JOURNAL OF CHEMOMETRICS
J. Chemometrics 2000; **14**: 15–36

Three-way data resolution by alternating slice-wise diagonalization (ASD) method

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 approach, the alternating slice-wise diagonalization (ASD) method, is developed for three-way data resolution. First, based on the least squares principle and the constraints inherent in the resolution of the trilinear model, a criterion, the slice-wise diagonalization (SD) loss, is proposed for trilinear analysis of three-way data. This criterion provides a natural way to avoid the two-factor degeneracy, which is difficult to handle for the PARAFAC algorithm. Second, by alternatingly minimizing the SD loss, a procedure is developed for identifying the parameters of the trilinear model. Experimental results show that the resolved profiles of chemical meaning are very stable with respect to the component number provided that the number is chosen to be equal to or greater than the actual one. This enables the ASD method to achieve resolution without concern about the actual component number. This approach is different from the traditional ones, since the determination of the actual component number is a critical step for conventional chemometric resolution techniques. Moreover, the convergence rate of the algorithm for the ASD method is much higher than that of the PARAFAC algorithm. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: three-way data resolution; alternating slice-wise diagonalization (ASD); PARAFAC; trilinear model; slice-wise diagonalization loss

INTRODUCTION

Identification of the components of interest in complex mixtures is a challenging problem in analytical chemistry. The common practice to attack the problem is to resort to some physical and chemical separation techniques. This solution, however, is always time-consuming and cost-expensive. Moreover, when there exists a chemical equilibrium in the mixtures, the equilibrium will be perturbed by the separation procedure, resulting in misleading quantification of the components of analytical interest. An alternative way to solve the problem is given by chemometric three-way data resolution methodologies. Under a certain condition, from three-way data experimentally available, these methods can extract the information which is adequate for qualitative and quantitative analysis of the components under consideration.

There are a variety of approaches to three-way data resolution in chemometrics. Among them the family of rank annihilation factor analysis (RAFA) methods plays a dominant role in chemometric

* Correspondence to: R.-Q. Yu, College of Chemistry and Chemical Engineering, Hunan University, Changsha 410082, People's Republic of China.

Contract/grant sponsor: National Natural Science Foundation of China; Contract/grant number: 29735150

CCC 0886–9383/2000/010015–22 $17.50
Received 10 May 1999
Accepted 12 October 1999

practice [1–12]. RAFA was originated by the work of Ho and co-workers [1–2]. Their method requires, however, the two-way response of the pure species under quantification to be known in advance. imposing a severe restriction on its applications. A most versatile and algebraically elegant formulation of RAFA, known as the generalized rank annihilation method (GRAM), was proposed by Sanchez and Kowalski [8–10]. Unfortunately, GRAM ignores the measurement errors in its formulation, which induces its solutions to have inflated variance and produce imaginary entries. Some efforts have been made to improve the performance of GRAM [11,12]. There is another avenue to three-way data resolution. A prominent example is the PARAFAC algorithm [13–28], as proposed by Harshman [14] and first used in analytical chemistry by Appellof and Davidson [15]. This algorithm is in essence a least squares method to estimate the parameters of a trilinear model, which yields the best fit to the three-way data array. Unfortunately, this algorithm, when trapped in computational swamps [18], will give chemically meaningless solutions. Additionally, the solutions finally acquired using this algorithm are rather unstable with respect to the number of components chosen for the model, creating a dilemma that is hard to handle in practical problem solving.

In this paper a new method, alternating slice-wise diagonalization (ASD), is developed for trilinear analysis of three-way data. Unlike existing approaches, the idea of the ASD method is to find two matrices such that the matrix slices along some order, when multiplied by these two matrices at two sides, are all fitted to diagonal form in the least squares sense. This formulation provides a natural way to circumvent the so-called two-factor degeneracy (2-FD) problem [18,19], which is difficult to combat for the PARAFAC algorithm. A salient virtue of the ASD method is that the resolved profiles of chemical meaning are very stable with respect to the model dimensionality when the dimensionality is chosen to be equal to or greater than the actual number of components. This enables the ASD method to achieve a resolution without concern about the actual component number. This approach is different from the traditional ones, since the determination of the actual component number is a critical step for conventional chemometric resolution techniques. In addition, the algorithm for the ASD method has a much higher convergence rate than the PARAFAC algorithm. Results of a simulated example and a real data set are presented to demonstrate the performance.

THEORY

Slice-wise diagonalization loss for trilinear resolution

Suppose that the data measured in a chemical process or experiment are collected in an $I \times J \times K$ three-way array $\underline{\mathrm{R}}$, which is generated by the trilinear model [19]

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

where $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$ are the profiles in three orders respectively of the nth component ($n = 1, 2, \ldots, N$), $\otimes$ denotes the tensor product and $\underline{\mathrm{E}}$ is the array of measurement errors. In general chemical practice one can assume that the profile matrices $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ have full column rank. To guarantee that the profiles of the components of interest are resolved uniquely, it can be further assumed that in the third order the profile of each component under concern is linearly independent of that of any other component. Note that this premise, which is stronger than Kruskal's conditions [20], is the case most generally considered in the chemometrics literature [21]. In matrix notation the trilinear model can be written as

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

where $\mathbf{R}_{..k}$ and $\mathbf{E}_{..k}$ are the kth slices of $\underline{\mathrm{R}}$ and $\underline{\mathrm{E}}$ respectively along the third order, the superscript T symbolizes the transpose of a vector or a matrix and diag($\mathbf{z}_{(k)}$) denotes the $N \times N$ diagonal matrix whose diagonal elements are the corresponding ones of $\mathbf{z}_{(k)}$. Here $\mathbf{z}_{(k)}$ is the kth row of the profile matrix $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$. The goal of trilinear resolution is to identify the profile matrices **X, Y** and **Z** given the data array $\underline{\mathrm{R}}$ experimentally measured.

It is important to note that, first, in apparent form the trilinear model, equation (1), treats its parameter matrices **X, Y** and **Z** in a symmetric way. However, in the case under consideration it is merely assumed that **X** and **Y** have full column rank, while **Z** does not necessarily have 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. Second, equation (1) only means that the data can be decomposed into several trilinear terms, not implying that the species all give trilinear responses. Therefore trilinear methods can not only be applied to data in which the species all exhibit ideal trilinear behaviour, but can also be used for data whose trilinear decomposition yields components of significant chemical meaning. In particular, one can treat partially trilinear data, in which the species of interest exhibit trilinear behaviour, despite the background components not behaving in a trilinear way, using trilinear methods. Third, the assumption that **X** and **Y** both have full column rank implies that $I \geq N$ and $J \geq N$. That is, under this premise the number of components is upper bounded by the minimum of I and J.

As it has been assumed that **X** and **Y** both have full column rank, it is known that there exist two matrices **P** ($l \times N$) and **Q** ($J \times N$), which belong to the subspaces of **X** and **Y** respectively, satisfying

$$\mathbf{P}^{\mathrm{T}}\mathbf{X} = \mathbf{1}_N \tag{3}$$

$$\mathbf{Y}^{\mathrm{T}}\mathbf{Q} = \mathbf{1}_N \tag{4}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix. Multiplying $\mathbf{R}_{..k}$ by $\mathbf{P}^{\mathrm{T}}$ at the left and by **Q** at the right and utilizing equations (2)–(4), one can obtain

$$\mathbf{P}^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{Q} = \mathrm{diag}(\mathbf{z}_{(k)}) + \tilde{\mathbf{E}}_{..k}, \quad k = 1, 2, \ldots, K \tag{5}$$

where $\tilde{\mathbf{E}}_{..k} = \mathbf{P}^{\mathrm{T}}\mathbf{E}_{..k}\mathbf{Q}$ ($k = 1, 2, \ldots, K$). To keep the values of the entries of $\tilde{\mathbf{E}}_{..k}$ at the level of measurement noises, one can assume that **P** and **Q** are column-wise normalized. That is, $\|\mathbf{p}_n\| = 1$ and $\|\mathbf{q}_n\| = 1$ ($n = 1, 2, \ldots, N$), where $\mathbf{p}_n$ and $\mathbf{q}_n$ are the nth columns of **P** and **Q** respectively and $\|\mathbf{a}\|$ designates the Euclidean norm of a vector **a**. With this scaling convention, $\tilde{\mathbf{E}}_{..k}$ s can be regarded as matrices of measurement errors. Therefore a statistically plausible approach to identify the parameter matrices **P, Q** and **Z** in equation (5) is the least squares method. That is, **P, Q** and **Z** are estimated by the minimizers of the criterion

$$E_{\mathrm{SD}}(\mathbf{P}, \mathbf{Q}, \mathbf{Z}) = \sum_{k=1}^{K} \|\mathbf{P}^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{Q} - \mathrm{diag}(\mathbf{z}_{(k)})\|^2 \tag{6}$$

where $\|\mathbf{A}\|$ denotes the Frobenius norm of a matrix **A**, i.e. $\|\mathbf{A}\|^2 = \mathrm{trace}\ (\mathbf{A}^{\mathrm{T}}\ \mathbf{A})$. Here trace(.) symbolizes the trace of a matrix. E_{SD} is called the slice-wise diagonalization (SD) error, as it measures the degree of discrepancy to which each slice matrix $\mathbf{R}_{..k}$ is fitted to diagonal form. The value of this least squares criterion, equation (6), depends on three parameter matrices **P, Q** and **Z**, from which the profile matrices **X, Y** and **Z** can be resolved directly. Consequently, minimization of this criterion (6) over **P, Q** and **Z** will actually yield the least squares solutions to the profile matrices to be identified. Note that in the formulation of this minimization problem one overlooks the constraints implied by equations (3) and (4), which are inherent in the resolution of the trilinear model (2). It is found in extensive simulations that, when the component number is correctly determined for the model, the parameters of the trilinear model can be resolved accurately even without concern about these constraints; when the model dimensionality is wrongly estimated, however, the algorithm for the minimization problem sometimes gets trapped in chemically meaningless solutions. Based on these findings, a new method has been developed for trilinear resolution. The idea of this method is to seek **P, Q** and **Z** minimizing the SD error (6) subjected to the constraints of equations (3) and (4). Since this method alternatingly exploits the matrices **P, Q** and **Z** such that all slices $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$) are fitted to diagonal form, it is called the alternating slice-wise diagnolization (ASD) method.

