JOURNAL OF CHEMOMETRICS
J. Chemometrics 2003; **17**: 237–253
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cem.789

Maximum likelihood parallel factor analysis (MLPARAFAC)

Lorenzo Vega-Montoto and Peter D. Wentzell*
Trace Analysis Research Centre, Department of Chemistry, Dalhousie University, Halifax, Nova Scotia, Canada B3H 4J3

Received 16 July 2002; Revised 16 January 2003; Accepted 29 January 2003

Algorithms for carrying out maximum likelihood parallel factor analysis (MLPARAFAC) for three-way data are described. These algorithms are based on the principle of alternating least squares, but differ from conventional PARAFAC algorithms in that they incorporate measurement error information into the trilinear decomposition. This information is represented in the form of an error covariance matrix. Four algorithms are discussed for dealing with different error structures in the three-way array. The simplest of these treats measurements with non-uniform measurement noise which is uncorrelated. The most general algorithm can analyze data with any type of noise correlation structure. The other two algorithms are simplifications of the general algorithm which can be applied with greater efficiency to cases where the noise is correlated only along one mode of the three-way array. Simulation studies carried out under a variety of measurement error conditions were used for statistical validation of the maximum likelihood properties of the algorithms. The MLPARAFAC methods are also shown to produce more accurate results than PARAFAC under a variety of conditions. Copyright © 2003 John Wiley & Sons, Ltd.

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

1. INTRODUCTION

With advancing technology of analytical instrumentation, data in the form of tensors of second order and higher has become more commonplace. Examples of such techniques include fluorescence excitation–emission spectroscopy and chromatography with multichannel detectors. In 1980, Hirschfeld [1] provided a very complete table of all the feasible combinations of techniques capable of providing second-order data at that time and estimated that about 60% of the techniques are bilinear under certain conditions. Extension to trilinear data is easily accomplished when several samples are analyzed by these methods. This list has continued to expand in terms of the number of techniques and possible analytical orders as this instrumentation becomes commonplace in chemistry laboratories. Ever since Appellof and Davison [2] provided the first application of trilinear decomposition to chemistry using both simulated and real LC/emission/excitation fluorescence data, the number of applications have expanded to many branches of chemistry, ranging from basic research to environmental and food chemistry.

Trilinear data (and multilinear tensors in general) share common properties with bilinear data that make the latter structure central to modern chemometrics. Both types of data can model deterministic relationships among variables, especially in cases where a high degree of collinearity exists. These types of models allow multivariate and multiorder data to 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 'true' rank 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. However, contrary to what happens in bilinear models, where the smaller set of variables are abstract solutions of the underlying physical factors which are not unique due to rotational ambiguities, the trilinear and higher multilinear models can produce unique and well-identified solutions (up to trivial differences in factor order and relative scaling across modes) [3]. In addition, the uniqueness of the solution gives rise to the 'second-order advantage' which allows the quantitation of an analyte in the presence of interferences with only one calibration sample.

A variety of algorithms have been developed to estimate

*Correspondence to: P. D. Wentzell, Trace Analysis Research Centre, Department of Chemistry, Dalhousie University, Halifax, Nova Scotia, Canada B3H 4J3.
E-mail: peter.wentzell@dal.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council (NSERC) of Canada.
Contract/grant sponsor: Dow Chemical Company.

the multilinear model, including parallel factor analysis [4] (PARAFAC), direct trilinear decomposition [5] (DTLD) and positive matrix factorization [6] (PMF3). These algorithms are based on different numerical approaches, namely alternating least squares (ALS), eigenproblem formulation and a Gauss–Newton approach, respectively. Each has its own advantages and disadvantages that make it suitable in a specific situation. However, PARAFAC (ALS) is currently the most widely used algorithm, mainly due to its good convergence properties. ALS, which was introduced by Yates [7] in 1933, works by simply dividing the parameters into several sets. Each set of parameters is estimated in a least-squares sense conditionally on the remaining parameters. The estimation of the parameters is repeated iteratively until a certain stop criterion is reached. In this way, a very complex non-linear problem becomes a sequence of simpler least-squares steps in which the parameter sets are improved in each step. As all estimates of parameters are least-squares estimates, the procedure can only improve the fit or keep it the same if converged. It follows from this that the objective function decreases monotonically and, since it is also bounded from below (the objective function cannot be less than zero), convergence is always reached. This does not imply that the global minimum is guaranteed, since a problem like this is characterized by several local minima. Global convergence can be assessed when repetitions using different starting points yield similar sets of parameters. In addition to the reliable convergence characteristics of ALS, it is also used because it yields maximum likelihood estimation under certain noise characteristics.

Methods such as PARAFAC give maximum likelihood estimates of the model parameters when the noise is independently and identically distributed with a normal distribution (*iid* normal). Noise can be broadly defined as an undesirable variation in a measured signal which obscures the measurement of interest, the true signal. Based on the specific advantages of multilinear data, this definition will be narrowed to undesirable variation attributable to non-chemical sources (e.g. instrumental sources). Noise can have many different origins, having a very complex range of properties and characteristics. Unfortunately, these properties and characteristics are not mutually exclusive making the number of possibilities of noise structures very large. The term *iid* has been coined in the chemometric literature to make a precise and concise description of the fundamental properties needed to characterize the instrumental noise in the 'ideal' case. It conveys information about independence (i.e. the error observed at any one channel is uncorrelated with the error observed at any other channel) and the homogeneity of distributions (i.e. identically distributed implies the error variance and distribution are the same for all measurements). Conventional least-squares approaches to trilinear decomposition are maximum likelihood methods only under *iid* conditions. These naive assumptions about the noise structure corrupting the multilinear data can lead to poor models, since all of the methods rely on a least-squares procedure. PARAFAC and DTLD are the most affected since both independence and homoscedasticity (identical distributions) need to be satisfied to yield the maximum likelihood solution. PMF3 can overcome the need for homoscedastic noise to yield the maximum likelihood solution because it applies a weighting scheme that solves this impediment. When minor variations from the assumption of *iid* normal noise are observed, some scaling techniques can be used with PARAFAC in order to alleviate the deviations from the *iid* condition, but this will only yield a maximum likelihood solution when the noise is uncorrelated and the heteroscedasticity follows a certain structure. A more general approach to tackle this problem, W-PARAFAC, was introduced in 1997 by Kiers [8], who used a weighted objective function to remedy the problem of heteroscedastic noise. The algorithm is based on a majorization procedure instead of an ALS algorithm. W-PARAFAC and PMF3 both overcome the heteroscedasticity of the noise using a weighted objective function, but the issue of the noise correlation is still a problem for both methods, since they cannot accommodate error covariance terms in the procedure.

The presence of covariance among measurement errors is a ubiquitous and pernicious effect produced by several sources ranging from the temporal correlation of pump noise in chromatography to the spatial correlation of array detectors in spectroscopy. Another important source of correlation in the measurement errors is signal processing, particularly electronic or digital smoothing filters. Because of all of these effects, correlated measurement errors are likely to be the rule rather than the exception for multivariate data sets, implying that standard methods of analysis (both two-way and multi-way) that make assumptions of *iid* normal noise are suboptimal. The only optimal means to account for the correlation in measurement errors is using a maximum likelihood approach to estimate model parameters that are most likely to give rise to the observed measurements. For bilinear data, this problem has been addressed through the development of maximum likelihood principal component analysis (MLPCA) [9], which has been shown to provide improved results where the effects of noise correlation are significant.

Correlation among measurement errors in three-way data is complicated by the unfolding/matricization process usually used in ALS algorithms. Elements with correlated measurement errors which may appear adjacent to one another in a 'slice' of the three-way array may become spatially separated from one another when the cube is unfolded in certain ways. Because of this, conceptualization and simplification of error covariance structures for three-way data are more difficult, and this has impeded the development of maximum likelihood methods for three-way data. Until recently, this problem was avoided by the standard estimation algorithms. Recently, a method called MILES [10], which is based on a majorization-ALS algorithm, was introduced to address the problem of correlated measurement errors for multilinear data. The extent to which this method yields maximum likelihood estimates is unclear since no validation of the results was done in this context and the theoretical foundation of the method is obscured by the complexity of the algorithm.

This paper introduces the theoretical foundations for maximum likelihood parallel factor analysis (MLPARA-

Table I. Standard MLPARAFAC algorithm (uncorrelated errors).

1. Given an $I \times J \times K$ cube of data $\underline{\mathbf{X}}$ and a corresponding $I \times J \times K$ cube $\underline{\boldsymbol{\Sigma}}$ of measurement error variances, the algorithm is initialized using random values of the correct dimensions or using estimates obtained by direct trilinear decomposition (TLD).

$$[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] = tld(\underline{\mathbf{X}}, P) \tag{T1}$$

2. Unfold $\underline{\mathbf{X}}$ and $\underline{\boldsymbol{\Sigma}}$ retaining the first order and calculate the maximum likelihood estimation of $\hat{\mathbf{A}}$ conditional on $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

$$\mathbf{X}_a = unfold(\underline{\mathbf{X}}, a); \boldsymbol{\Sigma}_a = unfold(\underline{\boldsymbol{\Sigma}}, a); {}^i\boldsymbol{\Psi}_a = diag({}^i\boldsymbol{\Sigma}_a) \tag{T2}$$

$${}^i\hat{\mathbf{a}}^{\mathrm{T}} = {}^i\mathbf{x}_a {}^i\boldsymbol{\Psi}_a^{-1}\hat{\mathbf{Z}}_a^{\mathrm{T}}(\hat{\mathbf{Z}}_a {}^i\boldsymbol{\Psi}_a^{-1}\hat{\mathbf{Z}}_a^{\mathrm{T}})^{-1} \tag{T3}$$

Here ${}^i\hat{\mathbf{a}}^{\mathrm{T}}$ is a row vector of $\hat{\mathbf{A}}$. Using this estimate and the estimates of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ the objective function can be calculated using Equation (T4).

$$S_a^2 = \sum_{i=1}^{I}({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a){}^i\boldsymbol{\Psi}_a^{-1}({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a)^{\mathrm{T}} \tag{T4}$$

3. Unfold $\underline{\mathbf{X}}$ and $\underline{\boldsymbol{\Sigma}}$ retaining the second order and calculate the maximum likelihood estimation of $\hat{\mathbf{B}}$ conditional on $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$.

$$\mathbf{X}_b = unfold(\underline{\mathbf{X}}, b); \boldsymbol{\Sigma}_b = unfold(\underline{\boldsymbol{\Sigma}}, b); {}^j\boldsymbol{\Psi}_b = diag({}^j\boldsymbol{\Sigma}_b) \tag{T5}$$

$${}^j\hat{\mathbf{b}}^{\mathrm{T}} = {}^j\mathbf{x}_b {}^j\boldsymbol{\Psi}_b^{-1}\hat{\mathbf{Z}}_b^{\mathrm{T}}(\hat{\mathbf{Z}}_b {}^j\boldsymbol{\Psi}_b^{-1}\hat{\mathbf{Z}}_b^{\mathrm{T}})^{-1} \tag{T6}$$

Here ${}^j\hat{\mathbf{b}}^{\mathrm{T}}$ is a row vector of $\hat{\mathbf{B}}$. Using this estimate and the estimates of $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$ the objective function can be calculated using Equation (T7).

$$S_b^2 = \sum_{j=1}^{J}({}^j\mathbf{x}_b - {}^j\hat{\mathbf{x}}_b){}^j\boldsymbol{\Psi}_b^{-1}({}^j\mathbf{x}_b - {}^j\hat{\mathbf{x}}_a)^{\mathrm{T}} \tag{T7}$$

4. Unfold $\underline{\mathbf{X}}$ and $\underline{\boldsymbol{\Sigma}}$ retaining the third order and calculate the maximum likelihood estimation of $\hat{\mathbf{C}}$ conditional on $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

$$\mathbf{X}_c = unfold(\underline{\mathbf{X}}, c); \boldsymbol{\Sigma}_c = unfold(\underline{\boldsymbol{\Sigma}}, c); {}^k\boldsymbol{\Psi}_c = diag({}^k\boldsymbol{\Sigma}_c) \tag{T8}$$

$${}^k\hat{\mathbf{c}}^{\mathrm{T}} = {}^k\mathbf{x}_c {}^k\boldsymbol{\Psi}_c^{-1}\hat{\mathbf{Z}}_c^{\mathrm{T}}(\hat{\mathbf{Z}}_c {}^k\boldsymbol{\Psi}_c^{-1}\hat{\mathbf{Z}}_c^{\mathrm{T}})^{-1} \tag{T9}$$

Here ${}^k\hat{\mathbf{c}}^{\mathrm{T}}$ is a row vector of $\hat{\mathbf{C}}$. Using this estimate and the estimates of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ the objective function can be calculated using Equation (T10).

$$S_c^2 = \sum_{k=1}^{K}({}^k\mathbf{x}_c - {}^k\hat{\mathbf{x}}_c){}^k\boldsymbol{\Psi}_c^{-1}({}^k\mathbf{x}_c - {}^k\hat{\mathbf{x}}_c)^{\mathrm{T}} \tag{T10}$$

5. Calculate the convergence parameters λ_1 and λ_2.

$$\lambda_1 = (S_b^2 - S_a^2)/S_a^2; \lambda_2 = (S_c^2 - S_a^2)/S_a^2 \tag{T11}$$

If λ_1 and λ_2 are less than the convergence limit (typically 10^{-8} in this work), terminate. Otherwise return to step 2.

