

**Separable nonlinear models:
theory, implementation and applications
in physics and chemistry**

This thesis was reviewed by:

prof.dr. J. de Leeuw (University of California, Los Angeles, USA)
prof.dr.ir. S. Van Huffel (Katholieke Universiteit Leuven, Belgium)
dr. R. Wehrens (Radboud University Nijmegen, The Netherlands)
prof.dr. M.L. Groot (Vrije Universiteit Amsterdam, The Netherlands)
dr. F. Bijma (Vrije Universiteit Amsterdam, The Netherlands)

Cover art by Arthur R. Mullen III.

ISBN 978-90-8659258-6

This research was financially supported by the Netherlands Organisation for Scientific Research (NWO) in the framework of the Computational Sciences Programme as project 635.000.014.



Netherlands Organisation for Scientific Research

VRIJE UNIVERSITEIT

**Separable nonlinear models: theory,
implementation and applications in
physics and chemistry**

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op maandag 17 november 2008 om 15.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Katharine Megan Mullen

geboren te Connecticut, Verenigde Staten van Amerika

promotor: prof.dr.ir. H.E. Bal

copromotor: dr. I.H.M. van Stokkum

Contents

1	Introduction	7
1.1	Theory	8
1.2	Implementation	9
1.3	Applications	9
1.4	Structure and notation	10
1.5	Synopsis	10
2	Variable projection and applications	11
2.1	Introduction	11
2.2	Variable projection	13
2.3	Spectroscopy applications	19
2.4	Microscopy applications	23
2.5	Mass spectrometry applications	26
2.6	Future work	29
2.7	Conclusions	30
3	Sum-of-exponentials models	31
3.1	Introduction	31
3.2	Gradient-based algorithms	33
3.3	Parameter precision	35
3.4	Data for a simulation study	37
3.5	Computational results	39
3.6	Conclusions	43
4	Constraints on the linear parameters	45
4.1	Introduction	45
4.2	Constraints on the linear parameters	48
4.3	Non-negativity constraints	50
4.4	Simulation study	55
4.5	Implementation	57
4.6	Conclusion	57

5	Implementation in the R package TIMP	59
5.1	Introduction	59
5.2	The implementation of TIMP	67
5.3	User-accessible functions	71
5.4	General model options	72
5.5	Kinetic models	76
5.6	Spectral models	88
5.7	Extension of supported model types	103
5.8	Conclusions	103
5.A	Comparison of implementations	105
5.B	New nls options	107
5.C	Data formats for input into TIMP	108
6	Separable nonlinear models for FLIM data	111
6.1	Introduction	111
6.2	Statistical model for FLIM data	113
6.3	FLIM data collection and analysis	114
6.4	Extension of TIMP for FLIM data analysis	115
6.5	A simulation study	118
6.6	Case study on measured CFP data	123
6.7	Conclusions	129
7	Models for time-resolved mass spectra	131
7.1	Introduction	131
7.2	Methods	132
7.3	Results and Discussion	135
7.4	Conclusions	145
8	Simulation studies in modeling GC/MS data	147
8.1	Introduction	147
8.2	MCR-ALS	150
8.3	Parametric model-based global analysis	153
8.4	Comparison of the methods	155
8.5	Implementation and future work	163
8.6	Conclusions	165
9	Summary	167
9.1	Future work	168
9.2	Samenvatting	169
	Acknowledgements	171
	Bibliography	184

Chapter 1

Introduction

This thesis is concerned with models consisting of a linear superposition of nonlinear functions with random effects made up of additive Gaussian white noise. Such models are very useful for modeling a wide range of physical systems, as will be explored herein. The inverse problem associated with estimating the most likely values for the parameters is an instance of separable nonlinear least squares, which is possible to address with the variable projection algorithm due to Golub and Pereyra (1973). This algorithm is described in detail in Chapter 2, along with several extensions.

Many varieties of measured data arising in physics and chemistry experiments represents a linear superposition of a number of components. Such measurements are often resolved with respect to multiple independent variables, such as time, wavelength, mass-to-charge ratio or location. In a selection of the independent variables, the behavior of each component may be possible to describe in terms of a nonlinear function, the form of which is suggested by physical first principles. However, in other independent variables a parsimonious nonlinear function to describe the component may not be feasible to construct. For example, measurements that arise in the time-resolved microscopy experiments described further in Chapters 2 and 6 represent the fluorescence decay of several components of a fluorescent protein after an initial laser excitation. These measurements are resolved with respect to both time and location. Each component is known to decay exponentially in time, and the measurement at each location is possible to describe well as a sum of exponential decays. However, the amplitude of each component at each location depends on unpredictable biological and experimental conditions that are difficult to represent in terms of a simple nonlinear function. In using a separable nonlinear model to describe such measurements, the amplitude of each component at each location may be treated as conditionally linear on estimates for the parameters associated with the nonlinear functions that describe the data in the time domain. This allows the data with respect to both time and location to be described in terms of a unified model having two types of parameters: intrinsically nonlinear parameters associated with the functions in the time domain, and conditionally linear

parameters associated with the amplitude of each component per-location.

The models considered here will share the feature that the parameters are possible to naturally separate into two groups, with estimates for the parameters in one group being easy to solve for giving fixed values of parameters in the other group. This feature is what defines a separable nonlinear model. Several advantages are to be had in using an algorithm for model fitting under least squares criteria that takes advantage of this special structure, as opposed to using a general nonlinear regression strategy to estimate all parameters directly. The parameter estimation task is rendered better-conditioned, meaning that it is easier to solve precisely using a digital computer. The problem is also rendered faster to solve, in that fewer iterations are typically required to converge on parameter values that represent a local solution.

Taking full advantage of the separable structure of nonlinear parameter estimation problems is increasingly important in the physical sciences, where improvements in instrumentation are making it possible to collect ever-larger datasets resolved with respect to many independent variables. The description of all measurements of interest in all independent variables in terms of a unified separable nonlinear model is a powerful means of extracting physically relevant information regarding a complex system of interest, as the applications studied here will attest.

1.1 Theory

Given a separable nonlinear model, what algorithms are available to estimate the parameters under least squares criteria? How precise will these parameter estimates be? Chapters 2 and 3 consider these questions. Much of the presentation centers on the variable projection algorithm, which has a long history of successful application to separable nonlinear least squares problems. Chapter 2 reviews variable projection with emphasis on modifications to fit multiway datasets in the absence of large memory resources.

The parameter precision associated with two different variants of variable projection is compared using the Fisher Information Matrix (FIM) of the parameters in Chapter 3. The FIM is again used in Chapter 4 to study the parameter precision for the case that non-negativity constraints on the linear parameters of a separable nonlinear model are active.

Sum of exponential fitting problems are important in applications, as Istratov and Vyvenko (1999) review. This is due to the fact that many physical phenomena are described by first-order differential equations, the solution of which is a sum of exponential decays. Linear approximation standard error estimates are compared to likelihood-based standard error estimates for a sum of exponentials fitting problem typical of those that arise in time-resolved spectroscopy applications in Chapter 3.

A variant of variable projection is developed in Chapter 4 that allows the application of constraints to the linear parameters in separable nonlinear models fit with least squares criteria. Constraining the linear parameters associated

with a separable nonlinear model is useful, for instance, if certain parameter values are not possible to physically interpret. For instance, the linear parameters associated with models for time-resolved fluorescence spectroscopy and mass spectrometry data often represent spectra resolved with respect to wavelength or mass to charge ratio, negative values of which have no physical interpretation.

1.2 Implementation

In order to fit separable nonlinear models under least squares criteria to experimental data, a problem-solving environment has been designed and implemented in the form of the package **TIMP** for the R language and environment for statistical computing (R Development Core Team, 2008). Implementation in R has several advantages. It is available under the terms of the GNU General Public License in source code form, and compiles and runs on all major operating systems. This means that R (and **TIMP**) need not be treated as a black box: if necessary, researchers can examine or modify the implementation at a very low level. The package system of R means that **TIMP** can be easily distributed, facilitating collaboration and the reproduction of research results. Furthermore, R's current position as the *lingua franca* of statistical computing means that **TIMP** can be easily integrated with a wide range of other packages.

TIMP was designed with extension to new types of data in mind. As Chapter 5 explains, the object-oriented structure allows much of the core functionality of the package to be re-used when writing extensions to handle new types of data and models.

The goal of the implementation is to support the process of separable nonlinear model specification, optimization under least squares criteria and validation. Several case studies in Chapter 5 illustrate the use of **TIMP** for these purposes.

1.3 Applications

Applications in multiway spectroscopy, time-resolved microscopy and time-resolved mass spectrometry have motivated much of the work described herein. Of these applications, those in multiway spectroscopy are the most well-established. In time-resolved spectroscopy in particular, separable nonlinear models have long been used to help gain an understanding of the dynamics of complex systems. Chapter 2 outlines some of the history of the use of separable nonlinear models for time-resolved spectroscopy data, and introduces the compartmental models that are often utilized in this application area by way of a case study on time-resolved fluorescence spectroscopy data. Further case studies in fitting multiway spectroscopy data are described in Chapter 5 and involve time-resolved difference absorption spectra, time-resolved fluorescence spectra and temperature-resolved absorption spectra.

Time-resolved microscopy applications are becoming increasingly important as the technique of Fluorescence Lifetime Imaging Microscopy (FLIM) matures.

FLIM measurements are useful for the study of cells and other biological systems *in vivo* and *in situ*, as introduced further in Chapter 2. The utility of the variable projection algorithm for fitting separable nonlinear models to FLIM data is explored in detail in Chapter 6 by way of simulation studies and a control experiment.

The first applications of separable nonlinear models to time-resolved spectrometry data are now being developed. Chapter 7 demonstrates the solution of component resolution problems in gas chromatography mass spectrometry (GC/MS) data via fitting a separable nonlinear model under least squares criteria. The approach is compared to a non-parametric algorithm for component resolution in Chapter 8 by way of simulation studies inspired by measured GC/MS data.

1.4 Structure and notation

Chapters 2-8 have been published or are submitted for publication as journal articles in slightly modified form. This leads to some repetition in the presentation. Each chapter may be read independently of the others, and the notation is not entirely uniform between chapters, (Chapter 8, for instance, uses notation that is common in the chemometrics literature). In some cases, when describing the estimate for a parameter θ , the notation $\hat{\theta}$ is not used.

Chapter 9 consists of a summary of the monograph in English and in Dutch, and some discussion of future work.

1.5 Synopsis

This thesis considers separable nonlinear models with random effects comprised of additive Gaussian white noise and applications thereof in physics and chemistry. Particular attention is devoted to the use of the variable projection algorithm to fit such models under least squares criteria to multiway data. Extensions of variable projection to constrain the conditionally linear parameters are developed, along with methodology to estimate the precision of parameter estimates. Software to fit separable nonlinear models under least squares criteria to data arising in physics and chemistry is implemented as the package **TIMP** for the R language and environment for statistical computing. Applications of separable nonlinear models to data arising in multi-way spectroscopy, time-resolved microscopy and time-resolved mass spectrometry experiments are investigated in-depth.

Chapter 2

The variable projection algorithm in time-resolved spectroscopy, microscopy and mass spectrometry applications¹

2.1 Introduction

An unconstrained nonlinear optimization problem

$$\min_{x \in \mathbb{R}^n} \gamma(x) \quad (2.1)$$

is *separable* if the variables $x \in \mathbb{R}^n$ can be partitioned into $x = (a, z)$ with $a \in \mathbb{R}^p$, $z \in \mathbb{R}^q$, $p + q = n$ such that the sub-problem

$$\min_{a \in \mathbb{R}^p} \gamma(a, z) \quad (2.2)$$

is *easy to solve* for fixed z . In other words, separable optimization problems are those in which it is possible to partition variables x into two sets a and z such that the variables a can be determined as a function of the variables z , written $a(z)$, in a way that is not too difficult to compute (in a sense that will be defined momentarily). A separable optimization problem can be formulated in the reduced space of z alone, as

$$\min_{z \in \mathbb{R}^q} \gamma(a(z), z) \quad (2.3)$$

¹A version of this chapter will appear as Mullen and van Stokkum (2008d) in *Numerical Algorithms*.

although every evaluation of the objective function requires computation of $a(z)$. What is meant by *easy to solve* or *not too difficult to compute* is simply that solving the separated Problem 2.3 is in some sense preferable in comparison to solving the unseparated Problem 2.1, e.g., in terms of execution time, conditioning, or ease of use, since starting values for the parameters a are not required. See, e.g., Ruhe and Wedin (1980); Parks (1985) for further introduction to separable optimization problems.

A common separable optimization problem is the task of fitting a linear combination of n_{comp} nonlinear functions to observations under least squares criteria, that is,

$$\min_{z \in \mathbb{R}^q} \left\| \psi - \sum_{i=1}^{n_{\text{comp}}} c_i(z) a[i] \right\|^2 \quad (2.4)$$

where ψ is an m -vector of observations, c_i is a nonlinear function of z that outputs an m -vector, $a[i] \in a$ for $i = 1, 2, \dots, n_{\text{comp}}$, and $\| \cdot \|$ is, as throughout, the 2-norm. This problem can be written in matrix notation as

$$\min_{z \in \mathbb{R}^q} \left\| \psi - C(z)a \right\|^2 \quad (2.5)$$

where column i of C corresponds to $c_i(z)$.

Golub and Pereyra (1972, 1973) developed an algorithm for Problem 2.5 termed *variable projection*, which eliminates the linear parameters a analytically from the problem, and assumes, as we do throughout, that the objective function is twice differentiable. Variable projection has been applied to the solution of instances of Problem 2.5, termed separable nonlinear least squares problems, in a wide variety of applications, as Golub and Pereyra (2003) review. Lukeman (1999) traces the history of developments of algorithms for Problem 2.5, and describes the connection to solving systems of n_{comp} nonlinear equations written as $C(z)a + \psi = 0$. Golub and LeVeque (1979) provide an extension of the algorithm to problems in which the same nonlinear functions are used to describe each column of matrix data Ψ but the conditionally linear parameters a vary per-column, so that the associated optimization problem is

$$\min_{z \in \mathbb{R}^q} \left\| \Psi - C(z)E^T \right\|^2 \quad (2.6)$$

where Ψ is $m \times n$, with column j storing ψ_j , and E is $n \times n_{\text{comp}}$, with row j storing $a_j = a_j[1], a_j[2], \dots, a_j[n_{\text{comp}}]$. Golub and LeVeque (1979) and Gay and Kaufman (1991) refer to Problem 2.6 as a separable nonlinear least squares problem with multiple right-hand sides. In the time-resolved spectroscopy and microscopy literature (e.g., Beechem *et al.* (1985)), Problem 2.6 is termed the problem of *global analysis*, referring to the assumption that the same nonlinear functions underlie each column of matrix data.

The statistical model associated with Problems 2.5 and 2.6 can be written (per-column j in the case of Problem 2.6) as

$$\psi_j = C_j(z)a_j + \epsilon \quad (2.7)$$

where each element of the noise vector ϵ represents an independent and identically distributed Gaussian random variable with expectation 0 and constant variance ($NID(0, \sigma^2)$). When matrix data is considered, the function C_j may be different for each j . Furthermore, Ψ in Equation 2.6 may be a *ragged* matrix (Black, 2004), meaning that the vectors ψ_i and ψ_j with $i \neq j$ may represent a different number of observations, which possibly arise under different experimental conditions. The connection between observations ψ_j for $j = 1, 2, \dots, n$ is via the vector of nonlinear parameters z . A given function C_j may depend on some or all of the parameters in z . In applications it is common that the observations take the form of vectors $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$ with j representing an independent variable (like a wavelength, or a location) measured many times under different conditions, indexed $1, 2, \dots, K$. An assumption that is often physically motivated is that the same vector of conditionally linear parameters a underlies $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$ (as in Equation 2.14). We will discuss these situations in turn.

A single residual function representing the sum of squared errors that result from using Equation 2.7 to model a collection of vectors ψ_j is formed and minimized under the variants of variable projection discussed in Section 2.2.1. The methodology presented there has allowed the extension of variable projection to d -way datasets, with $d > 2$, and offers many possibilities for the synthesis of data arising in different experiments via a unified model having a single vector z of associated nonlinear parameters.

Here we focus on the use of variable projection for solving instances of Problems 2.6 and variations thereof. Section 2.2 describes variable projection in some detail, including implementation strategies, estimation of linear approximation standard errors, and modifications that allow for the inclusion of constraints on the linear parameters. Applications in modeling time-resolved spectroscopy (Section 2.3), microscopy (Section 2.4) and mass spectrometry (Section 2.5) data are also discussed. Variable projection has been instrumental for model-based data analysis in these areas. The applications to microscopy and mass spectrometry data have been newly developed in the years since the review of Golub and Pereyra (2003).

2.2 Variable projection

Given fixed z and $C(z)$ of full rank, the problem

$$\min_{a \in \mathbb{R}^p} \|\psi - C(z)a\|^2 \quad (2.8)$$

is solved analytically as $a = C(z)^+ \psi$, where $C(z)^+$ is the Moore-Penrose pseudoinverse $C(z)^+ = (C(z)^T C(z))^{-1} C(z)^T$. The assumption that $C(z)$ has full rank is used throughout, though note that Golub and Pereyra (1973) also describe a variable projection algorithm for the case that the columns of $C(z)$ are not linearly independent. Using $a = C(z)^+ \psi$, the objective function to minimize

with respect to z for Problem 2.5 is

$$\begin{aligned} f(z) &= \frac{1}{2} \|\psi - C(z)a\|^2 \\ &= \frac{1}{2} \|(I - C(z)C(z)^+)\psi\|^2 \\ &= \frac{1}{2} \|Q_2 Q_2^T \psi\|^2. \end{aligned} \tag{2.9}$$

where the QR decomposition of $C(z) = QR = [Q_1 \ Q_2]R$ is used for numerical stability, so that Q_1 is $m \times n_{\text{comp}}$, Q_2 is $m \times (m - n_{\text{comp}})$, Q is orthogonal, and $R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}$, with R_{11} being $n_{\text{comp}} \times n_{\text{comp}}$ and upper triangular (Golub and van Loan, 1996). $f(z)$ is the variable projection functional, so named because $I - C(z)C(z)^+$ is the projector on the orthogonal complement of the column space of $C(z)$. Some computational efficiency may be gained by dropping the factor Q_2 , and formulating $f(z)$ as $f(z) = \frac{1}{2} \|Q_2^T \psi\|^2$ (Gay and Kaufman, 1991).

Standard algorithms for nonlinear least squares problems, namely Gauss-Newton and Levenberg-Marquardt, can be used to approach \hat{z} that minimize $f(z)$. However, these algorithms require the Jacobian matrix $J = \frac{\partial r(z)}{\partial z}$ of the residual vector $r(z) = Q_2 Q_2^T \psi$, (or, dropping the Q_2 factor, $r(z) = Q_2^T \psi$). Two classes of approaches to obtain J may be distinguished: methods that use a finite difference approximation, and methods that rely on an analytical expression. Many widely applied implementations of nonlinear least squares allow J to be determined by either approach, as in, e.g., the modification of Levenberg-Marquardt found in MINPACK (Moré, 1978) or the Gauss-Newton method employed by the R function `nls` (R Development Core Team, 2008).

If a finite difference approach is used to build up an approximation of J , $r(z)$ is repeatedly evaluated for different parameter vectors \check{z} obtained by perturbing z by an amount h . Using a forward difference approximation, the i th parameter in z is incremented by h , yielding \check{z} , and the i th column of J is determined as $J[:, i] = \frac{r(\check{z}) - r(z)}{h}$, requiring $\text{length}(z)$ evaluations of $r(\check{z})$ to calculate an update of J . The associated computational expense is often more than offset by the advantage of not having to derive and compute an analytic expression for J . For example, in the applications described in Sections 2.3 and 2.4, the method of determining $C(z)$ is often at least partially numerical, and there is no closed-form expression available for even $C(z)$. In these applications obtaining a closed-form expression for the three-dimensional tensor $\frac{\partial C(z)}{\partial z}$ needed for the determination of J via an analytical expression is difficult at best, and a finite difference approximation is always used. Minimization of the variable projection functional $f(z)$ with Levenberg-Marquardt using a finite difference approximation of J was described by Lawton and Sylvestre (1971).

If an analytical expression for $\frac{\partial C(z)}{\partial z}$ is available, then the method suggested

by Golub and Pereyra (1973) may be used to determine J per-column i as

$$J[, i] = - \left[\left(C(z)C(z)^+ \frac{\partial C(z)}{\partial z[i]} C(z)^+ \right) + \left(C(z)C(z)^+ \frac{\partial C(z)}{\partial z[i]} C(z)^+ \right)^T \right] \psi \quad (2.10)$$

where we follow the presentation in Golub and Pereyra (2003). Kaufman (1975) suggested that this expression for J could be well-approximated by only using the first term, i.e.,

$$J[, i] = - \left(C(z)C(z)^+ \frac{\partial C(z)}{\partial z[i]} C(z)^+ \right) \psi, \quad (2.11)$$

thereby introducing only a negligible loss in accuracy and significant savings in the number of computations required. The Kaufman approximation is discussed in detail in Bates and Lindstrom (1986); Gay and Kaufman (1991); Mullen *et al.* (2007).

Once J or an approximation of J has been determined, the standard algorithms for nonlinear least squares calculate the gradient ∇ of $f(\hat{z})$ as

$$\nabla f(z) = J^T r(z) \quad (2.12)$$

and also calculate an approximation for the Hessian $\nabla^2 f(z)$. $\nabla f(z)$ and $\nabla^2 f(z)$ allow determination of a direction and step size to move the current estimates \hat{z} in z -space so that $f(\hat{z})$ is reduced. From the new parameter estimates, the process of determining J and making a new step in z -space is iterated until stopping criteria are met. The details of the standard algorithms are found in e.g., Bates and Watts (1988); Seber and Wild (2003).

Several results justify and motivate the minimization of $f(z)$ as opposed to the unreduced objective function with parameters $x = (z, a)$. Golub and Pereyra (1973) give a proof that the stationary points of $f(z)$ are the same as for the unseparated problem when the rank of $C(z)$ is constant over z -space. Therefore given \hat{z} that determines either a local or global optimum in $f(z)$, we determine $\hat{a} = C(\hat{z})^+ \psi$, and arrive at estimates $\hat{x} = (\hat{a}, \hat{z})$ that define a local or global optimum in the objective function of the unseparated problem. Asymptotic convergence analysis under the Gauss-Newton algorithm by Ruhe and Wedin (1980) shows that variable projection has superlinear convergence whereas methods that alternate between solving the linear and nonlinear problems separately are only linearly convergent. This is because, as Sjöberg and Viberg (1997) show, the separated problem is better-conditioned than the unseparated problem. Obtaining a better-conditioned problem is indeed a primary motivation for minimizing the variable projection functional as opposed to the objective function for the unseparated problem. Krogh (1974) provides simple examples in which optimization of an unseparated nonlinear least squares problem results in divergence whereas optimization of the separated problem results in convergence in a small number of iterations. Osborne (2007) points to the extraordinary effectiveness of variable projection in least squares problems involving fitting the parameters of a linear combination of real or complex

exponential functions, which are ubiquitous in applications and notoriously difficult (Istratov and Vyvenko, 1999). Golub and Pereyra (2003) suggest that this success is due to the fact that the unseparated exponential fitting problem becomes increasingly ill-conditioned as (and if) the optimal parameters are converged upon, whereas the variable projection functional does not suffer from this problem.

2.2.1 Implementation

Problem 2.6 can be reformulated as an instance of Problem 2.5 by letting $\text{vec}(\Psi) = (I_n \otimes C(z))\text{vec}(E^T)$, where \otimes is used to denote the Kronecker product. Then variable projection as presented in Section 2.2 can be applied. However, as Golub and LeVeque (1979) realized, this does not take advantage of the special structure of Problem 2.6, and requires storing and operating on the matrix $(I_n \otimes C(z))$, which is large for large n . Golub and LeVeque (1979) thus suggested forming the residual vector associated with Problem 2.6 as

$$r(z) = \begin{bmatrix} Q_2^T \Psi[:, 1] \\ Q_2^T \Psi[:, 2] \\ \vdots \\ Q_2^T \Psi[:, n] \end{bmatrix} \quad (2.13)$$

from which $f(z)$ and J can be determined without ever storing or operating on $(I_n \otimes C(z))$.

Chapter 5 expands on the idea of determining $r(z)$ in a partitioned manner for the description of (ragged) matrix data per-column j using Equation 2.7. In the applications there, $C_j(z)$ varies per-column j , and the residual vector may be determined as in Equation 2.13, but Q_2^T must be calculated for each matrix $C_j(z)$ in $j = 1, 2, \dots, n$. It is also noted in Chapter 5 that in practice in the physical sciences, observations often take the form of vectors $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$ with j representing an independent variable (like a wavelength, or a location) measured many times under different conditions indexed $1, 2, \dots, K$. Then the assumption that the same vector of conditionally linear parameters a underlies $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$, can be accounted for by letting

$$\psi_j = \begin{bmatrix} \psi_{j_1} \\ \psi_{j_2} \\ \vdots \\ \psi_{j_K} \end{bmatrix} = \begin{bmatrix} C_{j_1}(z) \\ C_{j_2}(z) \\ \vdots \\ C_{j_K}(z) \end{bmatrix} a_j = C_j(z)a_j \quad (2.14)$$

where the second subscript on ψ and C is the dataset index. The residual vector associated with using Equation 2.14 to model the columns $j = 1, 2, \dots, n$ of (ragged) matrix data is also determined as in Equation 2.13 with the modification that Q_2^T is re-calculated for each matrix $C_j(z)$.

2.2.2 Standard error estimation

The matrix of first derivatives of the model function with respect to both the nonlinear parameters z and the conditional parameters a_j is

$$J_m = \begin{bmatrix} \frac{\partial C_j(z)a_j}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial C_j(z)}{\partial z} a_j & C_j(z) \end{bmatrix} \quad (2.15)$$

Under the assumption that the noise vector ϵ added to the model $\psi_j = C_j(z)a_j$ is such that every element ϵ_i is $NID(0, \sigma^2)$, the covariance matrix associated with both z and a is

$$\text{cov} \begin{bmatrix} z \\ a_j \end{bmatrix} = \hat{\sigma}^2 (J_m^T J_m)^{-1} \quad (2.16)$$

where $\hat{\sigma}^2 = RSS(\hat{x})/df$, RSS is the residual sum of squares and the degrees of freedom $df = (\sum_{j=1}^n \text{length}(\psi_j)) - \text{length}(z) - (\sum_{j=1}^n \text{length}(a_j))$ (Seber and Wild, 2003).

After writing the residual function as outlined in Section 2.2.1 and using a standard nonlinear least squares implementation to minimize the sum of squares of the residual vector with respect to z , $\text{cov}(\hat{z})$ is often returned along with \hat{z} , whereas $\text{cov}(a_j)$ must be determined. Using

$$J_m^T J_m = \begin{bmatrix} (\frac{\partial C_j(z)}{\partial z} a_j)^T \frac{\partial C_j(z)}{\partial z} a_j & (\frac{\partial C_j(z)}{\partial z} a_j)^T C_j(z) \\ C_j^T(z) \frac{\partial C_j(z)}{\partial z} a_j & C_j(z)^T C_j(z) \end{bmatrix} \equiv \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad (2.17)$$

we have, from the block matrix inversion theorem found in e.g., Seber and Wild (2003), Appendix A,

$$(J_m^T J_m)^{-1} = \begin{bmatrix} X_{11}^{-1} & X_{11}^{-1} X_{12} \\ -X_{21} X_{11}^{-1} & A_{22}^{-1} + X_{21} X_{11}^{-1} X_{12} \end{bmatrix} \quad (2.18)$$

where $X_{11} = A_{11} - A_{12} A_{22}^{-1} A_{21}$, $X_{12} = A_{12} A_{22}^{-1}$, and $X_{21} = A_{22}^{-1} A_{21}$. Then $\sigma^2 X_{11}^{-1} = \text{cov}(\hat{z})$ and we are interested in determining the bottom right block. Since $A_{22}^{-1} = (C_j(z)^T C_j(z))^{-1}$, we have

$$X_{21} = A_{22}^{-1} A_{21} \quad (2.19)$$

$$= (C_j(z)^T C_j(z))^{-1} C_j^T(z) \frac{\partial C_j(z)}{\partial z} a_j \quad (2.20)$$

$$= C_j(z)^+ \frac{\partial C_j(z)}{\partial z} a_j \equiv G_j \quad (2.21)$$

and $X_{12} = A_{12} A_{22}^{-1} \equiv G_j^T$, where G_j consists of columns $C_j^+ \frac{dC_j}{dz_i} a_j$, for each nonlinear parameter z_i . Hence it is possible to write

$$\text{cov}(\hat{a}_j) = \sigma^2 (C_j^+ C_j^{+T}) + G_j \text{cov}(\hat{z}) G_j^T. \quad (2.22)$$

This expression allows determination of $\text{cov}(\hat{a}_j)$ for all $j = 1, 2, \dots, n$ with modest memory resources even when n is large.

2.2.3 Modification for constraints on the conditionally linear parameters

It may be desirable to impose constraints on the linear parameters a and E in Problems 2.5 and 2.6, respectively. Consider the case that a is constrained to non-negative values. Then Problem 2.8 is replaced with the non-negative least squares (NNLS) problem

$$\begin{aligned} \min_{a^* \in \mathbb{R}^p} \quad & \| \psi - C(z)a^* \|^2 \\ \text{subject to} \quad & 0 \leq a_i^* \text{ for } i = 1, 2, \dots, n_{\text{comp}} \end{aligned} \quad (2.23)$$

Problem 2.23 must be solved in place of $a = C^+(z)\psi$ in the expression for $f(z)$, e.g., with the algorithm by Lawson and Hanson (1974), so that Equation 2.2 becomes

$$f(z) = \frac{1}{2} \| \psi - C(z)a^* \|^2 \quad (2.24)$$

Non-negativity constraints on a arise when ψ represents count data. Then the noise statistics are often best represented using the Poisson distribution, but for data comprised of large counts may be well-approximated by additive $NID(0, \sigma^2)$ noise, so that parameter estimation may proceed by minimization of $f(z)$. When using a finite difference method to obtain J , a variable projection algorithm that adds non-negativity constraints to the conditionally linear parameters a is obtained by using the definition of $f(z)$ given in Equation 2.24 in place of that given in Equation 2.2, as Mullen and van Stokkum (2008c) discuss. When using an analytical expression for J , an approximate expression based on the Jacobian in the absence of constraints may be used, as Sima and Van Huffel (2007) discuss.

In the case that the constraints applied to a do not ensure non-negativity, but rather some other property, the NNLS problem used to determine a^* must be replaced with the appropriate constrained optimization problem. The only restriction on the constraints applied is practical; the separated problem with constraints on a or E should remain easier to solve than the equivalent unseparated problem. Sima and Van Huffel (2006) have described the imposition of regularization constraints on a by replacing the least squares problem $a = C(z)^+ \psi$ in the variable projection functional with $a = (C(z)^T C(z) + m\lambda B)^{-1} C(z)^T \psi$, where the term $m\lambda B$ is used to impose a certain degree of smoothness on a .

Modification for equality constraints between conditionally linear parameters

Equality constraints that set $a_j[g]$ to zero for component g , or set $a_j[g] = a_j[h]$ for components g and h are often incorporated to make the estimation problem better determined or account for *a priori* knowledge of the system underlying the observations. Such constraints are common in time-resolved spectroscopy applications where $a_j[g]$ represents the spectrum of component g at wavelength

j . In the case of equality constraints that set $a_j[g]$ to zero, we remove column $C_j[,g](z)$ and element $a_j[g]$ from the model $\psi_j = C_j(z)a_j$. This results in a model with fewer free conditionally linear parameters. We refer to such equality constraints as *zero constraints*.

For the case of equality constraints that set $a_j[g]$ to be equal to $a_j[h]$, possibly with a linear scaling factor α_i , we let column $C_j[,h](z)$ be equal to $C_j[,g](z) * \alpha_i + C_j[,h](z)$, and then remove column $C_j[,g](z)$ from the model $\psi_j = C_j(z)a_j$. This also results in a better determined model. Note that α_i may be optimized as a nonlinear parameter.

2.3 Spectroscopy applications

Variable projection is often used to fit the free parameters of models for measurements of light, which often consist of a linear superposition of several nonlinear functions. The algorithm was first used to estimate the parameters of a model associated with measurements of light by Rust *et al.* (1976), who parametrize the decay of the luminosity of supernovae in time.