To solve the minimization problem with constraints, the penalty function method can be used to convert it into a problem without constraints. That is, one seeks **P, Q, Z, X** and **Y** minimizing the loss function

$$L_{\mathrm{SD}}(\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{P},\mathbf{Q}) = \sum_{k=1}^{K} \|\mathbf{P}^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{Q} - \mathrm{diag}(\mathbf{z}_k)\|^2 + \lambda\|\mathbf{P}^{\mathrm{T}}\mathbf{X} - \mathbf{I}_N\|^2 + \lambda\|\mathbf{Y}^{\mathrm{T}}\mathbf{Q} - \mathbf{I}_N\|^2 \qquad (7)$$

where λ is a predefined positive constant which determines the weights of the penalty terms $\|\mathbf{P}^{\mathrm{T}}\mathbf{X} - \mathbf{I}_N\|^2$ and $\|\mathbf{Y}^{\mathrm{T}}\mathbf{Q} - \mathbf{I}_N\|^2$. For simplicity this loss function is called the slice-wise diagonalization (SD) loss, since its core is the SD error. Minimization of this loss function over **P, Q, X, Y** and **Z** yields the resolution of the trilinear model (2).

Dimensionality reduction of SD loss

In the present study, instead of minimizing directly the SD loss (7), we first use some dimensionality reduction technique to transform the minimization problem into a reduced one.

Suppose $\mathbf{U}_x$ and $\mathbf{U}_y$ are the sets of orthonormal bases spanning the common column and row subspaces respectively of $\mathbf{R}_{..k}$ ($k = 1, 2, \ldots, K$). From equation (2) it is known that $\mathbf{R}_{..k}$s are all chemically rank-deficient, and the dimensionalities of the common column and row subspaces of $\mathbf{R}_{..k}$s are both intrinsically equal to N, the number of components underlying the data. Consequently, $\mathbf{U}_x$ and $\mathbf{U}_y$ are $l \times N$ and $J \times N$ matrices respectively. In common chemometric practice, $\mathbf{U}_x$ is estimated by the first N singular vectors of $\sum_{\mathrm{k=1}}^{\mathrm{K}} \mathbf{R}_{..k}\mathbf{R}_{..k}^{\mathrm{T}}$, and $\mathbf{U}_y$ is estimated by the first N singular vectors of $\sum_{\mathrm{k=1}}^{\mathrm{K}} \mathbf{R}_{..k}^{\mathrm{T}}\mathbf{R}_{..k}$ [9]. Because **X** and **Y** belong to the common column and row subspaces respectively one can estimate **X** and **Y** by

$$\mathbf{X} = \mathbf{U}_x\mathbf{A} \qquad (8)$$

$$\mathbf{Y} = \mathbf{U}_y\mathbf{B} \qquad (9)$$

where **A** and **B** are two $N \times N$ matrices which define the transformations from $\mathbf{U}_x$ to **X** and from $\mathbf{U}_y$ to **Y** respectively. Since **P** and **Q** belong to the subspaces of **X** and **Y** respectively, they can be estimated by

$$\mathbf{P} = \mathbf{U}_x\mathbf{G} \qquad (10)$$

$$\mathbf{Q} = \mathbf{U}_y\mathbf{H} \qquad (11)$$

where **G** and **H** are also two $N \times N$ matrices which define the transformations from $\mathbf{U}_x$ to **P** and from $\mathbf{U}_y$ to **Q** respectively. Substitution of equations (8)–(11) into equations (3) and (4) yields

$$\mathbf{A}^{\mathrm{T}}\mathbf{G} = \mathbf{I}_N \tag{12}$$

$$\mathbf{B}^{\mathrm{T}}\mathbf{H} = \mathbf{I}_N \tag{13}$$

These two equations give the dimensionality-reduced version of the constraints of equations (3) and (4). Noticing that **A, B, G** and **H** are square matrices, one obtains that **G** and **H** are actually the inverse matrices of $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{B}^{\mathrm{T}}$ respectively. Substituting equations (8)–(11) into equation (7) and letting

$$\tilde{\mathbf{R}}_{..k} = \mathbf{U}_x^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{U}_y \tag{14}$$

one obtains the loss function

$$L_{\mathrm{SD}}(\mathbf{A},\mathbf{B},\mathbf{Z},\mathbf{G},\mathbf{H}) = \sum_{k=1}^{K} \|\mathbf{G}^{\mathrm{T}}\tilde{\mathbf{R}}_{..k}\mathbf{H} - \mathrm{diag}(\mathbf{z}_{(k)})\|^2 + \lambda\|\mathbf{G}^{\mathrm{T}}\mathbf{A} - \mathbf{I}_N\|^2 + \lambda\|\mathbf{B}^{\mathrm{T}}\mathbf{H} - \mathbf{I}_N\|^2 \tag{15}$$

This loss function is the dimensionality-reduced version of the SD loss (7). As **X, Y** and **Z** can be estimated by the minimizers of the SD loss (7), minimization of this reduced SD loss over **A, B, Z, G** and **H** will yield the estimates of **A, B** and **Z**. Subsequently, from equations (8) and (9) one can achieve the resolution of the profile matrices **X, Y** and **Z**.

Alternating slice-wise diagonalization (ASD) method for trilinear resolution

In the preceding subsection it has been shown that in the ASD method the trilinear resolution of a three-way array is transformed to finding a set of parameter matrices **A, B, G, H** and **Z** minimizing the reduced SD loss (15). To exploit the solution minimizing this loss function, a procedure can be designed based on the alternating least squares (ALS) algorithm. This procedure is implemented by alternatingly minimizing L_{SD}(**A, B, G, H, Z**) over **Z** for fixed **A, B, G** and **H**, minimizing L_{SD}(**A, B, G, H, Z**) over **A** for fixed **B, G, H** and **Z**, minimizing L_{SD}(**A, B, G, H, Z**) over **B** for fixed **A, G, H** and **Z**, minimizing L_{SD}(**A, B, G, H, Z**) over **G** for fixed **A, B, H** and **Z**, and minimizing L_{SD}(**A, B, G, H, Z**) over **H** for fixed **A, B, G** and **Z**. Here the derivation of the updating equations for these parameter matrices is outlined.

The necessary condition for **G** minimizing L_{SD}(**A, B, G, H, Z**) for fixed **A, B, H** and **Z** is that

$$\frac{\partial L_{\mathrm{SD}}}{\partial \mathbf{G}} = 2\sum_{k=1}^{K}\tilde{\mathbf{R}}_{..k}\mathbf{H}[\mathbf{H}^{\mathrm{T}}\tilde{\mathbf{R}}_{..k}^{\mathrm{T}}\mathbf{G} - \mathrm{diag}(\mathbf{z}_{(k)})] + 2\lambda\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{G} - \mathbf{I}) = 0 \tag{16}$$

One can derive from equation (16) that

$$\mathbf{G} = \left(\sum_{k=1}^{K}\tilde{\mathbf{R}}_{..k}\mathbf{H}\mathbf{H}^{\mathrm{T}}\tilde{\mathbf{R}}_{..k}^{\mathrm{T}} + \lambda\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}\left(\sum_{k=1}^{K}\tilde{\mathbf{R}}_{..k}\mathbf{H}\mathrm{diag}(\mathbf{z}_{(k)}) + \lambda\mathbf{A}\right) \tag{17}$$

Equation (17) gives the updating equation of **G** for fixed **A, B, H** and **Z**. Analogously, the computing equation of **H** for fixed **A, B, G** and **Z** can be obtained as

$$\mathbf{H} = \left(\sum_{k=1}^{K} \tilde{\mathbf{R}}_{..k}^{\mathrm{T}} \mathbf{G}\mathbf{G}^{\mathrm{T}} \tilde{\mathbf{R}}_{..k} + \lambda \mathbf{B}\mathbf{B}^{\mathrm{T}} \right)^{-1} \left(\sum_{k=1}^{K} \tilde{\mathbf{R}}_{..k}^{\mathrm{T}} \mathbf{G} \mathrm{diag}(\mathbf{z}_{(k)}) + \lambda \mathbf{B} \right) \quad (18)$$