FAC). MLPARAFAC is an errors-in-variables modeling method in that it accounts for measurement errors in the estimation of model parameters. It is an optimal modeling method in a maximum likelihood sense for functional models with no errors in the model equations. The present method is a natural extension to PARAFAC of the MLPCA method introduced by Wentzell *et al.* [9]. The mathematical aspects of the algorithm are described in detail to allow the principles to be readily applied. The algorithm can accommodate heteroscedastic and correlated noise in one or more dimensions and has excellent convergence characteristics because its core is based on an alternating least-squares procedure. Although all the cases used in this paper will be three-way data this algorithm is extensible to N-way data.

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 indistinctively, and also 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 $\mathbf{X}_{\mathrm{a}}$. In the same way, matrices $\mathbf{X}_{\mathrm{b}}$ $(J \times IK)$ and $\mathbf{X}_{\mathrm{c}}$ $(K \times IJ)$ 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 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.

2. THEORY

PARAFAC is an acronym used to refer to two different, although closely related, concepts. It is used to describe the model that the trilinear structure of the data follows, and it is also used to refer to one of the various algorithms used to estimate the parameters of the aforementioned model. PARAFAC was originally introduced by Harshman [4] and simultaneously and independently by Carroll and Chang [11], who referred to it as canonical decomposition (CANDECOMP). The model can be seen as an extension of bilinear PCA to higher orders. The PARAFAC model for a three-way array is defined by three loading matrices, **A, B** and **C**, with elements a_{ip}, b_{jp} and c_{kp}. It can be written as a tensor product:

$$\underline{\mathbf{X}} = \sum_{p=1}^{P} \mathbf{a}_p \otimes \mathbf{b}_p \otimes \mathbf{c}_p + \underline{\mathbf{E}} \tag{1}$$

where $\mathbf{a}_p$, $\mathbf{b}_p$ and $\mathbf{c}_p$ are the pth columns of the loading matrices **A, B** and **C**, respectively. The model can also be expressed in scalar form:

$$x_{ijk} = \sum_{p=1}^{P} a_{ip} b_{jp} c_{kp} + e_{ijk} \tag{2}$$

where x_{ijk} is an element of the three-way array $\underline{\mathbf{X}}$ and e_{ijk} is an element of the corresponding residual matrix, $\underline{\mathbf{E}}$, where the indices refer to modes A, B and C, respectively.

Since most of the mathematical/statistical tools and concepts used in chemometrics rely on the foundations of linear algebra, a matrix representation of a three-way array is very useful. The process of converting a cube or higher order arrangement of data into a matrix is called unfolding or matricization and it can be done in at least as many ways as the array has orders. Equation (3) represents the unfolded data when the first order is retained:

$$\mathbf{X}_a = \mathbf{A}\mathbf{Z}_a + \mathbf{E}_a \tag{3}$$

The $\mathbf{X}_a$ matrix is obtained from the matrix multiplication of loading matrix **A** and a matrix $\mathbf{Z}_a$ which is formed from loading matrices **B** and **C**. The $\mathbf{Z}_a$ matrix can be obtained as a Khatri–Rao product [12] of matrices **B** and **C** or as a Kronecker product [13] of matrices **B** and **C** premultiplied by the unfolded superdiagonal 'identity' matrix of order P ($\mathbf{I}_a$). These alternative representations are shown in Equations (4) and (5), respectively:

$$\mathbf{Z}_a = (\mathbf{C}^{\mathrm{T}} | \otimes | \mathbf{B}^{\mathrm{T}}) \tag{4}$$

$$\mathbf{Z}_a = \mathbf{I}_a (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{B}^{\mathrm{T}}) \tag{5}$$

Analogous equations can be used to represent $\underline{\mathbf{X}}$ as the matrices obtained when the second and third orders are retained ($\mathbf{X}_b$ and $\mathbf{X}_c$).

Assuming **B** and **C** are known (or estimated) and *iid* noise conditions, then an estimate of **A** can be obtained solving the conditional least-squares problem to minimize the sum of the squares of the residuals in $\underline{\mathbf{E}}$. The solution to this problem is given by the equation

$$\hat{\mathbf{A}} = \mathbf{X}_a \hat{\mathbf{Z}}_a^{\mathrm{T}} (\hat{\mathbf{Z}}_a \hat{\mathbf{Z}}_a^{\mathrm{T}})^{-1} \tag{6}$$

This least-squares estimate of **A** can in turn be used to obtain estimates of **B** and **C** (given $\hat{\mathbf{C}}$ and $\hat{\mathbf{B}}$, respectively) by employing similar equations involving $\mathbf{X}_b$ and $\mathbf{X}_c$. This leads naturally to the iterative ALS procedure which can be used to estimate all of the loadings in a stepwise procedure.

2.1. Non-uniform measurement errors

Unfortunately, in cases where *iid* noise conditions are violated, the conventional ALS algorithm will produce suboptimal estimates of the loadings. In those cases where measurement errors remain independent but the condition of homoscedasticity is violated (i.e. each measurement can have a different variance), a more general objective function can be minimized to satisfy the maximum likelihood criterion. Consider the three-way array of measurements $\underline{\mathbf{X}}$ and an associated array $\underline{\boldsymbol{\Sigma}}$, which contains the variances of the measurements of the corresponding elements in $\underline{\mathbf{X}}$. For a given trial solution $\hat{\underline{\mathbf{X}}}$ (based on estimates of $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ such that $\hat{\mathbf{X}} = \hat{\mathbf{A}}\hat{\mathbf{Z}}_a$), Equation (7) gives the likelihood function in terms of the matrices unfolded in the A mode:

$$L = \prod_{i=1}^{I} \frac{1}{(2\pi)^{jk/2} |{}^i\boldsymbol{\Psi}_a|^{1/2}} \exp\left[-\frac{1}{2}({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a)\,{}^i\boldsymbol{\Psi}_a^{-1}({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a)^{\mathrm{T}}\right] \tag{7}$$

where ${}^i\mathbf{x}_a$ represents the ith row of the unfolded matrix $\mathbf{X}_a$ and ${}^i\hat{\mathbf{x}}_a$ represent the corresponding vector of estimates of $\hat{\mathbf{X}}_a$. The matrix ${}^i\boldsymbol{\Psi}_a$ is the measurement error covariance matrix for the ith row of $\mathbf{X}_a$, which in the case of uncorrelated errors will be a diagonal matrix ($JK \times JK$) containing the variance of the measurement errors of ${}^i\mathbf{x}_a$; that is, it is the diagonalized form of the ith row of $\boldsymbol{\Sigma}_a$. The error covariance matrix is defined according to

$${}^i\boldsymbol{\Psi}_a = E[({}^i\mathbf{x}_a - {}^i\mathbf{x}_a^{\circ})^{\mathrm{T}} \cdot ({}^i\mathbf{x}_a - {}^i\mathbf{x}_a^{\circ})] \tag{8}$$

where E designates an expectation value and ${}^i\mathbf{x}_a^{\circ}$ represents ith row of $\mathbf{X}_a^{\circ}$, which is the true or expectation value of $\underline{\mathbf{X}}^{\circ}$ unfolded in the A mode. Since $\underline{\mathbf{X}}^{\circ}$ is not normally known, it is normally estimated on the basis of mean values, or else ${}^i\boldsymbol{\Psi}_a$ is estimated on the basis of prior information (e.g. an assumption of proportional errors).

Obtaining the maximum likelihood estimates of $\hat{\mathbf{A}}, \hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ means maximizing the likelihood function in Equation (7) with respect to these loading parameters. This is equivalent to minimizing the logarithm of the likelihood function, which, when constant terms are ignored, results in the objective function in Equation (9):

$$\begin{aligned} S^2 &= \sum_{i=1}^{I} ({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a)\,{}^i\boldsymbol{\Psi}_a^{-1}({}^i\mathbf{x}_a - {}^i\hat{\mathbf{x}}_a)^{\mathrm{T}} = \sum_{i=1}^{I} S_i^2 \\ &= \sum_{i=1}^{I} ({}^i\mathbf{x}_a\ {}^i\boldsymbol{\Psi}_a^{-1}\,{}^i\mathbf{x}_a^{\mathrm{T}} - {}^i\mathbf{x}_a\ {}^i\boldsymbol{\Psi}_a^{-1}\,{}^i\hat{\mathbf{x}}_a^{\mathrm{T}} - {}^i\hat{\mathbf{x}}_a\ {}^i\boldsymbol{\Psi}_a^{-1}\,{}^i\mathbf{x}_a^{\mathrm{T}} + {}^i\hat{\mathbf{x}}_a\ {}^i\boldsymbol{\Psi}_a^{-1}\,{}^i\hat{\mathbf{x}}_a^{\mathrm{T}}) \end{aligned} \tag{9}$$

To minimize the objective function, S^2, with respect to the loadings $\hat{\mathbf{A}}$ given $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$, we first recognize that each term, S_i^2, in the summation is an independent function of the ith row of $\hat{\mathbf{A}}$, designated as ${}^i\hat{\mathbf{a}}$, and the given matrix $\hat{\mathbf{X}}_a$, that is ${}^i\hat{\mathbf{x}}_a = {}^i\hat{\mathbf{a}}\hat{\mathbf{Z}}_a$. This means that S^2 can be minimized by minimizing the individual terms, allowing each row of **A** to be estimated independently as shown in Equations (10)–

(12).

$$S^2 = {}^i\mathbf{x}_a \, {}^i\boldsymbol{\Psi}_a^{-1\,i}\mathbf{x}_a^{\mathrm{T}} - {}^i\,\mathbf{x}_a \, {}^i\boldsymbol{\Psi}_a^{-1}(\hat{\mathbf{a}}\mathbf{Z}_a)^{\mathrm{T}} - \hat{\mathbf{a}}\mathbf{Z}_a \, {}^i\boldsymbol{\Psi}_a^{-1i}\mathbf{x}_a^{\mathrm{T}} + \hat{\mathbf{a}}\mathbf{Z}_a \, {}^i\boldsymbol{\Psi}_a^{-1}(\hat{\mathbf{a}}\mathbf{Z}_a)^{\mathrm{T}} \tag{10}$$

$$\frac{\partial S^2}{\partial \hat{\mathbf{a}}} = 0 -{}^i\,\mathbf{x}_a \, {}^i\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}} - \mathbf{Z}_a \, {}^i\boldsymbol{\Psi}_a^{-1\,i}\mathbf{x}_a^{\mathrm{T}} + 2\mathbf{Z}_a \, {}^i\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}\hat{\mathbf{a}}^{\mathrm{T}} \tag{11}$$

$$\hat{\mathbf{a}} ={}^i\,\mathbf{x}_a \, {}^i\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}(\mathbf{Z}_a \, {}^i\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}})^{-1} \tag{12}$$

It should be emphasized that, in these equations, ${}^i\hat{\mathbf{a}}$ is used to designate a row of $\hat{\mathbf{A}}$ and does not represent a loading vector of $\hat{\mathbf{A}}$. From Equation (12), estimates of ${}^i\hat{\mathbf{a}}$ can be combined to give $\hat{\mathbf{A}}$. In cases where the error covariance matrix is the same for all the rows of $\mathbf{X}_a$, Equation (12) can be generalized to the matrix form represented in Equation (13).

$$\hat{\mathbf{A}} = \mathbf{X}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}(\mathbf{Z}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}})^{-1} \tag{13}$$

This equation can also be reduced easily to equation 6 in cases where the noise is the same (homoscedastic) for all the channels.

