JOURNAL OF CHEMOMETRICS
J. Chemometrics **13,** 557–578 (1999)

ALTERNATING COUPLED VECTORS RESOLUTION (ACOVER) METHOD FOR TRILINEAR ANALYSIS OF THREE-WAY DATA

JIAN-HUI JIANG, HAI-LONG WU, YANG LI AND RU-QIN YU*
College of Chemistry and Chemical Engineering, Hunan University, Changsha 410082, People's Republic of China

SUMMARY

A new method, alternating coupled vectors resolution (ACOVER), is developed for trilinear analysis of a three-way data array. First, based on the least-squares principle, four coupled vectors resolution (COVER) errors are proposed. Second, a procedure which resolves the profiles of each component successively is developed. This characteristic of successive resolution provides a natural way to avoid the two-factor degeneracy. Experimental results show that the ACOVER method has the advantage that the resolved profiles of analytical interest are very stable with respect to the estimated component number when the number of components is chosen to be equal to or greater than the actual model dimensionality. This circumvents the dilemma of determining a proper component number for the model, which is difficult to handle for the PARAFAC algorithm. Moreover, the method has much higher convergence rate than the PARAFAC algorithm. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: alternating coupled vectors resolution (ACOVER); trilinear model; three-way data analysis; coupled vectors resolution (COVER) error; PARAFAC algorithm

INTRODUCTION

With recent progress in high-order analytical instrumentation and data collection techniques, three-way data analysis has constituted an active trend in chemometric research.[1–11] There are two main types of methodologies for three-way data resolution in chemometrics. Methods of the first type are aimed at resolving directly the 'true' profiles on the basis of eigenanalysis or generalized eigenanalysis. Well-known examples are the generalized rank annihilation method (GRAM)[12–14] and its extension, the trilinear decomposition (TLD) method.[15,16] These methods yield direct solutions to the component profiles in each order. Unfortunately, in the procedures, two pseudosamples need be constructed to formulate eigenproblems, which leads to a loss of information in multiple samples. Moreover, the formulation of the generalized eigenproblems does not take the measurement errors into account. This induces the methods to produce imaginary solutions and exhibit inflated variance. Generally, these approaches work well only in situations where the signal-to-noise ratio is very high. Methods of the second type,[17–23] typically exemplified by the PARAFAC algorithm,[18–23] seek to fit a trilinear model to the data using an iterative algorithm. These methods allow full use of multisample

* Correspondence to: R.-Q. Yu, College of Chemistry and Chemical Engineering, Hunan University, Changsha 410082, People's Republic of China.
E-mail: rqyu@mail.hunu.edu.cn
Contract/grant sponsor: National Natural Science Foundation of China.
Contract/grant sponsor: 29735150.

CCC 0886–9383/99/060557–22 $17.50

Received 8 October 1998
Accepted 10 May 1999

information. However, the PARAFAC algorithm is generally computationally intensive and not guaranteed to locate the global optimum. In addition, the profiles finally estimated may deviate pronouncedly from the actual ones when the number of components is incorrectly chosen for the model, creating a dilemma hard to handle in practical applications.

In this paper a new approach, alternating coupled vectors resolution (ACOVER), to trilinear analysis of three-way data is developed. Unlike existing methods, ACOVER estimates the profiles of the components one at a time by minimizing two least squares-based coupled vectors resolution (COVER) errors. A salient virtue of ACOVER is that the resolved profiles of analytical interest are very stable with respect to the estimated component number when the component number is chosen to be equal to or greater than the actual one, easing the selection of a proper component number for the model. Additionally, the algorithm of ACOVER is computationally more efficient than the PARAFAC algorithm. Results of a simulated example and a real analytical system show that the proposed method provides acceptable performance in three-way data resolution.

THEORY AND METHODS

Least squares-based coupled vectors resolution (COVER) error

Suppose that the data measured on a chemical process or experiment are collected in an $I \times J \times K$ three-way array $\underline{\mathbf{R}}$ which is generated by the trilinear model[22]

$$\underline{\mathbf{R}} = \sum_{n=1}^{M} \mathbf{x}_n \otimes \mathbf{y}_n \otimes \mathbf{z}_n + \underline{\mathbf{E}} \tag{1}$$

where $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$ are the profiles in three orders respectively of the nth response-active species, $\otimes$ denotes the tensor product and $\underline{\mathbf{E}}$ is the array of measurement errors. In general analytical practice one can assume that the profile matrices $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M)$ have full column rank. To guarantee unique resolution of the trilinear data, one can further assume that in the third order the profile of each component is linearly independent of that of any other component. Notice that this premise is stronger than Kruskal's conditions;[24] however, it is the case most generally considered in the chemometric literature.[25] To achieve unique representation of $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$, it is assumed that $\mathbf{X}$ and $\mathbf{Y}$ are column-wise of unit length; that is, $|| \mathbf{x}_n || = 1$ and $|| \mathbf{y}_n || = 1$, $n = 1, 2, \ldots, M$, where $||.||$ denotes the Euclidean norm of a vector. The goal of trilinear analysis of a three-way data array is to resolve the 'true' profiles $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M)$ such that posterior to resolution the chemical process or experiment under measurement can be interpreted with mitigated difficulty.

By slicing the three-way data array $\underline{\mathbf{R}}$ along the third order, the trilinear model can be expressed in matrix notation as

$$\mathbf{R}_{..k} = \mathbf{X}\mathrm{diag}(\mathbf{z}_{(k)})\mathbf{Y}^{\mathbf{T}} + \mathbf{E}_{..k}, \quad k = 1, 2, \ldots, K \tag{2}$$

where $\mathbf{R}_{..k}$ and $\mathbf{E}_{..k}$ are the kth slices of $\underline{\mathbf{R}}$ and $\underline{\mathbf{E}}$ respectively along the third order, the superscript T symbolizes the transpose of a vector or a matrix, and $\mathrm{diag}(\mathbf{z}_{(k)})$ denotes the diagonal matrix whose diagonal entries are the corresponding ones of $\mathbf{z}_{(k)}$. Here $\mathbf{z}_{(k)}$ is the kth row of $\mathbf{Z}$. From (1) it can be seen that the trilinear model treats its parameter matrices, i.e. the profile matrices in three orders, in a symmetric way. Therefore, if $\underline{\mathbf{R}}$ is sliced along the first and second orders, the trilinear model can also be written as the matrix equations

$$\mathbf{R}_{j\cdot} = \mathbf{Z}\mathrm{diag}(\mathbf{y}_{(j)})\mathbf{X}^{\mathrm{T}} + \mathbf{E}_{j\cdot}, \quad j = 1, 2, \ldots, J \tag{3}$$

$$\mathbf{R}_{i\cdot\cdot} = \mathbf{Y}\mathrm{diag}(\mathbf{x}_{(i)})\mathbf{Z}^{\mathrm{T}} + \mathbf{E}_{i\cdot\cdot}, \quad i = 1, 2, \ldots, I \tag{4}$$

Since it has been assumed that $\mathbf{X}$ and $\mathbf{Y}$ both have full column rank, it is known that there exist vectors $\mathbf{q}_n$ and $\mathbf{p}_n$ ($n = 1, 2, \ldots, M$) satisfying

$$\mathbf{y}_n^{\mathrm{T}}\mathbf{q}_m = \delta_{mn}, \quad m, n = 1, 2, \ldots, M \tag{5}$$

$$\mathbf{x}_n^{\mathrm{T}}\mathbf{p}_m = \delta_{mn}, \quad m, n = 1, 2, \ldots, M \tag{6}$$

where $\delta_{mn} = 1$ if $m = n$ and $\delta_{mn} = 0$ if $m \neq n$. Therefore from (2) one can obtain

$$\mathbf{R}_{\cdot\cdot k}\mathbf{q}_n = z_{kn}\mathbf{x}_n + \mathbf{e}_{k,n}^{(x)}, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, M \tag{7}$$

$$\mathbf{R}_{\cdot\cdot k}^{\mathrm{T}}\mathbf{p}_n = z_{kn}\mathbf{y}_n + \mathbf{e}_{k,n}^{(y)}, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, M \tag{8}$$

It is noteworthy that using $\mathbf{q}_n$ and $\mathbf{p}_n$ to transform the trilinear model (1), which depends on all components, into the models (7) and (8), which are dependent merely upon one component, is a critical step in the development of the method. As will be shown later, based on fitting the single-component models (7) and (8), the trilinear resolution can be accomplished in a component-wise way.