If **Z** minimizes L_{SD}(**A, B, G, H, Z**) for fixed **A, B, G** and **H**, it is necessary for **Z** to satisfy the condition

$$\frac{\partial L_{\mathrm{SD}}}{\partial z_{kn}} = -2(\mathbf{g}_n^{\mathrm{T}} \tilde{\mathbf{R}}_{..k} \mathbf{h}_n - z_{kn}) = 0 \quad (19)$$

where $\mathbf{g}_n$ and $\mathbf{h}_n$ are the nth columns of **G** and **H** respectively. Then one has the updating equation of **Z** as

$$z_{kn} = \mathbf{g}_n^{\mathrm{T}} \tilde{\mathbf{R}}_{..k} \mathbf{h}_n, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots N \quad (20)$$

The necessary condition for **A** minimizing L_{SD}(**A, B, G, H, Z**) for fixed **B, G, H** and **Z** is that

$$\frac{\partial L_{\mathrm{SD}}}{\partial \mathbf{A}} = 2\lambda \mathbf{G}(\mathbf{G}^{\mathrm{T}}\mathbf{A} - \mathbf{I}_N) = 0 \quad (21)$$

Noticing that **G** is a square matrix, one can obtain the updating equation of **A** as

$$\mathbf{A} = (\mathbf{G}^{-1})^{\mathrm{T}} \quad (22)$$

Analogously, the updating equation of **B** for fixed **A, G, H** and **Z** is given by

$$\mathbf{B} = (\mathbf{H}^{-1})^{\mathrm{T}} \quad (23)$$

Having presented the updating equations for the parameter matrices, we can give the general algorithm for the ASD method as follows:

1. Estimate $\mathbf{U}_x$ and $\mathbf{U}_y$ and compute $\tilde{\mathbf{R}}_{..k}$ using equation (14).
2. Initialize **G** and **H** to be invertible matrices which are column-wise normalized and set λ to a small positive number.
3. Compute **Z** using equation (20).
4. Compute **A** and **B** using equations (22) and (23) respectively.
5. Compute **G** using equation (17) and subsequently scale **G** to be column-wise normalized.
6. Compute **Z** using equation (20).
7. Compute **H** using equation (18) and subsequently scale **H** to be column-wise normalized.
8. Repeat steps 3–7 until a stopping criterion is satisfied.

This procedure is virtually an ALS algorithm for minimizing the SD loss (15). It is important to point out that in the algorithm we follow the scaling convention that **G** and **H** are column-wise normalized. Because $\mathbf{U}_x$ and $\mathbf{U}_y$ are both column-wise orthonormal, one obtains from equations (10) and (11) that **P** and **Q** are also column-wise normalized. However, in common chemometric practice it is generally assumed that **X** and **Y** are column-wise normalized. Therefore, after the algorithm converges, one must rescale **X** and **Y** to be column-wise normalized so as to keep consistency with the common scaling convention. That is,

$$\mathbf{x}_n = \mathbf{U}_x\mathbf{a}_n/\|\mathbf{a}_n\| \tag{24}$$

$$\mathbf{y}_n = \mathbf{U}_y\mathbf{b}_n/\|\mathbf{b}_n\| \tag{25}$$

where $\mathbf{a}_n$ and $\mathbf{b}_n$ are the nth columns of $\mathbf{A}$ and $\mathbf{B}$ respectively. As a consequence, $\mathbf{z}_n$ given by the algorithm must also be rescaled by a factor of $\|\mathbf{a}_n\|\ \|\mathbf{b}_n\|$. Nevertheless, it is found in experiments that it is preferable to re-estimate $\mathbf{Z}$ using least squares regression of model (2) with $\mathbf{X}$ and $\mathbf{Y}$ thus resolved. That is, $\mathbf{Z}$ is estimated by

$$\mathbf{Z} = \mathbf{F}[(\mathbf{X}^{\mathrm{T}}\mathbf{X}) * (\mathbf{Y}^{\mathrm{T}}\mathbf{Y})]^{-1} \tag{26}$$

where $\mathbf{F}$ is a $K \times N$ matrix with entries given by

$$f_{kn} = \mathbf{x}_n^{\mathrm{T}}\mathbf{R}_{..k}\mathbf{y}_n, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, N \tag{27}$$

and $*$ signifies the element-wise (Hadamard) product of the matrices.

Thus far we have shown the ASD method for recovering the profile matrices from a given three-way array. Notice that, in model (2), postmultiplying $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ by diagonal matrices whose diagonal elements are 1 or -1 with the product being 1 does not alter the trilinear decomposition. That is, after the profile matrices have been identified, the signs of the profiles are still left undetermined. Therefore one needs to establish some sign convention to remove such a source of non-identifiability. In spectroscopic applications, since the physical profiles are non-negative, one can employ the sign convention that in each column of $\mathbf{X}$ and $\mathbf{Y}$ the element with the largest absolute value is positive. If several entries in a column have absolute values equal to the largest one, then the first such entry is positive. This sign convention defines the sign of the profiles in $\mathbf{X}$ and $\mathbf{Y}$, leaving the sign of the profiles in $\mathbf{Z}$ determined. To keep consistency with this sign convention, after the profile matrices have been identified, the following postprocessing is performed:

$$\mathbf{x}_{n,\mathrm{final}} = \mathbf{x}_{n,\mathrm{initial}}\ \mathrm{sign}(\mathbf{x}_{n,\mathrm{initial}}(MI_x)), \qquad n = 1, 2, \ldots, N \tag{28}$$

$$\mathbf{y}_{n,\mathrm{final}} = \mathbf{y}_{n,\mathrm{initial}}\ \mathrm{sign}(\mathbf{y}_{n,\mathrm{initial}}(MI_y)), \qquad n = 1, 2, \ldots, N \tag{29}$$

$$\mathbf{z}_{n,\mathrm{final}} = \mathbf{z}_{n,\mathrm{initial}}\ \mathrm{sign}(\mathbf{x}_{n,\mathrm{initial}}(MI_x))\ \mathrm{sign}(\mathbf{y}_{n,\mathrm{initial}}(MI_y)), \quad n = 1, 2, \ldots, N \tag{30}$$

where $\mathbf{x}_{n,\mathrm{initial}}$, $\mathbf{y}_{n,\mathrm{initial}}$ and $\mathbf{z}_{n,\mathrm{initial}}$ are the profiles initially resolved in x, y and z orders respectively by the ASD method, sign(x) designates the sign of x, MI_x and MI_y are the indices of the elements with the largest absolute values in $\mathbf{x}_{n,\mathrm{initial}}$ and $\mathbf{y}_{n,\mathrm{initial}}$ respectively and $\mathbf{x}_{n,\mathrm{final}}$, $\mathbf{y}_{n,\mathrm{final}}$ and $\mathbf{z}_{n,\mathrm{final}}$ are the profiles finally given in x, y and z orders respectively by the method. In the setting of spectroscopic applications, after this postprocessing, one can immediately obtain the 'true' physical profiles of each component in three orders.

It is important to note that, when two of the columns are collinear simultaneously in $\mathbf{P}$ and $\mathbf{Q}$, the SD error (6) will generally have a large value. Moreover, the penalty terms in the SD loss (7) cause the resolved $\mathbf{X}$ and $\mathbf{Y}$ to have full column rank. That is, there will be no collinearity between any two columns in $\mathbf{X}$ as well as in $\mathbf{Y}$. Therefore the ASD method is capable of circumventing the so-called 2-FD, which is 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 four samples of four species was simulated. The spectral profiles of these four species, $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and $\mathbf{s}_4$, were generated by

$$\begin{aligned}
s_{1,i} &= 0.2\ \mathrm{gs}(2i-1,30,30)+0.5\ \mathrm{gs}(2i-1,70,10), \quad && 1,2,\ldots,50\\
s_{2,i} &= 0.6\ \mathrm{gs}(2i-1,20,10)+0.3\ \mathrm{gs}(2i-1,80,30), \quad && 1,2,\ldots,50\\
s_{3,i} &= 0.7\ \mathrm{gs}(2i-1,40,20)+0.2\ \mathrm{gs}(2i-1,90,20), \quad && 1,2,\ldots,50\\
s_{4,i} &= 0.7\ \mathrm{gs}(2i-1,50,25), \quad && 1,2,\ldots,50
\end{aligned}$$

where gs(*x*, *a*, *b*) is the value at *x* of the 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

$$\begin{aligned}
c_{1,i} &= 0.5\ \mathrm{gs}(4i-3,40,5), \quad && i=1,2,\ldots,20\\
c_{2,i} &= 0.5\ \mathrm{gs}(4i-3,30,10), \quad && i=1,2,\ldots,20\\
c_{3,i} &= 0.5\ \mathrm{gs}(4i-3,40,10), \quad && i=1,2,\ldots,20\\
c_{4,i} &= 0.5\ \mathrm{gs}(4i-3,40,5), \quad && i=1,2,\ldots,20
\end{aligned}$$

The first two of the four simulated samples contained only the first three species, the concentrations of which are uniformly distributed in the range 0–1. The remaining two 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 equation (2), in which the random errors were normally distributed with mean zero and standard deviation 0·002. The data array was treated using the ASD method as well as the PARAFAC algorithm to resolve the profiles of each component in three orders.