Since the requirement for this development was independence of measurement errors, the error covariance matrices for all the orders are diagonals. Unfolding $\underline{\mathbf{X}}$ in the other two directions leads to similar equations for $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$, allowing an equivalent maximum likelihood estimation of $\underline{\hat{\mathbf{X}}}$ in all the spaces, subject to the constraint that two of the spaces remain fixed. This occurs because the objective function of $\underline{\mathbf{X}}$ unfolded in all the orders reduces to the same summation but in a different order. To obtain the unrestricted maximum likelihood estimation of $\underline{\hat{\mathbf{X}}}$, it is necessary to optimize the objective function with respect to all three sets of loading vectors. An alternative to such a direct optimization is an iterative approach using ALS.

The algorithm for the maximum likelihood PARAFAC in cases of heteroscedastic noise is given in Table I. The algorithm alternately uses the maximum likelihood estimates of two modes, say $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$, to update the estimates in the mode left out, say $\hat{\mathbf{A}}$. This procedure is carried out iteratively, using the previously estimated mode and one of the other two modes, say $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$, to estimate the other, say $\hat{\mathbf{B}}$. This procedure has been found to be simple, fast and reliable. Although global convergence is not guaranteed, it does not seem to be susceptible to local minima as is the case with gradient methods. In addition, this method is very attractive since its core is based on an ALS framework, which ensures an improvement of the solution in each step. The algorithm is easily applied in cases where there are missing values by incorporating large variances for the missing measurements. Convergence time depends on the dimensionality of the data, the degree of similarity of the components forming the system, the accuracy of the initial estimates and the structure of the errors. The two most important factors increasing the convergence time are the dimension of the model and the degree of similarity, especially the former, which makes each step longer and increases the necessity for more iterations. Some strategies have been reported to improve the efficiency of the algorithm [14], but these will not be incorporated here. Comparative data on convergence time will be reported in a future paper.

It is worth noting that the algorithm presented in Table I imposes restrictions on the presence of offsets in any mode. Normally, this would be equivalent to saying that the data have been properly mean centered [15], but in the case of non-uniform measurement errors, mean centering is not equivalent to eliminating offsets. The case of offsets will be treated from a more optimal, although incomplete perspective in Section 2.4.

Although the problem of heteroscedastic noise has been addressed in the literature using weighted PARAFAC algorithms, the description presented here marks the first time (to our knowledge) that a formal theoretical treatment of this problem from a maximum likelihood perspective has been given. It also represents a good starting point to generalize this algorithm to more complicated scenarios, such as systems affected by correlated noise and heteroscedastic and correlated noise in two or more dimensions.

2.2. Correlated measurement errors

The incorporation of uncorrelated, heteroscedastic measurement errors into the ALS framework as described in the preceding section is relatively straightforward. On the surface it may appear that extension to correlated measurement errors is a trivial matter, since the likelihood function expressed by Equation (7) should be equally applicable for error covariance matrices that are not diagonal. However, there is a critical difference that relates to the way in which the information in the error covariance matrices is transformed when the three-way array is unfolded. In the case of uncorrelated measurement errors, the diagonal error covariance matrices in each mode contain all of the information about the uncertainty in the measurements, although the order in which this information appears varies with the modes. In the case of correlated measurements, some of this information will be lost in one or more modes, making it impossible to maintain consistency in the ALS estimates obtained when using the same strategy as for independent errors.

To illustrate this point, consider the relatively simple case where the errors are correlated in one order only. For example, we may have a case where multiple samples of different composition are separated by chromatography with multichannel detection and there is significant correlation in the time domain due to pump noise. Alternatively, we could imagine fluorescence excitation–emission measurements for a series of samples, which are correlated in the emission domain due to source fluctuations, but uncorrelated in the excitation domain because it is scanned at longer time intervals as the second order. For convenience, we will say that the measurements along the rows which make up mode B are correlated, but there is no correlation among these rows in the three-way array. This situation is conceptually illustrated with a small 4 × 3 × 2 array in Figure 1. The elements of the array that are labeled with the same letters are considered to be correlated in this example. Considering unfolding in the A mode first, the figure shows the structure of the error covariance matrix for the first row of $\mathbf{X}_a$, which is block diagonal due to the presence of two sets of correlated measurement errors. The remaining three rows will have the same error covariance structure, resulting in 72 non-zero elements in total describing error covariance. On the other hand, the error covariance matrix for the first row of $\mathbf{X}_b$ has a diagonal form since the correlated measurements appear in the columns. Considering all three rows of $\mathbf{X}_b$, this results in only 24 non-zero elements describing the error structure. Information on the covariance has been lost in this representation. Finally, the error covariance matrix structure for the first row of $\mathbf{X}_c$ is band diagonal. The two error covariance matrices resulting from this unfolded matrix will have a total of 72 non-zero values describing the error covariance and contain the same information as the A mode, only in a different representation. However, because the error covariance matrices for $\mathbf{X}_b$ contain incomplete information, the sequence of steps in the ALS algorithm described in the previous section cannot be completed using this approach.

As correlation among the orders becomes more complex, the inability to represent this information becomes more obvious. This is clear if one realizes that a complete description of all correlations in the general case would require $(IJK)^2$ elements, but the total number of elements in the row covariance matrices for, say $\mathbf{X}_a$, is only $I(JK)^2$. In order to circumvent this problem a more general solution for correlated errors will be obtained redefining the problem and modeling the measurements as a single point in an IJK-dimensional space. To do this, $\underline{\mathbf{X}}$ (or alternatively any unfolded representation) is vectorized by applying the '*vec*' operator and the equations are adapted as necessary. The generalization of Equations (12) and (10) are

$$vec(\hat{\mathbf{A}}^{\mathrm{T}}) = (\mathbf{V}_a^{\mathrm{T}}\boldsymbol{\Omega}_a^{-1}\mathbf{V}_a)^{-1}\mathbf{V}_a^{\mathrm{T}}\boldsymbol{\Omega}_a^{-1}vec(\mathbf{X}_a^{\mathrm{T}}) \qquad (14)$$

$$S^2 = vec(\Delta\mathbf{X}_a^{\mathrm{T}})^{\mathrm{T}}\boldsymbol{\Omega}_a^{-1}vec(\Delta\mathbf{X}_a^{\mathrm{T}}) \qquad (15)$$

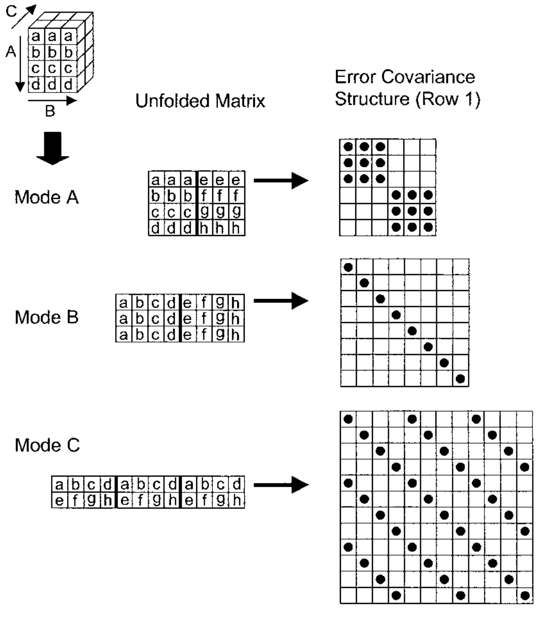

Figure 1. Illustration of the unfolding of a three-way array and its effect on the structure of the error covariance matrix for the first row of the unfolded matrix. Elements with correlated measurement errors are labeled with the same letter.

where

$$\mathbf{V}_a = \mathbf{I}_I \otimes \hat{\mathbf{Z}}_a^{\mathrm{T}} \qquad (16)$$

$$\boldsymbol{\Omega}_a = E[(vec((\mathbf{X}_a - \mathbf{X}_a^{\circ})^{\mathrm{T}})) \cdot (vec((\mathbf{X}_a - \mathbf{X}_a^{\circ})^{\mathrm{T}}))^{\mathrm{T}}] \qquad (17)$$

$$\Delta\mathbf{X}_a = (\mathbf{X}_a - \hat{\mathbf{X}}_a) \qquad (18)$$

In these equations, the '*vec*' operator reshapes a matrix into a column vector by taking the elements in sequence columnwise [13]. The symbol $\otimes$ as used here identifies the Kronecker product such that each element of $\mathbf{I}_I$ is multiplied by $\hat{\mathbf{X}}_a^{\mathrm{T}}$ therefore $\mathbf{V}_a$ is an $IP \times IJK$ matrix with $\mathbf{Z}_a^{\mathrm{T}}$ repeating along the diagonal. The matrix $\boldsymbol{\Omega}_a$ is the full error covariance matrix for $vec(\mathbf{X}_a^{\mathrm{T}})$, providing information about the error covariance among all the measurements. Similar equations can be obtained by making the appropriate substitutions for the second and third mode in a trilinear case, or to the other dimensions in a multilinear case. Based on this, an alternating regression algorithm similar to that in the preceding section can be formulated as shown in Table II.

In a manner analogous to the ALS algorithm for heteroscedastic errors, the generalized algorithm presented in Table II uses the maximum likelihood estimates in two spaces to estimate the solution in the other space. In order to exchange the solutions, the error covariance matrix for $vec(\mathbf{X}_a^{\mathrm{T}})$, given by $\boldsymbol{\Omega}_a$, needs to be modified to give the error covariance matrix for $vec(\mathbf{X}_b^{\mathrm{T}})$ and $vec(\mathbf{X}_c^{\mathrm{T}})$, given by $\boldsymbol{\Omega}_b$ and $\boldsymbol{\Omega}_c$, respectively. This can be done on an element-by-element basis; however, since these matrices contain the same elements in a different order, it is simpler to apply a special type of matrix called a permutation matrix to carry out the rearrangement. The permutation matrix is an orthonormal matrix that changes the arrangement of the elements. Conveniently, the same permutation matrix that is used to convert error covariance matrices can also be used to convert between the vectorized forms of $\mathbf{X}_a$,$\mathbf{X}_b$ and $\mathbf{X}_c$. Equations (19)–(22) show how this is done:

$$vec(\mathbf{X}_b^{\mathrm{T}}) = \mathbf{P}_b vec(\mathbf{X}_a^{\mathrm{T}}) \qquad (19)$$

$$vec(\mathbf{X}_c^{\mathrm{T}}) = \mathbf{P}_c vec(\mathbf{X}_a^{\mathrm{T}}) \qquad (20)$$

$$\boldsymbol{\Omega}_b = \mathbf{P}_b\boldsymbol{\Omega}_a\mathbf{P}_b^{\mathrm{T}} \qquad (21)$$

$$\boldsymbol{\Omega}_c = \mathbf{P}_c\boldsymbol{\Omega}_a\mathbf{P}_c^{\mathrm{T}} \qquad (22)$$

The construction of the permutation matrices $\mathbf{P}_b$ and $\mathbf{P}_c$, which consist only of ones and zeros, is conceptually straightforward but algorithmically involved, so the details of this will not be presented here.