Notice that in (7) $\mathbf{e}_{k,n}^{(x)} = \mathbf{E}_{\cdot\cdot k}\mathbf{q}_n$, which is a vector of errors; then a statistically plausible approach to estimate $\mathbf{q}_n$, $\mathbf{x}_n$ and $\mathbf{z}_n$ is the least squares principle. That is, $\mathbf{q}_n$, $\mathbf{x}_n$ and $\mathbf{z}_n$ can be estimated by the minimizers of the least squares criterion

$$E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n) = \sum_{k=1}^{K} \|z_{kn}\mathbf{x}_n - \mathbf{R}_{\cdot\cdot k}\mathbf{q}_n\|^2 \tag{9}$$

Note that criterion (9) is derived from (7) in terms of the least squares method; thus minimization of this criterion provides a least squares fit to model (7). Because the value of criterion (9) depends only on the three parameter vectors $\mathbf{x}_n$, $\mathbf{q}_n$ and $\mathbf{z}_n$, which are the 'true' profiles of one component to be resolved, minimization of the criterion over these parameters will actually yield the least squares solutions to the 'true' profiles $\mathbf{x}_n$ and $\mathbf{z}_n$ of one component. Unfortunately, there is no method available to determine directly the solutions of $\mathbf{x}_n$, $\mathbf{q}_n$ and $\mathbf{z}_n$ which minimize the criterion. One has to design an iterative algorithm to exploit the solutions. It will be shown that in the algorithm the three parameters are treated differently. The first two vectors, $\mathbf{x}_n$ and $\mathbf{q}_n$, are resolved in a coupled way. Accordingly, these two vectors are called the coupled vectors. Because the goal of the criterion is to resolve the coupled vectors, it is called the coupled vectors resolution (COVER) error.

Analogously, one can derive another COVER error from (2) and (8) as

$$E_{\mathrm{COVER}}(\mathbf{y}_n, \mathbf{p}_n; \mathbf{z}_n) = \sum_{k=1}^{K} \|z_{kn}\mathbf{y}_n - \mathbf{R}_{\cdot\cdot k}^{\mathrm{T}}\mathbf{p}_n\|^2 \tag{10}$$

Minimization of this least squares criterion over the parameters also gives the least squares estimates for the 'true' profiles $\mathbf{y}_n$ and $\mathbf{z}_n$ of one component. Parallelly, the first two parameter vectors, $\mathbf{y}_n$ and $\mathbf{p}_n$, are called the coupled vectors too.

In principle, utilizing COVER errors (9) and (10), one can develop a method for trilinear resolution

based on the ALS algorithm. Nevertheless, to make the algorithm convenient for implementation and derivation, the following two COVER errors are also used in the study:

$$E_{\mathrm{COVER}}(\mathbf{z}_n, \mathbf{q}_n; \mathbf{x}_n) = \sum_{i=1}^{I} \|x_{in}\mathbf{z}_n - \mathbf{R}_{i..}^{\mathrm{T}}\mathbf{q}_n\|^2 \tag{11}$$

$$E_{\mathrm{COVER}}(\mathbf{z}_n, \mathbf{p}_n; \mathbf{y}_n) = \sum_{j=1}^{J} \|y_{jn}\mathbf{z}_n - \mathbf{R}_{.j.}\mathbf{p}_n\|^2 \tag{12}$$

These two least squares criteria can be deduced from (3) and (4) in a manner analogous with the derivation of criterion (9). It is easy to show that COVER errors (11) and (12) are equivalent to criteria (9) and (10) respectively.

It is noteworthy that, first, in the presented study one does not make the assumption that **Z** has full column rank. In this sense the trilinear model is inherently symmetric only for the **x** and **y** orders and does not treat the three orders in an entirely symmetric way. Therefore only four least squares criteria can be derived for the resolution of the trilinear model. Second, the assumption that **X** and **Y** both have full column rank implies $I \geq M$ and $J \geq M$. Therefore under this premise the number of components is upper-bounded by the minimum of I and J.

ACOVER method for three-way data resolution

The rationale of the ACOVER method is to identify the 'true' profiles of the components one at a time. That is, after the 'true' profiles of one component, $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$, have been resolved at a time using the proposed procedure, the three-way array is deflated as

$$\underline{R}_1 = \underline{R} - \mathbf{x}_n \otimes \mathbf{y}_n \otimes \mathbf{z}_n \tag{13}$$

where $\underline{\mathrm{R}}_1$ is the three-way array in which the response of the component previously resolved is annihilated. Then the resolution process is repeated with the deflated array for recovering the remaining components. In the sequel we will present in detail the procedures for resolving the 'true' profiles of one component.

In terms of the COVER errors (9) and (10), two approaches can be developed for the identification of the 'true' profiles of one component. One approach is to estimate $\mathbf{x}_n$, $\mathbf{q}_n$ and $\mathbf{z}_n$ using the minimizers of the COVER error (9); then $\mathbf{y}_n$ is estimated by minimizing the COVER error (10) over $\mathbf{y}_n$ and $\mathbf{p}_n$ for $\mathbf{z}_n$ fixed at the previously estimated value. The other method is to estimate $\mathbf{y}_n$, $\mathbf{p}_n$ and $\mathbf{z}_n$, using the minimizers of the COVER error (10); then $\mathbf{x}_n$ is estimated by minimizing the COVER error (9) over $\mathbf{x}_n$ and $\mathbf{q}_n$ for $\mathbf{z}_n$ fixed at the previously estimated value. Now we separately describe these two approaches.

In principle, the minimizers of the COVER error (9) over $\mathbf{x}_n$, $\mathbf{z}_n$ and $\mathbf{q}_n$ can be approached based on a simple ALS scheme. That is, the minimizers can be exploited by alternately minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{x}_n$ for fixed $\mathbf{q}_n$ and $\mathbf{z}_n$, minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ and $\mathbf{x}_n$ and minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{z}_n$ for fixed $\mathbf{x}_n$ and $\mathbf{q}_n$. However, it was discovered in experiments that this scheme converges relatively slowly. Consequently, in the presented study we will utilize another version of the ALS algorithm to locate the minimizers of $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$. This algorithm is based on alternately minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{x}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ and minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$. Since the COVER error (11) is equivalent to the COVER error (9), the minimization of $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$, equation (9), over $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$ is equivalent to the minimization of $E_{\mathrm{COVER}}(\mathbf{z}_n, \mathbf{q}_n; \mathbf{x}_n)$, equation (11), over $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$.

Therefore the ALS algorithm for minimizing the COVER error (9) can be accomplished by alternately minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$, equation (9), over $\mathbf{x}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ and minimizing $E_{\mathrm{COVER}}(\mathbf{z}_n, \mathbf{q}_n; \mathbf{x}_n)$, equation (11), over $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$. Here it is noteworthy that such a transformation of the alternating minimization scheme is beneficial for the implementation of the algorithm. In fact, one can notice that in (9) and (11) the roles of $\mathbf{x}_n$ and $\mathbf{z}_n$ are just interchanged. Therefore, if the minimizers of (9) over $\mathbf{x}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ can be computed using some procedure, the solutions to $\mathbf{z}_n$ and $\mathbf{q}_n$ minimizing equation (11) for fixed $\mathbf{x}_n$ can also be calculated using the same procedure. This enables one to implement the updating scheme of $\mathbf{x}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ as well as the updating scheme of $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$ using a single program.

It is shown in Appendix I that the solutions to $\mathbf{x}_n$ and $\mathbf{q}_n$ minimizing $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ for fixed $\mathbf{z}_n$ can be computed as follows.

a1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}$$

corresponding to the largest eigenvalue.

a2. Calculate $\mathbf{q}_n$ by

$$\mathbf{q}_n = a\mathbf{U}_N\mathbf{S}_N^{-1/2}\mathbf{w}$$

a3. Calculate $\mathbf{x}_n$ by

$$\mathbf{x}_n = a\sum_{k=1}^{K} z_{kn}\mathbf{R}_{..k}\mathbf{q}_n \Big/ \sum_{k=1}^{K} z_{kn}^2$$

Here $\mathbf{S}_N$ is the $N \times N$ diagonal matrix whose diagonal elements are the first N singular values of $\sum_{k=1}^{K}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$, and $\mathbf{U}_N$ is the $J \times N$ matrix with the nth column being the nth singular vector of $\sum_{k=1}^{K}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$. That is, $\mathbf{U}_N$ and $\mathbf{S}_N$ are given by the truncated singular value decomposition (SVD) of $\sum_{k=1}^{K}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$. Here N is an estimate of the number of components underlying the data. Note that the ACOVER method resolves the profiles in a component-wise way, and the three-way array subjected to subsequent analysis is obtained by deflating the original three-way array $\underline{\mathrm{R}}$ successively using equation (13). Therefore, each time the three-way array is deflated, the number of components present in the obtained three-way array is decreased by one. That is, suppose the estimate of the component number in the original three-way array is *M;* at the time when *m* components have been resolved, the number of components underlying the deflated three-way array is $M - m + 1$. In steps a2 and a3, a is a constant to be determined. As it has been assumed that $\mathbf{x}_n$ is of unit length, this constant can be determined easily. From the computing procedure a1–a3 it can be concluded that the length of $\mathbf{z}_n$ will only affect the lengths of $\mathbf{x}_n$ and $\mathbf{q}_n$ and will not alter the directions of $\mathbf{x}_n$ and $\mathbf{q}_n$. Here the direction of a vector means the normalized vector; for example, the direction of $\mathbf{x}_n$ is $\tilde{\mathbf{x}}_n = \mathbf{x}_n/\|\mathbf{x}_n\|$. Since $\mathbf{x}_n$ is assumed to have unit length, then multiplying $\mathbf{z}_n$ by a scalar will not alter the solution of $\mathbf{x}_n$. Therefore, to calculate $\mathbf{x}_n$, one merely needs to know the normalized $\mathbf{z}_n$, i.e. $\tilde{\mathbf{z}}_n = \mathbf{z}_n/\|\mathbf{z}_n\|$.