Real excitation–emission fluorescence data

Acridine red, fluorescein sodium and rhodamine B are three fluorescent dyes coexisting in a dye laser. The steady state fluorescence of six samples of these three dyes was measured in 0·01 M NaOH solutions. The concentrations of these three species in the six samples are shown in Table I. The fluorescence was measured on a Hitachi 850 florescence spectrometer. The excitation and emission wavelengths were set from 450 to 600 nm and from 480 to 620 nm respectively, with a fixed interval of 5 nm. The scan rate was 240 nm min^{-1}. The effect of Rayleigh scattering on each response matrix was eliminated by subtracting the measurement matrix of a blank from the sample measurements. A $31 \times 29 \times 6$ three-way data array was thereby collected. This array was treated using the ASD method as well as the PARAFAC algorithm for recovering the profiles of each component in three orders.

All computer programs were written in MATLAB and run on a personal computer (Pentium Processor 233 MHz). The programs for the ASD method are given in the Appendix. The stopping criterion for the ASD algorithm was that the difference in the values of the SD loss (15) between

Table I. Compositions of six samples in real fluorescence measurement experiments.

	Concentration (μg g^{-1})					
Component	#1	#2	#3	#4	#5	#6
Acridine	0·00	0·00	0·00	0·00	0·24	0·12
Fluorescein	0·12	0·00	0·12	0·24	0·12	0·24
Rhodamine	0·00	0·11	0·22	0·11	0·22	0·22

consecutive iterations was less than a predefined constant, set to 10^{-10} in the simulations and 10^{-5} in real data analysis, or the total computational iterations were greater than 2000. The PARAFAC algorithm used in the study was the version given by Krijnen [22]. The stopping criterion for the PARAFAC algorithm was that the improvement in the PARAFAC error between consecutive iterations was less than a predefined constant, also set to 10^{-10} in the simulations and 10^{-5} in real data analysis, or the maximum computational epoch of 10000 was expired. In the PARAFAC algorithm the starting values of **X** and **Y** were $\mathbf{U}_x$ and U_y respectively. This starting configuration is commonly used in the implication of this algorithm.

RESULTS AND DISCUSSION

Simulated HPLC-DAD data

In the ASD method two parameters need to be determined preliminarily. One is the penalty weight λ and the other is the model dimensionality, i.e. the component number *N*. Because in the ASD algorithm the updates of **A** and **B** always keep the constraints (12) and (13) fulfilled, the value of λ will introduce no bias to the profiles finally recovered. Nevertheless, the value of λ has some effect on the convergence of the ASD algorithm. If λ is too large, the algorithm converges very slowly. In contrast, if λ is too small, as the iterative process proceeds, the matrices **G** and **H** are inclined to become singular and the iteration has to be interrupted owing to the computation of inverses of nearly singular matrices. This phenomenon induces the algorithm not to converge. However, it offers an indication for the determination of the value of λ. That is, one can start the algorithm with a relatively small value of λ. If the iterative process produces nearly singular matrices for **G** and **H**, one can break the iterations and restart the algorithm with λ ten times as great as the previous one. Using such a procedure, one can determine an appropriate value of λ with which the iteration process does not produce nearly singular matrices for **G** and **H** and converges sufficiently fast. It was discovered in simulations that $\lambda = 10^{-3}$ was a good initial guess for exploiting the proper values of λ.

For the PARAFAC method the component number is the parameter of dominant significance. If the component number is wrongly determined, the performance of PARAFAC generally collapses, creating a dilemma that is difficult to handle in practical problem solving. For comparison the effect of the component number on the solutions given by ASD was examined using the simulated data. With different numbers of components chosen for the model, the simulated data were analysed using the ASD method. In the investigation the initial estimates of **G** and **H** were both set to the identity matrices. The resolution errors, which are defined as the Euclidean distances between the resolved profiles and the actual ones, for the first species are plotted versus the component numbers in Figure 1. It can be seen that the resolution errors in three orders are undesirably large in cases where the component number determined is smaller than the actual number of components present in the samples, and the errors are stabilized to acceptable values as the component number is increased to be equal to or greater than the true model dimensionality. These findings were also observed for the other three species. These results imply that the profiles of chemical meaning recovered by the ASD

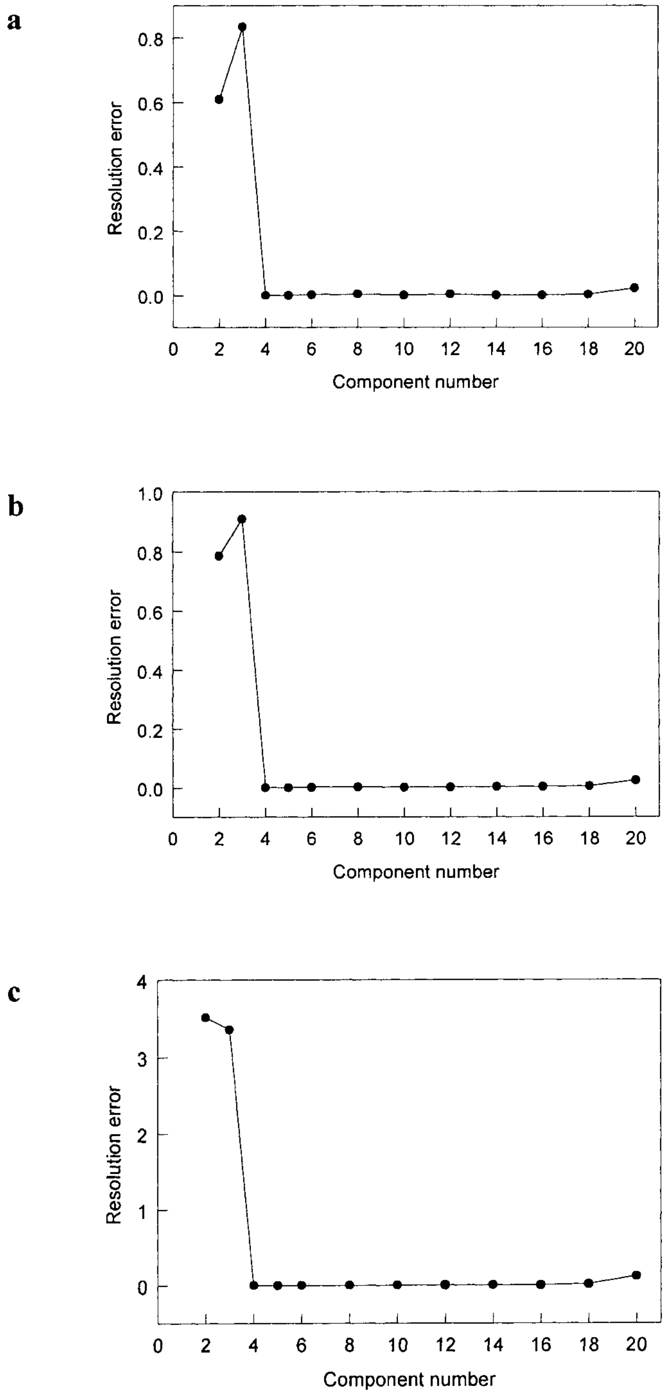


Figure 1. Relationship between component number and resolution errors of profiles of first species in simulated HPLC-DAD data for ASD method. The resolution error of a profile is the Euclidean distance between the resolved and the actual profile. (a) Resolution error of spectral profile. (b) Resolution error of chromatographic profile. (c) Resolution error of concentration profile.

Table II. Mean squared deviations (MSDs) from profiles resolved in ten runs to average ones and to actual ones. The deviation between two profiles is the Euclidean distance between them.

	MSDs to average profiles			MSDs to actual profiles		
Component	*x* order	*y* order	*z* order	*x* order	*y* order	*z* order
1	$1{\cdot}2 \times 10^{-13}$	$3{\cdot}3 \times 10^{-11}$	$3{\cdot}6 \times 10^{-10}$	$3{\cdot}3 \times 10^{-5}$	$4{\cdot}6 \times 10^{-5}$	$1{\cdot}2 \times 10^{-3}$
2	$4{\cdot}2 \times 10^{-9}$	$1{\cdot}1 \times 10^{-9}$	$3{\cdot}2 \times 10^{-7}$	$6{\cdot}5 \times 10^{-5}$	$1{\cdot}1 \times 10^{-4}$	$7{\cdot}4 \times 10^{-4}$
3	$2{\cdot}5 \times 10^{-13}$	$1{\cdot}0 \times 10^{-12}$	$2{\cdot}7 \times 10^{-10}$	$5{\cdot}2 \times 10^{-5}$	$5{\cdot}0 \times 10^{-5}$	$6{\cdot}6 \times 10^{-4}$
4	$1{\cdot}2 \times 10^{-8}$	$6{\cdot}1 \times 10^{-8}$	$3{\cdot}1 \times 10^{-7}$	$7{\cdot}0 \times 10^{-4}$	$5{\cdot}2 \times 10^{-4}$	$1{\cdot}8 \times 10^{-3}$

method are very stable to overestimates of the component number. Therefore with the ASD method one does not need to determine the component number accurately. One merely needs to estimate an upper bound for the component number. In the extreme one can simply take the smaller of *I* and *J* as the estimate. This conclusion is appealing from the practical point of view, since it indicates that the difficulty of determining a proper number of components for the model is circumvented in the ASD method. This offers a salient advantage over the PARAFAC algorithm.

