Università degli studi di Roma

“La Sapienza”

Dipartimento di Chimica

DOTTORATO DI RICERCA IN SCIENZE CHIMICHE

XXXII CICLO

ANNO ACCADEMICO: 2018 - 2019

Multivariate Analysis in the Study of the Chemical Processes

Candidate:

Anastasiia Dulina

Supervisor:

Professor Paola D’Angelo

PhD Coordinator:

Professor Osvaldo Lanzalunga

I sincerely thank everyone who, with a kind word and a kind deed, with a wise advice, has actualized this work:
the Rector Professor Eugenio Gaudio,
the Vice Rector for International Relations Professor Bruno Botta,
the Director of the department Luciano Galantini,
the PhD coordinator Professor Osvaldo Lanzalunga,
the supervisor Professor Paola D'Angelo,
the Student Defender Professor Giuseppe Familiari,
the Head of Ph.D. office Doctor Romina Caronna,
the Ph.D. student Andrea Martini,
the master student Francesco Tavani!

CONTENTS

INTRODUCTION 5

1. THE MATHEMATICS OF FAC-PACK 15

1.1. FAC-PACK: principles of computations. Two-component systems 15

1.1.1. The area of feasible solutions (AFS) for two-component systems 16

1.1.2. Feasibility of points in AFS 17

1.1.3. Peak group analysis 20

1.2. Three-component systems 20

1.2.1. Perron-Frobenius theory 20

1.2.2. Definition of AFS 22

1.3. The polygon inflation algorithms 26

1.3.1. Traces of the spectral data matrix 35

1.3.2. Borgen Plots 36

1.3.3. Weakly separated subsets of AFS 36

1.3.4. Straight-line segments of AFS 36

2. THE MATHEMATICS OF PYFITIT 38

2.1. Principal component analysis (PCA): definition and goals 38

2.1.1. Correlation Coefficient and Matrix in PCA 38

2.2. Mathematics of PCA 39

2.2.1. Interpreting the Meaning of the principal components (PCs) 43

2.3. Spectral unmixing using PyFitIt 46

2.4. Statistical criteria to determine number of principal components 50

3. ANALYSIS OF X-RAY ABSORPTION NEAR-EDGE STRUCTURE AND ULTRAVIOLET-VISIBLE SPECTRA 54

3.1. XANES: definition and detection modes 54

3.2. Basic principles of UV-Vis spectroscopy 59

3.2.1. Qualitative and quantitative analysis of UV-Vis spectroscopy 63

4. APPLIED CHARACTER OF XANES AND UV-VIS SPECTROSCOPY 66

4.1. Oxidation reactions catalyzed by nonheme iron complexes 66

4.2. Reactions investigated with XANES spectroscopy 66

4.2.1. Results of FAC-PACK 67

4.2.2. Results of PyFitIt 84
4.3. Reaction CNPhSMe+Fe^{IV}N4Py(O) investigated with UV-Vis spectroscopy 86
4.3.1. Results of FAC-PACK 86
4.3.2. Results of PyFitIt 88
5. PROBLEMS OF PROGRAMS: FAC-PACK AND PYFITIT 89
5.1. Problems of program FAC-PACK 89
5.1.1. Equality constraint 89
5.1.2. Complementarity theorem 91
5.1.3. Relations of spectral and concentration AFS 94
5.1.4. Presence of noise in AFS 96
5.1.5. Reduction of AFS 99
5.2. Calculations of the XANES spectra with the PyFitIt program 100
5.3. Problems of PCA in PyFitIt 101
6. A STRATEGY OF SCIENTIFIC IDENTIFICATION OF PURE SPECTRAL AND CONCENTRATION PROFILES 108
6.1. Dataset 1 109
6.2. Datasets 2-3 118
6.2.1. Dataset 2 119
6.2.2. Dataset 3 125
CONCLUSIONS 130
ABSTRACT 132
BIBLIOGRAPHY 134
Appendix A 143
Appendix B 158

INTRODUCTION

Multivariate analysis (MA) is a fundamental part of mathematics. It is an effective method of modern science researching. MA is applied: to analyze spectroscopic data in physical chemistry; in pure component analysis in analytical chemistry; in high performance liquid chromatography with diode array detection (HPLC–DAD) chromatographic data sets; for a treatment of hyperspectral images in hyperspectral imaging.

A spectral mixture is a dataset that results from the observation of a chemical system composed of (mixed) individual components and submitted to some variation. This variation is related to the change of an external factor, which is usually a physical or chemical variable. It can be for example sampling time, position, or pH. The spectral data thus consist of a superposition, or mixture, of the pure spectra of the individual components and their associated proportions. When dealing with evolving systems such as chemical reactions or processes, these proportions correspond to concentration profiles.

In chemistry, spectral mixture resolution corresponds to the resolution of complex mixture spectra into pure contributions, consisting of concentration distributions and spectra of the different chemical components. It is important to realize that, more than often, this decomposition is aimed at situations for which little a priori information is available. In practice, some physical perturbations or chemical interferences may complicate the ideal situation. The basic model underlying this decomposition, usually termed multivariate curve resolution (MCR) in chemometrics [53]. MCR is the generic denomination of a family of MA methods meant to solve the mixture analysis problem, i.e., able to provide a chemically (scientifically) meaningful additive bilinear model of pure contributions from the sole information of an original data matrix including a mixed measurement.

The main goal of the thesis is: to show the applications of the multivariate analysis in the study of the chemical processes.

The tasks of the thesis are:

1. Having the three species, coexisting in solution, and the time evolution of the total absorption spectra it is required to:

✓ Extract the concentration C_1 as it decreases, C_2 as it increases and then redecreases and C_3 as it increases over time;

✓ determine three forms of the spectra, that multiplied by C_1, C_2 and C_3 respectively at each instant of time, and summed up give the original mixture dataset.
The same goal is for the two species, coexisting in solution.

2. Determine the concentrations (in the function of the time), that is the matrix $C_{3\times time}$.

3. Determine the absorption spectra (in the function of energy), that is the matrix $A_{3\times energy}$.

4. Obtain pure spectral and concentration profiles for different X-ray absorption near-edge structure (XANES) and ultraviolet-visible (UV-Vis) datasets with different number of components using computer softwares FAC-PACK and PyFitIt.

5. Make a comparative analysis of the effectiveness of the obtained results in FAC-PACK and in PyFitIt.

6. Develop and implement the theory for application of the nonnegative matrix factorization (NMF) method, constrained nonlinear method, principal component analysis (PCA) method and test them for different datasets.

7. Identify the regions populated by t_{jk} values of the transformation matrix T capable to provide the non-negative specta and concentration profiles, which satisfy mass balance sheet.

8. For these t_{jk} values, which satisfy the previous point 7, isolate a set of "pure" spectra having a chemical-physical meaning.

9. To identify the maxima and the minima meaningful spectral and concentrations boundaries under constraints.

10. To write a computer program in programming language Python to identify pure spectral and concentration profiles and their related range of confidence for different XANES datasets with different number of components.

MA can be applied in situations where a reasonable approximation of the bilinear model, or any other fundamental basic equation that has the same mathematical structure, holds. Application of MA methods is broad, quite straightforward, and provides results which are readily chemically/physically interpretable. These assets explain why MA has spread in the chemical literature and in many other scientific fields. However, considering the mathematical conditions for exact resolution of the MCR problem, some theoretical issues remain and are currently the subject of intensive research.

The most puzzling of these issues is the so-called rotational ambiguity of the resolution. In more common words, this translates into the fact that a unique solution cannot be obtained in general. Then, particular attention should be paid at the initial conditions, or to the constraints applied during resolution, and it is important to assess the extent of rotational ambiguity before drawing any definitive conclusion. Considering these aspects, one may notice a certain antagonism in MA between wide applicability and high interpretability on the one hand, and mathematical complexity of the resolution on the other hand. This explains to a large extend the continuous development of this topic into a proper research field, still very much in progress.

The first applications of MCR were devoted to analyze spectroscopic data because of the identical underlying models of the measurement and the method [8]. A variance with other bilinear decomposition methods, MCR provides meaningful profiles because chemical properties related to the concentration profiles and spectra are actively incorporated in the optimization of the bilinear model.

There are many multicomponent systems described by spectroscopic data. MA adapts to this diversity by tuning the way of application of the method according to the characteristics of the spectroscopic technique and the concentration profiles. In a very general way, we can distinguish between process and mixture data sets. Process data sets show very structured concentration profiles, displaying a smooth variation as a function of a process variable. Typical examples may be an high performance liquid chromatography with diode array detection (HPLC–DAD) chromatographic data set, where the concentration (elution) profiles have a peak-shaped signal or reactions monitored by spectroscopic techniques. Instead, the concentration profiles of a mixture data set can vary in a non-patterned way.

This less structured variation can come from the nature of the data, e.g., a set of independent samples, or from the data set configuration needed for MA application, e.g., hyperspectral images (HSIs) are treated by using a data table of pixel spectra, which does not preserve the spatial organization of the original measurement. MA applies in all the scenarios described, but specificities of the different data set typologies should be taken into account in the data set configuration and in the different steps of application of the algorithm.

Before describing the details and application of MA algorithms, a necessary comment has to be made regarding one of the fundamental assumptions of MCR, i.e., the bilinearity of spectroscopic data. Indeed, the natural fulfillment of the Lambert-

Beer law may be affected by signal artifacts, such as scattering in near infrared spectroscopy, or fluorescence contributions in Raman spectra. Most of the times, the bilinear behavior is easily recovered by suitable preprocessing, e.g., scatter or baseline correction, adapted to the spectroscopic measurement of interest. Because of the nature of the spectroscopy used, some techniques need more dedicated and intensive preprocessing, like ultrafast spectroscopy measurements and, in extreme cases, nonlinear unmixing methods can be applied.

MA methods are key-tools in order to extract the pure component information (pure component spectra and the concentration profiles) from the chemical mixture (spectroscopic) data. The problem is to compute:

1. the number of independent components *s* and

2. the pure component factors C (concentration profiles) and A (spectra).

Any available information on the factors can and should be integrated into the MCR computations.

Typically, the data is taken by spectral observation of a chemical reaction on a time × frequency grid. If *k* spectra are measured, each at *n* frequencies, then the resulting matrix is a $k \times n$ matrix. The measured data result from a superposition of the pure component spectra.

The basic bilinear model underlying these methods is the Lambert-Beer law. In matrix notation, the Lambert- Beer law is a relation between *D* and the matrix factors $C \in \mathbb{R}^{k \times s}$ and $A \in \mathbb{R}^{s \times n}$ in the form

$$D = CA. \tag{0.1}$$

Therein *s* is the number of the pure or at least independent components. An error matrix $E \in \mathbb{R}^{k \times n}$ with entries close to zero can be added on the right-hand side of (0.1) in order to allow approximate factorizations in case of perturbed or noisy data *D*. In general, the matrices *C* and *A* are called abstract factors.

One is interested in finding a factorization *D=CA* with chemically interpretable *C* and *A*. Then the columns of $C \in \mathbb{R}^{k \times s}$ are the concentration profiles along the time axis of the pure components. And the rows of $A \in \mathbb{R}^{s \times n}$ are the associated pure component spectra.

The main questions are:

1. If inside the sample instead of a single substance there are several substances, each of them will absorb and there is a need to determine:

- ✓ the concentration C;
- ✓ the form of the absorption spectrum

for each of these substances!

2. To find a unique solution for identifying of pure spectral and concentration profiles and their related range of confidence.

The main hurdle for any MCR technique is the so-called rotational ambiguity of the solution. By applying additional hard- or soft-constraints to the pure component factorization problem, one can often determine a single solution by means of a regularized optimization problem. In case of proper constraints this solution can be the chemically correct one.

A large number of successful MCR methods has been developed. Some examples are methods as Multivariate curve resolution-alternating least squares (MCR-ALS), Resolving factor analysis (RFA), Simple to use interactive self-modelling algorithm (SIMPLISMA), Band-target entropy minimization (BTEM) and Pure component decomposition (PCD). Alternatively, one can give up the aim of determining only a single solution by solving a regularized optimization problem. Instead, one can follow the global approach of determining the full range of all nonnegative factorizations $D = CA$ with nonnegative rank-s matrices C and A. Such continua of possible nonnegative matrix factors can graphically be presented either by drawing the bands of possible concentration profiles together with the bands of possible spectra, or by plotting these sets of feasible factors by a certain low-dimensional representation, the so-called *Area of Feasible Solutions* (AFS).

The singular value decomposition (SVD) is a very powerful tool of numerical linear algebra to compute the left and right orthogonal bases for the expansion of the pure component factors $C \in \mathbb{R}^{k \times s}$ and $A \in \mathbb{R}^{s \times n}$. The SVD of D reads

$$D = U \Sigma V^T.$$

Therein $U \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices whose columns are the left and right singular vectors. The diagonal matrix Σ contains on its diagonal the singular values σ_i in decreasing order. The singular values are real and nonnegative.

For a *s*-component system the first *s* singular vectors and the associated singular values contain all information on the system. For data not including perturbations only the first *s* singular values are nonzero if the chemical system contains *s* independent chemical components. For data including noise some additional singular values are nonzero. In such cases the SVD allows one to compute optimal (with respect to least-squares) rank-*s* approximations of *D*. If in the case of noisy data the noise-to-signal ratio is not too large, then the number of independent chemical components *s* can often be determined from the SVD. Then the relevant and meaningful singular values are clearly larger as compared to the remaining nonzero singular values which highlight the influence of noise.

The first *s* singular vectors, namely the first *s* columns of *U* and the first *s* columns of *V*, are used as bases to expand the desired pure component factors *C* and *A*. For ease of notation we denote these submatrices of the SVD-factors again by *U* and *V*. Then $U \in \mathbb{R}^{k\times s}$ and $V \in \mathbb{R}^{n\times s}$. The matrices *C* and *A* are formed according to

$$C = U\Sigma T^{-1}, \quad A = TV^T \tag{0.2}$$

Therein $T \in \mathbb{R}^{s\times s}$ is a regular matrix which remains to be determined. From *C* and *A* the matrix *T* of expansion coefficients is accessible from Equation (0.2).

The basis expansion approach (0.2) drastically reduces the number of free variables of the pure component factorization problem. The crucial point is that the number of matrix elements of *C* and *A* is $(k + n)s$ whereas the representation by Equation (0.2) reduces the degrees of freedom to the s^2 matrix elements of *T*. Hence the representation (0.2) is a basic ingredient for the construction of computationally effective MA methods.

Hard- and soft constraints have a crucial role in the construction of MA methods. A very restrictive and often successful hard constraint is a kinetic model of the underlying chemical reaction. Only those concentration values *C* are acceptable which are consistent with the kinetic model. Typically, the rate constants are implicitly computed as a by-product of the model fitting process.

If no kinetic model is available for *C*, then soft constraints can be used in order to extract (from the set of all nonnegative factorizations) solutions with special properties. Typical examples of such soft constraints are those on the smoothness of

the concentration profiles in *C* or *A*, constraints on a small or large integral of the spectra in *A* (in order to favor solutions with few and sharp peaks or alternatively those with a large number of wide peaks), criteria on the closure of the concentration data and so on.

Even with proper constraint functions and proper weight factors, MA methods cannot always find the chemically correct or "true" solution. Thus one might follow the alternative idea to determine the set of *all* nonnegative factorizations $D = CA$. Such solutions which only fulfill the nonnegativity constraint are called *feasible* or *abstract* factors.

The global approach of computing all feasible factors provides an elegant way in order to survey the complete rotational ambiguity of the pure component factorization problem. However, the sets of feasible matrices $C \in \mathbb{R}^{k \times s}$ or $A \in \mathbb{R}^{s \times n}$ are difficult to handle. The key idea to make these sets of feasible factors accessible is their low-dimensional representation in terms of the AFS. The AFS refers to the representation of *C* and *A* as functions of *T* by Equation (0.2). Only a single row of *T* is sufficient to represent a feasible factor.

In the following, the MA aims at determining the AFS for the spectral factor *A* starting from a spectral data matrix *D*. This analysis can immediately be used to determine the concentration AFS containing the feasible factors *C*. Therefore, we apply the procedure to the transposed data matrix D^T since in $D = CA$ the factors change their places by the transposition $D^T = A^T C^T$.

The developments of the AFS concepts are closely related with the growth of effective numerical methods for its computation.

The AFS for two-component systems was first analyzed by Lawton and Sylvestre in 1971. The Lawton-Sylvestre plot is a 2D plot of the set of the two expansion coefficients (with respect to the basis of singular vectors) which result in nonnegativematrix factors *C* and *A*.

For three-component systems a direct analogue of the Lawton-Sylvestre plot would be a three-dimensional plot of feasible expansion coefficients. Such three-dimensional objects are somewhat more complicated to draw, to handle and to understand. However, there is a tricky dimension reduction (by a certain scaling) which allows one to represent these AFS sets for three-component systems only by two expansion coefficients (and thus by plots in 2D). This was suggested by Borgen

and Kowalski, who published in 1985 a geometric construction of these 2D AFS plots for three-component systems. These plots are called Borgen plots.

In addition to the geometric constructions of the Borgen plots, various techniques for a numerical approximation of the AFS for (s = 3)-component systems have been devised. These are the grid search method, the triangle enclosure method and the polygon inflation method.

One benefit of the numerical methods compared to the geometric methods is that the numerical methods are able to compute the AFS in the presence of noise.

The permutation of the columns of C and the same permutation applied to the rows of A does not provide any new information. This fact is known as the (trivial) permutation ambiguity. A consequence of this property is that the problem to find all feasible factors A is equal to the problem to determine the set of all first rows of the feasible factors A. Hence the set of feasible pure component spectra (also called feasible bands) for an s-component system reads

$$\mathcal{H} = \left\{ a \in \mathbb{R}^n : \text{exist } C, A \geq 0 \text{ with } A(1,:) = a \text{ and } D = CA \right\} \tag{0.3}$$

For the computation of $\mathcal{H}$ we prefer the SVD-based approach (0.2). A further reduction of the degrees of freedom is possible. This is explained next.

Equation (0.2) is a representation of the $s \times n$ matrix A by the matrix T with only s^2 matrix elements. These s^2 matrix elements are the expansion coefficients with respect to the basis of right singular vectors. As shown for the derivation of (0.3) only the first row of $A = TV^T$ is required in order to form the set $\mathcal{H}$ of feasible spectra. The first row of A equals the first row of TV^T. Hence only the s matrix elements of the first row of T are decisive. This reduces the degrees of freedom from s^2 down to s. These remaining s degrees of freedom for a s-component reaction can further be reduced to $s - 1$ by a certain scaling of the rows of A. Here, we follow the approach in Ref. [16] and use a scaling which sets the first column of T equal to ones, i.e.

$$T(i,1) = 1 \ \textit{for all} \ i = 1, \ldots, s \,. \tag{0.4}$$

This can be called the first-singular-vector scaling (FSV-scaling) since it uses for the first right singular vector the fixed expansion coefficient 1. A precise

justification for this choice is based on the Perron-Frobenius theory on spectral properties of nonnegative matrices.

With respect to the FSV-scaling (0.4) the matrix T in (0.2) has the form

$$T = \begin{pmatrix} 1 & x_1 & \cdots & x_{s-1} \\ 1 & & & \\ \vdots & & S & \\ 1 & & & \end{pmatrix} \tag{0.5}$$

Therein S is an $(s-1)\times(s-1)$matrix. Only the $s-1$ elements of the row vector $x = (x_1, \ldots, x_{s-1})$ are decisive for the representation of the set of all feasible solutions.

With these definitions the set $\mathcal{H} \subset \mathbb{R}^n$ by Equation (0.3) can be represented by the associated set of expansion vectors $x \in \mathbb{R}^{s-1}$. Such a set of $(s-1)$-dimensional vectors for a chemical reaction with s species is much easier to handle as compared to the subset $\mathcal{H}$ of the higher dimensional space $\mathbb{R}^n$. The set

$$\mathcal{M} = \{ x \in \mathbb{R}^{s-1} : \text{ exists } S \text{ so that } T \text{ in (0.5) fulfills } \operatorname{rank}(T) = s \text{ and } C, A \geq 0 \} \tag{0.6}$$

is called the AFS for the factor A or the spectral AFS.

The AFS has several interesting properties. Many of these properties derive from the Perron-Frobenius spectral theory of nonnegative matrices. This theory provides the justification for the scaling condition (0.4). The properties of the AFS are the following:

1) The set

$$\mathcal{M}^+ = \{ x \in \mathbb{R}^{s-1} : (1, x)V^T \geq 0 \} \tag{0.7}$$

is called FIRPOL. All points x in $\mathcal{M}^+$ result in nonnegative linear combinations of the right singular vectors, i.e. $(1, x)V^T$. Thus FIRPOL is a superset of the set of feasible spectra. The membership of a certain x to $\mathcal{M}^+$ does not guarantee that the

nonnegative spectrum $(1, x)V^T$ is part of a feasible pure component decomposition $D = CA$. The crucial point is that nonnegativity of $(1, x)V^T$ does not necessarily imply the nonnegativity of an associated concentration profile.

Further, the set $\mathcal{M}^*$ is to be introduced

$$\mathcal{M}^* = \{x \in \mathbb{R}^{s-1} : \text{ exists } S \text{ so that } T \text{ in (0.5) fulfills } \operatorname{rank}(T) = s \text{ and } C \geq 0, A(2:s,:) \geq 0\} \tag{0.8}$$

The two sets $\mathcal{M}^+$ and $\mathcal{M}^*$ are super-sets of the AFS $\mathcal{M}$. The definition of $\mathcal{M}^+$ includes only the nonnegativity constraint $A(1, :) \geq 0$. The definition of $\mathcal{M}^*$ includes the remaining constraints on nonnegativity and the rank condition. Thus $\mathcal{M} = \mathcal{M}^+ \cap \mathcal{M}^*$. Finally, the vectors

$$w(i,:) = \frac{D(i,:)V(:,2:s)}{D(i,:)V(:,1)} \in \mathbb{R}^{s-1} \text{ for } i = 1, \ldots, n \tag{0.9}$$

are introduced. The convex hull of these points $w(i,:), i = 1, \ldots, n$, is called INNPOL.

2) The origin (or null vector) is never contained in the AFS. The proof for this fact is given in [79]. It is based on the Perron-Frobenius theory and uses the irreducibility of the matrix $D^T D$.

3) The AFS is a bounded set. It is proved that $\mathcal{M}^+$ is a bounded set if and only if $D^T D$ is an irreducible matrix. Thus the AFS $\mathcal{M}$ is also a bounded set since $\mathcal{M} \subset \mathcal{M}^+$. This property makes possible a numerical approximation of the boundary of the AFS.

The AFS sets may have several shapes. For 3-component systems the most important cases are: a) AFS sets consisting of several isolated separated segments which do not contain any hole. In mathematics such sets are called *simply-connected*. The approach of inflating polygons can be used in order to approximate such AFS segments; b) AFS sets which have the form of a single topologically connected set with a single hole. Such a hole always contains the origin (null vector). The inverse polygon inflation algorithm is a modification of the polygon inflation algorithm which allows a fast and accurate numerical approximation of such one-segment AFS sets with a hole.

1. THE MATHEMATICS OF FAC-PACK

FAC-PACK is a software package for the computation of nonnegative multi-component factorizations and for the numerical approximation of the AFS. The software contains implementations of the polygon inflation method, the inverse polygon inflation method and the geometric constructive approach of generalized Borgen plots.

Given a nonnegative matrix $D \in R^{k \times n}$, which may even be perturbed in a way that some of its entries are slightly negative, a MCR technique can be used to find nonnegative matrix factors $C \in R^{k \times s}$ and $A \in R^{s \times n}$ so that

$$D \approx CA. \tag{1.1}$$

The most rigorous approach is to compute the complete continuum of nonnegative matrix factors (*C*, *A*) which satisfy (1).

1.1. FAC-PACK: principles of computations. Two-component systems.

FAC-PACK works as follows: first the data matrix *D* is loaded. The SVD is used to determine the number s of independent components underlying the spectral data in *D*. FAC-PACK can be applied to systems with *s=2*, *s=3* and *s=4* predominating components.

Noisy data are not really a problem for the algorithm as far as the SVD is successful in separating the characteristic system data (larger singular values and the associated singular vectors) from the smaller noise-dependent singular values. Then the SVD is used to compute a low rank approximation of *D*.

After this, an initial NMF is computed from the low rank approximation of *D*. This NMF is the starting point for the polygon inflation algorithm since it supplies two or three points within the AFS. From these points an initial polygon can be constructed, which is a first coarse approximation of the AFS. The initial polygon is inflated to the AFS by means of an adaptive algorithm. This algorithm allows one to compute all three segments of an AFS separately. Sometimes the AFS is a single topologically connected set with a hole. Then an “inverse polygon inflation” scheme is applied.

The program allows one to compute from the AFS the continuum of admissible spectra. The concentration profiles can be computed if the whole algorithm is applied to the transposed data matrix D^T. Alternatively, the spectral and the concentrational AFS can be computed simultaneously within the "Complementarity & AFS" graphical user interface (GUI).

FAC-PACK provides a live view mode which allows an interactive visual inspection of the spectra (or concentration profiles) while moving the mouse pointer through the AFS. Within the live view mode the user might want to lock a certain point of the AFS, for instance, if a known spectrum has been found. Then a reduced and smaller AFS can be computed, which makes use of the fact that one spectrum is locked. For a three-component system this locking and AFS reduction can be applied to a second point of the AFS.

Within the graphical user interface "Complementarity & AFS" the user can explore the complete factorization $D = CA$ simultaneously in the spectral and the concentration AFS. The factorization $D = CA$ is represented by two triangles. The vertices of these triangles can be moved through the AFS and can be locked to appropriate solutions. During this procedure the program always shows the associated concentration profiles and spectra for all components. In this way FAC-PACK gives the user a complete and visual control on the factorization. It is even possible to import certain pure component spectra or concentration profiles in order to support this selection-and-AFS-reduction process [81].

1.1.1. The area of feasible solutions (AFS) for two-component systems.

The AFS for systems with a number of $s = 2$ components can explicitly be described analytically. The numerical evaluation of the analytic formula results in the 1D AFS plots. The starting point for the case $s = 2$ is the matrix T by (0.5) which together with its inverse reads

$$T = \begin{pmatrix} 1 & x \\ 1 & S_{11} \end{pmatrix}, \quad T^{-1} = \frac{1}{S_{11} - x} \begin{pmatrix} S_{11} & -x \\ -1 & 1 \end{pmatrix}.$$

The nonnegativity for the factors C and A results in feasible intervals for x and S_{11}. With

$$a = -\min_{i \text{ with } V(i,2)>0} \frac{V_{i1}}{V_{i2}}, \quad b = \min_i \frac{U_{i2}\sigma_2}{U_{i1}\sigma_1},$$
$$c = \max_i \frac{U_{i2}\sigma_2}{U_{i1}\sigma_1}, \quad d = -\max_{i \text{ with } V(i,2)<0} \frac{V_{i1}}{V_{i2}} \tag{1.2}$$

the AFS for the two-component system has the form of two separated intervals

$$\mathcal{M} = [a, b] \cup [c, d]. \tag{1.3}$$

This result can be interpreted in a way that

$$x \in [a, b] \text{ and } S_{11} \in [c, d] \text{ and } S_{11} \in [a, b] \text{ and } x \in [c, d]$$

result in nonnegative factors. A certain choice $(x, S_{11}) \in [a, b] \times [c, d]$ completely determines a nonnegative factorization $D = CA$. The second choice $(S_{11}, x) \in [a, b] \times [c, d]$ does not provide any new information. Instead, the second solution is equal to the first solution after a row permutation in A and a column permutation in C. This fact justifies that the AFS for two-component systems is often represented by the rectangular $[a, b] \times [c, d]$.

1.1.2. Feasibility of points in AFS.

There is first the feasibility analysis for noise-free (model) data by considering a certain geometric construction. Alternatively, and with a focus on experimental spectral data, there is a numerical feasibility analysis which is based on the numerical solution of an optimization problem. Unfortunately, the numerical feasibility test can yield false results, if the numerical optimization procedure (e.g. due to convergence problems or poor initial estimates) is not successful.

This paragraph explains the feasibility checks of the polygon inflation algorithm by soft constraints, of the triangle enclosure technique as well as the grid search method and of the geometric-constructive Borgen plot approach.

Soft constraints can be added to the feasibility check on non-negativity. The aim of this approach is to compute the matrix elements of the submatrix S of (0.5) by solving a minimization problem for a certain target function which guarantees that CA is a good approximation of the initial matrix D. Simultaneously, various

constraints on C and A are to be satisfied. This feasibility test which also underlies the polygon inflation algorithm is explained in the following.

First it is introduced a small control parameter $\varepsilon \geq 0$ so that $-\varepsilon$ is a lower bound for the acceptable negative elements of C and A (in a relative sense related to the maximal value of a concentration profile or spectrum). Mathematically, these conditions read

$$\frac{\min_j C(j,i)}{\max_j C(j,i)} \geq -\varepsilon, \quad \frac{\min_j A(i,j)}{\max_j A(i,j)} \geq -\varepsilon$$

for all $i = 1,...,s$. The acceptance of small negative entries can often significantly stabilize the computation in case of noisy or perturbed (e.g. by a background subtraction) data.

The feasibility test for a point x is made in two steps. First, a rapid and computationally very cheap test is used in order to check whether x is contained in the set FIRPOL $\mathcal{M}^+$, see Equation (0.7). If x is not in FIRPOL, then x cannot be an element of the AFS $\mathcal{M}$. Once again, we accept small negative entries. To this end we use an approximate FIRPOL test in order to check whether or not

$$f_0(x) := \min\left(\frac{(1,x)V^T}{\left\|(1,x)V^T\right\|_\infty} + \varepsilon, 0\right) \tag{1.4}$$

satisfies that $f_0(x) \geq 0$ (in a component-wise sense). Therein, $\|\cdot\|_\infty$ is the maximum vector norm, which is the largest absolute value of all components of its argument. If this test is passed successfully, then a second much more expensive test follows. Therefore the soft constraint function

$$\begin{aligned} f(x,S) = &\sum_{i=1}^{s} \left\| \min\left(\frac{C(:,i)}{\| C(:,i) \|_\infty} + \varepsilon, 0\right)\right\|_F^2 + \\ &+ \sum_{i=2}^{s} \left\| \min\left(\frac{A(i,:)}{\| A(i,:) \|_\infty} + \varepsilon, 0\right)\right\|_F^2 + \left\| I_s - TT^+ \right\|_F^2 \end{aligned} \tag{1.5}$$

is considered with $T = T(x, S)$ by (0.5). Further, C, A are computed according to (0.2). If

$$\min_{S \in \mathbb{R}^{(s-1)\times(s-1)}} f(x,S) \le \varepsilon_{\text{tol}}, \tag{1.6}$$

then x has passed the feasibility test successfully. Therein ε_{tol} is a small positive control parameter, e.g. $\varepsilon_{\text{tol}} = 10^{-10}$.

To summarize, the approximate feasibility test with the control parameters $\varepsilon$and ε_{tol} results in the (approximate) AFS

$$\mathcal{M} = \left\{ x \in \mathbb{R}^{s-1} : x \text{ fulfills } f_0(x) \ge 0 \text{ and } \min_S f(x,S) \le \varepsilon_{\text{tol}} \right\}. \tag{1.7}$$

The *ssq*-function (ssq for sum-of-squares) approach evaluates the reconstruction functional

$$\text{ssq}(x,S) = \| D - \max(C,0) \cdot \max(A,0) \|_F^2$$

Therein max(C, 0) and max(A, 0) are the matrices whose negative entries are zeroed. The matrices C and A depend on $T = T(x, S)$ according to (0.5). The triangle enclosure algorithm and the grid search method are based on the evaluation of the *ssq*-function. Computationally, the *ssq*-evaluation is relatively expensive as the computation of O($k \cdot n$) squares is required whereas the evaluation of (1.5) needs only O($k + n$) squares. For large values of k and/or n this results in significantly different computation times.

Finally, the AFS can be written as

$$\mathcal{M} = \left\{ x \in \mathbb{R}^{s-1} : \min_S \text{ssq}(x,S) \le \varepsilon_{\text{tol}} \right\}$$

for a fixed small parameter $\varepsilon_{\text{tol}} > 0$.

The geometric feasibility test of a certain point $x \in \mathcal{M}^+$ for the case $s = 3$ consists of the following steps: First two tangents of INNPOL are constructed which run through the given point x and which (tightly) enclose INNPOL. Next the intersection of the first tangent with the boundary of $\mathcal{M}^+$ (the line segment between x and this point must touch at least one point w_i) is defined as P_1. The same is done for the second tangent. This results in the point P_2. Then x is a feasible point of the AFS if and only if the triangle with the vertices x, P_1 and P_2 includes the polygon

INNPOL. An extension of this geometric construction which is applicable to noisy or perturbed data is developed in Ref. [4].

1.1.3. Peak group analysis.

The Peak Group Analysis (PGA) has been presented as a numerical algorithm which allows a step-by-step computation of the pure component spectra from the initial spectral data set for the chemical mixture. A crucial requirement for a successful application of the PGA is that certain single peaks or isolated peak groups can be identified whose spectral profile is dominated by a single pure component. Then this peak or peak group is the starting point for a local optimization procedure which results in a global spectrum of a pure component. This global spectrum more or less reproduces the initial peak or peak group. The mathematical algorithm of the PGA is based on the minimization of a target function to which various weighted soft constraints are added.

PGA is a robust algorithm for medium-to-strong perturbed spectral data. The method also works very well in the case of systematic perturbations for example from a suboptimal baseline correction.

PGA may extract single pure components and is able to uncover correlations within highly overlapping peak groups. The technical reason for these abilities is the window analysis of the spectra and the reduced number of variables of the cost function.

1.2. Three-component systems.

1.2.1. Perron-Frobenius theory.

Definition 1.1. A matrix is called nonnegative (positive) if all its elements are nonnegative (positive).

Here, we consider only nonnegative square matrices of order n, i.e., matrices that have n rows and n columns.

Theorem 1.1 (Perron's Theorem). Let A be a positive matrix; then A has a positive eigenvalue λ_A such that

(a) λ_A is a simple root of the characteristic equation of A, and

(b) the value of λ_A is strictly greater than the absolute value of any other eigenvalue of A.

The eigenvalue λ_A corresponds to a unique (with an accuracy of up to a scalar factor) positive eigenvector x_A.

The set of nonnegative matrices has a special subset. Matrices from that subset, called irreducible (indecomposable) matrices, have many of the properties of positive matrices. The set of irreducible matrices is defined as the complement of the set of reducible (decomposable) matrices. Given a matrix that has zero elements, its (ir)reducibility is determined by the positions of those elements. A formal definition of a reducible matrix of order n (which is not necessarily nonnegative) will be reported below. In this definition, *N* denotes the index set $\{1,\ldots,n\}$.

Definition 1.2. Let A be a square matrix of order n, $n\geqslant 2$, formed by elements a_{ij}. The matrix A is called a reducible matrix if there exists a subset *S* of the index set *N* such that $a_{ij} = 0$ for all $j \in S$ and $i \notin S$.

The index set *S* that appears in this definition is called *isolated*. If A is a reducible matrix such that $S = \{1,\ldots,k\}$, then A has the following form:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \tag{1.8}$$

where A_{11} and A_{22} are square submatrices of order *k* and *n-k*, respectively. These submatrices can also contain zero elements. Moreover, these submatrices can, in turn, be reducible matrices. Any reducible matrix can be put in form (1.8) by simultaneously permuting its rows and columns.

It is clear that any positive matrix is irreducible. Any matrix of order 1 is considered to be reducible if and only if its (only) element equals zero.

G. Frobenius generalized Perron's theorem to irreducible matrices.

Theorem 1.2 (Frobenius' Theorem). Let A be an irreducible nonnegative matrix; then A has a positive eigenvalue λ_A such that

(a) λ_A is a simple root of the characteristic equation of A, and

(b) the value of λ_A is not less than the absolute value of any other eigenvalue of A.

The eigenvalue λ_A corresponds to a unique (with an accuracy of up to a scalar factor) positive eigenvector x_A.

Suppose A has m eigenvalues $\lambda_1,\ldots,\lambda_{m-1},\lambda_m=\lambda_A$ such that the absolute value of any of them equals λ_A. Then all these numbers are distinct and the set of these numbers, when considered as a system of points in the complex plane, goes over into itself under a rotation of the plane by the angle $2\pi/m$, i.e., these numbers are the roots of the equation $\lambda^m-(\lambda_A)^m=0$. If m > 1, by simultaneously permuting its rows and columns, A can be put in the following "cyclic" form:

$$A=\begin{bmatrix} 0 & A_{12} & 0 & \cdots & 0 \\ 0 & 0 & A_{23} & \cdots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots\ldots \\ 0 & 0 & 0 & \cdots & A_{m-1,m} \\ A_{m1} & 0 & 0 & \cdots & 0 \end{bmatrix},$$

where there are zero square blocks along the main diagonal.

The vector x_A and the number λ_A that appear in this theorem are called the Frobenius vector and the Frobenius eigenvalue of A, respectively. When it is important to stress that x_A is a column vector, it is referred to as the right Frobenius vector of *A*. It is clear that, given an irreducible nonnegative matrix *A*, its eigenvalue λ_A also corresponds to a unique (with an accuracy of up to a scalar factor) positive left eigenvector (row vector) $p_A: p_A A=\lambda_A p_A$ [54]. The vector p_A is called the left Frobenius vector of *A*.

1.2.2. Definition of AFS.

The starting point for the computation of the AFS is a SVD of the spectral data matrix $D\in R^{k\times n}$. Its SVD reads $U\Sigma V^T$ with orthogonal matrices *U*, *V* and the diagonal matrix Σ which contains the singular values σ_i on its diagonal. If *D* is a rank-*s* matrix, then it holds that $D=U\Sigma V^T=\tilde{U}\tilde{\Sigma}\tilde{V}^T$ with

$$\tilde{U} = U(:,1:s) \in R^{k\times s}, \tilde{\Sigma} = \Sigma(1:s,1:s) \in \mathbb{R}^{s\times s}, \tilde{V} = V(:,1:s) \in \mathbb{R}^{n\times s}.$$

The so-called abstract factors $\tilde{U}\tilde{\Sigma}$ and $\tilde{V}^T$ are usually poor approximations of the matrices *C* and *A*. A proper regular transformation by $T \in R^{s\times s}$ allows one to solve the reconstruction problem according to

$$C = \tilde{U}\tilde{\Sigma}T^{-1}, \quad A = T\tilde{V}^T. \tag{1.9}$$

Any pair of nonnegative matrices $\tilde{U}\tilde{\Sigma}T^{-1}$ and $T\tilde{V}^T$ is called a feasible solution. A feasible solution guarantees a correct reconstruction since $D = \left(\tilde{U}\tilde{\Sigma}T^{-1}\right)\left(T\tilde{V}^T\right)$. However, these factors may have no physical meaning.

For a two-component system Lawton and Sylvestre have represented the range of feasible solutions. For a three-component system the situation is more complicated but the purpose is still the same. All regular matrices *T* are to be found so that *C* and *A* in (1.9) are nonnegative matrices. The coefficients of *T* are the key for a low-dimensional representation of the AFS.

The Perron-Frobenius theory guarantees that the first singular vector *V*(:, 1) of *V* can be assumed to be a component-wise nonnegative vector; possibly a multiplication with −1 is to be applied to give a component-wise non-positive vector of the desired orientation. With (1.9) the i^{th} pure component spectrum *A*(*i*, :), *i* = 1, 2, 3, reads

$$A(i,:) = t_{i1}V(:,1)^T + t_{i2}V(:,2)^T + t_{i3}V(:,3)^T. \tag{1.10}$$

Since $V(:,1) \neq 0$ these spectra can be scaled so that t_{i1} = 1 for *i* = 1, 2, 3. Then *T* has still six degrees of freedom namely t_{i2} and t_{i3} with *i* = 1, 2, 3. The problem is forced to two dimensions by looking only for those $\alpha := t_{12}$ and $\beta := t_{13}$ so that

$$T = \begin{pmatrix} 1 & \alpha & \beta \\ 1 & s_{11} & s_{12} \\ 1 & s_{21} & s_{22} \end{pmatrix} \tag{1.11}$$

for proper s_{11}, s_{12}, s_{21} and s_{22} results in a feasible solution.

A point $(\alpha, \beta) \in \mathrm{R}^2$ is called *valid* if and only if there exists at least one regular matrix

$$S = \begin{pmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{pmatrix} \in \mathbb{R}^{2\times 2} \tag{1.12}$$

so that T is invertible and both $A = T\tilde{V}^T$ and $C = \tilde{U}\tilde{\Sigma}T^{-1}$ are nonnegative matrices. Hence the AFS can be expressed as the set

$$\mathcal{M} = \left\{(\alpha, \beta) \in \mathbb{R}^2 : \mathrm{rank}(T) = 3, C, A \geq 0\right\} \tag{1.13}$$

Under some mild assumptions the set $\mathcal{M}$ is bounded. The rows of S and α, β are coupled in the following sense: If $(\alpha, \beta) \in \mathcal{M}$, then the rows $S(i, :)$, $i = 1, 2$, of S are also contained in $\mathcal{M}$. The reason is that an orthogonal permutation matrix $P \in \mathrm{R}^{3\times 3}$ can be inserted in the admissible factorization

$$D = CA = \tilde{U}\tilde{\Sigma}T^{-1}T\tilde{V}^T = (\tilde{U}\tilde{\Sigma}\underbrace{T^{-1}P^T}_{(PT)^{-1}})\left(PT\tilde{V}^T\right).$$

The permutation of the rows of T is accompanied with the associated permutation of the columns of T^{-1} and the nonnegativity of the factors is preserved. Further PT and $T^{-1}P^T = \left(PT\right)^{-1}$ are a pair of transformation matrices with permuted rows/columns in a way that (s_{i1}, s_{i2}) can substitute (α, β) and vice versa.

The AFS for an s-component system is a subset of the R^{s-1} with the form

$$\begin{aligned} \mathcal{M} = \Big\{ & t \in \mathbb{R}^{1\times s-1} : \text{exists invertible } T \in \mathbb{R}^{s\times s} \\ & T(1,:) = (1, t), \tilde{U}\tilde{\Sigma}T^{-1} \geq 0 \text{ and } T\tilde{V}^T \geq 0\Big\} \end{aligned} \tag{1.14}$$

(1.14) implies that the rows a of a feasible factor A can be presented by linear combinations of the rows of $\tilde{V}^T$ in the form

$$a = (1, t) \cdot \tilde{V}^T. \tag{1.15}$$

The fixed 1 in the row vector $(1, t)$ guarantees that *any* spectrum has a contribution from the first right singular vector. This property is by no means evident and has to

be proved. Theorem 1.3 shows that the AFS representation (1.14) is valid if and only if D^TD is an irreducible matrix. For such matrices Theorem 1.4 shows that $\mathcal{M}$ is a bounded set.

Definition 1.3. Let P be an $n \times n$ permutation matrix, i.e. P is a column permutation of the identity matrix. An $n \times n$ matrix H with $n \geq 2$ is called reducible, if a permutation matrix P exists so that

$$PHP^T = \begin{pmatrix} H_{1,1} & H_{1,2} \\ 0 & H_{2,2} \end{pmatrix}.$$

Therein $H_{1,1}$ is an $m \times m$ submatrix and $H_{1,2}$ is an $m \times (n-m)$ submatrix with $1 \leq m < n$. If such a permutation matrix P does not exist, then H is called an irreducible matrix.

The next theorem proves that the Borgen and Kowalski approach (with 1s in the first column of T) is justified. Further, the result is used in Theorem 1.4 on the boundedness of the AFS.

Theorem 1.3. Let $D \in R^{k \times n}$ be a nonnegative matrix with *rank(D) = s* which has no zero column. Further let $U\Sigma V^T$ be a singular value decomposition of D and let $\tilde{V}$ be the submatrix of V formed by its first s columns. There exists a vector $t \in \mathbb{R}^{1 \times s-1} \setminus \{0\}$ with

$$(0,t) \cdot \tilde{V}^T \geq 0 \qquad (1.16)$$

(in words: any linear combination of the columns 2,...,s of $\tilde{V}$ has negative components) if and only if D^TD is an irreducible matrix.

Corollary 1.3. Let D^T satisfy the assumptions of Theorem 1.3. Then no $v \in \mathbb{R}^{s-1}$ exists with $\tilde{U}\tilde{\Sigma}\begin{pmatrix} 0 \\ v \end{pmatrix} \geq 0$ if and only if DD^T is irreducible.

The matrix D^TD can be assumed to be irreducible for spectroscopic applications. Otherwise, the series of spectra decomposes into apparently separated or noncoupled subblocks. A trivial example of a reducible matrix is the 3-by-3 identity matrix $D = I_3$ so that $D^TD = I_3$ for which T is not necessarily in the form (2) (since $V = T = T^{-1} = C = A = I_3 \geq 0$ is a feasible solution).

An important consequence of Theorem 1.3 is that the AFS is a bounded set. The AFS $\mathcal{M}$ is a subset of

$$\mathcal{M}^+ = \left\{ t \in \mathbb{R}^{1\times s-1} : (1,t)\tilde{V}^T \geq 0 \right\}, \tag{1.17}$$

which is closely related to FIRPOL in [78]. The set $\mathcal{M}^+$ stands for the non-negativity of the spectral factor *A* only, and $\mathcal{M}^+$ is the intersection of the *n* half-spaces

$$\left\{ t \in \mathbb{R}^{1\times s-1} : t\tilde{V}(i,2:s)^T \geq -\tilde{V}(i,1) \right\}, \quad i = 1,\ldots,n. \tag{1.18}$$

The next theorem shows that $\mathcal{M}$ and $\mathcal{M}^+$ are bounded sets for irreducible D^TD.

Theorem 1.4. Let *D* satisfy the assumptions of Theorem 1.3. Then $\mathcal{M}^+$ by (1.17) and $\mathcal{M}$ are bounded if and only if D^TD is an irreducible matrix.

With few additional assumptions one can show that the set $\mathcal{M}$ does not include the origin (i.e. the zero vector).

Theorem 1.5. Let $D \in R^{k\times s}$ be a nonnegative rank-s matrix so that D^TD and DD^T are irreducible matrices and that a factorization $D = CA$ with nonnegative factors exists. Then $0 \notin \mathcal{M}$.