Measurements of light resolved with respect to wavelength or wavenumber and one or more independent variables, such as time, are referred to as multiway spectroscopy data. Variable projection was first applied to multiway spectroscopy data by Golub and LeVeque (1979), who analyze time-resolved spectroscopy measurements describing the photocycle of bacteriorhodopsin. Nagle *et al.* (1982) also consider data on bacteriorhodopsin, and had a great influence in spreading the use of variable projection in the biophysics community. Part of the impact of this work can be explained by the authors' elaboration of the use of the nonlinear parameters z to represent a compartmental model for the kinetics, a theme which is further elaborated in Section 2.3.1. Nagle (1991b,a); van Stokkum *et al.* (1993); Nagle *et al.* (1995) further elaborate on the utility of variable projection for describing multiway spectroscopy data with emphasis on compartmental models for kinetics. The utility of compartmental models for describing time-resolved spectroscopy data representative of complex systems is stressed by Beechem *et al.* (1985); Holzwarth (1996). van Stokkum and Lozier (2002) describe an in-depth case study in using variable projection to fit a separable nonlinear model for 5-way spectroscopy data (resolved with respect to wavelength, time, temperature, pH and polarization) representing the photocycle of bacteriorhodopsin. van Stokkum *et al.* (2004); van Stokkum (2005) review separable nonlinear models for the description of time-resolved spectroscopy data, the free parameters of which are optimized with variable projection.

Golub and Pereyra (2003) discuss at length the application of variable projection to optimizing the parameters of models for Nuclear Magnetic Resonance (NMR) spectroscopy data, which often take the form of a linear superposition of complex damped exponentials. The algorithm has been important in this application domain since the influential work of van der Veen *et al.* (1988). Note that while the model describes complex data, the parameter estimation prob-

lem involves a residual vector and nonlinear parameters in the real domain, as described in detail in e.g., Sima and Van Huffel (2007). The latter also describe an extension to variable projection to account for non-negativity constraints on the linear coefficients, motivated by problems in modeling NMR spectra.

Multi-way spectroscopy data of all varieties can be stored as a ragged matrix Ψ , where each column represents a value of a spectroscopic property, and each row represents a value of some other independent variable. To fix ideas, consider data resolved with respect to time and wavelength, which arises in time-resolved spectroscopy experiments. The data often represents n_{comp} components, each with a distinct time profile $C[,j]$ and spectrum $E[,j]$. By the matrix analogue of the Beer-Lambert law for absorption, $C[,j]$ and $E[,j]$ often combine linearly, giving rise to the equation

$$\Psi = CE^T \quad (2.25)$$

where the $m \times n_{\text{comp}}$ matrix C and the $n \times n_{\text{comp}}$ matrix E represent the data in time and wavelength, respectively. It is also often the case that a parametric description of either C or E , but not both matrices, is available, so that parameter estimation requires solution of an instance of Problem 2.6 or generalizations thereof.

2.3.1 Example: Time-resolved fluorescence emission measurements of photosystem I

In order to give an idea of the possibilities that variable projection has opened up in the domain of modeling time-resolved spectroscopy data, we consider a case study in brief. Photosystem I (PS-I) is one of two photosystems in oxygenic photosynthesis, a process by which plants and green algae convert photons into chemical energy. The PS-I core is a distinct functional unit of PS-I. Gobets *et al.* (2001) describe a system consisting of PS-I cores in a buffer excited by a short laser pulse of femtosecond duration. Measurements of the fluorescence of the system at many wavelengths and times after excitation are then collected with a synchroscan streak camera in combination with a spectrograph, a technique which van Stokkum *et al.* (2006, 2008b) review. The observations considered here represent 48 wavelengths equidistant in the interval 626-785 nanometer (nm), and 914 timepoints in the range 0-200 picosecond (ps) after laser excitation, stored as a 914×48 matrix of data Ψ . The goal of data analysis is to describe the kinetics in terms of a model that parametrizes the formation and decay of each distinct state of the underlying system, while solving for the spectra E as linear coefficients.

In some systems, the exciting pulse results in the formation of all possible states simultaneously, as in the left panel of Figure 2.1. In other systems, a single state may be formed initially, the decay of which results in the formation of a second state, and so on, for all possible states of the system, as in the right panel of Figure 2.1. Alternatively, the states may be related in some other, arbitrarily complicated, way. Physical-chemical principles and ab-initio quantum mechanical calculations typically suggest a few models for the way

the possible states are related and how long they persist, and selection of the model best supported by the data and the underlying theory requires estimation of the free parameters of the possible kinetic models, $C(z)$, while the spectra are treated as the linear coefficients E^T in $\Psi = C(z)E^T$. Since a fluorescence spectrum $E[,g]$ represents a count of the relative number of emission photons contributed to the data by component g at some set of wavelengths, negative values of the spectra E are not interpretable, and thus these parameters are determined using the NNLS method discussed in Section 2.2.3. The reason that the spectra are not often described in terms of parametric model $E(z)$ is that their fine structure is difficult to represent via a function with a small number of free parameters.



Figure 2.1: Two possible compartmental models for a system represented by three components. In the left model, the initial excitation populates all three components, which decay in parallel. In the right model, the initial excitation populates the leftmost compartment, the decay of which populates the middle compartment, and so on, so that the compartments are said to be arranged sequentially.

A linear compartmental model (Godfrey, 1983; Seber and Wild, 2003; van Stokkum *et al.*, 2004) is used to describe the relationship between states like those shown graphically in Figure 2.1 and 2.2. The behavior of the compartmental model is given as a matrix $C(z)$ in which the concentration of a single compartment in time is represented by a column $C[,g]$, with

$$C(z) = \exp(K(z)t) \oplus j(t) \quad (2.26)$$

where z are free parameters, K is a transfer matrix that encodes the allowable transitions between components, and uses microscopic decay rate parameters $k \in z$ and scaling parameters $b \in z$ to describe the rate at which a state is formed and decays, and t is the vector of times that the rows of $C(z)$ represent. The vector $j(t)$ represents the proportion of the system in each compartment at the initial time point, multiplied by the instrument response function (IRF) $i(t)$. The operator \oplus stands for convolution, and in the case that the IRF is described as a Gaussian distribution or other simple function, it may be performed analytically. Here the IRF is described as a Gaussian with a location parameter that is a function of the wavelength, so that the matrix $C(z)$ must be determined per-wavelength as described in Section 2.2.1.

The compartmental model shown in Figure 2.2 was tested as a possible description for the kinetics of time-resolved spectroscopy data representing PS-I

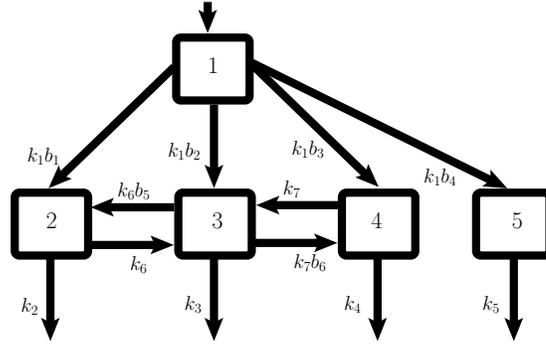


Figure 2.2: A compartmental model for the kinetics of PS-I core complexes. The decay rates of the components are parametrized by k_i . Where there is more than one allowable transition out of a compartment, parameters b_i are used to scale the decay rate k_i associated with the compartment. Here values of b_1, b_2, b_3, b_4 are fixed such that $b_1 + b_2 + b_3 + b_4 = 1$ and such that the spectra estimated for compartments 2 - 5 have approximately equal area, which is a physically motivated constraint.

cores by fitting the free parameters of the model with variable projection. Where there is more than one allowable transition out of a compartment, parameters b_i are used to scale the decay rate k_i associated with the compartment. Here values of b_1, b_2, b_3, b_4 are fixed such that $b_1 + b_2 + b_3 + b_4 = 1$ and such that the spectra estimated for compartments 2 - 5 have approximately equal area, which is expected from physical first principles. The parameters b_5 and b_6 describing the equilibrium between compartments 2 and 3 and compartments 3 and 4 are only possible to estimate by adding zero constraints (as described in Section 2.2.3) to some of the values in the matrix of spectra E . Here zero constraints are applied to all wavelengths of the spectrum for compartment 1, so that it is never emissive, to wavelengths of the spectrum of compartment 2 up to 690 nm, and to wavelengths of the spectrum of compartment 3 up to 697 nm.

The concentration profiles $C(z)$ and spectra E that result from application of this model are shown in Figure 2.3. Standard error estimates are shown on the spectra in Figure 2.3 as vertical bars, and are very small. In order to judge the quality of the fit, traces such as those in Figure 2.4 can be inspected, and the singular value decomposition of the residual matrix can be checked for evidence of systematic structure. The fit of the model described here was deemed satisfactory. The implementation of variable projection used is from the R package **TIMP** (Mullen and van Stokkum, 2007b), and a script to reproduce these results is included on the webpage of the package (Mullen and van Stokkum,

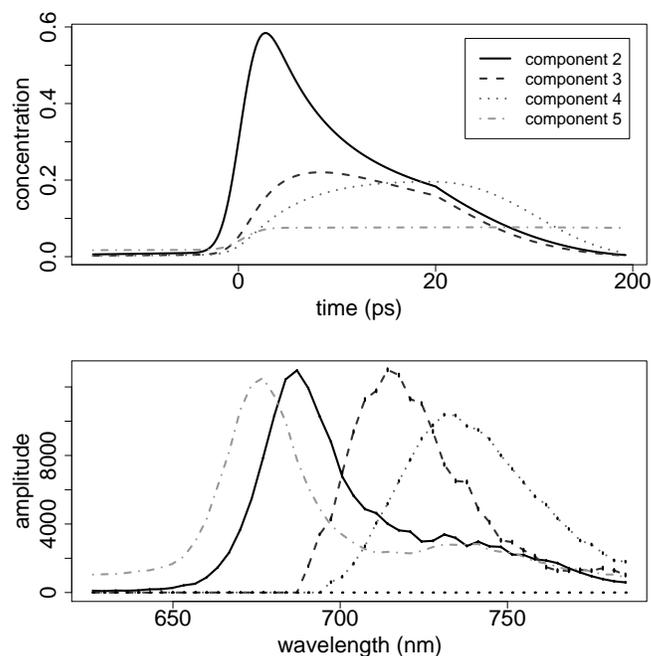


Figure 2.3: The concentration profiles $C(z)$ (upper panel) and spectra E (lower panel) that result from using the compartmental model shown in Figure 2.2 to describe the kinetics of PS-I core complexes. The legend shown in the upper panel applies to the spectra as well. The concentration profile and spectrum of compartment 1 is not shown, since it is non-emissive. Standard error estimates are shown on the spectra as vertical bars, and are so small as to be barely visible.

2008b).

2.4 Microscopy applications

Applications of variable projection to modeling time-resolved microscopy data have been developed since the review of applications of the algorithm by Golub and Pereyra (2003), and are becoming increasingly important as the technique of Fluorescent Lifetime Imaging Microscopy (FLIM) matures. FLIM is widely applied to detect interactions between fluorescently labeled biological molecules such as proteins, lipids, DNA and RNA, and results in a count of photons detected for many time points, at many spatial locations, often with 250 nanometer spatial resolution and sub-nanosecond temporal resolution. In many FLIM experiments, proteins of interest are genetically tagged with variants of the green fluorescent protein (GFP) (Tsien, 1998).

FLIM experiments give rise to a global analysis problem when the same

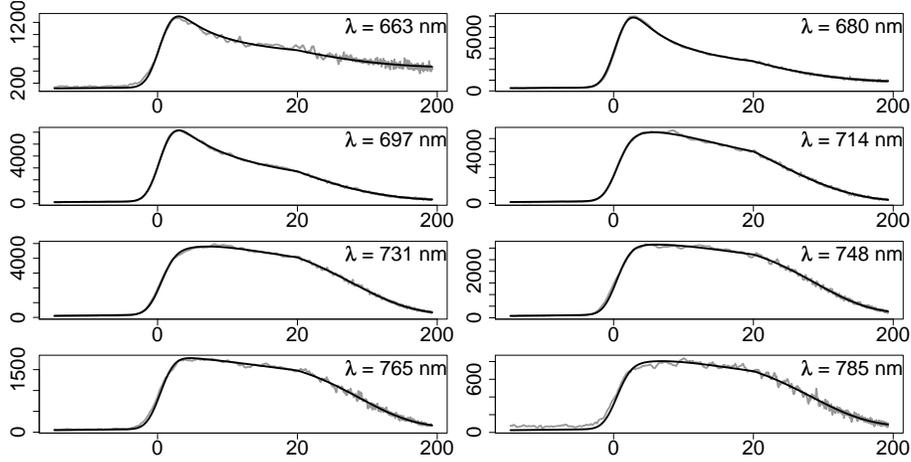


Figure 2.4: Columns of a matrix of time-resolved spectroscopy data (grey), each representing the wavelength λ in nanometers (nm) shown in the upper right corner of each panel. The x-axis represents time in picoseconds. The model is shown in black. Free parameters were fit using variable projection.

n_{comp} kinetic processes may be assumed to underlie the fluorescent decay at all locations. The decay of each kinetic process is exponential, but is complicated by the fact that it must be convolved with the time profile of the instrument response function (IRF). The IRF is often not described well by an analytical function with only a few parameters, and it is often necessary to make a measurement of the IRF time profile, and numerically convolve it with that of the exponential decay used to describe each kinetic process. Thus data analysis requires solution of an instance of Problem 2.6 where each kinetic process g is represented by a column of $C(z)$

$$C[, g] = \exp(-k_g t) \oplus i(t) \quad (2.27)$$

where t is the vector of time points represented by the rows of $C(z)$, $k_g \in z$ is a parameter to be estimated, \oplus indicates convolution and $i(t)$ is the measured time profile of the IRF. In many FLIM experiments, fluorescent dyes give rise to the observed fluorescence, which is measured at the maximum emission wavelength of the dye of interest, and can typically be described by two to four kinetic processes. The amplitude that each kinetic process contributes to the collected image at pixel j (that is, location j) is the row $E[j, \cdot]$ that appears in $\Psi = C(z)E^T$, where Ψ is the time-resolved image. The goal of data analysis is to estimate z and E .

Verveer *et al.* (2000) recognized that variable projection could be applied to global analysis problems arising in FLIM data analysis, but did not implement the partitioned technique described in Section 2.2.1 and therefore were stymied by the large memory resources required. Chapter 6 studies the utility of variable projection for modeling FLIM data via a number of simulation studies

and a control study in estimating the parameters describing the decay of Cyan Fluorescent Protein (CFP).

Note that FLIM data and fluorescence data in general represent a count of the number of photons detected at a given pixel and time. Poisson noise statistics apply to such non-negative count data. For datasets in which the counts are large, the assumption of $NID(0, \sigma^2)$ noise made by least squares methods is acceptable. However, for datasets in which most counts are not large, optimal estimates under least squares criteria do not well-approximate the true underlying parameter values, an issue that has been studied by Maus *et al.* (2001). This issue can be addressed to some extent by weighting each data point $\Psi[i, j]$ by $\frac{1}{\sqrt{\Psi[i, j]}}$, but in order to obtain fully correct estimates it would be necessary to develop an analogue of variable projection for the Poisson noise case.

2.4.1 Example: detection of protein-protein interactions

In this example we consider the simultaneous analysis of multiple FLIM images. Each pixel j in each image $1, 2, \dots, K$ is modeled using Equation 2.7. The nonlinear parameters $k_g \in z$ used to describe the fluorescent decay are estimated using all data included in the simultaneous analysis. To allow a physical interpretation, the vectors $a_{j_1}, a_{j_2}, \dots, a_{j_K}$, describing the amplitudes of the kinetic processes at pixel j in each dataset are constrained to non-negative values by the NNLS method, and the fluorescence decay rate parameters $k_g \in z$ are also constrained to non-negative values by a simple logarithmic transformation ($z_g = \log(k_g)$).

The experiments giving rise to the data involve two proteins known to be homogeneously distributed in the cell nucleus. In one set of experiments, the first protein is tagged with CFP. In the second set of experiments, the first protein is tagged with CFP, and the second protein of interest is tagged with yellow fluorescent protein (YFP). When intracellular dynamics bring the proteins within 1-10 nm of each other, the CFP molecule transfers energy to the YFP molecule. This results in an increase in the decay rate of CFP, which can be observed. The general process in which excited-state energy of a donor fluorophore, like CFP, is non-radiatively transferred to a ground-state acceptor molecule, like YFP, is termed Förster Resonance Energy Transfer (FRET). FRET as measured by FLIM is extensively used to detect protein-protein interactions, as Suhling *et al.* (2005) review.

Since CFP acts as a donor in the CFP-YFP FRET pair, we can use the FLIM set-up to measure only the wavelength at which CFP fluoresces, and examine whether the decay rate of CFP increases in the experiment with CFP and YFP tags as compared to in the experiment with only CFP tags. Such a decrease would be interpreted as evidence of FRET, which would imply that the proteins are often expressed in close proximity.

For the analysis, we select those pixels that represent the nucleus. The two CFP-only datasets whose intensity images are shown in Figure 2.5 (left) are

used in a simultaneous analysis to estimate the associated decay rates. The two CFP-YFP datasets shown in Figure 2.5 (right) are analyzed together in the same way. The decay of CFP is described by a bi-exponential decay in both pairs of datasets.

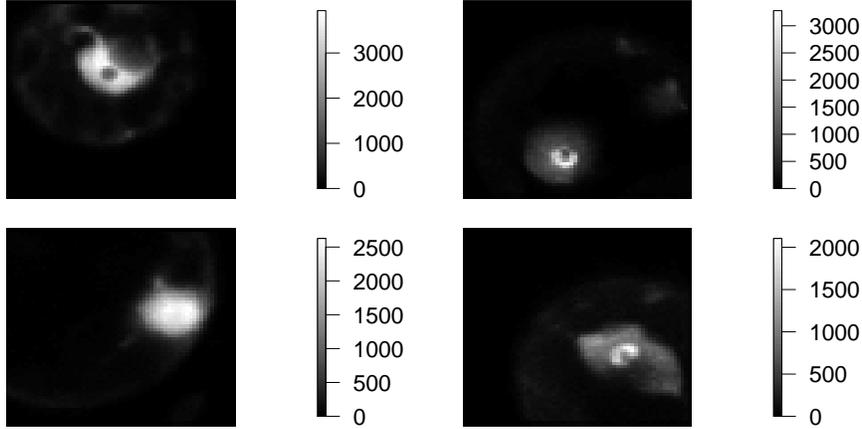


Figure 2.5: Intensity images resulting from FLIM experiments that measure fluorescence from cells at the wavelength at which CFP emission is maximum. Each pixel represents the sum of the data at that pixel over the 206 time-points in the entire time-resolved FLIM dataset. The left plots represent CFP tagged cells; the right plots represent cells tagged with both CFP and YFP. Only pixels representing the cell nucleus are subject to analysis.

The estimated decay rates for CFP in the first pair of datasets are $\hat{k} = \{1.53, 0.34\}$ whereas in the second pair of datasets these decay rates are estimated to be $\hat{k} = \{2.13, 0.44\}$. The average decay rate for pixel j is given as

$$\langle k_g \rangle = \frac{\sum_{g=1}^{n_{\text{comp}}} k_g a_j[g]}{\sum_{g=1}^{n_{\text{comp}}} a_j[g]} \quad (2.28)$$

Figure 2.6 shows the estimated average decay rate per-pixel for CFP in the CFP-only datasets (left panels) and in the CFP-YFP datasets (right panels). Clearly, the bi-exponential decay of CFP is significantly faster in the CFP-YFP datasets as compared to in the CFP-only datasets. We conclude that there is evidence of significant FRET, and that the two tagged proteins are often expressed simultaneously in close proximity.

2.5 Mass spectrometry applications

The experimental techniques of gas chromatography mass spectrometry (GC/MS) and liquid chromatography mass spectrometry (LC/MS) measure the mass spectrum of a complex sample as it elutes from a chromatography column (see e.g.,

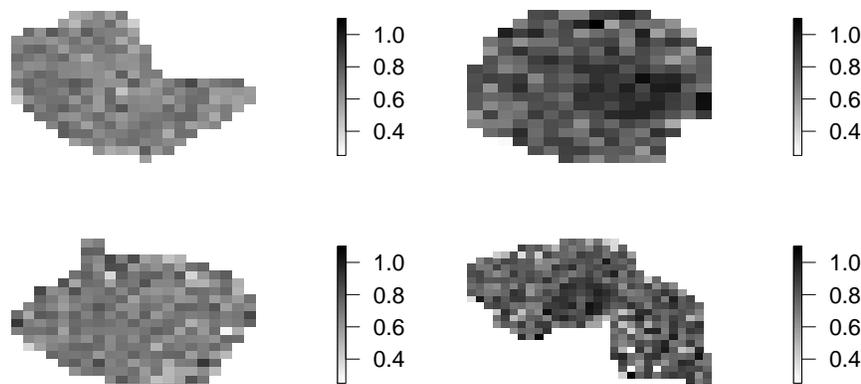


Figure 2.6: Average decay rate per pixel for the cells shown in Figure 2.5 after parameter estimation via global analysis. The higher decay rates in the datasets at right indicate protein-protein interactions.

the book by de Hoffmann and Stroobant (2007) for a discussion of the experimental technique). The mass spectrum of a chemical compound acts as its fingerprint: it allows the compound to be uniquely identified. The time a chemical compound takes to pass through a liquid or gas chromatographer, in contrast, may not be unique. Therefore, if more than one chemical compound elutes at the same time, the mass spectrum at those times represents a mixture of the pure mass spectra of the underlying compounds. Since the purpose of GC/MS and LC/MS experiments is typically to obtain the pure mass spectra of the underlying compounds of a sample in order to perform compound identification and possibly quantification, it is necessary to solve a component resolution problem if compounds are co-eluting.

GC/MS or LC/MS measurements of a single sample can be modeled as $\Psi = CE^T$ where C are the elution profiles of components, and E are the associated mass spectra, which are resolved with respect to mass-to-charge ratio (m/z). For the case that multiple datasets are modeled, the same mass spectra are assumed to be present in all samples $\Psi_1, \Psi_2, \dots, \Psi_K$, though the elution profiles are usually not assumed to be constant, so that Equation 2.14 applies. The inverse problem of estimating C and E , or in the multiple sample case C_1, C_2, \dots, C_K and E from Ψ or $\Psi_1, \Psi_2, \dots, \Psi_K$, respectively, is often tackled with the multivariate curve resolution alternating least squares (MCR-ALS) algorithm, as in, e.g., Tauler *et al.* (1995); Tauler (1995); de Juan and Tauler (2003); Jonsson *et al.* (2005); Garrido *et al.* (2008).

MCR-ALS is a non-parametric algorithm, and the number of free parameters involved in estimating an elution profile $C[, g]$ is $length(C[, g])$. In situations where the overlap of the elution profiles is large, using a parametric description

for $C[,g]$ may be desirable, as Chapters 7 and 8 explore. This reduces the number of free parameters associated with estimating an elution profile dramatically, since $C[,g]$ can often be described well by an exponentially modified Gaussian with only four parameters. The problem of estimating E and the nonlinear parameters z associated with the model for the elution profiles $C(z)$ is an instance of Problem 2.6, which may be addressed with variable projection.

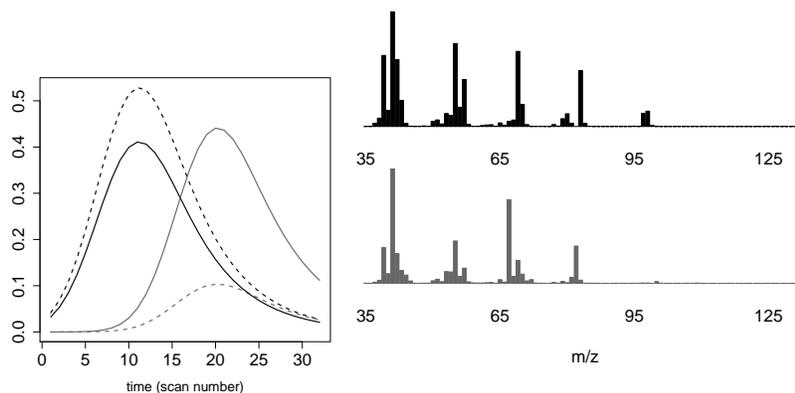


Figure 2.7: (left) Estimated concentration profiles for component 1 (black) and component 2 (grey). Solid lines are for Ψ_1 , while dashed lines are for Ψ_2 , and the profiles differ in the two datasets by amplitude only. (right) Mass spectra associated with component 1 (black) and component 2 (grey), which are assumed to underlie both Ψ_1 and Ψ_2 .

2.5.1 Example: component resolution in GC/MS data

We consider briefly GC/MS measurements Ψ_1 and Ψ_2 of two samples known to represent the same unknown compounds. We model the data using Equation 2.14. Each column $C_j[,g](z)$ is represented by an exponentially modified Gaussian (EMG) function with four parameters, for width, location, decay rate, and amplitude. The elution profiles $C_{j_1}[,g]$ and $C_{j_2}[,g]$ for component g in the two datasets at mass j are described with the same parameter values except for the amplitude, (because the elution profiles in these datasets are aligned; for datasets in which the locations of components are not aligned, the location parameter also varies per-dataset). A singular value decomposition of the data yields two singular values that clearly exceed the remaining values on a log scale; we therefore choose to model two components. It is furthermore known that data at 38 - 44 m/z represent a large peak at every timepoint. We therefore assign these m/z values a very low weight, so that the model for $C_j(z)$ is dependent on the m/z value j considered. The full model specification is found on the

webpage of the **TIMP** package. The estimated mass spectra and concentration profiles that result from fitting this model are shown in Figure 2.7. Data and model fit at selected masses are shown in Figure 2.8.

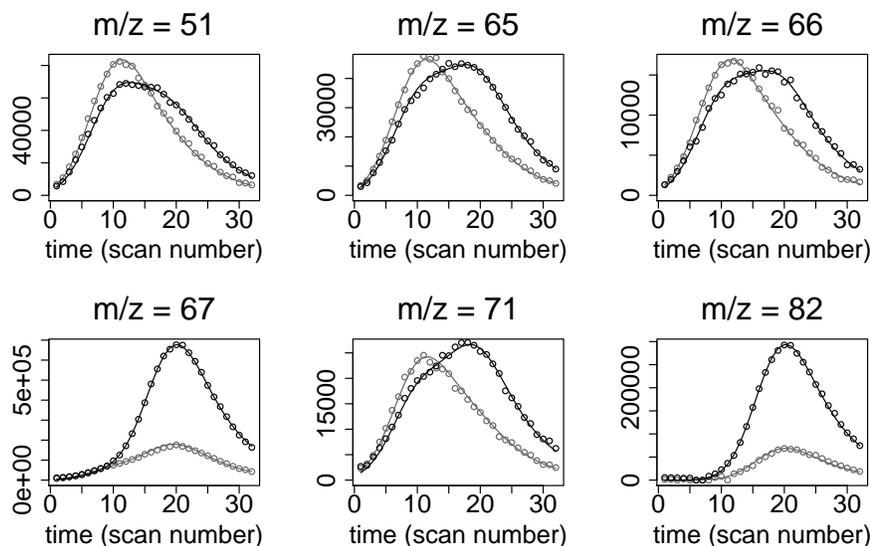


Figure 2.8: Data shown as points at selected mass values from two datasets Ψ_1 (black) and Ψ_2 (grey). The fitted model associated with a simultaneous analysis of both datasets is shown as lines.

2.6 Future work

We see several possibilities for further development of algorithms for separable nonlinear optimization problems. It would be of interest to develop methodology and software for separable optimization problems in which the noise is described by a non-Gaussian distribution. Of most practical importance would be the generalization to the Poisson noise case.

It would also be desirable to develop variable projection implementations to regularize ill-conditioned or over-parametrized models. Golub and Pereyra (2003) suggest that this be accomplished via use of a truncated singular value decomposition method to obtain the Jacobian. Recent work by Shen and Ypma (2007) is in this direction. However, to our knowledge no public domain general purpose optimization routines for the case that $C(z)$ is rank deficient have been made available.

2.7 Conclusions

We have surveyed the variable projection algorithm and its implementation with emphasis on matrix-data fitting problems. Methodology for standard error estimation was discussed, along with extensions to constrain the linear parameters.

The variable projection algorithm has been an important tool for modeling time-resolved spectroscopy data for many years, and we outlined some of the most influential work in this domain. Implementations allowing the flexible specification of models (e.g., the R package **TIMP**) are contributing to the spread of the use of the algorithm, and will continue to do so as such software further matures.

Two new application domains in which variable projection is making an impact are time-resolved fluorescence microscopy and GC/MS mass spectrometry data analysis. Datasets associated with microscopy and mass spectrometry techniques are often large, and more heuristic methods for data analysis have previously been popular largely to avoid the prohibitive execution time and memory needed for model-based analysis. Improvements in computer hardware are changing this and making model-based analysis more feasible. Here again the variable projection algorithm is posed to play a central role, since the parameter estimation task associated with many models applicable to microscopy and mass spectrometry data consists of fitting a linear combination of nonlinear functions.

Acknowledgements

We thank Bas Gobets and Rienk van Grondelle for providing the data in Section 2.3.1, Sergey Laptinok and Jan Willem Borst for the data in Section 2.4.1, and Yury Tikunov and Harrie A. Verhoeven for the data in Section 2.5.1.

Chapter 3

Algorithms for separable nonlinear least squares with application to modeling time-resolved spectra¹

3.1 Introduction

State-of-the-art dynamical experiments in photophysics result in datasets that represent a spectral property associated with a photoactive system at m times and n wavelengths by an $m \times n$ matrix Ψ . For typical experiments, m and n are of order 10^3 . A model-based analysis is mandatory for interactive validation of hypotheses regarding physicochemical mechanisms of the underlying system. The basic kinetic model applied to Ψ is

$$\Psi = CE^T + \Xi = \sum_{l=1}^{n_{comp}} c_l e_l^T + \Xi = \sum_{l=1}^{n_{comp}} \exp(-\phi_l t) e_l^T + \Xi \quad (3.1)$$

where column l of C represents the concentration in time of a spectrally distinct subsystem contributing a component to Ψ , column l of E describes the spectrum of that subsystem, n_{comp} is the number of contributing components, and Ξ is a matrix of residuals, the entries of which are Gaussian random variables with mean zero and constant standard deviation. Elements of Ψ , C , E and Ξ are in \mathbb{R} . Estimation of parameters ϕ under least-squares criteria is thus a multi-exponential analysis problem, the difficulty of which is well-known (Seber and Wild, 2003; Basu and Bresler, 2000). Problems in multi-exponential analysis

¹A version of this chapter appears as Mullen, Vengris, and van Stokkum (2007) in the *Journal of Global Optimization*, 38(2), 201-213.

are ubiquitous in physics applications in which data is modeled by the solution of first-order differential equations, as (Istratov and Vyvenko, 1999) review.

The estimation problem associated with estimating ϕ and E in Model (3.1) under least-squares criteria is

$$\text{Minimize } \| \text{vec}(C(\phi)E^T - \Psi) \|^2, \quad (3.2)$$

where, as throughout $\| \cdot \|$ is the 2-norm. The bilinear structure of Equation 3.2 means that it is a *separable* problem. An unconstrained optimization problem $\text{Minimize } \gamma(x)$, $x \in \mathbb{R}^n$ is separable if the variables separate into $x = (y, z)$ with $y \in \mathbb{R}^p$, $z \in \mathbb{R}^q$, $p + q = n$, and the subproblem

$$\text{Minimize } \gamma(y, z), \quad (3.3)$$

is easy to solve for fixed z . Such problems are connected to bilinear programming (Al-Khayyal, 1990; Brimberg *et al.*, 2002; Horst and Tuy, 1996). Separating the parameters reduces the n -dimensional unconstrained optimization problem to the q -dimensional unconstrained problem

$$\text{Minimize } \gamma(y(z), z), \quad (3.4)$$

where $y(z)$ denotes a solution of (3.3). In the considered application $y(z)$ is solved as the solution of a linear-least squares problem for fixed z , there are hundreds more conditionally linear parameters y than intrinsically nonlinear parameters z , and linear approximation standard error estimates about estimates for z are desired for model validation. These structural features of the problem and the requirement for standard error estimates make gradient-based algorithms that exploit the conditional linearity of Problem (3.2) attractive, though a variety of other algorithms, e.g., Branch and Cut methods (Audet *et al.*, 2000), evolutionary search (Wohlleben *et al.*, 2003; Fisz, 2006), or Prony-based methods (Osborne and Smyth, 1995) are also applicable. The development of gradient-based methods for the separable Problem (3.4) is chronicled in, e.g., Golub and Pereyra (2003); Parks (1985); Smyth (1996). The gradient-based algorithms most commonly applied to Problem (3.2) are based on alternating least squares (Bijlsma *et al.*, 2000, 2002; Jandanklang *et al.*, 2001; Dioumaev, 1997) or variable projection (Golub and LeVeque, 1979; Nagle, 1991b; van Stokkum *et al.*, 2004). These techniques have been numerically compared by Bates and Lindstrom (1986) for a single nonlinear parameter, and by Gay and Kaufman (1991) for small datasets (< 70 datapoints). Theoretical comparisons of gradient-based methods for separable problems have been made by Bates and Lindstrom (1986); Böckmann (1995); Kaufman (1975); Parks (1985); Ruhe and Wedin (1980). In this chapter we extend the literature comparing gradient-based methods for separable nonlinear optimization problems to Problem (3.2), the central estimation problem in fitting parametric kinetic models to time-resolved spectra.