The algorithm presented in Table II represents a completely general treatment for the case where correlation can exist among all of the measurement errors. Although it is presented for the trilinear case, extension to higher orders is trivial. The algorithm also has very good convergence characteristics and gives results that are identical to those obtained by the algorithm in Section 2.1 in the presence of uncorrelated noise. In practice, implementation of the algorithm is limited to some extent by the size and stability of the matrices and the convergence time. These three factors are not completely independent from one another. For example, as $\underline{\mathbf{X}}$ becomes large, the associated error covariance matrices tend to become ill-conditioned, causing convergence problems. A variety of approaches, such as compres-

Table II. General MLPARAFAC algorithm (correlated measurement errors).

1. Given an $I \times J \times K$ cube of data $\underline{\mathbf{X}}$, a corresponding $IJK \times IJK$ matrix $\mathbf{\Omega}_a$ of error covariances for $vec(\mathbf{X}_a)$ and two permutation matrices $\mathbf{P}_b$ and $\mathbf{P}_c$ to permute from $vec(\mathbf{X}_a)$ to $vec(\mathbf{X}_b)$ and $vec(\mathbf{X}_c)$, respectively, the algorithm is initialized using random values of the correct dimensions or using estimates obtained by TLD.

$$[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] = tld(\underline{\mathbf{X}}, P) \tag{T12}$$

2. Unfold and vectorize $\underline{\mathbf{X}}$ retaining the first order and calculate the maximum likelihood estimation of $\hat{\mathbf{A}}$ conditional on $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

$$vec(\mathbf{X}_a^{\mathrm{T}}) = vec(unfold(\underline{\mathbf{X}}, a)^{\mathrm{T}}); \hat{\mathbf{V}}_a = \mathbf{I}_I \otimes \hat{\mathbf{Z}}_a^{\mathrm{T}};\ \mathbf{\Omega}_a \tag{T13}$$

$$vec(\hat{\mathbf{A}}^{\mathrm{T}}) = (\hat{\mathbf{V}}_a^{\mathrm{T}} \mathbf{\Omega}_a^{-1} \hat{\mathbf{V}}_a)^{-1} \hat{\mathbf{V}}_a^{\mathrm{T}} \mathbf{\Omega}_a^{-1} vec(\mathbf{X}_a^{\mathrm{T}}) \tag{T14}$$

Here $vec(\hat{\mathbf{A}}^{\mathrm{T}})$ is the vectorized row form of $\hat{\mathbf{A}}$. Using this estimate and the estimates of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ the objective function can be calculated using Equation (T15).

$$\Delta\mathbf{X}_a = (\mathbf{X}_a - \hat{\mathbf{X}}_a);\ S_a^2 = vec(\Delta\mathbf{X}_a)^{\mathrm{T}} \mathbf{\Omega}_a^{-1} vec(\Delta\mathbf{X}_a) \tag{T15}$$

3. Vectorize $\underline{\mathbf{X}}$ retaining the second order and calculate the maximum likelihood estimation of $\hat{\mathbf{B}}$ conditional on $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$.

$$vec(\mathbf{X}_b^{\mathrm{T}}) = \mathbf{P}_b vec(\mathbf{X}_a^{\mathrm{T}}); \hat{\mathbf{V}}_b = \mathbf{I}_J \otimes \hat{\mathbf{Z}}_b^{\mathrm{T}};\ \mathbf{\Omega}_b = \mathbf{P}_b \mathbf{\Omega}_a \mathbf{P}_b^{\mathrm{T}} \tag{T16}$$

$$vec(\hat{\mathbf{B}}^{\mathrm{T}}) = (\hat{\mathbf{V}}_b^{\mathrm{T}} \mathbf{\Omega}_b^{-1} \hat{\mathbf{V}}_b)^{-1} \hat{\mathbf{V}}_b^{\mathrm{T}} \mathbf{\Omega}_b^{-1} vec(\mathbf{X}_b^{\mathrm{T}}) \tag{T17}$$

Here $vec(\hat{\mathbf{B}}^{\mathrm{T}})$ is the vectorized row form of $\hat{\mathbf{B}}$. Using this estimate and the estimates of $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$ the objective function can be calculated using Equation (T18).

$$\Delta\mathbf{X}_b = (\mathbf{X}_b - \hat{\mathbf{X}}_b);\ S_b^2 = vec(\Delta\mathbf{X}_b)^{\mathrm{T}} \mathbf{\Omega}_b^{-1} vec(\Delta\mathbf{X}_b) \tag{T18}$$

4. Vectorize $\underline{\mathbf{X}}$ retaining the third order and calculate the maximum likelihood estimation of $\hat{\mathbf{C}}$ conditional on $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

$$vec(\mathbf{X}_c^{\mathrm{T}}) = \mathbf{P}_c vec(\mathbf{X}_a^{\mathrm{T}}); \hat{\mathbf{V}}_c = \mathbf{I}_K \otimes \hat{\mathbf{Z}}_c^{\mathrm{T}};\ \mathbf{\Omega}_c = \mathbf{P}_c \mathbf{\Omega}_a \mathbf{P}_c^{\mathrm{T}} \tag{T19}$$

$$vec(\hat{\mathbf{C}}^{\mathrm{T}}) = (\hat{\mathbf{V}}_c^{\mathrm{T}} \mathbf{\Omega}_c^{-1} \hat{\mathbf{V}}_c)^{-1} \hat{\mathbf{V}}_c^{\mathrm{T}} \mathbf{\Omega}_c^{-1} vec(\mathbf{X}_c^{\mathrm{T}}) \tag{T20}$$

Here $vec(\hat{\mathbf{C}}^{\mathrm{T}})$ is the vectorized row form of $\hat{\mathbf{C}}$. Using this estimate and the estimates of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ the objective function can be calculated using Equation (T21).

$$\Delta\mathbf{X}_c = (\mathbf{X}_c - \hat{\mathbf{X}}_c);\ S_c^2 = vec(\Delta\mathbf{X}_c)^{\mathrm{T}} \mathbf{\Omega}_c^{-1} vec(\Delta\mathbf{X}_c) \tag{T21}$$

5. Calculate the convergence parameters λ_1 and λ_2.

$$\lambda_1 = (S_b^2 - S_a^2)/S_a^2; \lambda_2 = (S_c^2 - S_a^2)/S_a^2 \tag{T22}$$

If λ_1 and λ_2 are less than the convergence limit (typically 10^{-8} in this work), terminate. Otherwise return to step 2.

sion [14], line search extrapolation [16] and simplifications based on the error structure [17], may be adapted to the present algorithm to avoid these problems. The first two modifications will not be treated in this paper since the first is beyond the scope of the present work and the second is primarily an algorithmic modification to the ALS algorithm. However, the third approach has important practical implications and for this reason will be the focus of the next section.

2.3. Simplification: correlation along one order only

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 (i.e. in the second order, as illustrated in Figure 1) will be considered, since correlations along other orders can be rendered equivalent through permutation of the original array or appropriate adjustment of equations which will be presented. For this case, three common cases 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. This section will focus in the second and third cases, since the first case can only be treated using the general algorithm. To begin, however, it is helpful to examine the second case, which is more general and can be extended to the third case in a straightforward manner.

Imagine a trilinear data set such as the example presented in Section 2.2, where the error correlation can be expected to affect only one order, which we will assume to be the second order as noted above. 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 the same vertical slice of data. Considering that the correlation occurs along the rows of $\mathbf{X}_a$

and is the same in each row, all the covariance information is contained in a single $JK \times JK$ covariance matrix $\boldsymbol{\Psi}_a$ defined by

$$\boldsymbol{\Psi}_a = E[(\mathbf{x}_a - \mathbf{x}_a^{\circ})^{\mathrm{T}} \cdot (\mathbf{x}_a - \mathbf{x}_a^{\circ})] \tag{23}$$

where $\mathbf{x}_a$ and $\mathbf{x}_a^{\circ}$ can represent any row of $\mathbf{X}_a$ and $\mathbf{X}_a^{\circ}$, the unfolded forms of the measured data array and the error-free data array, respectively. Of course, $\mathbf{X}_a^{\circ}$ is not generally known, so in the absence of *a priori* knowledge of the error covariance matrix, $\boldsymbol{\Psi}_a$, might typically be estimated by obtaining replicates of the measurements for each row and using the means in place of $\mathbf{x}_a^{\circ}$, then pooling all of the results, as indicated in Equation (24):

$$\boldsymbol{\Psi}_a \approx \frac{1}{I}\sum_{i=1}^{I}\frac{1}{(N-1)}\sum_{n=1}^{N}({}^{in}\mathbf{x}_a - {}^{i}\bar{\mathbf{x}}_a)^{\mathrm{T}}({}^{in}\mathbf{x}_a - {}^{i}\bar{\mathbf{x}}_a) \tag{24}$$

where ${}^{in}\mathbf{x}_a$ is the nth replicate measurement of the ith row of $\mathbf{X}_a$ and ${}^{i}\bar{\mathbf{x}}_{\mathrm{a}}$ is the mean of the N replicates for that row. (Note that these replicates would likely be obtained through separate experiments for each of the K slices.) Other strategies are also possible, but these will not be discussed in detail here. The full covariance matrix, $\boldsymbol{\Omega}_a$, will now be block diagonal, consisting of I identical diagonal units of dimension $JK \times JK$. This situation offers a number of advantages to the algorithm. From a storage capacity point, the improvement is related to the reduction of the number of non-zero elements from a maximum of $(IJK)^2$ in the general case to $(JK)^2$, since $\boldsymbol{\Omega}_a$, that has the form represented in Equation (25), can also be represented as the sparse Kronecker product shown in Equation (26):

$$\boldsymbol{\Omega}_a = \begin{bmatrix} \boldsymbol{\Psi}_a & & & & \\ & \boldsymbol{\Psi}_a & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \boldsymbol{\Psi}_a \end{bmatrix} \tag{25}$$

$$\boldsymbol{\Omega}_a = \mathbf{I}_I \otimes \boldsymbol{\Psi}_a \tag{26}$$

Additionally, this improves the numerical stability of the algorithm since the Kronecker form allows $\boldsymbol{\Omega}_a$ to be inverted by inversion of the individual covariance matrix $\boldsymbol{\Psi}_a$ as shown in Equation (27):

$$\boldsymbol{\Omega}_a^{-1} = \mathbf{I}_I \otimes \boldsymbol{\Psi}_a^{-1} \tag{27}$$

The companion error covariance matrices for the other orders can be obtained using the permutation matrices via Equations (28) and (29):

$$\boldsymbol{\Omega}_b^{-1} = \mathbf{P}_b\boldsymbol{\Omega}_a^{-1}\mathbf{P}_b^{\mathrm{T}} \tag{28}$$

$$\boldsymbol{\Omega}_c^{-1} = \mathbf{P}_c\boldsymbol{\Omega}_a^{-1}\mathbf{P}_c^{\mathrm{T}} \tag{29}$$

Based on these equations and in the identical block diagonal form of $\boldsymbol{\Omega}_a$, it is easy to demonstrate that the maximum likelihood solution for the $\mathbf{A}$ loadings is obtained using Equation (30):

$$\hat{\mathbf{A}} = \mathbf{X}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}}(\mathbf{Z}_a\boldsymbol{\Psi}_a^{-1}\mathbf{Z}_a^{\mathrm{T}})^{-1} \tag{30}$$

Although the equation for order A under this assumption is analogous in form to Equation (13) for the heteroscedastic case, the rest of the equations needed to implement the ALS algorithm are different. In order to obtain these equations, it should first be realized that $\boldsymbol{\Omega}_c^{-1}$ can be represented as shown in Equation (31), as is apparent from Figure 1, whereas $\boldsymbol{\Omega}_b^{-1}$ cannot be similarly simplified under these circumstances.

$$\boldsymbol{\Omega}_c^{-1} = \begin{bmatrix} {}^{1}\boldsymbol{\Psi}_c^{-1} & & & & \\ & {}^{2}\boldsymbol{\Psi}_c^{-1} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & {}^{K}\boldsymbol{\Psi}_c^{-1} \end{bmatrix} \tag{31}$$

This leads to Equations (32) and (33) for the maximum likelihood estimation of $\mathbf{B}$ and $\mathbf{C}$, respectively, under this assumption:

$$vec(\hat{\mathbf{B}}^{\mathrm{T}}) = (\mathbf{V}_b^{\mathrm{T}}\boldsymbol{\Omega}_b^{-1}\mathbf{V}_b)^{-1}\mathbf{V}_b^{\mathrm{T}}\boldsymbol{\Omega}_b^{-1}vec(\mathbf{X}_b^{\mathrm{T}}) \tag{32}$$

$${}^{k}\hat{\mathbf{c}}^{\mathrm{T}} = \mathbf{x}_c\,{}^{k}\boldsymbol{\Psi}_c^{-1}\mathbf{Z}_c^{\mathrm{T}}(\mathbf{Z}_c\,{}^{k}\boldsymbol{\Psi}_c^{-1}\mathbf{Z}_c^{\mathrm{T}})^{-1} \tag{33}$$

In addition to the storage improvements achieved, speed enhancements are also realized since $\mathbf{A}$ can now be estimated projecting the data at once on to a smaller set of matrices. In order to estimate the $\mathbf{C}$ loading, a row-by-row procedure has to be implemented since the error covariance matrices change from slice to slice. The estimation of $\mathbf{B}$ has to be done using the full error covariance matrix as in the general case since the error covariance terms needed cannot be summarized in a more efficient manner. This algorithm for this simplified case is presented in Table III.

