAB. *Chemom. Intell. Lab. Syst.* 2000; **52**: 1.
30. Durell SR, Lee C, Ross RT, Gross EL. Factor analysis of the near-ultraviolet absorption spectrum of plastocyanyn using bilinear, trilinear and quadrilinear models. *Arch. Biochem. Biophys.* 1990; **278**: 148.

APPENDIX

Case 1B: Mode B

This scenario is represented by the following objective function:

$$f = \sum_{k=1}^{K} \operatorname{trace}[(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}] \tag{A1}$$

Defining: $\mathbf{M}_k = \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}$, Equation (A1) can be modified to yield:

$$f = \sum_{k=1}^{K} \operatorname{trace}(\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\mathbf{M}_k^{\mathrm{T}}) \tag{A2}$$

Using standard relations for derivatives of matrices and vectors [19], this gives:

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{B}_{jp}} &= \frac{\partial f}{\partial \mathbf{M}_k}\left(\frac{\partial \mathbf{M}_k}{\partial \mathbf{B}_{jp}}\right)^{\mathrm{T}} \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial \mathbf{M}_k^{\mathrm{T}}}{\partial \mathbf{B}_{jp}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})}{\partial \mathbf{B}_{jp}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}(-\mathbf{A}\mathbf{D}_k\mathbf{E}_{jp}^{\mathrm{T}})^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(2(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(-\mathbf{A}\mathbf{D}_k\mathbf{E}_{jp}^{\mathrm{T}})^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{K} \operatorname{trace}\left(-2\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}} + 2\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\right) \\
&= -2\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\right) \\
&\quad + 2\sum_{k=1}^{K} \operatorname{trace}(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}})
\end{aligned} \tag{A3}$$

Setting this derivative equal to zero to find the minimum leads to:

$$\begin{aligned}
\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}^{\mathrm{T}}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\right) &= \sum_{k=1}^{K} \operatorname{trace}(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}) \\
\sum_{k=1}^{K} \operatorname{trace}\left(\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\right) &= \sum_{k=1}^{K} \operatorname{trace}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{D}_k)
\end{aligned} \tag{A4}$$

Expressing the traces as the product of two vectors [20] yields:

$$\begin{aligned}
&\sum_{k=1}^{K} \operatorname{vec}(\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}}\mathbf{A}\mathbf{D}_k)^{\mathrm{T}}\operatorname{vec}(\mathbf{E}_{jp}) \\
&= \sum_{k=1}^{K} \operatorname{vec}(\mathbf{B})^{\mathrm{T}}\operatorname{vec}(\boldsymbol{\Psi}_k^{-1}\mathbf{E}_{jp}\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{D}_k) \\
&\operatorname{vec}\left(\sum_{k=1}^{K}(\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1})\right)^{\mathrm{T}}\operatorname{vec}(\mathbf{E}_{jp}) \\
&= \operatorname{vec}(\mathbf{B})^{\mathrm{T}}\sum_{k=1}^{K}(\boldsymbol{\Psi}_k^{-1}\otimes\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{D}_k)\operatorname{vec}(\mathbf{E}_{jp})
\end{aligned} \tag{A5}$$

Equation (A5) is one of the *JP* equations necessary to estimate the loadings of **B**. The rest of the equations are obtained as the right and left parts of this equation are multiplied by the different vectorized $\mathbf{E}_{jp}$ matrices. Since this term is completely independent in both sides of the equation, the process can be carried out in a straightforward manner using a matrix **E** formed as $[\operatorname{vec}(\mathbf{E}_{II})\ \operatorname{vec}(\mathbf{E}_{21})\ldots\operatorname{vec}(\mathbf{E}_{JP}]$. A closer look of this matrix shows that **E** is the identity matrix of order *JP*, making the multiplication theoretically sound but numerically unnecessary and providing Equation (A6) to estimate the loading of B:

$$\operatorname{vec}(\mathbf{B})^{\mathrm{T}} = \left(\sum_{k=1}^{K}(\boldsymbol{\Psi}_k^{-1}\otimes\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{B}\mathbf{D}_k)\right)^{-1}\operatorname{vec}\left(\sum_{k=1}^{K}(\mathbf{D}_k\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1})\right) \tag{A6}$$