A comparison of techniques in the photophysical modeling application domain is desirable due to the difficulty of Problem (3.2), which is not identifiable (Varah, 1985) and sensitive to starting values (Pettersson and Holmström, 1997;

van den Bos, 2007). Convergence issues due to ill-conditioning when two or more decay rate parameters ϕ_l are close are well-known (Osborne and Smyth, 1995; Petersson and Holmström, 1998). The stochastic noise term contained in measured Ψ introduces a further source of difficulty by complicating the sum of squared errors parameter surface of ϕ with local minima. The performance of alternating least squares and variable projection variants is studied here in such a way as to expose the vulnerabilities and strengths of the algorithms in the face of these difficulties as they occur in typical photophysical model fitting problems. To the best of our knowledge this is the first such comparison in the literature.

Alternating least squares and variable projection variants are presented in Section 3.2. The ability of the algorithms to deal with degeneracy in the case of similar decay rate parameters ϕ_l, ϕ_j is studied theoretically in Section 3.3 by comparison of Fisher information matrices (FIM) associated with parameter estimates under variable projection variants. This section contains a new derivation of the FIM under the full Golub-Pereyra variable projection functional. Section 3.4 discusses the simulation of realistic datasets of time-resolved spectra to be used in numerical comparison. A numerical study is made in Section 3.5 to highlight convergence issues and sensitivity to starting values. Section 3.5.2 contains a numerical comparison of variable projection techniques using FIMs as rate constants vary in such a way to make Problem (3.2) more nearly-degenerate.

3.2 Gradient-based algorithms for separable non-linear least squares

Gradient-based algorithms for solution of Problem (3.2) estimate E as $\hat{E}^T(\phi) = C^+\Psi$ where $+$ is the Moore-Penrose pseudoinverse, so that Problem (3.2) may be written as

$$\text{Minimize } \| (I - C(\phi)C^+(\phi))\Psi \|^2 . \quad (3.5)$$

The gradient-based techniques most often applied to Problem (3.5) are based on the Jacobian of either the alternating least squares (ALS) or variable projection functionals. ALS was introduced by Wold and Lyttkens (1969) as NIPALS and has a simple functional form which neglects the derivative of the pseudoinverse C^+ in an approximation of J . The analytical variable projection Jacobian (GP) makes use of the derivative of C^+ due to Golub and Pereyra (1972, 1973). The analytical approximation to the Jacobian given by GP introduced by Kaufman (1975) (KAUF) is more efficient to compute and for simple models has been shown to return nearly as precise parameter estimates as the full functional (Bates and Lindstrom, 1986; Gay and Kaufman, 1991).

In order to make clear the core differences between algorithms, we present ALS, KAUF and GP and a finite difference approximation of $(I - C(\phi)C^+(\phi))\Psi$ in terms of the Jacobian of the residuals $(I - C(\phi)C^+(\phi))\Psi$ with respect to the nonlinear parameters ϕ , using the notation of Bates and Lindstrom (1986). The

derivative of C with respect to the nonlinear parameters is denoted

$$C_\phi = \frac{dC}{d\phi^T} \quad (3.6)$$

so that for parameter ϕ_i it is $\frac{\partial C}{\partial \phi_i}$. Applying the QR decomposition, $C = QR = [Q_1 \ Q_2]R$, where Q_1 is $m \times n_{\text{comp}}$, Q_2 is $m \times (m - n_{\text{comp}})$, Q is orthogonal, and R is $m \times n_{\text{comp}}$, $R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}$, with R_{11} being $n_{\text{comp}} \times n_{\text{comp}}$ and upper triangular (Golub and van Loan, 1996). Assuming C is of full column rank, $C^+ = R_{11}^{-1}Q_1^T$. Then, where ‘‘convergence’’ is some appropriate stopping criterion and the iteration subscript s is suppressed, we have

ALGORITHMS ALS, KAUF, GP, NUM:

1. Choose starting ϕ approximately
2. For $s := 1, 2, \dots$ until convergence do
 - Calculate the residual vector as $res = (I - C(\phi)C^+(\phi))\Psi$
 - Calculate $J = \frac{dres}{d\phi}$ using one of the following prescriptions:
 - $J_{NUM} :=$ finite difference approximation of $\frac{d(I - CC^+)}{d\phi}\Psi$
 - $J_{GP} := Q_2Q_2^TC_\phi C^+\Psi - Q_1R_{11}^{-T}C_\phi^TQ_2Q_2^T\Psi$
 - $J_{KAUF} := Q_2Q_2^TC_\phi C^+\Psi$
 - $J_{ALS} := C_\phi C^+\Psi$
 - $\phi_{s+1} :=$ STEP, DIRECTION(ϕ_s, res, J, \dots)

The sub-routine *step, direction* that determines the direction and the step-sizes to move ϕ as a function of ϕ_s, res , and J is found in standard algorithms for nonlinear least squares such as Levenberg-Marquardt or Gauss-Newton (Bates and Watts, 1988). Here the step and direction are calculated with Gauss-Newton. Simulation studies indicate that for the numerical problems considered in Section 3.5, replacement of the Gauss-Newton STEP, DIRECTION with the Levenberg-Marquardt STEP, DIRECTION does not appreciably alter the performance of any of the algorithms.

For a numerical comparison we also consider varieties of ALS differing in the STEP method. The first (ALS-GN) makes a step in the direction determined by Gauss-Newton given J_{ALS} . The second (ALS-LS) makes a step in the direction determined by Gauss-Newton as well, but augments this step by a line search until the sum of squared errors (SSE) is seen to increase.

Implementation is straightforward using library subroutines for QR decomposition, finite difference derivatives, and nonlinear least squares. Such subroutines are found, for instance, in the *base* and *stats* packages of the R language and environment for statistical computing (R Development Core Team, 2008), where we base the implementation for numerical comparison. An analytical expression for C_ϕ is used for models based on a sum-of-exponentials. Under more complicated models for C a finite difference approximation of C_ϕ is often desirable.

We now summarize some prior results comparing subsets of the algorithms under consideration. Ruhe and Wedin (1980) have shown that for starting ϕ close to the solution, the asymptotic convergence rates of KAUF and GP are superlinear whenever application of Gauss-Newton to the unseparated parameter set $(\phi + E)$ has a superlinear rate of convergence, and that ALS always has only a linear rate of convergence. Bates and Lindstrom (1986) demonstrated that for a simple model having a single nonlinear parameter the performance of KAUF and GP was similar. Gay and Kaufman (1991) also performed a comparison of KAUF and GP on several small datasets, (< 70 data points), demonstrating that the time to compute KAUF was about 25% less than the time to compute GP for the range of problems considered.

3.3 Parameter precision under variable projection variants

When optimal estimates for two or more nonlinear parameters in a sum-of-exponential decays are close, so that the data are well-approximated by a lower-order sum-of-exponentials, Problem (3.2) is termed nearly-degenerate. In such situations the precision of nonlinear parameter estimates determines whether all exponential decays can be resolved. Section 3.4.1 further elaborates the importance of parameter precision in solving nearly-degenerate problems.

A means of quantifying the precision of a vector of parameter estimates is found in the FIM. The structure of the FIM provides insight into contributions to parameter precision, and FIMs may be numerically compared under different Jacobians, as in Section 3.5.2. The resolution limit of exponential analysis has been often studied in terms of FIMs and other information-theoretic metrics, as discussed in Istratov and Vyvenko (1999). Basu and Bresler (2000) have studied the connection between the stochastic stability of nonlinear least squares problems and the FIM with attention to separable problems such as Problem (3.2).

Definition 3.3.1: Define J as the Jacobian of the residual function with respect to the nonlinear parameters ϕ . Assume the model error σ^2 is estimated as $\sigma^2 = SSE(\phi)/df$, where df is the degrees of freedom of the model. Furthermore, assume as throughout, the noise Ξ is additive, with entries comprised of independent Gaussian random variables with mean zero and constant standard deviation. Then the FIM M may be defined as

$$M = \sigma^{-2} \text{vec}(J)^T \text{vec}(J) = \sigma^{-2} \tilde{M}. \quad (3.7)$$

When M is positive definite the covariance estimate of any unbiased estimator of parameter vector ϕ is bounded below by the inverse of M (the Cramér-Rao Bound), so that

$$\text{Cov}[\hat{\phi}] \geq M^{-1}. \quad (3.8)$$

We will now give functions for \tilde{M} under the variable projection algorithms *KAUF* and *GP*.

Proposition 3.3.1.

$$\tilde{M}_{KAUF} = \text{vec}(C_\phi)^T (E^T E \otimes P) \text{vec}(C_\phi). \quad (3.9)$$

Proof. J_{KAUF} is given as

$$J_{KAUF} = Q_2 Q_2^T C_\phi C^+ \Psi = P C_\phi E^T, \quad (3.10)$$

where $P = Q_2 Q_2^T$.

Writing J_{KAUF} in vectorized form,

$$\text{vec}(J_{KAUF}) = \text{vec}(P C_\phi E^T) \quad (3.11)$$

$$= (E \otimes P) \text{vec}(C_\phi). \quad (3.12)$$

Then from van Stokkum (1997),

$$\tilde{M}_{KAUF} = \text{vec}(J_{KAUF})^T \text{vec}(J_{KAUF}) \quad (3.13)$$

$$= ((E \otimes P) \text{vec}(C_\phi))^T ((E \otimes P) \text{vec}(C_\phi)) \quad (3.14)$$

$$= \text{vec}(C_\phi)^T (E^T E \otimes P) \text{vec}(C_\phi). \quad (3.15)$$

□

It is often convenient to consider \tilde{M} by entry \tilde{M}_{ij} . This is

$$(\tilde{M}_{KAUF})_{ij} = \text{vec}(C_{\phi_i})^T E^T E \otimes P \text{vec}(C_{\phi_j}), \quad (3.16)$$

where $\text{vec}(C_{\phi_i})$ is the vector representation of $\frac{dC}{d\phi_i}$.

For a two column matrix C in which $c_l = \exp(-t\phi_l)$, $\text{vec}(C_{\phi_1}) = \begin{pmatrix} g_1 \\ 0 \end{pmatrix}$

and $\text{vec}(C_{\phi_2}) = \begin{pmatrix} 0 \\ g_2 \end{pmatrix}$, where $g_i = -t \exp(-\phi_i t)$. For this case the expression for \tilde{M}_{KAUF} simplifies to

$$(\tilde{M}_{KAUF})_{ij} = e_i^T e_j g_i^T P g_j. \quad (3.17)$$

Proposition 3.3.2. *Writing \tilde{M}_{GP} per entry,*

$$(\tilde{M}_{GP})_{ij} = (\tilde{M}_{KAUF})_{ij} + \text{vec}(C_{\phi_i}^T)^T (P\Psi)(P\Psi)^T \otimes C^+ (C^+)^T \text{vec}(C_{\phi_j}^T). \quad (3.18)$$

Proof. The Jacobian J_{GP} of the residuals with respect to the nonlinear parameters contains the extra term $Q_1 R_{11}^{-T} C_\phi^T Q_2 Q_2^T \Psi$ as compared to J_{KAUF} , so that

$$J_{GP} = Q_2 Q_2^T C_\phi C^+ \Psi + Q_1 R_{11}^{-T} C_\phi^T Q_2 Q_2^T \Psi \quad (3.19)$$

$$= J_{KAUF} + (C^+)^T C_\phi^T P \Psi. \quad (3.20)$$

Vectorizing J_{GP} ,

$$\text{vec}(J_{GP}) = (E \otimes P) \text{vec}(C_\phi) + (P\Psi)^T \otimes (C^+)^T \text{vec}(C_\phi^T), \quad (3.21)$$

and

$$\text{vec}(J_{GP})^T = \text{vec}(C_\phi)^T(E^T \otimes P) + \text{vec}(C_\phi^T)^T(P\Psi) \otimes C^+. \quad (3.22)$$

Then, writing \tilde{M}_{GP} per entry,

$$(\tilde{M}_{GP})_{ij} = (\tilde{M}_{KAUF})_{ij} + \text{vec}(C_{\phi_i}^T)^T(P\Psi)(P\Psi)^T \otimes C^+(C^+)^T \text{vec}(C_{\phi_j}^T). \quad (3.23)$$

where we have used the orthogonality of J_{KAUF} and $(C^+)^T C_\phi P\Psi$. \square

For a two column matrix C in which $c_l = \exp(-t\phi_l)$, the expression for \tilde{M}_{GP} simplifies to

$$(\tilde{M}_{GP})_{ij} = (\tilde{M}_{KAUF})_{ij} + g_i^T P\Psi(P\Psi)^T g_j (R_{11}^T R_{11})_{ij}^{-1}. \quad (3.24)$$

The extra term in \tilde{M}_{GP} as compared to \tilde{M}_{KAUF} is associated with the more accurate representation of the Hessian of Problem (3.2) under J_{GP} as compared to under J_{KAUF} . The extent to which this extra term is of benefit in solving Problem (3.2) in practice is evaluated numerically in Section 3.5.2.

3.4 Data for a simulation study

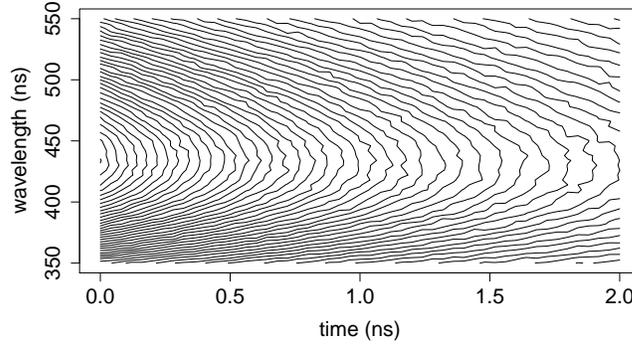


Figure 3.1: Contour map of typical simulated data Ψ used in computational study. Model fitting will resolve the two contributing components.

For a simulation study we used a model giving rise to a multi-exponential analysis problem involving two exponentials with rate constant parameters $\phi = \{k_1, k_2\}$. The generative model for the matrix C of concentrations is then $c_l = \exp(-k_l t)$, where t is a vector of times and $n_{comp} = 2$.

The spectra E associated with the exponential decays are modeled as a mixture of Gaussians in the wavenumber $\bar{\nu}$ (reciprocal of wavelength) domain, so that

$$e_l(\mu_{\bar{\nu}}, \Delta_{\bar{\nu}}) = a_l \bar{\nu}^5 \exp(-\ln(2)(2(\bar{\nu} - \mu_{\bar{\nu}})/\Delta_{\bar{\nu}})^2), \quad (3.25)$$

where e_l is column l of E describing the l th spectrum, with parameters $\mu_{\bar{\nu}}$, $\Delta_{\bar{\nu}}$, and a_l , for the location, full width at half maximum (FWHM), and amplitude, respectively. This underlying model for E is chosen because it is a simple model capable of representing real spectra in practice (van Stokkum, 1997), and because the use of Gaussians to represent spectral shapes is wide-spread, (as, e.g., van Stokkum *et al.* (2004) and references therein describe). The algorithms presented in Section 3.2 to solve Problem (3.2) treat the entries of E as conditionally linear parameters so that the spectral shapes are recoverable without specification of an underlying parametric model. This is often desirable because the set of parameters necessary to adequately describe the spectra of photophysical systems of interest is often large and more difficult to determine in comparison to the small and relatively simple parametrization ϕ of the concentrations C .

component	k	$\mu_{\bar{\nu}}$	$\Delta_{\bar{\nu}}$	a
1	.5	22	9	1
2	.6	18	8	2

Table 3.1: Rate constants, spectral parameters (in 10^3 cm^{-1}), and amplitudes for simulated Ψ

Given these models for C and E , data was generated with the parameter values in Table 3.1. Values for kinetic parameters k_1 and k_2 are similar and the spectral parameters represent overlapping spectral shapes. $n = 51$ time points equidistant in the interval 0-2 ns and $m = 51$ wavelengths equidistant in the interval 350-550 nm. These parameter values are inspired by real data (van Stokkum, 1997).

3.4.1 Degeneracy and multimodality due to noise

Measured time-resolved spectra Ψ always contain stochastic noise. The presence of noise may introduce stationary points where $\frac{dJ(\phi)}{d\phi} = 0$ at ϕ distinct from those values underlying the deterministic model, so that the algorithms presented in Section 3.2 are sensitive to starting values. This numerical identifiability problem is well-known in kinetic modeling (Godfrey, 1983). In the case of convergence to a local minimum introduced by noise, estimates for kinetic parameters and spectra are often implausible from physicochemical first principles. Uninterpretable parameter estimates typically allow spurious solutions to be recognized and discarded.

In fitting Model (3.1) to measured time-resolved spectra the signal-to-noise ratio may be such that degeneracy is a significant issue. That is, optimal estimates for two or more rate constant parameters in the vector of nonlinear parameters ϕ may be close enough that noise disrupts the SSE surface in such a way that the globally optimal solution is a sum of less than n_{comp} exponentials, as reviewed in van den Bos and Swarte (1993). Then the least-squares solution yields estimates with $k_1 = k_2$ for $\{k_1, k_2\} \in \phi$. In nearly-degenerate cases the least squares solution is with $k_1 \approx k_2$, and the parameters may be resolved if the

precision with which they are estimated is sufficiently high, as is studied numerically under the KAUF and GP algorithms in Section 3.5.2. For the simulated dataset described in Section 3.4 degeneracy is probable for noise with standard deviation of about 7×10^{-3} the maximal value in the deterministic data. The SSE surface of parameters ϕ for a noise realization that results in degeneracy is shown in Figure 3.2.

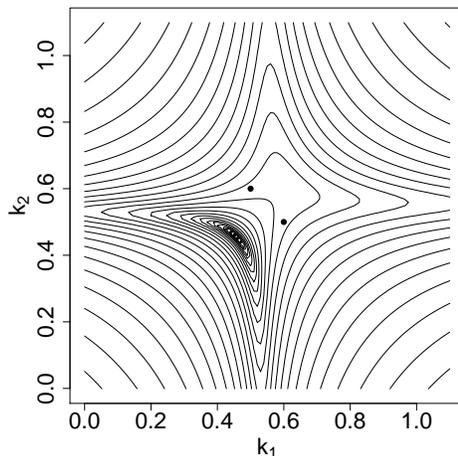


Figure 3.2: The SSE surface in parameter space for the contour plot of the dataset described in Section 3.4 with a stochastic noise term with Gaussian distribution and zero mean having standard deviation Δ equal to 7×10^{-3} the maximum of the deterministic dataset. The parameter values $\phi = \{.5, .6\}$ or symmetrically $\phi = \{.6, .5\}$ (closed circles) underlie the deterministic part of the data, and would be the globally optimal parameter estimates except for the effect of noise, which makes the lower order solution $\phi = \{.45, .45\}$ globally optimal.

3.5 Computational results

Model (3.1) was fit to the data described in Section 3.4 with a stochastic noise term with Gaussian distribution and zero mean having standard deviation Δ equal to 3×10^{-3} the maximum of the deterministic dataset using each of the algorithms described in Section 3.2. The convergence criterion was reduction of sum of squared errors (SSE) $\|vec(\Psi - CE^T)\|^2$ by a factor of less than $1/2^{10}$ between iterations. Estimated spectra found as conditionally linear parameters under KAUF, GP, ALS-LS or NUM well-represent the spectra used in generating the simulated data, as shown in Figure 3.3.

To visualize the progress of the algorithms per iteration, the SSE as a function of the rate constants k_1, k_2 is evaluated, with the result being the surface

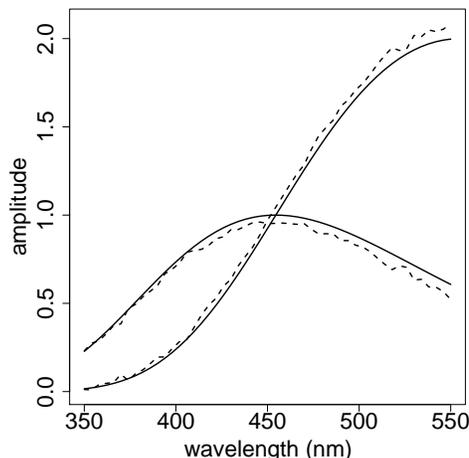


Figure 3.3: Estimated spectra (dashed lines) as found with KAUF, GP or NUM by fitting the simulated dataset depicted in Figure 3.1 with the two-component kinetic model described in Section 3.5. Spectra used to generate the deterministic part of the dataset (solid lines) are shown for comparison.

shown in Figure 3.4. Figure 3.4 also shows the values found by each algorithm under consideration for each of 50 iterations from the starting values $k_1 = .1, k_2 = 1$. KAUF, GP, ALS-LS and NUM converge to the same (globally optimal) solution in 4 iterations. ALS-GN does not generally converge after many hundreds of iterations, and from this case study and others we conclude that the Gauss-Newton step coupled with the Jacobian calculated under ALS is not sufficient for the solution of typical estimation problems in this domain.

Performance from a range of starting values and on variants of the dataset under different noise realizations was examined. For cases in which globally optimal parameter values are located at the end of a valley on the SSE surface with respect to the starting values, the performance of ALS-LS is very much hampered in terms of iterations required to convergence in comparison to KAUF, GP, and NUM. A plot of the SSE surface in this case shows that ALS-LS follows a zig-zagging path between the walls of the valley toward a globally optimal solution.

We conclude that ALS coupled with a line search and both variable projection methods KAUF and GP solve this problem for the considered data realizations. The KAUF algorithm typically requires the same number of iterations as the GP algorithm. ALS with line search converges in a greater or equal number of iterations as compared to KAUF and GP. The iterations required for ALS-LS are greater than for KAUF and GP when the globally optimal parameter values are at the end of a valley in SSE with respect to starting parameter estimates. Therefore in terms of iterations to convergence and sensitivity of computational efficiency to starting values, the variable projection-based algorithms demon-

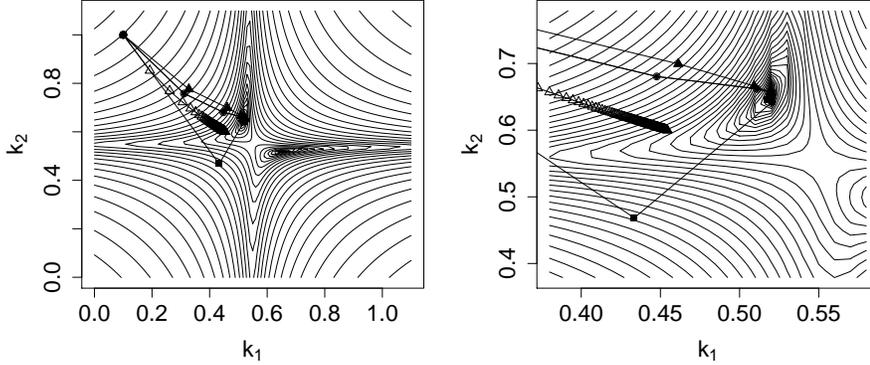


Figure 3.4: Contour map of the sum square of residuals $\|\text{vec}(\Psi - CE^T)\|^2$ as rate constants k_1, k_2 vary, at a relatively large (left panel) and relatively small (right panel) scale. The progress of ALS-GN (unfilled triangle), ALS-LS (square), KAUF (filled triangle), and GP/NUM (filled and unfilled circles) is depicted from starting values $k_1 = .1, k_2 = 1$; rate constant estimates are marked with the symbol associated with each algorithm after each iteration. Spacing between contour lines is not uniform.

strate the best performance.

3.5.1 Standard error estimates

In order to examine the properties of linear approximation standard error estimates as returned by the algorithms under consideration, 1000 realizations of the dataset described in Section 3.4 were simulated. For each realization, the *deviation*(k) = $\|\hat{k} - k\|$, where \hat{k} is the estimated rate constant value, and k is the value used in simulation, the linear approximation standard error ($\hat{\sigma}_{\hat{k}}$), derived from $\text{cov}(\hat{\phi}) = \hat{\zeta}^2(J^T J)^{-1}$, where $\hat{\zeta}^2$ denotes the estimated model variance and J is the Jacobian of the residuals evaluated at $\hat{\phi}$. The studentized parameter deviation is formed as the ratio (*deviation*/ $\hat{\sigma}_{\hat{k}}$) (Bates and Watts, 1988; Seber and Wild, 2003; van Stokkum, 1997). Table 3.2 reports root mean square (RMS) results for these quantities over 1000 noise realizations.

At the level of precision collated in Table 3.2, results for NUM, KAUF and GP are identical. NUM and GP only differ from KAUF in the 3rd decimal place of RMS (*deviation*/ $\hat{\sigma}_{\hat{k}}$), and from each other in the 6th.

RMS (*deviation*/ $\hat{\sigma}_{\hat{k}}$) is expected to be 1 in linear models, and hence the degree to which this ratio approximates 1 can be used as a measure of the applicability of the linear approximation standard error returned by the respective algorithms. Under ALS, $\hat{\sigma}_{\hat{k}}$ is much too small, and not useful as a measure of confidence in parameter value estimates.

		ALS-LS	KAUF/GP/NUM
RMS deviation ($\hat{k} - k$)	k_1	0.022	0.022
	k_2	0.025	0.025
RMS $\hat{\sigma}_{\hat{k}}$	k_1	0.00033	0.021
	k_2	0.00048	0.027
RMS (deviation/ $\hat{\sigma}_{\hat{k}}$)	k_1	55	1.3
	k_2	37	1.2

Table 3.2: Root mean square deviation and standard error of nonlinear parameters after fitting a two-component sum-of-exponentials model (Model 3.1) to 1000 realizations of the dataset described in Section 3.4.

Likelihood-based confidence regions may be constructed around parameter estimates based on the likelihood ratio between the sum of squared residuals $S(\hat{\phi}) = \|\text{vec}(\Psi - CE^T)\|^2$ at the solution and at values $S(\phi)$ around the solution as $\phi = \{k_1, k_2\}$ is varied. The confidence level $1 - \alpha$ is estimated as

$$1 - \alpha = F \left(P, N - P, (N - P)/P \frac{S(\phi) - S(\hat{\phi})}{S(\hat{\phi})} \right) \quad (3.26)$$

where F is the cumulative F -distribution, $P = n_{comp} = 2$, and N is the degrees of freedom in the model, which here is with $N = (times - n_{comp})(wavelengths) = (51 - 2)(51)$ (Bates and Watts, 1988; Seber and Wild, 2003). The resulting contour plot of confidence regions about the parameter estimates is shown in Figure 3.5(a). For comparison, the linear approximation confidence regions calculated from $cov(\phi)$ for KAUF, GP, or NUM are shown in Figure 3.5(b). Note that the linear approximation confidence regions are slightly too small as compared to the likelihood-based confidence regions, which is consistent with the slight underestimation of $\hat{\sigma}_{\hat{k}}$ in Table 3.2, as measured by the overshoot of deviation/ $\hat{\sigma}_{\hat{k}}$ to 1.

In conclusion, the standard error estimates returned by both variable projection variants are usable as a measure of confidence in the associated parameter estimates, and allow, e.g., the construction of confidence regions about parameter estimates. The standard error estimates returned by ALS with line search are so poor as to prohibit inference regarding the associated parameter estimates. Hence the variable projection-based algorithms also demonstrate better performance relative to ALS-based algorithms under the criteria of goodness of standard error estimates.

3.5.2 Numerical comparison of Fisher information matrices

The functional forms for the FIM are useful in accessing the loss of parameter precision under KAUF as compared to GP for typical problems. Relation (3.8) allows standard error bounds under both algorithms to be numerically

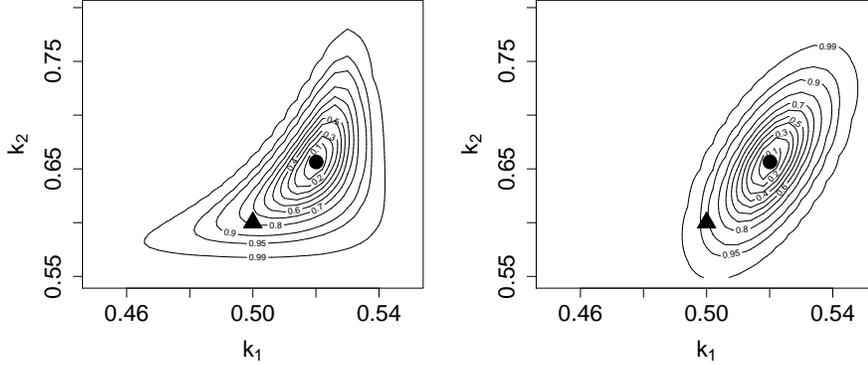


Figure 3.5: For the dataset depicted in Figure 3.4, (left panel) contour map of confidence levels $1 - \alpha$ as estimated by Equation 3.26 as rate constants k_1, k_2 vary, (right panel) linear approximation confidence regions as found using KAUF, GP, or NUM for the same levels as at left. In both panels a triangle marks the rate constant values used in simulation, and a circle marks the globally optimal values found by KAUF, GP, NUM, and ALS-LS.

compared. This comparison is of particular interest for estimation problems associated with a SSE surface of the nonlinear parameters ϕ with multiple closely spaced global minima.

For fitting Model (3.1) to the dataset described in Section 3.4 realized with a noise distribution having standard deviation 1×10^{-4} the maximum of the deterministic dataset, we studied the standard error bounds returned by KAUF and GP using Relation (3.8). We varied the separation between rate constants $k_2 - k_1$ by letting $k_1 = .5$ and varying k_2 between 1 and .5075. The standard error bounds under KAUF never increased by more than 5×10^{-4} percent in comparison to the bounds under GP, even when the separation $k_2 - k_1$ became very small. Hence the decrease in parameter precision under KAUF as compared to under GP is negligible even for nearly-degenerate instances of Problem (6.4). Since KAUF is faster to compute it may therefore be preferred for application.

3.6 Conclusions

Gradient-based algorithms for separable nonlinear least squares based on alternating least squares and variable projection were compared for an application in multi-exponential analysis that is common and important in fitting photophysical kinetic models to time-resolved spectra. The number of iterations required by the variable projection algorithms was found to be less sensitive to starting values as compared to the algorithms based on alternating least squares. The

linear approximation confidence regions about parameter estimates using variable projection variants were furthermore found to well-approximate likelihood-based confidence regions, while those based on an alternating least squares did not. Using a new derivation of the Fisher information matrix under the Golub-Pereyra variable projection algorithm, parameter precision under variable projection techniques was compared numerically. The loss of precision under the Kaufman approximation as compared to the Golub-Pereyra variable projection functional was found to be acceptable even on nearly-degenerate problems, so that the faster Kaufman approximation algorithm can be recommended for application to the problem in photophysical modeling considered here.

Chapter 4

Separable nonlinear least squares with constraints on the conditionally linear parameters¹

4.1 Introduction

Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} \gamma(x) \quad (4.1)$$

where γ is a nonlinear function of x . If Problem 4.1 is *separable*, the parameters $x \in \mathbb{R}^n$ separate into $x = (y, z)$ with $y \in \mathbb{R}^p$, $z \in \mathbb{R}^q$, $p + q = n$ such that the sub-problem

$$\min_{y \in \mathbb{R}^p} \gamma(y, z) \quad (4.2)$$

is *easy to solve* for fixed z . *Easy to solve* means that y can be treated as *dependent* on z , so that y is a function of z , which we write $y(z)$, and Problem 4.1 can be transformed to

$$\min_{z \in \mathbb{R}^q} \gamma(y(z), z) \quad (4.3)$$

with some advantage (such as speed, better conditioning, or not having to provide starting values for y) as compared to optimization with respect to x directly as in Problem 4.1.