Analogously, the computing procedure of $\mathbf{z}_n$ and $\mathbf{q}_n$ minimizing the COVER error (9), or

equivalently the COVER error (11), for fixed $\mathbf{x}_n$ is given as follows.

b1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{i,h=1}^{I} x_{in}x_{hn}\mathbf{R}_{i..}\mathbf{R}_{h..}^{\mathrm{T}}\mathbf{U}_N\mathbf{S}_N^{-1/2}$$

corresponding to the largest eigenvalue.

b2. Calculate $\mathbf{q}_n$ by

$$\mathbf{q}_n = b\mathbf{U}_N\mathbf{S}_N^{-1/2}\mathbf{w}$$

b3. Calculate $\mathbf{z}_n$ by

$$\mathbf{z}_n = b\sum_{i=1}^{I} x_{in}\mathbf{R}_{i..}^{\mathrm{T}}\mathbf{q}_n \Big/ \sum_{i=1}^{I} x_{in}^2$$

Here b is a constant to be determined. Note that in step b1 one has made use of the fact that $\sum_{k=1}^{K}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ has the same SVD as $\sum_{i=1}^{I}\mathbf{R}_{i..}\mathbf{R}_{i..}^{\mathrm{T}}$. Because it has been argued above that the solution of $\mathbf{x}_n$ minimizing the COVER error (9) for fixed $\mathbf{z}_n$ is uncorrelated to the length of $\mathbf{z}_n$, then when one aims at resolving $\mathbf{x}_n$ for fixed $\mathbf{z}_n$, the constant b in steps b2 and b3 can be arbitrarily selected. For simplicity one can choose b such that $\mathbf{z}_n$ is also of unit length.

So far it has been shown that the minimizers of $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{x}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{z}_n$ and the minimizers of $E_{\mathrm{COVER}}(\mathbf{x}_n, \mathbf{q}_n; \mathbf{z}_n)$ over $\mathbf{z}_n$ and $\mathbf{q}_n$ for fixed $\mathbf{x}_n$ can be computed using the procedures a1–a3 and b1–b3 respectively. Therefore the algorithm for minimizing the COVER error (9) can be implemented by alternately updating $\mathbf{x}_n$ and $\mathbf{q}_n$ using the computing procedure a1–a3 and updating $\mathbf{z}_n$ and $\mathbf{q}_n$ using the computing procedure b1–b3, in which the constants a and b are determined to make $\mathbf{x}_n$ and $\mathbf{z}_n$ normalized. Because this algorithm is essentially an ALS algorithm for the minimization problem, it can be concluded that this algorithm tends to decrease the COVER error (9) monotonically during the iterations. (Strictly speaking, the algorithm tends to decrease a rank-reduced version of the COVER error (9).) Therefore, after the algorithm converges, one can obtain the least squares estimates of $\mathbf{x}_n$, $\mathbf{z}_n$ and $\mathbf{q}_n$.

With $\mathbf{x}_n$ and $\mathbf{z}_n$ thus resolved, $\mathbf{y}_n$ can be estimated by the minimizers of the COVER error (10) over $\mathbf{y}_n$ and $\mathbf{p}_n$ for $\mathbf{z}_n$ fixed at the previously estimated value. Analogously, the computing procedure for $\mathbf{y}_n$ and $\mathbf{p}_n$ minimizing the COVER error (10) for fixed $\mathbf{z}_n$ is given as follows.

c1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{D}_N^{-1/2}\mathbf{V}_N^{T}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}\mathbf{R}_{..h}^{\mathrm{T}}\mathbf{V}_N\mathbf{D}_N^{-1/2}$$

corresponding to the largest eigenvalue.

c2. Calculate $\mathbf{p}_n$ by

$$\mathbf{p}_n = c\mathbf{V}_N\mathbf{D}_N^{-1/2}\mathbf{w}$$

c3. Calculate $\mathbf{y}_n$ by

$$\mathbf{y}_n = c \sum_{k=1}^{K} z_{kn} \mathbf{R}_{..k}^{\mathrm{T}} \mathbf{p}_n$$

Here $\mathbf{D}_N$ is the $N \times N$ diagonal matrix whose diagonal elements are the first N singular values of $\sum_{k=1}^{K} \mathbf{R}_{..k} \mathbf{R}_{..k}^{\mathrm{T}}$, $\mathbf{V}_N$ is the $J \times N$ matrix with the nth column being the nth singular vector of $\sum_{k=1}^{K} \mathbf{R}_{..k} \mathbf{R}_{..k}^{\mathrm{T}}$, and c is a constant which can be determined using equation (6); that is,

$$c = 1/(\mathbf{x}_n^{\mathrm{T}} \mathbf{V}_N \mathbf{D}_N^{-1/2} \mathbf{w}) \tag{14}$$

Note that in the aforementioned method, $\mathbf{z}_n$ resolved is normalized, while $\mathbf{y}_n$ is not normalized to unit length. To keep consistency with the scaling convention that $\mathbf{y}_n$ is of unit length, one can simply multiply $\mathbf{z}_n$ by the length of $\mathbf{y}_n$ and then normalize $\mathbf{y}_n$ to unit length.

As presented above, one approach to resolve the 'true' profiles of one component has been developed based on the COVER errors (9) and (10). This method estimates $\mathbf{x}_n$, $\mathbf{z}_n$ and $\mathbf{q}_n$ using the minimizers of the COVER error (9) and then estimates $\mathbf{y}_n$ and $\mathbf{p}_n$ using the minimizers of the COVER error (10) for fixed $\mathbf{z}_n$. Alternatively, another strategy can be proposed by estimating $\mathbf{y}_n$, $\mathbf{p}_n$ and $\mathbf{z}_n$ using the minimizers of the COVER error (10) and then estimating $\mathbf{x}_n$ and $\mathbf{q}_n$ using the minimizers of the COVER error (9) for fixed $\mathbf{z}_n$. Since this approach can be derived in a manner basically consistent with the aforementioned method, we will skip the derivation of this approach and merely outline its computing procedures.

The ALS algorithm for exploiting $\mathbf{y}_n$, $\mathbf{p}_n$ and $\mathbf{z}_n$ which minimize the COVER error (10) is based on alternately minimizing $E_{\mathrm{COVER}}(\mathbf{y}_n, \mathbf{p}_n; \mathbf{z}_n)$ over $\mathbf{y}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{z}_n$ and minimizing $E_{\mathrm{COVER}}(\mathbf{y}_n, \mathbf{p}_n; \mathbf{z}_n)$ over $\mathbf{z}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{y}_n$. The updating procedure of $\mathbf{y}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{z}_n$ is given as follows.

d1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{D}_N^{-1/2} \mathbf{V}_N^{\mathrm{T}} \sum_{k,h=1}^{K} z_{kn} z_{hn} \mathbf{R}_{..k} \mathbf{R}_{..h}^{\mathrm{T}} \mathbf{V}_N \mathbf{D}_N^{-1/2}$$

corresponding to the largest eigenvalue.

d2. Calculate $\mathbf{p}_n$ by

$$\mathbf{p}_n = \mathbf{V}_N \mathbf{D}_N^{-1/2} \mathbf{w}$$

d3. Calculate $\mathbf{y}_n$ by

$$\mathbf{y}_n = \sum_{k=1}^{K} z_{kn} \mathbf{R}_{..k}^{\mathrm{T}} \mathbf{p}_n$$

d4. Normalize $\mathbf{y}_n$ to unit length.

The updating procedure of $\mathbf{z}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{y}_n$ is as follows.

e1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{D}_N^{-1/2} \mathbf{V}_N^{\mathrm{T}} \sum_{j,h=1}^{J} y_{jn} y_{hn} \mathbf{R}_{j.}^{\mathrm{T}} \mathbf{R}_{.h.} \mathbf{V}_N \mathbf{D}_N^{-1/2}$$

corresponding to the largest eigenvalue.

e2. Calculate $\mathbf{p}_n$ by

$$\mathbf{p}_n = \mathbf{V}_N \mathbf{D}_N^{-1/2} \mathbf{w}$$

e3. Calculate $\mathbf{z}_n$ by

$$\mathbf{z}_n = \sum_{j=1}^{J} y_{jn} \mathbf{R}_{.j.} \mathbf{p}_n$$

e4. Normalize $\mathbf{z}_n$ to unit length.