A further simplification is possible when the error covariance matrix is the same for each row in all the slices, a situation which is not uncommon, at least to a first approximation. In this case Equations (27) and (30) can be used to estimate $\mathbf{A}$ as before, and analogous equations can be used to estimate $\mathbf{C}$ by making the appropriate substitutions, since all of the $\boldsymbol{\Psi}_c$s are now the same. The calculations are further simplified by realizing that $\boldsymbol{\Omega}_b^{-1}$, under these noise characteristics, can be expressed as in Equation (34), since the permutation matrix in this case is similar to the commutation matrix used in Reference 17, reducing the estimation of $\mathbf{B}$ to Equation (35):

$$\boldsymbol{\Omega}_b^{-1} = \boldsymbol{\Psi}_b^{-1} \otimes \mathbf{I}_J \tag{34}$$

$$\hat{\mathbf{B}} = \mathbf{X}_b\mathbf{Z}_b^{\mathrm{T}}(\mathbf{Z}_b\mathbf{Z}_b^{\mathrm{T}})^{-1} \tag{35}$$

Table IV gives the algorithm under these assumptions.

2.4. MLPARAFAC with offsets

So far, it has been assumed that the multilinear data are not affected by offsets in any mode. Unfortunately, it is not uncommon in chemical systems to have offsets affecting different orders. The sources of offsets range from instrumental artifacts, such as a spectral background for all samples or variations in cell position, to factors related to sample preparation. One general model to describe trilinear data affected by different kinds of offsets is represented by

Table III. Simplified MLPARAFAC algorithm (Simplification 1—same error covariance matrix for each row in a slice, but different between slices)

1. Given an $I \times J \times K$ cube of data $\underline{\mathbf{X}}$, a corresponding $IJK \times IJK$ matrix $\mathbf{\Omega}_b$ of error covariances for $vec(\mathbf{X}_b)$ and two permutation matrices $\mathbf{P}_b$ and $\mathbf{P}_c$ to migrate from $vec(\mathbf{X}_a)$ to $vec(\mathbf{X}_b)$ and $vec(\mathbf{X}_c)$ respectively. The algorithm is initialized using random values of the correct dimensions or using estimates obtained by TLD.

$$[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] = tld(\underline{\mathbf{X}}, P) \tag{T23}$$

2. Unfold $\underline{\mathbf{X}}$ retaining the first order and calculate the maximum likelihood estimation of $\hat{\mathbf{A}}$ conditional on $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$. Since $\mathbf{\Omega}_a$ is block diagonal as shown in T24, $\hat{\mathbf{A}}$ can be calculated at once.

$$\mathbf{X}_a = unfold(\underline{\mathbf{X}}, a);\ \mathbf{\Omega}_a = \mathbf{P}_b^{\mathrm{T}}\mathbf{\Omega}_b\mathbf{P}_b;\ \mathbf{\Omega}_a = \mathbf{I}_I \otimes \mathbf{\Psi}_a \tag{T24}$$

$$\hat{\mathbf{A}} = \mathbf{X}_a\mathbf{\Psi}_a^{-1}\hat{\mathbf{Z}}_a^{\mathrm{T}}(\hat{\mathbf{Z}}_a\mathbf{\Psi}_a^{-1}\hat{\mathbf{Z}}_a^{\mathrm{T}})^{-1} \tag{T25}$$

Using $\hat{\mathbf{A}}$ and the estimates of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ the objective function can be calculated using Equation (T26).

$$S_a^2 = tr((\mathbf{X}_a - \hat{\mathbf{X}}_a)\mathbf{\Psi}_a^{-1}(\mathbf{X}_a - \hat{\mathbf{X}}_a)^{\mathrm{T}}) \tag{T26}$$

3. Unfold $\underline{\mathbf{X}}$ retaining the second order and calculate the maximum likelihood estimation of $\hat{\mathbf{B}}$ conditional on $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$ using $\mathbf{\Omega}_b$.

$$vec(\mathbf{X}_b^{\mathrm{T}}) = vec(unfold(\mathbf{X}_b))^{\mathrm{T}}; \hat{\mathbf{V}}_b = \mathbf{I}_J \otimes \hat{\mathbf{Z}}_b^{\mathrm{T}};\ \mathbf{\Omega}_b \tag{T27}$$

$$vec(\hat{\mathbf{B}}^{\mathrm{T}}) = (\hat{\mathbf{V}}_b^{\mathrm{T}}\mathbf{\Omega}_b^{-1}\hat{\mathbf{V}}_b)^{-1}\hat{\mathbf{V}}_b^{\mathrm{T}}\mathbf{\Omega}_b^{-1}vec(\mathbf{X}_b^{\mathrm{T}}) \tag{T28}$$

Here $vec(\hat{\mathbf{B}}^{\mathrm{T}})$ is the vectorized row form of $\hat{\mathbf{B}}$. Using this estimate and the estimates of $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$ the objective function can be calculated using Equation (T29).

$$\Delta\mathbf{X}_b = (\mathbf{X}_b - \hat{\mathbf{X}}_b);\ S_b^2 = vec(\Delta\mathbf{X}_b)^{\mathrm{T}}\mathbf{\Omega}_b^{-1}vec(\Delta\mathbf{X}_b) \tag{T29}$$

4. Unfold $\underline{\mathbf{X}}$ retaining the third order and calculate the maximum likelihood estimation of $\hat{\mathbf{C}}$ conditional on $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.${}^{k}\mathbf{\Psi}_c$ is constructed taking the corresponding block of $\mathbf{\Omega}_c$ since it is block diagonal.

$$\mathbf{X}_c = unfold(\underline{\mathbf{X}}, c);\ \mathbf{\Omega}_c = \mathbf{P}_c\mathbf{P}_b^{\mathrm{T}}\mathbf{\Omega}_b\mathbf{P}_b\mathbf{P}_c^{\mathrm{T}};\ {}^{k}\,\mathbf{\Psi}_c \tag{T30}$$

$${}^{k}\hat{\mathbf{c}}^{\mathrm{T}} =^{k} \mathbf{x}_c^{k}\mathbf{\Psi}_c^{-1}\hat{\mathbf{Z}}_c^{\mathrm{T}}(\hat{\mathbf{Z}}_c^{k}\mathbf{\Psi}_c^{-1}\hat{\mathbf{Z}}_c^{\mathrm{T}})^{-1} \tag{T31}$$

Here ${}^{i}\hat{\mathbf{c}}^{\mathrm{T}}$ is a row vector of $\hat{\mathbf{C}}$. Using this estimate and the estimates of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ the objective function can be calculated using Equation (T32).

$$S_c^2 = \sum_{k=1}^{K}({}^{k}\mathbf{x}_c -^{k}\hat{\mathbf{x}}_c)^{k}\mathbf{\Psi}_c^{-1}({}^{k}\mathbf{x}_c -^{k}\hat{\mathbf{x}}_c)^{\mathrm{T}} \tag{T32}$$

5. Calculate the convergence parameters λ_1 and λ_2.

$$\lambda_1 = (S_b^2 - S_a^2)/S_a^2; \lambda_2 = (S_c^2 - S_a^2)/S_a^2 \tag{T33}$$

If λ_1 and λ_2 are less than the convergence limit (typically 10^{-8} in this work), terminate. Otherwise return to step 2.

Equation (36):

$$x_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + \sum_{p=1}^{P} a_{ip}b_{jp}c_{kp} \tag{36}$$

where μ is the grand mean of $\underline{\mathbf{X}}$ and α_i, β_j and γ_k represent the offsets for mode A, B and C, respectively. It has been reported [15] that, in cases where an overall offset exists, it can be removed by eliminating the offset associated with any mode. Therefore, the grand mean can be incorporated into any or all the offset terms affecting the individual modes. When the measurements in $\underline{\mathbf{X}}$ are corrupted by *iid* noise, proper mean centering to remove the offset is a convenient approach since this pre-processing step does not destroy the multilinear characteristics of the data. It is important to note, however, that mean centering will alter the structure of the loadings so that they may no longer be readily identified with real factors, counteracting one of the main benefits of trilinear decomposition.

From a mathematical point of view, the mean centering is equivalent to adding trilinear factors that are formed by the product of a vector of offsets and two other loading vectors set to ones as shown in Equation (37):

$$\underline{\mathbf{X}} = (\boldsymbol{\alpha} \otimes \mathbf{1}_J \otimes \mathbf{1}_K) + (\mathbf{1}_I \otimes \boldsymbol{\beta} \otimes \mathbf{1}_K) + (\mathbf{1}_I \otimes \mathbf{1}_J \otimes \boldsymbol{\gamma}) + \sum_{p=1}^{P} \mathbf{a}_p \otimes \mathbf{b}_p \otimes \mathbf{c}_p \tag{37}$$

Note that Equation (37) is a general formulation and in a given application, the offset affecting any of the modes could be set to zero, i.e. $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ or $\boldsymbol{\gamma}$ could be a zero vector. In addition, it could even be constrained to be a general offset affecting all the measurements equally and then, loadings representing each mode would be equal to a vector of ones and everything multiplied by a scalar representing the offset.