The problem of fitting a linear combination of nonlinear functions to data

¹A version of this chapter is submitted as Mullen and van Stokkum (2008c).

under least squares criteria, that is,

$$\min_{z \in \mathbb{R}^q} \left\| \psi - \sum_{i=1}^{n_{\text{comp}}} c_i(z) a[i] \right\|^2 \quad (4.4)$$

where ψ is an m -vector of data, c_i is a nonlinear function of z that outputs an m -vector, $a[i] \in y$ for $i = 1, \dots, n_{\text{comp}}$ and, as throughout $\| \cdot \|$ is the 2-norm is common in applications (Golub and Pereyra, 2003). We term $a[i]$ *conditionally linear parameters* since $a[i]$ that are optimal under least squares criteria may be obtained by solving a linear least squares problem given fixed z . Problem 4.4 can be written in matrix notation as

$$\min_{z \in \mathbb{R}^q} \left\| \psi - C(z)a \right\|^2 \quad (4.5)$$

where column i of C corresponds to $c_i(z)$, and $a = a[1], a[2], \dots, a[n_{\text{comp}}]$. If the same n_{comp} nonlinear functions $c_i(z)$ may be used to model n measured vectors ψ_j but the amplitude $a_j[i]$ with which each function $c_i(z)$ contributes to ψ_j must be estimated separately for $j = 1, \dots, n$, the optimization problem becomes

$$\min_{z \in \mathbb{R}^q} \left\| \Psi - C(z)E^T \right\|^2 \quad (4.6)$$

where Ψ is an $m \times n$ matrix of data, with column j storing ψ_j , and E is an $n \times n_{\text{comp}}$ matrix with row j storing $a_j = a_j[1], a_j[2], \dots, a_j[n_{\text{comp}}]$.

Attempting to solve Problem 4.6 is termed *global analysis* in the physics and chemistry literature, and is commonly performed in time-resolved spectroscopy (Golub and LeVéque, 1979; Nagle, 1991b) and fluorescence lifetime image measurement (FLIM) (Verveer *et al.*, 2000) data analysis. Chapters 5 and 6 describe global analysis of spectroscopy and FLIM measurements in detail. For physics and chemistry applications, c_i is very often based on a sum of distributions in the exponential family. Since many physical phenomena are described by first-order differential equations whose solution is an exponential decay function, c_i representing exponential decays are often used, as reviewed by Istratov and Vyvenko (1999), and discussed in the context of modeling time-resolved spectroscopy data in van Stokkum *et al.* (2004).

In certain situations, it is desirable to apply constraints to the conditionally linear parameters, leading, for instance, to the problem

$$\min_{z \in \mathbb{R}^q} \left\| \sum_{j=1}^n (\psi_j - \sum_{i=1}^{n_{\text{comp}}} a_j[i] c_i(z)) \right\|^2 \quad (4.7)$$

subject to $0 \leq a_j[i]$ for $i = 1, 2, \dots, n_{\text{comp}}$ and $j = 1, 2, \dots, n$

and more generally

$$\min_{z \in \mathbb{R}^q} \left\| \sum_{j=1}^n (\psi_j - \sum_{i=1}^{n_{\text{comp}}} a_j[i] c_i(z)) \right\|^2 \quad (4.8)$$

subject to $l_j[i] \leq a_j[i] \leq u_j[i]$ for $i = 1, 2, \dots, n_{\text{comp}}$ and $j = 1, 2, \dots, n$

which are represented in matrix notation as

$$\begin{aligned} & \min_{z \in \mathbb{R}^q} \| \Psi - C(z)E^T \|^2 \\ & \text{subject to } 0 \leq E^T[i, j] \text{ for } i = 1, 2, \dots, n_{\text{comp}} \text{ and } j = 1, 2, \dots, n \end{aligned} \quad (4.9)$$

and

$$\begin{aligned} & \min_{z \in \mathbb{R}^q} \| \Psi - C(z)E^T \|^2 \\ & \text{subject to } l_j[i] \leq E^T[i, j] \leq u_j[i] \text{ for } i = 1, 2, \dots, n_{\text{comp}} \text{ and } j = 1, 2, \dots, n \end{aligned} \quad (4.10)$$

respectively. Problem 4.9 often arises in optical spectroscopy, microscopy and mass spectrometry data analysis, where there are typically tens of intrinsically nonlinear parameters z and thousands of conditionally linear parameters $E^T[i, j]$.

In this chapter, we present a framework for addressing Problem 4.5 and its global analysis counterpart, Problem 4.6, in the case that constraints are to be applied to the linear coefficients a . In practice the constraints will be limited to linear or quadratic relations, since Problem 4.2 should remain easy to solve given fixed values of z . The considered approach elaborates on the gradient-based variable projection algorithm developed by Golub and Pereyra (1972, 1973). The history of the development of variable projection as well as an overview of its application in various fields was given by Golub and Pereyra (2003). Kaufman and Pereyra (1978) considered extensions of variable projection to account for nonlinear equality constraints. Parks (1985) noted that in order to enforce constraints on a , a may be determined as the solution of an arbitrary linear or quadratic programming problem. Sima and Van Huffel (2007) described an algorithm to constrain a to non-negative values, and also noted the possibility of a generalization of their approach to include other linear or quadratic constraints, as will be suggested here. The main contribution of this chapter is to explicitly present a framework for the introduction of constraints on the conditionally linear parameters a into variable projection, and to demonstrate the utility of the approach in the context of global analysis for some problems inspired by the analysis of time-resolved emission spectroscopy data.

While the functions c_i are always differentiable in applications of interest to us, so that gradient-based algorithms like variable projection are attractive, in the general case Problem 4.3 may be addressed by algorithms for the bilinear programming problem, as reviewed by Al-Khayyal (1990).

The remainder of this chapter is organized as follows: Section 4.2 describes a framework for modifying variable projection to address Problem 4.5 in the case that a is subject to constraints. Algorithms for Problem 4.7 and Problem 4.8 are described in detail. Section 4.3 discusses the effect on the precision with which nonlinear parameters z are estimated when a is subject to non-negativity constraints. Section 4.4 examines the deviation of estimates for z and a from values used in simulation of a dataset inspired by time-resolved emission

spectra obtained using variable projection with and without the modification to constrain a to non-negativity. Section 4.6 contains conclusions.

4.2 Variable projection with constraints on the conditionally linear parameters

Golub and Pereyra (1972, 1973) developed the variable projection algorithm based on the observation that given a bilinear model of form

$$\psi = C(z)a + \xi \quad (4.11)$$

where ξ is a vector comprised of Gaussian random variables with zero mean and constant standard deviation with the same length as the data vector ψ , a can be solved for in terms of $C(z)$ and ψ using the equality $a = C(z)^+\psi$, where $+$ denotes the Moore-Penrose pseudoinverse. Substituting $C(z)^+\psi$ for a in Problem 4.5 we obtain

$$\min_{z \in \mathbb{R}^q} \| (I - C(z)C(z)^+)\psi \|^2 \quad (4.12)$$

In the following we write $C(z)$ as simply C . Golub and Pereyra determined the Jacobian of $(I - CC^+)\psi$ with respect to z analytically as

$$J = \frac{d(I - CC^+)\psi}{dz} = -Q_2Q_2^TC_zC^+\psi - Q_1R_{11}^{-T}C_z^TQ_2Q_2^T\psi \quad (4.13)$$

where $C_z = \frac{dC}{dz^T}$ and the QR decomposition $C = QR = [Q_1 \ Q_2]R$ is used for numerical stability, where Q_1 is $m \times n_{\text{comp}}$, Q_2 is $m \times (m - n_{\text{comp}})$, Q is orthogonal, and R is $m \times n_{\text{comp}}$ is upper triangular and $R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}$, with R_{11} being $n_{\text{comp}} \times n_{\text{comp}}$ and upper triangular (Golub and van Loan, 1996). Kaufman (1975) suggested that the second term in the expression for the Jacobian of the residuals can be dropped, so that $J = -Q_2Q_2^TC_zC^+\psi$, without undue loss of precision and enabling faster computation, as shown empirically for various problems in Bates and Lindstrom (1986); Gay and Kaufman (1991); Mullen *et al.* (2007).

For many modern applications (van Stokkum *et al.*, 2004) the nonlinear functions on which C is based are sufficiently complex to render analytical derivation unattractive or impossible. A finite difference approximation of the Jacobian $J = \frac{d(I - CC^+)\psi}{dz}$ may be used in such cases to substitute for an analytical determination of J .

Variable projection may thus be formulated as

ALGORITHM VARPRO:

1. Choose starting z_0 .
2. For $s := 1, 2, \dots$ until stopping criteria are met do
 - 2a. Calculate the residual vector as $res = \psi - Ca = Q_2Q_2^T\psi$

- 2b. Determine $J = \frac{d \text{res}}{d z}$ using one of the following prescriptions
- $$J_{NUM} := \text{finite difference approximation of } \frac{d \text{res}}{d z}$$
- $$J_{GP} := -Q_2 Q_2^T C_z C^+ \psi - Q_1 R_{11}^{-T} C_z^T Q_2 Q_2^T \psi$$
- $$J_{KAUF} := -Q_2 Q_2^T C_z C^+ \psi$$
- $$z_{s+1} := \text{STEP, DIRECTION}(z_s, \text{res}, J, \dots)$$

Standard nonlinear least squares algorithms such as Levenberg-Marquardt or Gauss-Newton determine the direction and the step-size to move the parameter vector z as a function of z_s , res , and J (see e.g. Seber and Wild (2003) for a detailed formulation). Should simple bound constraints apply to the nonlinear parameters z , the VARPRO approach remains viable, though z_s should be subject to the appropriate transformation before being used as an argument to STEP, DIRECTION (e.g., by letting $z_i = \exp(\phi_i)$ to constrain ϕ_i to positive values, where ϕ_i is a nonlinear parameter of the function c_i).

Note that under the KAUF and NUM methods for determining J the residual vector may be projected into Q -space by letting $\text{res} = Q_2^T \psi$ in step 2a of VARPRO, and dropping the first factor of Q_2 in step 2b. This modification results in some gain in computational efficiency.

In order to modify VARPRO to constrain a , only the residual calculation in step 2a needs to be modified, provided that J is determined as J_{NUM} . Whereas calculating the residual as $\text{res} = \psi - Ca^T = Q_2 Q_2^T \psi$ is to solve for a as the solution of the least squares problem

$$\min_{a \in \mathbb{R}^p} \|\psi - Ca\|^2 \quad (4.14)$$

any desired constrained optimization algorithm may be substituted for Problem 4.14. The resulting constrained estimates a^* may be used in step 2a as

$$\text{res} = \psi - Ca^* \quad (4.15)$$

with the result being a modified algorithm that enforces the desired constraints on a . The only restriction on the constrained optimization algorithm that may be substituted for Problem 4.14 is practical: a^* and that z that are optimal under least squares criteria should remain easier to obtain using the modified variable projection approach in relation to solving a constrained optimization problem in which the variables are not separated.

Variable projection with constraints on the conditionally linear parameters a may thus be formulated as

ALGORITHM CONVARPRO:

1. Choose starting z_0 .
 2. For $s := 1, 2, \dots$ until stopping criteria are met do
 - 2a. Calculate the residual vector as $\text{res} = \psi - Ca^*$
 - 2b. Determine $J = \frac{d \text{res}}{d z}$ using J_{NUM} where

$$J_{NUM} := \text{finite difference approximation of } \frac{d \text{res}}{d z}$$
- $$z_{s+1} := \text{STEP, DIRECTION}(z_s, \text{res}, J, \dots)$$

To obtain a^* in the case that a is constrained to be non-negative, the sub-problem

$$\begin{aligned} \min_{a^* \in \mathbb{R}^p} \quad & \| \psi - Ca^* \|^2 \\ \text{subject to} \quad & 0 \leq a^*[i] \text{ for } i = 1, 2, \dots, n_{\text{comp}} \end{aligned} \quad (4.16)$$

is solved. The Lawson-Hanson non-negative least squares (NNLS) algorithm (Lawson and Hanson, 1974, 1995) is an efficient means of solving Problem 4.16. Sima and Van Huffel (2007) have applied the Lawson-Hanson NNLS algorithm to solve for a^* within a modified variable projection algorithm to problems in magnetic resonance spectroscopy data analysis. Bro and Jong (1997) and Benthem and Keenan (2004) have suggested some modifications to Lawson-Hanson NNLS that render the algorithm more efficient in some contexts.

To obtain a^* in the case that the desired constraints on a amount to lower and upper bounds on each element $a[i]$, the sub-problem

$$\begin{aligned} \min_{a^* \in \mathbb{R}^p} \quad & \| \psi - Ca^* \|^2 \\ \text{subject to} \quad & l_i \leq a^*[i] \leq u_i \text{ for } i = 1, 2, \dots, n_{\text{comp}} \end{aligned} \quad (4.17)$$

is solved. An efficient algorithm for this problem is the bounded variable least squares (BVLS) algorithm of Stark and Parker (1995), which is computationally more efficient than a standard simplex method or the ‘‘L-BFGS-B’’ box-constraint algorithm of Byrd *et al.* (1995).

Further varieties of constraints may be imposed on a by formulation of, e.g., an appropriate quadratic programming problem.

4.3 Non-negativity constraints on a : effect on precision of \hat{z}

Constraint of $a[i]$ to non-negativity in Problem 4.5 is motivated whenever the nonlinear functions c_i describe phenomena for which negative values are not interpretable. An example of such phenomena is a count of emission photons measured at $1, 2, \dots, m$ times. Such measurements often arise in time-resolved spectroscopy experiments.

In practice we observe that the addition of constraints such as imposing non-negativity on a in such situations increases the precision with which both a and the nonlinear parameters z are estimated. The increase in parameter precision under constraints to ensure the non-negativity of a may be studied using the Fisher information matrix (FIM). The FIM can be calculated for the case that a are estimated with NNLS criteria within the framework presented in Section 4.2, as well as for the case that the standard variable projection algorithm is used. Comparison of the resulting FIMs then will reveal any differences in the parameter precision possible to obtain under the two algorithms. van den Bos (2007) reviews the use of the FIM for the quantification of parameter precision.

Define J as the Jacobian of the residual function with respect to the nonlinear parameters ϕ . Assume the model error σ^2 is estimated as $\sigma^2 = SSE(\phi)/df$, where df is the degrees of freedom of the model. Furthermore, assume as throughout, the noise Ξ is additive, with entries comprised of independent Gaussian random variables with mean zero and constant standard deviation. Then the FIM M may be defined as

$$M = \sigma^{-2} \text{vec}(J)^T \text{vec}(J) = \sigma^{-2} \tilde{M}. \quad (4.18)$$

When M is positive definite the covariance estimate of any unbiased estimator of parameter vector ϕ is bounded below by the inverse of M (the Cramér-Rao Bound), so that

$$\text{Cov}[\hat{\phi}] \geq M^{-1}. \quad (4.19)$$

It has been shown in Chapter 3 that for variable projection in the absence of constraints on a , writing \tilde{M}_{KAUF} per entry (i, j) , with $P = Q_2 Q_2^T$ in the case that a is unconstrained and where $\text{vec}(C_{z_i})$ is the vector representation of $\frac{dC}{dz_i}$,

$$(\tilde{M}_{KAUF})_{ij} = \text{vec}(C_{z_i})^T a a^T \otimes P \text{vec}(C_{z_j}) \quad (4.20)$$

When the conditionally linear parameters are represented by the matrix E as for Problem 4.6 this is $(\tilde{M}_{KAUF})_{ij} = \text{vec}(C_{z_i})^T E^T E \otimes P \text{vec}(C_{z_j})$. The extra term in the FIM under the Golub-Pereyra gradient given in Chapter 5 is negligible for all problems known to exist in applications and $(\tilde{M}_{GP})_{ij}$ is nearly identical to $(\tilde{M}_{NUM})_{ij}$ when $(\tilde{M}_{NUM})_{ij}$ is obtained by inverting the linear approximation unscaled covariance matrix. Therefore we will consider $(\tilde{M}_{KAUF})_{ij}$ to well-approximate both $(\tilde{M}_{GP})_{ij}$ and $(\tilde{M}_{NUM})_{ij}$, and henceforth write $(\tilde{M}_{KAUF})_{ij}$ as simply \tilde{M}_{ij} .

Lawson-Hanson NNLS, which is an active-set algorithm, always solves Problem 4.16 provided C is of full rank, and terminates in a finite number of steps (Lawson and Hanson, 1974). At termination, Lawson-Hanson NNLS has classified each of the n_{comp} constraints $a_i^* \geq 0$ as either active or passive, where a constraint is active if in its absence $a[i] \leq 0$ and passive otherwise. The data ψ are projected into the subspace defined by the columns of C whose indices i appear in the passive set; we write the columns whose indices appear in the passive set C_p . Then $a[i]^*$ for i that occur in the passive set are obtained as $C_p^+ \psi$, and $a[i]$ for i that occur in the active set are zero. Writing $P_p = (I - C_p C_p^+)$, and a_p for those elements of a^* that are non-zero, \tilde{M}_{ij} for the case that a is constrained to non-negativity is,

$$(\tilde{M}_{CON})_{ij} = \text{vec}(C_{z_i})^T a_p a_p^T \otimes P_p \text{vec}(C_{z_j}) \quad (4.21)$$

In the case that the active set is non-empty, C_p represents fewer columns as compared to C , which is to say that the data are represented as a lower order sum of nonlinear functions c_i . The result is an increase in the precision of the nonlinear parameter estimates \hat{z} .

When nonlinear parameters z_i and z_j determine c_i and c_j , respectively, and are very close, variable projection may be unstable due to the near-collinearity of C . If c_j is a column that is present in C but not in C_p , the non-negativity constraints will render the constrained algorithm more stable, and problems due to the collinearity of C may be avoided.

Note that when the conditionally linear parameters a are represented by the matrix E as for Problem 4.6 and each nonlinear parameter z_i contributes to the determination of exactly one column of C , Equation 4.21 may be rewritten as

$$(\tilde{M}_{CON})_{ij} = \sum_{k=1}^n E^T[k, i] * E^T[k, j] * g_i^T P_{p_k} g_j \quad (4.22)$$

where $g_i = \frac{dc_i}{dz_i}$ and $g_j = \frac{dc_j}{dz_j}$, and P_{p_k} is the matrix P_p associated with row k of E^T .

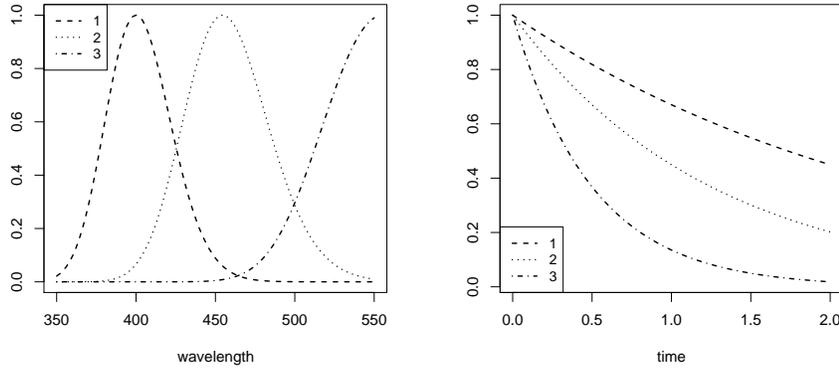


Figure 4.1: Simulated concentration profiles C (right) and spectra E (left). Both C and E are matrices comprised of three columns; each of the columns is plotted here, with its index given in the legend. The decay rates $z = [z_1, z_2, z_3] = [.4, .8, 2]$ determine column i of C as $\exp(-z_i * t)$, where t is a vector of times.

4.3.1 Numerical examples of an increase in parameter precision under non-negativity constraints

We now give an example of the increase in precision under variable projection with non-negativity constraints on the conditionally linear parameters as

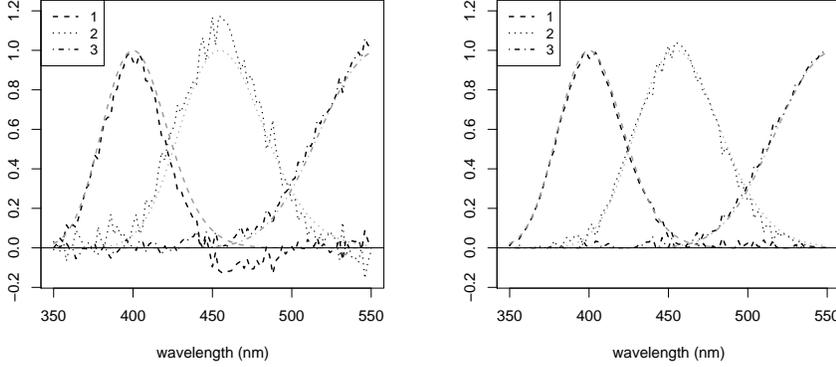


Figure 4.2: Estimates of E from VARPRO (left) and CONVARPRO with constraints to non-negativity on E (right) for a representative noise realization, with each column i of E plotted as a line. Values of E used in simulating the data are shown in grey. A solid line marks zero on the y-axis. The index i of the column of E corresponding to each line is given in the legend.

relative to unconstrained variable projection, as measured by the FIM under the respective algorithms.

Consider the dataset $\Psi = C(z)E^T$, where Ψ is a matrix of dimension $m \times n$ and C and E are matrices of dimension $m \times n_{\text{comp}}$ and $n \times n_{\text{comp}}$, respectively. Here $a = \text{vec}(E^T)$. Since VARPRO requires the data to be in vectorized form, we rewrite $\Psi = CE^T$ as

$$\text{vec}(\Psi) = \text{vec}(CE^T I_n) = (I_n \otimes C)\text{vec}(E^T) \quad (4.23)$$

A partitioned variable projection algorithm that avoids manipulating $(I_n \otimes C)$ directly and thereby allows datasets for which n is large to be modeled without huge memory resources is discussed in Chapter 5 and used in the code accompanying this example.

Let C be determined as $c_i(z_i) = \exp(-z_i t)$ where t is an m -vector. Let each column of E contain a Gaussian shape. C and E simulated using this model with $n_{\text{comp}} = 3$ are shown in Figure 4.1, where $z = [z_1, z_2, z_3] = [.4, .8, 2]$. Let independent, normally distributed noise be added with amplitude 1/100 of the maximum of the deterministic data. Variable projection without constraints may return \hat{E} with some estimates less than zero whereas solving Problem 4.16 within CONVARPRO returns $\hat{E} \geq 0$, as shown in the left and right panels of Figure 4.2, respectively.

The unconstrained and constrained results in Figure 4.2 are associated with

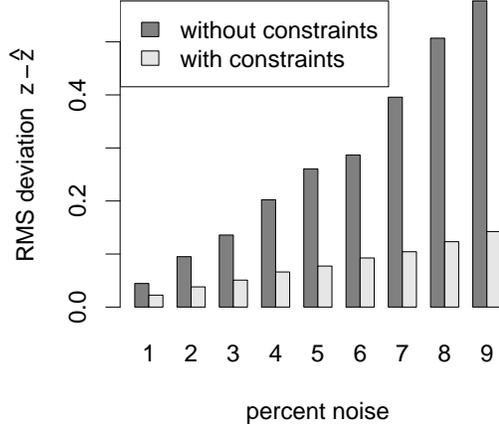


Figure 4.3: Mean RMS deviation $z - \hat{z}$ over 100 instances of the problem described in Section 4.4, for various noise amplitudes. The x -axis indicates the amplitude of the noise term as a percent of the maximum of the deterministic data.

the FIMs

$$\tilde{M} = \begin{matrix} & \begin{matrix} z_1 & z_2 & z_3 \end{matrix} \\ \begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix} & \begin{bmatrix} 0.0581 & -0.0060 & 0.0009 \\ -0.0060 & 0.0355 & -0.0049 \\ 0.0009 & -0.0049 & 0.0564 \end{bmatrix} \end{matrix} \quad (4.24)$$

and

$$\tilde{M}_{CON} = \begin{matrix} & \begin{matrix} z_1 & z_2 & z_3 \end{matrix} \\ \begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix} & \begin{bmatrix} 3.4733 & -0.1624 & -0.0150 \\ -0.1624 & 1.5372 & -0.0967 \\ -0.0150 & -0.0967 & 1.1612 \end{bmatrix} \end{matrix} \quad (4.25)$$

The diagonal elements which describe the precision with which each parameter z_i is estimated are much larger in \tilde{M}_{CON} as compared to \tilde{M} . For the unconstrained and constrained algorithms, the average contribution over n columns of E^T of the diagonal terms $g_i^T P g_j$ and $g_i^T P_{p_k} g_j$ where $i = j$ is .002862 and 0.5111, respectively, which illustrates the gain in precision achieved by the active non-negativity constraints on E .

Scripts to reproduce this numerical example are included in the supporting information (from Mullen and van Stokkum (2008b)).

4.4 Simulation study

To investigate the effect of the addition of non-negativity constraints on the conditionally linear parameters in the manner suggested in Section 4.2 on the accuracy of estimates for z and a for a problem that is typical of those encountered in the analysis of measured time-resolved emission spectroscopy data, we performed a simulation study. VARPRO and CONVARPRO with non-negativity constraints on the conditionally linear parameters were used to solve Problems 4.6 and 4.8, respectively.

The concentration profiles C and spectra E used to generate the deterministic part of the simulated data $\Psi = CE^T$ were similar to those shown in Figure 4.1, with a three-column matrix C in which each column represents an exponential decay, and a three-column E matrix with each column represented by a Gaussian shape. Independent, normally distributed stochastic noise was added to the data, with an amplitude that varied between 1% and 9% of the maximum of the deterministic data. The full specification of the simulation study is given in scripts contained in the supporting information (from Mullen and van Stokkum (2008b)).

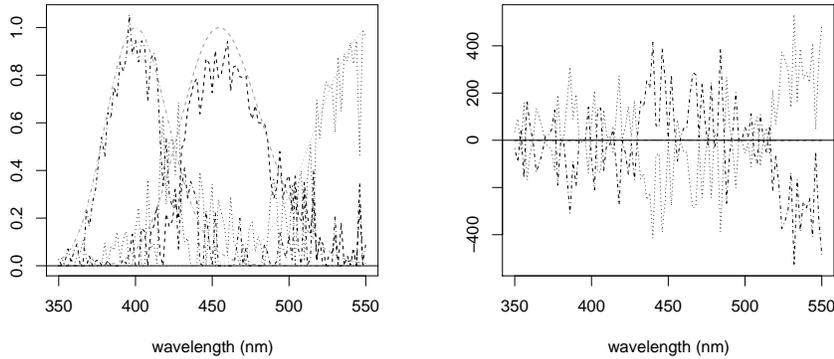


Figure 4.4: For an instance of the dataset in which the noise is 8% of the maximum of the deterministic data, the unconstrained algorithm results in catastrophically compensating estimates for E (right panel), whereas the constrained algorithm returns E (left panel) that qualitatively resemble the values used in simulating the data, which are shown in grey. Here the unconstrained algorithm returns $\hat{z} = [0.501, 2.237, 2.239]$, so that two columns of C are nearly collinear, and converges after 36 iterations. The constrained algorithm returns $\hat{z} = [0.345, 0.798, 2.220]$, which is much closer to the values of $z = [0.4, 0.8, 2]$ used in simulation, and converges after 6 iterations.

Figure 4.3 shows that for all noise amplitudes, the use of the constrained version of variable projection significantly increases the accuracy of \hat{z} .

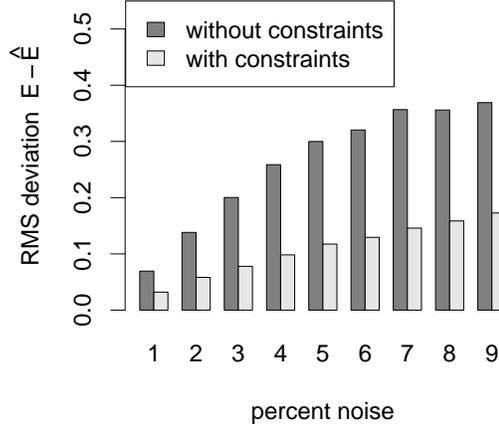


Figure 4.5: Mean RMS deviation $E - \hat{E}$ over 100 instances of the problem described in Section 4.4, for various noise amplitudes. The x -axis indicates the amplitude of the noise term as a percent of the maximum of the deterministic data. Results returned by the unconstrained algorithm that were catastrophically compensating were omitted.

The unconstrained algorithm will occasionally estimate very compensating values for the parameters represented by E , as shown by example in the right panel of Figure 4.4. These catastrophic failures of the unconstrained algorithm represent 0% of the solutions on problem instances with 1% noise, 9-12% of the problem instances with 2-4% noise, and 26-30% of the problem instances with 5-9% noise (where the definition of *catastrophic* was taken to be $\text{RMS } E - \hat{E} > 1$). In calculation of the mean RMS deviation $E - \hat{E}$ over the 100 instances for each noise level shown in Figure 4.5, the catastrophically compensating results were omitted.

That the unconstrained algorithm sometimes demonstrates catastrophic failure to return qualitatively correct estimates \hat{E} shows that VARPRO, like other algorithms for least squares regression, is *unstable* in the sense that small changes in the data (due to the peculiarities of the sample of random noise it contains) may result in huge changes in the RMS deviation of parameter estimates. This instability comes about when two columns of C , c_i and c_j are nearly collinear. For the simple exponential decay model here, this occurs when optimal \hat{z}_i and \hat{z}_j used as $c_i(z_i) = \exp(-z_i t)$ and $c_j(z_j) = \exp(-z_j t)$, respectively, are very similar under least squares criteria, so that \hat{z}_i and \hat{z}_j are not both resolvable, an issue that has been studied in great detail by van den Bos and Swarte (1993). The constrained algorithm eliminates the instability problem when \hat{z}_i and \hat{z}_j are

associated with compensating positive and negative values for $E[k, i]$ and $E[k, j]$ for some wavelength k , since in such situations the column of C associated with the negative value of E will in effect be removed from the model.

Scripts to reproduce these results are provided in the supporting information (from Mullen and van Stokkum (2008b)).

4.5 Implementation

TIMP, a package for the R language and environment for statistical computing (R Development Core Team, 2008) described in Chapter 5, has been developed to solve the problems discussed thus far for the case that $C(z)$ is based on functions useful for modeling the kinetics or the spectral dynamics of spectroscopy, microscopy or mass spectrometry data. Lawson and Hanson (1974) and Stark and Parker (1995) made FORTRAN implementations of Lawson-Hanson and BVLS, respectively, publicly available. The present authors have made an interface to this FORTRAN code available in R via the packages **nmls** and **bvls**, which may facilitate the embedding of these algorithms in the CONVARPRO algorithm discussed in Section 4.2. R and the packages **TIMP**, **nmls** and **bvls** are freely available under the terms of the GNU General Public License from the Comprehensive R Archive Network (<http://cran.r-project.org/>).

4.6 Conclusion

A framework for modification of the variable projection algorithm to include constraints on the conditionally linear parameters was presented. For the case that the modified algorithm enforces non-negativity constraints on the conditionally linear parameters, the precision with which the nonlinear parameters are estimated was shown to increase as compared to estimates obtained under classical variable projection. This result was explored via fitting simulated time-resolved emission spectroscopy datasets with and without non-negativity constraints on the spectra, which comprise the conditionally linear parameters. It was shown that in this context, the extra precision afforded by the non-negativity constraints helps avoid problems due to collinearity.

Future work will include application of the algorithms modified with constraints discussed here to the parametric modeling of mass spectrometry data.

Chapter 5

TIMP: an R package for fitting separable nonlinear models to data arising in physics and chemistry applications¹

5.1 Introduction

TIMP² is a package for the R language and environment for statistical computing (R Development Core Team, 2008) that enables the specification, fitting and validation of separable nonlinear models to data arising in physics and chemistry applications. It is a fully cross-platform problem-solving environment freely available under the terms of the GNU General Public License (GPL).

This chapter outlines the capabilities, structure, and application of the package. The introduction gives an overview of the problems in scientific model discovery that the package has been designed to address. Section 5.2 describes some aspects of the implementation, and includes a description of the partitioned variable projection algorithm that allows variable projection to be used without large memory requirements for parameter estimation problems that arise in the description of 2-way and n -way data. Section 5.3 describes in brief user-accessible functions. Section 5.4 describes general model options. Sections 5.5 and 5.6 describe the specification, fitting and validation of kinetic and spectral

¹A version of this chapter appears as Mullen and van Stokkum (2007b) in the *Journal of Statistical Software*, 18(3), 1-46.

²The name of the package refers to its origins in the **tim** collection of FORTRAN routines developed over the past fifteen years by Ivo H. M. van Stokkum to model *time*-resolved spectroscopy data; **TIMP** stands for *tim* package.

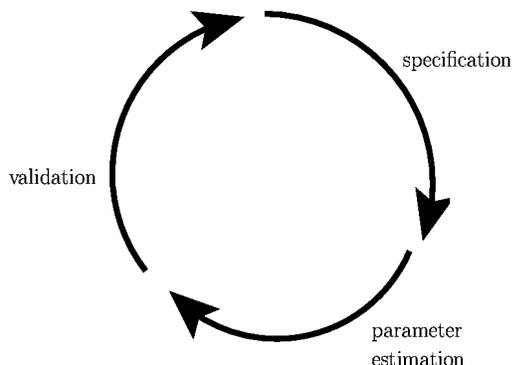


Figure 5.1: Scientific model discovery is often an iterative process of model specification, parameter estimation and validation.

models with the package, respectively. Both of these sections include case studies. Section 5.7 discusses in brief the extension of **TIMP** to new model types. Conclusions are contained in Section 5.8.