Note that in step e1 one utilizes the fact that $\sum_{k=1}^{K} \mathbf{R}_{..k} \mathbf{R}_{..k}^{\mathrm{T}}$ has the same SVD as $\sum_{j=1}^{J} \mathbf{R}_{j.}^{\mathrm{T}} \mathbf{R}_{j.}$.

The process of updating $\mathbf{y}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{z}_n$ as well as updating $\mathbf{z}_n$ and $\mathbf{p}_n$ for fixed $\mathbf{y}_n$ is continued until a stopping criterion is satisfied. After the algorithm converges, one obtains the estimates of $\mathbf{y}_n$ and $\mathbf{z}_n$. Then $\mathbf{x}_n$ is resolved by the following computing procedure.

f1. Compute the eigenvector $\mathbf{w}$ of the symmetric matrix

$$\mathbf{S}_N^{-1/2} \mathbf{U}_N^{\mathrm{T}} \sum_{k,h=1}^{K} z_{kn} z_{hn} \mathbf{R}_{..k}^{\mathrm{T}} \mathbf{R}_{..h} \mathbf{U}_N \mathbf{S}_N^{-1/2}$$

corresponding to the largest eigenvalue.

f2. Calculate $\mathbf{q}_n$ by

$$\mathbf{q}_n = d \mathbf{U}_N \mathbf{S}_N^{-1/2} \mathbf{w}$$

f3. Calculate $\mathbf{x}_n$ by

$$\mathbf{x}_n = d \sum_{k=1}^{K} z_{kn} \mathbf{R}_{..k} \mathbf{q}_n$$

Here d is a constant which can be determined using equation (5); that is,

$$d = 1/(\mathbf{y}_n^{\mathrm{T}} \mathbf{U}_N \mathbf{S}_N^{-1/2} \mathbf{w}) \tag{15}$$

Finally, to keep consistency with the common scaling convention that $\mathbf{x}_n$ is normalized, one can rescale $\mathbf{z}_n$ by simply multiplying it by the length of $\mathbf{x}_n$ and then normalizing $\mathbf{x}_n$ to unit length.

Thus far two approaches have been developed for resolving the 'true' profiles of one component. These two methods both comprise two steps. In the first step these methods involve an iterative exploitation of the 'true' profiles in two orders of one component using an ALS algorithm. In the second step, based on the minimization of another COVER error, these methods resolve the 'true' profile of the component in the third order using a direct procedure. Because the COVER errors are well defined in terms of the least squares principle, the minimizers of the COVER errors are the least squares estimates of the 'true' profiles of one component, and these two approaches actually give the resolution of the 'true' profiles for one component. Moreover, because these two approaches both give the least squares estimates for the 'true' profiles of one component, one can infer that the profiles of one component resolved by these two approaches will only have a very small difference. In fact, it

is found in simulations that the solutions given by these two approaches only exhibit a very small numerical difference. In this sense it can be concluded that minimization of COVER error (9) tends to minimize COVER error (10) and *vice versa*. On the other hand, since the COVER errors (9) and (10) are symmetrically derived from the trilinear model, a good estimate of $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$ should minimize these two symmetric criteria simultaneously. Based on this consideration, one can combine these two approaches developed above into one method, such that the resulting method can serve as a reduced version of the algorithm minimizing simultaneously criteria (9) and (10). This method is based on alternately updating $\mathbf{x}_n$ for fixed $\mathbf{z}_n$ using the computing procedure a1–a3, updating $\mathbf{z}_n$ for fixed $\mathbf{x}_n$ using the computing procedure b1–b3, updating $\mathbf{y}_n$ for fixed $\mathbf{z}_n$ using the computing procedure d1–d4, and updating $\mathbf{z}_n$ for fixed $\mathbf{y}_n$ using the computing procedure e1–e4. It is discovered in numerous experiments that this method always converges fast and shows improved stability, indicating it can serve as a reduced version of the algorithm minimizing simultaneously criteria (9) and (10). A MATLAB program for implementing this method is given in Appendix II, as a function named after ACOVER. This algorithm can be implemented using a program compactly structured. This algorithm is called the ACOVER algorithm since it resolves the coupled vectors in an alternating way.

Having developed the ACOVER algorithm for resolving the 'true' profiles of one component, one can easily construct a technique for trilinear analysis. The idea of this technique is that it estimates the components one at a time using the ACOVER algorithm, deflates the three-way array using equation (13), decreases the component number N by one and then continues with the deflated array. For simplicity this trilinear resolution technique is called the ACOVER method, as the ACOVER algorithm constitutes the core of the technique. In the ACOVER method, after the profiles of one component are resolved, the response of the component is subtracted from the three-way array, and the resolution of the remaining components is performed on the deflated three-way array in which the contribution of the components previously resolved is annihilated. Therefore the profiles of the components previously resolved will have almost no effect on the resolution of the rest components. This characteristic is attractive because it enables the ACOVER method to avoid the so-called two-factor degeneracy,[23] which is rather difficult to handle for the PARAFAC algorithm.

EXPERIMENTAL

Simulated HPLC–DAD data

A data set measured using a high-performance liquid chromatography (HPLC) system with diode array detection (DAD) on ten samples was simulated. The spectral profiles of four species, $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and $\mathbf{s}_4$, were generated by

Table 1. Compositions of nine mixtures in real HPLC–DAD experiments

	Concentration (μg ml^{-1})								
	1	2	3	4	5	6	7	8	9
p-CT	75·6	0·0	50·4	25·2	12·6	12·6	25·2	50·4	75·6
o-CT	0·0	91·2	30·4	60·8	15·2	152·0	91·2	30·4	60·8
o-DCB	0·0	0·0	0·0	0·0	152·2	15·2	60·8	91·2	30·4
CB	62·4	62·4	62·4	62·4	62·4	62·4	62·4	62·4	62·4

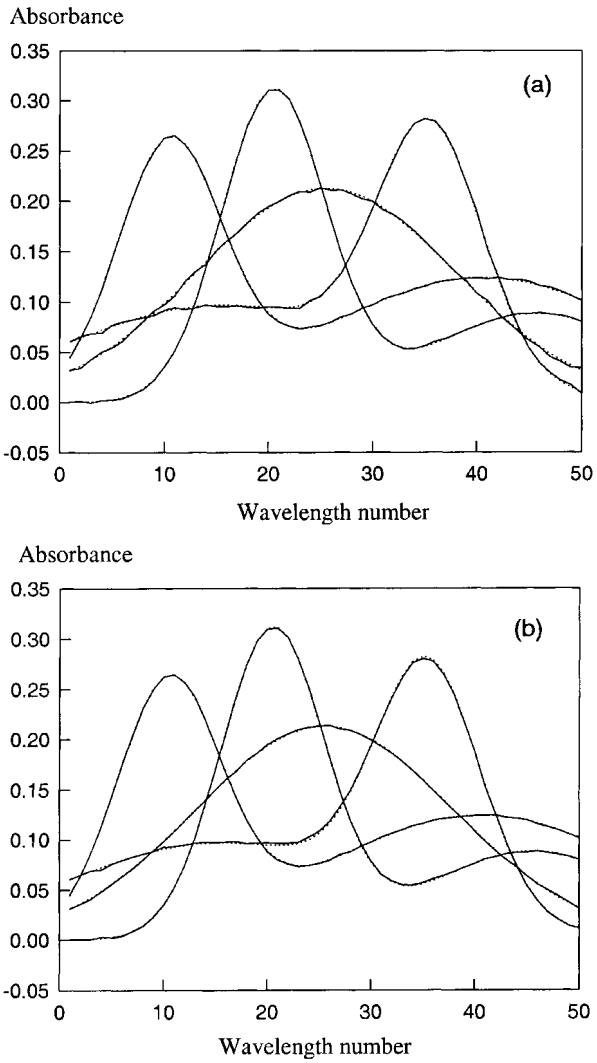

Figure 1. Profiles of simulated HPLC–DAD data resolved by ACOVER and PARAFAC methods. (a) Spectral profiles resolved by ACOVER method (full line) and actual spectral profiles (dotted line) for four components. (b) Spectral profiles resolved by PARAFAC algorithm (full line) and actual spectral profiles (dotted line) for four components. (c) Chromatographic profiles resolved by ACOVER method (full line) and actual chromatographic profiles (dotted line) for four components. (d) Chromatographic profiles resolved by PARAFAC algorithm (full line) and actual chromatographic profiles (dotted line) for four components

$$s_{1,i} = 0{\cdot}2\mathrm{gs}(2i-1,30,30) + 0{\cdot}5\mathrm{gs}(2i-1,70,10), \quad i = 1,2,\ldots,50$$

$$s_{2,i} = 0{\cdot}6\mathrm{gs}(2i-1,20,10) + 0{\cdot}3\mathrm{gs}(2i-1,80,30), \quad i = 1,2,\ldots,50$$