The ASD algorithm is an iterative procedure calling for starting values. That is, to start the algorithm, one needs to determine the initial estimates for **G** and **H**. In the simulated experiments the effect of the choice of these initial estimates on the convergence behaviour of this algorithm was investigated. This was performed by running the algorithm separately ten times. Since the performance of ASD was very stable to overestimates of the model dimensionality, for simplicity the component number was chosen to be four, which was the true dimensionality of the model, throughout this investigation. In each run the algorithm was started from initial estimates of **G** and **H** which took values randomly distributed in the domain $[-1, 1]$. It was discovered in the experiment that in each run the algorithm converged very fast, and the solutions given in different runs are almost the same. The average iteration number for ten runs was 46·6. (In this study, ASD required 10 494 floating point operations (FLOPs) per iteration.) This was much less than that of PARAFAC, as given below. The profiles resolved in ten runs were averaged. The Euclidean distances from the resolved profiles to the average ones as well as to the actual ones are listed in Table II. These figures verified that the solutions recovered using ASD in ten runs all provided accurate estimates for the actual profiles. Moreover, the profiles resolved in different runs exhibited very slight discrepancies. This gives evidence for the fact that the ASD algorithm had desirable convergence precision. Based on this finding, one can conclude that for the ASD method the initial estimates of **G** and **H** have little effect on the solution finally recovered. Therefore one can arbitrarily determine the initial estimates for **G** and **H** to start the algorithm. For simplicity in subsequent studies the initial estimates of **G** and **H** were both set to the identity matrices.

Finally, the simulated data were analysed using the ASD method as well as the PARAFAC algorithm. It was found with components less or more than four that some profiles recovered by PARAFAC deviated severely from the actual ones. The best resolution for PARAFAC was achieved in the case where the component number was chosen to be four, i.e. the actual number of components present in the four simulated mixtures. It took 4109 iterations for PARAFAC to achieve the resolution. (In the analysis, more than 170000 FLOPs were required for one PARAFAC iteration.) These resolved profiles in the spectral and chromatographic orders are depicted against the actual ones in Figure 2, and the concentration profiles computed are given in Table III. It can be seen that these profiles gave a very good fit to the actual ones. In contrast, when the component number was chosen to be equal to or greater than four, the profiles of chemical meaning were all accurately recovered by the ASD method. In the case where the component number was chosen to be five, the

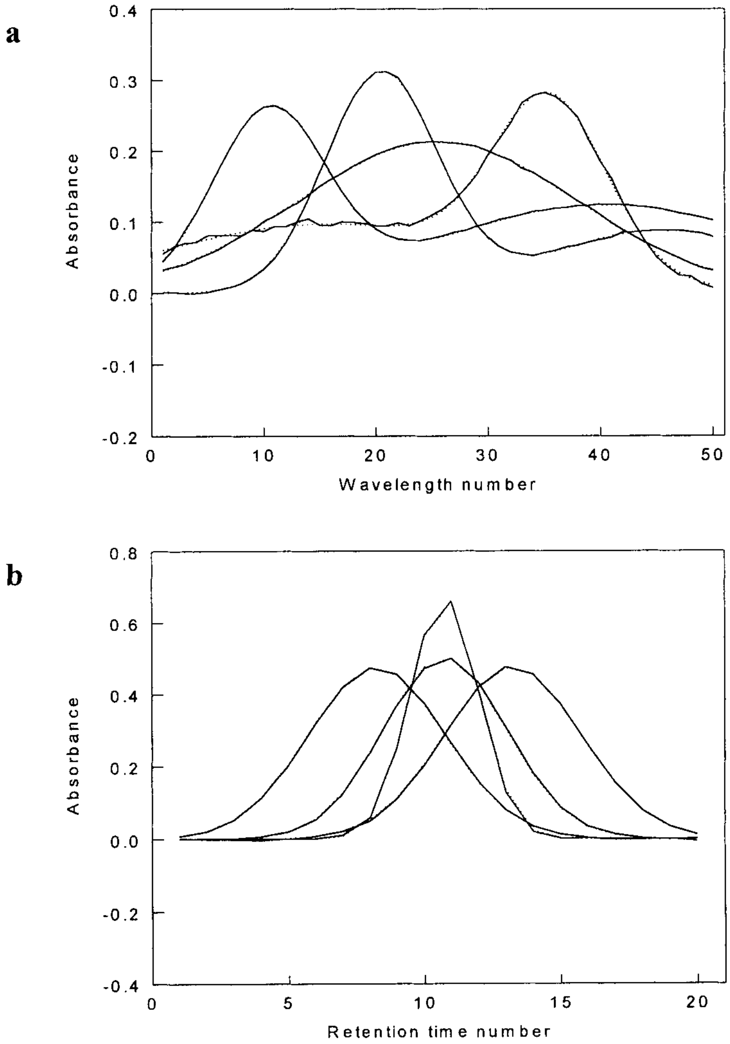


Figure 2. Profiles of simulated HPLC-DAD data resolved by ASD and PARAFAC methods. The component numbers for PARAFAC and ASD were four and five respectively. One noise component was extracted by ASD, since the actual component number was four. (a) Spectral profiles resolved by PARAFAC algorithm (full line) against actual ones (dotted line). (b) Chromatographic profiles resolved by PARAFAC algorithm (full line) against actual ones (dotted line). (c) Spectral profiles resolved by ASD method (full line) against actual ones (dotted line). (d) Chromatographic profiles resolved by ASD method (full line) against actual ones (dotted line).

resolved profiles are also shown in Figure 2 and Table III. Notice that in Table III the concentration profiles resolved by ASD have several negative elements. These negative values have resulted from the propagation of model errors, since a standard alternating least squares algorithm in which non-negativity is not imposed is used for the ASD algorithm. It took 319 iterations for ASD to obtain such resolution, indicating that ASD had a much higher convergence rate than PARAFAC. (For the five-component model, ASD required 19 879 FLOPs per iteration.) One can observe that the profiles of chemical meaning resolved by ASD (Figures 2c and 2d) fit the actual ones as well as those obtained using PARAFAC (Figures 2a and 2b). Because in the resolution the model dimensionality was set to

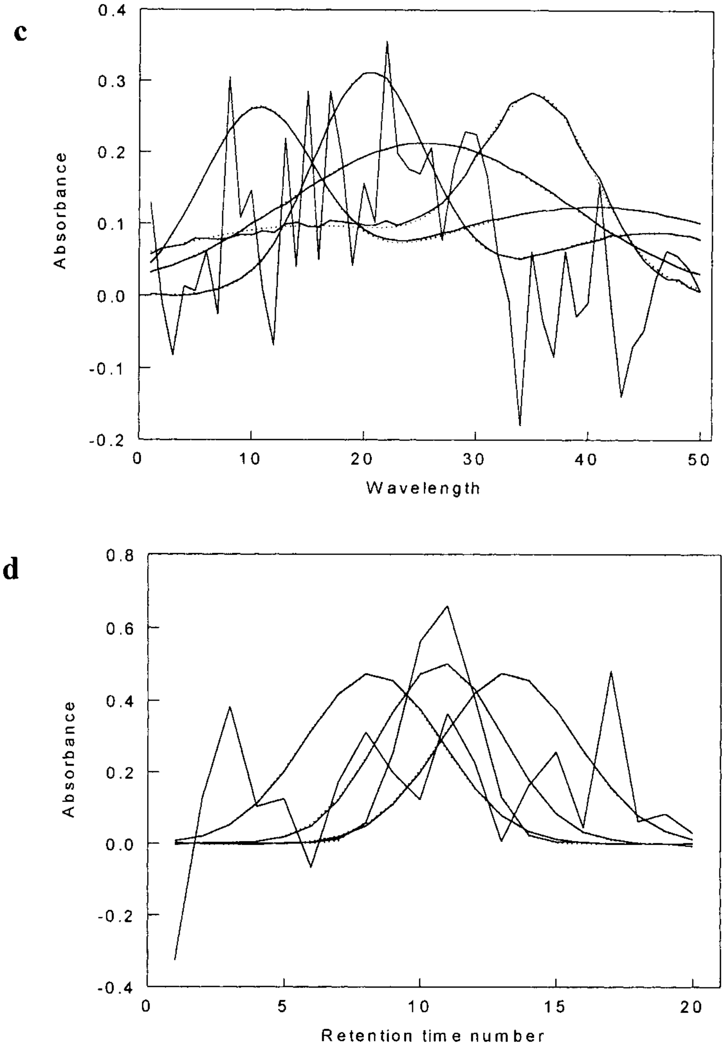


Figure 2. Continued.

five, one component of noise was extracted. It can be perceived from Table III that the entries in the concentration profile of this noise component are all nearly zero. This implies that this component made little contribution to accounting for the total variation in the data, indicating that only four components are present in the underlying model. These results showed that the ASD method was not only stable to the overestimate of the model dimensionality, but could also give an indication for the actual number of components present in the model.