Further, the first left singular vector $\tilde{U}(:,1)$ is not the concentration profile of a pure component and the first right $\tilde{V}(:,1)$ is not the spectrum of one of the pure components.

1.3. The polygon inflation algorithms.

For the computation of the AFS a procedure to classify points $(\alpha, \beta) \in \mathrm{R}^2$ as *valid*, if $(\alpha, \beta) \in \mathcal{M}$, or as *non-valid* in the other case has to be used. A procedure for this classification is developed next. Let $\varepsilon \geq 0$ be a small nonnegative real number. Then $-\varepsilon$ is used as a lower bound for the acceptable relative negativeness of the factors *C* and *A* in the following way

$$\frac{\min_j C_{ji}}{\max_j |C_{ji}|} \geq -\varepsilon, \quad \frac{\min_j A_{ij}}{\max_j |A_{ij}|} \geq -\varepsilon, \quad i = 1,2,3. \tag{1.19}$$

The acceptance of small negative components of C and A allows one to stabilize the computational process in the case of noisy data.

Let f be a target function which depends on the six degrees of freedom being α, β and $S \in \mathrm{R}^{2\times 2}$, see (1.11) and (1.12), so that

$$f : R \times R \times R^{2\times 2} \to R \text{ with}$$

$$f(\alpha,\beta,S) = \sum_{i=1}^{3}\sum_{j=1}^{k} \min\left(0, \frac{C_{ji}}{\|C(:,i)\|_\infty} + \varepsilon\right)^2 + \tag{1.20}$$

$$+ \sum_{i=1}^{3}\sum_{j=1}^{n} \min\left(0, \frac{A_{ij}}{\|A(i,:)\|_\infty} + \varepsilon\right)^2 + \left\|I_3 - TT^+\right\|_F^2 .$$

Therein C and A are formed according to (1.9), $I_3 \in \mathrm{R}^{3\times 3}$ is the 3×3 identity matrix, $\|\cdot\|_\infty$ is the maximum vector norm and $\|\cdot\|_F$ is the Frobenius matrix norm. Further T^+ is the pseudo-inverse of T. The last summand $\left\|I_3 - TT^+\right\|_F^2$ equals zero if T is an invertible matrix and is positive if T is singular; therefore $f = 0$ guarantees a regular T. The function f is used to form F as follows

$$F : \mathbb{R}^2 \to \mathbb{R}, \quad F(\alpha,\beta) = \min_{S\in\mathbb{R}^{2\times 2}} f(\alpha,\beta,S). \tag{1.21}$$

Computationally a point (α, β) is considered as valid if and only if $F(\alpha,\beta) \leq \varepsilon_{tol}$ *with* $\varepsilon_{tol} = 10^{-10}$. Hence,

$$\mathcal{M} = \left\{(\alpha,\beta) \in \mathbb{R}^2 : F(\alpha,\beta) \leq \varepsilon_{\mathrm{tol}}\right\}. \tag{1.22}$$

The evaluation of F requires the solution of a least-squares problem within 4 parameters and with $3(k + n + 3)$ variables. Our function F given in (1.21) is somewhat different from the pure sum of squares $ssq = \left\|D - C_+ A_+\right\|_F^2$; therein C_+ and

A_+ are derived from C and A by removing any negative entries. However, we prefer to use (1.21) for the reason of its numerical stability and as (1.21) requires a minimization of a sum of only O($k + n$) squares. In contrast to this, the minimization of ssq includes the much larger number of O($k \cdot n$) summands of squares. Here, the number of components is fixed at $s = 3$. If the approach is generalized to larger s, then the computational costs increase linearly in s.

The orientation of the AFS $\mathcal{M}$ depends on the orientation of the singular vectors. The orientation of a singular vector means that the sign of a singular vector is not uniquely determined in the sense that the simultaneous multiplication of the i^{th} left singular vector and the i^{th} right singular vector with -1 does not change the product $U\Sigma V^T$. However, the orientation of the first left singular vector and the first right singular vector can be fixed in advance by the Perron-Frobenius theory as these two vectors are sign-constant and can therefore be assumed in a component-wise nonnegative form [99]. In other words the SVD $U\Sigma V^T$ with $U \in \mathrm{R}^{k\times 3}$, $V \in \mathrm{R}^{n\times 3}$ is equivalent to the SVD $\hat{U}\Sigma\hat{V}^T$ with

$$\hat{U}(:,1:3) = U(:,1:3)\cdot \mathrm{diag}\left(1, p_1, p_2\right)$$

$$\hat{V}(:,1:3) = V(:,1:3)\cdot \mathrm{diag}\left(1, p_1, p_2\right)$$

and $p_1, p_2 \in \{-1, 1\}$. The signs of p_i are associated with a reflection of the AFS along the α- or the β-axes.

Here, the AFS $\mathcal{M}$ is related to feasible matrices representing the pure component spectra. If the AFS for the concentration factor C is of interest, then the whole procedure can be applied to the transposed data matrix D^T.

The algorithm starts with the construction of an initial triangle which is a first coarse approximation of a topologically-connected subset of the AFS. Therefore, an admissible factorization $D = CA$ with nonnegative factors C and A is needed. This factorization can be computed by any nonnegative matrix factorization tool. According to (1.10) the first row $A(1, :)$ reads

$$A(1,:) = V(:,1)^T + \alpha^{(0)}V(:,2)^T + \beta^{(0)}V(:,3)^T$$

Hence $(\alpha^{(0)}, \beta^{(0)}) \in \mathcal{M}$ are determined by

$$T(1,:) = \left(t_{11}, t_{12}, t_{13}\right) = A(1,:) \cdot V \text{ and } \alpha^{(0)} = \frac{t_{12}}{t_{11}}, \quad \beta^{(0)} = \frac{t_{13}}{t_{11}} .$$

This interior point ($\alpha^{(0)}$, $\beta^{(0)}$) is the basis for the construction of the three vertices P_1, P_2, P_3 of the initial triangle on the boundary $\partial\mathcal{M}$ of $\mathcal{M}$. Since $\mathcal{M}$ is a bounded set, P_1 and P_2 can be determined on the straight line along the α-axis through ($\alpha^{(0)}$,$\beta^{(0)}$) having the form

$$x = \begin{pmatrix} \alpha^{(0)} \\ \beta^{(0)} \end{pmatrix} + \gamma \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Hence, $\gamma \geq 0$ for P_1 and $\gamma \leq 0$ for P_2, see **Figure 1.1** for the construction. Then P_3 is one point of intersection of the mid-perpendicular of the line segment P_1P_2 having the form

$$x = M + \tilde{\gamma} v, \quad M = \frac{1}{2}\left(P_1 + P_2\right), \quad v \perp \overline{P_1P_2} .$$

Without loss of generality $\tilde{\gamma} \leq 0$ can be assumed.

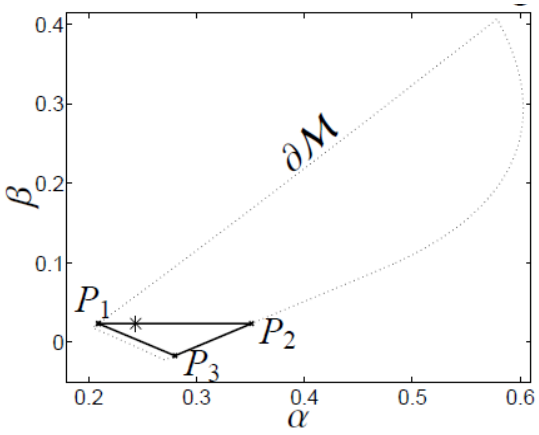


Figure 1.1. *Computation of an initial triangle in $\mathcal{M}$. Dotted line: Boundary of a subset of $\mathcal{M}$. Bold line: the initial triangle. Asterisk: Initial point ($\alpha^{(0)}$,$\beta^{(0)}$) = (0.2438, 0.0235) [78].*

The edges of the initial triangle and also the edges of refined polygons are subdivided by introducing new vertices in a way that the refined polygon is a better approximation of the AFS.

Next the adding of a new vertex is explained. Therefore, let the *m*-gon *P* with the vertices (P_1,...,P_m) be given. Then *P* is inflated to an (*m* + 1)-gon *P′* with the

vertices (P'_1,...,$P'm$+1). If the edge between P_i and P_{i+1} is selected for the refinement, then the new vertex P'_{i+1} is a point of intersection of the mid-perpendicular of the edge P_iP_{i+1} and the boundary $\partial\mathcal{M}$. The refined polygon has the vertices

$$\left(P_1', P_2', \ldots, P_{m+1}'\right) = \left(P_1, P_2, \ldots, P_i, P_{i+1}', P_{i+1}, \ldots, P_m\right).$$

If P approximates a topologically connected *convex* subset $\mathcal{M}$, then the new vertex P'_{i+1} is located not in the interior of P so that the new polygon P' contains P as a subset. In case of a concave boundary element the new polygon P' may have a smaller area than P. The mid-perpendicular of the edge $\overline{P_iP_{i+1}}$ has the form

$$M + \gamma v, \quad \gamma \in \mathbb{R} \text{ with } M = \frac{1}{2}\left(P_i + P_{i+1}\right), \quad v \perp \overline{P_iP_{i+1}}. \tag{1.23}$$

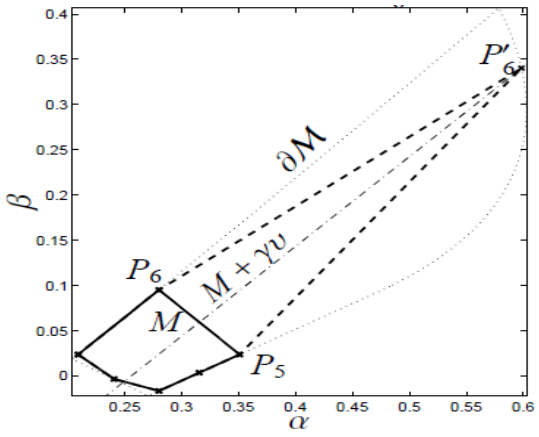


Figure 1.2. *Adding of the vertex* P_6' *which is located on the intersection of the mid-perpendicular through* P_5 *and* P_6 *and the boundary of* $\mathcal{M}$ *[78].*

The point of intersection of the straight line (1.23) and $\partial\mathcal{M}$ is not unique (there are two or more points of intersection); the new vertex P_{i+1}' is determined in a way that the Euclidean distance to M is minimized and that the polygon is not dissected into two parts (to avoid to find a new vertex on the opposite side of the polygon, i.e. $\overline{P_{i+1}'M}$ dissects P). **Figure 1.2** illustrates the refinement of a 6-gon to a 7-gon.

The accuracy of a new vertex depends on the function F which is to be minimized along the straight line (1.23). Numerically we use the relatively slow

converging bisection method for the root finding because of its simplicity and robustness. The iteration is stopped if a final accuracy ε_b is reached so that

$$P'_{i+1} \in \mathcal{M}, \quad \min_{x \notin \mathcal{M}} \left\| P'_{i+1} - x \right\|_2 < \varepsilon_b . \tag{1.24}$$

The number of iterations depends on ε_b and on the length $\left\| P_i - P_{i+1} \right\|_2$ of the edge. In our numerical calculations between 3 iterations (for $\varepsilon_b = 10^{-2}$) and 8 iterations (for $\varepsilon_b = 10^{-5}$) were needed to determine a vertex P'_{i+1}.

An adaptive process is used to determine those edges of the polygon whose subdivision promises to improve the approximation of the AFS in the best way. Next a selection strategy is introduced together with a termination criterion.

The central quantity which steers the refinement process is the change-of-area of the polygon which arises if an edge is subdivided. So if an edge $\overline{P_i P_{i+1}}$ is subdivided, then each of the new edges gets an equally weighted gain-of-area

$$\Delta_i = \frac{1}{4} \left\| P_i - P_{i+1} \right\|_2 \left\| M - P'_{i+1} \right\|_2 \tag{1.25}$$

as an attribute. In the next step an edge ℓ is selected for which $\ell = \arg\max_j \Delta_j$ in order to determine an edge which promises a maximal gain-of-area on the basis of its subdivision history. If there is no unique index ℓ, then the algorithm starts with the smallest index. As for the initial triangle no subdivision history is available, all three initial edges are subdivided at the beginning. The refinement process is stopped if the largest achievable gain-of-area drops below some final accuracy δ. The actual value of δ may depend on the problem. It is often used $\delta = \varepsilon_b$.

The polygon inflation algorithm works well for non-perturbed, as well as for noisy data. The parameter ε in (1.18) controls the allowance of relative negative contributions in C and A and, in our experiments, appears to cause a favorable numerical stability with respect to perturbations.

However, the noise level must be limited in a way that the first three singular vectors $V(:, i)$, $i = 1, 2, 3$, still contains the essential information on the system. If this is not guaranteed, then the expansion (1.10) cannot guarantee for a proper

reconstruction of C and A. Then even no regular transformation T may exist so that C and A are nonnegative matrix factors [78].

Three characteristic traits of the polygon inflation algorithm are compiled next and are compared with the triangle inclusion method.

1. The polygon inflation algorithm which uses the function (1.18) has to minimize sums of only $O(k+n)$ squares. In contrast to this the function ssq includes $O(kn)$ squares. (It is worth noting that by the definition of the Landau symbol it holds that $O(k + n) = O(s(k + n))$ where s is the number of components which equals 3 throughout this thesis.)

2. Negative entries of C and A larger than $-\varepsilon$ are not completely ignored in the polygon inflation algorithm but affect the minimum of F, see (1.21).

3. The polygon inflation algorithm results in a piecewise linear interpolation of the boundary of $\mathcal{M}$ by polygons. The local approximation error of a linear interpolation behaves like $O(h^2)$ if the nodes of the interpolant are assumed to be exact. In contrast to this, the enclosure of the boundary by a chain of equilateral triangles with the edge-length h results in a final accuracy which is bounded by the width $O(h)$ of this chain.

Further, the local adaptivity of the polygon inflation scheme even requires a small number of refinement steps if the boundary is locally more or less a straight line. A critical non-smooth region of the boundary can be resolved to any desired accuracy. This adaptive resolution of the boundary results in a cost-effective computational procedure. In contrast to this, the number of triangles needed for the triangle inclusion algorithm increases as $O(1/h)$ in the edge length h of the triangles.

Parameters of the polygon inflation algorithm are:

1. The parameter ε in (1.17) controls the degree of acceptable negative entries in the columns of C and the rows of A. Negative matrix elements are not penalized in (1.18) if their relative magnitude is larger than $-\varepsilon$. This parameter should be increased with growing perturbations in the spectral data. In our experience $0 \leq \varepsilon \leq 0.05$ seems to be working properly. For model problems and in absence of any errors $\varepsilon = 0$ can be used. By construction increasing ε enlarges the AFS.

2. The parameter ε_b in (1.24) controls the quality of the boundary approximation of the AFS. It is used $\varepsilon_b \leq 10^{-3}$ and sometimes $\varepsilon_b \leq 10^{-4}$. The influence of this parameter on the shape and size of the computed AFS is negligible.

3. The parameter δ defines a stopping criterion for the adaptive polygon refinement. If the largest gain-of -area (1.25) is smaller than δ, then the refinement can be stopped. It is often set $\delta = \varepsilon_b$ and state that the shape and size of the computed AFS is not sensitive to changes of δ.

Each step of an iterative minimization of F by (1.21) includes the solution of a nonlinear optimization problem. For a poorly conditioned problem the numerical solutions will scatter around the exact solution. Hence, a new vertex P'_{i+1} might be located in the interior of the AFS in the following sense $\min_{x \notin \mathcal{M}} \left\| P'_{i+1} - x \right\|_2 \geq \varepsilon_b$.

With such an inaccurate vertex the further refinement steps can result in a nonsmooth boundary which may even contain needles directing towards the inside of the AFS. To avoid such misplaced boundary points, the powerful optimization procedure NL2SOL is used and the iterative minimization has to start with a good initial guess [71]. A reasonable initial guess can be a convex combination of the numerical solutions which have previously been gained for nearby points. Further, it is applied some decision tree before accepting points as valid. Nevertheless, misplaced boundary points can be detected by looking for obtuse angles along the edges of the polygon. Then, suspicious vertices may be removed and the optimization can be restarted.

If parts of the boundary of two isolated subregions of the AFS are in close proximity, then the numerical algorithm tends to agglutinate these regions to a joint connected subset. However, for most of the practical problems the subsets of the AFS appear to be well separated.

As it was mentioned before the AFS is a bounded set which does not include the origin. A challenging question is: *What is the number of isolated segments an AFS may consist of?*

For s = 2-component systems the AFS consists of p = 2 separated intervals which are taken as the sides of a rectangular for its presentation. For s = 3-component systems experimental data and model data show that a number of p = 1, p = 3 or even p = 6 segments may occur.

It is well known that the AFS for three-component systems may consist of only one segment and that this segment can contain a hole, which surrounds the origin. Next it is described a variation of the polygon inflation algorithm which can be used to compute such AFS with a hole or an AFS with more than three isolated segments.

If for a three-component systems (s = 3) the entire AFS consists of one segment with a hole, then the triangle-enclosure algorithm and the polygon inflation algorithm are to be modified properly. For the triangle-enclosure algorithm two runs are necessary in order to cover the interior and the exterior boundary curve by sequences of triangles.

For the polygon inflation algorithm the exterior polygon and the interior polygon are to be treated differently but the geometric concept of inflation polygon is in each of these cases the same. Only the objective functions are changed.

First, the computational effort to compute the exterior polygon is very small since only $(1,t)\tilde{V}^T \geq 0$ is to be tested according to Equation (1.17). This exterior polygon is just the boundary of the set $\mathcal{M}^+$ defined in (1.17). The remaining conditions on t to be a valid vector, i.e. $t \in \mathcal{M}$, are used to define a further set

$$\mathcal{M}^* = \left\{ t \in \mathbb{R}^{s-1} : \min_S \| f(t,S) \|_2^2 = 0 \right\} \tag{1.26}$$

whose inner boundary is computed by the standard polygon inflation algorithm. For $t \in \mathcal{M}^*$ the definition of $\mathcal{M}^*$ guarantees that T is regular, $C \geq 0$ and $A(2:3,:) \geq 0$. The intersection of $\mathcal{M}^+$ and $\mathcal{M}^*$, which combines the conditions, results in the AFS

$$\mathcal{M} = \mathcal{M}^+ \cap \mathcal{M}^* . \tag{1.27}$$

In order to avoid any misinterpretation it is mentioned that $\mathcal{M}^*$ is very different to INNPOL as used in Ref. [88].

The algorithm to compute a polygon which approximates the boundary of $\mathcal{M}^+$ uses an objective function which guarantees (1.18) to hold. The starting point is the origin which is always in $\mathcal{M}^+$. The polygon inflation starts with a triangle enclosing the origin and whose vertices are located on the boundary of $\mathcal{M}^+$.

After this the interior boundary of $\mathcal{M}^*$ is computed by using the objective function

$$f : \mathbb{R}^{S-1} \times \mathbb{R}^{(s-1)\times(s-1)} \to \mathbb{R}^{ks+n(s-1)+1}. \qquad (1.28)$$

Therefore, the complement $\mathbb{R}^2 \setminus \mathcal{M}^*$ is approximated from the interior of this set. The starting point is, once again, the origin since Theorem 1.5 guarantees that

$$(0,0) \notin \mathcal{M}^*.$$

The use of the complement is the reason why we call the algorithm an inverse polygon inflation. The computation of the hole of the AFS by applying the polygon inflation to $\mathbb{R}^2 \setminus \mathcal{M}^*$ has the advantage that only few lines of program code are to be adapted. Further, the relevant regions of the two sets $\mathcal{M}^+$ and $\mathcal{M}^*$ can be computed in a stable way.

If *FAC-PACK* uses the standard polygon inflation algorithm and finds an AFS segment which has a nonzero intersection with at least three quadrants of the Cartesian coordinate system, then the algorithm automatically switches to the inverse polygon inflation. Thus a one-segment AFS is automatically computed by inverse polygon inflation.

1.3.1. Traces of the spectral data matrix.

The traces of spectral data matrix *D* can be drawn in $\mathcal{M}_A$ and $\mathcal{M}_C$. The traces in $\mathcal{M}_A$ are the normalized expansion coefficients of the rows of *D* with respect to the right singular vectors *V*(:, 2) and *V*(:, 3). The normalization is that the expansion coefficient for *V*(:, 1) equals 1. Thus the trace $\mathcal{M}_A$ is given by the *k* points

$$w_i = \left(\frac{(DV)_{i2}}{(DV)_{i1}}, \frac{(DV)_{i3}}{(DV)_{i1}} \right) = \left(\frac{D(i,:)\cdot V(:,2)}{D(i,:)\cdot V(:,1)}, \frac{D(i,:)\cdot V(:,3)}{D(i,:)\cdot V(:,1)} \right), \quad i = 1,\ldots,k\,.$$

Analogously, the traces of *D* in $\mathcal{M}_C$ are the normalized expansion coefficients of the columns of *D* with respect to the scaled left singular vectors $\sigma_2^{-1}U(:,2)$ and $\sigma_3^{-1}U(:,3)$

$$u_j = \left(\frac{\left(\Sigma^{-1}U^T D\right)_{2j}}{\left(\Sigma^{-1}U^T D\right)_{1j}}, \frac{\left(\Sigma^{-1}U^T D\right)_{3j}}{\left(\Sigma^{-1}U^T D\right)_{1j}} \right) = \left(\frac{\sigma_1 U(:,2)^T \cdot D(:,j)}{\sigma_2 U(:,1)^T \cdot D(:,j)}, \frac{\sigma_1 U(:,3)^T \cdot D(:,j)}{\sigma_3 U(:,1)^T \cdot D(:,j)} \right), j = 1,\ldots,n.$$

1.3.2. Borgen Plots.

Borgen plots are representations of the AFS which are generated by means of a geometric construction.

In FAC-PACK the generalized Borgen plots can be computed for two different scalings.

I. The Row Sum scaling (RS-scaling) scales the rows of data matrix *D* and spectral matrix factor *A* so that each row sum of these two matrices is equal to 1. For the case of nonnegative matrices the row sum scaling is equivalent to a normalization of rows with the 1-norm.

II. The First Singular Vector scaling (FSV-scaling) is a scaling which makes the expansion coefficient of each row of *A* with respect to the normalized first right singular vector of *A* equal to 1. The FSV-scaling has predominantly been used in the context of numerical methods to compute the AFS.

1.3.3. Weakly separated subsets of AFS.

If the segments of an AFS are only weakly separated (in a sense that the polygon inflation algorithm tends to glue separated segments of the AFS to a joint segment), then the numerical computation of $\mathcal{M}^+$ and $\mathcal{M}^*$, which is followed by their intersection, is a stable and favorable way to construct $\mathcal{M}$. The inverse polygon inflation procedure can even be applied to general situations with well separated segments - a situation we have often found for FT-IR spectral data. However, the computational procedure for the inverse polygon inflation is somewhat more expensive as compared to the direct computation of the three separated segments.

1.3.4. Straight-line segments of AFS.

An isolated segment of the AFS is most often either a set whose (mathematical surface) area is larger than zero or it is a single point. In some cases an isolated subset of the AFS appears to be a straight-line segment. In absence of rounding errors and perturbations its surface area equals zero; for slightly perturbed data such a segment practically is a long and narrow band. The polygon inflation method needs some algorithmic enhancement in order to compute such straight-line segments.

In *FAC-PACK* it is used an angle-search method for approximating such AFS segments. The starting point is a feasible coordinate $x = (\alpha, \beta)$ as computed by the NMF. Together with a small radius parameter r the function

$$g_{r,x}(\varphi) = x + r\begin{pmatrix}\sin(\varphi)\\ \cos(\varphi)\end{pmatrix}$$

is considered in order to compute a feasible angle φ so that $g_{r,x}(\varphi) \in \mathcal{M}$.

The numerical minimization of the objective function (1.25) is executed only if the rapid test (1.29) is passed, where the rapid test is

$$-t \cdot V(i, 2:s)^T \le V(i,1), \quad i = 1, \ldots, n\,. \tag{1.29}$$

The initial x might be one of the endpoints of the line segment or between them. In the latter case and if φ represents a feasible direction, then also $\varphi - \pi$ stands for a feasible direction. For these two oppositely oriented directions maximal values r_l and r_r are computed by the bisection method so that the desired line segment equals the union $\mathcal{L}_l \cup \mathcal{L}_r$ with

$$\mathcal{L}_l = \left\{ x + r\begin{pmatrix}\sin(\varphi-\pi)\\ \cos(\varphi-\pi)\end{pmatrix} \text{ with } r \in [0, r_l] \right\} \quad \mathcal{L}_r = \left\{ x + r\begin{pmatrix}\sin(\varphi)\\ \cos(\varphi)\end{pmatrix} \text{ with } r \in [0, r_r] \right\}.$$

2. THE MATHEMATICS OF PYFITIT

2.1. Principal component analysis (PCA): definition and goals.

The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data.

The goals of PCA are:

1. to extract the most important information from the data table;
2. to compress the size of the data set by keeping only this important information;
3. to simplify the description of the data set;
4. to analyze the structure of the observations and the variables;
5. to compress the data, by reducing the number of dimensions, without much loss of information;
6. to use in image compression.

2.1.1. Correlation Coefficient and Matrix in PCA.

Correlation Coefficient ρ is a measure of the linear relationship between two random variables ($-1 \le \rho \le 1$).

If the correlation between two variables is positive, then an increase (decrease) in the value of one variable corresponds to an increase (decrease) in the value of the other. Similarly, a negative correlation would mean that an increase (decrease) in the value of one variable will correspond to an decrease (increase) in the value of the other. In the case of independence when there is no relation between

two variables, the correlation is zero. The correlation coefficient denoted by ρ, is computed by

$$\rho_{XY} = corr(X,Y) = \frac{\text{cov}(X,Y)}{\sqrt{\text{var}(X)\,\text{var}(Y)}}. \tag{2.1}$$

Statistically

$$corr(X,Y) = r_{XY} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}. \tag{2.2}$$

Let $X = (X_1, \ldots, X_n)^T$ be an *n*-dimensional random sample, the correlation between random variables X_i and X_j is denoted by $r_{x_i x_j}$ and given by

$$corr(X_i, X_j) = r_{x_i x_j} = \frac{\sum_{k=1}^{n}(x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{n}(x_{ik} - \overline{x}_i)^2 \sum_{k=1}^{n}(x_{jk} - \overline{x}_j)^2}}.$$

Obtained $r_{x_i x_j}$ values can be represented in $n \times n$ matrix form

$$\mathbf{R} = \begin{pmatrix} r_{x_1x_1} & \cdot & \cdot & r_{x_1x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ r_{x_nx_1} & \cdot & \cdot & r_{x_nx_n} \end{pmatrix}. \tag{2.3}$$

2.2. Mathematics of PCA.

PCA is a procedure that seeks an *r-dimensional* basis that best captures the variance in the data. The vector that has the largest variance is called the first principal component. The orthogonal vector that captures the second largest variance is called the second principal component, and so on.

Prior to starting the PCA procedure, data are often pre-treated to transform them into a suitable form for the analysis.

Variables frequently have different numerical units, and different ranges. For example when there are two variables, the first one being a persons' weight and the second variable the height, the weight has a large range so it has a large variance, but the height has a small range, then it has small variance. Since PCA is a method of maximum variance projection, it follows that the variable which has large variance will contribute more than the variable with low-variance [29].

In the data matrix each element of a column is divided by the column standard deviation, see **Figure 2.1**.

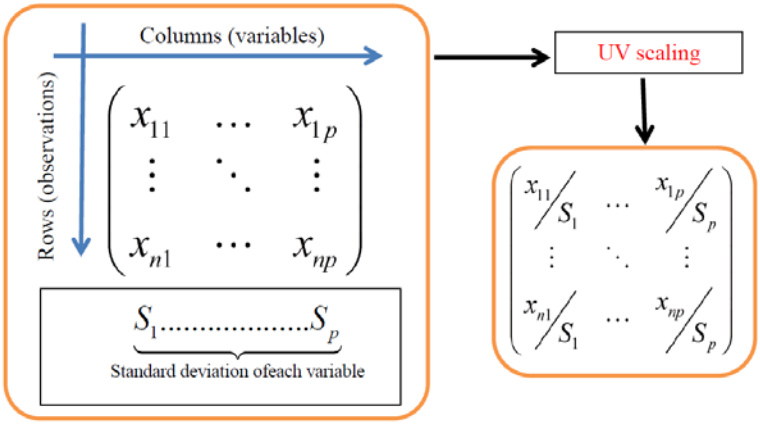


Figure 2.1. *Unit Variance (UV) scaling processing [2].*

Let $\mathbf{X}$ be $n \times p$ data matrix (p variables and n observations). The "center of gravity" of the columns is a vector $\bar{\mathbf{x}} = \left(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p\right)$ in $\mathbb{R}^p$ of the means $\bar{x}_j$ of the p variables (columns) which is given by:

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \cdot \\ \cdot \\ \bar{x}_p \end{pmatrix} = n^{-1}\mathbf{X}^T\mathbf{1}_n$$

where $\mathbf{1}_n$ is $n \times n$ unit matrix. The covariance matrix $\mathbf{S}$ can be written as

$$\mathbf{S} = n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X} - \overline{\mathbf{x}\mathbf{x}}^{\mathbf{T}} = n^{-1}\left(\mathbf{X}^{\mathbf{T}}\mathbf{X} - n^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{1}_n\mathbf{1}_n^T\mathbf{X}\right) = n^{-1}\mathbf{X}^{\mathbf{T}}\left(\mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^T\right)\mathbf{X} \ .$$

Hence, $\left(\mathbf{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^T\right)$ is a centering matrix denoted by **H.** Rewriting the covariance formula

$$\mathbf{S} = n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{H}\mathbf{X} \tag{2.4}$$

is obtained.

Note that **H** is symmetric and idempotent $\left(\mathbf{H} = \mathbf{H}^2\right)$. Then the standardized data matrix is denoted as $\mathbf{X}_*$ and given by

$$\mathbf{X}_* = n^{-1/2}\mathbf{H}\mathbf{X}\mathbf{D}^{-1/2} \tag{2.5}$$

where $\mathbf{D} = diag\left(s_{X_iX_i}\right)$.

Let $\mathbf{X}_c$ be the centered matrix of $\mathbf{X}$ $n \times p$ data matrix. The SVD of $\mathbf{X}_c$ is given as

$$\mathbf{X}_c = \mathbf{L}\Delta\mathbf{Q}^T. \tag{2.6}$$

The matrix $\mathbf{X}_c^T\mathbf{X}_c$ is defined as

$$\mathbf{X}_c^T\mathbf{X}_c = \left(\mathbf{L}\Delta\mathbf{Q}^T\right)^T\left(\mathbf{L}\Delta\mathbf{Q}^T\right) = \mathbf{Q}\Delta^T\mathbf{L}^T\mathbf{L}\Delta\mathbf{Q}^T = \mathbf{Q}\Delta^T\Delta\mathbf{Q}^T = \mathbf{Q}\Delta_{\mathbf{X}_c}^2\mathbf{Q}^T \tag{2.7}$$

where $\Delta_{\mathbf{x}_c}^2$ is $n \times n$ matrix with diagonal entries δ_i^2 for $i = 1, 2, \ldots, p$.

Since $\mathbf{X}_c$ is a centered data matrix, the covariance matrix is $\mathbf{\Sigma} = \frac{1}{n}\mathbf{X}_c^T\mathbf{X}_c$ by the principal axes theorem in Ref. [64]. This can be decomposed as $\mathbf{\Sigma} = \mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{U}$ then

$$\mathbf{X}_c^T\mathbf{X}_c = n\mathbf{\Sigma} = n\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{U} = \mathbf{U}^{\mathrm{T}}(n\mathbf{\Lambda})\mathbf{U}. \tag{2.8}$$

By (2.7) and (2.8), **Q** (*right singular vectors*) are the same of the eigenvectors of matrix **Σ**, additionally, the singular values of $\mathbf{X}_c$ are related to the eigenvalue of **Σ**.

$$n\lambda_i = \delta_i^2,\ \lambda_i = \frac{\delta_i^2}{n},\ i = 1, 2, \ldots, p.$$

The PCs are obtained from the SVD of the covariance matrix. In the principal component transformation, the estimator μ is replaced by $\bar{x}$ and $\mathbf{\Sigma}$ is replaced by **S**. Spectral decomposition of the covariance matrix can be written as

$$\mathbf{S} = \mathbf{GLG^T}. \tag{2.9}$$

Then the PCs are obtained by

$$Y = \left(X - \mathbf{1}_n \bar{x}^T\right)\mathbf{G} \tag{2.10}$$

where $\mathbf{L} = diag\left(\ell_1, \ell_2, \ldots, \ell_p\right)$ is the diagonal matrix of eigenvalues of **S** and $\mathbf{G} = \left(\mathbf{g_1}, \mathbf{g_2}, \ldots, \mathbf{g_p}\right)$ is a matrix of orthogonal eigenvectors $\mathbf{g_j}$ of **S.**

If all original *p* variables are uncorrelated (orthogonal, independent), then the variables themselves are the PCs. Hence **S** would have the form

$$\mathbf{S} = \begin{pmatrix} s_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_{pp} \end{pmatrix}$$

and the eigenvalues ℓ_j of the covariance matrix **S** will be $\ell_j = s_{jj}$ j=1, 2,..., p. Correspondingly, the normalized eigenvectors $\mathbf{g}_j$ which have 1 in the j^{th} position and zeros elsewhere, are

$$\mathbf{g}_j^T = (0, 0, \ldots, 1, 0, \ldots, 0),\ j = 1, 2, \ldots, p.$$

Thus, the j^{th} PC is $\mathbf{z}_j = \mathbf{g}_j^T\mathbf{X} = \mathbf{x}_j \quad j = 1, 2, \ldots, p$.

As another illustration, in the covariance **S** or correlation matrix **R**, a distinguishing pattern may be identified, from which formulation of the principal components can be deduced. For example, if one of the variables has the highest variance compared with others, this variable will dominate the first component, accounting for the majority of the variance.

Generally, the PCs are computed from **S** rather than **R**, specially if the PCs are used in farther computation. However, in some cases, the PCs will be more interpretable if calculated from **R**.

After centering the data matrix $\mathbf{X}_c = \mathbf{X} - \mathbf{1}_n \bar{\mathbf{x}}^T$, $\mathbf{X}_c^T \mathbf{X}_c$ is the covariance matrix which is used in PCA. When the variables are measured with different units, the data must be standardized by dividing each variable (each column) by the column standard deviation (2.4). In this case $\mathbf{X}_*^T \mathbf{X}_*$ is equal to the correlation matrix **R**. Then the analysis is referred to correlation PCA.

Let $X \sim (\mu, \mathbf{\Sigma})$, $\mathbf{\Sigma} = \mathbf{\Gamma}^T \mathbf{\Lambda} \mathbf{\Gamma}$ and $Y = \mathbf{\Gamma}^T (X - \mu)$ be a linear transformation, then the following properties applies:

1) $EY_j = 0 \quad j = 1, 2, \ldots, p$, $EY_j = E\left(\boldsymbol{\eta}_j^T (X - \mu)\right) = \boldsymbol{\eta}_j^T E(X - \mu) = 0$

2) $Var\left(Y_j\right) = \lambda_j \quad j = 1, 2, \ldots, p$,

$Var\left(Y_j\right) = Var\left(\boldsymbol{\eta}_j^T (X - \mu)\right) = \boldsymbol{\eta}_j^T Var(X) \eta_j = \lambda_j$ by the properties of variance

3) $Cov\left(Y_i, Y_j\right) = 0 \quad i \neq j$, $Cov\left(Y_i, Y_j\right) = E\left(Y_i Y_j\right) - E\left(Y_i\right) E\left(Y_j\right) = 0$

4) Let **S** be the covariance matrix of original variables, and let $\mathbf{Y} = \left(\mathbf{X} - \mathbf{1}_n \bar{x}^T\right) \mathbf{\Gamma}$.

The covariance matrix of the PCs is $\mathbf{S}_Y = \mathbf{\Lambda}$ where $\mathbf{\Lambda} = diag\left(\lambda_1, \lambda_2, \ldots, \lambda_p\right)$ is the eigenvalues of **S**

$$\mathbf{S}_Y = n^{-1} \mathbf{Y}^T \mathbf{H} \mathbf{Y} = n^{-1} \left(\left(\mathbf{X} - \mathbf{1}_n \bar{x}^T\right) \mathbf{\Gamma}\right)^T \mathbf{H} \left(\mathbf{X} - \mathbf{1}_n \bar{x}^T\right) \mathbf{\Gamma} = n^{-1} \mathbf{\Gamma}^T \mathbf{X}^T \mathbf{H} \mathbf{X} \mathbf{\Gamma} = \mathbf{\Gamma}^T \mathbf{S} \mathbf{\Gamma} = \mathbf{\Lambda}.$$

2.2.1. Interpreting the Meaning of the principal components (PCs).

PCA produces two items of basic information for interpreting the results. The first one is the correlation coefficients between the original variables and the PCs which are used to interpret the meaning of the PCs. The second one is each principal component associated with an eigenvalue which converts to the proportion of the variation that is explained by the PC.

The covariance between the original random variable *X* and the PC *Y* is given in Ref. [29] as

$$Cov(X,Y) = E\left(XY^T\right) - E(X)E\left(Y^T\right) = E\left(XY^T\right) =$$
$$= E\left(XX^T\boldsymbol{\Gamma}\right) - \boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{\Gamma} = Var(X)\boldsymbol{\Gamma} = \boldsymbol{\Sigma}\boldsymbol{\Gamma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda} \tag{2.11}$$

where $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^T$ is the covariance matrix; $\boldsymbol{\Lambda} = diag\left(\lambda_1, \lambda_2, \ldots, \lambda_p\right)$ is the eigenvalues; $\boldsymbol{\Gamma} = \left(\boldsymbol{\eta_1}, \boldsymbol{\eta_2}, \ldots, \boldsymbol{\eta_p}\right)$ is a matrix of orthogonal eigenvectors $\boldsymbol{\eta_j}$ of the covariance matrix.

The correlation between each PC and the original variables is denoted by $\rho_{X_iY_j}$ and given by

$$\rho_{X_iY_j} = \frac{\eta_{ij}\lambda_i}{\left(\sigma_{X_iX_i}\lambda_j\right)^{1/2}} = \eta_{ij}\left(\frac{\lambda_i}{\sigma_{X_iX_i}}\right)^{1/2} \quad \begin{matrix} i = 1,2,\ldots,p \\ j = 1,2,\ldots,q \end{matrix} \tag{2.12}$$

Using actual data, (2.12) translates to

$$r_{X_iY_j} = g_{ij}\left(\frac{\ell_j}{s_{X_iX_i}}\right)^{1/2}. \tag{2.13}$$

This correlation coefficient between the random variable *X* and PC is also called "*loading*". Note that sum of squares of loadings is equal to 1.

$$\sum_{j=1}^{p} r_{X_iY_j} = \frac{\sum_{j=1}^{p} \ell_j g_{ij}^2}{s_{X_iX_i}} = \frac{s_{X_iX_i}}{s_{X_iX_i}} = 1 \tag{2.14}$$

Most of the foundations of rotation were developed by Thurstone (1947) and Cattell (1978), who defended the use of rotation to make interpretation of PCs easier and more reliable. After the number of PCs has been selected, an attempt is made to facilitate interpretation and the analysis is often based on a rotation of the selected PCs.

There are two main kinds of rotations, the orthogonal and the oblique rotation.

1) Orthogonal Rotation. An orthogonal rotation method is described by a rotation matrix **R**, where the rows represent the original factors and the columns represent the new (rotated) factors. At the intersection of row *i* and column *j* we have the cosine of the angle θ between the original axis and the new axis.

$$\mathbf{R} = \begin{bmatrix} \cos\theta_{1,1} & \cos\theta_{1,2} \\ \cos\theta_{2,1} & \cos\theta_{2,2} \end{bmatrix} = \begin{bmatrix} \cos\theta_{1,1} & -\sin\theta_{1,1} \\ \sin\theta_{1,1} & \cos\theta_{1,1} \end{bmatrix}.$$

2) VARIMAX. VARIMAX is the most popular orthogonal rotation technique, which was developed by Kaiser (1958). In statistics, VARIMAX rotation means changing of coordinates used in PCA that maximizes the sum of variances of the squared loadings (squared correlations between variables and PCs).

$$v = \sum \left(q_{j,\ell}^2 - \overline{q}_{\ell}^2 \right)^2$$

where $q_{j,\ell}$ is the loading of j^{th} variable of matrix loadings matrix $\mathbf{Q}$ of PC ℓ and $\overline{q}_{\ell}^2$ is the squared mean of loading. VARIMAX simple solution implies that each PC has a small quantity of large loading and a large number of small (or zero) loading.

If the loadings in each column were approximately equal, the variance would be close to 0. As the squared loadings teands 0, the variance will approach a maximum. Thus the VARIMAX technique attempts to make the loadings either large or small to facilitate the interpretation [33].

The VARIMAX is available in most of factor PC analysis software programs, the output usually includes the rotated loading matrix $\mathbf{Q}^*$, the variance accounted for (sum of squares of each column of $\mathbf{Q}^*$), and the orthogonal rotation matrix $\mathbf{R}$ that is used to obtain $\mathbf{Q}^* = \mathbf{QR}$.

3) Oblique Rotation. The aim of using the Oblique Rotation is to get a simple structure by relocation of the factor axes. Oblique rotations relax the orthogonality constraint in order to gain simplicity in the interpretation. Oblique rotations are strongly recommended by Thurstone [92], but are used more rarely than their orthogonal counterparts.

2.3. Spectral unmixing using PyFitIt.

PyFitIt is the software based on the programming language Python. The main method of PyFitIt is the SVD of the experimental XANES data matrix in order to isolate pure XANES spectra and their related concentration profiles.

PyFitIt allows the user to estimate spectra and concentration of components by the usage of a (dynamical) rotational matrix modified by intuitively clear sliders. After this trial step, the obtained spectra and their related concentration profiles have already a chemical/physical meaning and can be directly analysed or used as input for a further refinement based on MCR-ALS algorithm.

We start from the hypothesis that each XANES spectrum, belonging to an experimental dataset $\boldsymbol{\mu}$ and acquired during determined experimental conditions, can be expressed as the weighted sum of *N* uncorrelated "pure" spectra (or main components) **S** for their related concentration profiles **C**:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{m1} & \mu_{m2} & \dots & \mu_{mn} \end{pmatrix} = \mathbf{S} \cdot \mathbf{C} =$$

$$= \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1N} \\ S_{21} & S_{22} & \dots & S_{2N} \\ \dots & \dots & \dots & \dots \\ S_{m1} & S_{m2} & \dots & S_{mN} \end{pmatrix} \cdot \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{N1} & C_{N2} & \dots & C_{Nn} \end{pmatrix} \tag{2.15}$$

Here $\boldsymbol{\mu}$ is characterized by m rows and n columns while matrices **S** and **C** have respectively the following dimensions $(m \times n)$ and $(n \times n)$. In order to realize the decomposition reported in equation (2.15) three main steps are needed.

First step:

The experimental dataset is decomposed through the SVD procedure:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \dots & \dots & \dots & \dots \\ \mu_{m1} & \mu_{m2} & \dots & \mu_{mn} \end{pmatrix} = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{V} = \tag{2.16}$$

$$= \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1m} \\ U_{21} & U_{22} & \dots & U_{2m} \\ \dots & \dots & \dots & \dots \\ U_{m1} & U_{m2} & \dots & U_{mm} \end{pmatrix} \cdot \begin{pmatrix} s_{11} & 0 & \dots & 0 \\ 0 & s_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & s_{nn} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \dots & \dots & \dots & \dots \\ V_{n1} & V_{n2} & \dots & V_{nn} \end{pmatrix}$$

Where **U** ($m \times m$) and **V** ($n \times n$) are orthogonal matrices formed, respectively, by the eigenvectors of matrixes $\boldsymbol{\mu} \cdot \boldsymbol{\mu}^t$ and $\boldsymbol{\mu}^t \cdot \boldsymbol{\mu}$. **Σ** is a rectangular matrix whose elements are the positive square roots of the eigenvalues of $\boldsymbol{\mu} \cdot \boldsymbol{\mu}^t$ (or $\boldsymbol{\mu}^t \cdot \boldsymbol{\mu}$), called singular values, sorted in descending order.

Second step:

After that the correct number (i.e. *N*) of significant singular values has been identified the remaining $n - N$ smaller elements of **Σ** are set to zero. This choice determines the dimensional reduction of matrices **U** and **V** from ($m \times m$) and ($n \times n$) to ($m \times N$) and ($N \times n$). For example, if two components are required (i.e. *N*=2), equation (2.16) acquires the following form:

$$\overline{\boldsymbol{\mu}} = \mathbf{U} \cdot \overline{\boldsymbol{\Sigma}} \cdot \mathbf{V} =$$

$$= \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1m} \\ U_{21} & U_{22} & \dots & U_{2m} \\ \dots & \dots & \dots & \dots \\ U_{m1} & U_{m2} & \dots & U_{mm} \end{pmatrix} \cdot \begin{pmatrix} s_{11} & 0 & 0 & \dots \\ 0 & s_{22} & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \dots & \dots & \dots & \dots \\ V_{n1} & V_{n2} & \dots & V_{nn} \end{pmatrix} = \tag{2.17}$$

$$= \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \\ \dots & \dots \\ U_{m1} & U_{m2} \end{pmatrix} \cdot \begin{pmatrix} s_{11} & 0 \\ 0 & s_{22} \end{pmatrix} \cdot \begin{pmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \end{pmatrix} = \overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}} \cdot \overline{\mathbf{V}} .$$

Where $\overline{\mathbf{U}}, \overline{\boldsymbol{\Sigma}}$ and $\overline{\mathbf{V}}$ are respectively the dimensional reduced form of **U**, **Σ** and **V**.

According to the Eckart-Young theorem, it is possible to assert that matrix $\overline{\boldsymbol{\mu}}$ is the best approximation of $\boldsymbol{\mu}$ among all other matrices with rank *N* [107]. If the new matrix $\overline{\boldsymbol{\mu}}$ is indistinguishable from $\boldsymbol{\mu}$ within the experimental noise, then the experimental set of data can be described by combination of only *N* components. Otherwise, the number of components must be increased by adding the next diagonal elements in $\boldsymbol{\Sigma}$. It is worth to mention that the variables $\overline{\mathbf{U}}, \overline{\boldsymbol{\Sigma}}$ and $\overline{\mathbf{V}}$ here reported correspond to the **U**, $\boldsymbol{\Sigma}$ and **V** factors present in equation (2.15).