Case 1B: Mode C

Similarly, this objective function is used to represent the following scenario. It is important to realize that it can be expressed as the summation over the *K* slices:

$$f = \sum_{k=1}^{K} \mathit{trace}[(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}] = \sum_{k=1}^{K} f_k \tag{A7}$$

Defining: $\mathbf{M}_k = \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}$, Equation A7 can be modified to yield:

$$f_k = \operatorname{trace}(\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\mathbf{M}_k^{\mathrm{T}}) \tag{A8}$$

Using standard relations for derivatives of matrices and vectors [19], this gives:

$$\begin{aligned}
\frac{\partial f_k}{\partial \mathbf{C}_{kp}} &= \frac{\partial f_k}{\partial \mathbf{M}_k}\left(\frac{\partial \mathbf{M}_k}{\partial \mathbf{C}_{kp}}\right)^{\mathrm{T}} \\
&= \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial \mathbf{M}_k^{\mathrm{T}}}{\partial \mathbf{C}_{kp}}\right) \\
&= \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}\frac{\partial(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}}{\partial \mathbf{C}_{kp}}\right) \\
&= \operatorname{trace}\left(2\mathbf{M}_k\boldsymbol{\Psi}_k^{-1}(-\mathbf{A}\mathbf{E}_{pp}\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}\right) \\
&= \operatorname{trace}\left(2(\mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_k^{-1}(-\mathbf{A}\mathbf{E}_{pp}\mathbf{B}^{\mathrm{T}})^{\mathrm{T}}\right) \\
&= \operatorname{trace}\left(-2\cdot\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}} + 2\cdot\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}}\right) \\
&= -2\cdot\operatorname{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}}\right) \\
&\quad + 2\cdot\operatorname{trace}(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}})
\end{aligned} \tag{A9}$$

Setting this derivative equal to zero to find the minimum leads to:

$$\begin{aligned}
\mathrm{trace}\left(\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}}\right) &= \mathrm{trace}(\mathbf{A}\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}})\\
\mathrm{trace}(\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}) &= \mathrm{trace}(\mathbf{D}_k\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}}\mathbf{A})\\
\mathrm{vec}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}}\mathbf{A})^{\mathrm{T}}\mathrm{vec}(\mathbf{E}_{pp}) &= \mathrm{vec}(\mathbf{D}_k)^{\mathrm{T}}\mathrm{vec}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\mathbf{E}_{pp}\mathbf{A}^{\mathrm{T}}\mathbf{A})\\
\mathrm{vec}(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{X}_k^{\mathrm{T}})^{\mathrm{T}}\mathrm{vec}(\mathbf{E}_{pp}) &= \mathrm{vec}(\mathbf{D}_k)^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\mathbf{A}\otimes\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B})\mathrm{vec}(\mathbf{E}_{pp})
\end{aligned} \tag{A10}$$

The last expression in Equation (A10) is one of the *PP* equations necessary to estimate the loadings of *k* row of matrix **C**. The rest of the equations are obtained as the right and left parts of this equation are multiplied by the different vectorized $\mathbf{E}_{pp}$ matrices. Since this term is completely independent in both sides of the equation, the process can be carried out in a straightforward manner, using a matrix **E** formed as $[\mathrm{vec}(\mathbf{E_{11}})\ \mathrm{vec}(\mathbf{E_{22}})\ldots\mathrm{vec}(\mathbf{E}_{PP})]$. Contrary to what happened in the estimation of mode B, matrix **E** is used to pick the relevant elements in both members, since we are only interested in the estimation of the elements located in the diagonal of $\mathbf{D}_k$. Therefore, Equation (A11) is used to estimate the loading of C in a row by row fashion:

$$\mathbf{c}_k = \left[\left(\mathbf{B}^{\mathrm{T}}\boldsymbol{\Psi}_k^{-1}\mathbf{B}\otimes\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathrm{vec}\left(\mathbf{A}^{\mathrm{T}}\mathbf{X}_k\boldsymbol{\Psi}_k^{-1}\mathbf{B}\right)\right]^{\mathrm{T}}\mathbf{E} \tag{A11}$$