Real excitation–emission fluorescence data

With the three-way fluorescence data experimentally measured, the goal of data analysis in the investigation was to recover the profiles of the species present in the six-sample system. For validating the resolution results, solutions of pure species involved were prepared and their excitation and emission spectra were measured. Two choices of the component number were employed for the

Table III. Concentration profiles of four components calculated using PARAFAC and ASD against actual ones in simulated data.

	Actual values				Values calculated by PARAFAC				Values calculated by ASD				
Sample	1	2	3	4	1	2	3	4	1	2	3	4	5
#1	0·4470	0·9359	2·2078	0	0·4454	0·9444	2·2028	0·0027	0·4485	0·9589	2·2045	−0·0090	−0·0216
#2	0·0743	2·3545	0·8657	0	0·0644	2·3612	0·8591	0·0039	0·0752	2·3795	0·8700	−0·0280	−0·0165
#3	0·0608	1·5216	0·5131	1·6158	0·0601	1·5287	0·4976	1·6190	0·0542	1·5487	0·5118	1·5985	−0·0093
#4	0·7754	2·0612	0·3245	2·5718	0·7701	2·0756	0·3048	2·5746	0·7634	2·1074	0·3257	2·5401	−0·0088
r^a					0·9999	1·0000	1·0000	1·0000	0·9999	0·9999	1·0000	1·0000	

[a] Correlation coefficients between the resolved profiles and the actual ones.

Table IV. Concentration profiles of three species calculated using PARAFAC and ASD against actual ones in real data. The component number was chosen to be three for PARAFAC and ASD. All the values listed are divided.

	Actual values			Values calculated by PARAFAC			Values calculated by ASD		
Sample	1	2	3	1	2	3	1	2	3
#1	0	1895	0	5	1941	4	5	1941	4
#2	0	0	1014	8	11	1022	6	11	1022
#3	0	1895	2028	1	1897	1851	−3	1897	1852
#4	0	3790	1014	1	3813	995	−2	3813	995
#5	1031	1895	2028	1059	1919	2233	1052	1923	2241
#6	515	3790	2028	472	3729	1998	478	3727	1994
r[a]				0·9986	0·9997	0·9893	0·9990	0·9997	0·9888

model. One choice for the component number was three, which is consistent with the actual dimensionality of the model. The other dimensionality chosen for the model was four. With these two choices of the component number the three-way array (31 × 29 × 6) was treated using the ASD method as well as the PARAFAC algorithm. When the component number was chosen to be three, it took 58 iterations for PARAFAC to converge to the solution. (For the three-component model, PARAFAC required more than 160 000 FLOPs per iteration.) The excitation and emission spectral profiles thus identified are depicted against those experimentally measured in Figures 3a and 3b, and the resolved concentration profiles are given in Table IV. One can observe that the resolved profiles show very small deviations from the actual ones. With the component number set to three, it took 38 iterations for ASD to achieve the profile estimates. (In resolving the three-component model, 6610 FLOPs were required for one ASD iteration.) The estimated excitation and emission spectral profiles are plotted in Figures 3c and 3d, and the concentration profiles are also shown in Table IV. It can be seen that the solutions recovered by ASD also exhibited very slight discrepancies compared with those experimentally measured. These results suggested that in the situation where the component number was correctly determined for the model, both ASD and PARAFAC yielded accurate estimates for the true profiles. On the other hand, it was found that when the component number was chosen to be four, the performance of PARAFAC turned out to be very poor. In this case it took 10000

Table V. Concentration profiles of three species calculated using PARAFAC and ASD against actual ones in real data. The component number was chosen to be four for PARAFAC and ASD. All the values listed are divided by 10^3.

	Actual values			Values calculated by PARAFAC				Values calculated by ASD			
Sampe	1	2	3	1	2	3	4	1	2	3	4
#1	0	1895	0	4	1022	−24	948	38	1941	5	−5
#2	0	0	1014	4	1571	1059	−1594	48	17	1022	25
#3	0	1895	2028	−2	3786	1891	−1923	−4	1897	1852	7
#4	0	3790	1014	−1	3490	974	346	−3	3808	996	−19
#5	1031	1895	2028	1047	4454	2289	−2575	1047	1940	2235	31
#6	515	3790	2028	472	4968	2013	−1250	471	3729	1995	−14
r[a]				0·9989	0·6165	0·9883		0·9971	0·9997	0·9891	

[a] Correlation coefficients between the resolved profiles and the actual ones.

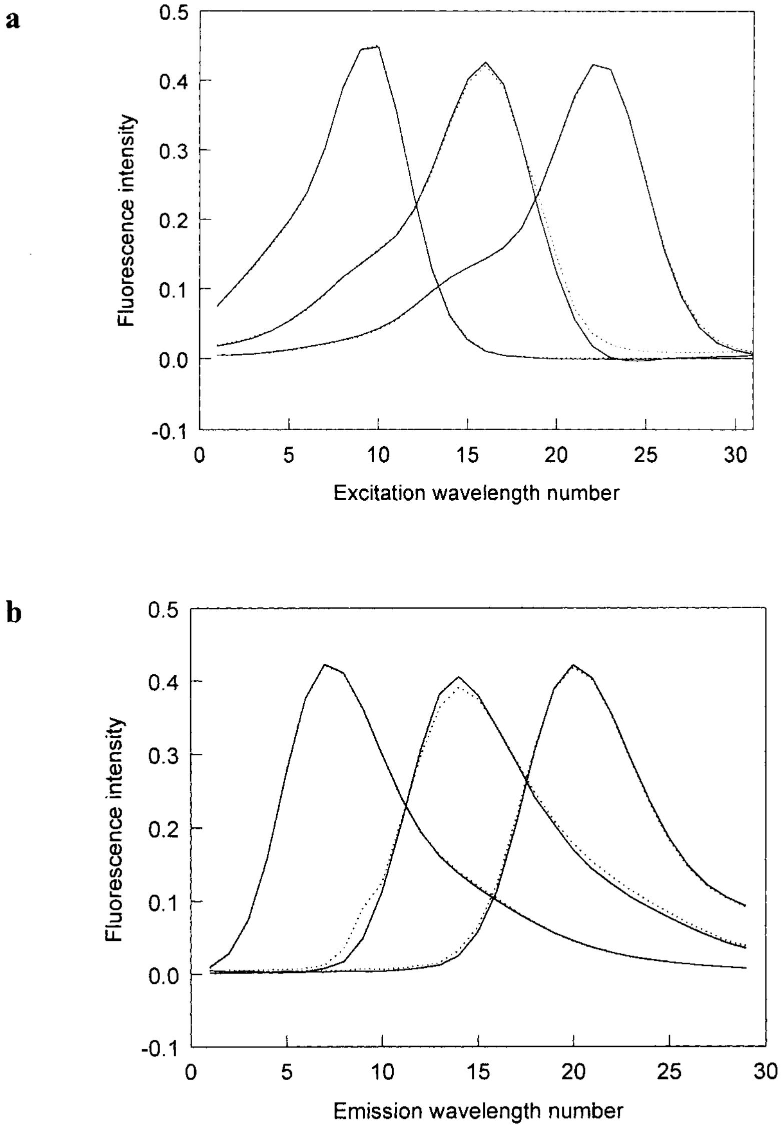


Figure 3. Spectral profiles of real fluorescence data resolved by ASD and PARAFAC methods when component number was set to three. (a) Excitation spectral profiles resolved by PARAFAC algorithm (full line) against excitation profiles experimentally measured (dotted line). (b) Emission spectral profiles resolved by PARAFAC algorithm (full line) against emission profiles experimentally measured (dotted line). (c) Excitation spectral profiles resolved by ASD method (full line) against excitation profiles experimentally measured (dotted line). (d) Emission spectral profiles resolved by ASD method (full line) against emission profiles experimentally measured (dotted line).

iterations for PARAFAC to give the solution. (For the four-component model, PARAFAC required more than 230 000 FLOPs per iteration.) These resolved excitation and emission profiles are shown in Figures 4a and IVb, and the concentration profiles are listed in Table V. One can notice that in each order only one profile fits the experimentally measured one with acceptable accuracy, while the other profiles deviate severely from those measured. These findings revealed the fact that in the case where