Third step:

The third step foresees the factorization of equation (2.17) in (2.15) by the following relation: $\mathbf{S} = \overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}}$ and $\mathbf{C} = \overline{\mathbf{V}}$. Under this representation matrix $\overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}}$ ($m \times N$) becomes a data matrix composed by *N* spectral profiles while $\overline{\mathbf{V}}$ contains its related concentration profiles. However, matrices $\overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}}$ and $\overline{\mathbf{V}}$ do not have any chemical/physical meaning. This problem can be overcome by the introduction of a "transformation" matrix **T** in equation (2.17) in the following way:

$$\overline{\boldsymbol{\mu}} = \overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}} \cdot \mathbf{T} \cdot \mathbf{T}^{-1} \cdot \overline{\mathbf{V}} \tag{2.18}$$

In this last equation it is possible to see that, globally, matrix **T** does not influence the entire decomposition reported in equation (2.17), however each element T_{ij} can be modified until reasonable spectral and concentration profiles are obtained, realising, in this way, equation (2.15).

The transformation matrix **T** is a squared matrix that the user can use for rotating and distorting the columns of matrix $\overline{\mathbf{U}} \cdot \overline{\boldsymbol{\Sigma}}$ and the rows of $\overline{\mathbf{V}}$. Clearly, the number of elements of **T** goes as N^2 [47]. For this reason, in order to reduce the number of elements of matrix that can be directly modified by the user, some constraints must be imposed. On this basis, the first or the last experimental spectrum (or even both) can be fixed. This procedure must be performed if and only if it has been attested that they represent some pure species in the chemical data mixture represented by $\boldsymbol{\mu}$. If these requirements are satisfied, the first or the last (or both) column/s of **S** in equation (2.15) are set equal to the corresponding column values of $\boldsymbol{\mu}$. Consequently, the related elements of **T** are obtained solving this set of linear equations:

$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1N} \\ S_{21} & S_{22} & \dots & S_{2N} \\ \dots & \dots & \dots & \dots \\ S_{m1} & S_{m2} & \dots & S_{mN} \end{pmatrix} = \bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}} \cdot \mathbf{T} =$$

$$= \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1N} \\ U_{21} & U_{22} & \dots & U_{2N} \\ \dots & \dots & \dots & \dots \\ U_{m1} & U_{m2} & \dots & U_{mN} \end{pmatrix} \begin{pmatrix} s_{11} & 0 & \dots & 0 \\ 0 & s_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & s_{NN} \end{pmatrix} \cdot$$

$$\cdot \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1N} \\ T_{21} & T_{22} & \dots & T_{2N} \\ \dots & \dots & \dots & \dots \\ T_{N1} & T_{N2} & \dots & T_{NN} \end{pmatrix} \tag{2.19}$$

$$\Rightarrow \begin{cases} S_{11} = \mu_{11} = \alpha_{11}T_{11} + \alpha_{12}T_{21} + \dots + \alpha_{1N}T_{N1} \\ S_{21} = \mu_{21} = \alpha_{21}T_{11} + \alpha_{22}T_{21} + \dots + \alpha_{2N}T_{N1} \\ \dots \\ S_{m1} = \mu_{m1} = \alpha_{N1}T_{11} + \alpha_{N2}T_{21} + \dots + \alpha_{NN}T_{N1} \end{cases};$$

$$\begin{cases} S_{1N} = \mu_{1N} = \alpha_{11}T_{1N} + \alpha_{12}T_{2N} + \dots + \alpha_{1N}T_{NN} \\ S_{2N} = \mu_{2N} = \alpha_{21}T_{1N} + \alpha_{22}T_{2N} + \dots + \alpha_{2N}T_{NN} \\ \dots \\ S_{mN} = \mu_{mN} = \alpha_{N1}T_{1N} + \alpha_{N2}T_{2N} + \dots + \alpha_{NN}T_{NN} \end{cases}$$

Here, symbols α_{ij} are the elements of the $\bar{\mathbf{U}} \cdot \bar{\mathbf{\Sigma}}$ matrix.

The number of unknown values in **T** can be further reduced by the normalization of the spectra. First, each experimental spectrum $\boldsymbol{\mu}_i$, constituting the input dataset **μ** should be normalized using this equation:

$$\boldsymbol{\mu}_{normalized} = \frac{\boldsymbol{\mu}_i}{\sigma_i};\ \sigma_i = \sqrt{\frac{1}{(E_{\max} - E_{\min})} \int_{E_{\min}}^{E_{\max}} dE[\boldsymbol{\mu}_i(E)]^2} \tag{2.20}$$

Where $E_{\min}$ and $E_{\max}$ are the minimum and maximum values of the energy range where the spectra are defined. As follows from equation (2.19) each spectrum of matrix **S** is written as a combination of a series of abstract spectral values α_{ij} multiplied by related the "transformation" elements T_{ij}. The highest contribution to

the construction of each "pure" spectrum comes from the $\boldsymbol{\alpha}_1$, which is characterized by the highest singular value s_{ii}. Actually $\boldsymbol{\alpha}_1$ is the only one component which resembles the XANES spectrum, while $\boldsymbol{\alpha}_2 \ldots \boldsymbol{\alpha}_N$ can be considered as some difference additives to $\boldsymbol{\alpha}_1$. Therefore normalization (2.20) can be applied to $\boldsymbol{\alpha}_1$ by calculating coefficient σ (2.21) and introducing it into matrix **T** (2.22):

$$\sigma = \sqrt{\frac{1}{(E_{\max} - E_{\min})} \int_{E_{\min}}^{E_{\max}} dE[\boldsymbol{\alpha}_1(E)]^2} \tag{2.21}$$

$$\begin{pmatrix} 1/\sigma & 1/\sigma & \ldots & 1/\sigma \\ T_{21} & T_{22} & \ldots & T_{2N} \\ \ldots & \ldots & \ldots & \ldots \\ T_{N1} & T_{N2} & \ldots & T_{NN} \end{pmatrix} \tag{2.22}$$

Finally, taking in account each possible combination of the constraints described above, it is possible to justify that the number of adjustable parameters (i.e. sliders in the program interface) that can be used in order to perform some transformation is given by the following formula:

$$N_{\text{param}} = \begin{cases} N^2 - N: \text{ Normalization imposed;} \\ N^2 - 2N + 1: \text{ Normalization imposed and first/ last spectrum fixed;} \\ N^2 - 3N + 2: \text{ Normalization imposed and both first and last spectrum fixed.} \end{cases} \tag{2.23}$$

2.4. Statistical criteria to determine number of principal components.

From a statistical point of view, the columns of matrix V in eq. (2.15), called loadings, are the principal axes which extend in the directions where the data variance is the highest, while the columns of matrix $\mathbf{U} \cdot \mathbf{\Sigma}$, is called "s**cores**" or components, consist of the data projection over all the n axes.

The matrix **Σ,** appearing in (2.15), is a diagonal matrix whose elements are called singular values. It is possible to demonstrate that each singular value s_{ii} is related to the eigenvalues (λ_i) of the covariance matrix of **μ** by the following relation:

$$\lambda_i = {s_{ii}}^2/(m-1) \tag{2.24}$$

It is worth noting that each λ_i term corresponds to the variance associated to the i[th] component. This means that the higher is the variance related to a determinate component the larger is its contribution in the reconstruction of the dataset. Vice versa, components characterized by a low variance will account for the noise contributes.

The variance values obtained from $\mathbf{\Sigma}$ are used in different statistical tests aimed at determining the correct number of principal component (PC) to consider (i.e. the components related to the real signal and not to the noise). Some of the most popular are: the scree plot, the imbedded error function (IE-test), the factor indicator function (IND-function) and the Malinowski F-Test.

In the scree plot, the variance associated with each component is used as a criterion for accepting or rejecting a determined component. The variance can be plotted against the number of components and the position of the elbow on the curve determines the border between the components having a real physical/chemical meaning and those related to the data noise. The latter components are called secondary components and the associated eigenvalues are called secondary eigenvalues: ${\lambda_i}^0$.

The Imbedded Error function (IE) is characterized by an expression given by:

$$\mathrm{IE} = \sqrt{\frac{k\sum_{i=k+1}^{n}\lambda_i}{mn(n-k)}} \tag{2.25}$$

where *k* represents the number of components used to reproduce the dataset $\boldsymbol{\mu}$. If the experimental errors are distributed randomly and uniformly along each spectrum of $\boldsymbol{\mu}$, then the sum of squares of the projections of the errors, defined as $e_{ij} = \mu_{ij} - \mu_{ij}^{\dagger}(k)$ (where $\mu_{ij}^{\dagger}(k)$ represents the ij element of $\boldsymbol{\mu}$ reconstructed with *k* components), onto each secondary eigenvector (i.e. noise related: ${\lambda_i}^0$) should be approximately the same. This means that: ${\lambda_i}^0 \cong {\lambda_{i+1}}^0 \cong \cdots \cong {\lambda_n}^0$hence, for $k > N$ equation (2.25) can be rewritten as:

$$\mathrm{IE} = n^{1/2} \cdot h \text{ ; with } h = \left(\frac{{\lambda_i}^0}{rc}\right)^{1/2} \tag{2.26}$$

It follows that increasing the number of components, for $k < N$ the IE function progressively decreases until $k = N$, where it reaches a minimum. Then, for $k > N$ the IE assumes a slow growing trend.

Malinowski discovered an empirical function called IND-function, which seems to be more sensitive than the IE function in its ability to pick-up the proper number of components. The IND-function is defined as:

$$\mathrm{IND} = \frac{1}{(n-N)^2}\sqrt{\frac{\sum_{i=N+1}^{n}\lambda_i}{m(n-N)}} \tag{2.27}$$

It is similar to the IE by definition and reaches a minimum when the correct number of components are employed. However, it has been observed that in this function the minimum is more pronounced and can appear in situation where the IE does not exhibit any minimum. More details about the IE and IND functions can be found in [85].

The Malinowski F-Test is a statistical method applied to determine the true dimensionality of a dataset [43]. It is based on the observation that the secondary eigenvalues expressed in the reduced form REV_i (see eq. (2.19)) should be statistically equal

$$\mathrm{REV}_i = \frac{\lambda_i}{(m-i+1)(n-i+1)}. \tag{2.28}$$

As the reduced eigenvalues are still proportional to a variance, a Fisher test can be applied. The test starts from the smallest eigenvalue, clearly associated with the noise, and proceeds to the eigenvalues with higher magnitude until the first significant one (*i.e.* first signal-related one) is found. The k^{th} component is considered significant on the basis of the Fisher test applied on its related standardized F-variable:

$$F(1, n-k) = \frac{\mathrm{REV}_k}{\sum_{k+1}^{n} \lambda_i} \left(\sum_{k-1}^{n} (m-i+1)(n-i+1) \right) \quad (2.29)$$

where the variable λ_i represents the noise-related (secondary) eigenvalues. If the percentage of significance level (%SL), associated with this variable, is lower than a pre-fixed value (usually it is fixed to 5%) then the k^{th} extracted component is accepted as a pure component.

Finally, it is worth to remember that the results coming from the F-Test, IND and IE factors must be considered with caution. These statistical criteria critically depend on the amount of noise in the dataset. In fact, a deviation from the real number of chemical/physical components occurs when the experimental noise (which is not known in advance) is close to the variation in the data, or when some component species have indistinct spectral features or when their fractional weight, in the data mixture, is statistically constant.

Therefore, the main idea of PyFitit is to provide to the user a visual interface for varying structural parameters in a broad range and monitoring the corresponding changes in the spectra. In such approach the user can manually obtain the best agreement between the theoretical and experimental spectra and then start automatically the fitting procedure for a XANES spectrum.

3. ANALYSIS OF X-RAY ABSORPTION NEAR-EDGE STRUCTURE (XANES) AND ULTRAVIOLET-VISIBLE (UV-VIS) SPECTRA

3.1. XANES: definition and detection modes.

X-ray absorption near-edge structure (XANES) spectroscopy using synchrotron radiation is a well-established technique providing information on the electronic, structural and magnetic properties of matter. In XANES, a photon is absorbed and an electron is excited from a core state to an empty state. To excite an electron in a given core-level, the photon energy has to be equal or higher than the binding energy of this core-level. This gives rise to the opening of a new absorption channel when the photon energy is scanned. The energy of an absorption edge therefore corresponds to the core-level energy, which is characteristic for each element, making XANES an element-selective technique.

In XANES, the changes in the absorption of X-rays due to the photoelectric effect is measured. The XANES spectrum is given by the absorption cross section μ. The Fermi Golden Rule states that the XANES intensity (I_{XANES}) for the transition from a system in its initial state Φ_i to a final state Φ_f is given by:

$$I_{XANES} \infty \left| \left\langle \varepsilon \left| \hat{e}_q \cdot r \right| v \right\rangle \right|^2 \rho$$

where $\hat{e}_q \cdot r$ is the electric dipole operator. The quadrupole transition is more than 100 times smaller and it often can be neglected. In the case of the pre-edge structures of the metal *K*-edges, the quadrupole transition is important because the 3*d* density of states is much larger than the 4*p* density of states and the quadrupole peaks appear in the pre-edge region where there is no 4*p* density of states [23].

If an assembly of atoms is exposed to X-rays it will absorb some of the incoming photons. At a certain energy a sharp rise in the absorption cross-section will be observed. This sharp rise in absorption is called the absorption edge. The energy of the absorption edge is determined by the binding energy of a core level. Exactly at the edge, the photon energy is equal to the binding energy, or more precisely the edge identifies transitions from the ground state to the lowest electron-hole excited state.

The core hole binding energy is formally defined in relation to the core electron ionization energy, as it is measured in an X-ray photoemission experiment. The ionization energy is the amount of energy required to remove an electron from an atom, as one would do in an X-ray photoemission spectroscopy (XPS) experiment. In case of a solid, the ionization energy is corrected by the work function, the energy difference between the lowest empty state and the vacuum level. The core hole binding energies of all metals are tabulated in the X-ray data booklet.

The XANES edge energy is not necessarily exactly the same as the core hole binding energy. The two processes are respectively:

$$\mathrm{XPS}: \Psi_0 > \Psi_0 \underline{c} + \varepsilon$$

$$\mathrm{XAS}: \Psi_0 > \Psi_0 \underline{c} V$$

In XPS the ground state (Ψ_0) is excited to the ground state plus a core hole ($\underline{c}$), where the electron (ε) is excited to higher energy, while in XAS the ground state is excited with a core-to-valence excitation ($\underline{c}V$).

The XPS binding (E_B) is defined as the photon energy (Ω) minus the measured kinetic energy of the electron (E_k) and corrected for the work function (ϕ):

$$E_\mathrm{B} = \Omega - E_\mathrm{k} - \phi$$

The work function is the minimal energy to emit an electron from the material. In metals the XAS edge energy can be assumed to be equal to the XPS binding energy, because exactly at the XPS binding energy a transition is possible to the lowest empty state.

Experimentally the XAS edge energy can be slightly higher than the XPS binding energy, for example if the transition to the lowest empty state is forbidden by selection rules.

In first approximation XANES can be described as the excitation of a core electron to an empty state. In the Fermi Golden rule, the initial state wave function is rewritten as a core wave function and the final state wave function (ε) as a valence electron wave function (ν). This implicitly assumes that all other electrons do not participate in the X-ray induced transition. In this approximation, the Fermi golden rule can be written as:

$$I_{XANES} \infty \left| \left\langle \varepsilon \left| \hat{e}_q \cdot r \right| \nu \right\rangle \right|^2 \rho$$

The X-ray absorption selection rules determine that the dipole matrix element is non-zero if the orbital quantum number of the final state differs by one from the initial state ($\Delta L = \pm 1$, i.e., $s \rightarrow p$, $p \rightarrow s$ or d, etc.) and the spin is conserved ($\Delta S = 0$). In the dipole approximation, the shape of the absorption spectrum should look like the partial density of the ($\Delta L = \pm 1$) empty states projected on the absorbing site, convoluted with a Lorentzian. This Lorentzian broadening is due to the finite lifetime of the core-hole, leading to an uncertainty in its energy according to Heisenberg's principle [30]. The single electron approximation gives an adequate simulation of the XANES spectral shape if the interactions between the electrons in the final state are relatively weak. This is the case for all excitations from 1*s* core states (*K*-edges).

The dominant method to calculate the density of states is Density Functional Theory (DFT) where either band structure, multiple scattering or chemical DFT codes can be used. Programs to calculate the X-ray absorption spectral shape include FEFF, Wien2k, QuantumEspresso, ADF and ORCA [105]. Depending on the specific method used one has to use a number of "technical" parameters such as the number of states used, the specific exchange-correlation potential and semi-empirical parameters such as the Hubbard U (the two-electron repulsion energy).

1) Core hole effects and hole-electron excitations. Following the final state rule (von Barth and Grossmann 1982), one has to calculate the distribution of empty states in the final state of the absorption process [6]. The final state includes a core hole on the absorbing site. The inclusion of the core hole introduces a significantly larger unit cell in case of reciprocal space calculations. In case of real-space calculations, the inclusion of a core hole is straightforward and only the potential of the central atom is modified. It has been shown for many examples that the inclusion of the core hole improves the agreement with experiment.

It is not well established if the inclusion of a full core hole gives the best description of the XANES spectral shape. For example, one can use the exchange core hole (XCH) method, or methods that explicitly calculate the creation of hole-electron excitation such as Time- Dependent DFT.

2) Multiplet effects. The core hole that is part of the XANES final state does not only have an influence on the potential, but the core state also has a wave function. In case of a 1*s* core state, this wave function can be neglected. In case of 2*s* and 3*s* wave functions the overlap is larger but the only effect that plays a role is the

spin-up or spin-down character of the core hole. The spin-moment of the core hole interacts with the valence electrons (or holes) giving rise to an exchange splitting in 2*s* and 3*s* XANES. The same exchange interaction plays a role in 2*s* and 3*s* XPS spectra, which are more often studied than their XANES counterparts (L_1 and M_1 edges). Things become dramatically more complex in those cases where a core hole carries an orbital momentum. The core hole spin-orbit coupling that separates the 2*p* XANES spectra into their $2p_{3/2}$ (L_3) and $2p_{1/2}$ (L_2) parts. In addition, the 2*p* wave function can have significant overlap with the valence electrons. In case of 3*d*-systems, the 2*p*3*d* electron-electron interactions are significant and significantly modify the spectral shape. The term "multiplet effect" is used to indicate this core-valence electron-electron interaction.

There are several XANES detection modes.

1) Transmission detection of XANES. A XANES spectrum originates from the fact that the probability of an electron to be ejected from a core level is dependent on the energy of the incoming beam. For this reason the energy of the X-rays is scanned during an experiment. The X-ray interacts with the sample of interest and the intensity after the sample is measured. An important factor of transmission detection is the requirement for a homogeneous sample. Variations in the thickness or pinholes are reasons for the so-called thickness effect that can significantly affect the spectral shape by introducing a non-linear response. Transmission experiments are standard for hard X-rays, but due to the attenuation length of less than one micron, soft X-ray XANES is usually not measured in transmission mode, except in the case of Transmission X-ray Microscopy (TXM) beamlines.

2) Electron yield detection of XANES. The decay of the core hole gives rise to an avalanche of electrons, photons and ions escaping from the surface of the substrate. By measuring any of these decay products, it is possible to measure samples of arbitrary thickness. An important prerequisite for the use of decay channels is that the channels that are measured are linearly proportional to the absorption cross section. With the total electron yield method, one detects all electrons that emerge from the sample surface, independent of their energy. One can detect the current that flows to the sample or detect the emitted electrons. The interaction of electrons with solids is much larger than the interaction of X-rays with solids. This implies that the electrons that escape from the sample must originate

close to the surface. The probing depth of total electron yield (TEY) lies in the range between approximately 1 to 10 nm, depending on the edge strength and the material studied. A quantitative study on the oxygen *K*-edge determined an electron escape depth of 1.9 nm.

3) Fluorescence yield detection of XANES. The fluorescence decay of the core hole can be used as the basis for the XANES measurement. The amount of fluorescent decay increases with energy and dominates over Auger decay for hard X-ray experiments. The photon created in the fluorescent decay has a mean free path of the same order of magnitude as the incoming X-ray, which implies that there will be saturation effects if the sample is not dilute. For materials dilute in the studied element the background absorption μB dominates the absorption at the specific edge and the measured fluorescence intensity (also known as fluorescence yield (FY)) is proportional to the absorption coefficient. For less dilute materials the spectral shape is modified and the highest peaks will appear compressed with respect to the lower peaks, an effect known as self-absorption or saturation. In case of the *L*-edges of transition metal compounds and the *M*-edges of rare earths the fluorescence decay is strongly energy dependent, which implies that for those systems FY detection is not directly proportional to the X-ray absorption spectral shape.

4) Partial Fluorescence Yield detection of XANES. Recently, a range of partial fluorescence yield methods have been developed. We briefly discuss two important approaches, respectively inverse partial fluorescence yield (IPFY) and high-energy fluorescence detection (HERFD). IPFY measured the integrated fluorescence of an element in the system other than the edge element that is measured. It can be shown that such an approach effectively yields a fluorescence yield spectrum that is effectively not sensitive to saturation effects. HERFD uses a high-resolution fluorescence detector to scan through a XANES spectrum. HERFD-XANES measurements are often performed in connection to resonant inelastic X-ray scattering (RIXS) experiments and relate to a constant-emission-energy cross-section through the RIXS plane. Furthermore, by extracting the constant-incident cross-section at the position *K*-edge pre-edge *L*-edge or *M*-edge-like spectra may be recorded. For systems where multiplet effects are important, the HERFD-XANES can be modified from the XANES spectral shape. HERFD-XANES effectively removes the lifetime broadening [51]. This largely enhances the accuracy in the

determination of the pre-edge spectral shapes and their intensities. HERFD detection also allows the detection of XANES spectra that are selective to the valence, spin-state or site symmetry. One can tune the energy of the fluorescence detector to the peak position of one valence and vary the energy of the incoming X-ray, thereby measuring the X-ray absorption spectrum of that particular valence.

It can be shown that electron energy loss spectroscopy (EELS) as detected in an electron microscope can measure exactly the same spectral shape as XANES. This is the case under the assumptions that the primary electron energy is higher than a few thousand eV and that the scattering angle is small.

X-ray Raman spectroscopy is the X-ray analog of optical and UV Raman. A hard X-ray, typically with an energy of about 10.000 eV impinges on the sample and the scattered radiation is measured at an energy lower than 10.000 eV. Like normal Raman one can study vibrational excitations (meV range) and electronic excitations (eV range). In addition, one can study core electron excitations that relate to energy losses of several hundred eV. Note that such core level X-ray Raman could also be named X-ray energy loss spectroscopy (XELS) and as such is the direct X-ray analog of electron energy loss spectroscopy (EELS). As is the case for EELS, the core excitation spectra from X-ray Raman spectroscopy can be described in analogy with XANES under the assumption of small scattering moments. X-ray Raman has a great potential for *in situ* measurements and it presents a hard X-ray alternative to conventional XANES experiments in the study of systems with light elements, including Li, B, C, N and O.

XANES experiments can be performed with X-ray microscopes. A number of transmission X-ray microscopes (TXM) or scanning transmission X-ray microscopes (STXM) exist for the soft X-ray range and the hard X-ray range. Typical resolutions that can be obtained are of the order of 20 nm. The TXM microscopes essentially allow doing XANES spectroscopy with 20 nm spatial resolution. If the sample is rotated in the beam the extension to XANES tomography is straightforward.

3.2. Basic principles of UV-Vis spectroscopy.

Ultraviolet (UV) and visible radiation comprises only a small part of the electromagnetic spectrum, which includes such other forms of radiation as radio, infrared (IR), cosmic, and X rays. The energy associated with electromagnetic

radiation is defined by the following equation: $E = h\nu$ where E is energy (in joules), h is Planck's constant (6.62×10^{-34} Js), and ν is frequency (in s^{-1}).

Electromagnetic radiation can be considered a combination of alternating electric and magnetic fields that travel through space with a wave motion. Because radiation acts as a wave, it can be classified in terms of either wavelength or frequency, which are related by the following equation: $\nu = c / \lambda$ where ν is frequency, c is the speed of light (3×10^{8} ms^{-1}), and λ is the wavelength (in meters). In UV-visible spectroscopy, wavelength usually is expressed in nanometers (1 nm = 10^{-9} m). It follows from the above equations that radiation with shorter wavelength has higher energy.

In UV-visible spectroscopy, the low-wavelength UV radiation has the highest energy. In some cases, this energy is sufficient to cause unwanted photochemical reactions when measuring sample spectra (remember, it is the UV component of light that causes sunburn).

When radiation interacts with matter, a number of processes can occur, including reflection, scattering, absorbance, fluorescence/phosphorescence (absorption and reemission), and photochemical reaction (absorbance and bond breaking). In general, when measuring UV-visible spectra, we want only absorbance to occur. Because light is a form of energy, absorption of light by matter causes the energy content of the molecules (or atoms) to increase. The total potential energy of a molecule generally is represented as the sum of its electronic, vibrational, and rotational energies:

$$E_{\text{total}} = E_{\text{electronic}} + E_{\text{vibrational}} + E_{\text{rotational}}$$

The amount of energy a molecule possesses in each form is not a continuum but a series of discrete levels or states. The differences in energy among the different states are in the order:

$$E_{\text{electronic}} > E_{\text{vibrational}} > E_{\text{rotational}}$$

In some molecules and atoms, photons of UV and visible light have enough energy to cause transitions between the different electronic energy levels [94]. The wavelength of light absorbed is that having the energy required to move an electron from a lower energy level to a higher energy level.

When light passes through or is reflected from a sample, the amount of light absorbed is the difference between the incident radiation (I_o) and the transmitted radiation (I). The amount of light absorbed is expressed as either transmittance or absorbance. Transmittance usually is given in terms of a fraction of 1 or as a percentage and is defined as follows:

$$T = I / I_{\mathrm{o}} \text{ or } \%T = \left(I / I_{\mathrm{o}}\right) \times 100 .$$

Absorbance is defined as follows:

$$A = -\log T .$$

For most applications, absorbance values are used since the relationship between absorbance and both concentration and path length normally is linear.

If a spectrum is expressed as absorbance (A) as a function of wavelength (λ), the derivative spectra are:

Zero order: $A = f(\lambda)$

First order: $\frac{dA}{d\lambda} = f'(\lambda)$

Second order: $\frac{d^2A}{d\lambda^2} = f''(\lambda)$

The first derivative is the rate of change of absorbance against wavelength. It starts and finishes at zero, passing through zero at the same wavelength as □max of the absorbance band. This derivative has a positive and a negative band with maximum and minimum at the same wavelengths as the inflection points in the absorbance band. This bipolar function is characteristic of all odd-order derivatives.

The most distinctive feature of the second-order derivative is a negative band with minimum at the same wavelength as the maximum on the zero-order band. This derivative also shows two positive satellite bands on either side of the main band. The fourth derivative shows a positive band with a maximum at the same wavelength as the maximum on the zero order band. Even-order derivatives show a negative or positive band with minimum or maximum at the same wavelength as □max on the absorbance band.

Optical, electronic, and mathematical methods all can be used to generate derivative spectra. Although optical and electronic techniques formed the basis of early UV-visible spectroscopy, these have been largely superseded by mathematical

methods. To calculate the derivative at a particular wavelength, a window of ± n data points is selected, and a polynomial

$$\lambda = a_0 + a_1\lambda + \ldots + a_l\lambda$$

is fitted by the least squares method. The coefficients $a_0 \ldots a_1$ at each wavelength are the derivative values, where a_1 is the first derivative, a_2 is the second derivative, and so on. Savitzky and Golay developed a highly efficient method to perform the calculations that is the basis of the derivatization algorithm in most commercial instruments. This method also smooths the data. If the polynomial order (l) is less than the number of data points (2n+1) in the window, the polynomial generally cannot pass through all data points [57]. Thus the least squares fit gives a smoothed approximation to the original data points. Although transforming a UV-visible spectrum to its first or a higher derivative usually yields a more complex profile than the zero-order spectrum, the intrinsic information content is not increased. In fact, it is decreased by the loss of lower-order data such as constant offset factors.

Derivative spectra can be used to enhance differences among spectra, to resolve overlapping bands in the qualitative analysis and, most importantly, to reduce the effects of interference from scattering, matrix, or other absorbing compounds in quantitative analysis.

An unwanted effect of the derivatization process is the decrease in signal-to-noise (S/N) with higher orders of derivatives. This decrease follows from the discrimination effect and from the fact that the noise always contains the sharpest features in the spectrum.

Thus, if the spectral data used in the derivative calculation are at 2-nm intervals, the noise has a 2-nm bandwidth. If the analyte band has a bandwidth of 20 nm, the S/N of the first derivative will be 10 times worse than with the zero-order spectrum. The smoothing properties of the Savitzky-Golay polynomial technique can be used to mitigate the decrease in S/N, but care must be taken as a too high degree of smoothing will distort the derivative spectrum.

The higher resolution of the derivative spectra places increased demands on the wavelength reproducibility of the spectrophotometer. Small wavelength errors can result in much larger signal errors in the derivative mode than in the absorbance mode. The negative effect of derivatization on S/N also places increased demands on

low-noise characteristics of the spectrophotometer. If the spectrophotometer can scan and average multiple spectra, S/N can be improved further prior to derivatization.

3.2.1. Qualitative and quantitative analysis of UV-Vis spectroscopy.

UV-visible spectra generally show only a few broad absorbance bands. Compared with techniques such as infrared spectroscopy, which produces many narrow bands, UV-visible spectroscopy provides a limited amount of qualitative information. Most absorption by organic compounds results from the presence of π (that is, unsaturated) bonds. A chromophore is a molecular group usually containing a π bond. When inserted into a saturated hydrocarbon (which exhibits no UV-visible absorbance spectrum), it produces a compound with absorption between 185 and 1000 nm. The presence of an absorbance band at a particular wavelength often is a good indicator of the presence of a chromophore. However, the position of the absorbance maximum is not fixed but depends partially on the molecular environment of the chromophore and on the solvent in which the sample is dissolved. Other parameters, such as pH and temperature, may also cause changes in both the intensity and the wavelength of the absorbance maxima. Conjugating the double bond with additional double bonds increases both the intensity and the wavelength of the absorption band. For some molecular systems, such as conjugated hydrocarbons or carotenoids, the relationship between intensity and wavelength has been systematically investigated. Transition metal ions also have electronic energy levels that cause absorption of 400–700 nm in the visible region.

Although UV-visible spectra do not enable absolute identification of an unknown species they frequently are used to confirm the identity of a substance through comparison of the measured spectrum with a reference spectrum.
UV-visible spectroscopy can be used to determine many physicochemical characteristics of the compounds and thus can provide information as to the identity of a particular compound.

If 100 photons of light enter a cell and only 50 emerge from the other side, the transmittance is 0.5, or 50 %. If these 50 photons then pass through an identical cell, only 25 will emerge, and so forth. Lambert (1760) generally is credited with the first mathematical formulation of this effect, although it now appears that Bouguer first stated it in 1729. The mathematical expression of the Lambert-Beer law is given by

$$A = \log\left(I_o / I_T\right) = C\varepsilon l \ .$$

The extinction coefficient (ε) is characteristic of a given substance under a precisely defined set of conditions, such as wavelength, solvent, and temperature. In practice, the measured extinction coefficient also depends partially on the characteristics of the instrument used. For these reasons, predetermined values for the extinction coefficient usually are not used for quantitative analysis. Instead, a calibration or working curve for the substance to be analyzed is constructed using one or more standard solutions with known concentrations of the analyte.

For electronic transitions, the difference in energy between ground and excited states is relatively large. Therefore, at room temperature, it is highly likely that all molecules are in the electronic ground state. Absorption and return to ground state are fast processes, and equilibrium is reached very quickly [106]. Thus, absorption of UV-visible light is quantitatively highly accurate. The simple linear relationship between absorbance and concentration and the relative ease of measurement of UV-visible light have made UV-visible spectroscopy the basis for thousands of quantitative analytical methods.

Multicomponent analyses using UV-visible spectra have been performed for almost as long as single-component analyses, but because the techniques used in multicomponent analysis often gave incorrect results (as detailed below), they were not widely applied. However, modern instruments yield more precise data, and modern curve-fitting techniques give more accurate results and—perhaps more importantly—indicate when results are incorrect. For these reasons, multicomponent UV-visible analyses are becoming more popular.

1) Principle of additivity. According to Beer's law, absorbance is proportional to the number of molecules that absorb radiation at the specified wavelength. This principle is true if more than one absorbing species is present. All multicomponent quantitative methods are based on the principle that the absorbance at any wavelength of a mixture is equal to the sum of the absorbance of each component in the mixture at that wavelength.

2) Simple simultaneous equations method. The simple approach to multicomponent analysis is based on measurements at a number of wavelengths equal to the number of components in the mixture. The wavelengths chosen usually are those of the absorbance maximum of each component. For calibration, the

absorbance of standards of known concentrations of pure components is measured to determine the extinction coefficient for each component at each wavelength selected. The absorbance of the mixture at each wavelength is the sum of the absorbance of each component at that wavelength, which in turn depends on the extinction coefficient and the concentration of each component. Thus for two components *x* and *y*, the equations are:

$$A'_{(x+y)} = A'_x + A'_y = \varepsilon'_x lc_x + \varepsilon'_y lc_y \text{ and } A''_{(x+y)} = A''_x + A''_y = \varepsilon''_x lc_x + \varepsilon''_y lc_y$$

where A' is absorbance at wavelength λ', A'' is absorbance at wavelength λ'', ε' is molar absorptivity at wavelength λ', ε'' is molar absorptivity at wavelength λ'', c is concentration, and l is path length.

These equations are easily solved to determine the concentration of each component. If measurements were always perfect, accurate results could be obtained even for complex mixtures of components with very similar spectra. In practice, however, measurement errors always occur. Such errors can affect significantly the accuracy of results when spectra overlap significantly.

3) Least squares method. The effect of random noise can be reduced through the use of additional spectral information, that is, a series of data points can be used for quantification instead of only two. In this so-called overdetermined system, a least squares fit of the standard spectra to the spectrum of the measured sample yields quantitative results.

This method enables the analysis of more complex mixtures and of simple mixtures of components with similar spectra. The residual from the least squares calculation is a good indicator of how well the standard spectra fit the sample spectra and is therefore a good indicator of the probable accuracy of the results. An example of multicomponent analysis is the quantification of five hemoglobins in blood with minimum sample preparation.

4) Other methods. Other statistical approaches to multicomponent analysis include the partial least squares (PLS), principle component regression (PCR), and multiple least squares (MLS) methods. In theory, these methods offer some advantages over those described above, but in practice the calibration process is much more complex.

4. APPLIED CHARACTER OF XANES AND UV-VIS SPECTROSCOPY

4.1. Oxidation reactions catalyzed by nonheme iron complexes.

Nonheme iron complexes constitute an important family of catalysts for the oxidation reactions of the bonds C-H and C=C in which the role of the oxidant is covered by H_2O_2 in most of the studies present in the literature, but can also be performed by alkyl peroxides, peracids, hypervalent iodine or persulphate [13, 56]. Two paradigmatic examples of nonheme iron complexes of this kind and relevant to the present thesis are shown in Figure 4.1.

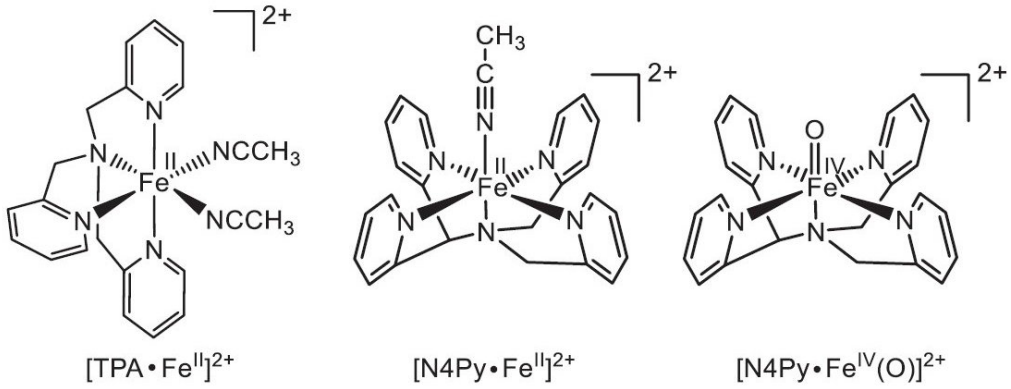


Figure 4.1. Molecular structures of the nonheme iron complexes.

In the present thesis there are 7 reactions belonging to the family of oxidation reactions catalyzed by nonheme iron complexes, which are investigated with XANES and UV-Vis Spectroscopies.

4.2. Reactions investigated with XANES spectroscopy.

In particular, nonheme iron complexes were studied with different substrates, namely:

1) para-cyano thioanisole (pCNPhSMe);
2) thioanisole (PhSMe);
3) para-methoxy thioanisole (MeOPhSMe);
4) benzyl alcohol ($PhCH_2OH$);
5) 9,10 – dihydroanthracene (DHA);
6) diphenylmethane (Ph_2CH_2).

The aforenamed reactions were analyzed for the cases of presence of two and three components.

4.2.1. Results of FAC-PACK.

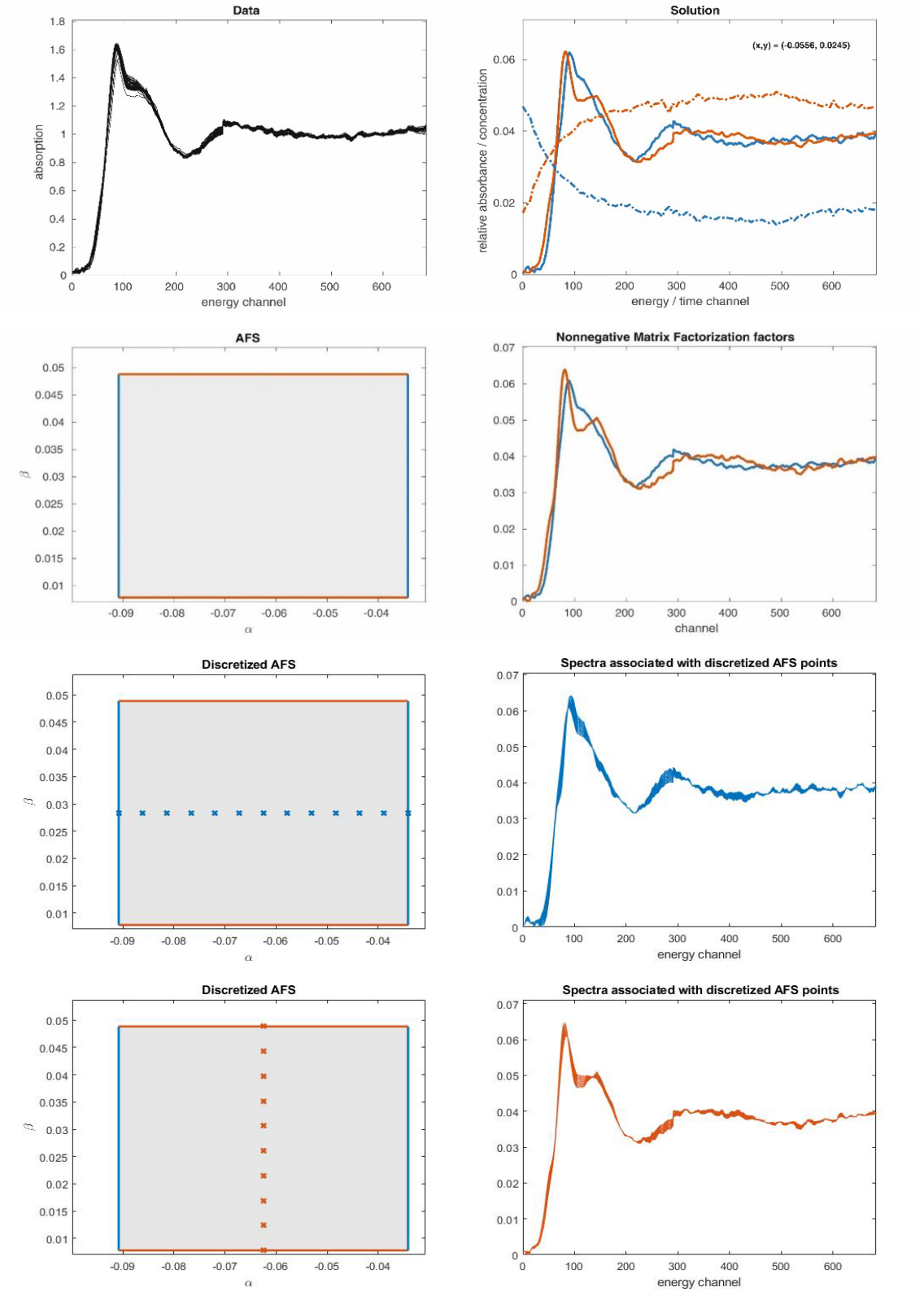
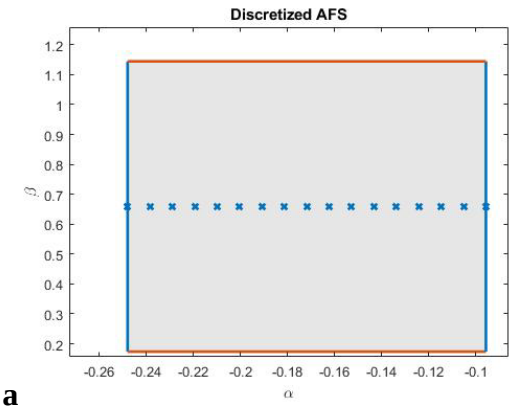
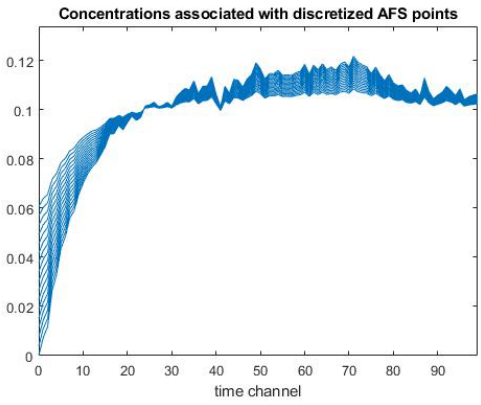


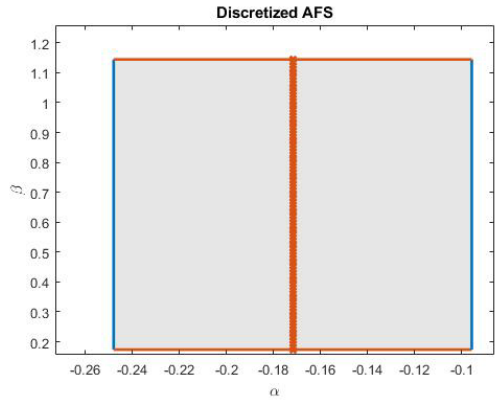
Figure 4.2. *Reaction with 1) para-cyano thioanisole (pCNPhSMe), 2-component case.*



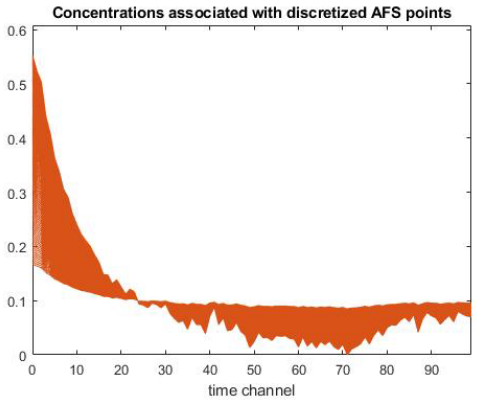
a



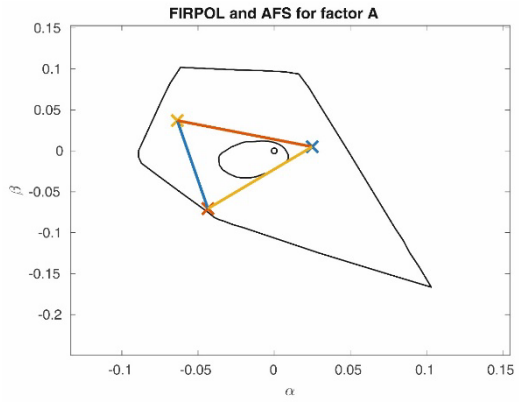
b



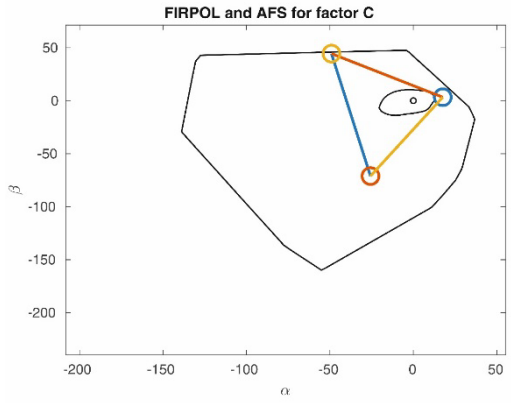
c



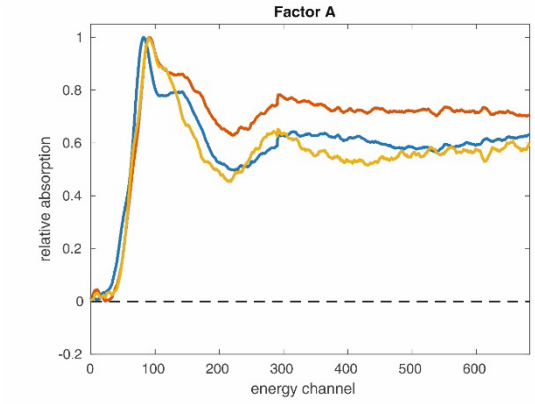
d



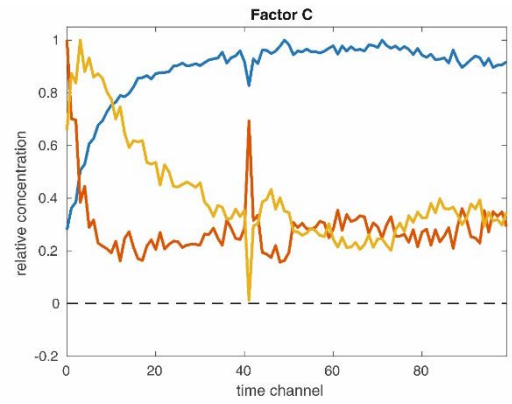
e



f



g



h

Figure 4.3. *Reaction with 1) para-cyano thioanisole (pCNPhSMe);* ***a-d:*** *2-component case;* ***e-h:*** *3-component case.*