Case 1C: Mode B

This scenario is well represented by the following objective function:

$$f = \mathrm{trace}[(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})^{\mathrm{T}}] \tag{A12}$$

In order to make the equation more tractable the following modifications were applied:

$$\mathbf{M} = (\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})^{\mathrm{T}}\text{and } \tilde{\mathbf{A}} = \mathbf{A}\mathbf{I}_a \text{ to yield :}$$

$$f = \mathrm{trace}(\mathbf{M}^{\mathrm{T}}\boldsymbol{\Psi}_a^{-1}\mathbf{M}) \tag{A13}$$

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{B}_{jp}} &= \frac{\partial f}{\partial \mathbf{M}}\left(\frac{\partial \mathbf{M}}{\partial \mathbf{B}_{jp}}\right)^{\mathrm{T}}\\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}\frac{\partial \mathbf{M}}{\partial \mathbf{B}_{jp}}\right)\\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}\frac{\partial\left[(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})^{\mathrm{T}}\right]}{\partial \mathbf{B}_{jp}}\right)\\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}(-\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}})\right)\\
&= \mathrm{trace}\left(2\cdot\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a^{\mathrm{T}} - (\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}})(-\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}})\right)\\
&= -2\cdot\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{X}_a^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}}\right)\\
&\quad + 2\cdot\mathrm{trace}(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}}
\end{aligned} \tag{A14}$$

Setting this derivative equal to zero to find the minimum leads to:

$$\begin{aligned}
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{X}_a^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}}\right) = \mathrm{trace}(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{E}_{jp})^{\mathrm{T}})\\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{E}_{jp})\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \mathrm{trace}(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{E}_{jp})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})\\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{I}_K\otimes\mathbf{E}_{jp})(\mathbf{C}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) =\\
&\mathrm{trace}(\boldsymbol{\Psi}_a^{-1}(\mathbf{I}_K\otimes\mathbf{E}_{jp})(\mathbf{C}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}^{\mathrm{T}}\otimes\mathbf{I}_P)(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}}))
\end{aligned} \tag{A15}$$

Equation (A15) becomes Equation (A18) using the matrices **L** and **R** as defined in Equation (A16) and (A17), respectively.

$$\mathbf{R} = (\mathbf{C}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a \tag{A16}$$

$$\mathbf{L} = (\mathbf{C}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{I}_p)^{\mathrm{T}} \tag{A17}$$

$$\begin{aligned}
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{I}_K\otimes\mathbf{E}_{jp})\mathbf{R}\right)\\
&\quad = \mathrm{trace}(\boldsymbol{\Psi}_a^{-1}(\mathbf{I}_K\otimes\mathbf{E}_{jp})\mathbf{L}(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}}))
\end{aligned} \tag{A18}$$

Equation (A18) can be expressed as Equation (A20) when the matrices forming both members of the previous equation are partioned as shown in Equation (A19). Matrices ${}^{mn}\boldsymbol{\Psi}_a^{-1}$, ${}^{nm}\mathbf{R}$ and ${}^{nm}\mathbf{L}$ have orders $J\times J$; $P\times J$, and $P\times P$ respectively.