$$s_{3,i} = 0{\cdot}7\mathrm{gs}(2i-1,40,10) + 0{\cdot}2\mathrm{gs}(2i-1,90,20), \quad i = 1,2,\ldots,50$$

$$s_{4,i} = 0{\cdot}7\mathrm{gs}(2i-1,50,25), \quad i = 1,2,\ldots,50$$

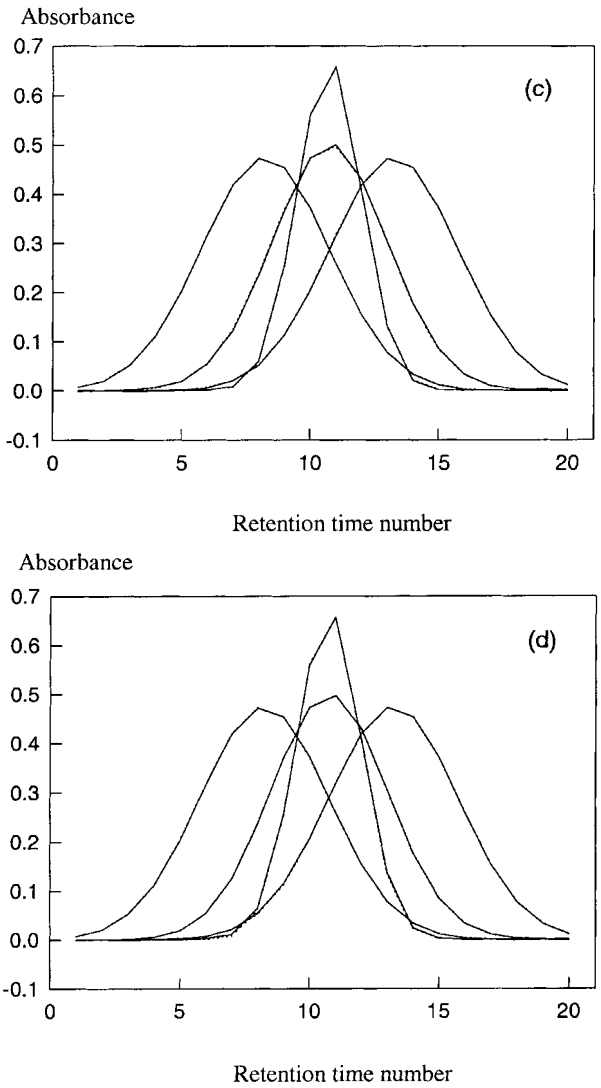

Figure 1. Continued.

where gs(x, a, b) is the value at x of a Gaussian function with centre a and standard deviation b, i.e. $\mathrm{gs}(x, a, b) = \exp[-(x-a)^2/2b^2]$. The chromatographic profiles of the components, $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ and $\mathbf{c}_4$, were simulated by

$$c_{1,i} = 0{\cdot}5\mathrm{gs}(4i-3, 40, 5), \quad i = 1, 2, \ldots, 20$$

$$c_{2,i} = 0{\cdot}5\mathrm{gs}(4i-3, 30, 10), \quad i = 1, 2, \ldots, 20$$

$$c_{3,i} = 0{\cdot}5\mathrm{gs}(4i-3, 50, 10), \quad i = 1, 2, \ldots, 20$$

$$c_{4,i} = 0{\cdot}5\mathrm{gs}(4i-3, 40, 9), \quad i = 1, 2, \ldots, 20$$

Table 2. Normalized concentration profiles of simulated HPLC–DAD data

Component		Normalized concentration values of ten samples									
		1	2	3	4	5	6	7	8	9	10
1	a.	0·1273	0·0274	0·3947	0·3950	0·5435	0·2230	0·3020	0·4832	0·0201	0·0311
	b.	0·1283	0·0269	0·4001	0·3968	0·5460	0·2192	0·2980	0·4789	0·0110	0·0239
	c.	0·1288	0·0250	0·3963	0·3934	0·5415	0·2247	0·3032	0·4833	0·0245	0·0314
2	a.	0·2849	0·3610	0·0041	0·2062	0·0359	0·2245	0·3694	0·3168	0·5004	0·4551
	b.	0·2851	0·3613	0·0032	0·2064	0·0347	0·2249	0·3696	0·3163	0·5004	0·4547
	c.	0·2855	0·3605	0·0036	0·2059	0·0346	0·2245	0·3697	0·3171	0·5006	0·4546
3	a.	0·2857	0·0499	0·3546	0·2256	0·3802	0·4936	0·4133	0·1423	0·0257	0·3991
	b.	0·2870	0·0496	0·3546	0·2254	0·3805	0·4936	0·4140	0·1390	0·0210	0·3988
	c.	0·2862	0·0489	0·3543	0·2262	0·3805	0·4935	0·4139	0·1419	0·0241	0·3983
4	a.	0	0	0	0	0	0·2215	0·4269	0·5105	0·6688	0·2465
	b.	0·0003	−0·0002	0·0012	0·0022	0·0035	0·2226	0·4255	0·5126	0·6673	0·2474
	c.	−0·0054	0·0008	−0·0081	−0·0063	−0·0086	0·2162	0·4232	0·5096	0·6739	0·2451

[a.] Actual concentration profiles.
[b.] Concentration profiles resolved by the ACOVER method.
[c.] Concentration profiles resolved by the PARAFAC algorithm.

The first five simulated samples contained only the first three species, the concentrations of which are uniformly distributed in the range 0–1. The remaining five samples contained all four components, with the concentrations of each component uniformly distributed in the range 0–1. The three-way response array was generated exactly in terms of (1), in which the random errors were normally distributed with mean zero and standard deviation 0·002. The data array was treated using the proposed method as well as the PARAFAC algorithm for comparison.

Real HPLC–DAD data

Nine mixtures of *p*-chlorotoluene (*p*-CT), *o*-chlorotoluene (*o*-CT) and *o*-dichlorobenzene (*o*-DCB) as well as an internal retention time standard, chlorobenzene (CB), were analysed using an HPLC system hyphenated with DAD. The concentrations of each component are shown in Table 1. Details of the experimental procedures are as given previously.[11] The data array ($50 \times 24 \times 9$) collected was treated using the developed method as well as the PARAFAC algorithm.

All computer programs were written in MATLAB and run on a personal computer (Pentium Processor 233 MHz). The programs for the ACOVER method are given in Appendix II. The stopping criterion for the ACOVER algorithm is that the differences in the entries of $\mathbf{z}_n$ between consecutive iterations are all less than 10^{-5}, or the total computational epoch, set to 200 in the study, is reached. The PARAFAC algorithm used in the study is the version given by Krijnen.[21] The stopping criterion for the PARAFAC algorithm is that the improvement in the PARAFAC error between consecutive iterations is less than 10^{-5}, or the total computational epoch is greater than a predefined maximum, set to 500 in the simulated experiments and 1000 in the real data analysis.

RESULTS AND DISCUSSION

Simulated HPLC–DAD data

Considering the core of the ACOVER method is an iterative procedure, firstly the convergence behaviour of the algorithm was investigated using the simulated data. This was performed by running

Table 3. Average distances between profiles resolved in ten runs and their average ones obtained using ACOVER method and PARAFAC algorithm for simulated HPLC–DAD data

	ACOVER			PARAFAC		
	X[a]	Y[a]	Z[b]	X[a]	Y[a]	Z[a]
Component 1	0·7892	0·3160	0·1900	0·2609	0·1646	0·4012
Component 2	0·2715	0·1989	0·0082	0·0087	0·0028	0·0011
Component 3	0·1113	0·0893	0·0171	0·0026	0·0287	0·0175
Component 4	0·5918	0·2040	0·0920	0·0471	0·0348	0·7246

[a] The distances below are all multiplied by 10^4.
[b] The distances below are all multiplied by 10^3.

the algorithm separately ten times. In the algorithm the number of components is chosen to be four, which is the true dimensionality of the underlying model. At each time the algorithm was started from initial estimates of concentrations, which took values randomly distributed in the domain [0, 1]. After the algorithm converged, it was found that the solutions obtained in ten runs exhibited a very small difference. The average iteration numbers in ten runs were 8·9, 6·4, 5·4 and 2·8 respectively for the four components. The profiles resolved in ten runs were averaged. The average profiles in spectral and chromatographic orders are depicted against the actual ones in Figures 1(a) and 1(c), and the average concentration profiles are shown in Table 2. One can observe that the average profiles show a very small discrepancy from the actual ones. In Table 3 the average Euclidean distances of the profiles computed in ten runs to their average ones are listed. These figures are all acceptably small, indicating that the algorithm has the desirable convergence precision. For comparison the above investigation was also conducted for the PARAFAC algorithm. In the algorithm the dimensionality of the model was also chosen to be four. It was found in the investigation that in ten runs the PARAFAC algorithm got trapped in meaningless solutions twice. With these two runs excluded, the average iteration cycle of the remaining eight runs was 360. The profiles resolved in the eight runs were averaged. The average profiles in spectral and chromatographic orders are plotted in Figures 1(b) and 1(d), and the concentration profiles are shown in Table 2. It can been seen that the average profiles also exhibit a slight divergence from the actual ones. These results indicate that the ACOVER method works as well as the PARAFAC algorithm in cases where the data are generated by an ideal trilinear model and the dimensionality is correctly chosen for the model, but the convergence rate of the ACOVER method is much higher than that of the PARAFAC algorithm, and the convergence is more stable with respect to the randomized initialization for the ACOVER algorithm than for the PARAFAC algorithm.