5.1.1 Interactive scientific model-discovery

We term *scientific model discovery* the identification of a statistical model able to reproduce experimentally collected measurements to a satisfactory degree of accuracy, with the additional constraint that the model be well-interpretable according to physico-chemical theory. Scientific model discovery very often requires iterating the steps of formulation of a candidate model, model fitting, and model validation. Postulation of a candidate model is guided by *a priori* knowledge of the system underlying the data as well as by exploratory analysis of the dataset (e.g., with decomposition techniques like the singular value decomposition). Model fitting provides estimates for free parameters that are more statistically likely (often under least squares criteria) than the estimates provided during postulation of the candidate model. Validation considers whether the fitted parameter values are precise and likely to be correct according to physico-chemical theory, whether the residuals are sufficiently small and unstructured, and whether adjustment of the candidate model is desirable. The cycle of model formulation, fitting and validation is often iterative because validation often helps identify a more appropriate candidate model.

Scientific model discovery is *interactive* in the case that the time to complete the model formulation, fitting, and validation cycle is determined primarily by the ability of the researcher to decide on and validate candidate models. To allow for interactive scientific model discovery the applied computer hardware and software must enable the researcher to quickly specify and fit a model, and

must provide information for model validation that allows for efficient evaluation of model fit and physical feasibility. **TIMP** provides a framework for interactive scientific model discovery in the multiway spectroscopy data modeling problem domain.

5.1.2 Multiway spectroscopy data and models

A major application domain of **TIMP** is in fitting models to multiway spectroscopy data. A two-way spectroscopy dataset Ψ_q arising under experimental conditions q may be represented as the matrix

$$\Psi_q = \begin{matrix} & \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{matrix} & \begin{bmatrix} \psi(t_1, \lambda_1) & \psi(t_1, \lambda_2) & \dots & \psi(t_1, \lambda_n) \\ \psi(t_2, \lambda_1) & \psi(t_2, \lambda_2) & \dots & \psi(t_2, \lambda_n) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(t_m, \lambda_1) & \psi(t_m, \lambda_2) & \dots & \psi(t_m, \lambda_n) \end{bmatrix} \end{matrix} \quad (5.1)$$

Each row of Ψ_q is a spectrum in the variable λ (which is often wavelength, but may be wavenumber or magnetic field strength; Chapter 6 describes the case that the variable is location). Spectra are represented at m instances of an independent experimental variable t such as time, pH, pD, temperature, excitation wavelength or quencher concentration. The independent experimental variable is chosen so as to monitor spectral change in a manner that provides information on the dynamics of the underlying system. More than one such variable may also be considered, as will be described shortly.

Ψ_q represents a contribution from n_{comp} spectrally distinct components. When a matrix analogue of the Beer-Lambert law for absorption applies to the data, the concentration and spectral property of each component may be represented as column l of matrices C and E , respectively, in the linear superposition model

$$\Psi_q = CE^T \quad (5.2)$$

Each column of C represents a concentration profile of a component in the independent variable t . Likewise, each column of the matrix E represents a spectrum of a component in a second independent variable λ .

The inverse problem of recovery of the entries of C or E in terms of physically significant parameters (descriptive of, e.g., the decay rate of a component, or the location of the maximum of a spectrum) is often of interest. Adequate parameterizations of either C or E are nonlinear, and are usually comprised of many submodels representative of various model aspects. Such parameterizations have been reviewed for the case of time-resolved spectroscopy data by van Stokkum *et al.* (2004); van Stokkum (2005). Very often a model-based description is possible for either C or E , but not both matrices. This gives rise to a separable nonlinear model

$$\Psi_q = C(\theta)E^T + \Xi \quad (5.3)$$

or symmetrically, in the case that the parametric description applies to the matrix E as opposed to the matrix C ,

$$\Psi_q = CE(\theta)^T + \Xi \quad (5.4)$$

The term Ξ represents a matrix of noise, each entry of which is assumed to be an independent Gaussian random variable with expectation zero and constant standard deviation. The assumption of noise with these properties will apply to all models discussed in this chapter.

Let us assume that it is the kinetics C that are described with a parametric model (the presentation for the case that the spectra E are described parametrically is symmetric). Then it may be a case that the parametrization for the kinetics changes per-wavelength represented in Ψ_q . This means that each column j of the data, ψ_j , is described as

$$\psi_j = C_j(\Theta)E[j,]^T \quad (5.5)$$

where the function C_j may vary for different columns j . Equation 5.5 also allows description of datasets Ψ_q that form a ragged matrix, with a different number of observations per-row and per-column. It furthermore allows description of many different datasets $\Psi_1, \Psi_2, \dots, \Psi_K$ with the same vector of nonlinear parameters Θ but using different functional descriptions C_{j_q} , where q is the dataset index. In applications it is common that observations take the form of vectors $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_x}$ with j representing an independent variable (e.g., wavelength) measured many times under different experimental conditions indexed $1, 2, \dots, x$. An assumption that is often physically motivated is that the same vector of conditionally linear parameters a underlies $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_x}$, as will be discussed further in Section 5.2.

As Golub and LeVeque (1979); Golub and Pereyra (2003) review, Θ and the conditionally linear parameters (the entries of the matrix E^T in Equation 5.3 and the entries of C in Equation 5.4) may be estimated from the data Ψ_q under least squares criteria using the variable projection algorithm, which has been shown to be very effective for fitting separable nonlinear models. **TIMP** includes a *partitioned* implementation of variable projection that allows the application of the algorithm to fitting the free parameters of models for matrix data without large memory resources. This method is compared to the standard variable projection implementation (found, e.g., in the `plinear` option of the `nls` function in the R environment) in detail in Appendix 5.A.

5.1.3 An introduction to modeling with TIMP

The application of **TIMP** to a very simple parameter estimation task on a single simulated time-resolved spectroscopy dataset serves to introduce the use of the package. As discussed in the previous section, a spectroscopy dataset Ψ_q can be considered to represent a linear superposition of the concentration profiles C and spectral properties E of components, so that $\Psi_q = CE^T$ (Equation 5.2).

We will simulate a dataset in which two spectrally distinct components contribute to Ψ_q , where C represents concentration in time and E represents spectra resolved with respect to wavenumber. The most simple realistic model for C lets the time-profile of each component c_l be described by an exponential decay with decay rate parameter k_l , so that

$$c_l(t) = \exp(-k_l t), \quad (5.6)$$

where t is time. (for elaboration on the use of exponential models for kinetic processes, see e.g., the review by Istratov and Vyvenko (1999)). We let the two components contributing to Ψ_q have associated decay rate parameters .5 and 1, and let the concentrations be measured at 51 times equidistant in the interval 0-2 ns. Then the following R commands calculate C .

```
R> C <- matrix(nrow = 51, ncol = 2)
R> k <- c(.5, 1)
R> t <- seq(0, 2, by = 2/50)
R> C[, 1] <- exp(- k[1] * t)
R> C[, 2] <- exp(- k[2] * t)
```

The most basic model for the spectrum e_l associated with a single component in wavenumber $\bar{\nu}$ is a Gaussian with parameters $\mu_{\bar{\nu}}$, $\Delta_{\bar{\nu}}$, and a_l , for the location, full width at half maximum (FWHM), and amplitude, respectively, so that

$$e_l(\bar{\nu}) = a_l \exp\left(-\ln(2) \left(2 \frac{(\bar{\nu} - \mu_{\bar{\nu}})}{\Delta_{\bar{\nu}}}\right)^2\right), \quad (5.7)$$

(see e.g., van Stokkum (1997) and references therein regarding the ubiquity of Gaussian models for spectra). Let us consider spectra represented by 51 wavenumbers equidistant in the interval 18000 - 28000 cm^{-1} , with locations 25000 and 20000, FWHMs 5000 and 7000, and amplitudes 1 and 2, respectively. In R we can then calculate E as

```
R> E <- matrix(nrow = 51, ncol = 2)
R> wavenum <- seq(18000,28000, by = 200)
R> location <- c(25000, 20000)
R> delta <- c(5000, 7000)
R> amp <- c(1, 2)
R> E[, 1] <- amp[1] * exp(- log(2) * (2 *
+ (wavenum - location[1])/delta[1])^2)
R> E[, 2] <- amp[2] * exp(- log(2) * (2 *
+ (wavenum - location[2])/delta[2])^2)
```

Given these R expressions for C and E , a dataset Ψ_q with Gaussian noise with zero mean and width $\sigma = .001$ may be generated and added to the data with

```
R> sigma <- .001
R> Psi_q <- C %*% t(E) + sigma * rnorm(nrow(C) * nrow(E))
```

Plots of C and E are shown in Figure 5.2. The simulated dataset Ψ_q is shown in Figure 5.3

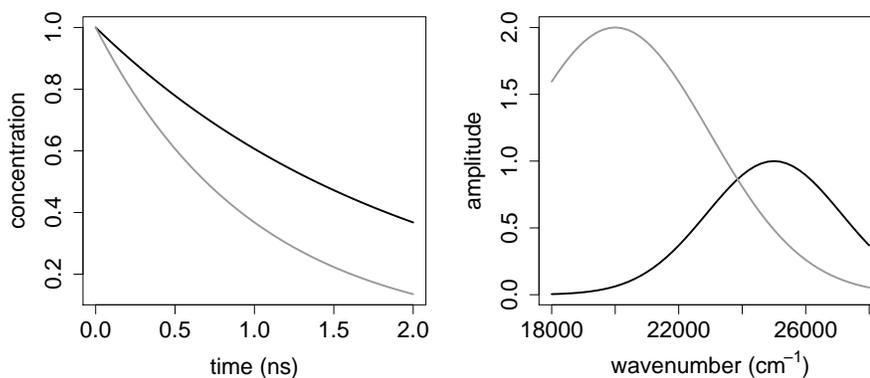


Figure 5.2: (Left) Simulated concentrations (Right) simulated spectra. Component 1 is in black; component 2 is in grey.

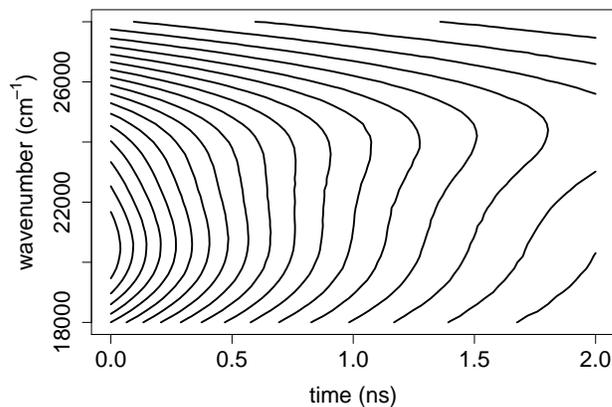


Figure 5.3: Simulated data; model fitting will resolve the two contributing components.

Given a dataset such as the simulated Ψ_q shown in Figure 5.3 it is often the case that either a kinetic model for C or a spectral model for E is desirable to apply to the data. Given estimates for the nonlinear model parameters that determine one of the two matrices, the entries of the other matrix may be solved for as conditionally linear parameters (clp). Let us assume that the kinetic model

used in simulating the data is known to describe `Psi_q`, and that approximate starting values for the two rate constants are known. Then the following R commands can be used to estimate the rate constants k and clp E .

TIMP is loaded with

```
R> library("TIMP")
```

The simulated dataset is placed into an instance of the class `dat` which is used to store data and model objects in **TIMP**. The `dat` object contains not only the data but also some information like its dimensions.

```
R> Psi_q_data <- dat(psi.df = Psi_q, x = t, nt = length(t),
+ x2 = wavenum, nl = length(wavenum))
```

A model to be applied to the data is initialized with the `initModel` function. The `seqmod = FALSE` option indicates that the components decay in parallel; the starting values for the decay rates are given in the vector `kinpar`.

```
R> kinetic_model <- initModel(mod_type = "kin", seqmod = FALSE,
+ kinpar = c(.1, 2))
```

With the next command model parameters are optimized over the course of four iterations using the `fitModel` function of **TIMP**.

```
R> kinetic_fit <- fitModel(data = list(Psi_q_data),
+                          modspec = list(kinetic_model),
+                          opt = kinopt(iter = 4))
```

The call to the `fitModel` function results in a composite plot displaying the fit of the model to the data at each wavenumber, a window showing parameter estimates for the two rate constants and information regarding the residuals, and a summary figure showing the estimated spectra and rate constants. For the case considered here the estimated spectra and concentration profiles well-approximate the entries of E and C used in simulation. **TIMP** employed the `nls` function in estimating the nonlinear parameters $k = \{k_1, k_2\}$.

This introduction to the application of **TIMP** shows how, given a single dataset Ψ_q , a parametric description of the concentration profiles of contributing components can be fit under least squares criteria, while the spectra E are solved for as clp. Subsequent sections will describe the application of **TIMP** to fitting more complex models to possibly many datasets.

5.1.4 Hierarchical models for possibly many datasets

The goal of data analysis given a collection of two-way spectroscopy datasets $\Psi = \{\Psi_1, \dots, \Psi_x\}$ is often to describe the totality of the data Ψ in terms of a unified separable nonlinear model, the parameters of which are estimated according to least squares criteria. Where Θ denotes the nonlinear parameters of such a model, the parameter estimation task requires solving $\min \|\hat{\Psi}(\Theta) - \Psi\|^2$, where, as throughout $\|\cdot\|$ denotes the 2-norm. The model specifies a prescription

to determine residual matrices $Z_1(\Theta), \dots, Z_x(\Theta)$ associated with datasets Ψ_q . The model may be such that residual matrix Z_q depends on only a subset of Θ . Parameters in Θ that determine the residuals associated with multiple residual matrices Z_q (or in formulations considered later, residual vectors) are said to be *linked* between the datasets. Parameters that do not determine the residuals of a given dataset are said to be *unlinked* to the dataset. See e.g., Soo and Bates (1992) for further discussion of models with this structure.

Models applied to data Ψ may be comprised of many submodels, each of which describe a distinct aspect of the underlying system giving rise to the measurements. Examples of submodels include a parametric description of an instrument response function (IRF), a coherent artifact, or the shape of a spectrum. Submodels may also include a prescription for the transformation of parameters in order to enforce constraints or apply relations between parameters, as illustrated in the case study contained in Section 5.5.6. Description in terms of a tree is often well-representative of such parameterizations. As an example, a tree representation for a model describing two datasets $\Psi = \{\Psi_1, \Psi_2\}$ is given in Figure 5.4. The parameters associated with models for the individual datasets Ψ_1 and Ψ_2 are themselves comprised of submodels for various aspects of the underlying system. Note that distinct submodels may depend on the same parameter $\theta_i \in \Theta$, so that parameters are linked between datasets, as Figure 5.5 illustrates diagrammatically.

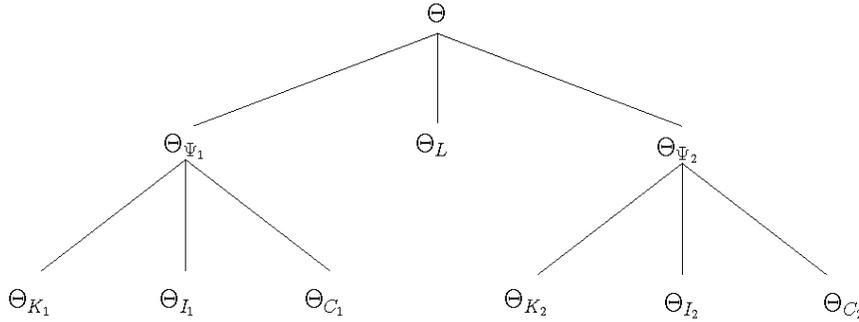


Figure 5.4: A hierarchical model Θ for datasets $\Psi = \{\Psi_1, \Psi_2\}$. The model is comprised of a submodel for each dataset with associated nonlinear parameters Θ_{Ψ_1} and Θ_{Ψ_2} . These submodels are each comprised of a submodel for the kinetic decay rates parametrized with Θ_{K_1} and Θ_{K_2} , a submodel for the IRF, parametrized with Θ_{I_1} and Θ_{I_2} , and a submodel for a coherent artifact parametrized with Θ_{C_1} and Θ_{C_2} .

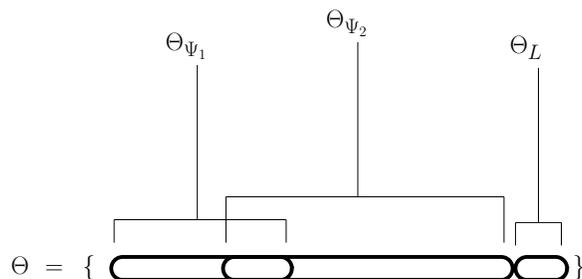


Figure 5.5: The vector of nonlinear model parameters Θ may parametrize a model for multiple datasets. Some parameters $\theta \in \Theta$ may be used to model more than one dataset, so that θ is *linked* between datasets, while other parameters may be used to model only one dataset. Still other parameters may be used to model a relationship *between* datasets, such as a linear scaling. In the above figure, the vectors of parameters Θ_{Ψ_1} and Θ_{Ψ_2} parametrizing the models for datasets Ψ_1 and Ψ_2 , respectively, include some of the same elements from the full vector of nonlinear model parameters Θ . Parameters to determine a relationship between datasets are written as Θ_L , and apply to how the model functions associated with datasets are scaled.

5.2 The implementation of TIMP

TIMP has been designed to facilitate interactive scientific model discovery of the multidataset hierarchical models introduced in Section 5.1.4 and reviewed in van Stokkum *et al.* (2004). In this section we attempt to give a brief overview of the design of the package.

5.2.1 The role of S4 classes and methods in TIMP

S4 classes and methods are the preferred means by which object-oriented programming is implemented in R. See e.g., Chambers (1998) for a review of the S4 classes and methods system, and e.g., Bates and DebRoy (2003) for a description of their utility within a large R package.

S4 classes and methods are central to the implementation of **TIMP**. Slots of S4 classes are used to store each component of a hierarchical model. In the course of optimization (initiated with the `fitModel` function), an object is associated with the parametrization of the multidataset model and the current values of the parameters associated with this model, respectively. The object containing a prescription for the multidataset model is of class `multimodel`. The

Selected S4 classes defined by TIMP	
class	represents
dat	single dataset Ψ_q and an associated model
kin	kinetic model specification, inherits from dat
spec	spectral model specification, inherits from dat
theta	parameter estimates for model for single dataset Ψ_q
multitheta	parameter estimates for (possibly) multiple dataset model
multimodel	multiple dataset model and fit
res	results of fitting (possibly) many datasets

Table 5.1: Some S4 classes used in **TIMP**.

object used to store the current parameter estimates associated with this model is of class **multitheta**.

During each iteration of parameter estimation, **TIMP** updates the S4 object representing the current nonlinear parameter estimates using the updated vector of nonlinear parameter estimates returned by a routine to perform nonlinear least squares. The update is performed by the function **getThetaCl**. The S4 object representing the parameter estimates is subsequently used to form the residual vector to be minimized with respect to the nonlinear parameter vector. By determining the residuals as a function of this object as opposed to as a direct function of the “raw” vector of nonlinear parameter estimates Θ (which is typically of length $10^1 - 10^2$), the residual function implementation is relatively concise and readable. All bookkeeping necessary to account for fixed parameters, parameter relations, and parameter constraints is performed in **getThetaCl**, not the residual function. This allows for the rapid implementation of support for model types associated with new residual functions, as elaborated in Section 5.7.

As elaborated in Section 5.4, aspects of model parametrization not specific to model type (e.g., a weighting specification, constraints, or a specification of fixed parameter values) are specified in the slots of the class **dat**. Aspects of model parametrization specific to a given sort of model (e.g., a model in which the nonlinear parameters apply to the kinetics) are specified in the slots of classes inheriting from **dat**. S4 methods switch the definition of the function that determines the residuals and the output/plotting based on the class of the model type.

5.2.2 Model specification

A model in **TIMP** is specified using the function **initModel**. Either **initModel** can be used to specify multiple models to be applied to the list of datasets, or per-dataset model differences from a single model may be given as the **modeldiffs** argument to the function **fitModel**, or these possibilities may be combined with each other.

To be concrete, the first four arguments of **fitModel** are **data**, **modspec**, **datasetind** and **modeldiffs**. The argument **data** is a list of dataset objects,

each of which is of class `dat` and is typically created using the `readData` function. The argument `modspec` is a list of model objects, each of which also has class `dat` and is created via the function `initModel`. The argument `datasetind` is a vector that has the same length as `data`; it specifies the model each dataset in `data` should have as an index into `modspec`; if not given, then the first model in `modspec` is applied to all datasets. Where the same model is applied to multiple datasets, the argument `modeldiffs` allows the specification of any per-dataset differences such as a parameter that is free to vary per-dataset. `modeldiffs` also allows the specification of parameters used for scaling between datasets.

5.2.3 Parameter estimation

The variable projection algorithm for separable nonlinear least squares problems is central to parameter estimation in **TIMP**. The conditionally linear parameters in Equation 5.3 and 5.4 are possible to determine given estimates for Θ as $\beta = X(\Theta)^+ \psi_j$ where $X(\Theta)$ is the matrix explicitly determined by nonlinear parameters Θ , and ψ_j is a vector of data. Note that we use the notation X and β in place of C and E here in order to stress that a nonlinear model may be applied to describe either C or E directly, resulting in the same parameter estimation task.

The parameter estimation task is possible to formulate in terms of Θ alone, as

$$\text{Minimize } \|\psi_j - X(\Theta)\beta\|^2 \quad (5.8)$$

where $\|\cdot\|$ is, as throughout, the 2-norm. Using the QR decomposition, $C = QR = [Q_1 \ Q_2]R$, where Q_1 is $m \times n_{\text{comp}}$, Q_2 is $m \times (m - n_{\text{comp}})$, Q is orthogonal, and R is $m \times n_{\text{comp}}$, $R = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}$, with R_{11} being $n_{\text{comp}} \times n_{\text{comp}}$ and upper triangular (Golub and van Loan, 1996), the residuals $\psi_j - X(\Theta)\beta$ appearing in the objective function can be written as $r(\Theta) = (I - X(\Theta)X^+(\Theta))\psi_j = Q_2 Q_2^T \psi_j$. For computational efficiency, the first factor of Q_2 may be dropped with no effect on the final parameter estimates, so that $r(\Theta) = Q_2^T \psi_j$. By solving for the conditionally linear parameters as a function of the intrinsically nonlinear parameters Θ , the nonlinear search space is often very significantly reduced, leading to a much better-determined model that is faster to fit.

$(I - X(\Theta)X^+(\Theta))\psi_j$ is termed the variable projection functional. Golub and Pereyra (1972, 1973) give an analytical expression for the Jacobian of first derivatives of this vector with respect to Θ . This functional may also be minimized, as in **TIMP**, using a finite difference approximation for the Jacobian.

The Jacobian allows the determination of the gradient and an approximation of the Hessian in a standard nonlinear least squares algorithm. **TIMP** allows either the Gauss-Newton implementation in `nls` function of **R**, the `nlsol` implementation also in `nls`, or the Levenberg-Marquardt implementation in the **R** function `nls.lm` from the package `minpack.lm` (Elzhov and Mullen, 2008) to be used for the minimization of $r(\Theta)$ with respect to Θ under least squares criteria.

5.2.4 Partitioned variable projection

The models 5.3 and 5.4 discussed in the introduction are for matrix data. When Equation 5.5 is applied to account for a dependence of the model $X_j(\Theta)$ on the column j of the (ragged) matrix data being modeled, the residual vector is formed as

$$r(z) = \begin{bmatrix} Q_2^T \Psi[1] \\ Q_2^T \Psi[2] \\ \vdots \\ Q_2^T \Psi[n] \end{bmatrix}. \quad (5.9)$$

Above, the matrix Q_2^T is calculated for each matrix X_j in $1, 2, \dots, n$.

For the case of modeling multiple datasets $\Psi_1, \Psi_2, \dots, \Psi_x$, observations often take the form of vectors $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$ with j representing an independent variable (like a wavelength, or a location) measured many times under the different conditions indexed $1, 2, \dots, x$. Then the assumption that the same vector of conditionally linear parameters a underlies $\psi_{j_1}, \psi_{j_2}, \dots, \psi_{j_K}$, is possible to account for by letting

$$\psi_j = \begin{bmatrix} \psi_{j_1} \\ \psi_{j_2} \\ \vdots \\ \psi_{j_K} \end{bmatrix} = \begin{bmatrix} X_{j_1}(\Theta) \\ X_{j_2}(\Theta) \\ \vdots \\ X_{j_K}(\Theta) \end{bmatrix} a_j = X_j(\Theta) a_j \quad (5.10)$$

where the second subscript on ψ and X is the dataset index. The residual vector associated with using Equations 5.10 to model the columns $j = 1, 2, \dots, n$ of (ragged) matrix data is also determined as in Equation 5.9 with the modification that Q_2^T is re-calculated for each matrix $X_j(\Theta)$.

We refer to this partitioned strategy for forming the residual vector per-part p as `PARTITIONEDVARPRO`. The alternative to this strategy is to formulate the matrix data fitting problem in vectorized form, as explained in the Appendix 5.A. The vectorized formulation of the model and the data is required by standard variable projection implementations such as that found in e.g., the `plinear` function of `nls` in R. The advantage of `PARTITIONEDVARPRO` is that significantly less memory is required, allowing application of the algorithm on large datasets on a personal computer.

To examine the implementation of partitioned variable projection in `TIMP`, see the `rescomp` function which collects the residual vector Z and one of the `S4` methods for `residPart`, which returns the residuals (with a contribution from possibly many datasets) associated with a single part p .

5.2.5 Validation

Model validation in `TIMP` may be performed via a variety of means. Nonlinear parameter estimates as returned by `nls` may be validated via the linear ap-

proximation standard error estimates returned. Relatively large standard errors indicate an over-parametrization of the model.

For each model type, an S4 method `plotter` is implemented to output model type-specific results, both in the form of plots and in the form of ASCII files representing the model fit and other estimates.

Analysis of the residuals is an important aspect of model validation. This analysis is often facilitated by taking a singular value decomposition (SVD) of the residual matrix, which allows structure in the misfit of the model to the data to be readily observed. Also important for an evaluation of the model fit are plots of the fit of the model to the data for each row or column of the data (possibly for each of multiple datasets).

It is often desirable in scientific modeling applications to perform model validation after the satisfaction of *stopping* criteria, as opposed to convergence criteria, are satisfied. For example, it is often desirable to validate the model fit after a set number of iterations, or at the starting values of parameters to be optimized. In order to allow validation to be performed after a set number of iterations, the `nls` function of R was extended with new options, which are described in Appendix 5.B. These options allow the `fitModel` of **TIMP** to return information validating the model fit after any desired number of iterations.

5.3 User-accessible functions

TIMP is currently structured around five core user-accessible functions. More complete information regarding function arguments and output is found in the `help` functions of the package; here a higher-level description of the purpose and structure of the arguments and output is given.

5.3.1 `readData`

`readData` takes as an argument a string containing the path to an ASCII-file containing data and reads the data into R. The supported data formats, are described in Appendix 5.C. The data (and inferred attributes, such as the number of wavelengths by which spectra are represented, etc.) are returned as an object of class `dat`.

5.3.2 `preProcess`

`preProcess` takes an argument of class `dat` and a specification of the desired data sampling, selection, baseline correction, or axis scaling, and the dimension in which to perform the preprocessing (which may be the spectral dimension or the dimension in which spectra are resolved, e.g., time). The `preProcess` function returns an object of class `dat`.

5.3.3 `initModel`

The `initModel` function is used to specify a model. A string `mod_type` giving the class of the model being specified, ("`kin`" for kinetic models, "`spec`" for spectral models, and so forth) is a mandatory argument. Additional arguments may be any model options described in the `help` page for the `dat` class, plus options described on the `help` page of the desired model class given as `mod_type`. Output is an object of the desired model class, which inherits from `dat`.

5.3.4 `fitModel`

`fitModel` performs optimization of a model to an arbitrary number of datasets. Arguments to `fitModel` include those described in Section 5.2.2. Options to control the number of iterations, and various other aspects of the optimization and output of results are specified in the `opt` argument, which should be an object of class `opt`. Subclasses of class `opt` include `kinopt` and `specopt` which allow options specific to kinetic and spectral models, respectively, to be set.

5.3.5 `examineFit`

`examineFit` takes as input the list returned by `fitModel` and re-calls the plotting and functions to write output. This function is useful for the comparison of fit of several models. An output object is not returned.

5.4 General model options

This section seeks to outline at a higher level than that found in the package's `help` pages model parametrization options that may be applied to all model types. The specification of each such option becomes a slot in the class `dat`, possibly after processing (within the `initModel` or `getModel` functions). Options that are specific to a given model type that inherits from `dat` (e.g., the class `kin` for kinetic model options or the class `spec` for spectral model options) are described in later sections.

The function `initModel` takes as arguments a specification of model options. Note that for multidataset models, any aspect of a parametrization of a model possible to specify in `initModel` may be modified, removed or added to the prescription of per-dataset model differences given as the `modeldiffs` argument to the fitting function `fitModel`.

5.4.1 Data weighting

A weighting scheme W has the form of an m by n matrix (where m by n is the dimension of the data matrix Ψ). W may lessen the weight of portions of the data known to contain less information regarding the model parameters. Such data may result from noisy experimental conditions, for instance. The application of a weighting scheme may also be desirable from first principles,

e.g., in the case that the data are known to have a variance related to their magnitude as in single photon counting (SPC) experiments, in which the data are Poisson distributed, with

$$\hat{\sigma}_{\Psi(t_i, \lambda_j)} = \sqrt{\Psi(t_i, \lambda_j)}. \quad (5.11)$$

For the case of Poisson distributed count data, $W(i, j)$ is

$$W(i, j) = \frac{1}{\hat{\sigma}_{\Psi(t_i, \lambda_j)}}. \quad (5.12)$$

Once W has been determined, the Hadamard product (element-by-element product) of W and Ψ is taken, so that

$$\Psi^W(t_i, \lambda_j) = \Psi(t_i, \lambda_j) * W(t_i, \lambda_j) \quad (5.13)$$

which we write as

$$\Psi^W = \Psi \circ W \quad (5.14)$$

In the case that Ψ is transformed by a weight matrix W into Ψ^W , the associated concentration matrix becomes C^W where

$$C_{\lambda_j}^W(t_i, l) = C_{\lambda_j}(t_i, l) \circ W_j = C \circ W_j \quad (5.15)$$

where W_j is the column of W corresponding to λ_j .

Ψ^W and C^W may then be used in place of Ψ and C in parameter estimation.

Specification in TIMP: Weighting

The list argument `weightpar` specifies a prescription for the matrix of weights to be applied to the dataset. `weightpar` is a list of vectors. The vectors have form

`c(first_x, last_x, first_x2, last_x2, weight)`. `first_x` and `last_x` are the least and greatest times (or other variable with which spectra are resolved) having weight `weight`; `first_x2` and `last_x2` are the least and greatest values of the spectral variable having weight `weight`.

Note that if vector elements 1-4 are NA, the first point of the data is taken for elements 1 and 3, and the last points are taken for 2 and 4. For example, for a dataset in which spectra are measured in wavelength at many different times in picoseconds, the specification `weightpar = list(c(40, 1500, 400, 600, .9), c(NA, NA, 700, 800, .1))` will weight data between 40-1500 ps and 400 and 600 nm by .9, and will weight data at all times between 700 and 800 nm by .1.

For single photon counting data or other types of count datasets, `weightpar = list(poisson = TRUE)` will apply Poisson weighting to all non-zero elements of the data.

5.4.2 Fixed parameters

It is often of interest to set nonlinear model parameters to fixed values. Fixing model parameters may make use of a *priori* knowledge of true parameter values, or may be performed to decrease the number of free parameters of the model.

Specification in TIMP: Fixed parameters

Every model parametrization option with an associated list or vector of nonlinear parameter starting values is named. For instance, the name of the starting values for kinetic decay rates is "`kinpar`". In order to fix nonlinear parameters, the name of their list or vector of starting values is given, along with the indices into the list or vector at which parameter values should be fixed. This specification is contained in a list `fixed`.

For instance `fixed = list(kinpar = c(1,3,5), parmu = list(c(1,1), c(1,2), c(1,3)))` will fix the 1st, 3rd, and 5th elements of the `kinpar` vector of starting values for kinetic decay rates, and the 1st, 2nd, and 3rd elements of the 1st list of parameters in the `parmu` list of starting values for parameters describing wavelength-dependence of the IRF.