$$\begin{aligned}
&\mathrm{trace}\left(\left[\begin{array}{c|c|c} {}^{11}\boldsymbol{\Psi}_a^{-1} & \cdots & {}^{1K}\boldsymbol{\Psi}_a^{-1}\\ \hline \vdots & \ddots & \vdots\\ \hline {}^{K1}\boldsymbol{\Psi}_a^{-1} & \cdots & {}^{KK}\boldsymbol{\Psi}_a^{-1}\end{array}\right]\left[\begin{array}{c|c|c} \mathbf{E}_{jp} & & \\ \hline & \mathbf{E}_{jp} & \\ \hline & & \mathbf{E}_{jp}\end{array}\right]\left[\begin{array}{c|c|c} {}^{11}\mathbf{R} & \cdots & {}^{1K}\mathbf{R}\\ \hline \vdots & \ddots & \vdots\\ \hline {}^{K1}\mathbf{R} & \cdots & {}^{KK}\mathbf{R}\end{array}\right]\right)\\
&=\mathrm{trace}\left(\left[\begin{array}{c|c|c} {}^{11}\boldsymbol{\Psi}_a^{-1} & \cdots & {}^{1K}\boldsymbol{\Psi}_a^{-1}\\ \hline \vdots & \ddots & \vdots\\ \hline {}^{K1}\boldsymbol{\Psi}_a^{-1} & \cdots & {}^{KK}\boldsymbol{\Psi}_a^{-1}\end{array}\right]\left[\begin{array}{c|c|c} \mathbf{E}_{jp} & & \\ \hline & \mathbf{E}_{jp} & \\ \hline & & \mathbf{E}_{jp}\end{array}\right]\left[\begin{array}{c|c|c} {}^{11}\mathbf{L} & \cdots & {}^{1K}\mathbf{L}\\ \hline \vdots & \ddots & \vdots\\ \hline {}^{K1}\mathbf{L} & \cdots & {}^{KK}\mathbf{L}\end{array}\right](\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}})\right)
\end{aligned} \tag{A19}$$

$$\begin{aligned}
&\mathrm{trace}\left(\sum_{m=1}^{K}\sum_{n=1}^{K}{}^{mn}\boldsymbol{\Psi}_a^{-1}\mathbf{E}_{jp}{}^{nm}\mathbf{R}\right) = \mathrm{trace}\left(\left(\sum_{n=1}^{K}\sum_{m=1}^{K}{}^{mn}\boldsymbol{\Psi}_a^{-1}\mathbf{E}_{jp}{}^{nm}\mathbf{L}\right)\mathbf{B}^{\mathrm{T}}\right)\\
&\mathrm{trace}\left(\sum_{m=1}^{K}\sum_{n=1}^{K}\mathbf{E}_{jp}^{nm}\mathbf{R}^{mn}\boldsymbol{\Psi}_a^{-1}\right) = \mathrm{trace}\left(\mathbf{B}^{\mathrm{T}}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}{}^{mn}\boldsymbol{\Psi}_a^{-1}\mathbf{E}_{jp}{}^{nm}\mathbf{L}\right)\right)\\
&\mathrm{vec}(\mathbf{E}_{pj})^{\mathrm{T}}\mathrm{vec}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}{}^{nm}\mathbf{R}^{mn}\boldsymbol{\Psi}_a^{-\mathrm{T}}\right)\\
&\ = \mathrm{vec}(\mathbf{B})^{\mathrm{T}}\mathrm{vec}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}{}^{mn}\boldsymbol{\Psi}_a^{-1}\mathbf{E}_{jp}^{nm}\mathbf{L}\right)\\
&\mathrm{vec}(\mathbf{E}_{pj})^{\mathrm{T}}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}({}^{mn}\boldsymbol{\Psi}_a^{-\mathrm{T}}\otimes{}^{nm}\mathbf{R})\right)\mathrm{vec}(\mathbf{I}_J)\\
&\ = \mathrm{vec}(\mathbf{B})^{\mathrm{T}}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}({}^{nm}\mathbf{L}^{\mathrm{T}}\otimes{}^{mn}\boldsymbol{\Psi}_a^{-1})\right)\mathrm{vec}(\mathbf{E}_{jp})\\
&\mathrm{vec}(\mathbf{E}_{pj})^{\mathrm{T}}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}({}^{mn}\boldsymbol{\Psi}_a^{-\mathrm{T}}\otimes{}^{nm}\mathbf{R})\right)\mathrm{vec}(\mathbf{I}_J)\\
&\ = \mathrm{vec}(\mathbf{E}_{jp})^{\mathrm{T}}\left(\sum_{n=1}^{K}\sum_{m=1}^{K}({}^{nm}\mathbf{L}\otimes{}^{mn}\boldsymbol{\Psi}_a^{-\mathrm{T}})\right)\mathrm{vec}(\mathbf{B})
\end{aligned} \tag{A20}$$