Another goal of the simulated experiments was to examine the effect of the component number on the solutions given by the ACOVER method. With different numbers of components selected for the model, the simulated data were analysed using the ACOVER method. In the investigation the initial concentration estimates from which the algorithm was started were determined by the first principal component of the third order, as used in the presented algorithm. The resolution errors, defined as the Euclidean distances between the resolved and the actual profiles, for component one are plotted versus the component number in Figure 2. It can be observed that the distances between the resolved and the actual profiles are undesirably large in the case where the component number is chosen to be smaller than the true dimensionality of the model, and the performance is stabilized when the number of components is increased to be equal to or greater than the actual dimensionality. These findings were also observed for the other components. These results imply that the ACOVER method is very stable to overestimates of the model dimensionality. Therefore with the ACOVER method one need

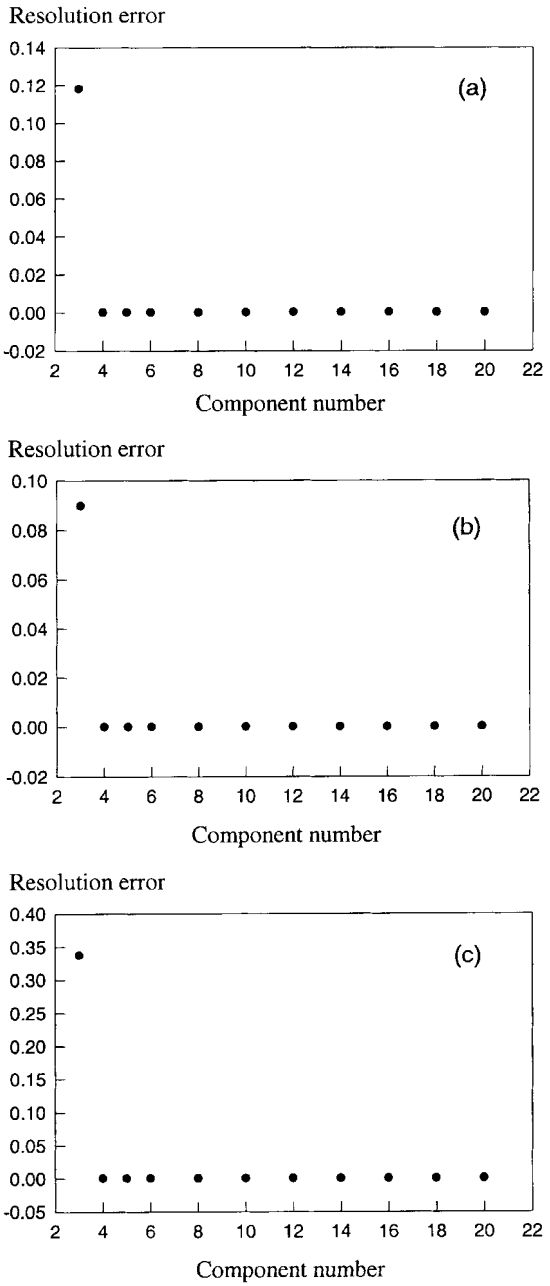

Figure 2. Relationship between component number and resolution errors of profiles of first component in simulated HPLC–DAD data. The resolution error of a profile is the Euclidean distance between the resolved and the actual profile. (a) Resolution error of spectral profile. (b) Resolution error of chromatographic profile. (c) Resolution error of concentration profile

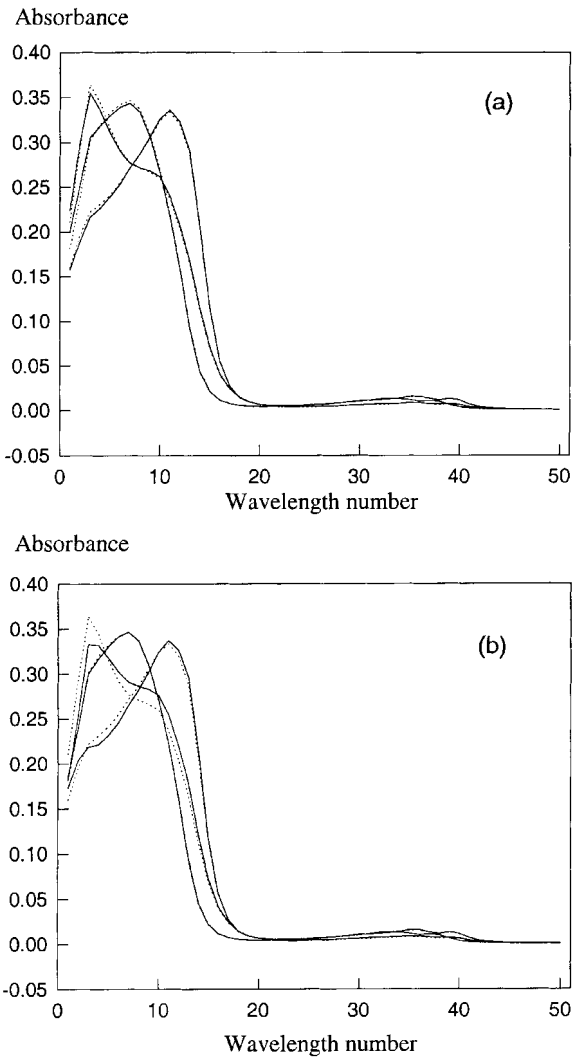

Figure 3. Spectral profiles of real HPLC–DAD data resolved by ACOVER and PARAFAC methods. (a) Spectral profiles resolved by ACOVER method (full line) and spectral profiles measured experimentally (dotted line) for *p*-CT, *o*-CT and *o*-DCB. (b) Spectral profiles resolved by PARAFAC algorithm (full line) and spectral profiles measured experimentally (dotted line) for *p*-CT, *o*-CT and *o*-DCB

not determine the component number accurately. One need only estimate an upper bound for the component number. In the extreme one can simply take the smaller of I and J as the estimate of the component number. This conclusion is appealing from the practical point of view, since it circumvents the problem of determining the component number before resolution, which is rather hard to handle for the PARAFAC algorithm.

Real HPLC–DAD data

In the investigation the data array was treated as a black system and the goal of analysis is to resolve

Table 4. Normalized concentration profiles of real HPLC–DAD data

Component		Normalized concentration values of nine samples								
		1	2	3	4	5	6	7	8	9
p-CT	a.	0·5620	0	0·3746	0·1873	0·0937	0·0937	0·1873	0·3746	0·5620
	b.	0·5717	0·0029	0·3821	0·1925	0·0751	0·1051	0·1621	0·3591	0·5641
	c.	0·5841	−0·0028	0·3885	0·1929	0·0467	0·0788	0·1314	0·3609	0·5610
o-CT	a.	0	0·4111	0·1370	0·2741	0·0685	0·6852	0·4111	0·1370	0·2741
	b.	0·0051	0·4345	0·1482	0·2914	0·0700	0·6387	0·4314	0·1277	0·2985
	c.	0·0076	0·4363	0·1505	0·2934	0·0490	0·6416	0·4230	0·1265	0·3031
DCB	a.	0	0	0	0	0·7985	0·0797	0·3190	0·4785	0·1595
	b.	0·0153	0·0013	0·0107	0·0060	0·7242	0·0911	0·3682	0·5410	0·1966
	c.	0·0423	0·0206	0·0351	0·0278	0·7272	0·1279	0·3853	0·5020	0·2239

[a.] Actual concentration profiles.
[b.] Concentration profiles resolved by the ACOVER method.
[c.] Concentration profiles resolved by the PARAFAC algorithm.