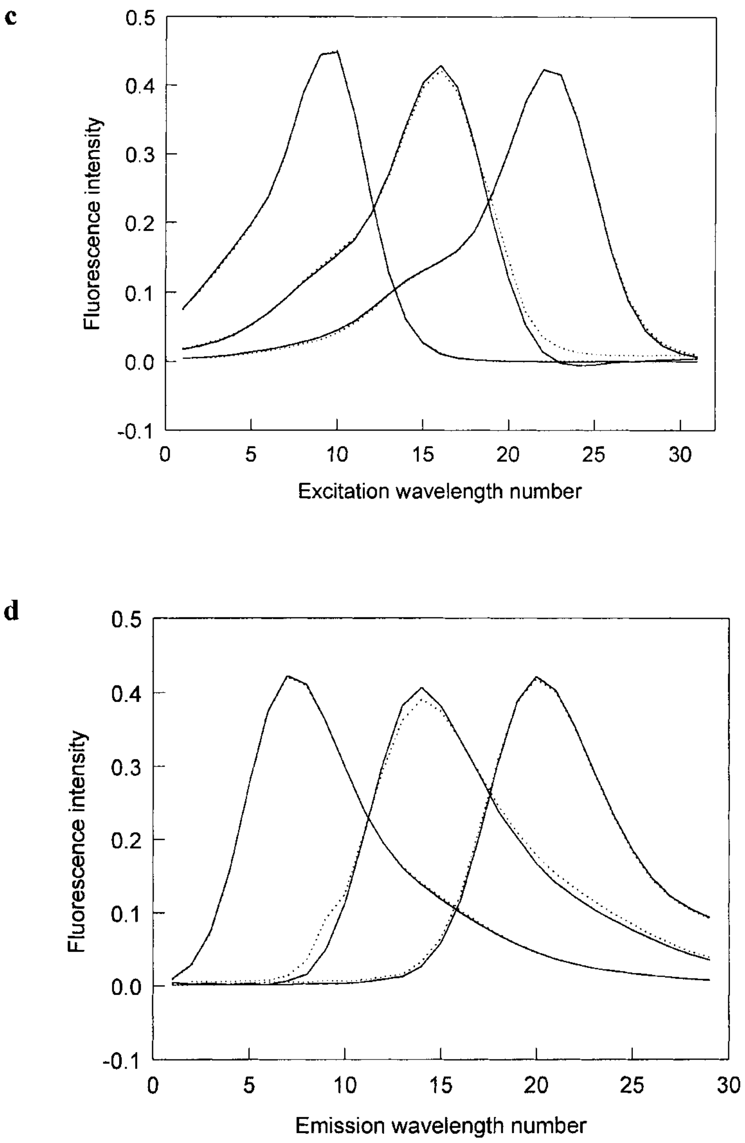


Figure 3. Continued.

the component number was incorrectly ascertained, PARAFAC generally produced a frustrating performance. In contrast, it was discovered in the study that with the model dimensionality set to four, the ASD method still exhibited desirable stability in the profile resolution. As can be seen in Figures 4c and 4d and Table V, the true profiles in three orders are accurately recovered by the ASD method, disclosing that using ASD the resolved profiles of chemical meaning are very stable in the situation where the component number is overestimated. Moreover, one can recognize that in Table V the elements of the concentration profile of the fifth component, which is associated with the measurement background and noise, are all insignificant in comparison with those of the components of chemical meaning. This characteristic is of particular significance in practical problem solving,

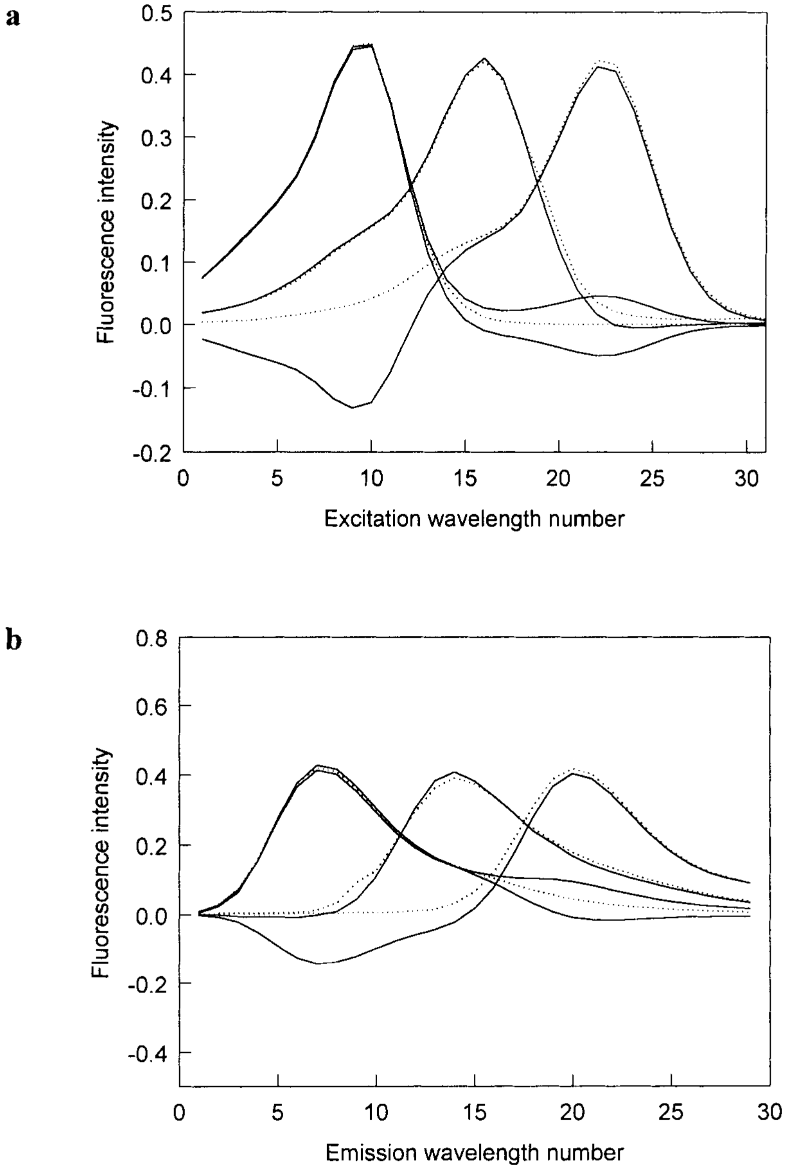
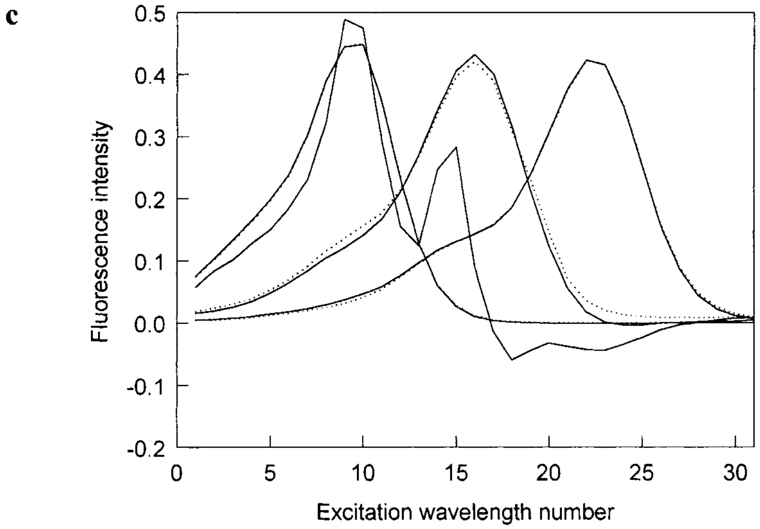


Figure 4. Spectral profiles of real fluorescence data resolved by ASD and PARAFAC methods when component number was set to four.(a) Excitation spectral profiles resolved by PARAFAC algorithm (full line) against excitation profiles experimentally measured (dotted line).(b) Emission spectral profiles resolved by PARAFAC algorithm (full line) against emission profiles experimentally measured (dotted line).(c) Excitation spectral profiles resolved by ASD method (full line) against excitation profiles experimentally measured (dotted line).(d) Emission spectral profiles resolved by ASD method (full line) against emission profiles experimentally measured (dotted line).

since it hints at the actual component number in the model. In this investigation the iteration number for ASD was merely 111. (In resolving the four-component model, ASD required 14,864 FLOPs per iteration.) This re-verified that ASD had a much higher convergence rate than PARAFAC.



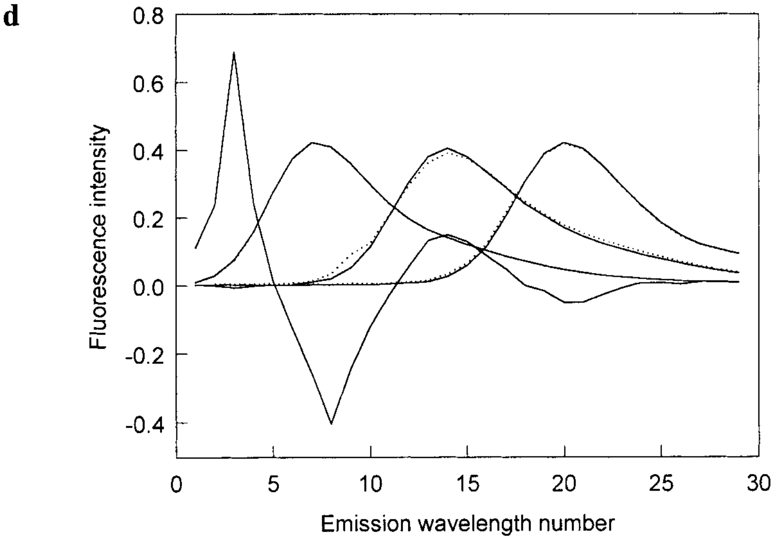


Figure 4. Continued.