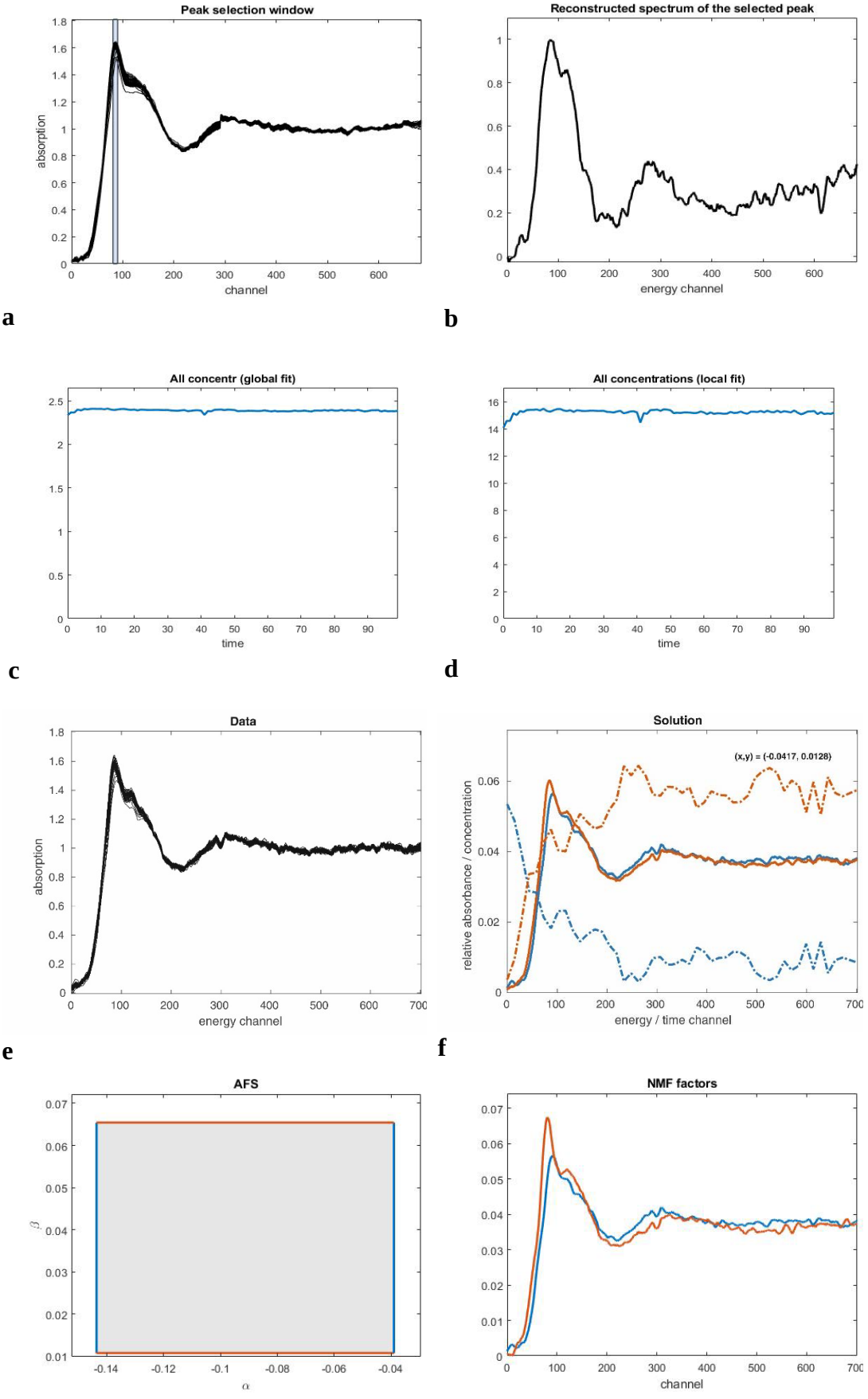


Figure 4.4. ***a-d:*** *Peak group analysis for the reaction with 1) para-cyano thioanisole (pCNPhSMe);* ***e-h:*** *reaction with 2) thioanisole (PhSMe), 2-component case.*

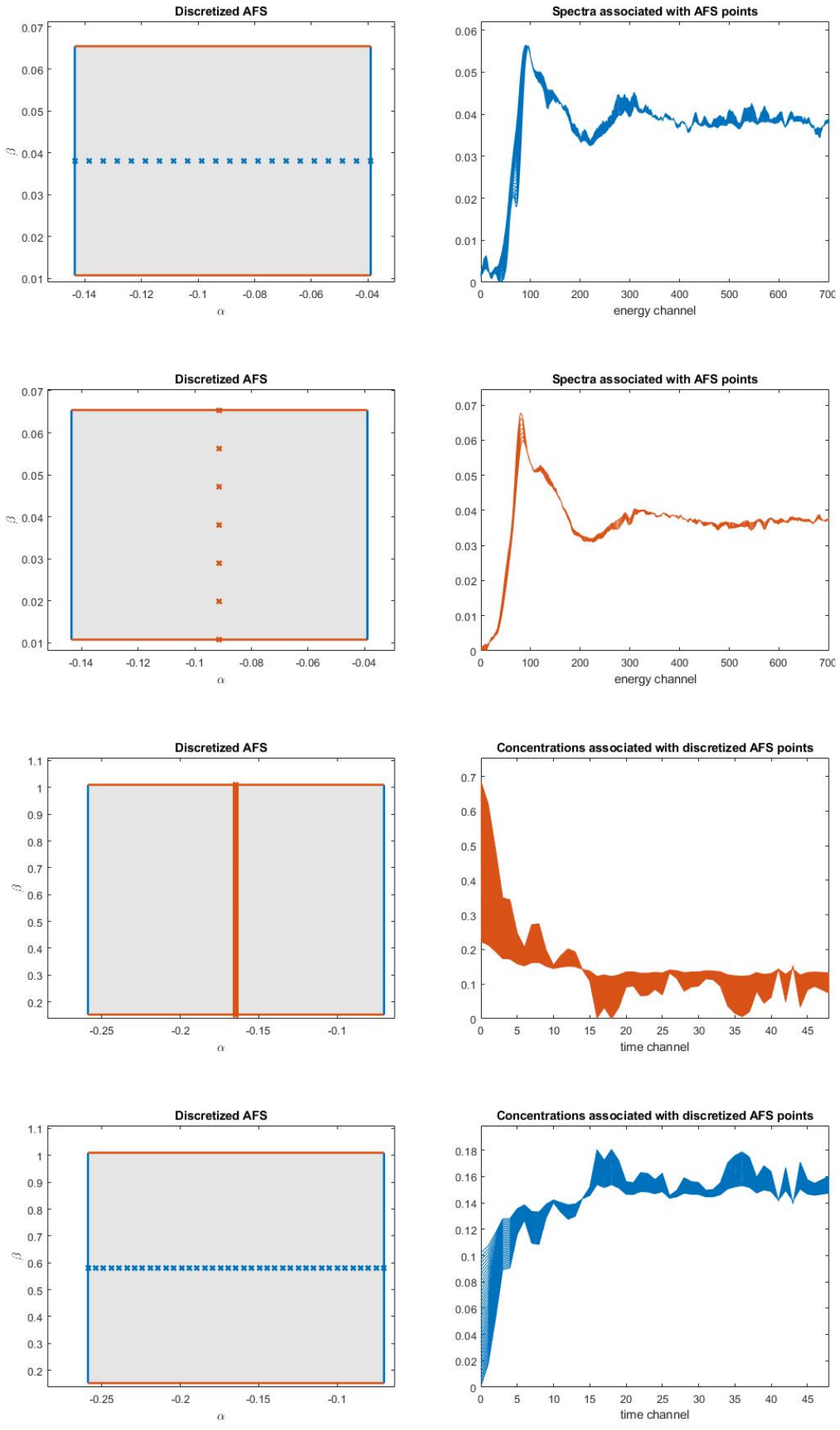


Figure 4.5. *Reaction with 2) thioanisole (PhSMe), 2-component case.*

The description of the contents of some figures:

as it was mentioned in the introduction the soft constraints can be used in order to extract (from the set of all nonnegative factorizations) solutions with special properties [82].

In the present chapter the following constrain functions are applied to spectral and concentrational AFS:

• monotone concentration profiles: monotonously increasing or decreasing profiles are favored and are shown in light gray or white rispectively in **Figure 4.7 (d)**;

• smooth concentration profiles: profiles with a small Euclidean norm of the discrete second derivative are favored (shown white in **Figure 4.7 (b)**);

• exponentially decaying concentration profiles: for every point of a grid in the (α, β)-plane the approximation error for an exponentially decaying function (with an optimized decay constant) is calculated. This allows to favor reactants decaying exponentially, which can be found in the light gray or white areas in the AFS for factor C in **Figure 4.7 (c)** and **Figure 4.21**;

• smooth spectra: spectra with a small Euclidean norm of the discrete second derivative are favored. They are shown in **Figure 4.6 (h)** and **Figure 4.21**;

• small norm spectra: spectra with a small Euclidean norm of the representing vector are favored. Thus, in **Figure 4.6 (g)** spectra with few isolated and narrow peaks are shown in light gray compared to those with wide absorbing peaks (shown in darker gray).

In some graphics (**Figure 4.16 (b, d, f, h)** and others) it is shown that the parts of the concentrational profiles and the parts of the spectra meet at one point. These points are called inversion points [93]. It happens when the values of the transformation matrix $T = \begin{pmatrix} 1 & x \\ 1 & S_{11} \end{pmatrix}$ coincide: $x = S_{11}$.

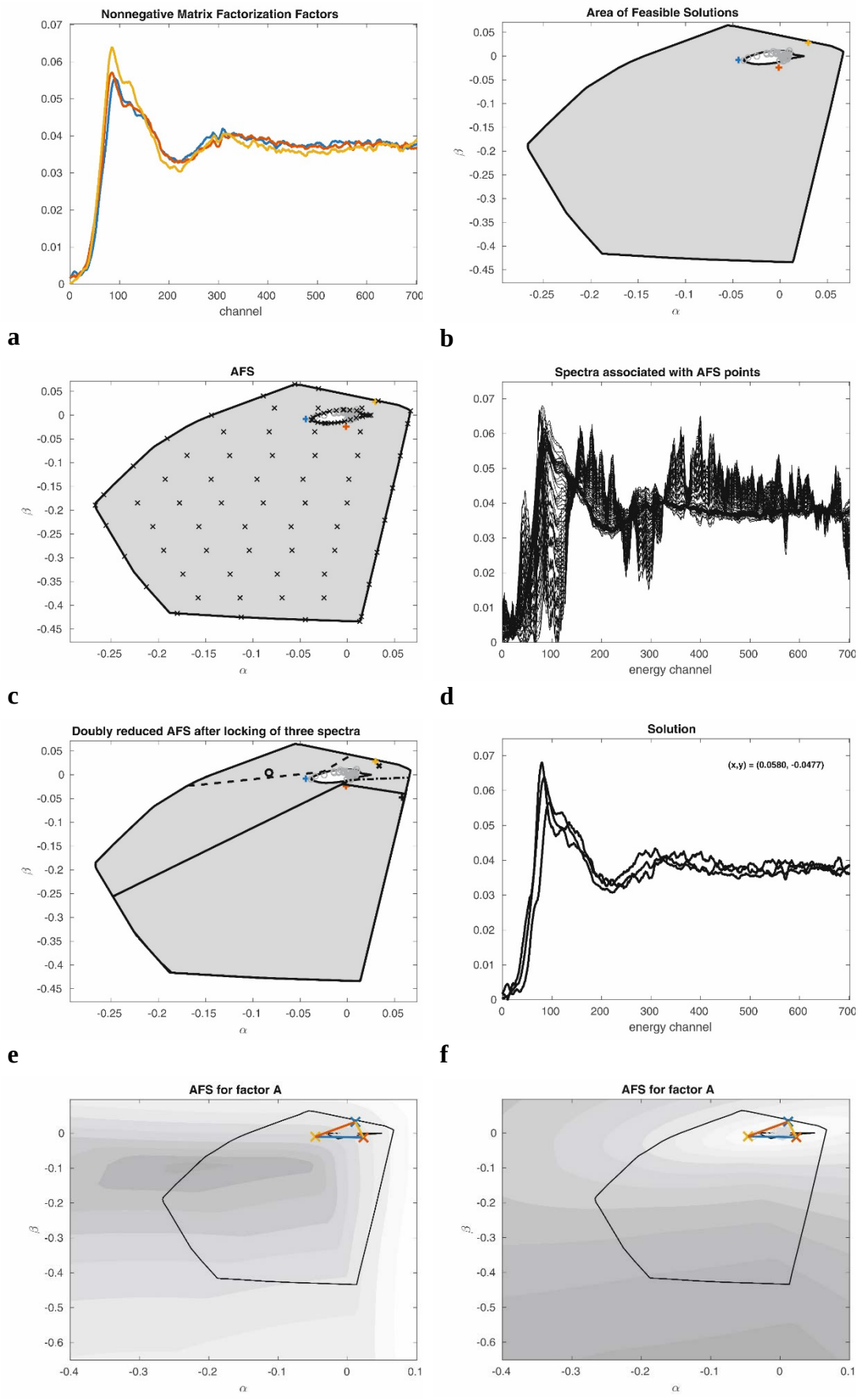


Figure 4.6. *Reaction with 2) thioanisole (PhSMe), 3-component case.* ***b, c, e, g, h:*** *spectral area of feasible solutions with the traces of the spectral data matrix;* ***g:*** *small norm spectra;* ***h:*** *smooth spectra.*

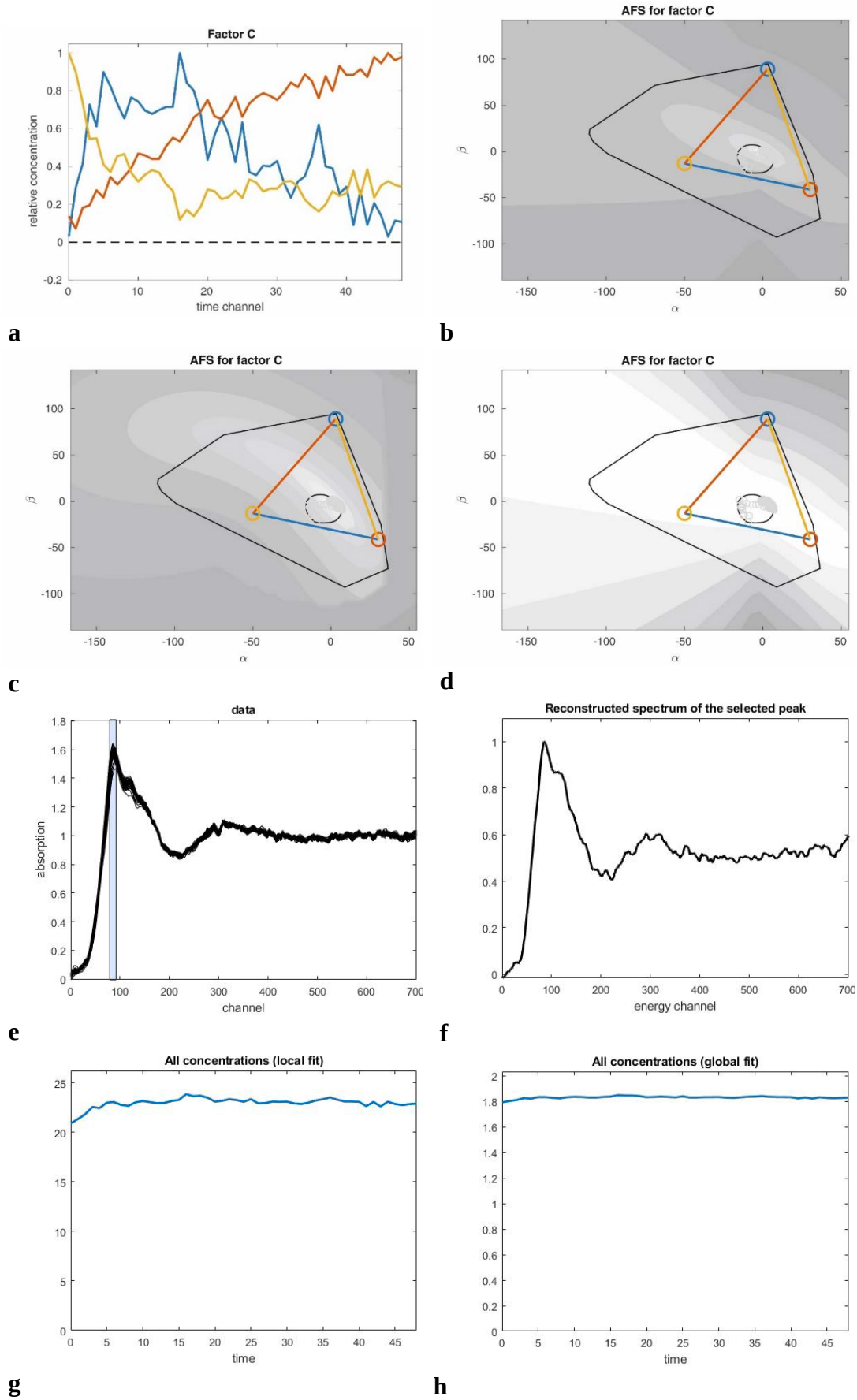


Figure 4.7. *Reaction with 2) thioanisole (PhSMe), 3 componenti;* ***b, c, d:*** *concentration area of feasible solutions;* ***b:*** *smooth concentration profiles;* ***c:*** *exponentially decaying concentration profiles;* ***d:*** *monotone concentration profiles;* ***e-h:*** *peak group analysis.*

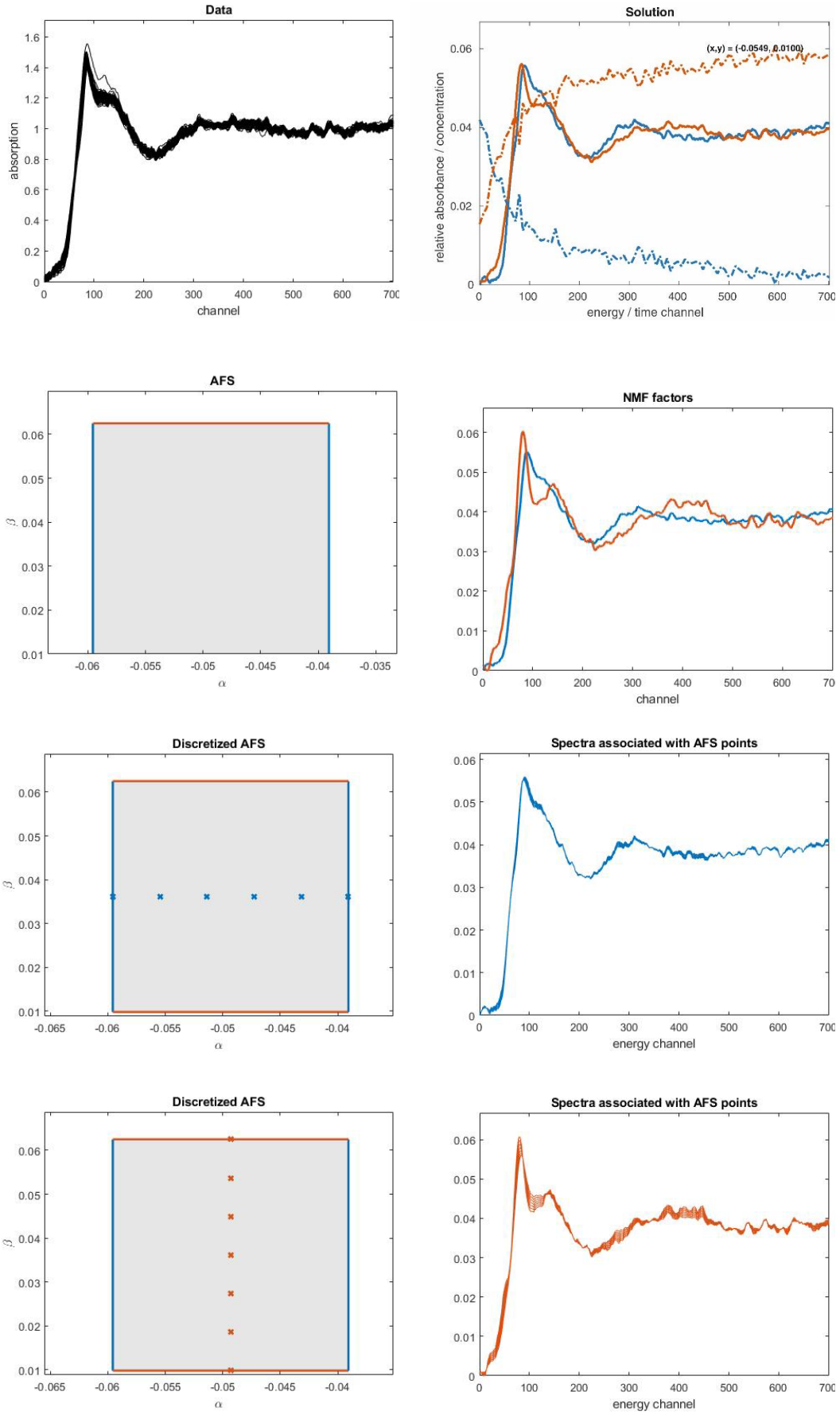


Figure 4.8. *Reaction with 3) para-methoxy thioanisole (MeOPhSMe), 2-component case.*

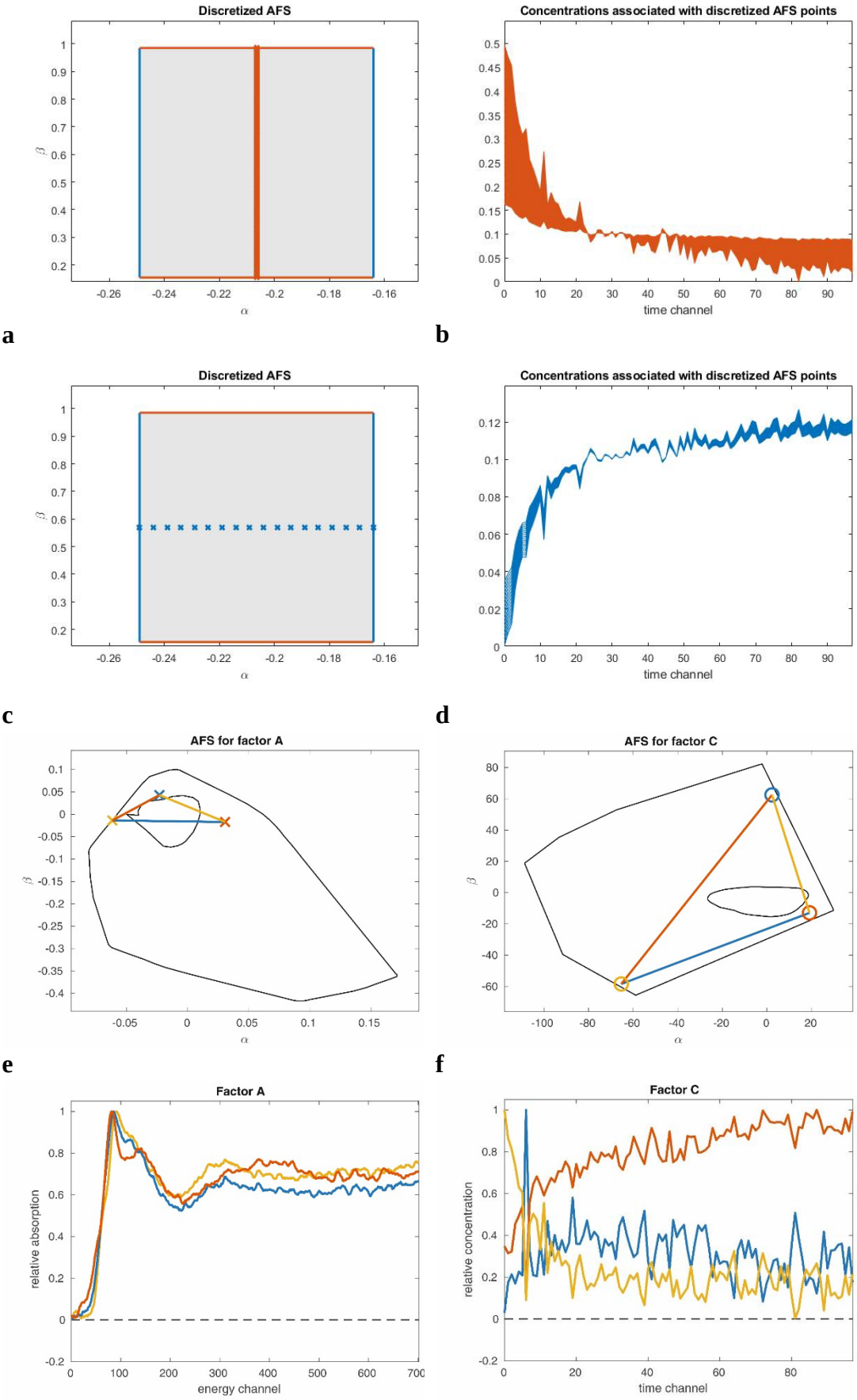


Figure 4.9. *Reaction with 3) para-methoxy thioanisole (MeOPhSMe);* ***a-d:*** *2-component case;* ***e-h:*** *3-component case.*

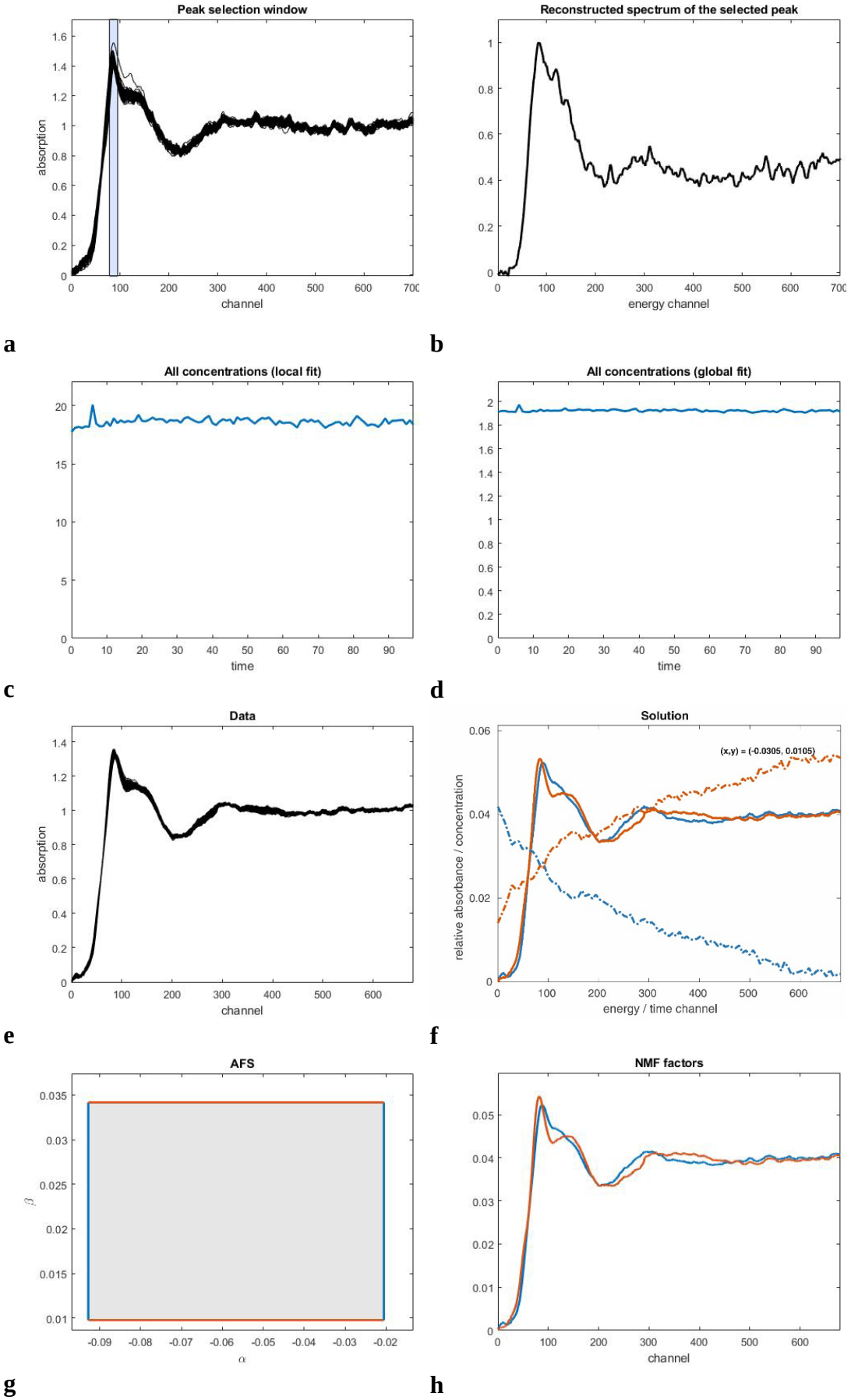


Figure 4.10. ***a-d:*** *Reaction with 3) para-methoxy thioanisole (MeOPhSMe), peak group analysis;* ***e-h:*** *Reaction with 4) benzyl alcohol ($PhCH_2OH$), 2-component case.*

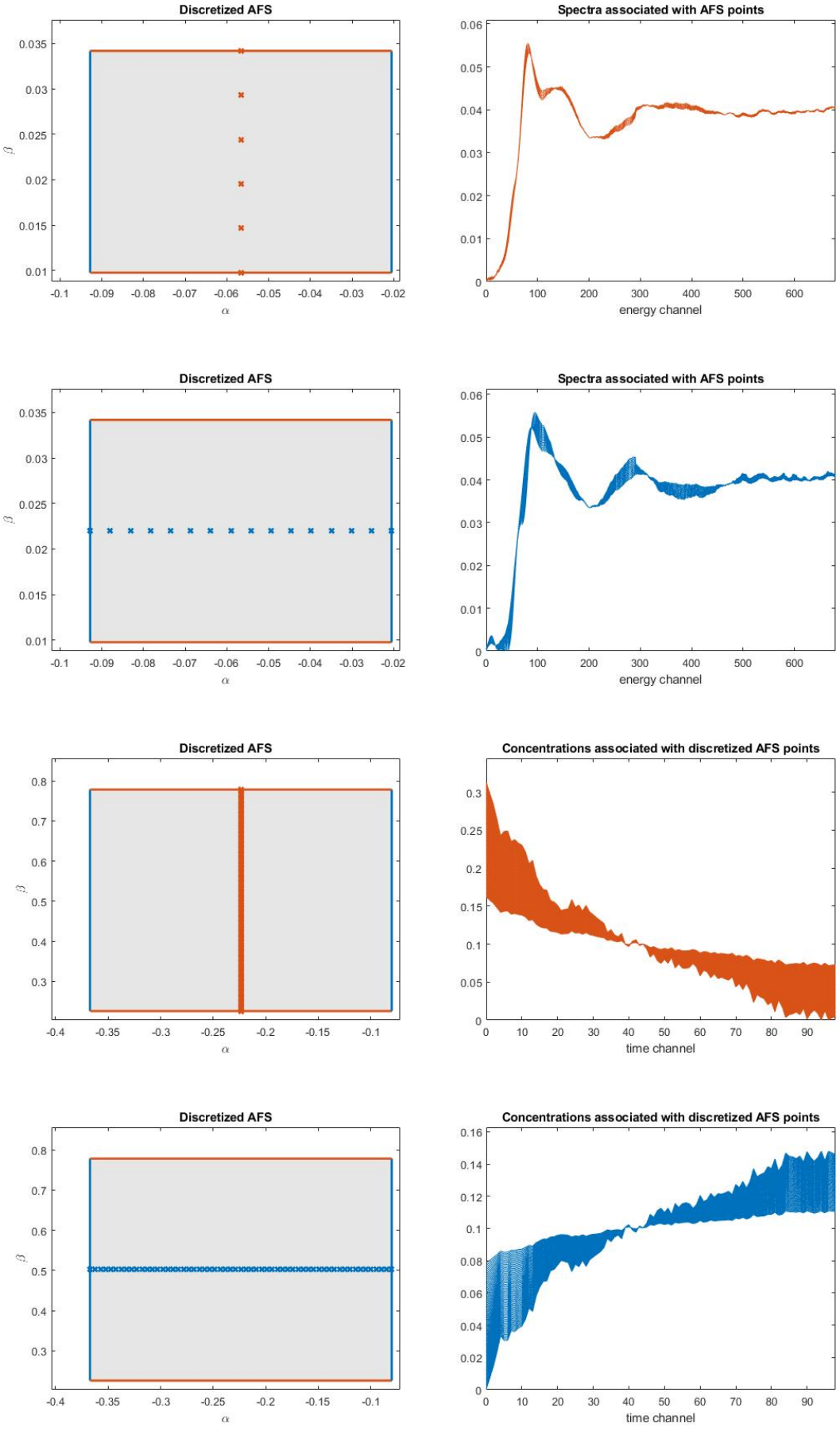


Figure 4.11. *Reaction with 4) benzyl alcohol ($PhCH_2OH$), 2-component case.*

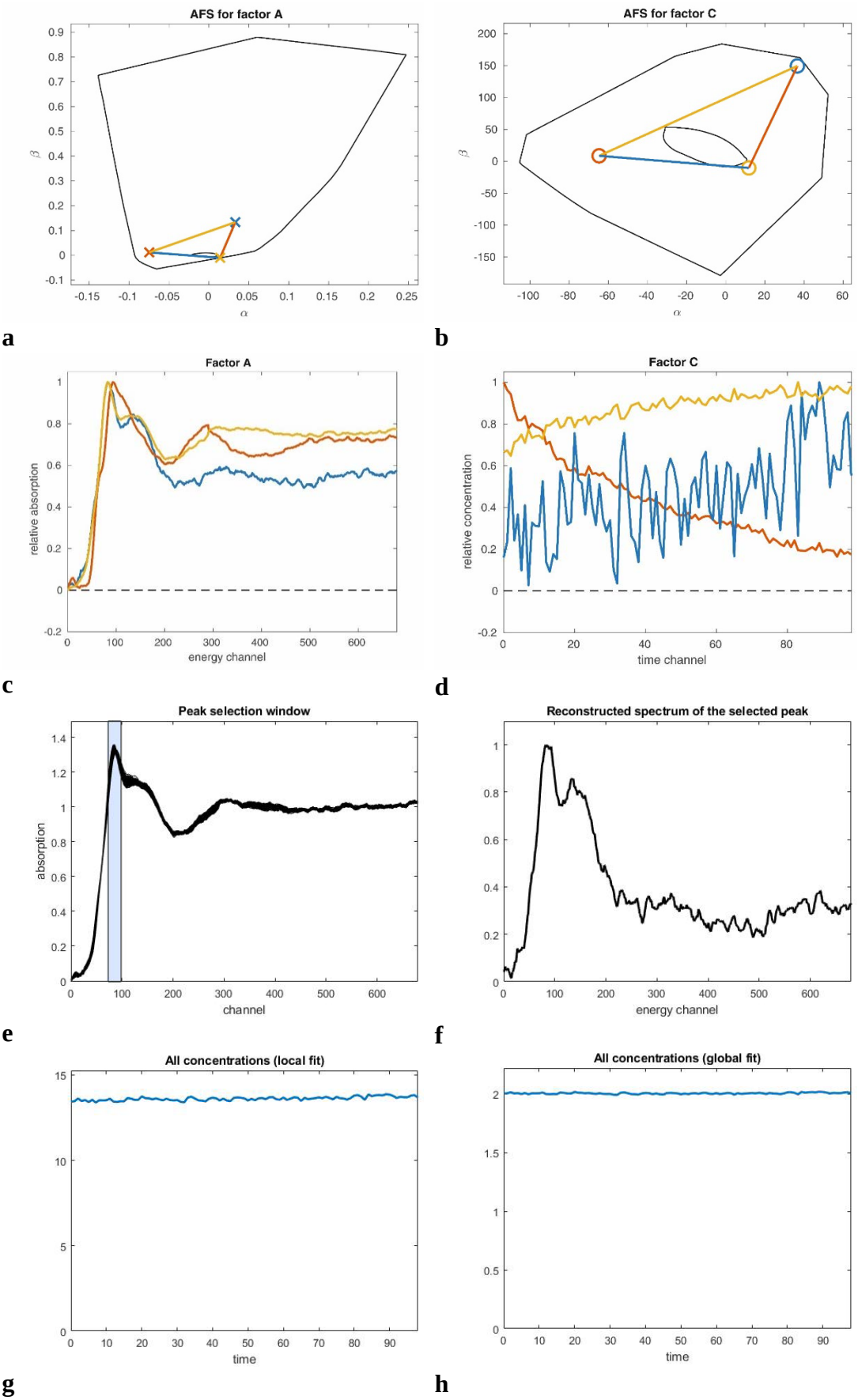


Figure 4.12. *Reaction with 4) benzyl alcohol ($PhCH_2OH$); **a-d:** 3-component case; **e-h:** peak group analysis.*

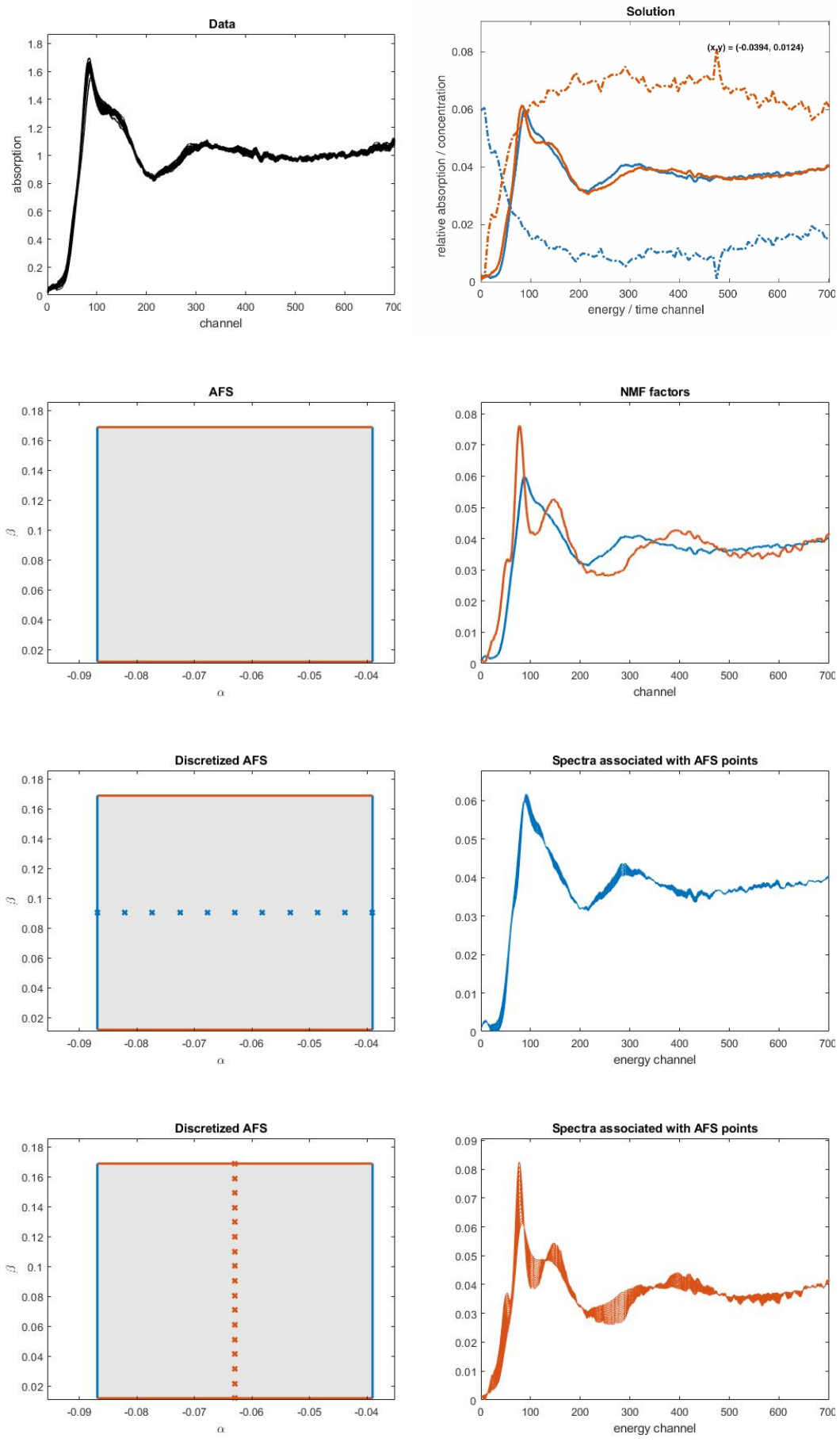


Figure 4.13. *Reaction with 5) 9,10 – dihydroanthracene (DHA), 2-component case.*

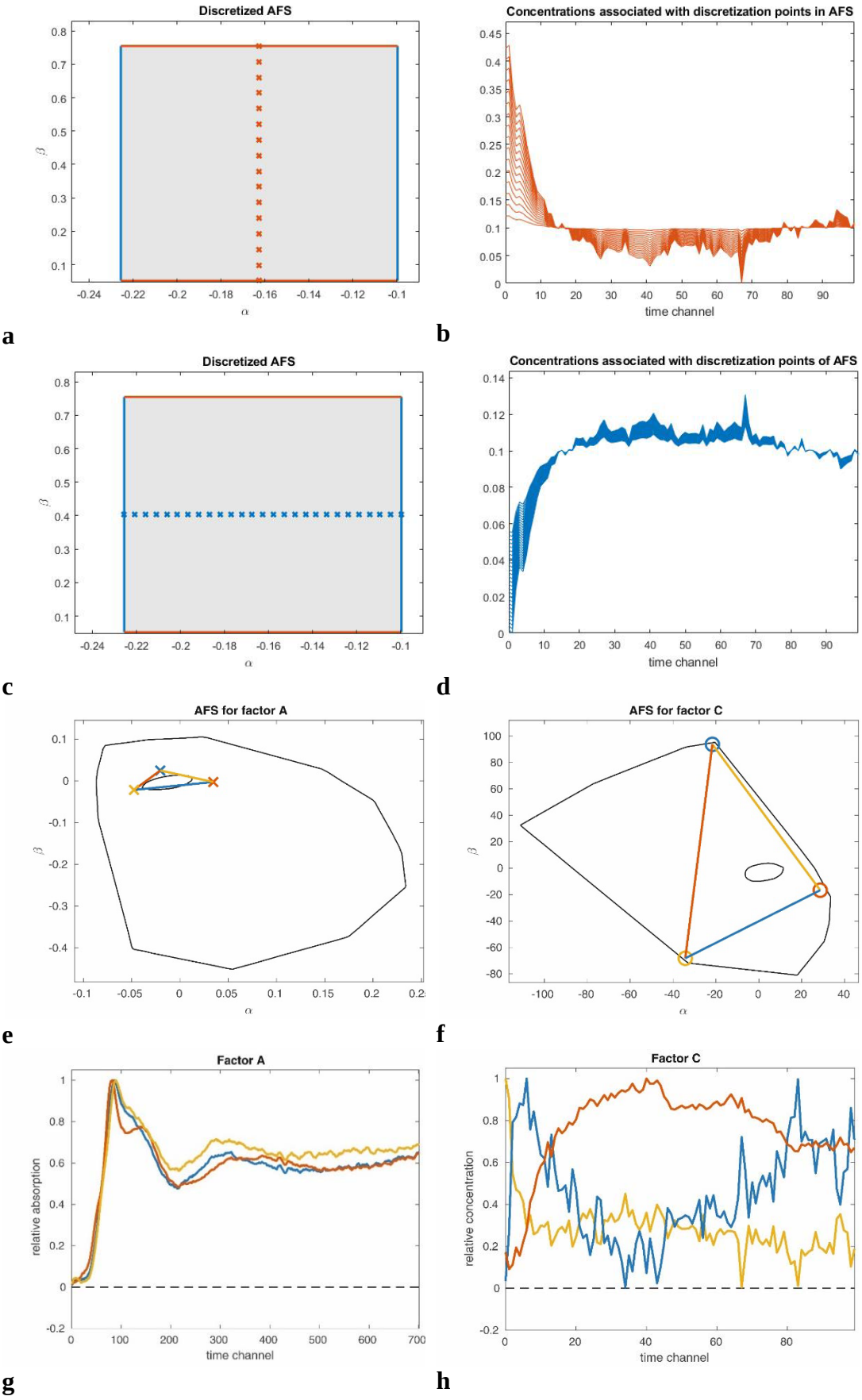


Figure 4.14. *Reaction with 5) 9,10 – dihydroanthracene (DHA),*

a-d: *2-component case;* ***e-h:*** *3-component case.*

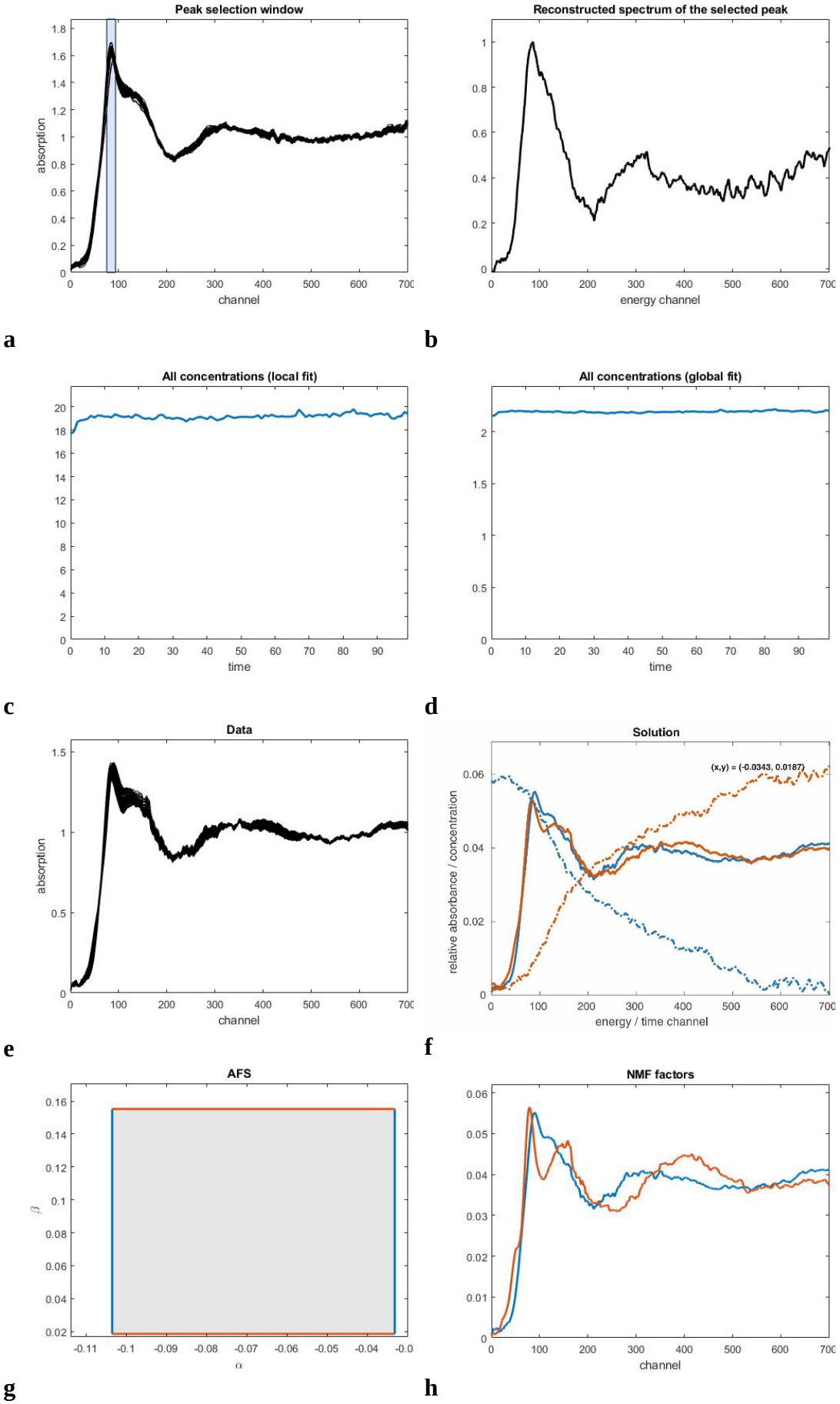


Figure 4.15. ***a-d:*** *Reaction with 5) 9,10 – dihydroanthracene (DHA), peak group analysis;* ***e-h:*** *Reaction with 6) diphenylmethane (Ph_2CH_2), 2-component case.*