Equation (A20) is one of the *JP* equations necessary to estimate the loadings of **B**. The rest of the equations are obtained as the right and left parts of this equation are multiplied by the different vectorized $\mathbf{E}_{pj}$ and $\mathbf{E}_{jp}$ matrices, respectively. Since these terms are completely independent on both sides of the equation, the process can be carried out in a straightforward manner using matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ formed as $[\mathrm{vec}(\mathbf{E}_{11})\ \mathrm{vec}(\mathbf{E}_{21})\ldots\mathrm{vec}(\mathbf{E}_{PJ})]$ and $[\mathrm{vec}(\mathbf{E}_{11})\ \mathrm{vec}(\mathbf{E}_{21})\ldots\mathrm{vec}(\mathbf{E}_{JP})]$, respectively. A closer look at these matrices shows that $\mathbf{E}_1$ is the identity matrix of order *JP* while $\mathbf{E}_2$ is equal to the commutation matrix $\mathbf{K}_{PJ}$. When the equation is rearranged to estimate the loading of B, Equation (A 21) is obtained:

$$\mathrm{vec}(\mathbf{B}^{\mathrm{T}}) = \left(\sum_{m=1}^{K}\sum_{n=1}^{K}\left(\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes \mathbf{L}_{mn}\right)\right)^{-1} \times \left(\sum_{m=1}^{K}\sum_{n=1}^{K}\left(\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes \mathbf{R}_{mn}\right)\right)\mathrm{vec}(\mathbf{I}_J) \tag{A21}$$

Case 1C: Mode C

This scenario is well represented by the following objective function:

$$f = \mathrm{trace}\left[\left(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}}\right)\boldsymbol{\Psi}_a^{-1}\left(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}}\right)^{\mathrm{T}}\right] \tag{A22}$$

In order to make the equation more tractable the following modifications were applied: $\mathbf{M} = (\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})^{\mathrm{T}}$ and $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{I}_a$ to yield:

$$f = \mathrm{trace}(\mathbf{M}^{\mathrm{T}}\boldsymbol{\Psi}_a^{-1}\mathbf{M}) \tag{A23}$$

It is worth noting two important manipulations carried out in Equation (A25). First, the commutation matrices are introduced in order to invert the order of the Kronecker products $(\mathbf{E}_{kp}\otimes\mathbf{I}_J)$ and $(\mathbf{C}\otimes\mathbf{I}_J)$. Second, due to the sparse nature of $\tilde{\mathbf{A}}$, the following equality holds:

$$\mathbf{K}_{PP}\tilde{\mathbf{A}}^{\mathrm{T}} = \tilde{\mathbf{A}}^{\mathrm{T}} \tag{A26}$$

Equation (A25) becomes Equation (A31) using the matrices **T**, **S**, **L**, and **R** as defined in Equations (A27) to (A30):

$$\mathbf{L} = \boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{A27}$$

$$\mathbf{R} = (\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a \tag{A28}$$

$$\mathbf{S} = (\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}} \tag{A29}$$

$$\mathbf{T} = (\mathbf{B}^{\mathrm{T}}\otimes\mathbf{I}_K)\mathbf{K}_{JK}\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{A30}$$

$$\mathrm{trace}\left(\mathbf{L}(\mathbf{I}_J\otimes\mathbf{E}_{kp})\mathbf{R}\right) = \mathrm{trace}\left((\mathbf{I}_J\otimes\mathbf{C})\mathbf{S}(\mathbf{I}_P\otimes\mathbf{E}_{pk})\mathbf{T}\right) \tag{A31}$$

Equation (A31) can be expressed as Equation (A33) when the matrices forming both members of the previous equation are partioned as shown in Equation (A32). Matrices ${}^{mn}\mathbf{L}$, ${}^{nm}\mathbf{R}$, ${}^{mn}\mathbf{S}$, and ${}^{nm}\mathbf{T}$ have dimensions $K\times K$; $P\times K$; $P\times P$ and $K\times K$ respectively.