CONCLUSIONS

A new method, alternating slice-wise diagonalization (ASD), has been developed for three-way data resolution. The presented results have shown that the developed ASD method can render accurate resolution for the component profiles, and the performance of the method is very stable with respect to overestimates of the component number. Moreover, the algorithm of ASD has shown a much higher convergence rate than PARAFAC. 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.

ACKNOWLEDGEMENTS

The work is supported by the National Natural Science Foundation of China (29735150).

APPENDIX: PROGRAMS IN MATLAB CODE FOR ASD

```
% Notations:
  % rrzk is the response matrix of the kth sample.
  % KK is the number of samples. KK > = 2. In the study we use KK = 4.
  % rrzr = [rrz1 rrz2 … rrzKK];
  % rrzl = [rrz1′ rrz2′ … rrzKK′];
  % II is the number of variables in x order. In the study II = 50.
  % JJ is the number of variables in y order. In the study JJ = 20.
  % XX, YY and ZZ are the resolved profiles in x, y and z orders respectively.
  % NN is an estimate of the number of components.
  % uux and uuy are the first N singular vectors of rrzr * rrzr′ and rrzl * rrzl′ respectively.
  % weight is the penalty weight.
  % AA = uux′* XX; BB = uuy′* YY; GG = inv(AA′); HH = inv(BB′).

  function [XX, YY, ZZ, errflow] = asdmain(rrzr, rrzl, II, JJ, KK, NN)

  [rrzr2, rrzl2, uux, uuy] = svddr(rrzr, rrzl, NN, JJ, KK);
  GG = eye(NN); HH = eye(NN);
  err1 = 100; derr = 1; cyc = 0; weight = 1e − 3;
  errflow = [];

  while derr > 1e − 10 & cye < 2000
    cyc = cyc + 1
    err0 = errl;
    ZZ = gghhtozz(rrzr2, GG, HH, NN, KK);
    AA = inv(GG′); BB = inv(HH′);
    errl = errorl(rrzr2, GG, HH, KK, NN);
    HH = ggzztohh(rrzr2, GG, ZZ, BB, weight, NN, KK);
    HH = HH * diag(ones(1, NN) ./ sqrt(sum(HH * HH)));
    ZZ = gghhtozz(rrzr2, GG, HH, NN, KK);
    GG = ggzztohh(rrzl2, HH, ZZ, AA, weight, NN, KK);
    GG = GG * diag(ones(1, NN) ./ sqrt(sum(GG. * GG)));
    errflow = [errflow, err1];
    derr = abs(err0 − err1);
  end

  XX = uux * AA; YY = uuy * BB;
  XX = XX * diag(ones(1, NN) ./ sqrt(sum(XX. * XX)));
  YY = YY * diag(ones(1, NN) ./ sqrt(sum(YY. * YY)));
  ZZ = xxyytozz(rrzr, XX, YY, JJ, KK, NN);

  % Post-processing to keep sign convention.
  [maxx, indx] = max(abs(XX));
  [maxy, indy] = max(abs(YY));
  xsign = ones(NN, 1); ysign = ones(NN, 1);
  for nn = 1:NN
```

```
    xsign(nn) = sign(XX(indx(nn), nn));
    ysign(nn) = sign(YY(indy(nn), nn));
end
XX = XX * diag(xsign);
YY = YY * diag(ysign);
ZZ = ZZ * diag(xsign) * diag(ysign);

function err1 = error1(rrzr, pp, qq, KK, NN)
drrz = [];
for kk = 1:KK
    rrz = pp′ * rrzr(:, NN * (kk − 1) + 1:NN * kk) * qq;
    drrz = [drrz, rrz − diag(diag(rrz))];
end    err1 = sum(sum(drrz. * drrz));

function ZZ = gghhtozz (rrzr, GG, HH, NN, KK)
for kk = 1:KK
    ZZ(kk, :) = diag (GG′ * rrzr (:, NN * (kk − 1) + 1:NN * kk) * HH)′;
end

function HH = ggzztohh (rrzr, GG, ZZ, BB, weight, NN, KK)
tt1 = zeros (NN, NN);
tt2 = zeros (NN, NN);
for kk = 1:KK
    tt1 = tt1 + rrzr (:, NN * (kk − 1) + 1:NN * kk)′ * GG * GG′ * rrzr (:, NN * (kk − 1) + 1:NN * kk);
    tt2 = tt2 + rrzr (:, NN * (kk − 1) + 1:NN * kk)′ * GG * diag(ZZ(kk,:));
end
HH = inv (tt1 + weight * BB * BB′) * (tt2 + weight * BB);

function [rrzr2, rrzl2, uux, uuy] = svddr (rrzr, rrzl, NN, JJ, KK)
[uu, ss, vv] = svd (rrzl * rrzl′);
uuy = uu (:, 1:NN);
[uu, ss, vv] = svd (rrzr * rrzr′);
uux = uu (:, 1:NN);

rrzr2 = []; rrzl2 = [];
for kk = 1:KK
    rrzr2 = [rrzr2 uux′ * rrzr (:, JJ * (kk − 1) + 1:JJ * kk) * uuy];
    rrzl2 = [rrzl2 uuy′ * rrzr (:, JJ * (kk − 1) + 1:JJ * kk)’ * uux];
end

function ZZ = xxyytozz (rrzr, XX, YY, JJ, KK, NN)
FF = zeros (KK, NN);
for kk = 1:KK
    FF(kk,:) = diag (XX′ * rrzr (:, JJ * (kk − 1) + 1:JJ * kk) * YY)′;
end
ZZ = FF * inv ((XX′ * XX). * (YY′ * YY));
```

REFERENCES

1. Ho C – N, Christian GD, Davidson ER. *Anal. Chem.* 1978; **50**:1108–1113.
2. Appellof CJ, Davidson ER. *Anal. Chim. Acta* 1983; **146**:9–14.
3. Lorber A. *Anal. Chim. Acta* 1984; **164**:293–297.

4. Grung B, Kvalheim OM. *Chemometrics Intell. Lab. Syst.* 1995; **29**:213–221.
5. Millican DW, McGown LB, *Anal. Chem.* 1990; **62**:2242–2247.
6. Leurgans SE, Ross RT, Abal RB. *SIAM J. Matrix Anal. Appl.* 1993; **14**:1064–1083.
7. Juan AD, Rutan SC, Tauler R, Massart DL, *Chemometrics Intell. Lab. Syst.* 1998; **40**:19–32.
8. Sanchez E, Kowalski BR. *Anal. Chem.* 1986; **58**:499.
9. Wilson B, Sanchez E, Kowalski BR. *J. Chemometrics* 1989; **3**:493.
10. Sanchez E, Kowalski BR. *J. Chemometrics* 1990; **4**:29.
11. Li S, Hamilton C, Gemperline P. *Anal. Chem.* 1992; **64**:599.
12. Booksh KS, Lin Z, Wang Z, Kowalski BR. *Anal. Chem.* 1994; **66**:2561.
13. Carroll JD, Chang JJ. *Psychometrica* 1970; **35**:283.
14. Harshman RA. *UCLA Working Papers Phonet.* 1970; **16**:1.
15. Appellof CJ, Davidson ER. *Anal. Chem.* 1981; **53**:2053–2056.
16. Geladi P. *Chemometrics Intell. Lab. Syst.* 1989; **7**:11–30.
17. Smilde AK, Doornbos DA. *J. Chemometrics* 1991; **5**:345–360.
18. Mitchell BC, Burdick DS. *J. Chemometrics* 1994; **8**:155–168.
19. Bro R. *Chemometrics Intell. Lab. Syst.* 1997; **38**:149–171.
20. Kruskal JB. *Linear Algebra Appl.* 1977; **18**:95.
21. Kiers HAL, Smilde AK. *J. Chemometrics* 1995; **9**:179–195.
22. Krijnen WP. *The Analysis of Three-way Arrays by Constrained PARAFAC Methods*. DSWO Press:Leiden, 1993.
23. Paatero P. *Chemometrics Intell. Lab. Syst.* 1997; **38**:223.
24. Wu HL, Shibukawa M, Oguma K. *J. Chemometrics* 1998; **12**:1.
25. Bro R, Andersson CA. *Chemometrics Intell. Lab. Syst.* 1998; **42**:105–113.
26. Linder M, Sundberg R. *Chemometrics Intell. Lab. Syst.* 1998; **42**:159–178.
27. Hopke PK, Paatero P, Jia H, Ross RT, Harshman RA. *Chemometrics Intell. Lab. Syst.* 1998; **43**:25–42.
28. Beltran JL, Guiteras J, Ferrer R. *Anal. Chem.* 1998; **70**:1949–1955.