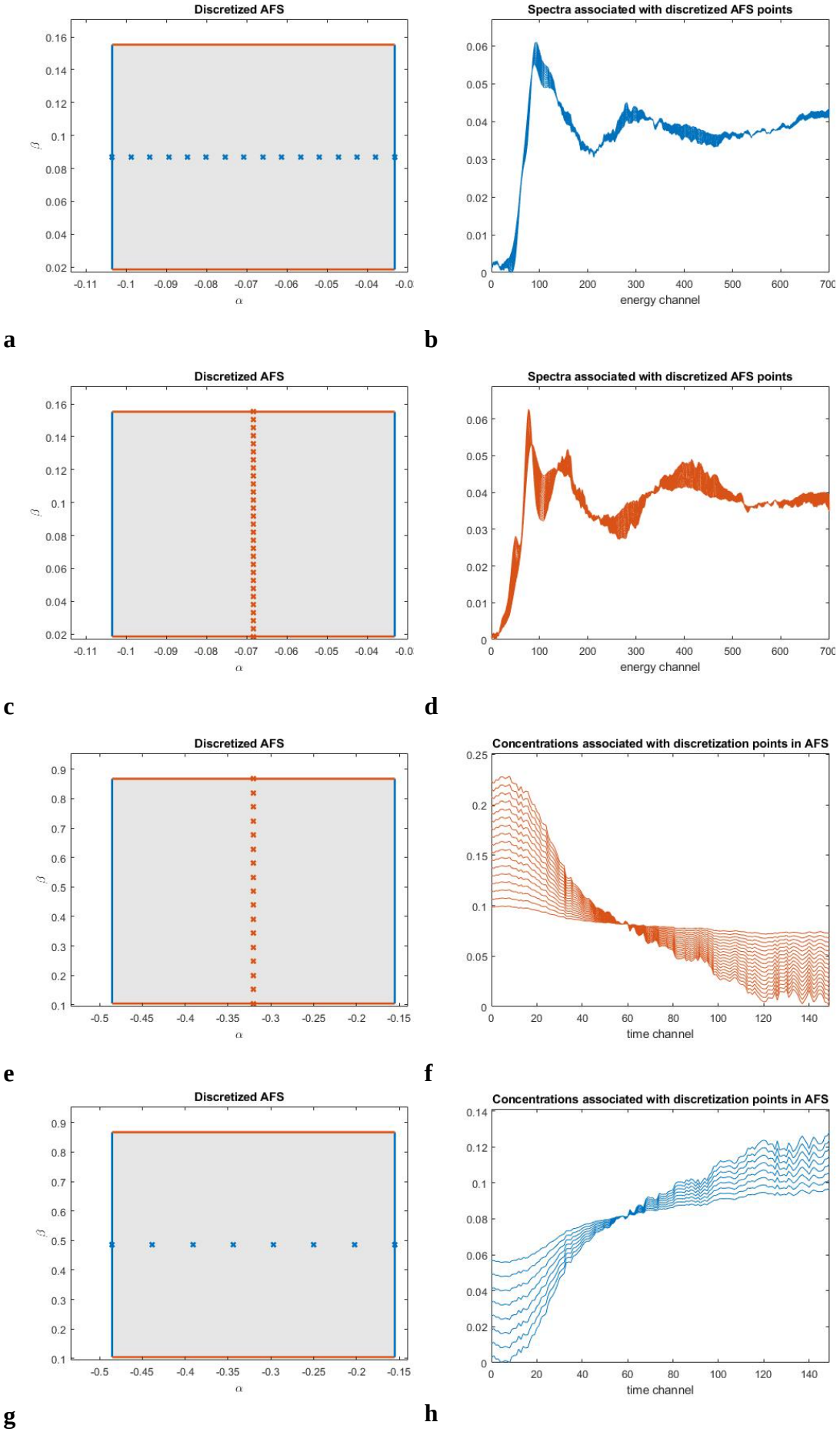


Figure 4.16. *Reaction with 6) diphenylmethane (Ph_2CH_2), 2-component case.*

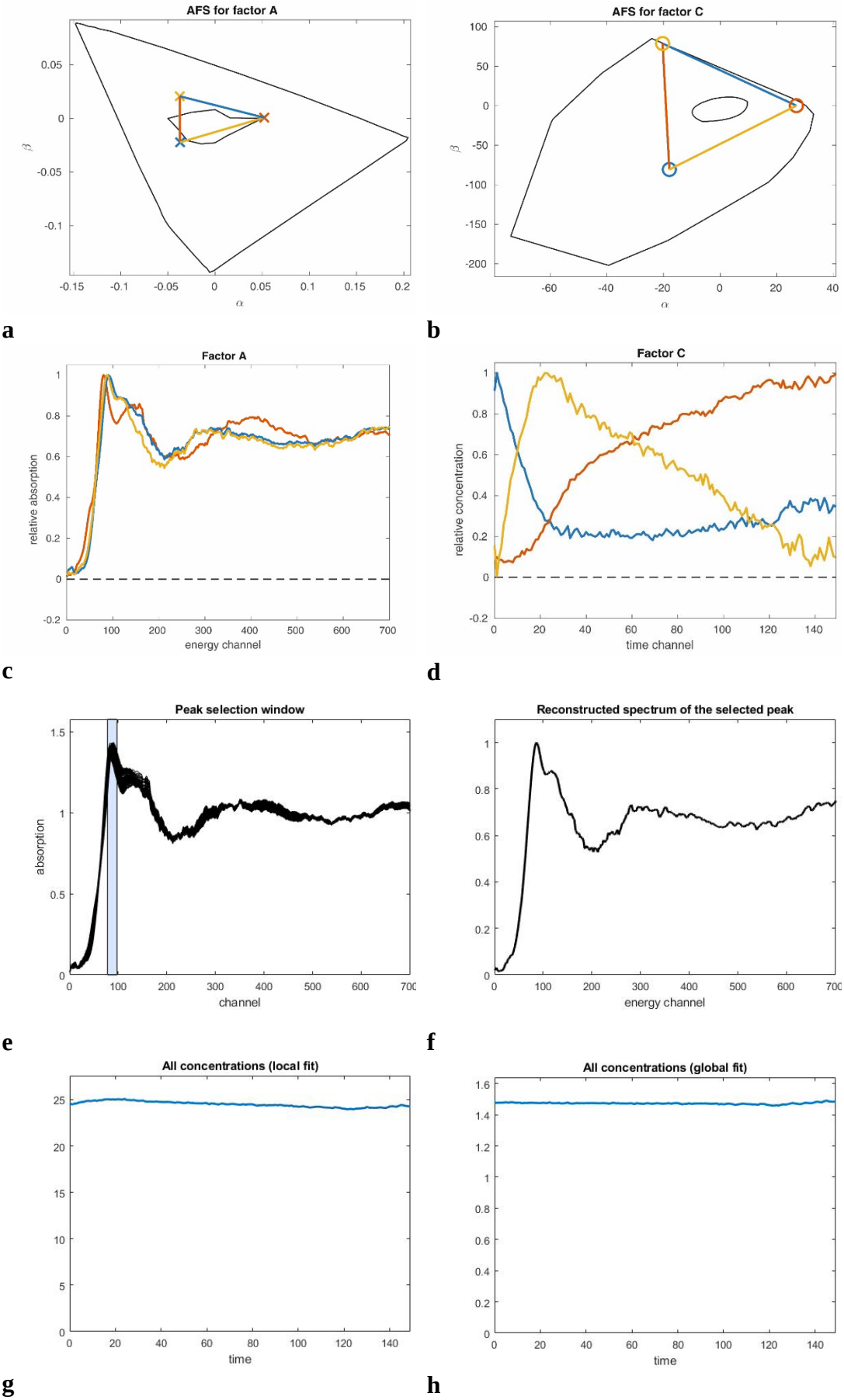


Figure 4.17. *Reaction with 6) diphenylmethane (Ph_2CH_2); **a-d:** 3-component case; **e-h:** peak group analysis.*

4.2.2. Results of PyFitIt.

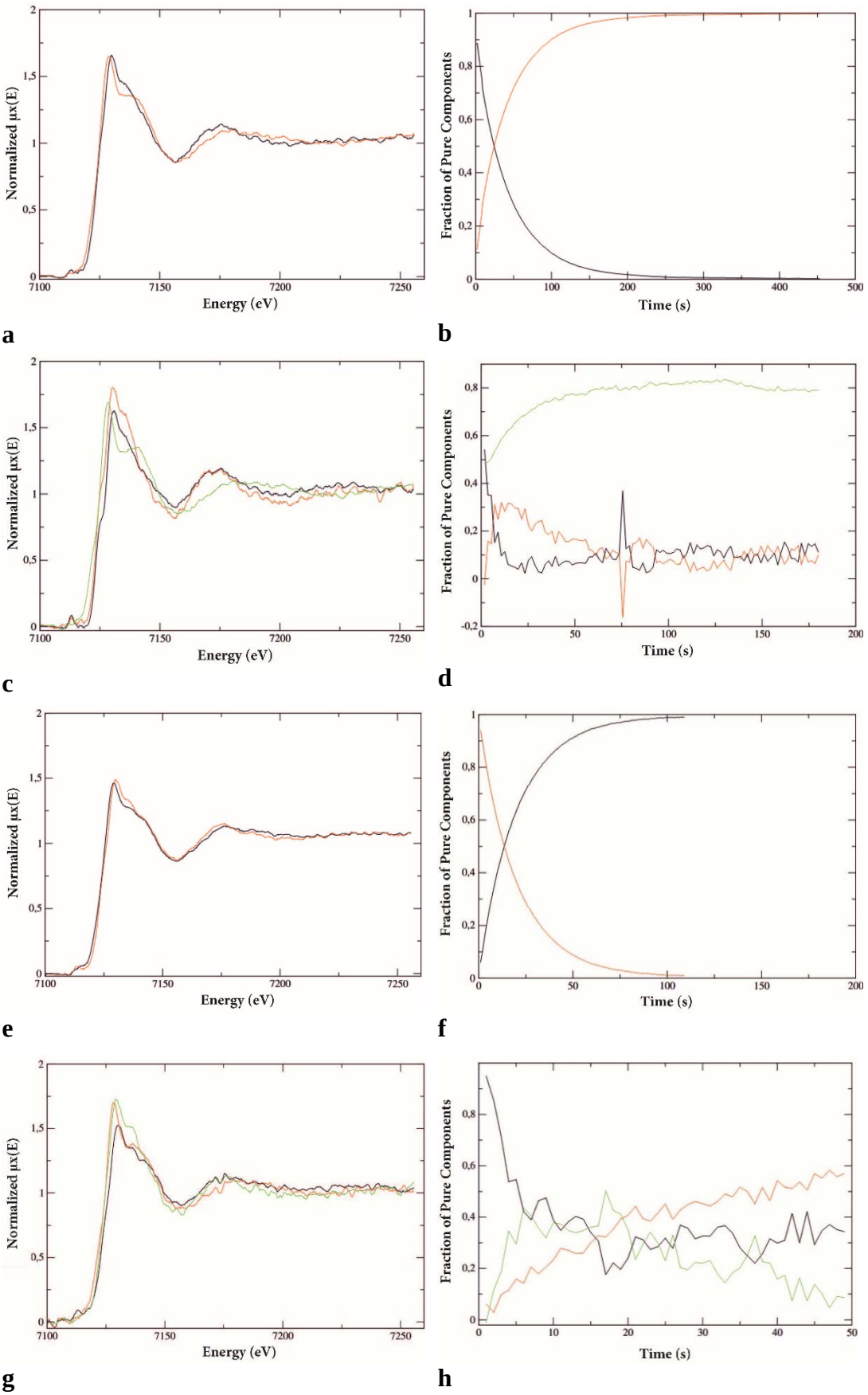


Figure 4.18. *Two-component case: I. a, b – 1) para-cyano thioanisole (pCNPhSMe); II. e, f – 2) thioanisole (PhSMe). Three-component case: I. c, d – 1) para-cyano thioanisole (pCNPhSMe); II. g, h – 2) thioanisole (PhSMe).*

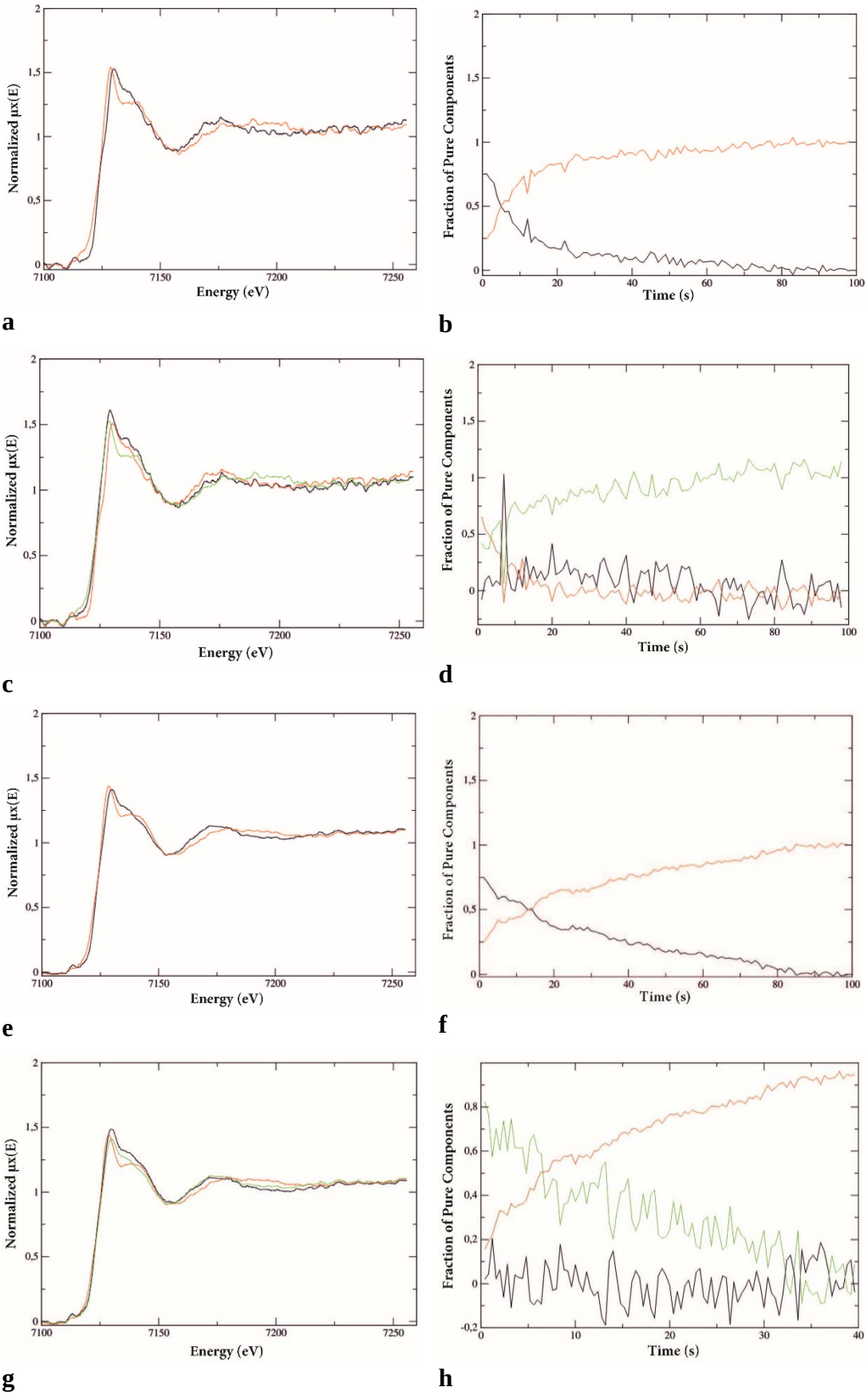


Figure 4.19. *Two-component case: I. a, b – 3) para-methoxy thioanisole (MeOPhSMe); II. e, f – 4) benzyl alcohol ($PhCH_2O$).*
Three-component case: I. c, d – 3) para-methoxy thioanisole (MeOPhSMe); II. g, h – 4) benzyl alcohol ($PhCH_2O$).

4.3. Reaction with CNPhSMe+Fe^{IV}N4Py(O) investigated with UV-Vis spectroscopy.

4.3.1. Results of FAC-PACK.

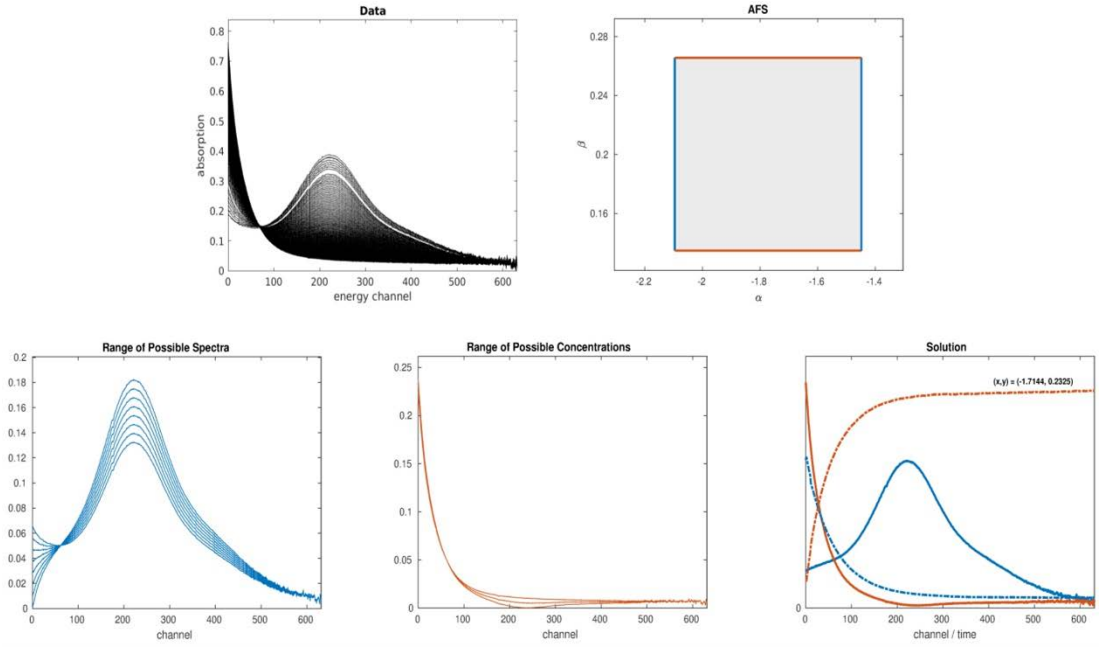


Figure 4.20. *Reaction with CNPhSMe+Fe^{IV}N4Py(O), 2-component case.*

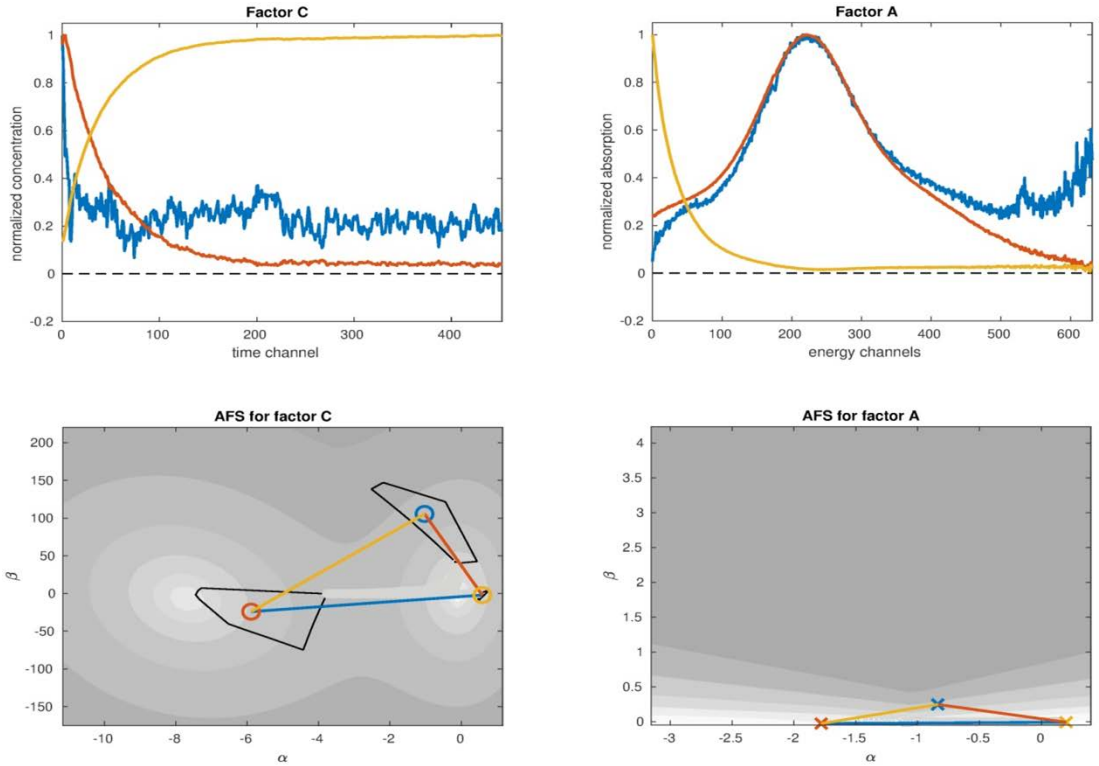


Figure 4.21. *Reaction with CNPhSMe+Fe^{IV}N4Py(O), 3-component case;*
AFS for factor C represents exponentially decaying concentration profiles; AFS for factor A represents smooth spectra.

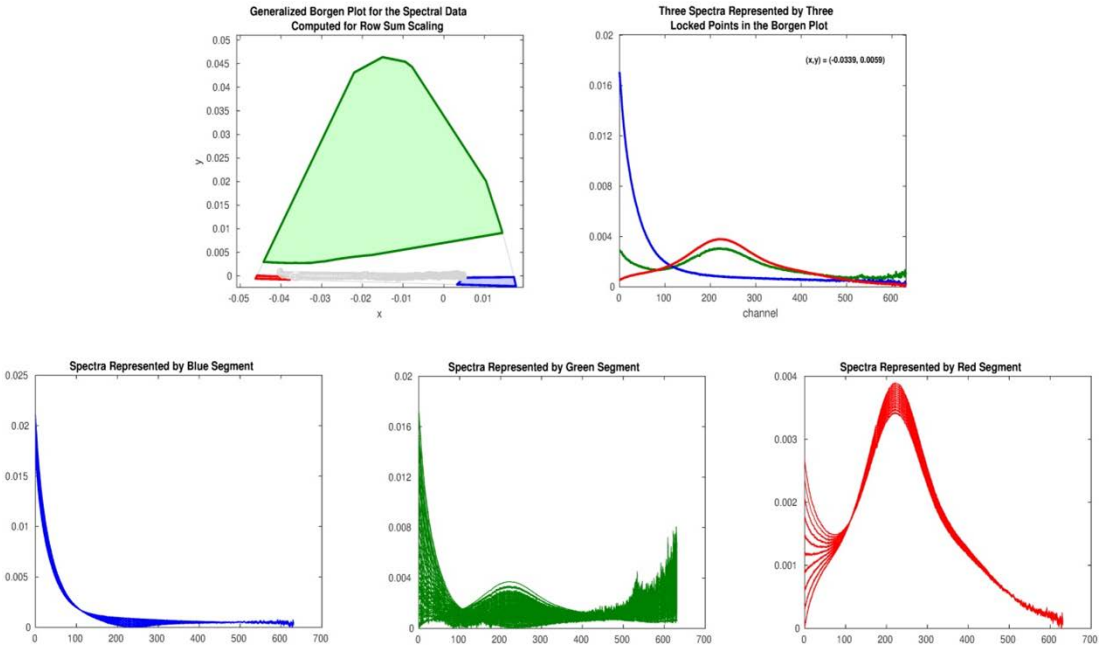


Figure 4.22. *Reaction with CNPhSMe+Fe^{IV}N4Py(O). Generalized Borgen plot for the spectral data matrix computed for RS-scaling.*

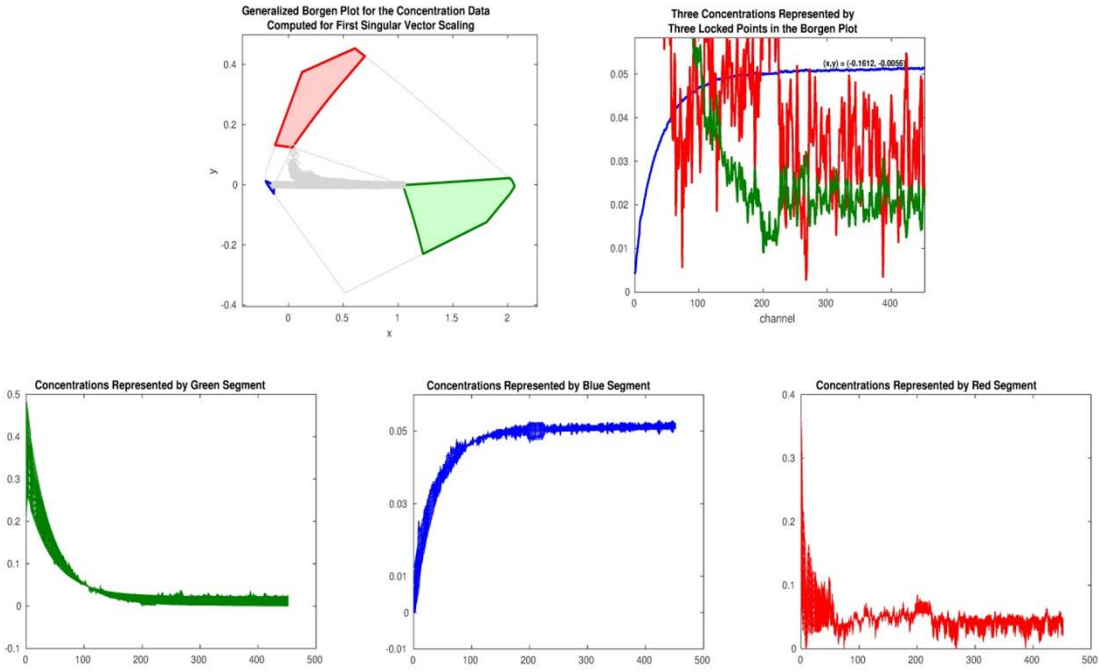


Figure 4.23. *Reaction with CNPhSMe+Fe^{IV}N4Py(O). Generalized Borgen plot for the concentration data matrix computed for FSV-scaling.*

4.3.2. Results of PyFitIt.

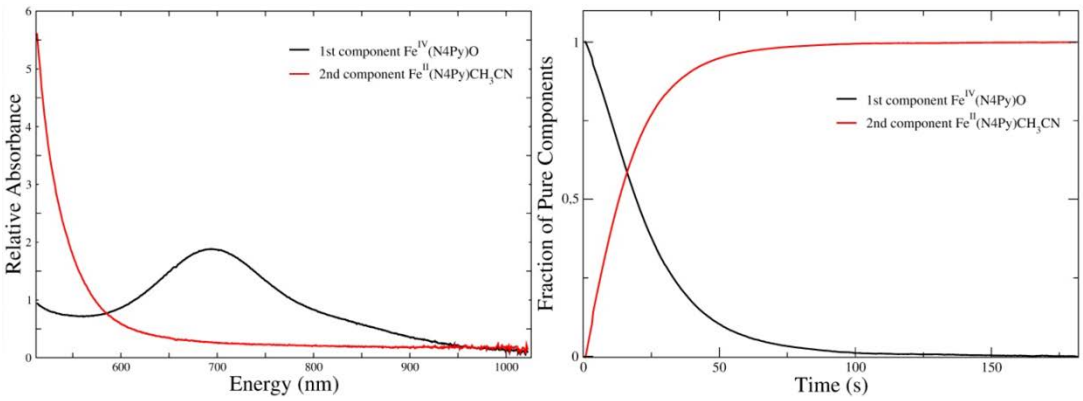


Figure 4.24. *Reaction with CNPhSMe+Fe^{IV}N4Py(O), 2-component case, first and last spectrum fixed.*

5. PROBLEMS OF PROGRAMS: FAC-PACK AND PYFITIT

The main problem of FAC-PACK as well as that of PyFitIt is the absence of a unique univocal solution.

Typically the factorization (1.1) does not result in unique nonnegative matrix factors *C* and *A*. Instead a continuum of possible solutions exists; this non-uniqueness is called the rotational ambiguity of MCR solutions. Sometimes additional information can be used to reduce this rotational ambiguity.

5.1. Problems of program FAC-PACK.

For instance known concentration profiles or pure component spectra lead to a significant reduction of the rotational ambiguity by means of the duality- and complementarity theory. Alternatively, soft constraints can be very useful for extracting chemically meaningful solutions from the AFS.

5.1.1. Equality constraint.

The knowledge of a certain pure component spectrum is often called an equality constraint. It means that a certain point of the AFS is fixed (or locked). The effect on the remaining components is a reduction of the rotational ambiguity or equivalently a reduction of the area of the AFS segments. The reduced AFS can either be constructed geometrically (then one vertex in the simplex rotation algorithm is fixed) or numerically (then a certain row of *T* is fixed). One observes that the reduction effect on the remaining segments of the AFS is relatively large if a point close to the origin is fixed. All these concepts also apply to known concentration profiles. Then the whole procedure works with $D^T = A^T C^T$ where *C* and *A* have changed their places.

The representation of the AFS for the spectral factor is based on the expansion of the spectra with respect to the basis of right singular vectors given by *V*. In a similar way the AFS for the concentration factor rests on an expansion of the concentration profiles with respect to the basis of left singular vectors given by *U* [5].

In equation (5.1)

$$D \approx \underbrace{U\Sigma T^{-1}}_{C}\underbrace{TV^T}_{A} \tag{5.1}$$

the rows (spectra) of A are represented as linear combinations of the right singular vectors, which are the columns of V. The i^{th} row of $A = TV^T$ reads

$$A(i,:) = (t_{i1},\ldots,t_{is})V^T = t_{i1}\left(1,\frac{t_{i2}}{t_{i1}},\ldots,\frac{t_{is}}{t_{i1}}\right)V^T = t_{i1}(1,x)V^T. \tag{5.2}$$

Therein $t_{i1} \neq 0$. Equation (5.2) shows that the i^{th} spectrum $A(i, :)$ aside from scaling is uniquely determined by the row vector $x \in \mathrm{R}^{s-1}$ of the expansion coefficients. The scaling constant t_{i1} in (5.2) can be written as

$$t_{i1} = (T)_{i1} = (AV)_{i1} = (AV(:,1))_i. \tag{5.3}$$

The construction for the factor C is similar. The j^{th} column of $C = U\Sigma T^{-1}$ with $\left(T^{-1}\right)_{ij} = \overline{t_{ij}}$ reads

$$C(:,j) = U\Sigma\left(\overline{t_{1j}},\ldots,\overline{t_{sj}}\right)^T = \overline{t_{1j}}U\Sigma\left(1,\frac{\overline{t_{2j}}}{t_{1j}},\ldots,\frac{\overline{t_{sj}}}{\overline{t_{1j}}}\right)^T = \overline{t_{1j}}U\Sigma(1,y)^T. \tag{5.4}$$

Once again, $\overline{t_{1j}} \neq 0$ is guaranteed by the Theorem 1.3 in the first chapter. It holds that

$$\overline{t_{1j}} = \left(T^{-1}\right)_{1j} = \left(\Sigma^{-1}U^TC\right)_{1j} = \sigma_1^{-1}U(:,1)^TC(:,j). \tag{5.5}$$

It is useful to represent C and A and some of their submatrices by their expansion coefficients x and y according to (5.2) and (5.4). We call this the *block representation of truncated expansion coefficients* with respect to the basis of singular vectors.

Definition 5.1. Let s_0 be an integer with $1 \leq s_0 \leq s$ and let for $i = 1, \ldots, s_0$ the row vector $x^{(i)} \in \mathrm{R}^{s-1}$ be the truncated vector of expansion coefficients of $A(i, :)$ with

respect to the right singular vectors in the sense of (5.2). Considering s_0 rows of $A(1 : s_0, :)$ simultaneously yields

$$X = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(s_0)} \end{pmatrix} \in \mathbb{R}^{s_0 \times (s-1)}$$

as the block representation of truncated expansion coefficients. In the same way let $y^{(j)}$ be the representative of $C(:, j)$ in the sense of (5.4). Then

$$Y = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(s_0)} \end{pmatrix} \in \mathbb{R}^{s_0 \times (s-1)}$$

is the block representation of $C(:, 1 : s_0)$ [16].

Remark 5.1. If $s_0 = s$, then the block representations $X, Y \in \mathrm{R}^{s \times (s-1)}$ define two simplices in the R^{s-1} whose vertices are the row vectors of either X or Y.
Further, Equations (5.2) and (5.3) result in

$$A(i,:) = \left(AV(:,1)_i \left(1, x^{(i)}\right)\right) V^T$$

This yields for $s_0 = s$ and with the s-dimensional 1- vector $e = (1,...,1)^T \in R^s$

$$A = M_1(e, X)V^T \text{ with } M_1 = \mathrm{diag}(AV(:,1))$$

Similarly, Equations (5.4) and (5.5) result in

$$C(:, j) = \left(\sigma_1^{-1} U(:,1)^T C(:, j)\right) U\Sigma \left(1, y^{(j)}\right)^T$$

so that for $s_0 = s$

$$C = U\Sigma \begin{pmatrix} e^T \\ Y^T \end{pmatrix} M_2 \text{ with } M_2 = \mathrm{diag}\left(\sigma_1^{-1} U(:,1)^T C\right).$$

5.1.2. Complementarity theorem.

Next the complementarity theorem is reproduced.

Theorem 5.1. Let $D \in \mathbb{R}_+^{k \times n}$ be a matrix of rank s, which is assumed to be decomposable in the form $D = CA$ with nonnegative factors $C \in \mathbb{R}_+^{k \times s}$ and $A \in \mathbb{R}_+^{s \times n}$. Let be $U\Sigma V^T$ a singular value decomposition of D. Further let the rows $A(i,:)$ for $i = 1, \ldots, s_0$ be given.

Then all the complementary concentration profiles $C(:, j)$ for $j = s_0 + 1,...,s$ are contained in the $(s - s_0)$- dimensional linear subspace

$$\left\{ c \in \mathbb{R}^k : \text{ chas the form } c = U\Sigma y \text{ for a vector } y \in \mathbb{R}^s \text{ which satisfies } A(1:s_0,:Vy = 0\right\} . \tag{5.6}$$

The complementarity theorem shows how pre-given spectra for certain pure components restrict the concentration profiles for the remaining components and vice versa. A comparable observation has been made for three-component systems. Next it is shown how such a pre-given spectrum, which is represented by a single point in the spectral AFS $\mathcal{M}_A$ reduces the AFS $\mathcal{M}_C$ for the complementary components.

Theorem 5.2. Let a spectrum/row $A(i_0, :)$ be given. According to (5.2) it holds $A(i_0,:) = t_{i_0 1}(1, x)V^T$ and x specifies a point in the spectral AFS $\mathcal{M}_A$.
Then all concentration profiles $C(:, j)$ with $j \neq i_0$ are represented in the sense of (5.4) by points y which are elements of the s − 2-dimensional affine subspace

$$C^{(i_0)} = \left\{ y \in \mathbb{R}^{s-1} : \sum_{\ell=1}^{s-1} x_\ell y_\ell = -1 \right\}. \tag{5.7}$$

Thus all feasible concentration profiles $C(:, j)$ with $j \neq i_0$ have the form

$$U\Sigma(1, y)^T \text{ with } y \in C^{(i_0)}.$$

Proof. For given $A(i_0, :)$ Theorem 4.2 from Ref. [75] can be applied (for the case that $1 : s_0$ is substituted by i_0). This theorem guarantees that the complementary concentration profiles $C(:, j)$ for $j \neq i_0$ are elements of the space

$$\left\{ U\Sigma\tilde{y} : \text{ for } \tilde{y} \in \mathbb{R}^s \text{ with } A(i_0,:)V\tilde{y} = 0 \right\}. \tag{5.8}$$

Therein $\tilde{y}$ is a column vector in the R^s.

Insertion of $A(i_0,:) = t_{i_0 1}(1, x)V^T$ into (5.8) shows that

$$t_{i_0 1}(1,x)V^T V\tilde{y} = t_{i_0 1}(1,x)\tilde{y} = 0 \qquad (5.9)$$

is the decisive condition, which is now transformed in order to prove (5.7). First, Equation (5.4) allows one to write the concentration profile $C(:,j) = U\Sigma\tilde{y}$ in the form

$$U\Sigma\tilde{y} = \overline{t}_{1j} U\Sigma(1,y)^T$$

with the row vector $y \in \mathrm{R}^{s-1}$. Thus $\tilde{y} = \overline{t}_{1j}(1,y)^T$. Inserting this into (5.9) yields

$$t_{i_0 1}\overline{t_{1j}}(1,x)(1,y)^T = t_{i_0 1}\overline{t_{1j}}\left(1 + \sum_{\ell=1}^{s-1} x_\ell y_\ell\right) = 0\,.$$

Since $t_{i_0 1}\overline{t}_1 j \neq 0$, the second factor equals 0, i.e. $\sum_{\ell=1}^{S-1} x_\ell y_\ell = -1$, which proves (5.7).

Finally, the dimension of $\mathrm{C}^{(i0)}$ equals $s-2$ because the vector $y \in \mathrm{R}^{s-1}$ has to satisfy one linear constraint. □

The set $\mathrm{C}^{(i0)}$ is an $(s-2)$-dimensional affine subspace which is a hyperplane in R^{s-1} and which intersects $\mathcal{M}_C$. Further, Theorem 5.2 also applies to the situation in which $\mathcal{M}_A$ and $\mathcal{M}_C$ have changed their places [80]. This fact does not require a separate proof but is now stated explicitly.

Corollary 5.1. Theorem 5.2 is applicable to the case in which *A* and *C* are swapped. Then a given representative y for a concentration profile $C(:, i_0)$ results in the set

$$\mathcal{H}^{(i_0)} = \left\{ x \in \mathbb{R}^{s-1} : x \cdot y^T = \sum_{\ell=1}^{s-1} x_\ell y_\ell = -1 \right\}$$

of representatives for the complementary pure component spectra $A(j,:)$ with $j \neq i_0$.

Theorem 5.2 constitutes a relation between a certain point in either the spectral or concentration AFS with an affine subspace in the concentration or spectral AFS. For a two-component system a certain point $x \in \mathcal{M}_A$ is directly related to another point $y \in \mathcal{M}_C$. For a three-component system a certain point $x \in \mathcal{M}_A$ is connected with a straight line in $\mathcal{M}_C$. For an (s = 4)-component system a certain point $x \in \mathcal{M}_A$ is related to a plane. If more than one spectrum or concentration

profile of the pure components is known, then $\mathcal{M}_A$ and $\mathcal{M}_C$ can be further reduced. Then, in the best case, even a unique decomposition can be determined.

Corollary 5.2. For given s_0 rows $A(i, :)$, $i = 1,...,s_0$, let $x^{(i)} \in \mathrm{R}^{s-1}$ be the representatives in the sense of (5.2). Let $X \in \mathrm{R}^{s0\times(s-1)}$ be the block representation of these coefficients according to Definition 5.1.

Then the representatives $y \in \mathcal{M}_C$ of the complementary columns $C(:, j)$, $j = s_0 + 1,...,s$, are elements of the $(s - s_0 - 1)$-dimensional affine subspace

$$C^{(1:s_0)} = \left\{ y \in \mathbb{R}^{s-1} : Xy^T = (-1,\ldots,-1)^T \right\}. \qquad (5.10)$$

Proof. Let $C(:, j)$ with $j > s_0$ be a concentration profile and let $y \in \mathcal{M}_C$ be its representative, see Equation (5.4). Theorem 5.2 imposes the conditions $x^{(i)}y = -1$ for $i = 1,..., s_0$, which gives (5.10). The dimension of $C^{(1:s_0)}$ equals $s - s_0 - 1$ since s_0 linear equations are imposed on $y \in \mathrm{R}^{s-1}$.

Remark 5.2. The dimension $s-s_0-1$ of $C^{(1:s_0)}$ is consistent with the dimension $s-s_0$ in Equation (7) of Theorem 4.2 in Ref. [75]. The reason that the dimension of $C^{(1:s_0)}$ is reduced by 1 is that the block representation of the expansion coefficients in Definition 5.1 includes the fixed scaling of the first left singular vector. In other words, $(1, y)$ is the vector of expansion coefficients under scaling assumptions and $\omega(1, y)$, for $\omega \in \mathrm{R}$, is the full subspace without any scaling. Corollary 5.2 can also be formulated in a way in which C and A have changed their places.

5.1.3. Relations of spectral and concentration AFS.

In this paragraph it is assumed that so much information on a factor is available that the second factor is completely determined by the complementarity theorem. A well known fact on MCR factorizations $D = CA$ is that one given factor determines the second factor. The second factor can be determined as follows: in the case of noise-free data D a linear system of equations is to be solved if CA has the full rank of D. In the case of noisy data or if a low-rank approximation of D is considered, then the second factor can be computed by solving least-squares

problems. In any case the knowledge of a full factor completely determines the system.

A successful factorization means that in the AFS $\mathcal{M}_A$ and the AFS $\mathcal{M}_C$ each s point is specified. These points are the vertices of two simplices, one in $\mathcal{M}_A$ and one in $\mathcal{M}_C$. For the factor A the simplex in $\mathcal{M}_A$ has the vertices $x^{(i)}$, $i = 1, \ldots, s$, see (5.2). The block representation of these vertices is $X \in \mathrm{R}^{s\times(s-1)}$ according to Remark 5.1. Analogously, the factor C defines a simplex in $\mathcal{M}_C$ with the vertices $y^{(j)}$ given by the rows of Y.

For a two-component system the simplex in R is a line segment. For a three-component system the simplex in R^2 is a triangle and its edges are determined by the complementarity theorem 5.2. For four-components systems the simplex in R^3 is a tetrahedron and its side surfaces, the triangles, are determined by the complementarity theorem once again. All this is analyzed and demonstrated in the following. First the relation of the simplex defined by X to the simplex defined by Y is described in Theorem 5.3.

Theorem 5.3. Let $X \in \mathrm{R}^{s\times(s-1)}$ be the block representation of A as introduced in Definition 5.1. Then the vertices $Y(j, :), j = 1,\ldots,s$, can be computed by solving s linear systems of equations. For $j = 1,\ldots,s$ and $Y(j, :) = y^{(j)}$ the linear system of equations reads

$$\begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(j-1)} \\ x^{(j+1)} \\ x^{(j+1)} \\ \vdots \\ x^{(s)} \end{pmatrix} \left(y^{(j)}\right)^T = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}. \tag{5.11}$$

The assertion also holds if X and Y are interchanged.