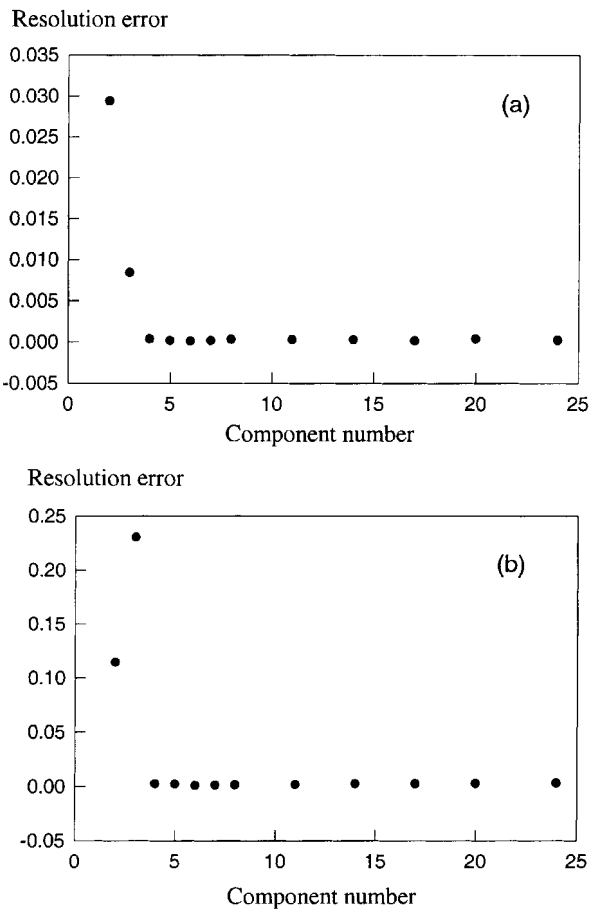

Figure 4. Relationship between component number and resolution errors of profiles of *p*-CT in real HPLC–DAD data. The resolution error of a profile is the Euclidean distance between the resolved and the actual profile. (a) Resolution error of spectral profile. (b) Resolution error of concentration profile

the profiles in each order of *p*-CT, *o*-CT and *o*-DCB. For validating the resolution results, spectra were measured using DAD on pure compounds of these three species. The data were analysed using the developed ACOVER method as well as the PARAFAC algorithm. In the PARAFAC algorithm the starting values of $\mathbf{X}$ and $\mathbf{Y}$ are $\mathbf{U}_N$ and $\mathbf{V}_N$ respectively. This starting configuration is commonly used in the practice of the algorithm. Different component numbers were chosen for these two algorithms. It was found that with a component number greater than four or less than three the profiles resolved by the PARAFAC algorithm deviated greatly from those experimentally measured or prepared. The best resolution for the PARAFAC algorithm was achieved in the case where the model dimensionality was chosen to be three. It took 774 iterations for the PARAFAC algorithm to achieve the resolution. These resolved spectral profiles are depicted against those measured experimentally in Figure 3(b), and the concentration profiles computed are given in Table 4. The profiles in the second order are not given, because no references are available to validate the performance. In contrast, in the investigation one still observes the stability of the ACOVER method with respect to overestimates of the component number. As can be seen in Figure 4, the resolution error of the ACOVER method is stabilized when the model dimensionality is greater than four. The spectral profiles resolved by the ACOVER method with the model dimensionality set to five are plotted in Figure 3(a) and the concentration profiles are shown in Table 4. It can be seen that the resolution accuracy of the ACOVER method is much better than that given by the PARAFAC algorithm. In practical problem solving, data are frequently confounded by some instrumental drift, which causes the data to deviate to some degree from the ideal trilinear model. The above results indicate that the ACOVER method is more stable to model deviations than the PARAFAC algorithm. In the resolution the computational cycles for the ACOVER method were eleven, nine and three respectively for the three components. These results reconfirm the conclusion that the ACOVER method has a much higher convergence rate than the PARAFAC algorithm.

CONCLUSIONS

A new method, alternating coupled vectors resolution (ACOVER), has been developed for trilinear analysis of three-way data array. The presented results have shown that the developed ACOVER method can give resolution with acceptable accuracy for the component profiles, and the performance of the method is very stable with respect to overestimates of the component number. This method provides a valuable tool for second-order calibration and for the study of complex chemical systems or processes which can be characterized by a three-way data array. Though the theoretical nature of the ACOVER method calls for further investigation, its idea will still provide an extra dimension in the development of novel trilinear resolution procedures.

ACKNOWLEDGEMENTS

The work is supported by the National Natural Science Foundation of China (29735150). The authors thank Professor P. J. Gemperline for thoroughly examining the algorithm and debugging the MATLAB program. The insightful comments of Professor P. J. Gemperline and two anonymous reviewers were helpful for our improving the presentation of the method.

APPENDIX I: DERIVATION OF THE COMPUTING PROCEDURE FOR THE SOLUTIONS TO $\mathbf{x}_n$ AND $\mathbf{q}_n$ MINIMIZING THE COVER ERROR (9) FOR FIXED $\mathbf{z}_n$

If $\mathbf{x}_n$ and $\mathbf{q}_n$ minimize the COVER error (9) for fixed $\mathbf{z}_n$, it is necessary for them to satisfy the conditions

$$\frac{\partial E_{\mathrm{COVER}}}{\partial \mathbf{x}_n} = 2\sum_{k=1}^{K} z_{kn}(z_{kn}\mathbf{x}_n - \mathbf{R}_{..k}\mathbf{q}_n) = 0 \tag{16}$$

$$\frac{\partial E_{\mathrm{COVER}}}{\partial \mathbf{q}_n} = -2\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}(z_{kn}\mathbf{x}_n - \mathbf{R}_{..k}\mathbf{q}_n) = 0 \tag{17}$$

One can derive from (16) and (17) that

$$\frac{\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}}{\sum_{k=1}^{K} z_{kn}^2}\mathbf{q}_n = \sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{q}_n \tag{18}$$

$$\mathbf{x}_n = \sum_{k=1}^{K} z_{kn}\mathbf{R}_{..k}\mathbf{q}_n \Big/ \sum_{k=1}^{K} z_{kn}^2 \tag{19}$$

Note that the matrices $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$) and $\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ are chemically rank-deficient, and their rank is intrinsically the number of components underlying the data. This results in ill-conditioning of the eigenproblem (18). Therefore rank-reduced approximations to $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$) and $\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ should be used for combating ill-conditioned solutions. Let the singular value decomposition (SVD) of $\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ be

$$\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k} = \mathbf{U}\mathbf{S}\mathbf{U}^{\mathrm{T}}$$

where $\mathbf{U}$ and $\mathbf{S}$ are both $J \times J$ matrices. These matrices are truncated by removing the corresponding right-hand columns and the bottom rows to give $\mathbf{U}_N$ ($J \times N$) and $\mathbf{S}_N$ ($N \times N$). The rank-reduced approximation of $\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ is thus given by the truncated SVD

$$\sum_{k=1}^{K} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k} = \mathbf{U}_N\mathbf{S}_N\mathbf{U}_N^{\mathrm{T}} \tag{20}$$

Because $\mathbf{U}_N$ spans the common subspace of the rows of $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$), the rank-reduced approximations to $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$) are

$$\mathbf{R}_{..k} = \mathbf{R}_{..k}\mathbf{U}_N\mathbf{U}_N^{\mathrm{T}} \tag{21}$$

Substituting equation (20) and (21) into (18) and letting

$$\mathbf{w} = \mathbf{S}_N^{1/2}\mathbf{U}_N^{\mathrm{T}}\mathbf{q}_n$$

one can obtain

$$\frac{\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}}{\sum_{k=1}^{K} z_{kn}^2}\mathbf{w} = \mathbf{w} \tag{22}$$

Now it can be concluded that $\mathbf{w}$ is the eigenvector of the symmetric matrix

$$\frac{\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}}{\sum_{k=1}^{K} z_{kn}^2}$$

with the associated eigenvalue of one. It is shown in Appendix III that all eigenvalues of the symmetric matrix are not greater than one. Therefore $\mathbf{w}$ is the eigenvector of the symmetric matrix corresponding to the largest eigenvalue. Since $\sum_{k=1}^{K} z_{kn}^2$ is a scalar quantity, $\mathbf{w}$ is also the eigenvector of the matrix

$$\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}$$

corresponding to the largest eigenvalue. One can easily solve the eigenproblem and reach the solution of $\mathbf{w}$. Then $\mathbf{q}_n$ can be calculated by

$$\mathbf{q}_n = a\mathbf{U}_N\mathbf{S}_N^{-1/2}\mathbf{w} \tag{23}$$

where a is a constant to be determined. Substitution of (23) into (19) yields

$$\mathbf{x}_n = a\sum_{k=1}^{K} z_{kn}\mathbf{R}_{..k}\mathbf{U}_N\mathbf{S}_N^{-1/2}\mathbf{w}\Big/\sum_{k=1}^{K} z_{kn}^2 \tag{24}$$

Therefore one obtains the computing procedure for the solutions to $\mathbf{x}_n$ and $\mathbf{q}_n$ minimizing the COVER error (9) for fixed $\mathbf{z}_n$.

Note that in deriving the algorithm, one does not impose the constraint (5) on $\mathbf{q}_n$. As is shown from the computing results, not imposing this constraint does not induce the resolved profiles to deviate from the actual ones. This indicates that the constraint (5) is mathematically of trivial significance in the ACOVER method.