Table IV. Simplified MLPARAFAC algorithm (Simplification 2—same error covariance matrix for each row in each slice)

1. Given an $I \times J \times K$ cube of data $\underline{\mathbf{X}}$, and the error covariance matrices $\boldsymbol{\Psi}_a$ and $\boldsymbol{\Psi}_c$ for the A and C orders, respectively. The algorithm is initialized using random values of the correct dimensions or using estimates obtained by TLD.

$$[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] = tld(\underline{\mathbf{X}}, P) \tag{T34}$$

2. Unfold $\underline{\mathbf{X}}$ retaining the first order and calculate the maximum likelihood estimation of $\hat{\mathbf{A}}$ conditional on $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

$$\mathbf{X}_a = unfold(\underline{\mathbf{X}}, a);\ \boldsymbol{\Psi}_a \tag{T35}$$

$$\hat{\mathbf{A}} = \mathbf{X}_a \boldsymbol{\Psi}_a^{-1} \hat{\mathbf{Z}}_a^{\mathrm{T}} (\hat{\mathbf{Z}}_a \boldsymbol{\Psi}_a^{-1} \hat{\mathbf{Z}}_a^{\mathrm{T}})^{-1} \tag{T36}$$

Using $\hat{\mathbf{A}}$ and the estimates of $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ the objective function can be calculated using Equation (T37).

$$S_a^2 = tr((\mathbf{X}_a - \hat{\mathbf{X}}_a)\boldsymbol{\Psi}_a^{-1}(\mathbf{X}_a - \hat{\mathbf{X}}_a)^{\mathrm{T}}) \tag{T37}$$

3. Unfold $\underline{\mathbf{X}}$ retaining the second order and calculate the maximum likelihood estimation of $\hat{\mathbf{B}}$ conditional on $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$.

$$\mathbf{X}_b = unfold(\underline{\mathbf{X}}, b) \tag{T38}$$

$$\hat{\mathbf{B}} = \mathbf{X}_b \hat{\mathbf{Z}}_b^{\mathrm{T}} (\hat{\mathbf{Z}}_b \hat{\mathbf{Z}}_b^{\mathrm{T}})^{-1} \tag{T39}$$

Using $\hat{\mathbf{B}}$ and the estimates of $\hat{\mathbf{C}}$ and $\hat{\mathbf{A}}$ the objective function can be calculated using Equation (T40).

$$S_b^2 = tr((\mathbf{X}_b - \hat{\mathbf{X}}_b)(\mathbf{X}_b - \hat{\mathbf{X}}_b)^{\mathrm{T}}) \tag{T40}$$

4. Unfold $\underline{\mathbf{X}}$ retaining the third order and calculate the maximum likelihood estimation of $\hat{\mathbf{C}}$ conditional on $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

$$\mathbf{X}_c = unfold(\underline{\mathbf{X}}, c);\ \boldsymbol{\Psi}_c \tag{T41}$$

$$\hat{\mathbf{C}} = \mathbf{X}_c \boldsymbol{\Psi}_c^{-1} \hat{\mathbf{Z}}_c^{\mathrm{T}} (\hat{\mathbf{Z}}_c \boldsymbol{\Psi}_c^{-1} \hat{\mathbf{Z}}_c^{\mathrm{T}})^{-1} \tag{T42}$$

Using $\hat{\mathbf{C}}$ and the estimates of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ the objective function can be calculated using Equation (T43).

$$S_c^2 = tr((\mathbf{X}_c - \hat{\mathbf{X}}_c)\boldsymbol{\Psi}_c^{-1}(\mathbf{X}_c - \hat{\mathbf{X}}_c)^{\mathrm{T}}) \tag{T43}$$

5. Calculate the convergence parameters λ_1 and λ_2.

$$\lambda_1 = (S_b^2 - S_a^2)/S_a^2;\ \lambda_2 = (S_c^2 - S_a^2)/S_a^2 \tag{T44}$$

If λ_1 and λ_2 are less than the convergence limit (typically 10^{-8} in this work), terminate. Otherwise return to step 2.

However, the presence of non-uniform and/or correlated error distribution makes mean centering no longer optimal from a maximum likelihood point of view. This can be understood considering that mean centering in any mode is the projection of $\underline{\mathbf{X}}$ unfolded in this mode onto the null space spanned by the vector of ones corresponding to this mode. Therefore, this projection will only be optimal under *iid* conditions. In order to mean center optimally, the procedure should be incorporated into the ALS algorithm. Contrary to what happens in MLPCA, where the loadings are constrained to be orthogonal, PARAFAC does not impose any constraints on the estimation of the loadings, making the inclusion of offsets in the ALS algorithm a more straightforward task. Additionally, an important benefit is that the offsets may often be uniquely determined because of the uniqueness of the PARAFAC model.

A relatively simple approach to handling offsets can be used when the offsets follow the structure represented by Equation (37). It is clear from this equation that the offsets can be incorporated by using from one to three more factors (in the trilinear case) than the number of factors expected in the absence of offsets. The number of additional factors which should be added depends on how many modes exhibit the offsets in Equation (37). This means of dealing with offsets is easily incorporated into the MLPARAFAC algorithm, and will yield maximum likelihood estimates of the loadings in accordance with the model, but is not the best approach. This is because the loadings in the two modes other than the one in which the offsets occur are allowed to 'float,' that is, they are not constrained to unity (or, more generally, a constant value). While these loadings may be nearly constant and will constitute a maximum likelihood solution to the expanded-factor model, all of the loadings in this case should experience a greater variance than would be expected with the true model. The situation is analogous to fitting simple bivariate straight line data with an intercept of zero to linear models. The data could be fit using only a slope term (intercept forced to zero), or with a slope and intercept term. Both approaches will yield a maximum likelihood solution, but the latter strategy (which has a closer fit but fewer degrees of freedom) will produce a larger variance in the slope, so it is the less preferred method given *a priori* knowledge of a zero intercept. Likewise, if we have prior knowledge of a structure such as that in Equation (37), it is better to incorporate this into the modeling process.

Equation (37) is only one of many possible constrained structures that can exist in trilinear models, and it is clear that the question of offsets is part of a more general issue of constrained factors. The nature of these constraints is very application dependent and relies on prior information. While such constraints can be incorporated into the MLPARAFAC algorithm, a general discussion of strategies is premature and beyond the scope of the current paper. However, one example will be presented in Section 4.5 to demonstrate the performance of MLPARAFAC in the presence of offsets.

2.5. Estimation of error covariance matrices

The error covariance matrix is of critical importance in maximum likelihood methods such as MLPCA and MLPARAFAC. Consequently, questions often arise related to procedures used to estimate the error covariance matrix, the quality of these estimates, and the implications of this on the subsequent analysis. While the emphasis of this work is on the development of the algorithm, it is legitimate to raise these concerns, so they will be addressed here, although only briefly.

Perhaps the most obvious way to estimate the error covariance matrix is through the use of replicates, as indicated in the discussion related to Equation (24). In practice, such an approach may be limited by experimental design considerations or realistic constraints on the number of experiments that can be conducted. Covariance estimates, like variance estimates, are notoriously imprecise unless a large number of replicates are employed. This is often impractical, although pooling can sometimes be used. The question then becomes whether it is better to employ traditional methods (which assume an *iid*-normal error structure) or maximum likelihood methods with a noisy error covariance matrix. Maximum likelihood methods will generally be favored in situations where the number of replicates is large and/or the level of heteroscedastic/correlated noise is high. The precise point at which the use of maximum likelihood methods becomes advantageous will depend on the particular application and a detailed examination of this is beyond the scope of the present work.

An alternative to the often unpopular approach of measuring replicates is to characterize the error covariance structure for a particular instrument or application based on empirical evidence or theoretical considerations. In the same way that certain instruments are known to exhibit proportional noise, it may be possible to obtain a functional form for the error covariance in certain types of applications. This is already done to some degree when multiplicative signal correction (MSC) is applied to near-infrared data dominated by scatter noise. Furthermore, in some circumstances, it may be possible to describe covariance arising from techniques such as filtering or transformation on a purely theoretical basis. By using such approaches, more reliable error covariance matrices can be obtained that are not subject to the statistics of a small number of replicates.

For the work presented here, which is intended to validate the algorithm rather than to demonstrate its practical application, the theoretical error covariance matrix based on noise simulation was used. This removed any uncertainties associated with the error covariance in the statistical validation.

3. EXPERIMENTAL

3.1. Data sets

Since the objective of this work is to describe the theoretical basis of the MLPARAFAC algorithm and to validate its capabilities, all of the data sets employed in this work were simulated so that the rank and error structure could be known with confidence. Future studies will 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 Set 1 was a rank-three data set of dimensions $8 \times 7 \times 4$ used to test the degrees of freedom with conventional PARAFAC algorithm under conditions of *iid* normal noise and compare it with the new algorithms. The loadings for mode A were represented by an 8×3 matrix drawn from a uniform distribution of random numbers from zero to three ($U(0,3)$). Similarly, $\mathbf{B}$ was a 7×3 matrix from $U(0,2)$ and $\mathbf{C}$ was a 4×3 matrix from $U(0,5)$. The error-free data were generated using Equation (3), yielding the 8×28 matrix of error-free data, unfolded to maintain the A mode. The matrix of measurement errors was an 8×28 matrix of normally distributed random numbers ($\mu = 0$, $\sigma = 0.1$, or $N(0,0.1)$), which was added to the error-free data to generate the unfolded form of Data Set 1. This matrix was then folded into a three-way array and passed to the PARAFAC algorithms.

Data Set 2 was a rank-three data set of dimensions $6 \times 7 \times 3$ and was used to test the algorithm under conditions of heteroscedastic but uncorrelated noise. The error-free data were generated in the same fashion as Data Set 1, with the same ranges of loadings but using the corresponding dimensions. The matrix of measurement errors was created by the Hadamard (element-by-element) multiplication of a 6×21 matrix of normally distributed random numbers drawn from $N(0,1)$ and a 6×21 matrix of random numbers, $\mathbf{Q}_a$, drawn from $U(0,0.1)$, representing the matrix of standard deviation for each measurement in $\mathbf{X}_a$. The noise matrix and the error-free data set were added and the resultant matrix was folded.

The error-free part of Data Set 3, which was used to test the general algorithm for correlation in multiple orders, is identical with Data Set 1. The noise matrix was created to introduce non-uniform and correlated noise at the same time. Initially, an 8×28 matrix of normally distributed random numbers drawn from $N(0,1)$ was generated and multiplied in an element-by-element fashion by one-tenth of the value of the error-free measurements. The resultant matrix was treated with a 15-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 to the opposite side in order to eliminate edge effects. Since the error matrix was unfolded to maintain mode A, this approach produced correlation among the measurements in the two other modes. Although this approach is not particularly realistic, it represents a general case for which the covariance structure could be easily predicted. Again, the error-free data set was added to the noise matrix in order to generate the data set.

Data Sets 4 and 5 were $5 \times 8 \times 4$ matrices, again formed by three components in the same manner as already described for error-free data. These data sets were used to test simplifications to the general algorithm related to the error covariance structure. In both cases, the error-free data were the same and only the measurement error matrices differed. The noise matrix for Data Set 4 was generated to simulate a system where the errors are correlated along only one order (the B mode) and the error covariance matrix is identical for each vector in this mode. To do this, four 5×8 matrices of normally distributed random numbers drawn from $N(0,0.1)$ were generated and all of them were individually treated with a five-point moving average filter along the rows. The filtered error matrix was added to the error-free matrix and used in the simulations. For Data Set 5, the correlated errors were also only in one order and all the vectors in a given 'slice' (mode C fixed) had the same error covariance structure, but this structure varies from slice to slice. The measurement error matrices for this data set were generated in the same manner as for Data Set 4, but the standard deviation of the normal distribution and the filter width were varied between slices (σ = 0.15, 0.2, 0.1, 0.05; w = 3, 5, 7, 3).

Data Set 6, which was used to test the performance in the presence of offsets, was constructed from a $7 \times 8 \times 4$ rank three matrix with the same distribution of loading values and the same noise correlation structure as Data Set 3—heteroscedastic and correlated in two orders. In this case, however, a single vector offset was added to the second order, that is, a 1×8 vector of values drawn from $U(0,2)$ was added to each row of the three way array.

3.2. Computational aspects

All the calculations were carried out on a Sun Ultra 60 workstation with 2×300 MHz processors and 512 MB of RAM and a 700 MHz Pentium-III PC with 128 MB of RAM. All programs were written in-house using Matlab 6.0 (The MathWorks, Natick, MA, USA).

4. RESULTS AND DISCUSSION

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. 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, 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 included in Matlab as shown in Equation (38):

$$P(S^2|v) = \Gamma_{inc}\left(\frac{S^2}{2}, \frac{v}{2}\right) \tag{38}$$

where v is the number of degrees of freedom. If the two distributions are the same, a plot of the observed probabilities vs 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. It should be pointed out that the only way to employ this approach is to use simulated data where the true noise characteristics are known. Because error estimates for virtually all experimental measurements will have some (often substantial) degree of uncertainty, the resulting S^2 values will not follow a χ^2 distribution. (For this reason, it can be argued that the present methods are not truly 'maximum likelihood,' since they should also estimate the error covariance, but this is not practical in most situations.)