Proof. Corollary 5.2 for $s_0 = s - 1$ results in a 0- dimensional affine subspace $\mathrm{C}^{(1:s-1)}$ which is just the single vertex $Y(s, :) = y^{(s)}$ and proves the case $j = s$. The argument can also be applied for the remaining indexes j [28]. □

Remark 5.3. Theorem 5.3 in Equation (5.11) formulates a relation between the simplices in $\mathcal{M}_C$ and $\mathcal{M}_A$ which are defined by *X* and *Y*. This relation cannot immediately be translated into a factorization of *D* since the feasible factorizations

$$D = \underbrace{U\Sigma T^{-1}}_{C}\underbrace{TV^T}_{A} = \underbrace{U\Sigma Z}_{C'}\underbrace{Z^{-1}V^T}_{A'}$$

with *T* and *Z* defined in (5.12)

$$T(1,:) = (1, x), \quad Z(:,1)^T = (1, y) \tag{5.12}$$

include a specific scaling of the rows of *A* and columns of *C*. Thus in general $C'A = U\Sigma ZTV^T \neq D$, *D* holds. What is needed for a correct representation of the factorization are the two diagonal matrices M_1 and M_2 as introduced in Remark 5.1. With these matrices and with *T = (e, X)* and $Z^T = (e, Y)$ for $e = (1,\ldots,1)^T \in \mathbb{R}^s$ it holds that $D = U\Sigma ZM_2M_1TV^T$.

FAC-PACK includes an algorithmic implementation of the complementarity theorem which allows one to import known spectra or known concentration profiles, to mark their representatives in the AFS and to construct as well as to draw the complementary affine spaces.

5.1.4. Presence of noise in AFS.

Theorem 5.1 and Equation (5.7) can be used to derive a relation on the sensitivity of the AFS with respect to noise.

Lemma 5.1. Let $x \in \mathcal{M}_A$ be given and let $y \in \mathcal{M}_C$ be in the complementary space of concentration profiles as given by (5.7). If the perturbation of x due to noise is given by δ_x, then the induced perturbation δ_y of *y* is bounded as follows

$$\left\|\delta_y\right\| \geq \frac{\left|\left(\delta_x\right) y^T\right|}{\| x \|} \tag{5.13}$$

if the quadratic term $(\delta_x)(\delta_y)^T$ is ignored. The inequality also holds with (x, δ_x) and (y, δ_y) having changed their positions.

Proof. Let a certain spectrum be given and let its representative be $x \in \mathcal{M}_A$. Theorem 5.2 shows that the representatives y of the complementary concentration profiles fulfill $xy^T = -1$. Let $\delta_x \in \mathrm{R}^{s-1}$ be a perturbation (row) vector of x and δ_y be the resulting perturbation for y. From

$$(x + \delta_x)(y + \delta_y)^T = -1$$

one gets after subtraction of $xy^T = -1$

$$(\delta_x)\, y^T + x(\delta_y)^T = -(\delta_x)(\delta_y)^T = O\left(\|\delta_x\|\|\delta_y\|\right)$$

where O is the Landau symbol and where $\|\cdot\|$ is the Euclidean vector norm. (The Landau or big O notation is used to describe the asymptotic behavior of a function; here it expresses that $(\delta_x)(\delta_y)^T$ is a mixed quadratic term in the δ-perturbations, which quadratically tends to 0, if $\delta_x \to 0$ and $\delta_y \to 0$.)

Next the second order term of perturbations on the right-hand side is ignored. Application of the Cauchy- Schwarz inequality leads to

$$\|x\|\|\delta_y\| \geq \left|x(\delta_y)^T\right| = \left|(\delta_x)\, y^T\right|.$$

This proves that $\|\delta_y\| \geq \left|(\delta_x)\, y^T\right| / \|x\|$. □

Inequality (5.13) shows that the resulting perturbation $\|\delta_y\|$ is bounded from below by $\left|(\delta_x)\, y^T\right| / \|x\|$. This lower bound is reciprocal to $\|x\|$ which is the Euclidean distance of x to the origin [66]. An interpretation of this result is as follows: for points x far away from the origin the influence of perturbations δ_x on y decreases. However, any x close to the origin appears to be sensitive with respect to perturbations.

This perturbation argument is consistent with spectroscopic observations: for IR-spectra with narrow localized peaks next to non-absorbing frequency bands, the representatives x are often far away from the origin. Hence the sensitivity with respect to noisy data is relatively small. In contrast to this UV-Vis data often has wide absorbing frequency bands without non-absorbing bands. Then the representing vectors x of the true solutions are often close to the origin and the reliability of the results of Theorem 5.1 for noisy data decreases.

The reciprocal relation between $\|x\|$ and the perturbation $\|\delta_y\|$ which is expressed by Equation (5.13) has some structural resemblance to the observation of Windig, Keenan et. al. [98] namely that in MCR techniques high contrast solutions in the *C*-space are related to low contrast solutions in the *A*-space and vice versa.

The next lemma shows that the acute angle which is enclosed by x and $x + \delta_x$ in the *A*-space equals the acute angle which is enclosed by the associated onedimensional affine spaces in the *C*-space and vice versa. This result can be interpreted as a bound on the potential perturbation δ_y resulting from a given perturbation δ_x. The application of this result to the AFS plots in the current paper requires that the α and β axes are scaled to the same length units.

Lemma 5.2. For ($s = 3$)-component system let x and $x + \delta_x$ be given in $\mathcal{M}_A$. Further, let C_y and $C_{y+\delta y}$ be the associated one-dimensional affine linear subspaces as determined by Theorem 5.1. Then it holds that

$$\angle\left(x, x+\delta_x\right) = \angle\left(C_y, C_{y+\delta_y}\right).$$

The relation also holds if x and y interchange their positions.

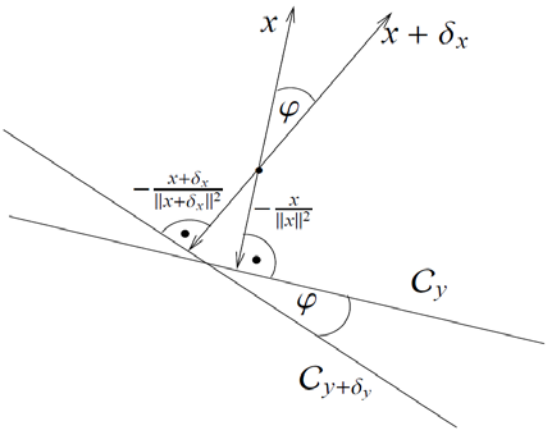


Figure 5.1. *The geometric construction underlying Lemma 5.2 [80].*

Proof. For a given x in $\mathcal{M}_A$ any element y of the complementary space C_y satisfies $(x, y) = x_1 y_1 + x_2 y_2 = -1$. This relation can be rewritten in the Hesse normal form of a straight line

$$\left(-\frac{x}{\|x\|}, y\right) = +\frac{1}{\|x\|}.$$

This means that the C_y is a straight line which is orthogonal to $-x/\|x\|$ and whose smallest distance to the origin is $1/\|x\|$.

Similarly the relation $(x + \delta_x, y + \delta_y) = -1$ can be rewritten as

$$\left(-\frac{x+\delta_x}{\|x+\delta_x\|}, y+\delta_y\right) = +\frac{1}{\|x+\delta_x\|}$$

so that $C_{y+\delta y}$ is a straight line which is orthogonal to $-(x+\delta_x)/\|x+\delta_x\|$ and whose smallest distance to the origin is $1/\|x+\delta_x\|$. The geometric setup is shown in **Figure 5.1**. Simple geometric arguments (on the sum of angles in a triangle) show that the acute angle φ which is enclosed by x and $x+\delta_x$ equals the acute angle enclosed by C_y and $C_{y+\delta y}$. □

5.1.5. Reduction of AFS.

The complementarity theory is only one source for an AFS reduction from pre-given information. Next, three different sources for a reduction of the AFS are listed. We always assume that a single spectrum is known, i.e. a single point in the AFS $\mathcal{M}_A$ is determined. (The same arguments apply if a single concentration profile or single point in the AFS $\mathcal{M}_C$ is given.)

Then three different sources for restrictions on $\mathcal{M}_A$ and $\mathcal{M}_C$ are:

1. Restrictions on the AFS segments of the complementary components in $\mathcal{M}_C$.
2. Restrictions on the concentration profile of the component for which the spectrum is given.
3. Restrictions on the AFS segments for the remaining components in $\mathcal{M}_A$.

In any case the predictions for $\mathcal{M}_C$ by the complementarity theorem are much more restrictive compared to the other criteria.

There are various further options for the reduction of the rotational ambiguity in multivariate curve resolution techniques. For instance hard-modeling by means of a kinetic model and the restrictions on the concentration profiles in *C* have been presented in Ref. [15]. Other important and well established techniques for the ambiguity reduction are the evolving factor analysis (EFA) and window factor

analysis (WFA) and techniques which exploit local rank information in order to extract single pure component spectra and single concentration profiles. For these techniques the Manne theorems are key tools.

It is worth noting that the complementarity theory is a hard constraint due to known spectra and concentration profiles which makes predictions on the remaining unknown parts of *C* and *A*. In this sense EFA and WFA are related to the complementarity theory.

5.2. Calculations of the XANES spectra with the PyFitIt program.

The calculation of theoretical XANES spectra can be carried out using different approaches:

(1) Gradient descent approach. This strategy includes many branches with trial and error steps to optimize the structure. It is based on the comparison of a XANES experimental spectrum with several theoretical calculations by varying selected structural parameters starting from a defined initial geometrical configuration around the absorber. The optimization process is based on the square of the residual function in the parameters space. Multi-dimensional minima search is performed by MINUIT function developed in CERN [25]. Typically, hundreds theoretical spectra are needed to obtain the best fit of the data.

(2) Indirect approach: prediction of a XANES spectrum for a given set of structural parameters. Approach (1) may lead to inconsistent results when several local minima exist in the region of variation of structural parameters. The repetition of the descendent procedure, starting from different initial conditions, can help to find the global minimum but increases significantly the computational time. Alternatively, the studied space of structural parameters can be sampled with some points where XANES spectra are calculated *a priori* before comparison with experimental data. Such approach is realized in FitIt [83]. It performs the multi-dimensional polynomial interpolation of spectra as a function of structural parameters. Once the polynomial interpolation has been constructed, the minima of discrepancy between the experimental XANES spectrum and the interpolated one are searched, by varying the structural parameters, under the application of a gradient

descendent algorithm. In this way, the required number of *ab initio* calculated XANES spectra is considerably smaller with respect to the approach (1).

(3) Direct approach: prediction of structural parameters for a given XANES spectrum. Approach (2) can be inverted to establish correspondence between points in an experimental XANES spectrum $\mu(E_i)$ and related structural parameters. This eliminates the trial-error search of structural parameters providing the minimal discrepancy between the theoretical and experimental XANES spectrum. The structural parameters are predicted directly as a function of the $\mu(E_i)$ variables. Several authors described the application of the direct method to the XANES analysis. Zheng et al. [100] created a large dataset of computed references of XAS spectra calculated in the *muffin-tin* approximation. Afterwards, they applied an Ensembled-Learned Spectra-Identification (ELSI) algorithm to predict the oxidation state and the local environment for a wide set of compounds. The algorithm combines 33 weak "learners" comprising a set of pre-processing steps and a similarity metric, and it can achieve up to 84.2% accuracy. Timoshenko et al. [90] used supervised machine learning (SML) to study 3D structure of supported platinum nanoparticles. The authors constructed an artificial neural network (NN) in Wolfram Mathematica, using an input layer composed by 129 nodes and two hidden layers having 339-387 nodes in each. A hyperbolic tangent function was used for the activation function while the NN was trained creating a dataset of *ab initio* calculated XANES calculations on different size/shape nanoparticles. Their approach allowed to reconstruct the average size, shape, and morphology of well-defined platinum nanoparticles from their XANES spectra [89]. The same group successively extended the method to the EXAFS part of the spectrum processed by wavelets.

5.3. Problems of PCA in PyFitIt.

Since PyFitit is based on PCA it has the following disadvantages, which are:

1) The covariance matrix is difficult to be evaluated in an accurate manner.

2) Even the simplest invariance could not be captured by the PCA unless the training data explicitly provides this information.

Variance is another measure of the spread of data in a data set. It is defined as the mean of the deviation of each term from its arithmetic mean of whole data. In fact it is almost identical to the standard deviation. The formula is this

$$s^2 = Var(X) = \frac{\sum_{i=1}^{n}(X_i - X')(X_i - X')}{(n-1)}$$

X' = Arithmetic mean

$X_i - X'$ = Deviation of individual observation from arithmetic mean,

n = Number of observations.

Many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions. For example, we might have as our data set both the height of all the students in a class and the mark they received on genetics paper. We could then perform statistical analysis to see if the height of a student has any effect on their mark. Standard deviation and variance only operate on one dimension, so that we could only calculate the standard deviation for each dimension of the data set independently of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other. Covariance is such a measure. Covariance is always measured between 2 dimensions. If we calculate the covariance between one dimension and itself, we will get the variance. So, if we had a 3-dimensional data set (A, B, C), then we could measure the covariance between the A and B dimensions, the B and C dimensions, and the A and C dimensions. Measuring the covariance between A and A, or B and B, or C and C would give us the variance of the A, B and C dimensions respectively. The formula for covariance is very similar to the formula for variance. The formula for covariance with respect to variance could also be written

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - X')(Y_i - Y')}{(n-1)}$$

where, X' = Arithmetic mean of data X, Y' = Arithmetic mean of data Y, n = Number of observation.

As mentioned above if we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated. For example, from

a 3 dimensional data set (dimensions *x, y, z*) we could calculate the *cov*(*x*,*y*), *cov*(*y*,*z*) and *cov*(*x*,*z*). In fact, for an *n*-dimensional data set, we can calculate $\frac{n!}{(n-2)!*2}$ different covariance values [96].

A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix. So, by definition the covariance matrix for a set of data with *n*-dimensions is

$$C^{M*N} = \left(c_{i,j}, c_{i,j} = \mathrm{cov}\left(Dim_i, Dim_j\right)\right).$$

Where, C^{M*N} is a matrix with *n* rows and *n* columns, and (*Dim x*) is the x^{th} dimension.

This typical formula says that if you have an *n*-dimensional data set, then the matrix has *n* rows and columns (so is square) and each entry in the matrix is the result of calculating the covariance between two separate dimensions.

E.g. the entry on row 2, column 3, is the covariance value calculated between the 2nd dimension and the 3rd dimension.

Properties of Variance and Covariance.

1) $Var\left(\mathbf{a}^T X\right) = \mathbf{a}^T Var(X)\mathbf{a} = \sum_{i,j} a_i a_j \sigma_{X_i X_j}$

2) $Var(\mathbf{A}X + b) = \mathbf{A}Var(X)\mathbf{A}^T$

3) $Var(X + Y) = Var(X) + Cov(X, Y) + Cov(Y, X) + Var(Y)$

4) $Cov(X + Y, Z) = Cov(X, Z) + Cov(Y, Z)$

5) $Cov(\mathbf{A}X, \mathbf{B}Y) = \mathbf{A}Cov(X, Y)\mathbf{B}^T$

The correlation matrix **R** formula can be rewritten in algebra matrix

$$r_{X_i X_j} = corr\left(X_i, X_j\right) = \frac{\mathrm{cov}\left(X_i, X_j\right)}{\sqrt{\mathrm{var}\left(X_i\right)\mathrm{var}\left(X_j\right)}} = \tag{5.14}$$

$$= \frac{1}{\sqrt{\mathrm{var}\left(X_i\right)}}\mathrm{cov}\left(X_i, X_j\right)\frac{1}{\sqrt{\mathrm{var}\left(X_j\right)}}.$$

Let **D** be a diagonal matrix such that the diagonal elements are the same as those of the covariance matrix **S** i.e. ($d_{ii} = s_{ii}$). From (5.14) the relation between the correlation matrix and the covariance matrix is given by (5.15).

$$\mathbf{R} = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2}. \tag{5.15}$$

Assume a random vector X, taking values in $\Re^m$, has a mean and covariance matrix of μ_X and $\sum_X$, respectively. $\lambda_1 > \lambda_2 > \cdots > \lambda_m > 0$ are ordered eigenvalues of $\sum_X$, such that the i-th eigenvalue of $\sum_X$ means the i-th largest of them. Similarly, a vector α_i is the i-th eigenvector of $\sum_X$ when it corresponds to the i-th eigenvalue of $\sum_X$. To derive the form of principal components (PCs), consider the optimization problem of maximizing $\text{var}[\alpha_1^T X] = \alpha_1^T \sum_X \alpha_1$, subject to $\alpha_1^T \alpha_1 = 1$. The Lagrange multiplier method is used to solve this question.

$$L(\alpha_1, \phi_1) = \alpha_1^T \sum_X \alpha_1 + \phi_1(\alpha_1^T \alpha_1 - 1)$$

$$\frac{\partial L}{\partial \alpha_1} = 2\sum_X \alpha_1 + 2\phi_1\alpha_1 = 0 \Rightarrow \sum_X \alpha_1 = -\phi_1\alpha_1 \Rightarrow \text{var}[\alpha_1^T X] = -\phi_1\alpha_1^T\alpha_1 = -\phi_1.$$

Because $-\phi_1$ is the eigenvalue of $\sum_X$, with α_1 being the corresponding normalized eigenvector, $\text{var}[\alpha_1^T X]$ is maximized by choosing α_1 to be the first eigenvector of $\sum_X$. In this case, $z_1 = \alpha_1^T X$ is named the first PC of X, α_1 is the vector of coefficients for z_1, and $\text{var}(z_1) = \lambda_1$ [38].

To find the second PC, $z_2 = \alpha_2^T X$, we need to maximize $\text{var}[\alpha_2^T X] = \alpha_2^T \sum_X \alpha_2$ subject to z_2 being uncorrelated with z_1. Because $\text{cov}(\alpha_1^T X, \alpha_2^T X) = 0 \Rightarrow \alpha_1^T \sum_X \alpha_2 = 0 \Rightarrow \alpha_1^T \alpha_2 = 0$, this problem is equivalently set as maximizing $\alpha_2^T \sum_X \alpha_2$, subject to $\alpha_1^T \alpha_2 = 0$, and $\alpha_2^T \alpha_2 = 1$. We still make use of the Lagrange multiplier method.

$$L(\alpha_2, \phi_1, \phi_2) = \alpha_2^T \sum_X \alpha_2 + \phi_1\alpha_1^T\alpha_2 + \phi_2(\alpha_2^T\alpha_2 - 1)$$

$$\frac{\partial L}{\partial \alpha_2} = 2\sum_X \alpha_2 + \phi_1\alpha_1 + 2\phi_2\alpha_2 = 0$$

$$\Rightarrow \alpha_1^T(2\sum_X \alpha_2 + \phi_1\alpha_1 + 2\phi_2\alpha_2) = 0 \Rightarrow \phi_1 = 0$$

$$\Rightarrow \sum_X \alpha_2 = -\phi_2\alpha_2 \Rightarrow \alpha_2^T \sum_X \alpha_2 = -\phi_2 .$$

Because $-\phi_2$ is the eigenvalue of $\sum_X$, with α_2 being the corresponding normalized eigenvector, $\text{var}[\alpha_2^T X]$ is maximized by choosing α_2 to be the second eigenvector of $\sum_X$. In this case, $z_2 = \alpha_2^T X$ is named the second PC of X, α_2 is the vector of coefficients for z_2, and $\text{var}(z_2) = \lambda_2$. Continuing in this way, it can be shown that the i-th PC $z_i = \alpha_i^T X$ is constructed by selecting α_i to be the i-th eigenvector of $\sum_X$, and has variance of λ_i [70]. The key result in regards to PCA is that the principal components are the only set of linear functions of original data that are uncorrelated and have orthogonal vectors of coefficients.

Proposition 5.1 [Jolliffe, 2002]. For any positive integer $p \le m$, let $B = [\beta_1, \beta_2, \ldots, \beta_p]$ be an real $m \times p$ matrix with orthonormal columns, i.e., $\beta_i^T \beta_j = \delta_{ij}$, and $Y = B^T X$. Then the trace of covariance matrix of Y is maximized by taking $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$, where α_i is the i-th eigenvector of $\sum_X$.

Proof. Because $\sum_X$ is symmetric with all distinct eigenvalues, so $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ is an orthonormal basis with α_i being the i-th eigenvector of $\sum_X$, and we can represent the columns of B as

$$\beta_i = \sum_{j=1}^{m} c_{ji}\alpha_j , \quad i = 1, \ldots, p .$$

So we have

$$B = PC$$

where $P = [\alpha_1, \ldots, \alpha_m]$, $C = \{c_{ij}\}$ is an $m \times p$ matrix. Then, $P^T \sum_X P = \Lambda$, with Λ being a diagonal matrix whose k-th diagonal element is λ_k, and the covariance matrix of Y is,

$$\sum_Y = B^T \sum_X B = C^T P^T \sum_X PC = C^T \Lambda C = \lambda_1 c_1 c_1^T + \cdots + \lambda_m c_m c_m^T$$

where c_i^T is the i-th row of C. So,

$$trace(\sum_Y) = \sum_{i=1}^{m} \lambda_i trace(c_i c_i^T) = \sum_{i=1}^{m} \lambda_i trace(c_i^T c_i) = \sum_{i=1}^{m} \lambda_i c_i^T c_i = \sum_{i=1}^{m} (\sum_{j=1}^{p} c_{ij}^2) \lambda_i .$$

Because $C^T C = B^T P P^T B = B^T B = I$, so $trace(C^T C) = \sum_{i=1}^{m} \sum_{j=1}^{p} c_{ij}^2 = p$, and the columns of C are orthonormal. By the Gram-Schmidt method, C can expand to D, such that D has its columns as an orthonormal basis of $\Re^m$ and contains C as its first p columns. D is square shape, thus being an orthogonal matrix and having its rows as another orthonormal basis of $\Re^m$. One row of C is a part of one row of D, so $\sum_{j=1}^{p} c_{ij}^2 \le 1$, $i = 1, \ldots, m$. Considering the constraints $\sum_{j=1}^{p} c_{ij}^2 \le 1$, $\sum_{i=1}^{m} \sum_{j=1}^{p} c_{ij}^2 = p$ and the objective $\sum_{i=1}^{m} (\sum_{j=1}^{p} c_{ij}^2) \lambda_i$. We derive that $trace(\sum_Y)$ is maximized if $\sum_{j=1}^{p} c_{ij}^2 = 1$ for $i = 1, \ldots, p$, and $\sum_{j=1}^{p} c_{ij}^2 = 0$ for $i = p+1, \ldots, m$. When $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$, straightforward calculation yields that C is an all-zero matrix except $c_{ii} = 1$, $i = 1, \ldots, p$. This fulfills the maximization condition. Actually, by taking $B = [\gamma_1, \gamma_2, \ldots, \gamma_p]$, where $\{\gamma_1, \gamma_2, \ldots, \gamma_p\}$ is any orthonormal basis of the subspace of $span\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$, the maximization condition is also satisfied, thus yielding the same trace of covariance matrix of Y. □

Proposition 5.2 [Jolliffe, 2002]. Suppose that we wish to approximate the random vector X by its projection onto a subspace spanned by columns of B, where $B = [\beta_1, \beta_2, \ldots, \beta_p]$ is a real $m \times p$ matrix with orthonormal columns, i.e., $\beta_i^T \beta_j = \delta_{ij}$. If σ_i^2 is the residual variance for each component of X, then $\sum_{i=1}^{m} \sigma_i^2$ is minimized if $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$, where $\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ are the first p eigenvectors of $\sum_X$. In other words, the trace of convariance matrix of $X - BB^T X$ is minimized if $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$. When $E(X) = 0$, which is a commonly applied preprocessing step in data analysis methods, this property is saying that

$E\left\|X - BB^T X\right\|^2$ is minimized if $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$.

Proof. The projection of a random vector X onto a subspace spanned by columns of B is $\hat{X} = BB^T X$. Then the residual vector is $\varepsilon = X - BB^T X$, which has a convariance matrix

$$\sum_{\varepsilon} = (I - BB^T)\sum_X (I - BB^T).$$

Then,

$$\sum_{i=1}^{m} \sigma_i^2 = trace(\sum_{\varepsilon}) = trace(\sum_X - \sum_X BB^T - BB^T \sum_X + BB^T \sum_X BB^T).$$

Also, we know

$$trace(\sum_X BB^T) = trace(BB^T \sum_X) = trace(B^T \sum_X B)$$

$$trace(BB^T \sum_X BB^T) = trace(B^T \sum_X BB^T B) = trace(B^T \sum_X B).$$

The last equation comes from the fact that B has orthonormal columns.

So,

$$\sum_{i=1}^{m} \sigma_i^2 = trace(\sum_X) - trace(B^T \sum_X B).$$

To minimize $\sum_{i=1}^{m} \sigma_i^2$, it suffices to maximize $trace(B^T \sum_X B)$. This can be done by choosing $B = [\alpha_1, \alpha_2, \ldots, \alpha_p]$, where $\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$ are the first p eigenvectors of $\sum_X$, according to Proposition 5.1 stated above. □

6. A STRATEGY OF SCIENTIFIC IDENTIFICATION OF PURE SPECTRAL AND CONCENTRATION PROFILES

Let's consider a XANES spectrum $\boldsymbol{\mu}_i$ characterized by L energy points. It is part of a XANES data matrix $\mathbf{X}$ composed by M spectra (i.e. $\dim(\mathbf{X}) = \mathrm{L} \times \mathrm{M}$). By the Lambert-Beer equation (0.1), it is possible to express $\boldsymbol{\mu}_i$ as the sum of N pure spectra s_i weighted by their related concentration profiles c_{ij}:

$$\boldsymbol{\mu}_i = \sum_{j=1}^{N} c_{ij}\boldsymbol{s}_j \tag{6.1}$$

It is possible to demonstrate by means of PCA that each spectrum $\boldsymbol{s}_j$ is rewritable in the following way:

$$\boldsymbol{s}_j = \sum_{k=1}^{N} t_{jk}\boldsymbol{v}_k \tag{6.2}$$

Where $\boldsymbol{v}_k$ are a set of orthogonal vector that can be obtained by the diagonalization of the covariance matrix associated to $\mathbf{X}$ i.e.

$$R = \frac{\mathbf{X} \cdot \mathbf{X}^t}{M}$$

Matrix t_{jk} is an unknown matrix. In PyFitIt it is possible to access to its elements by means of sliders. Moreover, their number can be reduced through constraints. One of the most widely used is the so called "normalization" that fixes the first row of t_{jk} as:

$$t_{jk} = \begin{pmatrix} 1/a & \dots & 1/a \\ t_{21} & \dots & t_{2N} \\ \dots & \dots & \dots \\ t_{N1} & \dots & t_{NN} \end{pmatrix} \tag{6.3}$$

$$a = \sqrt{\frac{1}{E_{max} - E_{min}} \int_{E_{min}}^{E_{max}} dE\,\boldsymbol{\mu}_i(E)^2} \tag{6.4}$$

Where a is the normalization parameter and E_{max}, E_{min} are respectively the maximum and minimum values of energy.

To retrieve a set of chemical/physical meaningful spectra and their related concentration profiles it is necessary to satisfy the following requirements:

1) $\boldsymbol{s}_j \geq 0$: Non-negativity of spectral profiles
2) $0 \leq c_{ij} \leq 1$ with $\sum_{j=1}^{N} c_{kj} = 1$ for every k point: non negativity of the concentration profiles
3) Mass balance condition

These constraints can influence the values of t_{jk} leading to the formation of a determined AFS. These regions can be obtained via "brute force" (i.e. varying randomly each element t_{jk} and collecting the values satisfying points number 1) and 2) and 3)) or minimizing a penalty function.

This second strategy seems to be more accessible especially because it allows one to save an higher amount of computational time. The AFS, then, can be recovered minimising the following equation:

$$P(t_{21} \dots t_{2N} \dots t_{N1} \dots t_{NN}) = \sum_{i=1}^{L} \sum_{j=1}^{N} H_s(s_{ij})(s_{ij})^2 + \sum_{k=1}^{M} \sum_{j=1}^{N} H_c(c_{kj})(c_{kj})^2 \quad (6.5)$$

Where P is the "penalty function", while H_s and H_c are two step function, which preserve requirements 1) and 2):

$$H_s(x) = \begin{cases} 1 \; if \; x < 0 \\ 0 \; if \; x \geq 0 \end{cases}$$

$$H_c(x) = \begin{cases} 1 \; if \; x < 0 \; or \; x > 1 \\ 0 \; if \; 0 \leq x \leq 1 \end{cases}$$

6.1. Dataset 1.

As a first test dataset for this method, we considered a set of XANES composed of 30 spectra acquired during the MTM process (methane to methanol conversion: CH_4 loading and CH_3OH extraction by means of water) catalysed by the

Cu-zeolite characterized by the Ferrierite (FER) framework: 0.20Cu-H, Na-FER(11). The details of the experiment together with the description of the reaction can be found in Ref. [60].

The test of this dataset is reported below:

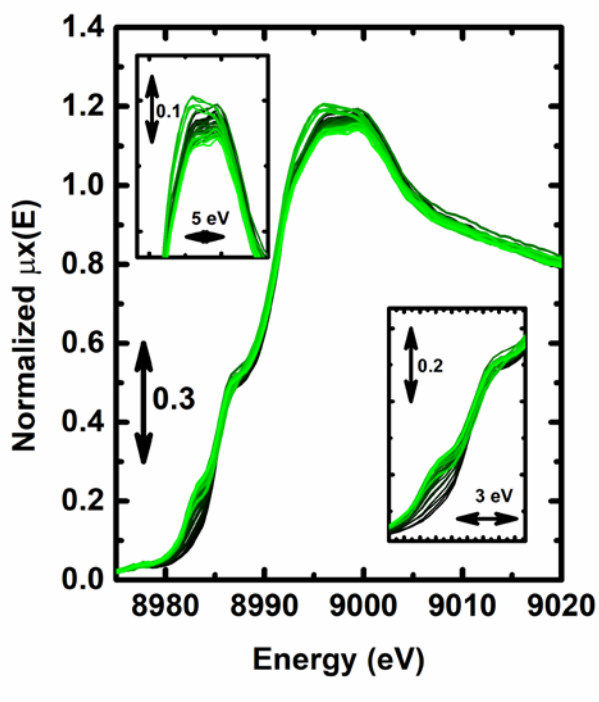


Figure 6.1. *XANES dataset used for this study composed of 30 spectra acquired during the MTM conversion.*

As it is possible to see from **Figure 6.1**, the total dataset variation is quite low. The PCA executed on it reveals the presence of three main components:

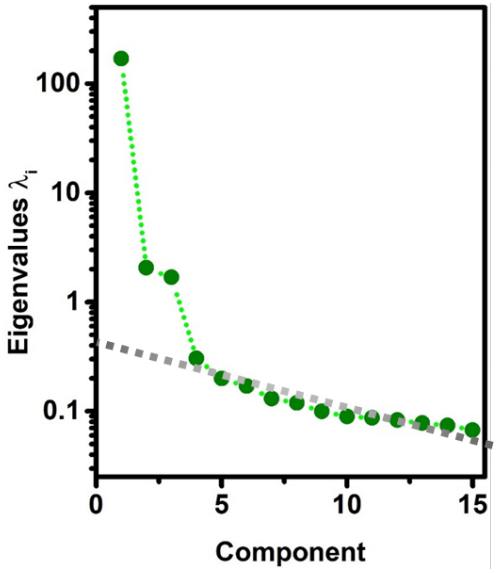


Figure 6.2. *Scree Plot of the analysed dataset. As it is possible to see the elbow of this curve is localised in proximity of the third PC.*

A first attempt for the spectral separation can be realised by means of the MCR-ALS approach, initializing the ALS routine using the SIMPLISMA algorithm [26]. The results are reported in **Figure 6.3**:

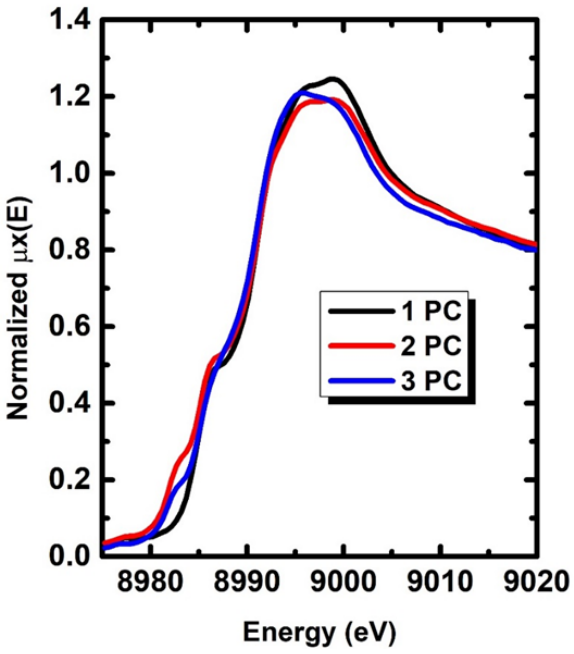


Figure 6.3. *Principal components/pure spectra extracted from MCR-ALS.*

As it is possible to observe, the pure spectral profiles extracted by this technique do not have any chemical/physical meaning. Thus, we proceeded to apply the PyFitIt method based on the application of a transformation matrix.

Considering 3PCs the number of sliders to move corresponds to nine. However, imposing the normalization constraint this number is reduced to six. Due to the finer energy step (0.005 eV), using the normalization, condition 3) is satisfied. Sliders are then moved basing on the the spectral and concentration profiles non negativity trends. Finally, we required, for all the components, that the XANES white line magnitude could not be higher than 2 (unrealistic solution for this system) and that the highest value of concentration is one. All these constraints have been implemented in a proper penalty function using equation (6.5).

The AFS obtained with this method are reported in **Figure 6.4**:

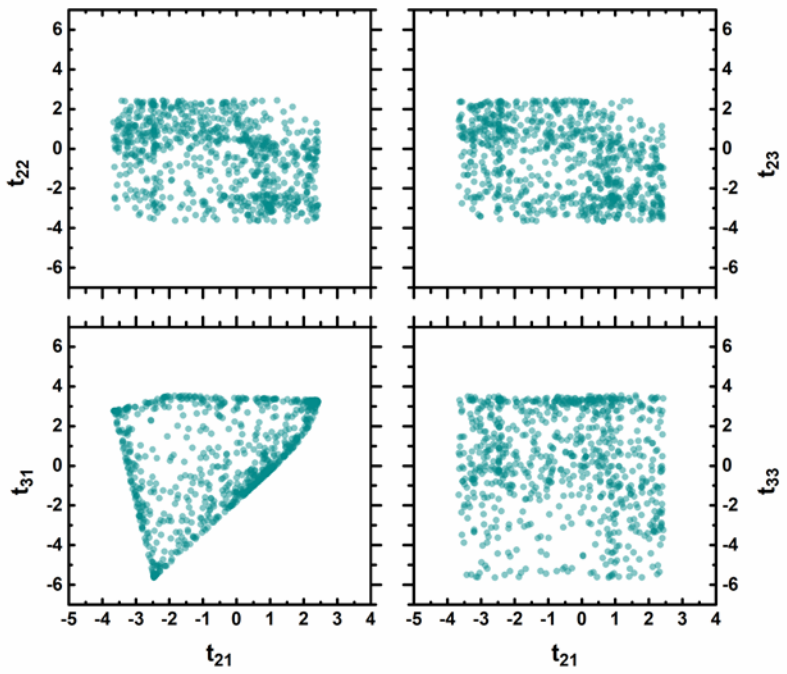


Figure 6.4. *Some selected plots representing the AFS obtained minimizing function (6.5) using 1000 initialization set of points chosen randomly between -10 and 10. The minimization procedure has been realised using the Nelder-Mead algorithm.*

Once the AFS have been defined, it is possible to identify, moving the elements of matrix (6.3), the set of spectra $\boldsymbol{s}_j$ closest to given references. It is worth noting that the results obtained by this method are not unique and this approach cannot be defined as a "blind method" as MCR-ALS.

The first method foresees the comparison of each spectrum calculated for every point with the reference. The comparison is performed minimizing the L^2 norm of the following difference:

$$\min_{(t_{21}\dots t_{2N}\dots t_{N1}\dots t_{NN})} \left(\sqrt{\sum_{i=1}^{L} (s_i(t_{21}\dots t_{2N}\dots t_{N1}\dots t_{NN}) - XANES_i)^2 \Delta E} \right) \quad (6.6)$$

In case of a multi-values solution, a second reference spectrum can be added to the minimization then equation (6.6) becomes:

$$\min_{(t_{21}\dots t_{2N}\dots t_{N1}\dots t_{NN})} \left(\sqrt{\sum_{j=1}^{N}\sum_{i=1}^{L} \left(s_{ij}(t_{21}\dots t_{2N}\dots t_{N1}\dots t_{NN}) - XANES_{ij}\right)^2 \Delta E} \right) \quad (6.7)$$

Generally, the PyFitIt user must exploy his/her chemical/physical intuition in order to isolate some results that can be interpretable. A meaningful solution is reported in **Figure 6.5** together with some selected references used for the comparison.

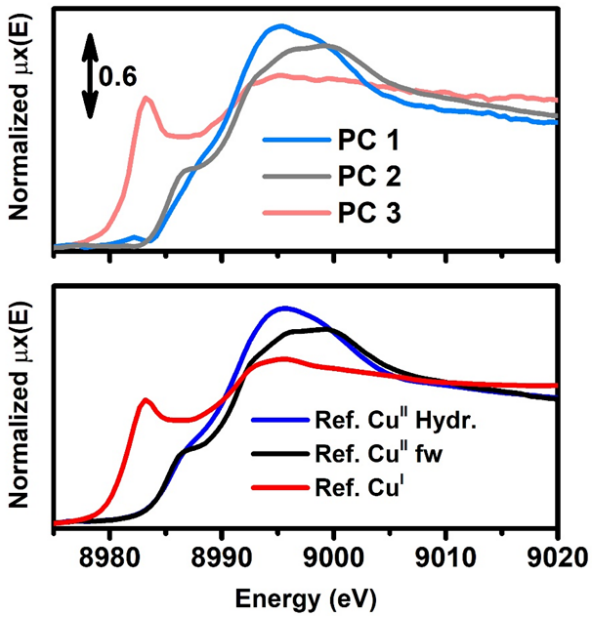


Figure 6.5. *In the upper panel a feasible set of spectra is reported, having a chemical/physical meaning while in the bottom panel three XANES references are shown used for the comparison. The Cu^{II} hydrated state has been taken at RT (25 °C)*

in a water solution of Cu acetate. The Cu^{I} reference is referring to a state of a Cu site in the same FER framework taken at 400 °C after two hours of activation in vacuum. The black spectrum is related to a Cu^{II} site of the FER framework. This spectrum has been obtained at 400 °C in oxygen atmosphere after two hours of activation.

How it is possible to see from **Figure 6.5**, the similarity between the pure spectral set extracted by PyFitIt and the references spectra is extremely high. Looking at the related concentrations profiles in **Figure 6.6**, more information about the chemistry of the process can be extracted:

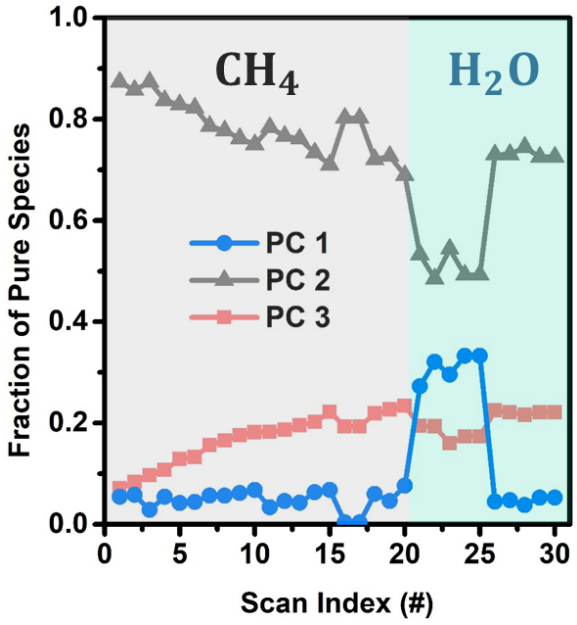


Figure 6.6. *Concentration profiles extracted by PyFitIt during the loading of CH_4 and the extraction of CH_3OH.*

After the interaction of the O_2-activated catalyst with CH_4 at 200 °C it is possible to observe the partial reduction of Cu^{II} to Cu^{I} . The reduction of Cu^{II} to Cu^{I} during CH_4 loading is in agreement with the formation of a reaction intermediate i.e. methoxy species, as suggested in the literature [3, 46, 58-59, 86].

Recently, the percentage of reduced Cu from the interaction of active sites with CH_4 has been taken as a descriptor of the productivity. Finally, the introduction of steam in the reactor used to release CH_3OH, correlates with the formation of an hydrate state (the blue-one component) and involves the oxidation of a fraction of the

Cu(I) ions. More information about the chemical results obtained by these experiments can be found in [60].

As it is possible to see, the solution reported in **Figure 6.5** and **Figure 6.6** is not unique. The idea to minimize the R-factor related to the reconstruction of the dataset, that should be able to provide a well defined solution, can not be followed. The reason is easily explained having a look at the cumulative variance explained by the three components (that is higher than 90%) and remembering that the total variation to the right and left singular values provided by sliders is zero. At the moment, the possibility to find a global objective function that minimized, under constraints, is able to provide a unique, chemical/physical meaningful solution, is still under development. However, we were able to identify the maxima and the minima meaningful spectral and concentrations boundaries under constraints.

We defined an objective function as the ratio between the Frobenius norm of the signal contribution of a particular species with respect to the Frobenius norm of the whole signal for all the considered species. In this way, the objective function has a scalar output within 0 and 1, that is a good approach for an further optimization process.

$$f_k(t_{21} \dots t_{2N} \dots t_{N1} \dots t_{NN}) = \frac{\|\boldsymbol{s}_k \boldsymbol{c}_k{}^{\mathrm{T}}\|}{\|\mathbf{X}\|} \tag{6.8}$$

The optimization of this objective function, either maximized or minimized, will give respectively the maximum and the minimum boundary for the *k*-species. The definition of this objective function requires the initial values of the variables in the transformation matrix together with the initial values for the species profiles, $\boldsymbol{s}_k$ and $\boldsymbol{c}_k$. When the optimization is implemented as a minimization of the objective function it is appropriate to find the minimum band boundaries, whereas if the optimization is implemented as a maximization of the objective function, the functions given in Equations (6.8) and should be changed in sign.

Clearly, without constraints, equation (6.8) can not be minimized; the usage of some constraints is then required. Moreover, a proper set of spectra as initialization profiles must be provided in order to find meaningful boundaries. The extension of the minimization of equation (6.8) in a large range of variation of elements of the transformation matrix can lead to an overestimation of the boundaries profiles. For this reason, we considered a hypercube, having an initial

side of 0.25 and centered in the set of parameters that provides **Figure 6.5** and **Figure 6.6**. In order to get meaningful boundaries, equation (6.8) has been minimized together with equation (6.5) in the following way:

$$f_k(t_{21} \dots t_{2N} \dots t_{N1} \dots t_{NN}) = \frac{\|\boldsymbol{s}_k \boldsymbol{c}_k{}^{\mathrm{T}}\|}{\|\mathbf{X}\|} + \gamma P(t_{21} \dots t_{2N} \dots t_{N1} \dots t_{NN}) \qquad (6.9)$$

Herein, the penalty function guarantees to obtain a solution located in the AFS. The parameter γ works as a Lagrange multiplier. We minimized equation (6.9) iteratively, using the Sequential Least Squares Programming Algorithm (SLSQP) provided by the Sci.py package of python, which allows one to impose constraints on the variation of the t_{ij} elements. For each iteration, the solution of equation (6.9) is used as the initialization of the following cycle. At the same time, the γ term is progressively increased with a step of 10 for each cycle. This algorithm is globally quite long to find a convergence (usually in 10 iterations), however it allows to reach it. A proper set of boundaries is reported below:

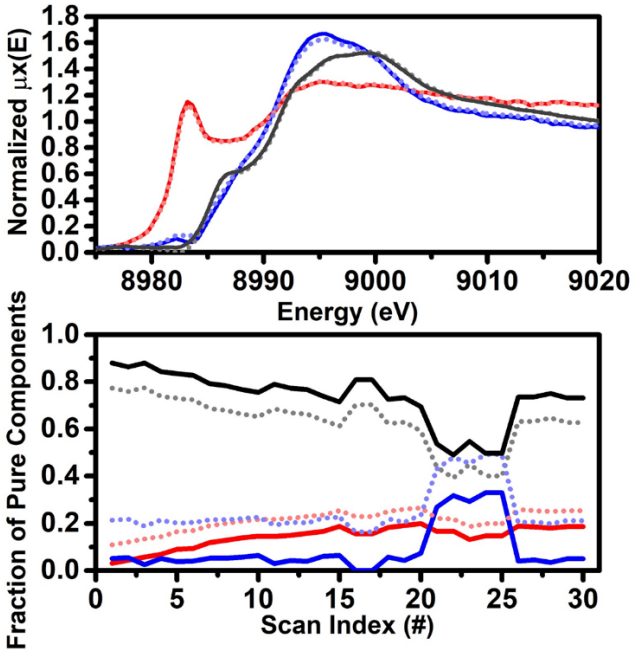


Figure 6.7. *Spectral and concentration boundaries obtained minimizing equation (6.9) and using as initial guess spectra the set reported in* **Figure 6.5**. *Colour legend: blue tonality: Cu^{II} hydr. specie, red tonality Cu^{I} specie, black tonality Cu^{II} framework specie.*

It is interesting to note that with this kind of constraints, only small spectral variations are detectable. The most evident regards the white line of the Cu^{II} hydr. state and the pre edge features of Cu^{II}-oxo specie. On the contrary, in the field of concentration profiles, high variations are evident, in particular the critical point is related to the maximum Cu^{II} hydr. boundary profile which seems to be overestimated in proximity of the first scan number (#). In fact the first scan should correspond to an activated state obtained cooling down the sample from 200 °C to 400 °C in oxygen. The Cu^{II} hydr. state is an indicator of water in the system and, for this reason, an amount comparable to 20% of the mixture is unfeasible. The range of variation of sliders can be then reduced to 0.1. Using this constraint, the total amount the Cu^{II} hydr. component, in the first scan goes under the 8%. This maximum estimation is reasonable and can be interpreted considering the fact that cooling down the temperature, some Cu sites can rearrange their local environment assuming an hexahedral/octahedral configuration or that all the water molecules are not completely deleted from the framework. A similar result can be obtained fitting the first spectrum of the dataset using references of **Figure 6.5** as was made by Pappas et al in [60]. The resulting concentrations profiles are reported below:

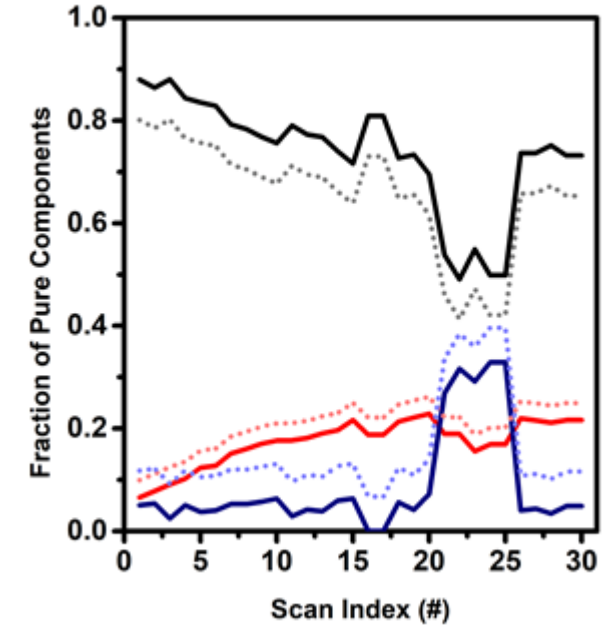


Figure 6.8. *Concentrations profiles retrieved considering a hypercube of side 0.1.*

It is interesting to observe that reducing the side of the hypercube of variation the concentration profiles of the Cu^{I} and Cu^{II} species do not change, remaining globally stable.