5.4.3 Constraint of `clp`

The basic superposition model $\Psi = CE^T$ (Equation 5.2) is of bilinear form, where the matrix C describes concentrations and the matrix E describes spectra. A nonlinear model may be used to describe either C or E , and the entries of the remaining matrix estimated as `clp`, as described in Section 5.2.3.

It is often desirable to constrain `clp` to account for a *priori* knowledge or to reduce the number of free parameters in the model. **TIMP** currently allows `clp` to be constrained to a linear relationship with a scaling parameter (that may be fixed at 1 to equate `clp`), or to be constrained to zero. All `clp` may be also constrained to non-negative values via the methodology described in Chapter 4.

It is often useful to name the `clp` to be constrained in terms of the component they represent, (i.e., the column of C or E they are contained in).

Specification in TIMP: Constraint of `clp` to zero

The list `clp0` contains lists that specify `clp` to constrain to zero. The elements of these lists are named `low`, `high`, `comp`, specifying the least and greatest absolute values of the `clp` dimension to constrain to zero, and the component to which to apply the zero constraint, respectively. For example, where `clp` represent spectra in wavelength, `clp0 = list(list(low=400, high = 600, comp=2), list(low = 600, high = 650, comp = 4))` applies zero constraints to the spectra associated with component 2 between 400 and 600 nm, and to the spectra associated with component 4 between 600 and 650 nm.

Specification in TIMP: Constraint of clp to a linear relationship

The list `clpequspec` contains lists that specify collections of clp to relate. The elements of these lists are named `to`, `from`, `low`, `high`. An optional element named `dataset` specifies the dataset from which to get the reference clp determining the relationship. `to` is the component from which clp are to be fixed in relation to clp from some other component; `from` is the reference component. `low` and `high` are the least and greatest absolute values of the clp dimension to constrain. For example, where clp represent spectra in wavelength, `clpequspec = list(list(low = 400, high = 600, to = 1, from = 2))` will constrain the spectra associated with the first and the second components to equality between 400 and 600 nm according to $\epsilon_1 \leftarrow \epsilon_2 \theta_\epsilon$ where ϵ_1 and ϵ_2 are the spectra associated with components 1 and 2, \leftarrow indicates that ϵ_1 is dependent on ϵ_2 , and θ_ϵ parametrizes the linear relation.

The vector `clpequ` contains `length(clpequspec)` numerics, where the i th numeric is a starting value θ_{ϵ_i} parametrizing the linear relation specified between clp by the i th list in `clpequspec`. Fixing the starting value of an element of `clpequ` at 1 constrains the associated clp to equality.

Specification in TIMP: Constraint of clp to non-negativity

To constrain clp to non-negativity, the `opt` argument of `fitModel` should set `nnls = TRUE`, as in `opt = kinopt(nnls = TRUE)`.

5.4.4 Relations between nonlinear parameters

It may be desirable to enforce a relationship between two nonlinear parameters θ_1 and θ_2 so that $\theta_2 = f(\theta_1)$. The relationship of nonlinear parameters is usually performed in order to take into account *a priori* knowledge of the system being modeled.

A linear relationship between parameters is currently implemented, (and other functional relationships will be added).

Specification in TIMP: Nonlinear parameter relations

As in the case of fixing parameters, in order to relate nonlinear parameters, the name of the associated list or vector of starting values is given, along with the indices into the list or vector at which parameter values are related. This specification is contained in a list `prelspec`.

Each element of `prelspec` is a list having elements named `what1` (a character string describing the parameter type to relate, e.g., "kinpar"), `what2` the parameter type on which the relation is based; usually the same as `what1`), `ind1` (an index into `what1`) and `ind2` (an index into `what2`), and the optional argument `rel` (a character string to specify the functional relation type, by default "linear"). For examples, `prelspec = list(list(what1 = "kinpar", what2 = "kinpar", ind1 = 1, ind2 = 5))` relates the 1st element of `kinpar`

to the 5th element of `kinpar` according to `kinpar[2] ← kinpar[1]` where `←` indicates that `kinpar[2]` is dependent on `kinpar[1]`.

The vector `prel` is of length `length(prelspec)` and contains numeric starting values parametrizing the linear relationships described in `prelspec`.

5.4.5 Constraint of nonlinear parameters to positivity

It may be known *a priori* that the nonlinear parameters associated with a sub-model should be positive. Then optimizing on the log of these parameters and transforming the results by exponentiation of the estimated parameters is a means of enforcing the desired constraint.

Specification in TIMP: Constraint of parameters to positivity

The vector `positivepar` includes a character string containing the name of the vector or list of starting parameters that should be constrained to positivity, e.g., `positivepar = c("kinpar")`.

5.5 Kinetic models

Kinetic models describe the concentrations of components in time. For multiway spectroscopy data modeling applications the basic model for the kinetics of each component (i.e., each column of the matrix C) is an exponential decay in time t , so that

$$\Psi = CE^T = \sum_{l=1}^{n_{\text{comp}}} c_l \epsilon_l^T = \sum_{l=1}^{n_{\text{comp}}} (\exp(-\phi_l t) \oplus i(t)) \epsilon_l^T \quad (5.16)$$

where i is the instrument response function (IRF) and \oplus is convolution. The parameter estimation problem of optimal values for the amplitudes ϵ_l of the exponential decays (i.e., the spectra E) along with the decay rates ϕ_l under least squares criteria is called the multiexponential analysis problem, and is ubiquitous in physics applications in which data is modeled by the solution of first-order differential equations, as Istratov and Vyvenko (1999) review.

Kinetic models are represented in **TIMP** with the class `kin`. Options for the parametrization of kinetic models in **TIMP** are here outlined.

5.5.1 Model for the decay of components

The basic model for the concentration matrix C is a sum of n_{comp} exponential decays parametrized as $\Theta_K = (k_1, \dots, k_{n_{\text{comp}}})$, so that the entries of the concentration matrix $C(i, l)$ are given as

$$C(i, l) = \exp(-k_l t_i). \quad (5.17)$$

Specification in TIMP: Kinetic decay rates

The vector `kinpar` contains numeric starting values for the kinetic decay rates, which in the absence of a compartmental scheme parametrize exponential decays. The number of values given determines n_{comp} , the number of components dedicated to modeling kinetic decays.

5.5.2 Instrument response models

Multisway spectroscopy experiments often employ a short laser pulse to excite the system under study and measure the resulting spectra in time. The convolution of the shape of this exciting pulse and the detector response is the IRF. With pump-probe spectroscopy the IRF is given by the convolution of a pump and a probe pulse. With Gaussian-shaped pump and probe pulses, the convolution of the two will again be Gaussian-shaped, but with an increased width.

An IRF may either be parametrized in the model, or measured. A parametric model for the IRF i as a Gaussian in the time dimension t gives the following model for column l of the concentration matrix C

$$c_l(t, k_l, \mu, \Delta) = \frac{\exp(-k_l t)}{2} \exp(k_l(\mu + k_l \tilde{\Delta}^2/2)) \left\{ 1 + \operatorname{erf} \left[\frac{t - (\mu + k_l \tilde{\Delta}^2)}{\sqrt{2} \tilde{\Delta}} \right] \right\} \quad (5.18)$$

where t is time, k_l is the l th decay rate, μ and Δ are the location and full width half maximum (FWHM) parameters of the Gaussian distribution, respectively, $\tilde{\Delta}$ is the Gaussian width parameter (such that $\tilde{\Delta} = \Delta/2\sqrt{2\ln 2}$). Recall that the FWHM of the Gaussian distribution is related to the standard deviation σ as $FWHM = 2\sqrt{2\ln 2}\sigma$.

In the case that a measured IRF is used as opposed to a model for the IRF with parameters to be estimated, its numerical convolution with the exponential decay function yielding the kinetic decays is required. **TIMP** allows for either methods of including the effects of the IRF to be applied.

Specification in TIMP: Gaussian IRF

The vector `irfpar` contains starting values for the parametric description of the IRF in terms of a Gaussian. The vector is ordered `irfpar = c(μ , Δ)`. For example, `irfpar = c(-2, .05)` specifies a starting location parameter μ as -2 and a width Δ as .05, where the starting values are in the unit of time represented by the data, e.g., picoseconds.

Specification in TIMP: Measured IRF

The vector `measured_irf` is of length equal to the number of times represented by the data, and contains the measured IRF; if specified, it will be applied to the data.

The integer `convalg` is between 1-4 and determines the numerical convolution algorithm used. `convalg = 1` is the default and is recommended.

5.5.3 Models for dependence of IRF parameters on spectral variable

In the case that an IRF is included in the model using a parametric description, it is often desirable that the parameters involved are dependent on a spectral variable (e.g., wavelength). The dependence of IRF on the spectral variable is termed dispersion.

Time-gated spectra are measurements of an optical property (e.g., emission or absorption) as a function of a spectral variable like wavelength taken simultaneously across some range, repeated at many distinct times. In kinetic models of time-gated spectra, dispersion is often well-modeled as a smooth function of the spectral variable. Polynomial functions of degree `nparmu` and `partau` are often used to describe the variance in the spectral variable of the IRF location parameter μ and FWHM parameter Δ , the coefficients of which become additional parameters of Θ_I . Then the IRF μ and Δ are calculated per wavelength as

$$\mu(\lambda) = \mu_0 + \sum_{i=1}^{nparmu} \mu_i \left(\frac{\lambda - \lambda_c}{100} \right)^i, \quad (5.19)$$

$$\Delta(\lambda) = \Delta_0 + \sum_{i=1}^{partau} \Delta_i \left(\frac{\lambda - \lambda_c}{100} \right)^i \quad (5.20)$$

where μ is a function that takes a wavelength or wavenumber and gives the location of the IRF, μ_0 is the location of the IRF at λ_c , Δ is a function that takes a wavelength or wavenumber and gives the scaled width of the IRF, and Δ_0 is the scaled width of the IRF at λ_c .

For data collected by measuring decay traces at many different wavelengths dispersion is often well-modeled by shift parameters for μ and Δ per-wavelength. Where `nl` is the number of points whereby spectra are represented, this results in `nl` shift parameters parametrize the dispersion of μ and `nl` shift parameters parametrizing the dispersion of Δ . We refer to models of dispersion employing a shift parameter per-wavelength as *discrete*.

Specification in TIMP: Dispersion models

The character strings `dispmufun` and `disptaufun` determine the functional form of the dispersion of the IRF location parameter. The default is a polynomial description. If `dispmufun` or `disptaufun` is equal to "discrete" a discrete model for dispersion of the corresponding variable is applied. For the discrete case `parmu` or `partau` should contain a starting value for the shift for every point by which spectra are represented (e.g., for each wavelength).

The numeric `lambdac` is supplied if a polynomial description of either dispersion of location or FWHM is applied. Then `lambdac` is the center-wavelength in this description (λ_c in Equations 5.19 and 5.20).

The list `parmu` contains starting values for the parameters of the model for dispersion of the IRF location. The vector `partau` contains starting values for the parameters of the model for dispersion of the IRF FWHM.

5.5.4 Coherent artifact/scatter models

Should the measurements contain an instantaneous response or coherent artifact due to Raman scatter, it may be desirable to include in the model for the concentrations C its description in time. This is done by adding components (which are columns of the C matrix) to represent its contribution.

A commonly used model for coherent artifact/scatter has the time characteristics of the IRF, in which case a single column is appended to the concentration matrix with the IRF time profile. A variation of this model maintains separate coherent artifact spectra for each of x datasets Ψ_q .

Another commonly used model type employs a sequential scheme with, e.g., femtosecond lifetimes, in which the signs of the amplitude of consecutive components alternates. This model type often well-describes an oscillatory coherent artifact. In the case that the instantaneous response of the exciting pulse has both scatter and coherent response components, a linear superposition of the model that follows the IRF time profile and a model based on a sequential scheme may be applied.

An aside on the implementation of ultra-fast coherent artifact lifetimes

Models for the coherent artifact model type that employ a sequential kinetic scheme are often well-fit with ultra-fast lifetimes, and hence large exponential decay rates. For very large decay rates under Equation 5.18, it may be desirable to employ the exponentially scaled error function $\exp(x^2)\text{erfc}(x)$, as in the `decayirf` function of **TIMP**. An implementation of the exponentially scaled error function (`erfce`) is currently unavailable within R. Its use in **TIMP** is made possible by porting the implementation found in the Cephes Mathematical Library (Moshier, 1992) to an R shared library in C.

Specification in **TIMP**: Coherent artifact/scatter models

The list `cohspec` describes the model for coherent artifact/scatter component(s). If `cohspec$type` is "irf", the coherent artifact/scatter has the time profile of the IRF. If `cohspec$type` is "freeirfdisp" the coherent artifact/scatter has a Gaussian time profile whose location and width are parametrized in the vector `coh` (independent from the IRF parameters). If `cohspec$type` is "irfmulti" the time profile of the IRF is used for the coherent artifact/scatter model, but the IRF parameters are taken per dataset (for the multidataset case), and

`cohspec$numdatasets` must be equal to the number of datasets modeled. If `cohspec$type` is "seq" a sequential exponential decay model is applied, whose parameters are contained in `coh`. If `cohspec$type` is "mix" a sequential exponential decay model is applied along with a model that follows the time profile of the IRF; the coherent artifact/scatter contribution is then a linear superposition of these two models.

The vector `coh` contains starting values for the parametrization of a coherent artifact/scatter model.

5.5.5 Compartmental models

A linear time-invariant compartmental model may be used to describe allowed transitions between components, as Godfrey (1983) reviews. Transitions between compartments are described by microscopic rate constants which constitute the off-diagonal elements of the transfer matrix K . The diagonal elements of K contain the total decay rates of each compartment. The concentrations of the compartments in continuous time are described by a vector $c(t) = [c_1(t), \dots, c_{n_{\text{comp}}}(t)]^T$. Thus, a linear compartmental model with n_{comp} compartments is described by a differential equation for these concentrations

$$\frac{d}{dt}c(t) = Kc(t) + j(t) \quad (5.21)$$

where the input to the system is described by a vector $j = i(t)[j_1 \ j_2 \ \dots \ j_{n_{\text{comp}}}]$ such that $\sum_{l=1}^{n_{\text{comp}}} j_l = 1$, and $j_{n_{\text{comp}}} \equiv 1 - \sum_{l=1}^{n_{\text{comp}}-1} j_l$. Also, generally $j_l \geq 0$. The IRF $i(t)$ describes the time-profile of the inputs. Under the assumption that the eigenvalues of K are different, and that $c(-\infty) = 0$, Equation 5.21 is solved as

$$c(t) = \exp(Kt) \oplus j(t). \quad (5.22)$$

Equation 5.22 is used as the prescription for the concentrations of the components in discrete time. Note that Equation 5.22 requires evaluation of the exponential of the K matrix. This exponentiation is implemented in **TIMP** as described in van Stokkum (2005).

Analysis in the absence of a compartmental model is equivalent to the application of a K matrix with non-zero elements only on the diagonal (Figure 2.1, left panel). A commonly applied compartmental model is termed a *sequential model*, in which $n_{\text{comp}} - 1$ components decay to a single other component, one component decays directly to the ground state and no loops are present (Figure 2.1, right panel).

Specification in TIMP: Parallel/sequential compartmental model

The logical `seqmod` determines whether a sequential compartmental model is applied. If `seqmod = FALSE` then a parallel compartmental model is applied. The default is `seqmod = TRUE`.

Specification in TIMP: Full compartmental model

The array `kmat` has dimension attribute $c(2, n_{\text{comp}}, n_{\text{comp}})$. That is, `kmat` consists of two matrices of dimension $n_{\text{comp}} \times n_{\text{comp}}$. The matrix `kmat[1, 1 : ncomp, 1 : ncomp]` is l at position `kmat[1, j, i]` if a transition from component i to component j is allowed, and is parameterized by the l th element of `kinpar`, and else is 0. Non-zero elements `kmat[1, j, j]` on the diagonal indicate transfer out of the system (to a ground state).

The matrix `kmat[2, 1 : ncomp, 1 : ncomp]` is k at position `kmat[2, j, i]` if the transition from component i to component j is parametrized by a branching parameter from `kinscal` with index k , and else is 0.

The vector `jvec` contains the j vector descriptive of the inputs to the transfer matrix K .

The vector `kinscal` is descriptive of starting values for branching parameters of K .

5.5.6 Case study: Multiexperiment analysis with a kinetic model

An example of multiexperiment kinetic modeling of data $\Psi = \{\Psi_1, \Psi_2\}$ is considered in this section. Time-resolved difference absorption spectroscopy datasets Ψ_1 and Ψ_2 representing wavelengths in the visible range were collected under the same experimental conditions except the excitation laser intensity was doubled during collection of Ψ_1 relative to the laser intensity used during collection of Ψ_2 . The challenge in modeling is to obtain a parametric description of the kinetics of the underlying system as evidenced by both datasets, as well as a parametric description of how the difference in laser intensity affects the system.

Data input

The data $\Psi = \{\Psi_1, \Psi_2\}$ is read into **TIMP** in the *time explicit* format described in Appendix 5.C via the commands

```
R> psi_1 <- readData("psi_1.txt")
Read 1 item
Read 2385 items

R> psi_2 <- readData("psi_2.txt")
Read 1 item
Read 2385 items
```

where Ψ_1 is stored in the file “`psi_1.txt`” and Ψ_2 is stored in the file “`psi_2.txt`” (distributed with the package).

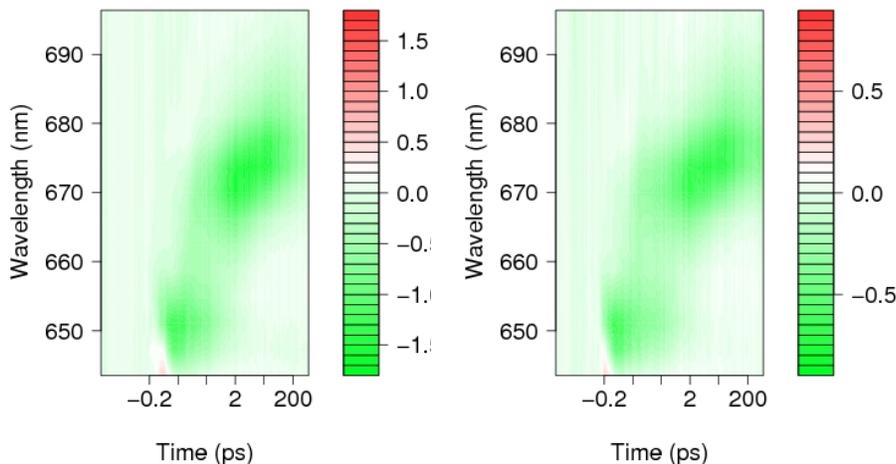


Figure 5.6: (Left) Dataset Ψ_1 measured using twice the laser intensity used to measure (Right) dataset Ψ_2 . The color palette used to display these datasets is generated with the `diverge_hsv` function of the `vcd` package (Meyer *et al.*, 2006).

Data preprocessing

The wavelength axis of the data is scaled via the prescription $x = 3.78x + 643.5$.

```
R> psi_1 <- preProcess(data = psi_1, scalx2 = c(3.78, 643.5))
R> psi_2 <- preProcess(data = psi_2, scalx2 = c(3.78, 643.5))
```

An initial model

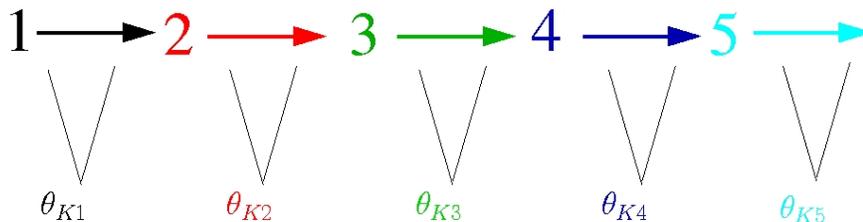


Figure 5.7: Kinetics are described by a five-component sequential compartmental model with identical decay rate parameters for both datasets.

It is known *a priori* that the underlying system is likely to contain five components decaying sequentially, that a Gaussian IRF model is likely to be

appropriate, and that a contribution from a coherent artifact must be accounted for. Approximate starting estimates for parameter values are also known.

This leads to the following model specification in **TIMP**.

```
R> model1 <- initModel(mod_type = "kin",
+ kinpar = c(7.9, 1.08, 0.129, .0225, .00156),
+ irfpar = c(-.1018, 0.0434),
+ parmu = list(c(.230)),
+ lambdac = 650,
+ positivepar = c("kinpar"),
+ cohspec = list(type = "irf"))
```

The vector `kinpar` contains the starting values for five kinetic components. These parameters are constrained to positive values during fitting by including "kinpar" in the vector argument `positivepar`. The `seqmod` argument is by default set to `TRUE`, and determines that the kinetics are described with a sequential compartmental model. The `irfpar` argument gives starting values for the parameters of the default Gaussian IRF model. The `parmu` argument gives starting values for the default model for dispersion in terms of a polynomial, in this case of first-order. The `lambdac` argument gives the center wavelength for the polynomial description of dispersion. The `cohspec` argument determines that a model for a coherent artifact/scatter component is to be added with the time-profile of the IRF.

Fitting and validating the initial model

In the absence of information regarding the effect of laser intensity on the underlying system, the model initialized in Section 5.5.6 may be fit to both dataset Ψ_1 and dataset Ψ_2 simultaneously, with the addition of a dataset scaling parameter Θ_L to account for the difference in amplitude between datasets due to laser intensity.

A call to the fitting function `fitModel` fits the initial model `model1` to both datasets, with the argument `modeldiffs` specifying per-dataset differences in the applied model. By visual inspection of the data it is clear that the intensity of dataset Ψ_2 is approximately half that of dataset Ψ_1 . It is not obvious from visual inspection what other per-dataset differences to include in the model. Therefore `modeldiffs` only specifies that the second dataset is scaled to .5 times the first dataset (via the argument `dscal = list(list(to = 2, from = 1, value = .5))`). The input argument `opt` specifies plotting and output options.

```
R> denRes <- fitModel(data = list(denS4, denS5), list(model1),
+ opt = kinopt(iter = 5, divdrel = TRUE, linrange = .2,
+ makeps = "den1", selectedtraces = c(1,5,10), plotkinspec = TRUE,
+ output="pdf", xlab = "time (ps)", ylab = "wavelength"))
```

Figure 5.8 shows the fit of selected traces after fitting for five iterations. Large misfits are present. Misfits around time zero are likely to be due to an

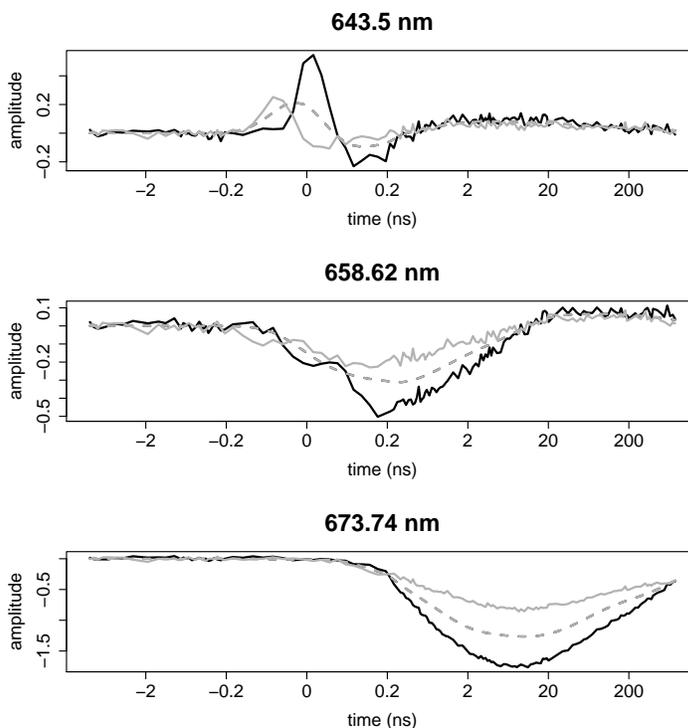


Figure 5.8: A plot of three selected traces resulting from the fit of the initial model to the data $\Psi = \{\Psi_1, \Psi_2\}$. Black represents data Ψ_1 (solid) and the fit of the initial model to Ψ_1 (dotted); Grey represents data Ψ_2 (solid) and the fit of the initial model to Ψ_2 (dotted). The RMS error associated with this fit is .040.

insufficient IRF model, whereas misfits at later times are likely to be due to differences in the kinetics of the two measured datasets. The root mean square (RMS) error associated with this fit is .040. Note that the datasets appear to have the same amplitudes due to the argument `divrel` to `opt` in the call to `fitModel`, which divides Ψ_2 and the fit of the model to Ψ_2 by the estimate for the dataset scaling parameter.

Model refinement and re-validation

Fitting the IRF location parameters and the slowest two kinetic decay rate parameters per-dataset attempts to address the inadequacies of the model identified in validation.

```
R> denRes <- fitModel(data = list(denS4, denS5),
+ modspect = list(model1),
```

```

+ modeldiffs = list(dscal = list(list(to = 2,from = 1,
+ value = .457)),
+ free = list(
+ list(what = "irfpar", ind = 1, dataset = 2, start = -.1932),
+ list(what = "kinpar", ind = 5, dataset = 2, start = .0004),
+ list(what = "kinpar", ind = 4, dataset = 2, start = .0159)
+ )),
+ opt = kinfopt(iter = 5, divdrel = TRUE, linrange = .2,
+ xlab = "time (ps)", ylab = "wavelength", output = "pdf",
+ makeps = "den2", selectedtraces = c(1,5,10)))

```

The resulting fit is improved over the initial model, though a misfit remains evident at early times. The data has an oscillatory nature in some traces indicative of a contribution from a coherent artifact, as evident in Figure 5.9. The RMS error associated with this fit is .027.

A satisfactory model

The model for the coherent artifact has followed the time profile of the IRF, which is not sufficient to account for the oscillatory nature of its apparent contribution to the data. Therefore the coherent artifact model is replaced via a re-definition of the model (which could also be performed by specifying a different model prescription for both datasets in the model differences list):

```

R> model2 <- initModel(mod_type = "kin",
+ kinpar = c(7.9, 1.08, 0.129, .0225, .00156),
+ irfpar = c( -.1018, 0.0434),
+ parmu = list(c(.230)),
+ lambdac = 650,
+ positivepar = c("kinpar", "coh"),
+ cohspec = list(type = "seq", start = c(8000, 1800)))

```

The new model is fit to the datasets. Again the IRF location and the two slowest data rates are fit per-dataset. The estimated value of the data scaling parameter between Ψ_1 and Ψ_2 obtained in fitting the refinement of the initial model is used as a starting value.

```

R> denRes <- fitModel(data = list(denS4, denS5), list(model2),
+ modeldiffs = list(dscal = list(list(to = 2,from = 1,
+ value = .457)),
+ free = list(
+ list(what = "irfpar", ind = 1, dataset = 2, start = -.1932),
+ list(what = "kinpar", ind = 5, dataset = 2, start = .0004),
+ list(what = "kinpar", ind = 4, dataset = 2, start = .0159)
+ )),
+ opt = kinfopt(iter = 5, divdrel = TRUE, linrange = .2,
+ makeps = "den3", selectedtraces = c(1,5,10), plotkinspec = TRUE,

```

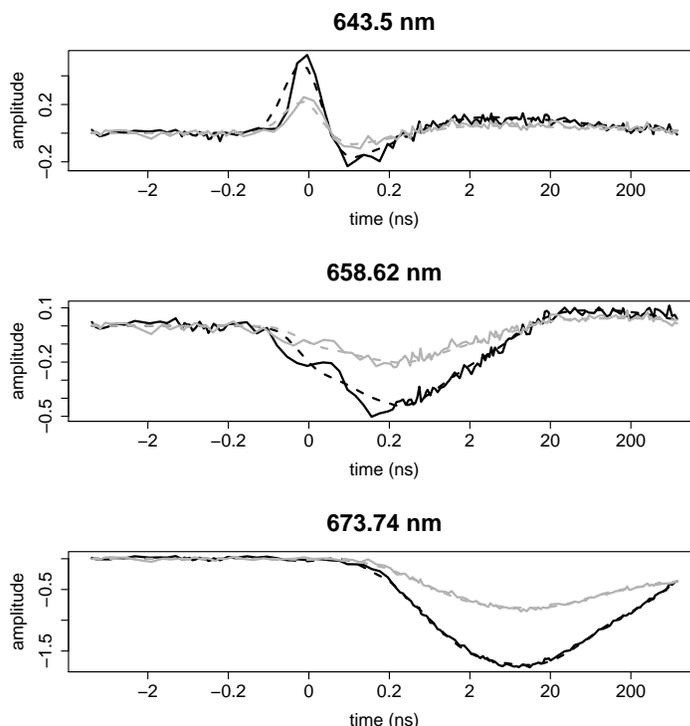


Figure 5.9: A plot of three selected traces resulting from the fit of the refined model to the data $\Psi = \{\Psi_1, \Psi_2\}$, with the IRF location parameters and the two slowest decay rates fit per-dataset. Black represents data Ψ_1 (solid) and the fit of the initial model to Ψ_1 (dotted); Grey represents data Ψ_2 (solid) and the fit of the initial model to Ψ_2 (dotted). The RMS error associated with this fit is .027.

```
+ stderrclip = TRUE, kinspecerr = TRUE, output = "pdf",
+ xlab = "time (ps)", ylab = "wavelength",
+ breakdown = list(plot = c(643.50, 658.62, 677.52))))
```

The resulting parameter estimates mapped to a hierarchical representation of the model are shown in Figure 5.13. For clarity the standard error estimates returned by `nls` have been omitted. These standard errors are typically 1-5 in the last significant digit reported, except for θ_C (which is not of interest) where they are huge.

Figure 5.11 shows just the estimates of the kinetic decay rates, with the color of each component the same as in Figure 5.14 and Figure 5.12. Figure 5.12 shows the contribution to the fit of the kinetic decay components and the coherent artifact (which is in pink) at three different wavelengths. The fit associated with the model after five iterations shown at three selected wavelengths in Figure 5.10

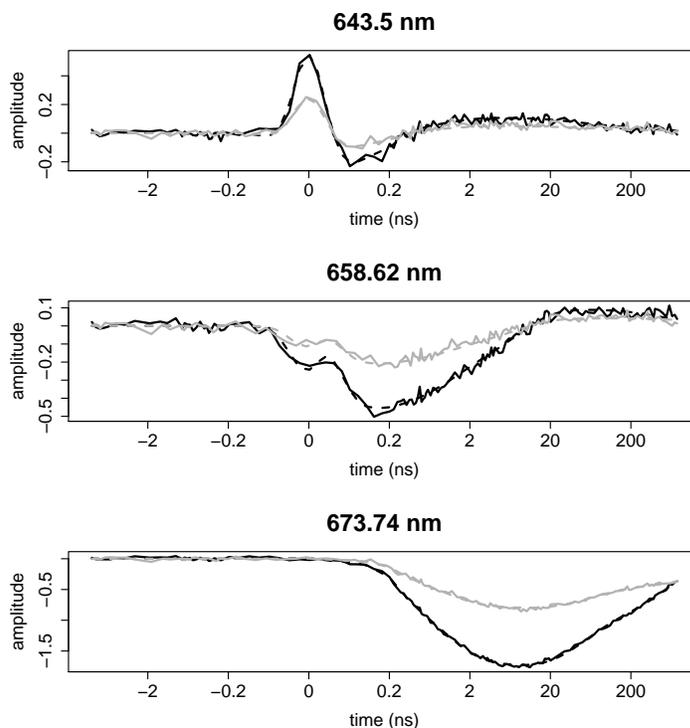


Figure 5.10: A plot of three selected traces resulting from the fit of a satisfactory model to the data $\Psi = \{\Psi_1, \Psi_2\}$, with the IRF location parameters and the two slowest decay rates fit per-dataset, and under application of an oscillatory coherent artifact model. Black represents data Ψ_1 (solid) and the fit of the initial model to Ψ_1 (dotted); Grey represents data Ψ_2 (solid) and the fit of the initial model to Ψ_2 (dotted). The RMS error associated with this fit is .025.

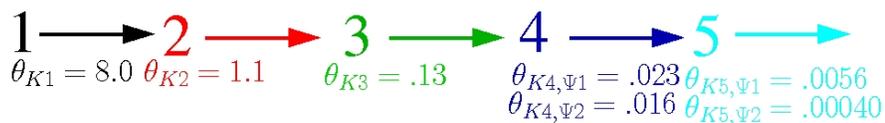


Figure 5.11: Transitions between the five components of the compartmental model are now fit with decay rates as labeled; the slowest two decays are fit independently for each dataset.