$$\begin{aligned}
&\mathrm{trace}\left(\left[\begin{array}{c:c:c} {}^{11}\mathbf{L} & \cdots & {}^{1J}\mathbf{L} \\ \hdashline \vdots & \ddots & \vdots \\ \hdashline {}^{J1}\mathbf{L} & \cdots & {}^{JJ}\mathbf{L} \end{array}\right]\left[\begin{array}{c:c:c} \mathbf{E}_{kp} & & \\ \hdashline & \mathbf{E}_{kp} & \\ \hdashline & & \mathbf{E}_{kp} \end{array}\right]\left[\begin{array}{c:c:c} {}^{11}\mathbf{R} & \cdots & {}^{1J}\mathbf{R} \\ \hdashline \vdots & \ddots & \vdots \\ \hdashline {}^{J1}\mathbf{R} & \cdots & {}^{JJ}\mathbf{R} \end{array}\right]\right) \\
&= \mathrm{trace}\left((\mathbf{I}_J\otimes\mathbf{C})\left[\begin{array}{c:c:c} {}^{11}\mathbf{S} & \cdots & {}^{1P}\mathbf{S} \\ \hdashline \vdots & \ddots & \vdots \\ \hdashline {}^{J1}\mathbf{S} & \cdots & {}^{JP}\mathbf{S} \end{array}\right]\left[\begin{array}{c:c:c} \mathbf{E}_{pk} & & \\ \hdashline & \mathbf{E}_{pk} & \\ \hdashline & & \mathbf{E}_{pk} \end{array}\right]\left[\begin{array}{c:c:c} {}^{11}\mathbf{T} & \cdots & {}^{1J}\mathbf{T} \\ \hdashline \vdots & \ddots & \vdots \\ \hdashline {}^{P1}\mathbf{T} & \cdots & {}^{PJ}\mathbf{T} \end{array}\right]\right)
\end{aligned} \tag{A32}$$

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{C}_{kp}} &= \frac{\partial f}{\partial \mathbf{M}}\left(\frac{\partial \mathbf{M}}{\partial \mathbf{C}_{kp}}\right)^{\mathrm{T}} = \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}\frac{\partial \mathbf{M}^{\mathrm{T}}}{\partial \mathbf{C}_{kp}}\right) \\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}\frac{\partial\left[(\mathbf{X}_a - \tilde{\mathbf{A}}(\mathbf{C}\otimes\mathbf{B})^{\mathrm{T}})\right]}{\partial \mathbf{C}_{kp}}\right) \\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}\mathbf{M}(-\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}})\right) \\
&= \mathrm{trace}\left(2\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a^{\mathrm{T}} - (\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}})(-\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}})\right) \\
&= -2\cdot\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{X}_a^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}}\right) \\
&\quad + 2\cdot\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}}\right)
\end{aligned} \tag{A 24}$$

Setting this derivative equal to zero to find the minimum leads to:

$$\begin{aligned}
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{X}_a^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}}\right) = \mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{kp}\otimes\mathbf{B})^{\mathrm{T}}\right) \\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{E}_{kp}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \mathrm{trace}\left((\mathbf{C}\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{pk}\otimes\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}\right) \\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}(\mathbf{E}_{kp}\otimes\mathbf{I}_J)(\mathbf{I}_P\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \\
&\mathrm{trace}\left((\mathbf{C}\otimes\mathbf{I}_J)(\mathbf{I}_P\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{E}_{pk}\otimes\mathbf{I}_P)(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}\right) \\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}(\mathbf{I}_J\otimes\mathbf{E}_{kp})\mathbf{K}_{JP}(\mathbf{I}_P\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \\
&\mathrm{trace}((\mathbf{I}_J\otimes\mathbf{C})\mathbf{K}_{JP}(\mathbf{I}_P\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}\mathbf{K}_{PP}(\mathbf{I}_P\otimes\mathbf{E}_{pk})\mathbf{K}_{PK}(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}) \\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}(\mathbf{I}_J\otimes\mathbf{E}_{kp})\mathbf{K}_{JP}(\mathbf{I}_P\otimes\mathbf{B})\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \\
&\mathrm{trace}\left((\mathbf{I}_J\otimes\mathbf{C})(\mathbf{B}\otimes\mathbf{I}_P)\mathbf{K}_{PP}\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}\mathbf{K}_{PP}(\mathbf{I}_P\otimes\mathbf{E}_{pk})\mathbf{K}_{PK}(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}\right) \\
&\mathrm{trace}\left(\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}(\mathbf{I}_J\otimes\mathbf{E}_{kp})(\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{X}_a\right) = \\
&\mathrm{trace}\left((\mathbf{I}_J\otimes\mathbf{C})(\mathbf{B}\otimes\mathbf{I}_P)\tilde{\mathbf{A}}^{\mathrm{T}}\tilde{\mathbf{A}}(\mathbf{I}_P\otimes\mathbf{E}_{pk})\mathbf{K}_{PK}(\mathbf{I}_K\otimes\mathbf{B}^{\mathrm{T}})\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ}\right)
\end{aligned} \tag{A 25}$$