The method still remains not unique. In fact the size of the minimization range must be defined by user. However, a general procedure can be described in this way. It is possible to increase progressively the size of the minimization range. For each minimization cycle, determined spectral or concentration features (such us some unfeasible pecks) can be monitored. Their appearance will correspond to the maximum variation for the set of transformation elements associated with the spectra having these features. Once the limits of variation of each element of the transformation matrix have been defined, the minimization process can be applied as showed before.

The program code for the aforesaid strategy was developed in programming language Python in web-based, interactive computing notebook environment Jupyter in collaboration with researches of the university of Turin.

The program code is reported in **Appendix A**.

6.2. Datasets 2-3.

The next datasets are represented by the spectra XAS for the reactions:

1) PhSMe + Fe^{IV}N4Py(O) – **dataset 2**;
2) $PhCH_2OH$ + Fe^{IV}N4Py(O) – **dataset 3**.

These reactions are the reactions of the substrates PhSMe and $PhCH_2OH$ for which the kinetic study has been already published in Ref. [11] so that it is possible to obtain the XANES spectra of the pure species. The concentrations were obtained by applying the program PyFitIt to the succession of the UV-Vis spectra of the reactions, and normalizing the concentrations to 1. In both reactions the number of components is equal to 2: the forma is relative to the reagent and the latter to the product.

The process for the identification of the pure spectral and concentration profiles and their related range of confidence for dataset 2 and dataset 3 is analogical. The program code for this process for dataset 2 and dataset 3 is identical and it was

written in programming language Python in web-based, interactive computing notebook environment Jupyter in collaboration with researches of the university of Turin. The program code is reported in **Appendix B**.

6.2.1. Dataset 2.

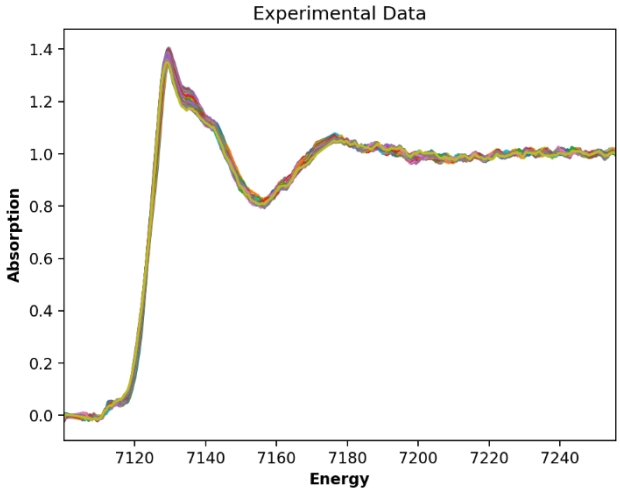


Figure 6.9. *XAS spectra for the reaction PhSMe + Fe^{IV}N4Py(O).*

In **Figure 6.9** the XAS dataset composed by 49 spectra.

In order to perform a correct decomposition of the input dataset in a reduced set of *pure* independent spectra, each experimental spectrum $\boldsymbol{\mu}_i$ must be normalised using the formula (6.4). It is worth noting that the higher the number of points characterizing each experimental spectrum, the more accurate the normalization results. For this reason, before this step, the experimental spectra were interpolated with a finer sampling interval changing the step parameter in the interpolation function (energy, data, step=0.005). By this function, the user (modifying the energy step between two consecutive points by the "step" parameter) can increase the number of points for each spectrum in the experimental dataset [84]. On the other hand, if the user wishes to use the original energy range, this step can be skipped.

The normalized spectra are shown in **Figure 6.11** and their related normalized concentration profiles are shown in **Figure 6.10**.

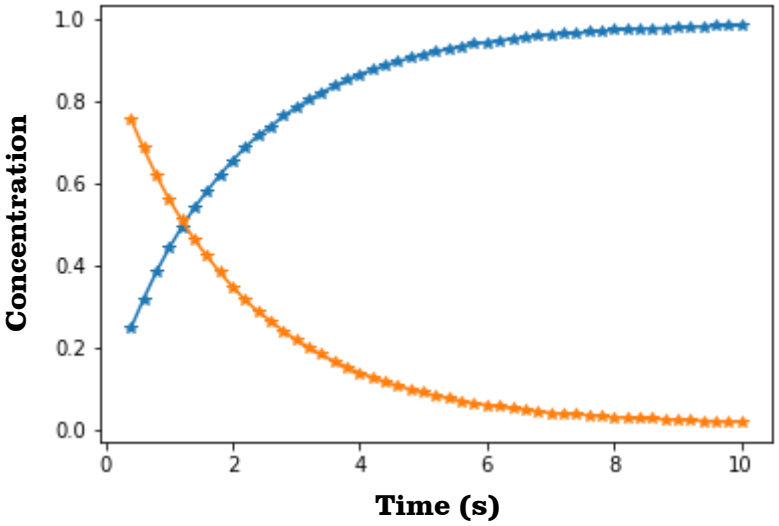


Figure 6.10. *Normalized concentration profiles.*

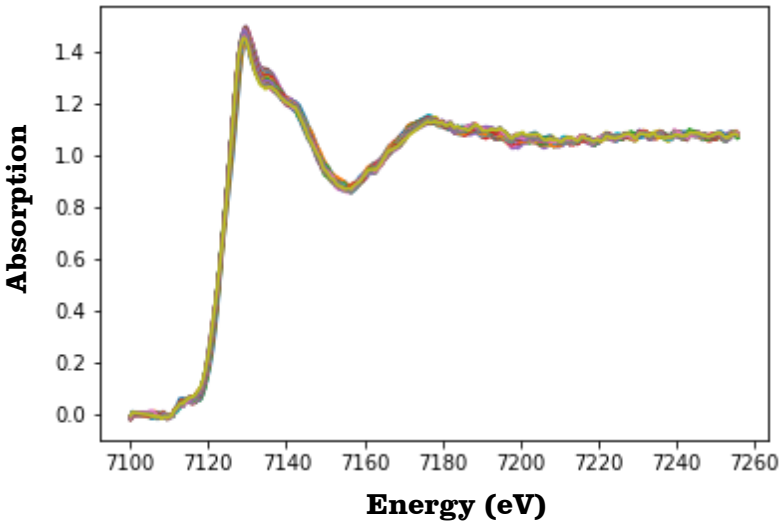


Figure 6.11. *Normalized spectra.*

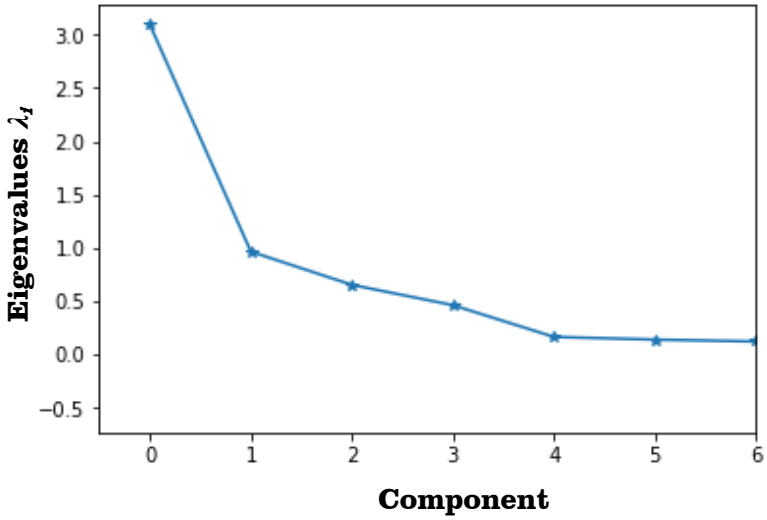


Figure 6.12. *Number of principal components.*

In **Figure 6.12** the number of principal components is reported and it is equal to 2. The qualitative analysis of the PCs extracted by SVD can help to identify the correct number of components related to physical variation of signals. It is worth noting that the number of points characterising each PC is equal to the number of points of the energy column. This fact implies that each PC can be plotted *vs* the column energy and interpreted as an *abstract spectrum* without any chemical/physical meaning.

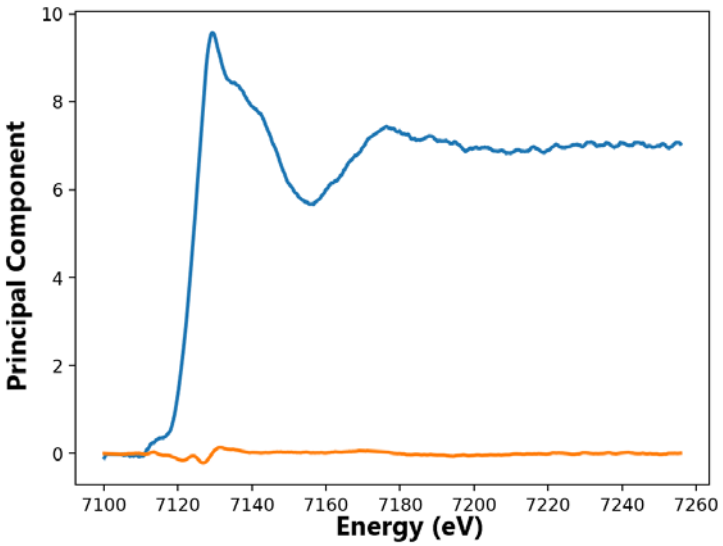


Figure 6.13. *Example of the spectra to be multipled with the elements of the transformation matrix T.*

As it is possible to observe, the pure spectral profiles in **Figure 6.13** do not have any chemical/physical meaning.

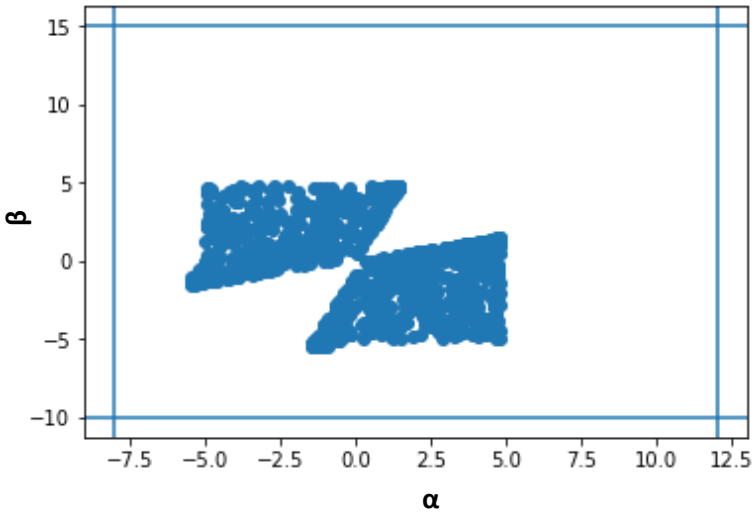


Figure 6.14. *The AFS obtained by minimizing function (6.5).*

The AFS in **Figure 6.14** is obtained using 1000 initialization sets of points chosen randomly between -5 and 5. The minimization procedure has been carried out using the Nelder-Mead algorithm.

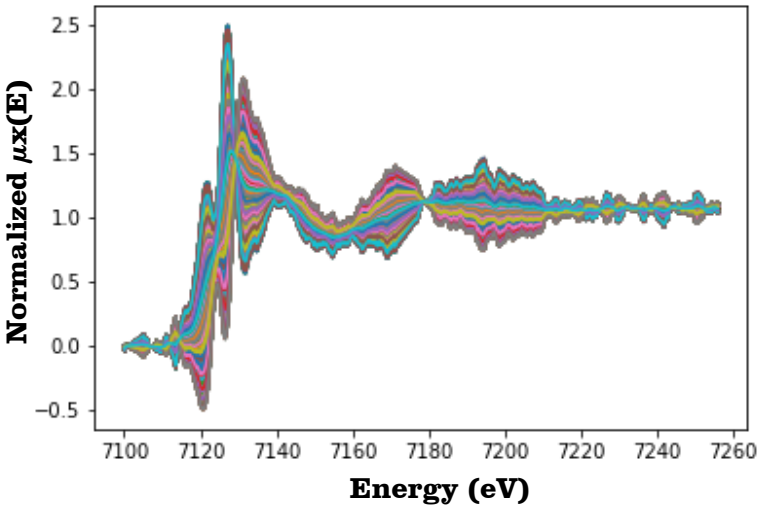


Figure 6.15. *Variation of the spectra as a function of estimated parameters.*

Estimated parameters are the matrix containing 1000 rows and 2 columns. The first row of this matrix contains estimated parameters t_1, t_2 obtained during the first iteration of the AFS calculation process. The second row of this matrix contains estimated parameters t_1, t_2 obtained during the second iteration of the AFS calculation. And so on until the 1000^{th} row of the matrix, which contains estimated parameters t_1, t_2 obtained during the 1000^{th} iteration of the AFS calculation. Spectra of estimated parameters are represented in **Figure 6.15**.

In **Figure 6.16** the maximum and the minimum meaningful boundary of concentrations profiles are reported. In **Figure 6.17** the maximum and the minimum meaningful boundary of spectral profiles are reported.

The results shown in **Figures 6.16-6.17** were obtained by minimizing equation (6.9) iteratively, using the Sequential Least Squares Programming Algorithm (SLSQP). Minimization was performed by the Sci.py package of python, which imposes constraints on the variation of the t_{ij} elements. The set reported in **Figure 6.15** was used as initial guess spectra.

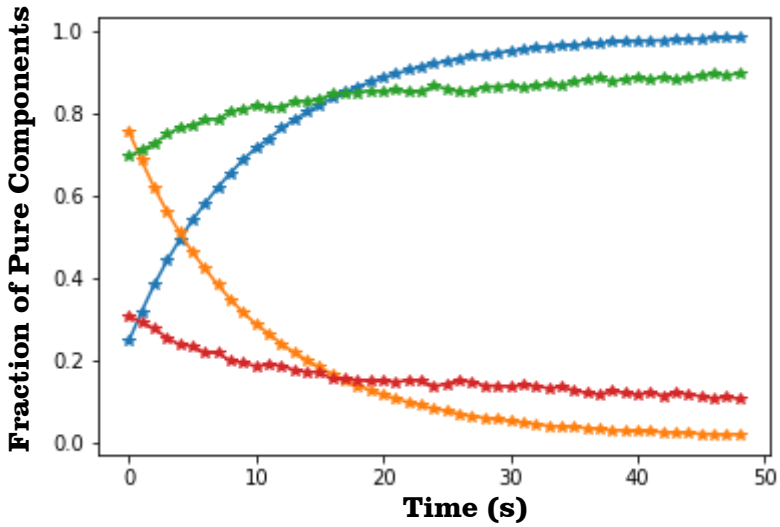


Figure 6.16. *Concentration boundaries obtained minimizing the objective function (6.9) with constraints by method SLSQP.*

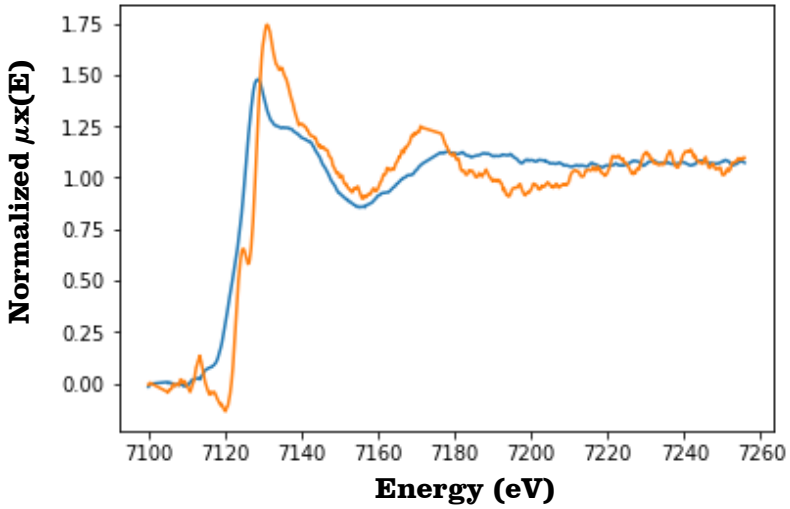


Figure 6.17. *Spectral boundaries obtained minimizing the objective function (6.9) with constraints by method SLSQP.*

It is interesting to note that with this kind of constraints, a big concentration profiles variation is detectable in **Figure 6.16** as compared to the recovered concentration profiles reported in **Figure 6.18**.

A big spectral variation is detectable in **Figure 6.17** as compared to the recovered spectra reported in **Figure 6.19**.

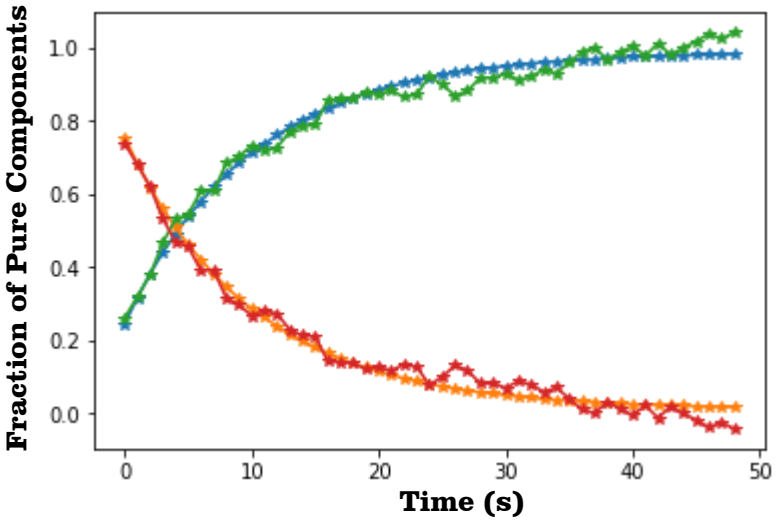


Figure 6.18. *Concentration profiles of the species present in solution during the reaction.*

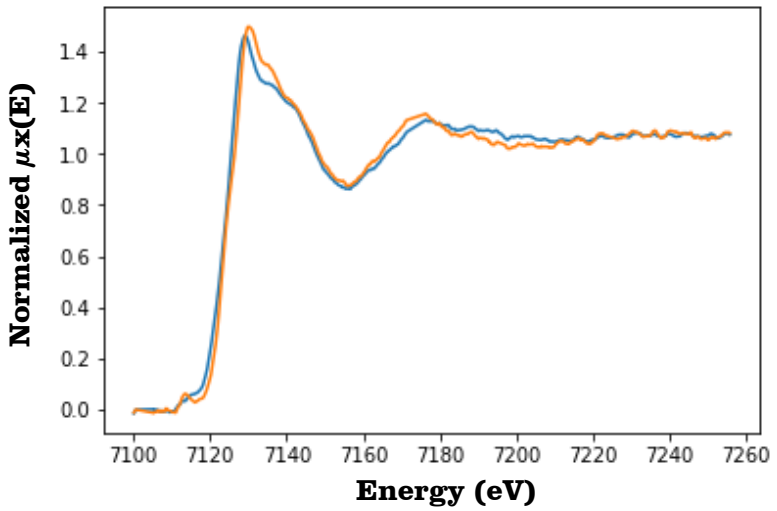


Figure 6.19. *Spectra of the species present in solution during the reaction.*

In **Figure 6.18** the recovered concentration profiles are reported. In **Figure 6.19** the recovered spectra are reported. The results shown in **Figures 6.18-6.19** were obtained by the optimization of the objective function using the method of Nelder-Mead. The optimization of the objective function was implemented as its minimization performed by the Sci.py package of python.

6.2.2. Dataset 3.

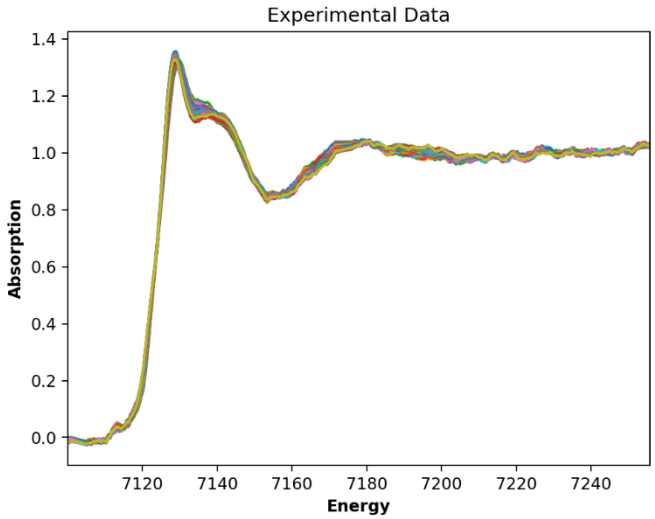


Figure 6.20. *XAS spectra for the reaction $PhCH_2OH + Fe^{IV}N4Py(O)$.*

In **Figure 6.20** the XAS dataset composed by 99 spectra.

Since the strategy for obtaining of the desired results is similar and the program code for this strategy is identical for dataset 2 and dataset 3, the descriptions of all reported below graphics of this paragraph are similar to their descriptions in the paragraph 6.2.1.

The normalized experimental spectra using the formula (6.4) are shown in **Figure 6.22** and their related normalized experimental concentration profiles are shown in **Figure 6.21**. Before the normalization the experimental spectra were interpolated with the step parameter equal to 0.005.

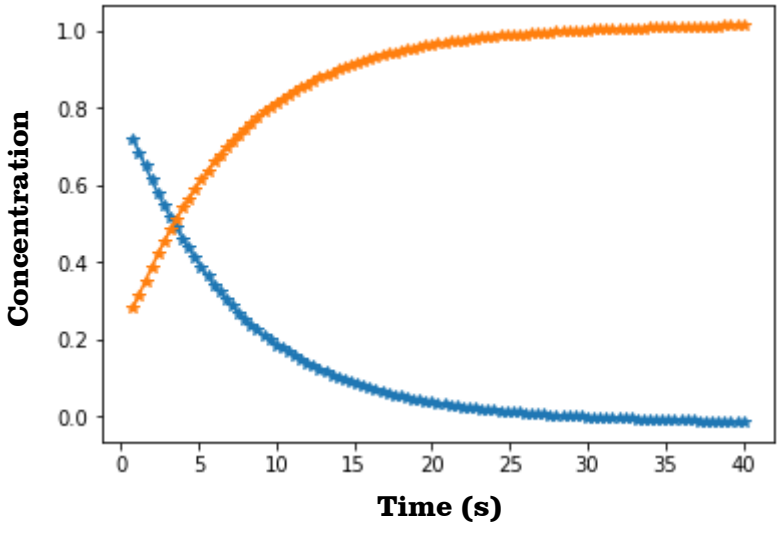


Figure 6.21. *Normalized concentration profiles.*

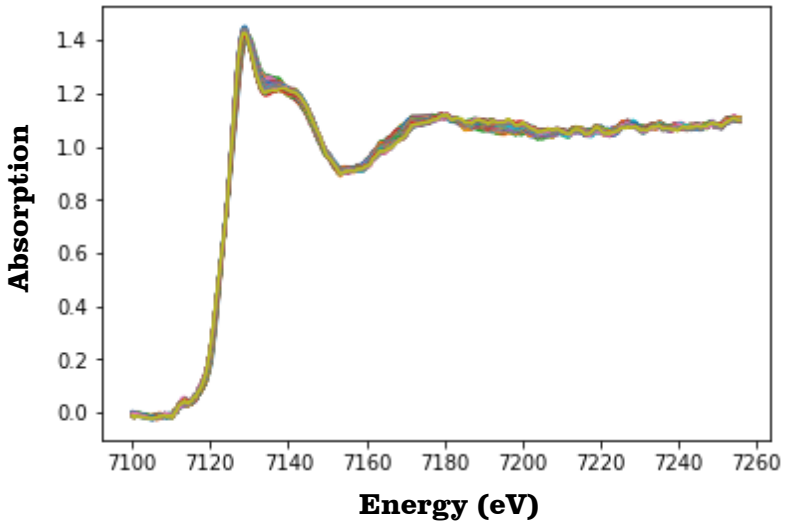


Figure 6.22. *Normalized spectra.*

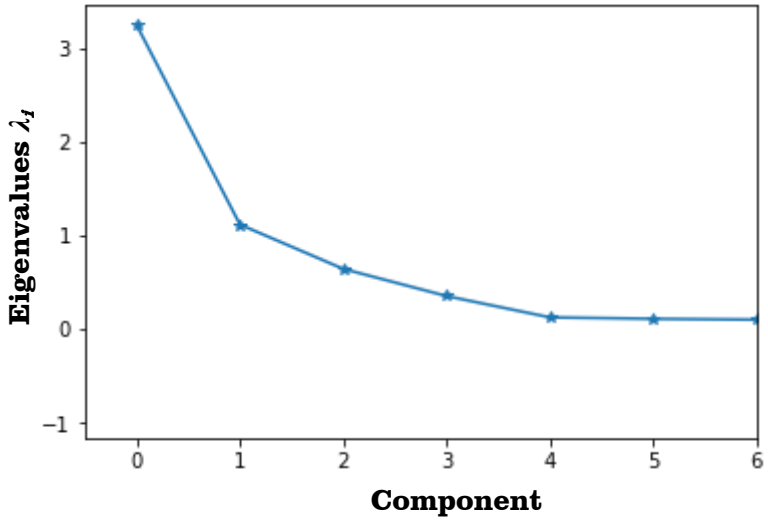


Figure 6.23. *Number of principal components.*

In **Figure 6.23** the number of principal components is reported which is equal to 2.

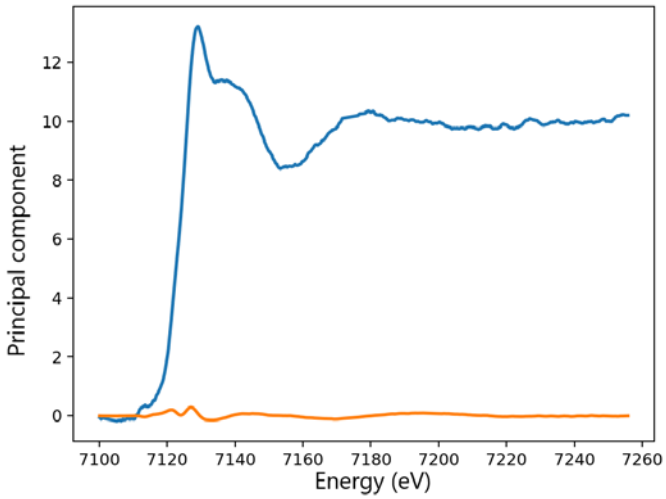


Figure 6.24. *Example of the spectra to be multipled with the elements of the transformation matrix T.*

As it is possible to observe, the pure spectral profiles in **Figure 6.24** do not have any chemical/physical meaning.

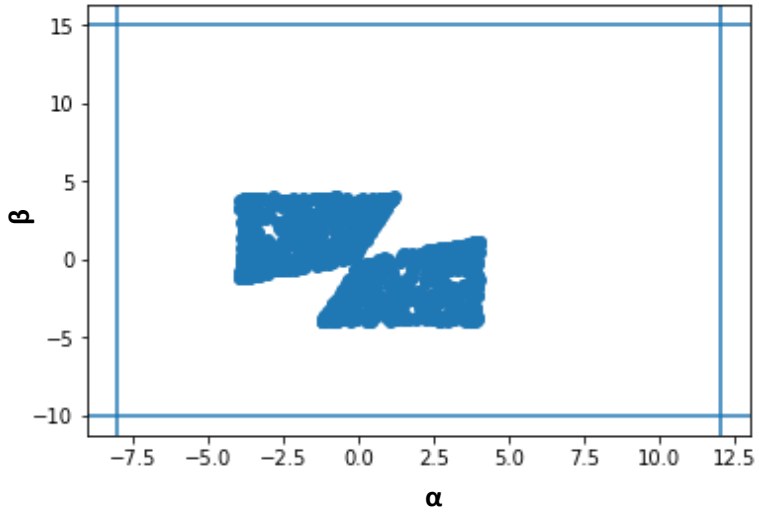


Figure 6.25. *The AFS obtained by minimizing function (6.5).*

The AFS in **Figure 6.25** is obtained using 1000 initialization sets of points chosen randomly between -5 and 5. The minimization procedure has been carried out using the Nelder-Mead algorithm.

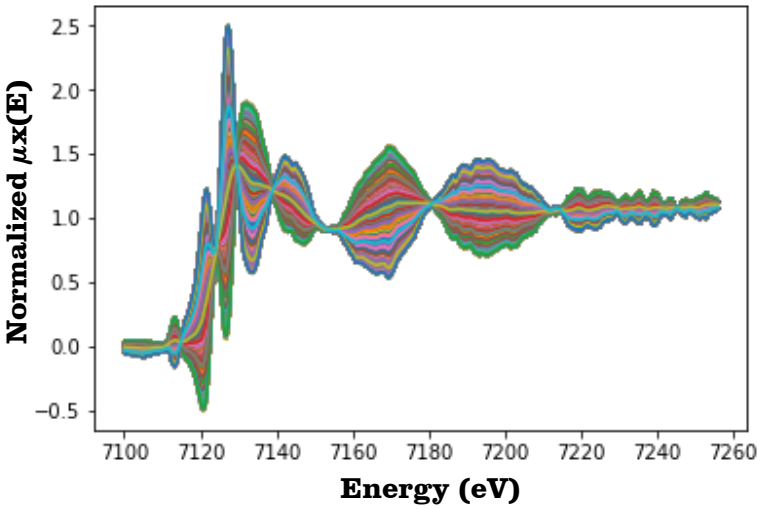


Figure 6.26. *Variation of the spectra as a function of estimated parameters.*

Spectra of estimated parameters are represented in **Figure 6.26**.

In **Figure 6.27** the maximum and the minimum meaningful boundary of concentrations profiles are reported.

In **Figure 6.28** the maximum and the minimum meaningful boundary of spectral profiles are reported.

The set reported in **Figure 6.26** was used as initial guess spectra.

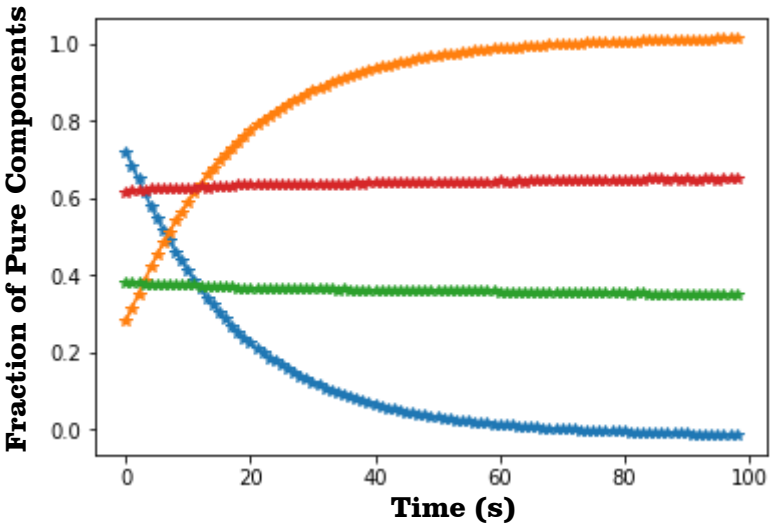


Figure 6.27. *Concentration boundaries obtained minimizing the objective function (6.9) with constraints by method SLSQP.*

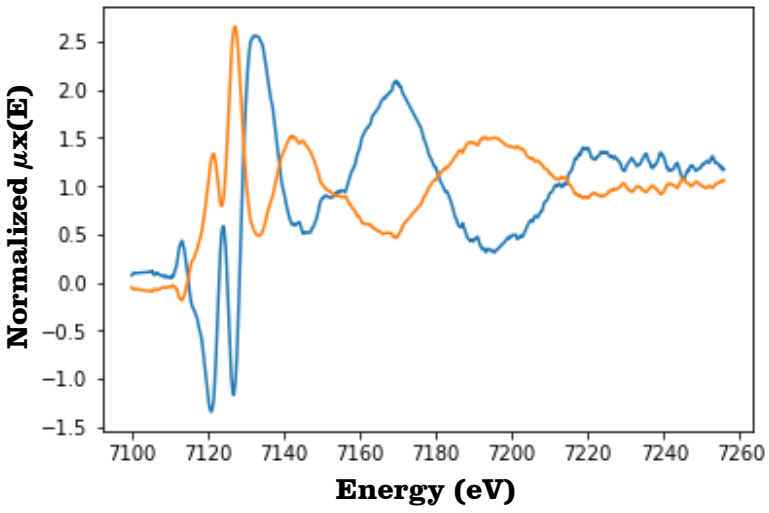


Figure 6.28. *Spectral boundaries obtained minimizing the objective function (6.9) with constraints by method SLSQP.*

With this kind of constraints, a big concentration profiles variation is detectable in **Figure 6.27** as compared to the recovered concentration profiles reported in **Figure 6.29**.

A big spectral variation is detectable in **Figure 6.28** as compared to the recovered spectra reported in **Figure 6.30**.

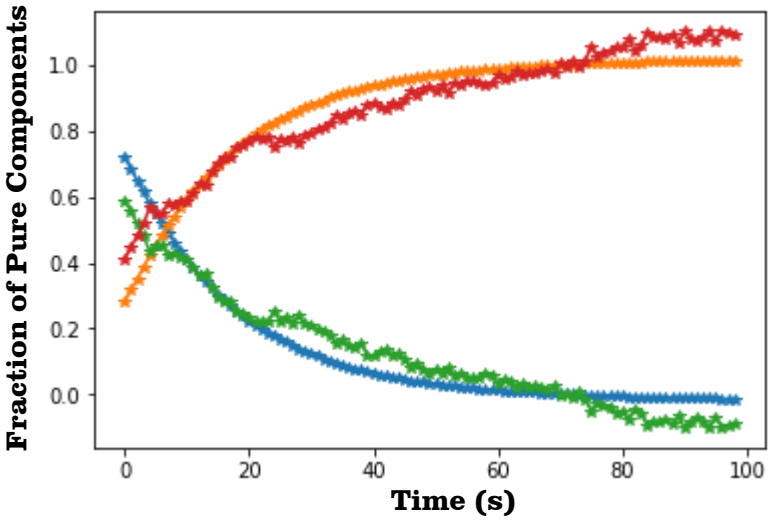


Figure 6.29. *Concentration profiles of the species present in solution during the reaction.*

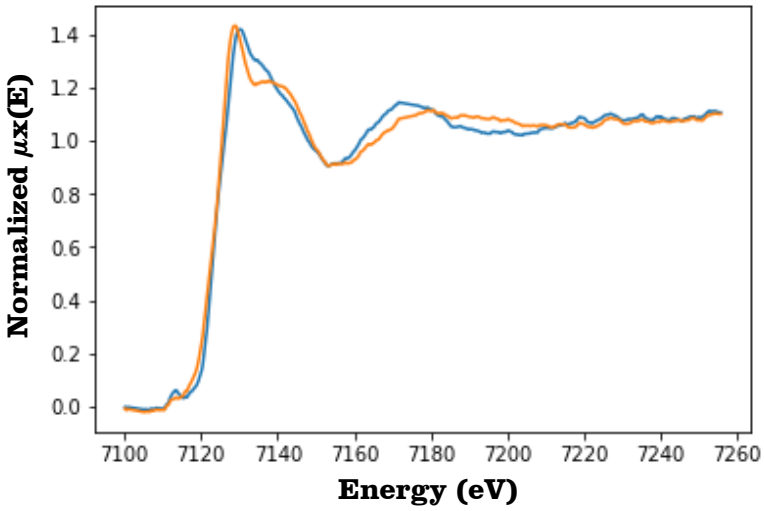


Figure 6.30. *Spectra of the species present in solution during the reaction.*

In **Figure 6.29** the recovered concentration profiles are reported. In **Figure 6.30** the recovered spectra are reported.

CONCLUSIONS

In this thesis we developed and applied a deconvolution procedure to the XAS spectra of different species undergoing a chemical reaction.

In the thesis the following tasks were realized:

1) Having three species, coexisting in solution, and the time evolution of the total absorption spectra:

✓ three concentration variations with time were extracted (C_1 as it decreases, C_2 as it increases and then redecreases and C_3 as it increases over time);

✓ three forms of the XANES spectra were determined, that multiplied by C_1, C_2 and C_3 respectively at each instant of time, and summed up gave the original mixture dataset.

The same goal was realized when two species were present in solution.

2) The concentrations (as a function of the time), that is the matrix $C_{3\times time}$, were determined.

3) The absorption spectra (as a function of the energy), that is the matrix $A_{3\times energy}$, were determined.

4) Pure spectral and concentration profiles were obtained for different XANES and UV-Vis datasets with different number of components using the computer softwares FAC-PACK and PyFitIt.

5) A comparative analysis of the effectiveness of the obtained results in FAC-PACK and in PyFitIt was made.

6) The theory for application of the NMF method, constrained nonlinear method, PCA method was developed and implemented. These methods were tested for different datasets.

7) The regions populated by t_{jk} values of the transformation matrix T capable to provide the non-negative specta and concentration profiles, which satisfy mass balance sheet were identified.

8) A set of “pure” spectra having a chemical-physical meaning was isolated for these t_{jk} values, which satisfy the previous point 7).

9) The maxima and the minima meaningful spectral and concentration boundaries under constraints were identified.

10) A computer program was written in programming language Python to identify pure spectral and concentration profiles and their related range of confidence for different XANES datasets with different number of components.

ABSTRACT

The classical problem of the determination of the spectral components from a data set is very often a badly conditioned problem without univocal solutions. Very often one deals with experimental data which are a superposition, or mixture, of the pure spectra of the individual components and their associated proportions. When dealing with evolving systems such as chemical reactions several compounds are coexisting and if one collects UV-Vis or any other kind of absorption spectra they will be the overimposition of the singles spectra associated with the chemical species present in the reaction bath with an intensity that depends on their concentration profiles. In this case it is very important to be able to extract each single component that is the absorption spectrum associated with each species, and the concentration evolution with time. More than often, this decomposition is aimed at situations for which little a priori information is available. Several theoretical approaches have been developed to achieve this goal and the multivariate analysis is one of the most promising approach. The multivariate curve resolution (MCR) is the generic denomination of a family of methods meant to solve the mixture analysis problem that is able to provide a chemically meaningful additive bilinear model of pure contributions from the sole information of an original data matrix including a mixed measurement. MCR methods are key-tools in order to extract the pure component information (pure component spectra and the concentration profiles) from the chemical mixture (spectroscopic) data. The problem is to compute: 1) the number of independent components *s* and 2) the pure component factors C (concentration profiles) and A (spectra).

An actuality of the problem of the determination of the spectral components from a chemical-physical data is driven by its broad range of applications in situations where a reasonable approximation of the bilinear model, or any other fundamental basic equation that has the same mathematical structure, holds. Also, the bilinear model can be extended for the analysis of multiple data sets that are meant to connect different experiments together.

The main goal of the thesis is to develop and apply an innovative MCR approach for the multivariate analysis of chemical systems. In particular, this method has been developed to have an *ad hoc* tool to be used for chemical reactions for which a combined UV-Vis and X-ray absorption (XAS) investigation is carried out.

The final goal is to extract the UV-Vis and XAS spectra associated with all the intermediates that are formed during bimolecular chemical reactions occurring in the ms time range.

In particular, in the first part of the thesis I have implemented the non negative matrix factorization (NMF) method to a newly written code, PyFitIt, that has been specifically developed for the multivariate analysis of XAS spectra. This method has been then applied to different sets of experimental data collected on different bimolecular chemical reactions in solution for which UV-Vis and XAS spectra were collected simultaneously.

By using this new approach it was possible to extract, from the initial experimental data sets, both the XAS spectra and the concentration profiles of the intermediate species that were formed during the reaction. From the analysis of the X-ray absorption near edge structure (XANES) spectra it will be possible to determine the oxidation state of the absorption element, the nature of the short-lived intermediates that are formed during the reaction, and their three-dimensional structure with a picometric accuracy. To test the reliability of the entire procedure ten different systems have been investigated and the principal components and concentration profiles have been successfully extracted for all the investigated systems. The new procedure and code that has been developed in this thesis will be a valid tool that can be applied to different XAS data sets in the future.

BIBLIOGRAPHY

1. H. Abdollahi, M. Maeder, and R. Tauler, Calculation and meaning of feasible band boundaries in multivariate curve resolution of a two-component system, *Analytical Chemistry*, 2009, **81.** 2115–2122.

2. Ahmed S. A. Alani, Principal component analysis in statistics, Master thesis, 2014.

3. E. M. C. Alayon, M. Nachtegaal, A. Bodi, J.A. van Bokhoven, Reaction conditions of methane-to-methanol conversion affect the structure of active copper sites, *ACS Catalysis*, 2014, **4**. 16-22.

4. S. Beyramysoltan, H. Abdollahi, and R. Rajk´o, Newer developments on self-modeling curve resolution implementing equality and unimodality constraints, *Anal. Chim. Acta*, 2014, **827.** 1–14.

5. S. Beyramysoltan, R. Rajk´o, and H. Abdollahi, Investigation of the equality constraint effect on the reduction of the rotational ambiguity in three-component system using a novel grid search method, *Anal. Chim. Acta*, 2013, **791.** 25–35.

6. A. Bianconi, Surface X-ray absorption spectroscopy: surface EXAFS and surface XANES, *Appl. Surf. Sci*., 1980, **6.** 392-418.

7. X-Ray absorption and X-Ray emission spectroscopy, ed. J. A. Van Bokhoven and C. Lamberti, Wiley, New Jersey, 2016, pp. 25-40.

8. O. S. Borgen, N. Davidsen, Z. Mingyang, and O. Oyen, The multivariate N-component resolution problem with minimum assumptions, *Microchimica Acta*, 1986, **89.** 63–73.

9. O. S. Borgen and B.R. Kowalski, An extension of the multivariate component-resolution method to three components, *Analytica Chimica Acta*, 1985, **174.** 1–26.

10. R. G. Brereton, Chemometrics: data analysis for the laboratory and chemical plant, John Wiley & Sons, 2003.

11. G. Capocasa, F. Sessa, F. Tavani, M. Monte, G. Olivo, S. Pascarelli, O. Lanzalunga, S. Di Stefano and P. D'Angelo, *J. Am. Chem. Soc*., 2019, **141**, 2299-2304.

12. W. Chew, E. Widjaja, and M. Garland, Band-target entropy minimization (BTEM): an advanced method for recovering unknown pure component spectra.

Application to the FT-IR spectra of unstable organometallic mixtures, *Organometallics*, 2002, **21.** 1982–1990.

13. M. S. Chen and M. C. White, *Science*, 2007, **318**. 783-787.

14. L. Eniksson, J. Byme, J. Trygg and E. Johansson, Multi- and megavariate data analysis basic principles and applications part1, New York: Umetrics, Inc., 2006.

15. P.J. Gemperline and E. Cash. Advantages of soft versus hard constraints in self-modeling curve resolution problems, alternating least squares with penalty functions. *Anal. Chem.*, 2003, **75.** 4236–4243.

16. A. Golshan, H. Abdollahi, and M. Maeder, Resolution of rotational ambiguity for three-component systems, *Anal. Chem*., 2011, **83.** 836–841.

17. A. Golshan, M. Maeder, and H. Abdollahi, Determination and visualization of rotational ambiguity in four-component systems, *Anal. Chim. Acta*, 2013, **796.** 20–26.

18. G.H. Golub and C.F. Van Loan, Matrix computations, Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

19. A. A. Guda, S. A. Guda, K. A. Lomachenko, M. A. Soldatov, I. A. Pankin, A. V. Soldatov, L. Braglia, A. L. Bugaev, A. Martini, M. Signorile, E. Groppo, A. Piovano, E. Borfecchia, C. Lamberti, Quantitative structural determination of active sites from in situ and operando XANES spectra: from standard ab initio simulations to chemometric and machine learning approaches, *Catal. Today*, (2018).

20. H. Haario and V.M. Taavitsainen, Combining soft and hard modelling in chemical kinetics, *Chemometr. Intell. Lab.*, 1998, **44.** 77–98.

21. K. Hayakawa, K. Hatada, S. Della Longa, P. D'Angelo, M. Benfatto, Progresses in the MXAN fitting procedure, in: B. Hedman, P. Painetta (Eds.) X-Ray absorption fine structure-Xafs13, *Amer Inst Physics*, Melville, 2007, **882**. 111-113.

22. B. Hemmateenejad, Z. Shojaeifard, M. Shamsipur, K. Neymeyr, M. Sawall, and A. Mohajeri, Solute-induced perturbation of methanol-water association, *RSC Adv.*, 2015, **5.** 71102–71108.

23. Grant S. Henderson, Frank M. F. de Groot, Benjamin J. A. Moulton**,** X-ray absorption near-edge structure (XANES) spectroscopy, *Reviews in Mineralogy & Geochemistry*, 2014, **78.** 75-138.

24. R. C. Henry, Duality in multivariate receptor models, *Chemom. Intell. Lab. Syst.*, 2005, **77.** 59–63.

25. F. James, M. Roos, MINUIT: a system for function minimization and analysis of the parameter errors and corrections, *Comput. Phys. Commun*., 1975, **10**. 343-367.