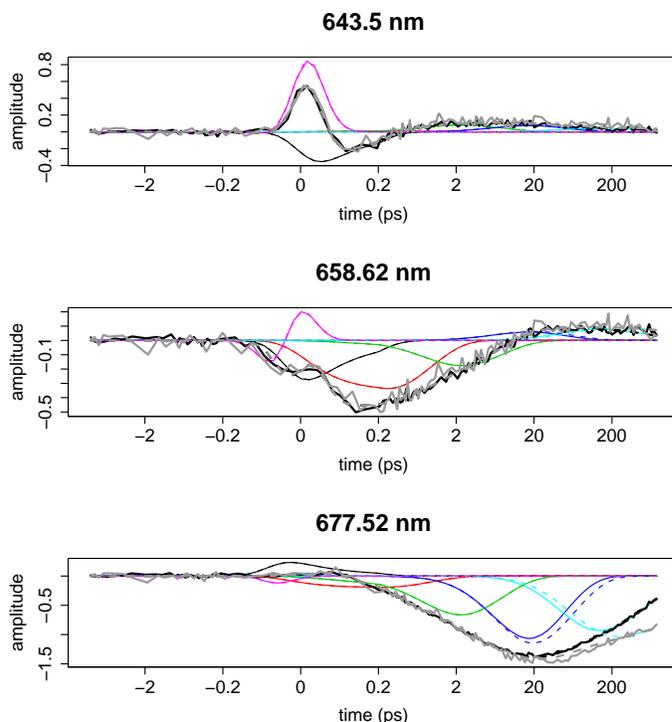


Figure 5.12: Contributions to fit per component show evolution. Pink represents the coherent artifact component; key to other colors is in Figure 5.11. Dashed lines indicate the fit of the second dataset Ψ_2 , which has slower decay rate estimates.

is deemed acceptable. The RMS error associated with this fit is .025. The spectra associated with the kinetic decay components (Figure 5.14) have physically plausible shapes. The discovery of an appropriate model for the data allows the differences in the slow rate constant estimates between datasets to be attributed to the effect of a difference in laser power. The parameter Θ_L is also interpretable as quantifying this difference.

5.6 Spectral models

Spectral models are those models in which the nonlinear parameters determine the matrix of spectra E . The spectral bandshapes are typically described in terms of a linear superposition of standard band shapes (e.g., Gaussian, Lorentzian, Voigt, skewed Gaussian) or in terms of splines. In the case that

a superposition of standard band shapes is used, each column of E , ϵ_l , is modeled as

$$\epsilon_l = (\text{amp}_1)(g_{1l}) + \dots + (\text{amp}_h)(g_{hl}) \quad (5.23)$$

where $\text{amp}_1, \dots, \text{amp}_h$ are amplitude parameters, and g is the band shape function. The concentration profiles are then estimated as `clp`.

Spectral models are represented in **TIMP** with the class `spec`. Model parametrization options for spectral models are here outlined.

5.6.1 Bandshape models

The most commonly applied bandshape model is a superposition of skewed Gaussians. This underlying model for E is chosen because it is a simple model capable of representing real spectra in practice and because the use of (skewed) Gaussians to represent spectral shapes is wide-spread, (see, e.g., van Stokkum *et al.* (2004) and van Stokkum (1997) and references therein).

The model for ϵ_l under a single skewed Gaussian distribution where $\bar{\nu} = 10^7/\lambda$ is

$$\epsilon_l(\bar{\nu}_{max}, \Delta\bar{\nu}, b) \equiv \bar{\nu}^{-n} \exp\left(-\ln(2) \left(\ln\left(1 + \frac{2b(\bar{\nu} - \bar{\nu}_{max})}{\Delta\bar{\nu}}\right)/b\right)^2\right), \quad (5.24)$$

except if $1 + (2b(\bar{\nu}_i - \bar{\nu}_{max}))/\Delta\bar{\nu} \leq 0$, in which case $\epsilon_l(\bar{\nu}_{max}, \Delta\bar{\nu}, b) \equiv 0$. When skewness $b = 0$, (and the skewed Gaussian distribution reduces to the Gaussian distribution), $\bar{\nu}_{max}$ corresponds to the maximum of the distribution, and $\Delta\bar{\nu}$ corresponds to the full width at half maximum (FWHM). When $b \neq 0$, $\bar{\nu}_{max}$ and $\Delta\bar{\nu}$ do not have this exact correspondence. Typically, $n = 1$ for absorption and $n = 5$ for emission, as discussed by van Stokkum *et al.* (2004). The FWHM may then be calculated as $\Delta\bar{\nu} \sinh(b)/b$. The average wavenumber of the skewed Gaussian is given by

$$\bar{\nu}_{avg} = \bar{\nu}_{max} + \frac{\Delta\bar{\nu}}{2b} \left(\exp\left(-\frac{3b}{4\ln(2)}\right) - 1\right) \quad (5.25)$$

A linear superposition of spectra as in Equation 5.24 is a common model for a single spectrum ϵ_l .

Specification in TIMP: Bandshape model

The list `specpar` contains vectors of starting values for spectral parameters; the number of vectors gives the number of components in the resulting spectral model. Each vector contains the parameters associated with a component. e.g., `specpar = list(c(20000, 3000, .3, 21000, 2000, .4), c(18000, 1000, .2))`; the parameters in each vector are grouped `c(location_spectra, width_spectra, skew_spectra)`. The location and width parameters are given in wavenumbers. Note that this means that each component may be modeled with a superposition of many skewed Gaussians.

The character string `specfun` specifies the model used in the description of bandshapes.

Inclusion of the amplitude parameters is not yet implemented, so that all skewed bandshapes contributing to a single component spectrum ϵ_l contribute equally.

5.6.2 Dependence of bandshape parameters on independent variable

It is often desirable to model bandshape parameters of spectra as a function of the independent variable in which spectra are resolved, (e.g., time or temperature). Parametrization of this variation is a means of studying protein conformational stability (for temperature-resolved data), and vibrational relaxation (for time-resolved data).

Dependence on the variable in which spectra are resolved may often be well-described by a polynomial model. Where θ is some parameter contributing to the determination of a bandshape (i.e., either a location, width, skewness or amplitude parameter), t is a value of the independent variable with which spectra are resolved, v is the number of coefficients parametrizing the polynomial (in addition to θ), t_{ref} is the center-point of the polynomial, and $\alpha_1, \dots, \alpha_v$ parametrize the dependence, then

$$\theta(t) = \theta + \sum_{i=1}^v \alpha_i ((t - t_{\text{ref}})/100)^i. \quad (5.26)$$

An exponential model of dependence on the variable in which spectra are resolved is also often of interest. Then where t_{ref} is a reference time, and other variables are as in the polynomial description,

$$\theta(t) = \theta + \alpha_1 \exp(-\alpha_2(t - t_{\text{ref}})). \quad (5.27)$$

A multiexponential model may also be of use in some cases. For the biexponential case

$$\theta(t) = \theta + ((\theta_0 - \theta) \frac{\exp(-k_1(t - t_{\text{ref}})) + \alpha_2 \exp(-k_2(t - t_{\text{ref}}))}{1 + \alpha_2}), \quad (5.28)$$

where $\theta(t_0) = \theta_0$ and $\theta(\infty) = \theta$.

For the general case of a multiexponential decay,

$$\theta(t) = \theta + (\theta_0 - \theta) \frac{\sum (\alpha_i \exp(-k_i(t - t_{\text{ref}})))}{\sum \alpha_i} \quad (5.29)$$

where $\alpha_1 \equiv 1$.

Specification in TIMP: Dependence of bandshape parameters on independent variable

The character string `parmufunc` determines the function modeling dependence of the bandshape parameters on the independent variable. Options are "poly" for the polynomial case (Equation 5.26) "exp" for the single exponential description (Equation 5.27) and "multiexp" for the multiexponential description (Equation 5.29).

The numeric `specref` gives the index of the center variable, t_{ref} in Equations 5.26, 5.27 and 5.29).

The list `specdispindex` defines those indices of `specpar` whose dependence on the variable in which spectra are resolved is to be modeled. For example, `specdispindex = list(c(1,1), c(1,2), c(1,3))` indicates that parameters 1-3 of spectra 1 are to be modeled as variable.

The list `specdisppar` contains vectors of the parameters describing time-dependence. One vector of parameters is given for each vector of indices in `specdispindex`. These parameters describe a polynomial time-dependence by default. There are three ways to interpret these vectors, depending on the value of `parmufunc`. If `parmufunc` is "poly" for the polynomial case, the vectors are of the length of the desired degree of the polynomial parametrization; e.g., use `specdisppar = list(c(-2000, 1, .1), c(1, .1, .01), c(.2, .1))` for a 3rd order dependence of two spectral parameters, and a 2nd order dependence on one spectral parameter. If `parmufunc` is "exp" the first parameter is a linear coefficient and the second is a rate; if all but the first vector have the rate omitted then rates will be linked across all parameters that are time-dependent; otherwise rates will be fit per-parameter that is dependent on the variable with which spectra are resolved. If `parmufunc` is "multiexp" an arbitrary number of vectors of coefficients and rates in the form $c(\alpha_1, k_1, \alpha_2, k_2, \dots)$ may be specified as elements of the `specdisppar` list.

5.6.3 Case study: Time-dependence of spectral parameters

A spectral model is fit to the time-resolved dataset shown in Figure 5.15. A goal is the parametric description of the relaxation of the bandshape in time. As in the kinetic modeling case study, (Section 5.5.6) this section describes the use of **TIMP** for interactive model discovery.

Data input

The data $\Psi = \{\Psi_1\}$ is read into **TIMP** in the *time explicit* format described in Appendix 5.C via the command

```
R> psi_1 <- readData("psitspec.txt")
Read 1 item
Read 181063 items
```

where Ψ_1 is stored in the file "psitspec.txt", (distributed with the package).

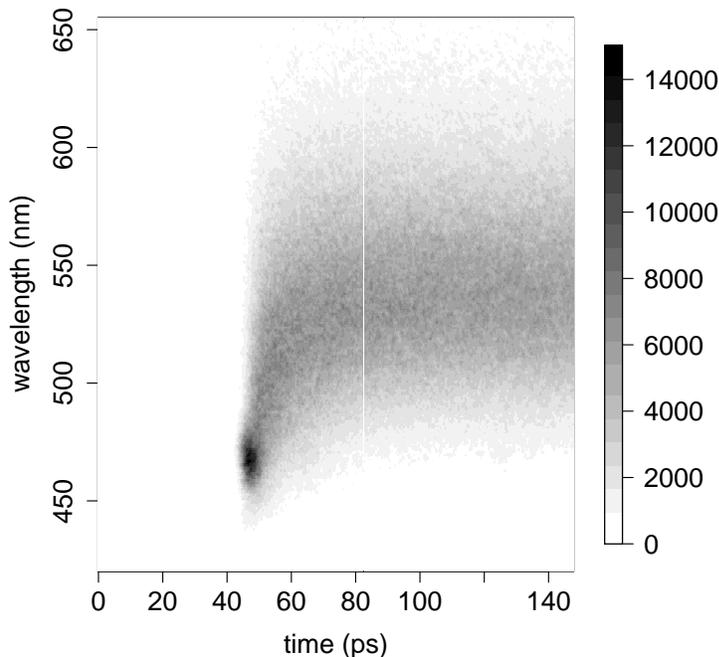


Figure 5.15: A dataset of time-resolved spectra. We are interested in application of a spectral model to the selection of this data shown in Figure 5.16.

Data preprocessing

The dataset was truncated to the wavelength range 440-640 nm, and to times after 53 ps (from time indices 178 to 478), since we are only interested in modeling processes occurring in this range.

```
R> psi_1_full <- preProcess(psi_1, sel_time = c(178, 478),
+ sel_lambda_ab = c(440, 640))
```

To expedite parameter estimation only every fifth timepoint is sampled from the selected dataset. When a model likely to be satisfactory is identified on the sampled dataset, it may be applied to the unsampled selected dataset.

```
R> psi_1_sampled <- preProcess(psi_1, sel_time = c(178, 478),
+ sel_lambda_ab = c(440, 640), sample_time = 5)
```

Figure 5.16 shows the selected and sampled dataset resulting from the above calls to the `preProcess` function.

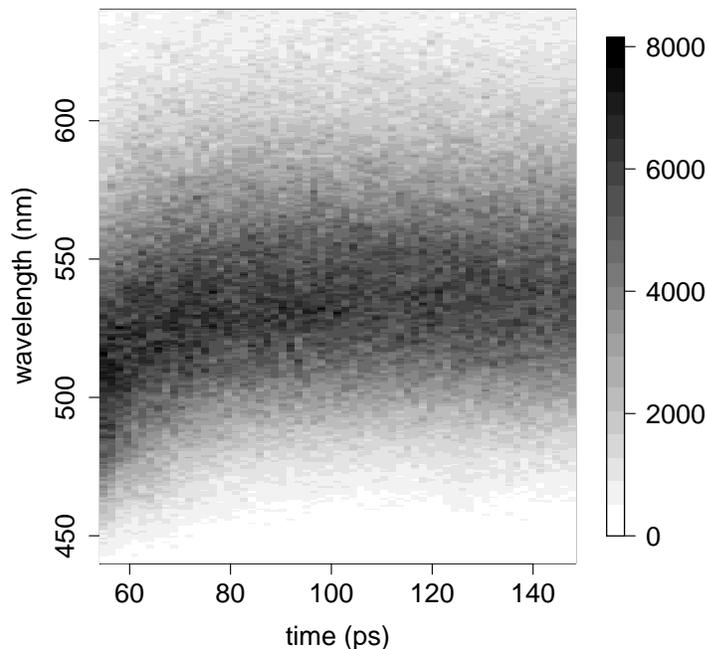


Figure 5.16: The dataset in Figure 5.15 selected to include only the range of times and wavelengths of interest for modeling, and sampled to include only every fifth timepoint to expedite parameter estimation.

Initial model: Polynomial dependence of spectral variables on time

A spectra with bandshape model comprised of a single skewed Gaussian is initialized, first with linear polynomial dependence of the spectral variables on time.

```
R> model_polylin <- initModel(mod_type = "spec",
+ specpar = list(c(20000,3100,-.3) ), specdisp = TRUE,
+ specdispindex = list(c(1,1), c(1,2), c(1,3)),
+ specdisppar = list(c(-2000),c(1), c(.2)),
+ specref = 53, specfun = "gaus")

R> res_polylin <- fitModel(data = list(psi_1_sampled),
+ modspect = list(model_polylin),
+ opt = specopt(iter = 7, linrange = 20, stderrclp = TRUE,
+ plotkinspec = TRUE, kinspecerr = TRUE,
+ makeps = "polylin",
```

```

+ title = "Polynomial parameterization of time dep.",
+ selectedspectra = seq(1, psi_1_sampled@nt, by = 7),
+ residplot = TRUE,
+ xlab = "time", ylab = "wavelength")

```

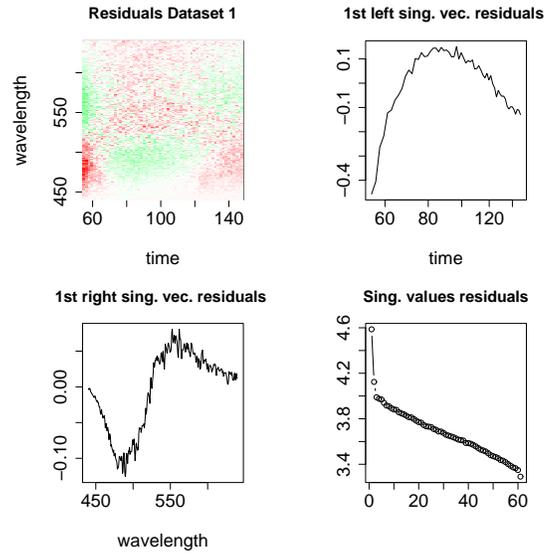


Figure 5.17: Residuals associated with the fit of the initial spectral model in which dependence of the spectral bandshape on parameters is described with a first-order polynomial. The first left and right singular vector and the singular values associated with their SVD are also plotted. Note the structure in the first left singular vector of the residuals, indicating inadequacy in the model fit. The RMS error associated with this fit is 414.

Structure in the residuals as evidenced by their SVD in Figure 5.17 indicates an inadequate model. A different parametrization of the dependence of spectral parameters on time will be applied as a model refinement.

Refined model: Exponential dependence of spectral variable on time

The initial model is refined to describe an exponential dependence of spectral variables on time, in an attempt to address the inadequacy in the fit of the initial model. The applied model is such that α_2 from Equation 5.27 is equated for all spectral bandshape parameters.

```
R> model_exp_linkedrates <- initModel(mod_type = "spec",
+ specpar = list(c(18000, 3200, -.1)), specdisp = TRUE,
+ specdispindex = list(c(1,1), c(1,2), c(1,3)),
+ specdisppar = list(c(600,1/20), c(400), c(.1)),
+ specref = 53, specfun = "gaus", parmufunc = "exp")

R> res_model_exp_linkedrates <- fitModel(data = list(psi_1_sampled),
+ modspec = list(model_exp_linkedrates),
+ opt = specopt(iter = 5, linrange = 20, residplot = TRUE,
+ makeps = "explinked", stderrclp = TRUE,
+ title = "Exponential parameterization of time dep., linked rates",
+ plotkinspec = TRUE, kinspecerr = TRUE, superimpose = 1,
+ selectedspectra = seq(1, psi_1_sampled@nt, by = 7),
+ xlab = "time", ylab = "wavelength"))
```

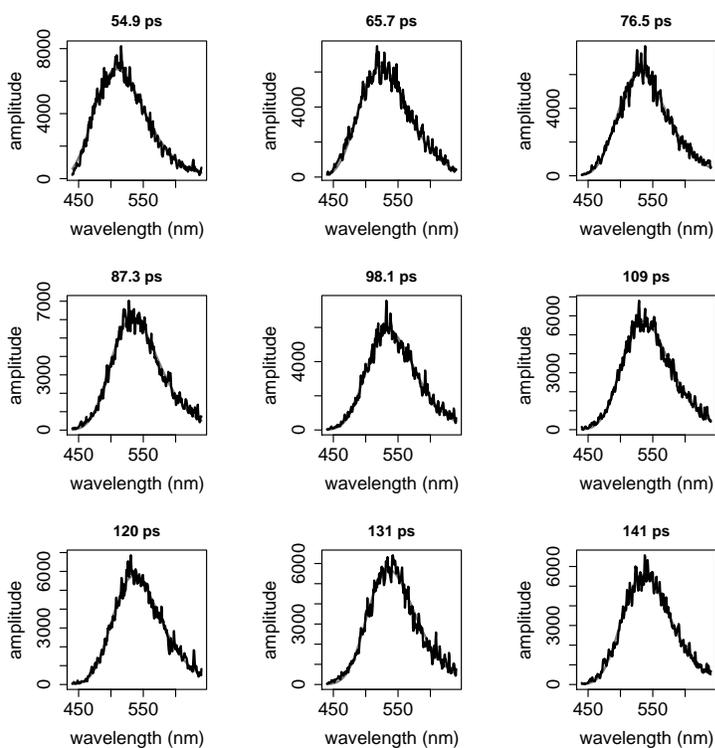


Figure 5.18: (Left) A plot of selected spectra resulting from the fit (dashed line) of a spectral model in which time-dependence of spectral bandshape parameters is modeled with an exponential function to data $\Psi = \{\Psi_1\}$ (solid line). Each sub-plot represents the data at the timepoint on the left axis. The RMS error associated with this fit is 312.

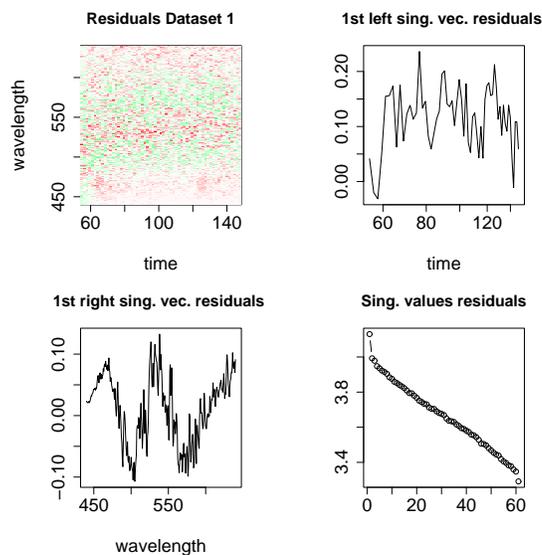


Figure 5.19: Residuals associated with the fit of the spectral model, and the first left and right singular vector and the singular values associated with their SVD. Note the lack of structure in the first left singular vector of the residuals, a sign that the model fit is satisfactory. There remains some structure in the first right singular vector, but we accept the model as satisfactory nonetheless.

The fit of the refined spectral model that employs an exponential description of time-dependence of spectral bandshape parameters is well-fit to the data, as evidenced by plots showing the fit of the model to the data in Figure 5.18. Further evidence for the satisfactory model fit is contained in plots regarding the residuals in Figure 5.19. An SVD decomposition of the residuals shows little structure in the first left singular vector. Some structure remains evident in the first right singular vector, indicating that there remains room for improvement in the model. Despite this, we conclude that the model fit is satisfactorily descriptive of the data. The exponential rate estimate $\alpha_1 = .07$ is therefore useful as a descriptor of the relaxation of the spectral bandshape parameters in time.

5.6.4 Case study: Temperature-dependence of spectral parameters

Protein conformational stability may be studied spectroscopically by monitoring spectra as a function of temperature. For the study considered here, a protein in solution was heated from 25°C to 75°C. Absorption spectra were measured at 57 temperatures as the temperature rose. Then the protein was cooled to 25°C, and the absorption spectra was re-measured. During heating the protein is denatured. The data is shown in Figure 5.20.

The spectral model fit to the data has two components, comprised of a linear superposition of three and two skewed Gaussian shapes, respectively. The location and width parameter of each skewed Gaussian is modeled as having a linear temperature-dependence. An increase in concentration of the second component and a decrease in concentration of the first component will be interpreted as indicative of a heat-induced conformational change in the protein.

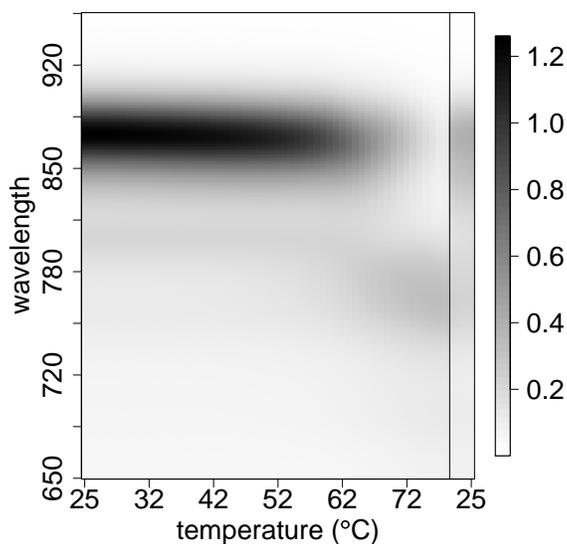


Figure 5.20: A dataset of temperature-resolved spectra. Temperatures to the left of the vertical black line represent the absorption spectra of a protein as it is heated. Temperatures to the right of the vertical black line represent the same protein after cooling.

Data input

The data Ψ is read into R with the command

```
R> G9 <- readData("spectresG9")
Read 1 item
```



```

+           c(1, 12), c(2, 1), c(2,2), c(2, 5),
+           c(2, 6), c(2, 8)),
+   specdispar = dispar, specref = 25,
+   specfun = "gaus", nupow = 1)

```

The model is fit to the data with the command

```

R> ranres <- fitModel(data = list(G9_p), modspec = list(modelF),
+                   opt = specopt(iter = 10, makeps = "final2",
+                   xlab = "temperature", ylab = "amplitude",
+                   title = "Linear dispersion model",
+                   plotkinspec = TRUE, kinspecerr = TRUE,
+                   stderrclp = TRUE))

```

The concentration profiles that result appear in Figure 5.21. The rise of the second component and the fall of the first is interpreted as indicative of a conformational change in the protein.

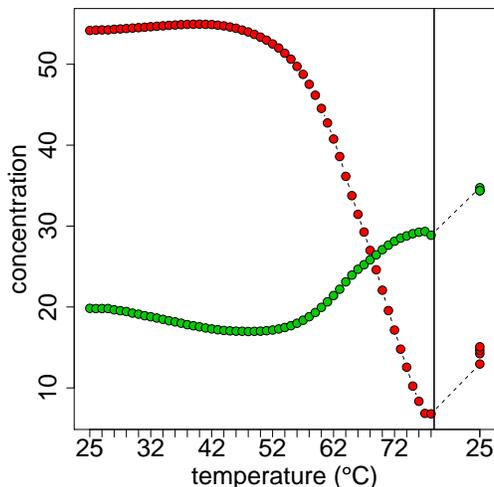


Figure 5.21: Concentration profiles of two components estimated using a spectral model. Temperatures to the left of the vertical black line represent the absorption spectrum of a protein as it is heated. Temperatures to the right of the vertical black line represent the same protein after cooling.

The spectra that are estimated depend on the temperature considered. Figure 5.22 shows the spectra at nine temperatures as the protein is heated.

The model described in Section 5.4 represents the data very well, with the exception of the spectra taken after the protein has been heated and then re-cooled, as shown by data at a selection of temperatures in Figure 5.23. After heating (and denaturation), irreversible changes in the sample occur, so that

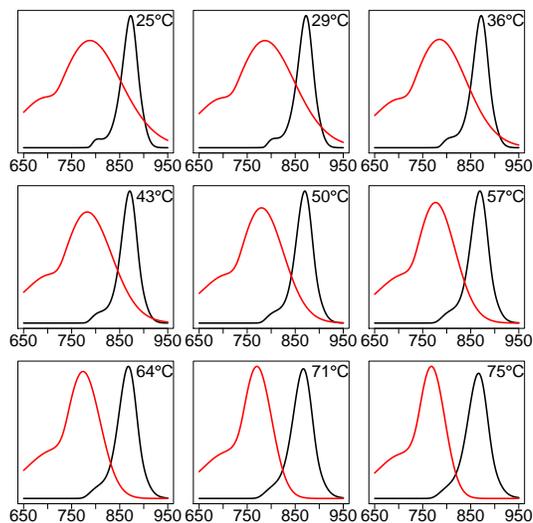


Figure 5.22: Estimated absorption spectra at nine temperatures as the system giving rise to the measurements is heated.

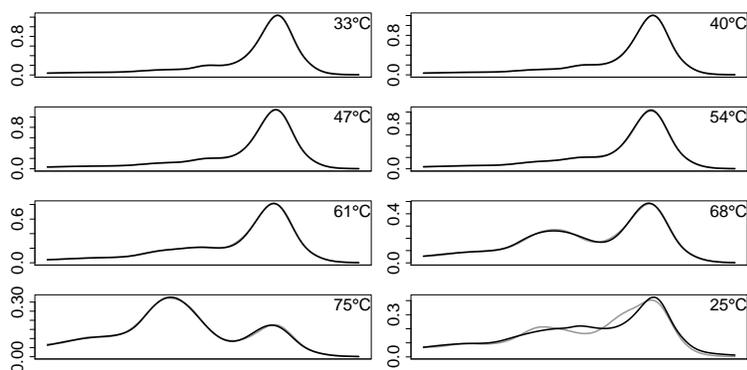


Figure 5.23: Data (grey) and model fit (black) at selected temperatures. Only the data after cooling contains significant misfits.

the data taken at 25°C before and after heating require different models. The formulation of an appropriate model for the data after cooling remains as future work. Note that Pandit *et al.* (2001) discuss an association model for assembly of the antenna with intermediate steps. We find hints of intermediates only in the cooling spectra, where the chosen two-component spectral model is not sufficient.

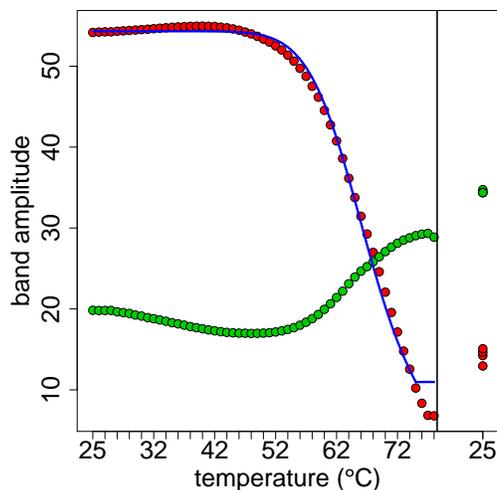


Figure 5.24: Estimated band amplitude curves derived from the model fit in Section 5.6.4 as points. A thermodynamic model is fit to the band amplitude curve for the first component in the heating regime, and is shown as a solid curve. Temperatures to the left of the vertical black line represent the absorption spectra of a protein as it is heated. Temperatures to the right of the vertical black line represent the same protein after cooling.

Extraction of thermodynamic information from model fit

The area of the bandshape for each component at each temperature is used to scale the corresponding concentration profile, resulting in a band area curve that is indicative of the oscillator strength f_{ab} of the component per-temperature (van Stokkum *et al.*, 1995). Oscillator strength f_{ab} compares the intensity of absorption to that expected from a three-dimensional harmonic oscillator, which can be shown to be

$$f_{ab} = 4.315 \times 10^{-9} \int \epsilon(\nu) d\nu \quad (5.30)$$

(Cantor and Schimmel, 1980).

The transition defined by the decrease of the band area curve associated with the first component and the increase in the band area curve associated with the second component is an important feature of the fitted model. Whether the band area curves as fitted correspond to those expected from a pure two-state transition is of interest. A thermodynamic model for a two-state transition in band area curves may be written

$$ba(T) = \frac{a}{1 + \exp\left(-\Delta H_m \left(\frac{1}{T} - \frac{1}{T_m}\right)\right)} \quad (5.31)$$

where ba is the band area curve, T_m is the temperature marking the midpoint of the transition, a is an amplitude parameter indicative of the difference between the band areas before and after the transition, and ΔH_m is the apparent enthalpy change at T_m divided by the gas constant R van Stokkum *et al.* (1995).

Taking ba as the estimated band area code for the first component for the temperatures that represent the heating regime, we obtain estimates for T_m , a , and ΔH_m in Equation 5.31 that are optimal under least squares criteria. Figure 5.24 shows the band amplitude curves, with the fit of the thermodynamic model shown as a line superimposed on component 1. The fitted curve is a rough approximation of the band area curve. The small misfit indicates some complexity in the transition not described by the simple two-state thermodynamic model.

5.7 Extension of supported model types

TIMP has been designed to allow for rapid implementation of new options and new model types. The process of adding an entirely new model type `newmodel` can be described in terms of four steps:

- definition of a new class for the model type `newmodel` that inherits from `dat`
- for every slot `par` representing a list/vector of nonlinear parameter value starting values in the definition of `newmodel` adding a slot for a list/vector of the same name `par` to the class `theta` so that the vector of parameters Θ can be inferred, and so that updated parameter estimates can be plugged back into the model specification each iteration
- definition of methods for `residPart` (and/or `getClpIndepX` depending on whether `clp` are involved) that supply a prescription for the calculation of residuals for a single dataset Ψ_q given a model
- definition of desired output plots and other information via a method for `plotter`
- definition, if desired, of a new subclass inheriting from class `opt` that allows options for plotting and model optimization that are specific to the new model type to be defined

The remaining code of **TIMP** should not require any modification.

5.8 Conclusions

TIMP, an R package for interactive scientific model discovery for multiway spectroscopy data has been introduced. The design of the package has been outlined. The partitioned variable projection algorithm that is central to solving separable nonlinear least squares parameter estimation problems with **TIMP** was presented.

General options for models in **TIMP** were introduced, along with options specific to kinetic and spectral model types. A case study in application of a kinetic model to two datasets simultaneously illustrated many kinetic model options. A case study in application of a spectral model illustrated many spectral model options.

TIMP is in active development. Future work includes the development of new model types for data collected in multipulse laser experiments, anisotropy experiments, and experiments designed to extract information on photocycles, as well as the implementation of additional options to support model specification and validation. The development of a GUI to support interactivity is also planned.

Acknowledgments

Mikas Vengris, Denitsa Grancharova and Rienk van Grondelle provided the data modeled in Section 5.5.6. Rob Koehorst, Bart van Oort, Sergey Laptinok, Ton Visser and Herbert van Amerongen provided the data modeled in Section 5.6.3. Maïté Paternostre and Jérôme Seguin provided the data in Section 5.6.4. Joris Snellenburg is thanked for constructive comments on the text. Uwe Ligges and Martin Mächler collaborated in the implementation of the `nls` options described in Section 5.B. Achim Zeileis contributed helpful suggestions regarding the figures.