$$\begin{aligned}
&\mathrm{trace}\left(\sum_{m=1}^{J}\sum_{n=1}^{J}{}^{mn}\mathbf{L}\mathbf{E}_{kp}{}^{nm}\mathbf{R}\right) = \mathrm{trace}\left(\mathbf{C}\sum_{n=1}^{J}\sum_{m=1}^{P}{}^{mn}\mathbf{S}\mathbf{E}_{pk}{}^{nm}\mathbf{T}\right) \\
&\sum_{m=1}^{J}\sum_{n=1}^{J}\mathrm{trace}\left(\mathbf{E}_{kp}{}^{nm}\mathbf{R}^{mn}\mathbf{L}\right) = \mathrm{vec}(\mathbf{C}^{\mathrm{T}})^{\mathrm{T}}\mathrm{vec}\left(\sum_{n=1}^{J}\sum_{m=1}^{P}{}^{mn}\mathbf{S}\mathbf{E}_{pk}^{nm}\mathbf{T}\right) \\
&\mathrm{vec}(\mathbf{E}_{pk})^{\mathrm{T}}\left(\sum_{m=1}^{J}\sum_{n=1}^{J}\left({}^{mn}\mathbf{L}^{\mathrm{T}}\otimes{}^{nm}\mathbf{R}\right)\right)\mathrm{vec}(\mathbf{I}_K) \\
&= \mathrm{vec}(\mathbf{E}_{pk})^{\mathrm{T}}\left(\sum_{n=1}^{J}\sum_{m=1}^{P}\left({}^{nm}\mathbf{T}\otimes{}^{mn}\mathbf{S}^{\mathrm{T}}\right)\right)\mathrm{vec}(\mathbf{C}^{\mathrm{T}})
\end{aligned} \tag{A33}$$

Equation (A33) is one of the *KP* equations necessary to estimate the loadings of **C**. The rest of the equations are obtained as the right and left parts of this equation are multiplied by the vectorized $\mathbf{E}_{pk}$ matrix. Since these terms are completely independent on both sides of the equation, the process can be carried out in a straightforward manner using matrices $\mathbf{E}_1$ formed as $[\mathrm{vec}(\mathbf{E}_{11})\ \mathrm{vec}(\mathbf{E}_{21})\ldots\mathrm{vec}(\mathbf{E}_{KP})]$. A closer look at these matrices shows that $\mathbf{E}_1$ is the identity matrix of order *PK*, providing Equation (A 34) to estimate the loadings of C:

$$\mathrm{vec}(\mathbf{C}^{\mathrm{T}}) = \left(\sum_{m=1}^{P}\sum_{n=1}^{J}\left(\mathbf{T}_{mn}\otimes\mathbf{S}_{nm}^{\mathrm{T}}\right)\right)^{-1} \times \left(\sum_{m=1}^{J}\sum_{n=1}^{J}\left(\mathbf{L}_{mn}^{\mathrm{T}}\otimes\mathbf{R}_{nm}\right)\right)\mathrm{vec}(\mathbf{I}_K) \tag{A34}$$