26. J. Jaumot, R. Gargallo, A. de Juan, and R. Tauler, A graphical user-friendly interface for MCR-ALS: a new tool for multivariate curve resolution in MATLAB, *Chemom. Intell. Lab. Syst.*, 2005, **76**. 101–110.

27. J. Jaumot, P. J. Gemperline, and A. Stang, Non-negativity constraints for elimination of multiple solutions in fitting of multivariate kinetic models to spectroscopic data, *J. Chemometrics*, 2005, **19.** 97–106.

28. J. Jaumot and R. Tauler, MCR-BANDS: A user friendly MATLAB program for the evaluation of rotation ambiguities in multivariate curve resolution, *Chemom. Intell. Lab. Syst.*, 2010, **103**. 96–107.

29. I. Jolliffe, Principal component analysis, New York: Springer, 2002.

30. Y. Joly, S. Grenier, Theory of X-Ray absorption near edge structure, in: J.A. van Bokhoven, C. Lamberti (Eds.) X-Ray absorption and X-Ray emission spectroscopy: theory and applications, John Wiley & Sons, Chichester (UK), 2016, pp. 73-97.

31. A. de Juan, M. Maeder, M. Mart´ınez, and R. Tauler, Combining hard and soft-modelling to solve kinetic problems, *Chemometr. Intell. Lab.*, 2000, **54**. 123–141.

32. A. Jürß, M. Sawall, and K. Neymeyr, On generalized Borgen plots. I: From convex to affine combinations and applications to spectral data, *J. Chemometrics*, 2015, **29**. 420–433.

33. H. F. Kaiser, The varimax criterion for analytic rotation, *Psychometrika,* 1958, **23**. 187–200.

34. S. Kawata, H. Komeda, K. Sasaki, and S. Minami, Advanced algorithm for determining component spectra based on principal component analysis, *Applied Spectroscopy*, 1985, **39**. 610-614.

35. H. Kim, H. Park, Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method, *SIAM J. Matrix. Anal. Appl.*, 2008, **30.** 713–730.

36. D. C. Lay, Linear Algebra and its applications, New York: Pearson Education Inc, 2012.

37. W.H. Lawton, E.A. Sylvestre, Self modelling curve resolution, *Technometrics*, 1971, **13.** 617–633.

38. C. Li, Y. Diao, H. Ma, Y. Li, A statistical PCA method for face recognition, *Intelligent Information Technology Application*, 2008. 376-380.

39. M. Maeder, Evolving factor analysis for the resolution of overlapping chromatographic peaks, *Analytical Chemistry*, 1987, **59**. 527–530.

40. M. Maeder, Y.M. Neuhold, *Practical data analysis in chemistry*, Elsevier, Amsterdam, 2007.

41. M. Maeder, A. D. Zuberbuehler, The resolution of overlapping chromatographic peaks by evolving factor analysis, *Analytica Chimica Acta*, 1986, **181**. 287–291.

42. E. R. Malinowski, Window factor analysis: theoretical derivation and application to flow injection analysis data, *J. Chemometrics*, 1992, **6**. 29–40.

43. E. Malinowski, Factor analysis in chemistry, Wiley, New York, 2002.

44. A. Manceau, M. Marcus, T. Lenoir, Estimating the number of pure chemical components in a mixture by X-ray absorption spectroscopy, *J. Synchrot. Radiat.*, 2014, **21**. 1140-1147.

45. R. Manne, On the resolution problem in hyphenated chromatography, *Chemometrics and Intelligent Laboratory Systems*, 1995, **27**. 89–94.

46. A. Martini, E. Alladio, E. Borfecchia, Determining Cu-speciation in the Cu-CHA zeolite catalyst: the potential of multivariate curve resolution analysis of in situ XAS data, *Top. Catal*., 2018, **61**. 1396-1407.

47. A. Martini, S. A. Guda, A. A. Guda, G. Smolentsev, A. Algasov, O. Usoltsev, M. A. Soldatov, A. Bugaev, Y. Rusalev, C. Lamberti and A. V. Soldatov, PyFitit: The software for quantitative analysis of XANES spectra using machine-learning algorithms, In Press.

48. C. Mason, M. Maeder, A. Whitson, Resolving factor analysis, *Anal. Chem.*, 2001, **73**. 1587–1594.

49. H. Minc, Nonnegative matrices, JohnWiley & Sons, New York, 1988.

50. J. R. Movellan, Introduction to probability theory and statistics, Javier R. Movellan, 2008.

51. C. R. Natoli, Distance dependence of continuum and bound state of excitonic resonances in X-Ray absorption near edge structure (XANES), in: K.O.

Hodgson, B. Hedman, J.E. Penner-Hahn (Eds.) EXAFS and near edge structure III. Springer Proc. Phys., Vol 2. , Springer, Berlin, 1984, pp. 38-42.

52. K. Neymeyr, M. Sawall, D. Hess, Pure component spectral recovery and constrained matrix factorizations: concepts and applications, *J. Chemom.*, 2010, **24**. 67–74.

53. K. Neymeyr, M. Sawall, On the set of solutions of the nonnegative matrix factorization problem, *SIAM J. Matrix Anal. Appl.*, 2018, **39.** 1049-1069.

54. Pham Huu Anh Ngoc, Byung Soo Lee, Nguyen Khoa Son, Perron Frobenius theorem for positive polynomial matrice, *Vietnam Journal of Mathematics,* 2004, **32**. 475–481.

55. G. Olivo, A. Barbieri, V. Dantignana, F. Sessa, V. Migliorati, M. Monte, S. Pascarelli, T. Narayanan, O. Lanzalunga, S. Di Stefano and P. D'Angelo, *The Journal of Physical Chemistry Letters*, 2017, **8**, 2958-2963.

56. G. Olivo, L. Cussò, M. Costas, *Chem. - Asian J.*, 2016, **11**, 792-793.

57. T. Owen, Fundamentals of UV-visible spectroscopy: A Primer, Agilent Technologies, 2000.

58. D. K. Pappas, E. Borfecchia, M. Dyballa, K. A. Lomachenko, A. Martini, G. Berlier, B. Arstad, C. Lamberti, S. Bordiga, U. Olsbye, S. Svelle, P. Beato, Understanding and optimizing the performance of Cu-FER for the direct CH4 to CH3OH conversion, *ChemCatChem*, 2019, **11**. 621-627.

59. D. K. Pappas, E. Borfecchia, M. Dyballa, I. A. Pankin, K. A. Lomachenko, A. Martini, M. Signorile, S. Teketel, B. Arstad, G. Berlier, C. Lamberti, S. Bordiga, U. Olsbye, K. P. Lillerud, S. Svelle, P. Beato, Methane to methanol: structure activity relationships for Cu-CHA, *J. Am. Chem. Soc.*, 2017, **139**. 14961-14975.

60. D. K. Pappas, E. Borfecchia, K.A. Lomachenko, A. Lazzarini, E. S. Gutterød, M. Dyballa, A. Martini, G. Berlier, S. Bordiga, C. Lamberti, B. Arstad, U. Olsbye, P. Beato, S. Svelle, Cu-exchanged ferrierite zeolite for the direct CH4 to CH3OH conversion: insights on Cu speciation from X-Ray absorption spectroscopy, *Top. Catal.*, 2019, **62**. 712-723.

61. Park J., Morimoto Y., Lee Y.-M., Nam W., Fukuzumi S., Metal ion-coupled electron transfer of a nonheme oxoiron(IV) complex: remarkable enhancement of electron-transfer rates by Sc^{3+}, *J. Am. Chem. Soc.*, 2011, **133**, 403-405.

62. Park J., Morimoto Y., Lee Y.-M., Nam W., Fukuzumi S., Proton-promoted oxygen atom transfer vs proton-coupled electron transfer of a non-heme iron(IV)-oxo complex, *J. Am. Chem. Soc.*, 2012, **134**. 3903-3911.

63. Park J., Morimoto Y., Lee Y.-M., Nam W., Fukuzumi S., Unified view of oxidative C-H bond cleavage and sulfoxidation by a non heme iron(IV)-oxo complex via Lewis acid-promoted electron transfer, *Inorg. Chem.*, 2014, **53**. 3618-3628.

64. D. Poole, Linear Algebra: a modern introduction, 3rd edition, Bosten USA: Brooks/Cole, Cengage Learning, 2011.

65. P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min and W. Worek, Overview of the face recognition grand challenge, *Computer vision and pattern recognition*, 2005. 947-954.

66. N. Rahimdoust, M. Sawall, K. Neymeyr, and H. Abdollahi, Investigating the effect of constraints on the accuracy of the area of feasible solutions in the presence of noise with an improved cost function of the polygon inflation algorithm; soft versus hard constrained SMCR, Technical Report, Universität Rostock and IASBS Zanjan, 2015.

67. R. Rajk´o, Additional knowledge for determining and interpreting feasible band boundaries in self-modeling/multivariate curve resolution of two-component systems, *Anal. Chim. Acta*, 2010, **661**. 129–132.

68. R. Rajk´o, Natural duality in minimal constrained self modeling curve resolution, *J. Chemom.*, 2006, **20**. 164–169.

69. R. Rajkó, K. István, Analytical solution for determining feasible regions of self-modeling curve resolution (SMCR) method based on computational geometry, *J. Chemometrics*, 2005, **19**. 448–463.

70. A. C. Rencher, Methods of multivariate analysis, Brigham Young University: A John Wiley & Sons, Inc. Publication, 2002.

71. C. Ruckebusch, Resolving spectral mixtures with applications from ultrafast time-resolved spectroscopy to super-resolution imaging, Elsevier, 2016.

72. K. Sasaki, S. Kawata, S. Minami, Constrained nonlinear method for estimating component spectra from multicomponent mixtures, *Applied Optics*, 1983, **22**. 3599-3603.

73. K. Sasaki, S. Kawata, S. Minami, Estimation of component spectral curves from unknown mixture spectra, *Applied Optics*, 1984, **23**. 1955-1959.

74. M. Sawall, A. Börner, C. Kubis, D. Selent, R. Ludwig, K. Neymeyr, Model-free multivariate curve resolution combined with model-based kinetics: Algorithm and applications, *J. Chemom.*, 2012, **26**. 538–548.

75. M. Sawall, C. Fischer, D. Heller, K. Neymeyr, Reduction of the rotational ambiguity of curve resolution technqiues under partial knowledge of the factors. Complementarity and coupling theorems, *J. Chemometrics*, 2012, **26**. 526–537.

76. M. Sawall, A. Jürß, H. Schröder, K. Neymeyr, On the analysis and computation of the area of feasible solutions for two-, three- and four-component systems, Book contribution in volume 30 of data handling in science and technology, "Resolving spectral mixtures" edited by Cyril Ruckebusch, Elsevier 2016.

77. M. Sawall, C. Kubis, E. Barsch, D. Selent, A. Börner, K. Neymeyr, Peak group analysis for the extraction of pure component spectra, *Journal of the Iranian Chemical Society*, 2015, **13**. 1–15.

78. M. Sawall, C. Kubis, D. Selent, A. Börner, K. Neymeyr, A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. I: Concepts and applications, *J. Chemom.*, 2013, **27**. 106–116.

79. M. Sawall, K. Neymeyr, A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. II: Theoretical foundation, inverse polygon inflation, and FAC-PACK implementation, *J. Chemom.*, 2014, **28**. 633–644.

80. M. Sawall, K. Neymeyr, On the area of feasible solutions and its reduction by the complementarity theorem, *Anal. Chim. Acta*, 2014, **828**. 17–26.

81. M. Sawall, K. Neymeyr, A. Jürß, Users' guide to FAC-PACK, 2014. http://www.math.uni-rostock.de/facpack/

82. M. Sawall, N. Rahimdoust, C. Kubis, H. Schröder, D. Selent, D. Hess, H. Abdollahi, R. Franke, A. Börner, K. Neymeyr, Soft constraints for reducing the intrinsic rotational ambiguity of the area of feasible solution, Technical Report, *Chemom. Intell. Lab. Syst*., 2015, **149**. 140-150.

83. G. Smolentsev, A. V. Soldatov, FitIt: New software to extract structural information on the basis of XANES fitting, *Comput. Mater. Sci.*, 2007, **39**. 569-574.

84. G. Smolentsev, A. Soldatov, Quantitative local structure refinement from XANES: multi-dimensional interpolation approach, *J. Synchrot. Radiat.*, 2006, **13**. 19-29.

85. M. A. Soldatov, A. Bugaev, Y. Rusalev, C. Lamberti and A. V. Soldatov, *Computer Physics Communications*, In Press.

86. V. L. Sushkevich, D. Palagin, M. Ranocchiari, J.A. van Bokhoven, Selective anaerobic oxidation of methane enables direct synthesis of methanol, *Science*, 2017, **356**. 523-527.

87. F. Tavani, Studio di reazioni chimiche attraverso tecniche di spettroscopia XAS, UV-Vis e di analisi statistica multivariata, Tesi di laurea magistrale, 2019.

88. R. Tauler, Calculation of maximum and minimum band boundaries of feasible solutions for species profiles obtained by multivariate curve resolution, *Journal of Chemometrics*, 2001, **15**. 627-646.

89. J. Timoshenko, A. Anspoks, A. Cintins, A. Kuzmin, J. Purans, A. I. Frenkel, Neural network approach for characterizing structural transformations by X-Ray absorption fine structure spectroscopy, *Physical Review Letters*, 2018, **120**. 1-6.

90. J. Timoshenko, D. Y. Lu, Y. W. Lin, A. I. Frenkel, Supervised machine-learning-based determination of three-dimensional structure of metallic nanoparticles, *J. Phys. Chem. Lett.*, 2017, **8.** 5091-5098.

91. J. Timoshenko, C. J. Wrasman, M. Luneau, T. Shirman, M. Cargnello, S. R. Bare, J. Aizenberg, C. M. Friend, A. I. Frenkel, Probing atomic distributions in mono- and bimetallic nanoparticles by supervised machine learning, *Nano Lett.*, 2019, **19**. 520-529.

92. L. L. Thurstone, Multiple-factor analysis, Chicago: University of Chicago press, 1947. pp. 535.

93. M. Vosough, C. Mason, R. Tauler, M. Jalali-Heravi, M. Maeder, On rotational ambiguity in model-free analyses of multivariate data, *J. Chemom.*, 2006, **20**. 302–310.

94. B. M. Weckhuysen, In-situ spectroscopy of catalysts, Amer. Scientific. Pub., 2004.

95. E. Widjaja, C. Li, W. Chew, M. Garland, Band target entropy minimization. A robust algorithm for pure component spectral recovery. Application to complex randomized mixtures of six components, *Anal. Chem.*, 2003, **75**. 4499–4507.

96. L. J. Williams, H. Abdi, Principal Component Analysis, *John Wiley & Sons*, 2010. p. 433.

97. W. Windig, J. Guilment, Interactive self-modeling mixture analysis, *Anal. Chem.*, 1991, **63**. 1425–1432.

98. W. Windig, M.R. Keenan, Angle-constrained alternating least squares, *Appl. Spectrosc.*, 2011, **65**. 349–357.

99. Winkler G., The Perron-Frobenius Theorem, In: Image Analysis, Random Fields and Dynamic Monte Carlo Methods. Applications of Mathematics (Stochastic Modelling and Applied Probability), vol 27. Springer, Berlin, Heidelberg, 1995.

100. C. Zheng, K. Mathew, C. Chen, Y. M. Chen, H. M. Tang, A. Dozier, J. J. Kas, F. D. Vila, J. J. Rehr, L. F. J. Piper, K. A. Persson, S. P. Ong, Automated generation and ensemble-learned matching of X-ray absorption spectra, *npj Comput. Mater.*, 2018, **4**.

101. Visible and ultraviolet spectroscopy. https://www2.chemistry.msu.edu/faculty/reusch/VirtTxtJml/Spectrpy/UV-Vis/spect-rum.htm

102. Блохин М.А. Физика рентгеновских лучей. – М.: Гостехиздат, 1957. – 518 с.

103. Ведринский Р.В. EXAFS-спектроскопия – новый метод структурного анализа // Соросовский Образовательный Журнал. 1996. №5. С. 79-84.

104. Кочубей Д.И., Бабанов Ю.А., Замараев К.И. и др. EXAFS-спектроскопия. Новосибирск: Наука, 1988. 306 с.

105. А. В. Солдатов. От спектроскопии EXAFS к спектроскопии XANES: новые возможности исследования материи. Соросовский образовательный журнал, №12, 1998.

106. Spettroscopia UV – visibile. http://www.chimicavolta.com

107. Visione computazionale: tecniche di ricostruzione tridimensionale. https://scaricarelibri.net/andrea-fusiello.html

APPENDIX A

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
import random
import copy
import collections
from scipy.optimize import basinhopping
import scipy.interpolate
from mpl_toolkits import mplot3d

# Import of the experimental dataset
dataset=np.loadtxt('dataset_Fe.dat')

# Procedure of the normalization
def normalization(energy,data):
    scaled=np.zeros(np.shape(data)[1])
    for i in range(np.shape(data)[1]):
        scaled[i]=np.sqrt((1./((1./(np.max(energy)-
np.min(energy)))*(np.trapz((data[:,i])**2,energy)))))
    for i in range(np.shape(data)[1]):
        data[:,i]=data[:,i]*scaled[i]
    return data

def sigle_norm(energy,spectrum):
    scale=np.sqrt((1./((1./(np.max(energy)-
np.min(energy)))*(np.trapz((spectrum)**2,energy)))))
    return scale*spectrum

# Procedure of the interpolation
def interpolation(energy,data,step):
    e_valor=np.arange(min(energy),max(energy),step)
    dat_valor=[]
```

```
    col=np.shape(data)[1]
    for i in range(col):
        d=np.interp(e_valor,energy,data[:,i])
        dat_valor.append(d)
    data=np.transpose(dat_valor)
    energy=e_valor
    return energy, data

# Definition of the spectral step function
def H_s(x):
    if x<0 or x>2:
        return 1
    elif x>=0 and x<=2:
        return 0

# Definition of the concentration step function
def H_c(x):
    if x<0 or x>1:
        return 1
    elif x>=0 and x<=1:
        return 0

# Import of the references
ref_1_imp=np.loadtxt('Cu_Hydr.dat')
ref_2_imp=np.loadtxt('Cu_I_ammino.dat')
ref_3_imp=np.loadtxt('Cu_II.dat')

ref_p10=np.loadtxt('pappas_1_Hydr.dat')
ref_p20=np.loadtxt('pappas_2_ammino.dat')
ref_p30=np.loadtxt('pappas_3_Cu_II.dat')

ref_oxo_1=np.loadtxt('Cu_I_oxo.dat')
ref_oxo_2=np.loadtxt('Cu_II_oxo.dat')

ref_2Al_I=np.loadtxt('CHA_2AL.dat')
```

```
energia=dataset[:,0] # energia
X_old=dataset[:,1:]
energia,X_int=interpolation(energia,X_old,0.05)
X=normalization(energia,X_int)
ref_1=np.interp(energia,ref_1_imp[:,0],ref_1_imp[:,1])
ref_2=np.interp(energia,ref_2_imp[:,0],ref_2_imp[:,1])
ref_3=np.interp(energia,ref_3_imp[:,0],ref_3_imp[:,1])

ref_1p=np.interp(energia,ref_p10[:,0],ref_p10[:,1])
ref_2p=np.interp(energia,ref_p20[:,0],ref_p20[:,1])
ref_3p=np.interp(energia,ref_p30[:,0],ref_p30[:,1])

ref_2Al=np.interp(energia,ref_2Al_I[:,0],ref_2Al_I[:,1])

ref_Cu_I_oxo=np.interp(energia,ref_oxo_1[:,0],ref_oxo_1[:,1])
ref_Cu_II_oxo=np.interp(energia,ref_oxo_2[:,0],ref_oxo_2[:,1])
plt.xlabel("Energy (eV)")
plt.ylabel("Norm. XANES")
_=plt.plot(energia,X)
ref_Hydr=sigle_norm(energia,ref_1)
ref_Cu_I=sigle_norm(energia,ref_2)
ref_Cu_II=sigle_norm(energia,ref_3)

ref_H_p=sigle_norm(energia,ref_1p)
ref_Cu_I_p=sigle_norm(energia,ref_2p)
ref_Cu_II_p=sigle_norm(energia,ref_3p)

ref_Cu_OXO_1=sigle_norm(energia,ref_Cu_I_oxo)
ref_Cu_OXO_2=sigle_norm(energia,ref_Cu_II_oxo)

ref_2AL=sigle_norm(energia,ref_2Al)

# Plotting of the references used for the comparison
plt.plot(energia,ref_Cu_OXO_1)
plt.plot(energia,ref_Cu_OXO_2)
```

```
_=plt.plot(energia,X)
np.savetxt('X.dat',X)
_=plt.plot(energia,X)
plt.plot(energia,ref_H_p)
plt.plot(energia,ref_Cu_I_p)
plt.plot(energia,ref_Cu_II_p)
plt.plot(energia,ref_Hydr)
plt.plot(energia,ref_Cu_I)
plt.plot(energia,ref_Cu_II)
_=plt.plot(energia,X)
plt.plot(energia,ref_Hydr)
plt.plot(energia,ref_Cu_I)
plt.plot(energia,ref_Cu_II)
plt.plot(energia,ref_2AL)
plt.plot(energia,ref_Hydr)
plt.plot(energia,ref_Cu_I)
plt.plot(energia,ref_Cu_II)
plt.plot(energia, ref_Cu_II)

# Singular value decomposition
U,S,VT=np.linalg.svd(X, full_matrices=False)
_=plt.plot(S,'-*')
plt.yscale("log")
plt.xlim(-0.50,8)
plt.ylim(10**-3,10**4)
np.savetxt('S.dat',S)

# Principal components number
nPC=3
nPC_n=4
u=U[:,:nPC]
vt = VT[:nPC,:]
s=np.diag(S[:nPC])
u_n=U[:,:nPC_n]
vt_n = VT[:nPC_n,:]
```

```
s_n=np.diag(S[:nPC_n])
us_n=np.dot(u_n,s_n)
us=np.dot(u,s)
_=plt.plot(us)
a=-1*np.sqrt((1./((1./(np.max(energia)-
np.min(energia)))*(np.trapz((us[:,0])**2,energia)))))
print(a)
plt.plot(a*us[:,0])
Z = np.full(3, a)
parametri=[3.4,-1.05,-0.7,0.45,1.5,-0.3]
def spectra(params):
    params=np.reshape(params,(2,3))
    t=np.vstack((Z,params))
    return np.dot(us,t)
def concentrations(params):
    params=np.reshape(params,(2,3))
    t=np.vstack((Z,params))
    return np.dot(np.linalg.pinv(t),vt)

plt.plot(concentrations(parametri).T)
np.savetxt('s_X.dat',spectra(parametri))
L,M=np.shape(X)
N=nPC
print(L,M)
L,M=np.shape(X)

# Definition of the penalty function
def P(params):
    term_S=np.zeros((L,N))
    term_C=np.zeros((M,N))
    sp=spectra(params)
    cp=concentrations(params).T
    for j in range(N):
        for i in range(L):
            term_S[i,j]=H_s(sp[i,j])*(sp[i,j])**2
```

```
    for i in range(N):
        for j in range(M):
            term_C[j,i]=H_c(cp[j,i])*(cp[j,i])**2
    penalty=(((sum(sum(term_S)))+(sum(sum(term_C)))))
    return penalty
plt.plot(ref_3p)

# Definition of the norm L^2
def L2_1(params):
    spettri_1=spectra(params)
    return np.sqrt((1/(np.max(energia)-np.min(energia)))*(np.trapz((spettri_1[:,0]-
ref_2p)**2,energia)))
def L2_2(params):
    spettri_2=spectra(params)
    return np.sqrt((1/(np.max(energia)-np.min(energia)))*(np.trapz((spettri_2[:,1]-
ref_1p)**2,energia)))
def L2_3(params):
    spettri_3=spectra(params)
    return np.sqrt((1/(np.max(energia)-np.min(energia)))*(np.trapz((spettri_3[:,2]-
ref_3p)**2,energia)))
def L2(spectrum,ref):
    return np.sqrt((1/(np.max(energia)-np.min(energia)))*(np.trapz((spectrum-
ref)**2,energia)))

# Definition of the second derivative
def Second_D(spectra):
    secondS=np.zeros((np.shape(spectra)[0]-2,np.shape(spectra)[1]))
    for i in range(np.shape(spectra)[1]):
        dS=np.diff(spectra[:,i],1)
        dE=np.diff(energia,1)
        S_first=dS/dE
        E_first=0.5*(energia[:-1]+energia[1:])
        dS_first=np.diff(S_first,1)
        dE_first=np.diff(E_first,1)
        S_second=dS_first/dE_first
```

```
        E_second=0.5*(E_first[:-1]+E_first[1:])
        scale_derivative=abs(np.sqrt((1./((1./(np.max(E_second)-
np.min(E_second)))*(np.trapz((S_second)**2,E_second))))))
        secondS[:,i]=abs(S_second)*scale_derivative
    for n in range(np.shape(spectra)[1]):
        for m in range(np.shape(spectra)[0]-2):
            if secondS[m,n]==0:
                secondS[m,n]=10**-9
    return secondS
energy_derivative=copy.copy(energia)
e_Min= 8980 ;
e_Max= 8987
e_Min_near=min(energy_derivative, key=lambda x:abs(x-e_Min));
e_Max_near=min(energy_derivative,
key=lambda x:abs(x-e_Max))
p_Min=list(energy_derivative).index(e_Min_near)
p_Max=list(energy_derivative).index(e_Max_near)

# Definition of the fourth derivative
def Fourth_D(spectra):
    second_derivative_spectra=Second_D(spectra)
    fourthS=np.zeros((np.shape(spectra)[0]-4,np.shape(spectra)[1]))
    for i in range(np.shape(spectra)[1]):
        E_first=0.5*(energia[:-1]+energia[1:])
        E_second=0.5*(E_first[:-1]+E_first[1:])
        dS2=np.diff(second_derivative_spectra[:,i],1)
        dE2=np.diff(E_second,1)
        S_third=dS2/dE2
        E_third=0.5*(E_second[:-1]+E_second[1:])
        dS_third=np.diff(S_third,1)
        dE_third=np.diff(E_third,1)
        S_fourth=dS_third/dE_third
        E_fourth=0.5*(E_third[:-1]+E_third[1:])
        scale_derivative_4=abs(np.sqrt((1./((1./(np.max(E_fourth)-
np.min(E_fourth)))*(np.trapz((S_fourth)**2,E_fourth))))))
```

```
        fourthS[:,i]=abs(S_fourth)*scale_derivative_4
    for n in range(np.shape(spectra)[1]):
        for m in range(np.shape(spectra)[0]-4):
            if fourthS[m,n]==0:
                fourthS[m,n]=10**-9
    return fourthS

def Second_D_Reduced(spectra):
    secondS=np.zeros((np.shape(spectra[p_Min:p_Max,:])[0]-
2,np.shape(spectra[p_Min:p_Max,:])[1]))
    for i in range(np.shape(spectra[p_Min:p_Max,:])[1]):
        dS=np.diff(spectra[p_Min:p_Max,i],1)
        energy_R=energy_derivative[p_Min:p_Max]
        dE=np.diff(energy_R,1)
        S_first=dS/dE
        E_first=0.5*(energy_R[:-1]+energy_R[1:])
        dS_first=np.diff(S_first,1)
        dE_first=np.diff(E_first,1)
        S_second=dS_first/dE_first
        E_second=0.5*(E_first[:-1]+E_first[1:])
        scale_derivative=abs(np.sqrt((1./((1./(np.max(E_second)-
np.min(E_second)))*(np.trapz((S_second)**2,E_second))))))
        secondS[:,i]=abs(S_second)*scale_derivative
    for n in range(np.shape(secondS)[1]):
        for m in range(np.shape(secondS)[0]):
            if secondS[m,n]==0:
                secondS[m,n]=10**-9
    return seconds
energy_derivative=copy.copy(energia)
e_Min= 8979
e_Max= 8983
e_Min_near=min(energy_derivative, key=lambda x:abs(x-e_Min));
e_Max_near=min(energy_derivative,
key=lambda x:abs(x-e_Max))
p_Min=list(energy_derivative).index(e_Min_near)
```

```
p_Max=list(energy_derivative).index(e_Max_near)

def L2_1_red(params):
    energia_red=energia[p_Min:p_Max]
    spettri_1=spectra(params)
    spettri_1_red=spettri_1[p_Min:p_Max,:]
    return np.sqrt((1/(np.max(energia_red)-
np.min(energia_red)))*(np.trapz((spettri_1_red[:,0]-
ref_Hydr[p_Min:p_Max])**2,energia_red)))

def L2_2_red(params):
    energia_red=energia[p_Min:p_Max]
    spettri_2=spectra(params)
    spettri_2_red=spettri_2[p_Min:p_Max,:]
    return np.sqrt((1/(np.max(energia_red)-
np.min(energia_red)))*(np.trapz((spettri_2_red[:,0]-
ref_Cu_II[p_Min:p_Max])**2,energia_red)))
def entropy(params):
    spettri_entropy=spectra(params)
    der_S=Fourth_D(spettri_entropy)
    matrixP=np.zeros(np.shape(der_S))
    for j in range(np.shape(der_S)[1]):
        for i in range(np.shape(der_S)[0]):
                matrixP[i,j]=-1*der_S[i,j]*(np.log(der_S[i,j]))
    entropia=sum(sum(matrixP))
    return entropia

def similarity(params):
    matrix_sim=np.zeros((L,nPC))
    spettri_s=spectra(params)
    average_s=np.sum(X,axis=1)/(np.shape(X)[1])
    for i in range(np.shape(matrix_sim)[1]):
        matrix_sim[:,i]=(spettri_s[:,i]-average_s)**2
    return sum(sum(matrix_sim))/sum(sum(X**2))
```

```
# AFS calculation. Warning! Long calculation!
# Definition of the matrix of the estimated parameters
parametri_stimati=[]
for j in range(1000):
    params_init=np.zeros(6)
    for i in range(6):
        params_init[i]=random.uniform(-5, 5)
    res = scipy.optimize.minimize(P, params_init,method='nelder-
mead',options={'xtol': 1e-6})
    parametri_stimati.append(res.x)
for i in range(1000):
    plt.plot(spectra(parametri_stimati[i]))
np.shape(spectra(parametri_stimati[1]))[0]
parametri_stimati_s=copy.copy(parametri_stimati)
parametri_da_tenere_s=[]
for i in range(1000):
    spettri=spectra(parametri_stimati_s[i])
    s1=0
    for j in range(nPC):
        for l in range(L):
            if spettri[l,j]>=0 and spettri[l,j]<=2:
                s1=s1+1
    if s1==nPC*L:
        parametri_da_tenere_s.append(parametri_stimati_s[i])
for i in range(np.shape(parametri_da_tenere_s)[0]):
    plt.plot(spectra(parametri_da_tenere_s[i]))
parametri_stimati_c=copy.copy(parametri_da_tenere_s)
parametri_da_tenere_c=[]
for i in range(np.shape(parametri_da_tenere_s)[0]):
    concentrazioni=concentrations(parametri_stimati_c[i])
    concentrazioni=np.transpose(concentrazioni)
    c1=0
    for j in range(nPC):
        for l in range(M):
            if concentrazioni[l,j]>=0 and concentrazioni[l,j]<=1:
```

```
            c1=c1+1
    if c1==nPC*M:
        parametri_da_tenere_c.append(parametri_da_tenere_s[i])
np.shape(parametri_da_tenere_c)
np.shape(parametri_da_tenere_s)
for i in range(np.shape(parametri_da_tenere_c)[0]):
    _=plt.plot(energia,spectra(parametri_da_tenere_c[i]))
for i in range(np.shape(parametri_da_tenere_c)[0]):
    plt.plot(concentrations(parametri_da_tenere_c[i]).T)

# Finish long calculation of the AFS
np.savetxt('good_parameters.dat',parametri_da_tenere_c)
parametri_da_tenere_c=np.loadtxt('good_parameters.dat')
spettri_t=np.zeros((np.shape(X)[0],3))
for i in range(len(parametri_da_tenere_c)):
    spettri_s=spectra(parametri_da_tenere_c[i])
    spettri_t=np.hstack((spettri_t,spettri_s))
p0=[3.4,-1.05,-0.7,0.45,1.5,-0.3]
v=1
bounds=[[p0[0]-v,p0[0]+v],[p0[1]-v,p0[1]+v],[p0[2]-v,p0[2]+v],[p0[3]-
v,p0[3]+v],[p0[4]-v,p0[4]+v],[p0[5]-v,p0[5]+v]]
bounds[5]
_=plt.plot(spectra(p0))
matrix_t_list=np.zeros(np.shape(parametri_da_tenere_c))
for j in range(6):
    for i in range(np.shape(parametri_da_tenere_c)[0]):
        matrix_t_list[i,j]=parametri_da_tenere_c[i][j]
matrix_P0=np.zeros((5,2*nPC))
for i in range(2*nPC):
    matrix_P0[:,i]=np.linspace(p0[i]-v,p0[i]+v,5)

# Definition of the elements of the transformation matrix
t21=matrix_P0[:,0]
t22=matrix_P0[:,1]
t23=matrix_P0[:,2]
```

```
t31=matrix_P0[:,3]
t32=matrix_P0[:,4]
t33=matrix_P0[:,5]
p_values=np.vstack(np.meshgrid(t21,t22,t23,t31,t32,t33)).reshape(6,-1).T
plt.scatter(p0[0],p0[5],color='red')
plt.plot(matrix_t_list[:,0],matrix_t_list[:,5],'*')
plt.show()
np.shape(p_values)
plt.scatter(p_values[:,0],p_values[:,1])

# Objective Function
nT=np.shape(parametri_da_tenere_c)[0]
def obiettivo(params):
    return L2_1(params)+L2_2(params)+L2_3(params)+gamma*P(params)
gamma=1
cons = []
for factor in range(len(bounds)):
    lower, upper = bounds[factor]
    l = {'type': 'ineq',
         'fun': lambda x, lb=lower, i=factor: x[i] - lb}
    u = {'type': 'ineq',
         'fun': lambda x, ub=upper, i=factor: ub - x[i]}
    cons.append(l)
    cons.append(u)
cons
parameters_to_save=[]
params=p0
nT=np.shape(p_values)[0]
for i in range(1):
    gamma=1
    print('ciclo',i)
    params=p_values[random.randint(0,nT-1),:]
    iteration=0
    while iteration<15:
        print(params)
```

```
# Minimization of the objective function using SLSQP algorithm
opt=scipy.optimize.minimize(obiettivo,params,method='SLSQP',constraints=cons)
        params_min=opt.x
        params=params_min
        gamma=gamma*10
        iteration=iteration+1
    parameters_to_save.append(params)
plt.plot(energia,spectra(parameters_to_save[0]))
plt.plot(energia, ref_H_p)
plt.plot(energia, ref_Cu_I_p)
plt.plot(energia, ref_Cu_II_p)
plt.plot(concentrations(parameters_to_save[0]).T)
parameters_to_save[0].reshape((2,3))
t1=np.linalg.lstsq(us_n,spectra(parameters_to_save[0])[:,0],rcond=None)[0]
t2=np.linalg.lstsq(us_n,spectra(parameters_to_save[0])[:,1],rcond=None)[0]
t3=np.linalg.lstsq(us_n,spectra(parameters_to_save[0])[:,2],rcond=None)[0]
Z4=np.transpose(np.array([a,0,0,0]))
Msave=np.vstack([t1,t2,t3,Z4]).T

# Plot Objective
np.savetxt("spectra_Reerences.dat",spectra(parameters_to_save[12]))
np.savetxt("parameters_References.dat",parameters_to_save[12])
np.savetxt("concentrations_References.dat",concentrations(parameters_to_save[12]).
T)
parametri_stimati_Ob=[]
for j in range(1000):
    params_init_Ob=np.zeros(6)
    for i in range(6):
        params_init_Ob[i]=random.uniform(bounds[i][0],bounds[i][1])
    res_Ob = scipy.optimize.minimize(P, params_init_Ob,
method='SLSQP',constraints=cons,tol=1e-6)
    parametri_stimati_Ob.append(res_Ob.x)

# Definition of the procedure of the refinement
def Refinement(parametri_stimati):
```

```
    parametri_stimati_s=copy.copy(parametri_stimati)
    parametri_da_tenere_s=[]
    for i in range(1000):
        spettri=spectra(parametri_stimati_s[i])
        s1=0
        for j in range(nPC):
            for l in range(L):
                if spettri[l,j]>=0 and spettri[l,j]<=2:
                    s1=s1+1
        if s1==nPC*L:
            parametri_da_tenere_s.append(parametri_stimati_s[i])
    parametri_stimati_c=parametri_da_tenere_s
    parametri_da_tenere_c=[]
    for i in range(np.shape(parametri_da_tenere_s)[0]):
        concentrazioni=concentrations(parametri_stimati_c[i])
        concentrazioni=np.transpose(concentrazioni)
        c1=0
        for j in range(nPC):
            for l in range(M):
                if concentrazioni[l,j]>=0 and concentrazioni[l,j]<=1:
                    c1=c1+1
        if c1==nPC*M:
            parametri_da_tenere_c.append(parametri_da_tenere_s[i])
    return parametri_da_tenere_c
Ps=Refinement(parametri_stimati_Ob)
Pz=np.zeros((len(Ps),6))
for i in range(len(Ps)):
    Pz[i,:]=Ps[i]
plt.scatter(Pz[:,0],Pz[:,1])
Oz=np.zeros(np.shape(Pz)[0])
for i in range(len(Oz)):
    Oz[i]=obiettivo(Pz[i,:])
x=Pz[:,0]
y=Pz[:,1]
xi, yi = np.linspace(x.min(), x.max(), 100), np.linspace(y.min(), y.max(), 100)
```

```
xi, yi = np.meshgrid(xi, yi)
rbf = scipy.interpolate.Rbf(x, y, Oz,function='linear')
zi = rbf(xi, yi)
plt.contourf(zi, vmin=Oz.min(), vmax=Oz.max(), origin='lower',extent=[x.min(),
x.max(), y.min(), y.max()])
plt.colorbar()
plt.show()

# Finish Plot Objective
```

APPENDIX B

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
import random
import copy
import collections
from scipy.optimize import basinhopping
import scipy.interpolate
from mpl_toolkits import mplot3d

# Import of the experimental dataset
dataset_S=np.loadtxt('norm_5spettri_XAS_PhSMe_0.35M_49spettri_da7100eV.dat')
            ('norm_deg_99_spettri_PhCH2OH_2.19M_nuovoformato_da7100eV.dat')
data=dataset_S[:,1:]
dataset_C=np.loadtxt('PhSMe_Pure_Concentrations.dat')
                    ('PhCH2OH_Concentrations_XAS.dat')
energia=dataset_S[:,0]
scan_index=dataset_C[:,0]
concentrazioni=dataset_C[:,1:]

# Procedure of the normalization
def normalization(energy,data):
   scaled=np.zeros(np.shape(data)[1])
   for i in range(np.shape(data)[1]):
      scaled[i]=np.sqrt((1./((1./(np.max(energy)-
np.min(energy)))*(np.trapz((data[:,i])**2,energy)))))
   for i in range(np.shape(data)[1]):
      data[:,i]=data[:,i]*scaled[i]
   return data

# Procedure of the interpolation
def interpolation(energy,data,step):
```

```
    e_valor=np.arange(min(energy),max(energy),step)
    dat_valor=[]
    col=np.shape(data)[1]
    for i in range(col):
        d=np.interp(e_valor,energy,data[:,i])
        dat_valor.append(d)
    data=np.transpose(dat_valor)
    energy=e_valor
    return energy, data
def spectra(params):
    params=np.reshape(params,(1,2))
    t=np.vstack((Z,params))
    return np.dot(us,t)
def concentrations(params):
    params=np.reshape(params,(1,2))
    t=np.vstack((Z,params))
    return np.dot(np.linalg.pinv(t),vt)

# Definition of the penalty function
def P(params):
    term_S=np.zeros((L,N))
    term_C=np.zeros((M,N))
    sp=spectra(params)
    cp=concentrations(params).T
    for j in range(N):
        for i in range(L):
            term_S[i,j]=H_s(sp[i,j])*(sp[i,j])**2
    for i in range(N):
        for j in range(M):
            term_C[j,i]=H_c(cp[j,i])*(cp[j,i])**2
    penalty=(((sum(sum(term_S)))+(sum(sum(term_C)))))
    return penalty

# Definition of the spectral step function
def H_s(x):
```

```
    if x<-0.5 or x>2.5:
        return 1
    elif x>=-0.5 and x<=2.5:
        return 0

# Definition of the concentration step function
def H_c(x):
    if x<-0.5 or x>1.5:
        return 1
    elif x>=-0.5 and x<=1.5:
        return 0

def Obiettivo(params):
    conc=np.transpose(concentrations(params))
    return sum(sum((conc-concentrazioni)**2))/sum(sum(concentrazioni)**2)

# Definition of the procedure of the refinement
def Refinement(parametri_stimati):
    parametri_stimati_s=copy.copy(parametri_stimati)
    parametri_da_tenere_s=[]
    for i in range(100):
        spettri=spectra(parametri_stimati_s[i])
        s1=0
        for j in range(nPC):
            for l in range(L):
                if spettri[l,j]>=0 and spettri[l,j]<=2:
                    s1=s1+1
        if s1==nPC*L:
            parametri_da_tenere_s.append(parametri_stimati_s[i])
    parametri_stimati_c=parametri_da_tenere_s
    parametri_da_tenere_c=[]
    for i in range(np.shape(parametri_da_tenere_s)[0]):
        concentrazioni=concentrations(parametri_stimati_c[i])
        concentrazioni=np.transpose(concentrazioni)
        c1=0
```

```
        for j in range(nPC):
            for l in range(M):
                if concentrazioni[l,j]>=0 and concentrazioni[l,j]<=1:
                    c1=c1+1
        if c1==nPC*M:
            parametri_da_tenere_c.append(parametri_da_tenere_s[i])
    return parametri_da_tenere_c

# Normalization of the concentrations
lista_somma=[]
for i in range(np.shape(concentrazioni)[0]):
    f_r=concentrazioni[i,0] + concentrazioni[i,1]
    lista_somma.append(f_r)
concentrazioni[:,0]=concentrazioni[:,0]/lista_somma
concentrazioni[:,1]=concentrazioni[:,1]/lista_somma
_=plt.plot(scan_index,concentrazioni,'-*')

# Normalization of the spectra
energia,data=interpolation(energia,data,step=0.005)
spettri=normalization(energia,data)
L,M=np.shape(spettri)
nPC=2
N=nPC
_=plt.plot(energia,spettri)

# Singular value decomposition
U,S,VT=np.linalg.svd(spettri, full_matrices=False)
_=plt.plot(np.log10(S),'-*')
plt.xlim(-0.50,6)

# Principal components number
nPC=2
u=U[:,:nPC]
vt = VT[:nPC,:] # cij
s=np.diag(S[:nPC])
```

```
us=np.dot(u,s)
_=plt.plot(us)

# Parameters of the spectral normalization
a=-1*np.sqrt((1./((1./(np.max(energia)-
np.min(energia)))*(np.trapz((us[:,0])**2,energia)))))
Z = np.full(2, -a)
print(Z)
parametri=[1,2]

# AFS calculation. Warning! Long calculation!
parametri_stimati=[]
for j in range(1000):
    print(j)
    params_init=np.zeros(2)
    for i in range(2):
        params_init[i]=random.uniform(-5, 5)
    res = minimize(P, params_init, method='nelder-mead',options={'xtol': 1e-6})
    parametri_stimati.append(res.x)
for i in range(len(parametri_stimati)):
    plt.plot(energia,spectra(parametri_stimati[i]))
Pz=np.zeros((len(parametri_stimati),2))
for i in range(len(parametri_stimati)):
    Pz[i,:]=parametri_stimati[i]
plt.scatter(Pz[:,0],Pz[:,1])
plt.xlabel("Energy (eV)")
plt.ylabel("Norm. XANES")

plt.axhline(y=-10)
plt.axhline(y=15)
plt.axvline(x=-8)
plt.axvline(x=12)
plt.show()

# Constraints for Minimization
```

```
p0=[1,2]
v01x=-8
v02x=12
v11y=-10
v12y=15
bounds=[[v01x,v02x],[v11y,v12y]]
cons = []
for factor in range(len(bounds)):
    lower, upper = bounds[factor]
    l = {'type': 'ineq',
         'fun': lambda x, lb=lower, i=factor: x[i] - lb}
    u = {'type': 'ineq',
         'fun': lambda x, ub=upper, i=factor: ub - x[i]}
    cons.append(l)
    cons.append(u)
opt=scipy.optimize.minimize(Obiettivo,p0,constraints=cons,method='SLSQP')
plt.plot(concentrazioni,'-*')
plt.plot((concentrations(opt.x).T),'-*')
_=plt.plot(energia,(spectra(opt.x)))

# Nelder-Mead Optimization
p0=[1,2]
opt=scipy.optimize.minimize(Obiettivo,p0,method='nelder-mead',options={'xtol': 1e-
6})
opt.x
plt.plot(concentrazioni,'-*')
plt.plot((concentrations(opt.x).T),'-*')
_=plt.plot(energia,(spectra(opt.x))) ## obtained spectra

# 2D Map
matrix_P0=np.zeros((50,2*nPC))
v_range=2
for i in range(2*nPC):
    matrix_P0[:,i]=np.linspace(-v_range,v_range,50)
t21=matrix_P0[:,0]
```

```
t22=matrix_P0[:,1]
p_values=np.vstack(np.meshgrid(t21,t22)).reshape(2,-1).T
Oz=np.zeros(np.shape(p_values)[0])
for i in range(len(Oz)):
    Oz[i]=Obiettivo(p_values[i,:])
x=p_values[:,0]
y=p_values[:,1]
xi, yi = np.linspace(x.min(), x.max(), 100), np.linspace(y.min(), y.max(), 100)
xi, yi = np.meshgrid(xi, yi)
rbf = scipy.interpolate.Rbf(x, y, Oz,function='linear')
zi = rbf(xi, yi)
plt.contourf(zi, vmin=Oz.min(), vmax=Oz.max(), origin='lower',extent=[x.min(),
x.max(), y.min(), y.max()])
plt.colorbar()
plt.show()
np.savetxt('x.dat',x)
np.savetxt('y.dat',y)
np.savetxt('z.dat',Oz)
```