The issue of degrees of freedom for trilinear data is far from being trivial. Bro suggested that degrees of freedom do not exist *a priori* [18], but have to be determined from the specific data. This situation arises from the fact that the rank of a trilinear data set cannot be calculated based on the same approach used in bilinear data. For instance, the maximum rank of a $3 \times 3 \times 3$ array is five [19], contrary to what happens in bilinear data, where the maximum rank of a 3×3 matrix is always three. Unfortunately, there is no simple rule for calculating the maximum rank of arrays in general, except for the bilinear case and some simple trilinear arrays. However, Durell *et al.* [20]. reported two equations to calculate the degrees of freedom in trilinear and quadrilinear models, as given in Equation (39) and (40):

$$v(\underline{\mathbf{X}})_{3-\mathrm{way}} = IJK - P(I + J + K - 2) \tag{39}$$

$$v(\underline{\mathbf{X}})_{4-\mathrm{way}} = IJKL - P(I + J + K + L - 3) \tag{40}$$

The theoretical foundation of these equations is not completely clear, but it has been suggested in the literature that they might be used for exploratory (qualitative) purpose. In other words, they should not be used as the statistically correct number of degrees of freedom. In the present work, the approach was to use Equation (39) as estimator of the statistically meaningful number of degrees of freedom for a trilinear case. In order to assess the merit of this approach, trilinear data corrupted with *iid* normal noise, such as Data Set 1, were submitted to the standard PARAFAC algorithm, which is well known to yield the maximum likelihood solution under these noise characteris-

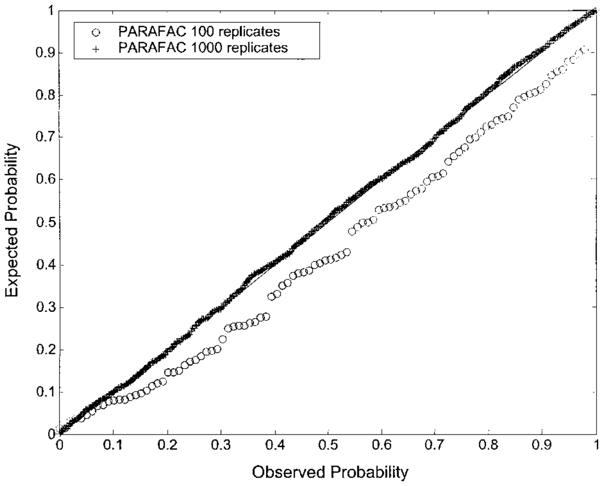

Figure 2. Probability plot for PARAFAC results under conditions of *iid* normal errors for 100 (○) and 1000 (+) replicates. The solid line with unity slope indicates ideal behavior for maximum likelihood estimation.

tics. The replication procedure described above was performed using 100 and 1000 replicates and the probability plot, shown in Figure 2, was constructed. It is observed that the plot follows the theoretical slope very closely for 1000 replicates, indicating that Equation (39) provides a credible number of degrees of freedom, at least for the purposes of this study. For 100 replicates, the agreement is not as good owing to the smaller sample size, but these results are included as a point of reference for other studies that involve only 100 replicates. It is worth noting that, even though the results are not shown, analysis of all of the trilinear data structures used in this work was carried out under *iid* conditions using PARAFAC to confirm the estimated degrees of freedom.

4.2. Non-uniform (uncorrelated) measurement errors: Data Set 2

In order to test the validity of the algorithm depicted in Table I, Data Set 2, which was corrupted with heteroscedastic error, was employed. Since the main objective of this study is the statistical validation of the algorithm, the theoretical error covariance matrix obtained from the simulation parameters was employed. The theoretical error covariance matrix for each row is calculated using the equation

$$^{i}\mathbf{\Psi}_a = diag(^{i}\mathbf{q}_a)^2 \tag{41}$$

where *diag*() represents the diagonalization operator that transforms the vector argument into a diagonal matrix. The result is a diagonal matrix with the squared elements of $^{i}\mathbf{q}_a$ (the *i*th row of $\mathbf{Q}_a$, the matrix of standard deviations unfolded in the A mode) along the diagonal. Error covariance matrices for the other orders were obtained using the same equation applied to $\mathbf{Q}_b$ and $\mathbf{Q}_c$, respectively.

Figure 3 shows the results obtained for the analysis of Data Set 2 using PARAFAC and the version of MLPARAFAC designed to accommodate heteroscedastic noise. The S^2 values in both cases were calculated in the same manner, i.e. using Equation (9) with either the PARAFAC or MLPARAFAC estimates of $\hat{\mathbf{X}}_a$. It is clear from the figure that the estimates obtained using MLPARAFAC follow the expected behavior for maximum likelihood estimation, with only minor deviations attributable to the statistical limitations of the study. On the other hand, the models obtained by PARAFAC do not adequately account for the systematic variance in the data set, producing suboptimal solutions that deviate radically from the line representing expected χ^2 distribution in the probability plot.

Although this data set was not designed to test the more general algorithm depicted in Table II, it was also analyzed using that algorithm to test its generality. The general algorithm produced exactly the same set of solutions, indicating that the two algorithms are equivalent under these noise characteristics.

4.3. Non-uniform and correlated measurement errors: Data Set 3

In the preceding section, it was noted that the general MLPARAFAC algorithm for correlated errors was also able to handle the case of uncorrelated errors. Data Set 3 was designed to test the general algorithm in the presence of errors which were correlated and heteroscedastic. Again, the theoretical error covariance matrix was used. For this specific data set, the covariance matrix in the A mode is given by

$$\mathbf{\Omega}_a = \begin{bmatrix} ^{1}\mathbf{\Psi}_a & & & \\ & ^{2}\mathbf{\Psi}_a & & \\ & & \cdot & \\ & & & ^{I}\mathbf{\Psi}_a \end{bmatrix} \tag{42}$$

where $^{i}\mathbf{\Psi}_a$ represents the error covariance matrix of the *i* row of $\mathbf{X}_\mathrm{a}$ and was calculated using the equation

$$^{i}\mathbf{\Psi}_a = \mathbf{F}^\mathrm{T}(diag((0.1)\cdot {}^{i}\mathbf{x}^{\circ}_a))^2\mathbf{F} \tag{43}$$

where $\mathbf{F}$ is the 28 × 28 filter matrix designed to carry out the 15-point moving average smooth on the noise ($\mathbf{e}_{corr} = \mathbf{e}_{iid}\ \mathbf{F}$),

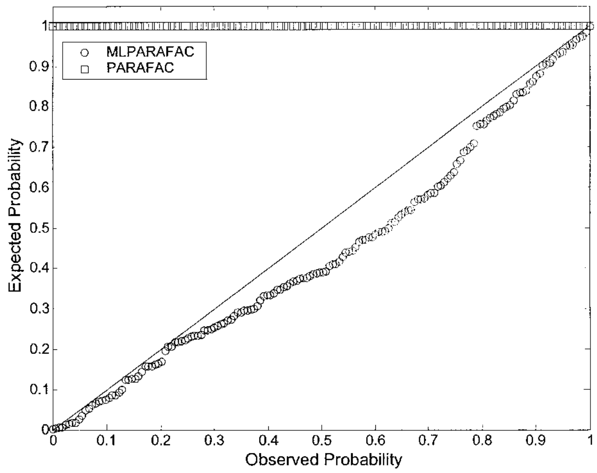

Figure 3. Probability plot for the analysis of 100 replicates of Data Set 2 (non-uniform, uncorrelated errors) by MLPARAFAC (○) and PARAFAC (□).

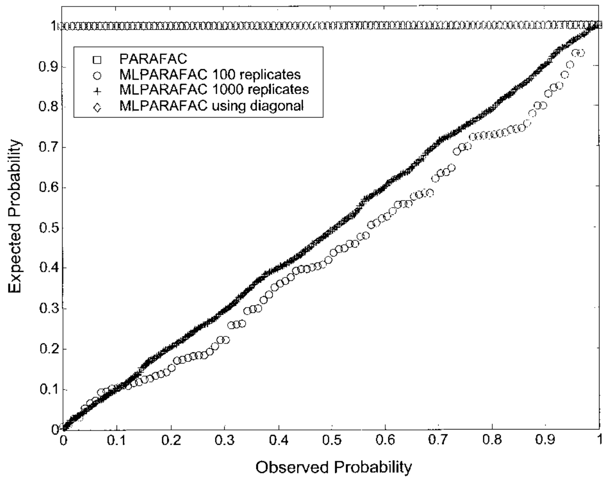

Figure 4. Probability plot for the analysis of Data Set 3 (correlated measurement errors in two modes) using the general MLPARAFAC algorithm with 100 (○) and 1000 (+) replicates, the standard MLPARAFAC algorithm for uncorrelated errors with 100 replicates (◇) and PARAFAC with 100 replicates (□).

and second term is a diagonal matrix of the variance of the noise in the *i*th row of noise matrix prior to smoothing, equal to 10% of the error-free measurement squared. The companion error covariance matrices, $\boldsymbol{\Omega}_b$ and $\boldsymbol{\Omega}_c$, were calculated using their respective permutation matrices as shown in Equations (21) and (22).

Figure 4 shows the probability plots obtained using Data Set 3. Results for the general MLPARAFAC algorithm, which can accommodate any covariance structure, are shown for both 100 and 1000 replicates. Both of these show good agreement with the expected slope of unity, indicating that a maximum likelihood solution has been obtained. In contrast, it is clear that the PARAFAC model has substantial systematic error, since it generates a maximum expected probability of unity across all observed probabilities. In order to test whether the superior performance of the general MLPARAFAC algorithm was due to its inclusion of the error covariance structure or simply because it accounts for heteroscedasticity, results were also generated using the version of MLPARAFAC designed to accommodate only heteroscedasticity. For this analysis, only the diagonal elements of the full error covariance matrix ($\boldsymbol{\Omega}$) were employed. Like the standard PARAFAC algorithm, these models result in systematic errors, indicating that modeling the covariance structure is a critical factor.

4.4. Simplified error covariance structures: Data Sets 4 and 5

While the general MLPARAFAC algorithm should be able to deal with any error covariance structure, in many cases it may be possible to use the simplified algorithms presented in Tables III and IV. These algorithms were tested using Data Sets 4 and 5. Data Set 4, which has a simple error covariance structure consisting of correlation in one mode only and identical error covariance matrices for all the vectors in this mode, was used to test the corresponding algorithm in Table IV, which will be referred to as Simplification 2. The probability plots for this study are shown in Figure 5, together with the results of the generalized algorithm and conventional PARAFAC. Note that the results of the general algorithm and Simplification 2 are identical, confirming that the latter is a special case of the former, and that both appear to produce the maximum likelihood results. As before, the performance of PARAFAC is suboptimal.

Simplification 1, which appears in Table III, is designed to handle the case where (i) error correlation exists in one mode only and (ii) the error covariance structure differs from vector to vector along one of the remaining modes, but is the same along the other remaining mode. Data Set 5, which was simulated to test this algorithm, was created such that errors were correlated along the rows (mode B) and the error covariance matrix was identical for rows within the same slice (mode A), but different across different slices (mode C). The results from analysis of 100 replicates are summarized in Figure 6. As with Simplification 2, the figure shows the identical results for Simplification 1 and the generalized algorithm, both of which produce maximum likelihood estimates, and poor results for PARAFAC.

4.5. MLPARAFAC with offsets: Data Set 6

As noted in Section 2.4, the inclusion of certain kinds of offsets in the trilinear structure can be modeled by using an expanded rank model. This can be demonstrated with Data Set 6, which has offsets added to one order (i.e. $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are zero in Equation (37), but $\boldsymbol{\beta}$ is not). Therefore, expansion of the PARAFAC model to rank four should accommodate the offsets. This is demonstrated with the probability plots in Figure 7, which compares the results of MLPARAFAC (general algorithm) with conventional PARAFAC, both with rank four models. It is clear that MLPARAFAC produces the maximum likelihood solution while PARAFAC does not. Furthermore, this approach to handling offsets is superior to mean-centering in that the integrity of the loading vectors is retained.