Case 1D

This scenario is structurally similar to the previous case, but more complex since the error covariance matrix changes from row to row. Therefore, the estimation process cannot be carried out in one step, but rather as a sequence of I independent problems solved by minimizing Equation (A 35):

$$f = \sum_{i=1}^{I} \left({}^{i}\mathbf{x}_a - {}^{i}\tilde{\mathbf{a}}(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}}\right) {}^{i}\boldsymbol{\Psi}_a^{-1} \left({}^{i}\mathbf{x}_a - {}^{i}\tilde{\mathbf{a}}(\mathbf{C} \otimes \mathbf{B})^{\mathrm{T}}\right)^{\mathrm{T}} \tag{A35}$$

As mentioned before, the only difference in the minimization process between Equation (A35) and Equation (A12) is the sequential manner in which the former is solved. This situation leads to estimation equations that are similar to the previous case, but solved in a sequential manner. Mathematically, this is carried out by solving row by row in mode A and solving over a sequence of I summations for mode B and C as shown next:

$${}^{i}\mathbf{a} = {}^{i}\mathbf{x}_a {}^{i}\boldsymbol{\Psi}_a^{-1} \mathbf{Z}_a^{\mathrm{T}} \left(\mathbf{Z}_a {}^{i}\boldsymbol{\Psi}_a^{-1} \mathbf{Z}_a^{\mathrm{T}}\right)^{-1} \tag{A36}$$

$$\begin{aligned} \mathrm{vec}(\mathbf{B}^{\mathrm{T}}) = & \left(\sum_{i=1}^{I} \sum_{m=1}^{K} \sum_{n=1}^{K} \left({}^{i}\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes {}^{i}\mathbf{L}_{mn}\right) \right)^{-1} \\ & \times \left(\sum_{i=1}^{I} \sum_{m=1}^{K} \sum_{n=1}^{K} \left({}^{i}\boldsymbol{\Psi}_{nm}^{-\mathrm{T}} \otimes {}^{i}\mathbf{R}_{mn}\right) \right) \mathrm{vec}(\mathbf{I}_J) \end{aligned} \tag{A37}$$

where

$${}^{i}\mathbf{R} = (\mathbf{C} \otimes \mathbf{I}_P)^{i}\tilde{\mathbf{a}}^{\mathrm{T}i}\mathbf{x}_a \tag{A38}$$

$$\mathbf{L} = (\mathbf{C} \otimes \mathbf{I}_P)^{i}\tilde{\mathbf{a}}^{\mathrm{T}i}\tilde{\mathbf{a}}(\mathbf{C} \otimes \mathbf{I}_p)^{\mathrm{T}} \tag{A39}$$

and

$$\begin{aligned} \mathrm{vec}(\mathbf{C}^{\mathrm{T}}) = & \left(\sum_{i=1}^{I} \sum_{m=1}^{P} \sum_{n=1}^{J} \left({}^{i}\mathbf{T}_{mn} \otimes {}^{i}\mathbf{S}_{nm}^{\mathrm{T}}\right) \right)^{-1} \\ & \times \left(\sum_{i=1}^{I} \sum_{m=1}^{J} \sum_{n=1}^{J} \left({}^{i}\mathbf{L}_{mn}^{\mathrm{T}} \otimes {}^{i}\mathbf{R}_{nm}\right) \right) \mathrm{vec}(\mathbf{I}_K) \end{aligned} \tag{A40}$$

$${}^{i}\mathbf{L} = {}^{i}\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{A41}$$

$${}^{i}\mathbf{R} = (\mathbf{B} \otimes \mathbf{I}_P)^{i}\tilde{\mathbf{a}}^{\mathrm{T}i}\mathbf{x}_a \tag{A42}$$

$${}^{i}\mathbf{S} = (\mathbf{B} \otimes \mathbf{I}_P)^{i}\tilde{\mathbf{a}}^{\mathrm{T}i}\tilde{\mathbf{a}} \tag{A43}$$

$${}^{i}\mathbf{T} = (\mathbf{B}^{\mathrm{T}} \otimes \mathbf{I}_K)\mathbf{K}_{JK}{}^{i}\boldsymbol{\Psi}_a^{-1}\mathbf{K}_{KJ} \tag{A44}$$