APPENDIX II: PROGRAM WRITTEN IN MATLAB FOR ACOVER METHOD

```
%NOTATION:
  % rrzk is the kth slice of R along z order.
  % rrxi is the ith slice of R along x order.
  % rryj is the jth slice of R along y order.
  % uuNy and ssNsqy are U and S^1/2 respectively.
  % uuNx and ssNsqx are V and D^1/2 respectively.
```



```
% Xcal, Ycal and Zcal are the calculated profile matrices in x, y and z orders respectively.
% xn, yn and zn are the profiles of the nth component in x, y and z orders respectively.
% II, JJ and KK are the dimensionalities of xn, yn and zn respectively.
% NN is the number of components chosen for the model.
% MM is the number of components to be resolved.
% rrzr = [rrz1 rrz2 … rrzKK];
% rrzl = [rrz1′ rrz2′ … rrzKK′].
% rryl = [rry1′ rry2′ … rryJJ′].
% rrxl = [rrx1′ rrx2′ … rrxII′].
% rrxr = [rrx1 rrx2 … rrxII].

function [Xcal, Ycal, Zcal] = main(rrzr, rrzl, II, JJ, KK, NN, MM)
Xcal = zeros(II, MM);
Ycal = zeros(JJ, MM);
Zcal = zeros(KK, MM);

for nn = 1 : MM
[Xcal(:, nn), Ycal(:, nn), Zcal(:, nn)] = ACOVER(rrzl, rrzr, II, JJ, KK, NN);

rrznl = []; rrznr = [];
for kk = 1 : KK
rrzn = Zcal(kk, nn) * Xcal(:, nn) * Ycal(:, nn)′;
rrznl = [rrznl rrzn′];
rrznr = [rrznr rrzn];
end

rrzl = rrzl - rrznl;
rrzr = rrzr - rrznr;
NN = NN - 1;
end

function [xn, yn, zn] = ACOEVR(rrzl, rrzr, II, JJ, KK, NN)
rryl = reshape(rrzl′, II, KK * JJ);
rrxl = reshape(rryl′, KK, JJ * II);
rrxr = reshape(rrzr′, JJ, KK * II);

[uu, ss, vv] = svd(rrzl, 0);
uuNy = uu(:, 1 : NN);
ssNsqy = ss(1 : NN, 1 : NN);
[uu, ss, vv] = svd(rrzr, 0);
uuNx = uu(:, 1 : NN);
ssNsqx = ss(1 : NN, 1 : NN);
[uu, ss, vv] = svd(rrx1, 0);
zn = uu(:, 1);

dzn = 1; cyc = 0;
while dzn > le−5
zn0 = zn; cyc = cyc + 1

[qn, tt] = absolve(rrzl, II, KK, zn, uuNy, ssNsqy);
xn = tt′ * qn;
xn = xn / sqrt(xn′ * xn);
```



```
[qn, tt] = absolve(rrxr, KK, II, xn, uuNy, ssNsqy);
zn = tt′ * qn;
zn = zn / sqrt(zn′ * zn);

[qn, tt] = absolve(rrzr, JJ, KK, zn, uuNx, ssNsqx);
yn = tt′ * qn;
yn = yn / sqrt(yn′ * yn);

[qn, tt] = absolve(rryl, KK, JJ, yn, uuNx, ssNsqx);
qn = qn / (qn′ * xn);
zn = tt′ * qn;

dzn = max(abs(zn − zn0));
end

function [qn, tt] = absolve(rrl, II, KK, zn, uuN, ssNsq)
rrleye = [];
for kk = 1 : KK
rrleye = [rrleye, eye(II) * zn(kk)];
end

tt = rrl * rrleye′;
ttt = inv(ssNsq) * uuN′ * tt * tt′ * uuN * inv(ssNsq);
[ff, gg, hh] = svd(ttt);
alpha = ff(:, 1);

qn = uuN * inv(ssNsq) * alpha;
```

APPENDIX III: PROOF THAT THE EIGENVALUES OF THE SYMMETRIC MATRIX IN (22) ARE NOT LARGER THAN ONE

To prove the proposition, it is sufficient to show that for any vector α,

$$\alpha^{\mathrm{T}}\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha \Big/ \sum_{k=1}^{K} z_{kn}^2 \leq \alpha^{\mathrm{T}}\alpha$$

Let $\mathbf{A} = (\mathbf{R}_{..1}^{\mathrm{T}}, \ldots, \mathbf{R}_{..K}^{\mathrm{T}})$ and $\mathbf{B} = (z_{1n}\mathbf{I}, z_{2n}\mathbf{I}, \ldots, z_{Kn}\mathbf{I})^{\mathrm{T}}$, where $\mathbf{I}$ is the $I \times I$ identity matrix; then one has

$$\alpha^{\mathrm{T}}\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha \Big/ \sum_{k=1}^{K} z_{kn}^2$$

$$= \alpha^{\mathrm{T}}\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\mathbf{A}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha \Big/ \sum_{k=1}^{K} z_{kn}^2$$

$$= \mathrm{Trace}\left(\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha\alpha^{\mathrm{T}}\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\mathbf{A} \Big/ \sum_{k=1}^{K} z_{kn}^2\right)$$

Letting

$$\beta = \mathbf{A}^{\mathrm{T}}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha$$

and noticing

$$(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1} = 1\Big/\sum_{k=1}^{K} z_{kn}^2$$

one obtains

$$\alpha^{\mathrm{T}}\mathbf{S}_N^{-1/2}\mathbf{U}_N^{\mathrm{T}}\sum_{k,h=1}^{K} z_{kn}z_{hn}\mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..h}\mathbf{U}_N\mathbf{S}_N^{-1/2}\alpha\Big/\sum_{k=1}^{K} z_{kn}^2$$

$$= \mathrm{Trace}[\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\beta\beta^{\mathrm{T}}]$$

$$= \beta^{\mathrm{T}}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\beta$$

Since $\mathbf{B}\,(\mathbf{B}^{\mathrm{T}}\,\mathbf{B})^{-1}\,\mathbf{B}^{\mathrm{T}}$ is a projection matrix, one has

$$\beta^{\mathrm{T}}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\beta \leq \beta^{\mathrm{T}}\beta$$

It is easy to show that $\beta^{\mathrm{T}}\,\beta = \alpha^{\mathrm{T}}\,\alpha$, therefore one finishes the proof.

REFERENCES

1. C.-N. Ho, G. D. Christian and E. R. Davidson, *Anal. Chem.* **50**, 1108 (1978).
2. A. Lorber, *Anal. Chim. Acta*, **164**, 293 (1984).
3. S. Wold, P. Geladi, K. Esbensen and J. Ohman, *Chemometrics Intell. Lab. Syst.* **4**, 29 (1989).
4. P. Geladi, *Chemometrics Intell. Lab. Syst.* **7**, 11 (1989).
5. Y. Zeng and P. K. Hopke, *Chemometrics Intell. Lab. Syst.* **7**, 237 (1989).
6. R. Tauler, B. R. Kowalski and S. Fleming, *Anal. Chem.* **65**, 2040 (1993).
7. A. K. Smilde, R. Tauler, J. M. Henshaw, L. W. Burgess and B. R. Kowalski, *Anal. Chem.* **66**, 3345 (1994).
8. K. S. Booksh, J. M. Henshaw, L. W. Burgess and B. R. Kowalski, *J. Chemometrics*, **9**, 263 (1995).
9. R. Bro, *J. Chemometrics*, **10**, 47 (1996).
10. A. K. Smilde, *J. Chemometrics*, **11**, 307 (1997).
11. H.-L. Wu, M. Shibukawa and K. Oguma, *J. Chemometrics*, **12**, 1 (1998).
12. E. Sanchez and B. R. Kowalski, *Anal. Chem.* **58**, 499 (1986).
13. B. Wilson, E. Sanchez and B. R. Kowalski, *J. Chemometrics*, **3**, 493 (1989).
14. S. Li, C. Hamilton and P. Gemperline, *Anal. Chem.* **64**, 599 (1992).
15. E. Sanchez and B. R. Kowalski, *J. Chemometrics*, **4**, 29 (1990).
16. K. S. Booksh, Z. Lin, Z. Wang and B. R. Kowalski, *Anal. Chem.* **66**, 2561 (1994).
17. P. Paatero, *Chemometrics Intell. Lab. Syst.* **38**, 223 (1997).
18. J. D. Carroll and J. J. Chang, *Psychometrica*, **35**, 283, (1970).
19. R. A. Harshman, *UCLA Working Papers Phonet.* **16**, 1 (1970).
20. H. A. L. Kiers and W. P. Krijnen, *Psychometrica*, **56**, 147, (1991).
21. W. P. Krijnen, *The Analysis of Three-way Arrays by Constrained PARAFAC Methods*, DSWO Press, Leiden (1993).
22. R. Bro, *Chemometrics Intell. Lab. Syst.* **38**, 149 (1997).
23. W. S. Rayens and B. C. Mitchell, *Chemometrics Intell. Lab. Syst.* **38**, 173 (1997).
24. J. B. Kruskal, *Linear Alg. Appl.* **18**, 95 (1977).
25. H. A. L. Kiers and A. K. Smilde, *J. Chemometrics*, **9**, 179 (1995).