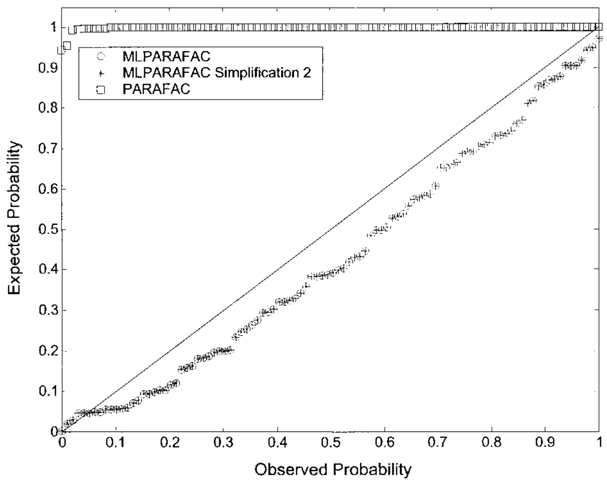

Figure 5. Probability plot for the analysis of 100 replicates of Data Set 4 (identical row correlations) using the general MLPARAFAC algorithm (○), Simplification 2 of the general MLPARAFAC algorithm (*) and PARAFAC (□).

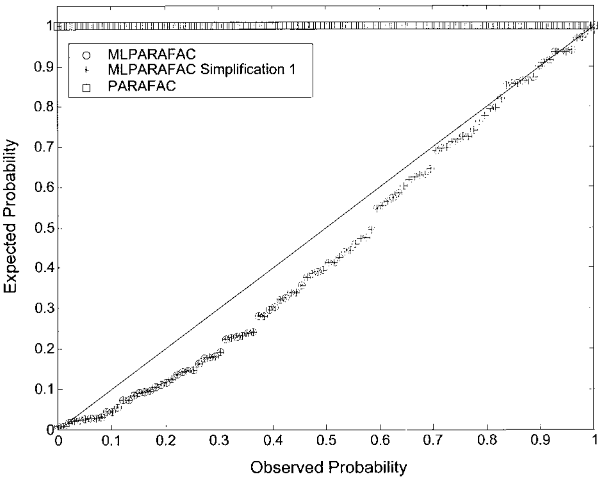

Figure 6. Probability plot for the analysis of 100 replicates of Data Set 5 (different row correlations along mode A, same row correlations along mode C) using the general MLPARAFAC algorithm (○), Simplification 1 of the MLPARAFAC algorithm (*) and PARAFAC (□).

As noted in Section 2.4, the maximum likelihood solution extracted in this manner does not represent the 'best' solution in this application because information about constraints on the loading vectors in the A and C modes of the offset factor (i.e. that they are fixed) is not incorporated into the ALS algorithm. While it is possible to do this, the inclusion of constrained factors adds algorithmic complications and introduces questions regarding degrees of freedom, so this issue will not be dealt with in this paper.

4.6. Model quality

The preceding sections 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 the implication has been that the MLPARAFAC solutions are better, two reasonable questions that arise are (1) are the MLPARAFAC estimates closer to the true underlying factors than the PARAFAC estimates?, and (2) do the MLPARAFAC estimates offer a significant advantage over the estimates obtained by PARAFAC?. The first question can be answered easily using simulated data. The second question is more general in essence and it can only be addressed on a case-by-case basis since the advantages gained by MLPARAFAC will strongly depend on the type and magnitude of error corrupting the data and the correct use of a number of parameters related to the estimation of the model. Some of the parameters determining the success of MLPARAFAC over PARAFAC are the number of components, accuracy of the estimation of the error covariance matrix, and the use of the correct algorithm based on the error structure present.

The first issue, the closeness of estimates to the true factors, will be addressed 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

$$\theta_p^a = \cos^{-1}\left(\frac{\hat{\mathbf{a}}_p^{\mathrm{T}}\mathbf{a}_p}{\|\hat{\mathbf{a}}_p\|\|\mathbf{a}_p\|}\right) \tag{44}$$

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 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.

To evaluate the accuracy of the loadings extracted by MLPARAFAC and PARAFAC under different conditions, loadings extracted from 100 replicates of Data Sets 2–6 by both MLPARAFAC and PARAFAC were used to calculate vector angles for each of the loadings. These angles were then averaged over the 100 replicates to give nine mean angles and their standard deviations (3 modes × 3 factors) for each method. These results are summarized in compressed form in Table V, which, in the interest in saving space, shows only the results for the first loading vector in each mode. The uncertainty given is the population standard deviation.

The results in Table V support the general view that MLPARAFAC produces more accurate estimates of the loading vectors than PARAFAC. Both the mean vector angles and their uncertainties are smaller in all cases for MLPARAFAC, although the degree to which this is true varies with the data set. For Data Set 2, the differences between the two methods is relatively small. This might be expected, however, since this data set contains heteroscedastic errors only with no correlated errors, and the degree of heteroscedasticity, arising from proportional errors, is not very large. Nevertheless, differences are statistically significant (note that the standard deviation of the mean will be

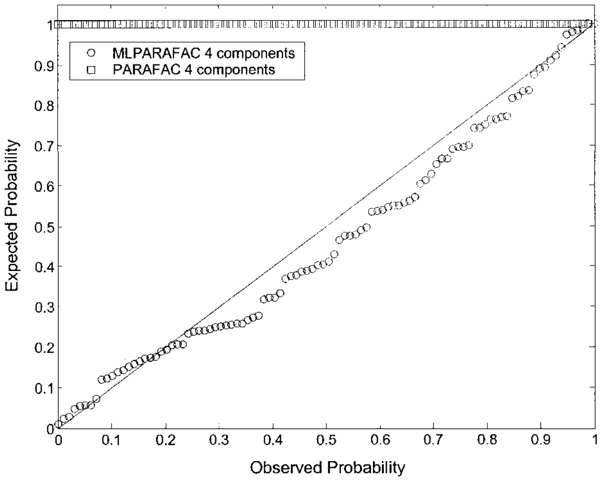

Figure 7. Probability plot for the analysis of 100 replicates of Data Set 6 (correlation along modes B and C plus offset on modeB) using the general MLPARAFAC algorithm (○) and PARAFAC (□).

Table V. Comparison of vector angle accuracies for PARAFAC and MLPARAFAC: results are based on 100 replicates and uncertainties are given as standard deviations

	Mean angular deviation (°)					
	PARAFAC			MLPARAFAC		
Data set	**A**	**B**	**C**	**A**	**B**	**C**
2	0.27 ± 0.15	0.33 ± 0.13	0.21 ± 0.18	0.17 ± 0.09	0.19 ± 0.08	0.14 ± 0.11
3	0.90 ± 0.36	0.61 ± 0.34	0.58 ± 0.37	0.08 ± 0.02	0.14 ± 0.05	0.09 ± 0.04
4	0.17 ± 0.07	0.27 ± 0.14	0.21 ± 0.16	0.07 ± 0.04	0.19 ± 0.08	0.10 ± 0.09
5	0.25 ± 0.12	0.43 ± 0.23	0.32 ± 0.25	0.10 ± 0.05	0.23 ± 0.16	0.16 ± 0.16
6	1.77 ± 1.30	3.04 ± 1.16	1.52 ± 1.03	0.24 ± 0.12	0.47 ± 0.14	0.31 ± 0.19

the value reported in the table divided by 10). The differences are much more dramatic for Data Set 3, which has correlated errors in two modes, and illustrates the importance of modeling error covariance. To further emphasize this point, the analysis of Data Set 3 by MLPARAFAC assuming only heteroscedastic errors (i.e. using only the diagonal) produced corresponding vector angles of 0.92 ± 0.37, 0.59 ± 0.34 and 0.57 ± 0.35, which are not significantly different from the PARAFAC results. Data Sets 4 and 5, which exhibit a smaller degree of error covariance than Data Set 3, also show less dramatic differences between MLPARAFAC and PARAFAC, but the angular differences are still about a factor of two and are statistically very significant. The analysis of these two data sets employed the simplified algorithms, but it should be noted that the general algorithm produced identical results, as expected. In Data Set 6, the addition of a fourth factor representing the offset decreases the quality of the estimates in general compared to Data Set 3 (the most similar data set). Because of the highly correlated error structure, this data set exhibits a difference of a factor of five or more in the mean vector angles obtained by the two methods. For comparison purposes, the corresponding vector angles for the rank three MLPARAFAC model are 1.52 ± 0.55, 3.79 ± 0.17 and 1.44 ± 0.42, indicating that the inclusion of the fourth factor to model the offset is essential.

These results clearly demonstrate that improved estimates of loadings can be obtained from the trilinear model when information about the measurement error structure is available 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 nature of the application and the characteristics of the noise. Furthermore, the results presented here were obtained assuming an absolute knowledge of the measurement error covariance matrix, but in practice this is typically estimated on the basis of replicate measurements and hence may be less reliable. The benefits of including measurement error information must therefore be weighed against the detrimental effects of including poor quality information. The development of the algorithms presented here has demonstrated the potential for improvements that could be achieved and facilitates application to more practical situations in which an experimental assessment of their benefits can be made.

5. CONCLUSIONS

Four algorithms for carrying out MLPARAFAC based on an ALS framework have been described. The simplest of these is 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 was significantly better than for the PARAFAC models in cases where the error covariance matrix is known.

Although the principles of MLPARAFAC have been established here, a number of more practical aspects related to its implementation remain to be examined. These include issues related to the computational efficiency and stability of the algorithms for large arrays, the estimation of error covariance matrices for three-way data, and the implementation of constraints on the loadings within the algorithms. These subjects will be the focus of future investigations.

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. Hirschfeld T. The hy-phen-ated methods. *Anal. Chem.* 1980; **52**: 297A–312A.
2. Apellof CJ and Davison ER. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. *Anal. Chem.* 1981; **53**: 2053–2056.
3. Leurgans S and Ross RT. Multilinear models: applications in spectroscopy. *Statist. Sci.* 1992; **7**: 289–319.
4. Harshman RA. Foundations of the PARAFAC procedure: model and conditions for an 'explanatory' multimode factor analysis. *Work. Pap. Phonetics* 1970; **16**: 1–84.
5. Sanchez E and Kowalski BR. Tensorial resolution: a direct trilinear decomposition. *J. Chemom.* 1990; **4**: 29–45.
6. Paatero P. A weighted non-negative least squares

algorithm for three-way "PARAFAC" factor analysis, *Chemom. Intell. Lab. Syst.* 1997; **38**: 223–242.

7. Yates F. The analysis of replicated experiments when the field results are incomplete. *Emp. J. Exp. Agric.* 1933; **1**: 129–142.
8. Kiers HAL. Weighted least squares fitting using ordinary least squares algorithms. *Psychometrika* 1997; **62**: 251–266.
9. Wentzell PD, Andrews DT, Hamilton DC, Faber K and Kowalski BR. Maximum likelihood principal component analysis. *J. Chemom.* 1997; **11**: 339–366.
10. Bro R, Sidiropoulos ND and Smilde AK. Maximum likelihood fitting using simple least squares algorithms. *J. Chemom.* 2002; **16**: 387–400.
11. Carroll JD and Chang J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 1970; **35**: 283–319.
12. Rao CR and Mitra S. *Generalized Inverse of Matrices and its Applications*. Wiley: New York, 1971.
13. Magnus JR and Neudecker H. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley: Chichester, 1988.
14. Andersson CA and Bro R. Improving the speed of multiway algorithms. Part I: Tucker3. *Chemom. Intell. Lab. Syst.* 1998; **42**: 93–103.
15. Bro R and Smilde AK. Centering and scaling in component analysis. *J. Chemom.* 2003; **17**: 16–33.
16. Ross RT and Leurgans S. Component resolution using multilinear models. *Methods Enzymol.* 1995; **246**: 679–700.
17. Wentzell PD and Lohnes MT. Maximum likelihood principal component analysis with correlated measurement errors: theoretical and practical considerations. *Chemom. Intell. Lab. Syst.* 1999; **45**: 65–85.
18. Bro R. *Multi-way analysis in the food industry: models, algorithms and applications*. PhD Thesis, Univerteit van Amsterdam, 1998.
19. Kruskal JB. Rank, decomposition, and uniqueness for 3-way and N-way arrays. In: *Multiway Data Analysis*, Coppi R, Bolasco S (eds). North-Holland: Amsterdam, 1989; 7.
20. Durell SR, Lee C, Ross RT and Gross EL. Factor analysis of the near-ultraviolet absorption spectrum of plastocyanyn using bilinear, trilinear and quadrilinear models. *Arch. Biochem. Biophys.* 1990; **278**: 148–160.