5.A Partitioned vs. unpartitioned variable projection algorithms

This appendix considers in detail the memory requirements of the partitioned variable projection algorithm introduced in Section 5.2.4, and compares these requirements to those of the standard (unpartitioned) variable projection algorithm that applies to data that is in vector form. In order to fit a model for matrix data using the standard variable projection algorithm, the data and the model must first be vectorized. Let us consider the case that a different function C_j is used to describe each column ψ_j of a matrix dataset, where $\psi_j = C_j(\Theta)E[j, \cdot]$ as in Equation 5.5. Then to convert the data and the model into a vectorized format, we can use $\text{vec}(\Psi) = \text{vec}(C_{\text{super}}E^T I_n) = (I_n \otimes C_{\text{super}})\text{vec}(E^T)$ where

$$C_{\text{super}} = \begin{bmatrix} C_1(\Theta) \\ C_2(\Theta) \\ \vdots \\ C_n(\Theta) \end{bmatrix}. \quad (5.32)$$

However, large memory resources are required to form and operate on the matrix $(I_n \otimes C_{\text{super}})$ for the case that n is large. This is a significant disadvantage and may prohibit the use of the approach (as e.g., in the experience of Verveer *et al.* (2000)).

The ability to apply the variable projection functional without operating on large matrices is the main motivation for introduction of the partitioned variant of the algorithm. `PARTITIONEDVARPRO` and the standard implementation of variable projection, as found for instance in the `plinear` function of `nls`, return the same results. The algorithms differ only in the memory resources required.

5.A.1 Empirical comparison of partitioned and unpartitioned variable projection algorithms

The `nls` function of R allows application of the variable projection algorithm via the `plinear` option. We show here how the simple kinetic modeling problem considered in Section 5.1.3 may be fit using `nls` and the `plinear` option. We then compare the memory requirements under `plinear` to those under the partitioned variable projection algorithm implemented in `TIMP`.

We assume that a dataset `Psi_q` is simulated in R as described in Section 5.1.3. Commands to simulate the dataset and set up the workspace are contained in full in the file “`memory_prof_plin.R`” (distributed with the package). In order to use `nls` under the `plinear` option, the model for concentrations is placed into a function `calcC`, which is also found in `TIMP`, as follows.

```
"calcC" <- function (k, t)
{
  tfun <- function(t,k) exp(-k*t)
  ## get C with tfun
```

```

  mapply(tfun, k = k, MoreArgs = list(t = t))
}

```

The sum-of-exponentials model can then be fit to the data using `calcC`, with the `plinear` option using the standard variable projection functional to determine the residuals, and with the spectra as conditionally linear parameters.

```

R> psi_q_vector <- as.vector(Psi_q)

R> onls <- nls(psi_q_vector ~ kronecker(diag(length(wavenum))),
+ calcC(k, t)), data.frame(psi_q_vector),
+ start = list(k = c(.1,2)), alg = "plinear", trace = T)

```

To profile the memory allocated in the course of solving this problem, we apply the `gc` function, (note that more refined memory profiling is possible with the `Rprof` and `Rprofmem` functions under builds of R compiled to enable memory profiling). Before and after the call to `nls`, the results of a call `gc(verbose = TRUE)` on our system are

```

Garbage collection 11 = 9+0+2 (level 2) ...
6.4 Mbytes of cons cells used (58%)
0.9 Mbytes of vectors used (14%)

```

and

```

Garbage collection 100 = 45+26+29 (level 2) ...
6.6 Mbytes of cons cells used (53%)
13.2 Mbytes of vectors used (32%)

```

respectively. This shows that under the `plinear` implementation of variable projection about 12 Mbytes of vector space is allocated in the course of solving this example problem.

To contrast this with the memory allocated under the partitioned variable projection implementation found in **TIMP** on the same system, we load the package and initialize a model object as described in Section 5.1.3 and in the file “`memory_prof_pvarpro.R`”. A call to `gc(verbose = TRUE)` before calling the `fitModel` function that applies the partitioned variable projection algorithm to fitting the sum-of-exponentials model with the spectra as `clp` has the following result

```

Garbage collection 35 = 31+2+2 (level 2) ...
7.5 Mbytes of cons cells used (68%)
1.0 Mbytes of vectors used (16%)

```

Then fitting the model as in Section 5.1.3 with

```

kinetic_fit <- fitModel(data = list(Psi_q_data),
                        modspect = list(kinetic_model),
                        opt = kinopt(iter = 4, plot = FALSE))

```

and subsequently calling `gc(verbose = TRUE)` results in

```
Garbage collection 52 = 46+3+3 (level 2) ...
7.8 Mbytes of cons cells used (62%)
1.5 Mbytes of vectors used (25%)
```

This shows that in the course of applying the partitioned variable projection implementation to the problem, about .5 Mbytes of vector space is allocated, about 20 times less than under `plinear` on the same problem.

The savings in memory allocated via the use of the partitioned variable projection algorithm found in **TIMP** is very significant for problems of interest in the multiway spectroscopy modeling domain. As larger amounts of data are involved the memory requirements of the standard non-partitioned implementation found in the `plinear` option grow so large as to prohibit its use on a modern personal computer, while the memory requirements of the partitioned version of the algorithm found in **TIMP** remain modest.

5.B New nls options

It is often desirable in scientific modeling applications to terminate the iterative optimization of free model parameters when *stopping* criteria are met, as opposed to when *convergence* criteria are met. For instance, it is often desirable to evaluate the fit of a model at a given set of starting estimates, or after fitting for a modest number of iterations. Then the stopping criteria is completion of a maximum number of iterations, after which output is desired, even though the fit may be far from satisfying convergence criteria.

It is often also desirable to examine output in the case that the fitting algorithm encountered a problem and terminated fitting with an error. For instance, if the gradient of the residual vector with respect to nonlinear parameter estimates becomes singular, examination of the current parameter estimates may shed light on how the model can be modified to be better determined with respect to the data.

The R function `nls` is widely applied in scientific model discovery and is used in **TIMP** to iteratively improve nonlinear parameter estimates. Prior to R version 2.5 `nls` did not return output in the case that any of the following conditions are met

- the maximum number of iterations x is met (as specified with `nls(..., control = list(maxiter = x, ...))`)
- the step-size is below the minimum x (as specified with `nls(..., control = list(minFac = x, ...))`)
- a singular gradient occurs

In all these cases return of the output object may be valuable for scientific model discovery, for the reasons sketched above. We have implemented the

option ‘warnOnly’ to determine if an output object is returned in the case that one of the above conditions is met. The implementation of this option has been incorporated into R version 2.5. A logical slot in the class `nls.control` is used to toggle the ‘warnOnly’ option. To output a result object even in the case that an error is triggered, `nls` is called with `nls(..., control = list(warnOnly = TRUE, ...))`.

A better understanding of the residual surface on which optimization occurs is sometimes gained by knowledge of how many times `nls` halves the step-size in the descent direction. We have implemented the option ‘printEval’ to print the number of evaluations (of the step-size) required each iteration; this option is also included in R version 2.5. A logical slot in the class `nls.control` is used to toggle the ‘printEval’ option. To print the number of evaluations required (as well as the achieved convergence tolerance), `nls` may be called with `nls(..., control = list(warnOnly = TRUE, ...))`.

5.C Data formats for input into TIMP

Currently **TIMP** allows the input of data using the `readData` function, and supports the following formats.

5.C.1 Plain format

The plain format is an ASCII file in which the first row represents column labels and the first column represents row labels, and remaining entries represent data. For time-resolved spectroscopy data the first row should represent wavelengths, and the first column represents times. This data is read into **TIMP** via `readData` with the argument `typ` set to ‘plain’. The argument `sep` may be used to specify the delimiter in the case that the ASCII data is not space delimited.

5.C.2 Time explicit format

The *time explicit* format for data input contains 5 lines and then a matrix of data in which each row represents a time profile, and each column represents a measured spectrum.

```

Heading line 1
Heading line 2
Time explicit
Intervalnr 5

```

	t_1	t_2	...	t_m
λ_1	$\Psi(t_1, \lambda_1)$	$\Psi(t_2, \lambda_1)$...	$\Psi(t_m, \lambda_1)$
λ_2	$\Psi(t_1, \lambda_2)$	$\Psi(t_2, \lambda_2)$...	$\Psi(t_m, \lambda_2)$
...
λ_n	$\Psi(t_1, \lambda_n)$	$\Psi(t_2, \lambda_n)$...	$\Psi(t_m, \lambda_n)$

Data matrix elements are space-delimited. ‘Heading line 1’ and ‘Heading line 2’ are two lines that may be filled as desired (e.g., with a data file title).

The string ‘Time explicit’ indicates the data format. The string ‘Intervalnr’ and a scalar m indicates the number of distinct points m at which spectra were measured, (note that the number of wavelengths n need not be specified). The following line contains the real-valued variable values (such as times) t_1, \dots, t_m at which measurements were taken. The first value of each of the remaining lines represents the wavelength at which the concentration profile contained on that row was taken. The rest of each remaining row represents a (space-delimited) concentration profile $\Psi(t_1, \lambda), \Psi(t_2, \lambda), \dots, \Psi(t_m, \lambda)$.

5.C.3 Wavelength explicit format

The *wavelength explicit* format for data input contains 5 lines and then a matrix of data in which each row represents a measured spectrum, and each column represents a time profile.

```

Heading line 1
Heading line 2
Wavelength explicit
Intervalnr 5
       $\lambda_1$        $\lambda_2$       ...   $\lambda_n$ 
 $t_1$        $\Psi(t_1, \lambda_1)$    $\Psi(t_1, \lambda_2)$   ...   $\Psi(t_1, \lambda_n)$ 
 $t_2$        $\Psi(t_2, \lambda_1)$    $\Psi(t_2, \lambda_2)$   ...   $\Psi(t_2, \lambda_n)$ 
...      ...      ...      ...  ...
 $t_m$        $\Psi(t_m, \lambda_1)$    $\Psi(t_m, \lambda_2)$   ...   $\Psi(t_m, \lambda_n)$ 

```

All entries above are space delimited. ‘Heading line 1’ and ‘Heading line 2’ are two lines that may be filled as desired (e.g., with a data file title). The string ‘Wavelength explicit’ indicates the format that the input data is to take. The string ‘Intervalnr’ and a scalar n indicates the number of distinct wavelengths n at which measurements were taken, (note that the number of time points m need not be specified). The following line contains the real-valued wavelengths $\lambda_1, \lambda_2, \dots, \lambda_n$ at which measurements were taken. The first value of each of the remaining lines represent the independent variable value (such as a time) at which the spectrum contained on that row was taken. The rest of each remaining row represents a (space-delimited) spectrum $\Psi(t, \lambda_1), \Psi(t, \lambda_2), \dots, \Psi(t, \lambda_n)$.

5.C.4 FLIM format

Fluorescence Lifetime Imaging Microscopy (FLIM) data is read into **TIMP** in the format described in Chapter 6 of this monograph.

Chapter 6

Fluorescence Lifetime Imaging Microscopy (FLIM) data analysis with **TIMP**¹

6.1 Introduction

This chapter describes the utility of the **TIMP** package for the R language and environment for statistical computing (R Development Core Team, 2008) for the global analysis of images collected by Fluorescence Lifetime Imaging Microscopy (FLIM) experiments. FLIM experiments typically measure the fluorescence of biological objects at 250 nm lateral resolution and with (sub-) nanosecond temporal resolution. FLIM has been widely applied in cell biology to detect interactions between fluorescently labeled biological molecules such as proteins, lipids, DNA and RNA.

One experimental technique that is particularly useful for the study of protein interactions in particular is the detection of Förster Resonance Energy Transfer (FRET). FRET is a bi-molecular process in which the excited-state energy of a donor fluorophore is non-radiatively transferred to a ground-state acceptor molecule by dipole-dipole coupling. The FRET efficiency varies with the inverse 6th power of the distance between donor and acceptor and is usually negligible when the distance is larger than 10 nm. FRET can be identified by a shorter fluorescence lifetime of the donor molecule. That is, it is a fluorescence quenching process. FLIM experiments involving the detection of FRET typically use genetically modified cells in which two intracellular proteins of interest are tagged with variants of the green fluorescent protein (GFP) (Tsien, 1998)

¹A version of this chapter appears as Laptanok, Mullen, Borst, van Stokkum, Apanasovich, and Visser (2007) in the *Journal of Statistical Software*, 18(8), 1-20.

that act as a donor-acceptor FRET pair. Spectral variants cyan fluorescent protein (CFP, donor) and yellow fluorescent protein (YFP, acceptor) are the FRET-pair that is most commonly used in practice. When the fluorescently tagged proteins are within 1-10 nm of each other, FRET occurs. This can be detected by estimating the fluorescent lifetime of the donor in the FRET pair, and observing that the estimated lifetime is shortened in comparison to the lifetime estimated in a control experiment in which only the donor fluorescent tag is used. FRET as measured by FLIM experiments can therefore be used as a “spectroscopic ruler” to map protein-protein interactions inside cells. For recent applications see Barber *et al.* (2005) and Grailhe *et al.* (2006). Suhling *et al.* (2005) have comprehensively reviewed different FLIM methods, FLIM and FRET examples and other FLIM applications.

Well-designed data analysis techniques are required to process the measured FLIM images to estimate the lifetimes associated with the fluorescence. It is often the case that the dynamics of the system are well-described by a model in which a small number of exponential decays with equal decay rates across all pixels underlie the measured fluorescence, with amplitude parameters for the exponentials that vary per-pixel. Then estimates for the amplitude parameters are conditionally linear on estimates for the decay rate parameters, allowing application of the variable projection algorithm (Golub and LeVeque, 1979), which has been shown to have many desirable properties (Golub and Pereyra, 2003; Mullen *et al.*, 2007) for problems of this form. A disadvantage of the variable projection method that has prevented its application in this problem domain (Verwee *et al.*, 2000) is that large memory resources are required. The **TIMP** package contains an implementation of a partitioned variable projection algorithm that returns the same results as the standard variable projection algorithm but requires much less memory, as described in Chapter 5. The ability to apply the variable projection functional to estimation problems in the absence of large memory resources is a primary advantage of the application of **TIMP** in the FLIM image analysis problem domain. A further primary advantage is the support the package provides for visual interpretation and validation of the results of model fit.

The organization of the chapter is as follows. Section 6.2 describes the sum-of-exponentials model that is often used to describe FLIM images and the optimization problem associated with fitting the parameters of this model. Section 6.3 discusses approaches to the parameter estimation task associated with the analysis of FLIM images, including the partitioned variable projection approach employed by **TIMP**. Section 6.4 describes extensions to **TIMP** implemented to support FLIM image analysis. Section 6.5 contains a simulation study in the application of the package to the analysis of datasets inspired by measured FLIM data. Section 6.6 describes the use of **TIMP** to fit a measured FLIM image. Section 6.7 contains conclusions.

6.2 Statistical model for FLIM data

FLIM images represent the decay of fluorescence in time at many different locations in the underlying system. Each location is represented by a pixel x , so that the image may be represented as a matrix

$$\Psi = \begin{bmatrix} & x_1 & x_2 & \dots & x_n \\ t_1 & \psi(t_1, x_1) & \psi(t_1, x_2) & \dots & \psi(t_1, x_n) \\ t_2 & \psi(t_2, x_1) & \psi(t_2, x_2) & \dots & \psi(t_2, x_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_m & \psi(t_m, x_1) & \psi(t_m, x_2) & \dots & \psi(t_m, x_n) \end{bmatrix}. \quad (6.1)$$

Each column of Ψ represents a fluorescence decay in time at a given pixel x .

The decay of fluorescence data in time $\psi(t)$ at pixel x can often be satisfactorily modeled as a sum of n_{comp} first-order kinetic processes convolved with an instrument response function (IRF) $g(t)$, so that

$$\psi_x(t) = \sum_{l=1}^{n_{\text{comp}}} c_l a_{x,l} = \sum_{l=1}^{n_{\text{comp}}} \exp(-t/\tau_l) \oplus g(t) a_{x,l} \quad (6.2)$$

where c_l represents the contribution to the data from process l in time t , $a_{x,l}$, represents the amplitude of decay l at pixel x , and \oplus is the convolution operator. The model parameters to be fit are then the lifetimes τ_l and the associated linear coefficients $a_{x,l}$ representative of intensity.

When the same kinetic processes underlie the fluorescence at all n locations, Equation 6.2 can be applied globally to the image Ψ , so that the deterministic aspects of the data are modeled as

$$\Psi = CE^T = \sum_{l=1}^{n_{\text{comp}}} c_l a_l^T = \sum_{l=1}^{n_{\text{comp}}} (\exp(-t/\tau_l) \oplus g(t)) a_l^T \quad (6.3)$$

where C is a matrix in which column l represents the time-profile of the l th kinetic process, and E is a matrix in which column l represents the intensity of kinetic process l across pixels. Then the parameter estimation task is *global analysis*: estimation of the n_{comp} lifetimes τ associated with the image as a whole and the n_{comp} amplitude parameters $a_{x,l}$ associated with each pixel x (so that $n * n_{\text{comp}}$ amplitude parameters are estimated in total). Under least-squares criteria this is

$$\min \|C(\tau)E^T - \Psi\|^2 \quad (6.4)$$

where $\|\cdot\|$ is the 2-norm. This is an instance of the multi-exponential analysis problem, which is common in physics applications. Its difficulty is well-known, as Istratov and Vyvenko (1999) review.

Ψ represents the number of photons fluorescing from the location represented by pixel x at time t , and is therefore count data, the noise associated with which is assumed to be Poisson distributed. That we fit the data using least squares

criteria introduces a bias, which is small in the case that the counts in the data are large. Maus *et al.* (2001) have investigated this issue. Note that the decay of fluorescence in proteins like CFP and YFP is typically described by two or more exponentially decaying kinetic process each (so that the decay is said to be bi-or-multi-exponential).

6.3 Methods for FLIM data collection and analysis

FLIM data is collected by exciting a sample (such as a cell) to fluoresce (i.e., emit photons) using a laser pulse. After a laser pulse the time t until the first photon is detected at pixel x is measured; The process of applying a laser pulse and recording the time of arrival of the first photon is repeated many times. The resulting histograms of arrival times represent fluorescence decays per pixel x . The FLIM experiments we consider here detect photons at a single wavelength.

Since samples of interest are *in vivo*, the power of the laser light used must be low if the sample is to remain alive throughout the measurement. The use of laser light of low power results in the arrival of few photons per location x . The requirement to keep the sample alive and in the same condition also means that the acquisition time cannot be long, (i.e., the sample cannot be subjected to too many laser pulses).

To improve signal-to-noise ratio (SNR), the time resolution may be decreased, since under a wider histogram channel more photons will be collected. The FLIM experiment is thus always compromising between time resolution and SNR. FLIM experiments that measure process with sub-nanosecond time resolution often have low SNRs (6-15 is typical). Further discussion of the methodology of FLIM experiments may be found in, e.g., Becker and Bergmann (2003).

Methods for the analysis of FLIM data commonly applied (Becker *et al.*, 2001, 2002) fit the model given by Equation 6.2 independently to each of the n pixels in an image. This yields estimates for the decay rates of the n_{comp} kinetic process as well as n_{comp} estimates of the amplitude of each process, *for each pixel in the image*. This is not desirable if the assumption that the same n_{comp} kinetic processes underlie measurements at all pixels is valid.

Data analysis methods that acknowledge the desirability of global analysis, which assumes that the underlying kinetic processes have the same lifetimes but different amplitudes across all pixels measured, as in Equation 6.3, often restrict themselves to the bi-exponential instance of the model (Barber *et al.*, 2005; Pelet *et al.*, 2000). Verveer *et al.* (2000) acknowledge that the global analysis associated with Problem 6.4 is a separable nonlinear least-squares problem that may be solved using variable projection, though the authors state that for the large number of variables involved in fitting typical FLIM data, the memory requirements prohibit the approach.

TIMP allows an arbitrary number of exponentials to be fit to the data, though under experimentally realistic SNRs it is most often possible to well-

estimate the parameters of only one or two such decays. **TIMP** applies a partitioned variable algorithm to the global analysis problem associated with the analysis of FLIM images. This algorithm forms the residual vector prescribed by the variable projection functional without the need to store and operate on prohibitively large matrices, as is described in detail in Mullen and van Stokkum (2007b) and Chapter 5 of this monograph. The present study of the application of **TIMP** to FLIM data is to the best of the authors' knowledge the first application of variable projection to FLIM data in the literature.

6.4 Extension of TIMP for FLIM data analysis

Several new capabilities were added to the **TIMP** package to facilitate the analysis of FLIM data. As described in Section 6.4.1, a file format was defined for the input of FLIM images into **TIMP**. A method for numerical convolution of an exponential decay with a measured IRF was added to the kinetic model options of the package, as Section 6.4.2 elaborates. New options to visually validate the results of fitting were also added, and are discussed in Section 6.4.3.

6.4.1 Data format

Given a FLIM image, it is often desirable to select those pixels associated with the subject of interest for modeling. For example, given a FLIM image of a cell, only those pixels interior to the cell wall are typically representative of the fluorescence decay of interest, and accordingly only these pixels are usually selected for modeling. Pre-processing dedicated to pixel selection is currently performed outside of **TIMP**. The indices of selected pixels are then included in the ASCII input file.

The format of the input file is as follows.

```

line 1:    reserved for comments, not read
line 2:    reserved for comments, not read
line 3:    the character string "FLIM Image"
line 4:    dimension of image as  $x y$  (space-delimited)
line 5:    number of times  $t$  in image
line 6:    number of pixels  $x$  selected for analysis
line 7:    vector of the times at which measurements were made
line 8:    index of selected pixel and decay trace from this pixel  $\Psi(, p)$ 
...
Line (8+number of selected pixels): the character string "Intensity map"
Remaining lines: FLIM intensity image as matrix of dimension  $x \times y$ 

```

Examples of this file format are included with this contribution. Files in this format may be read into R using the **TIMP** function `readData`. Section 6.4.3 defined the meaning of the intensity image; Section 6.6 contains an example of the use of the `readData` function.

6.4.2 Options for numerical convolution with a measured IRF

The analysis of FLIM data typically employs a measured IRF $g(t)$ in fitting the exponential decay model contained in Equation 6.3. Evaluation of Equation 6.3 requires the numerical convolution of $g(t)$ with an exponential decay. Methods to perform this convolution have been addressed in the literature at least since the seminal paper of Grinvald and Steinberg (1974), as Bajzer *et al.* (1995) discuss. For FLIM data (in which $g(t)$ and the exponential decay are very often represented by 256 or less time points), we have found that methods based on a Fourier transformation are problematic, and that iterative methods give better results. Since an iterative method for the convolution of a vector and an exponential decay was not found implemented in R or in openly available scientific programming libraries, a method based on an iterative technique suggested in Grinvald and Steinberg (1974) was implemented in the shared C library used by **TIMP**, as the function `Conv1`.

To validate that the implementation of this iterative convolution technique returns an un-biased result, we considered its operation on the convolution of an IRF $g(t)$ simulated as a Gaussian with location μ and full width half maximum Δ parameters inspired by values occurring in FLIM experiments. The convolution of a Gaussian with an exponential decay is determined by the analytical expression

$$c_l(t) = \frac{\exp(-k_l t)}{2} \exp(k_l(\mu + k_l \tilde{\Delta}^2/2)) \left\{ 1 + \operatorname{erf} \left[\frac{t - (\mu + k_l \tilde{\Delta}^2)}{\sqrt{2} \tilde{\Delta}} \right] \right\} \quad (6.5)$$

where $\tilde{\Delta} = \Delta/(2\sqrt{2\log(2)})$ and erf is the error function. Note that Equation 6.5 uses the decay rate k_l (which is also the parameter estimated), whereas its reciprocal $\tau_l = 1/k_l$ is commonly reported. For times and decay rates k_l inspired by values in measured data of interest, the results determined by the implementation of the iterative technique are unbiased as compared to results obtained using the analytical expression.

6.4.3 Model validation

Model validation in the FLIM image analysis application domain is ideally largely visual. The magnitude of residuals and fitted parameter estimated are possible to map per-pixel onto the modeled image as colors, allowing the results of fitting to be quickly evaluated. Several options for this display are implemented in **TIMP**. The analysis of a FLIM image with the **TIMP** function `fitModel` results in a multipanel summary plot as shown in Figure 6.1, whose components will be explained in turn.

First histograms of the estimated amplitudes associated with each component, with the corresponding global lifetime estimate on the bottom are displayed. In Figure 6.1 these are the two plots contained in row 1, columns 1 and 2. These

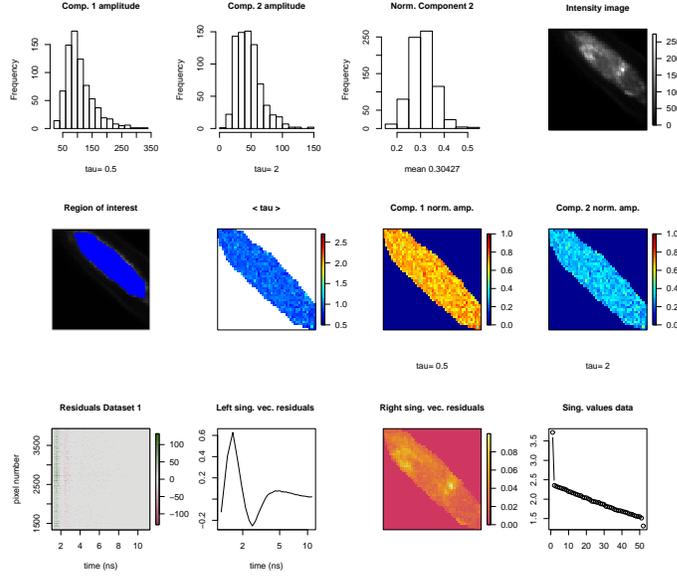


Figure 6.1: An example multipanel summary plot of residuals and fit of a bi-exponential model for measured FLIM data. Individual plots are explained in the text of Section 6.4.3. The image is taken from a fixed BHK (baby hamster kidney) cell with CFP expressed.

plots allow for an impression of the absolute contributions of the components across all pixels of the image.

The following plots in the summary figure are $n_{\text{comp}} - 1$ histograms of the relative contribution p_l of component l , where

$$p_l = \frac{a_l}{\sum_i^{n_{\text{comp}}} a_i} \quad (6.6)$$

In Figure 6.1 this is the plot contained in row 1, column 3. These plots allow for an impression of the relative contribution of the component l across all pixels of the image.

A plot of the intensity image is then given. This intensity image includes those pixels not selected for modeling, and represents the number of photons per pixel measured over the course of all times t represented by the dataset. From this intensity image only some pixels are typically selected for modeling. The selected pixels are shown in the next plot in blue. The intensity image and the intensity image with selected pixels in blue are contained in row 1, column 4 and row 2, column 1, respectively, in Figure 6.1.

The plot entitled “ $\langle \tau \rangle$ ” presents the average lifetime for each pixel x from the selected region, where the average lifetime is given as

$$\langle \tau \rangle = \frac{\sum_l^{n_{\text{comp}}} \tau_l a_{x,l}}{\sum_l^{n_{\text{comp}}} a_{x,l}} \quad (6.7)$$

(row 2, column 2 of Figure 6.1). The average lifetime may allow insight into the rate of energy transfer in processes on a per-pixel basis.

The next n_{comp} plots show normalized amplitudes in a color code mapped to the associated image, for each component l . In Figure 6.1 these are the plots contained in row 2, columns 3 and 4. The normalized amplitude plots allow insight into spatial patterns in the contribution of components. For example, these plots may allow identification of specific structures in a cell where the contribution of a given component is large.

Next the residuals associated with each pixel from the selected region are given as a color image, providing information on the quality of the fit both spatially and temporally, (row 3, column 1 of Figure 6.1). The first left singular vector of the residuals as results from a singular value decomposition (SVD) is plotted next (row 3, column 2 of Figure 6.1). This plot allows insight into structure in the residuals in time. For typical FLIM experiments, this structure is large around time 0, where the exponentially decaying components and the IRF contribute most. Structure in the left singular vectors after time 0 may be indicative of an inadequacy in the applied model. The next plot shows the first right singular vector associated with the SVD of the residuals mapped to the pixels selected for analysis, which provides information on the quality of the fit per pixel, and allows determination of whether the lack of fit is spatially structured. The last plot shows the singular values associated with the SVD of the data. The number of singular values that stand out in this plot indicate how many spatially and temporally independent components are present in the data. Further discussion of the use of the rank of the data in the estimation of the number of components can be found in e.g., Henry (1997).

6.5 A simulation study

A study of the application of **TIMP** to the analysis of simulated FLIM images was made in order to investigate the capabilities of the package in the problem domain. The study was designed in two parts.

The first part, described in Section 6.5.1, examines the ability of the software to estimate the lifetimes associated with bi-exponential decays in which the decay of fluorescence in time was measured over 64 and 256 times (which we refer to as *channels* throughout). 64 and 256 channel data is commonly collected in FLIM experiments, and thus was of particular interest. Simulation of bi-exponential decays was performed because Gratton *et al.* (2003) have shown that resolution of more than two components is not possible over this number of channels for experimentally realistic lifetime values and signal-to-noise ratios.

The second part of the simulation study, described in Section 6.5.2, smoothly varies the two amplitude parameters associated with bi-exponential decays across columns of the image for the purpose of examining whether the software is able to accurately estimate the relative contribution of the components.

Images $\Psi(t, x)$ were simulated using Equation 6.2, as shown in Figure 6.2. Each pixel is associated with a decay in the time window 12.5 ns, over either 64 channels or 256 channels (equidistant in the interval 0-12.5 ns). The IRF $g(t)$ was simulated as a Gaussian with mean 9 and 34 and standard deviation .4 and 1 for the 64 channel and 256 channel cases, respectively, in units of channels. Note that non-zero contribution of the IRF in both the 64 channel and the 256 channel case is represented by very few channels (3-8), as is commonly the case in FLIM experiments. Poisson noise was added to each decay trace $\psi_x(t)$ to obtain data of the desired signal-to-noise ratio (SNR) (using the R function `rpois`). The result may be considered as count data where $\Psi(t, x)$ represents the number of photons collected at a given pixel x and time t , as in measured time-correlated single photon counting data. The SNRs of simulated images were chosen to reflect those commonly obtained in FLIM experiments. Note that the optimization of parameters under least squares criteria will result in biased estimates, since the noise applying to the data is Poisson and not Gaussian distributed, as Maus *et al.* (2001) have studied, but for data comprised mostly of large counts this bias is small.

6.5.1 A simulation study in the resolution of bi-exponential decays

This part of the simulation study examines the ability of **TIMP** to recover satisfactory estimates for the lifetimes underlying simulated images representative of two components. Images simulated with three pairs of lifetimes (in nanoseconds) collated in Table 6.1 were studied. For each pair of lifetimes studied, the

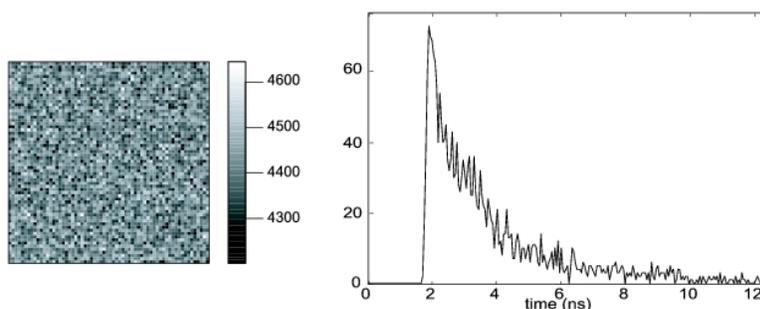


Figure 6.2: (Left) Intensity image of simulated data comprised of 1600 pixels in a 40×40 pixel arrangement, where *intensity* means the total photons summed over all channels. (Right) A fluorescence decay trace over 256 channels in the interval 0-12.5 ns is associated with each of the 1600 pixels comprising an image.

relative contribution of the two components was varied between .1 and .9, so that 9 different images were simulated using each pair of lifetime values. The lifetime values are experimentally motivated (Borst *et al.*, 2005). The images were simulated for both the SNR 8 and the SNR 15 case; the SNR 8 case is average for typical FLIM experiments, while the SNR 15 is higher than average.

A bi-exponential model was fit to the images, with the relative contribution of the two components being estimated as conditionally linear on values for the nonlinear lifetime estimates. The results are shown for images simulated with the pairs of lifetimes on row 1 and 2 of Table 6.1 in Figure 6.3. Note that each boxplot describes the variance in lifetime estimates as the relative amplitude of the components is varied. Our criteria for a satisfactory lifetime estimate is that the estimate is $\pm 5\%$ of the lifetime value used in simulation for data containing 256 channels, and within $\pm 10\%$ of the lifetime value used in simulation for data containing 64 channels. Under these criteria, the lifetime estimates obtained and shown graphically in Figure 6.3 are satisfactory. The small bias is attributed to using the number of photons at the leftmost point of each bin of times comprising a time-channel as representative of the average lifetime within the bin; because the data is exponentially decaying, there are always more photons to the left of the bin than to the right, and the average lifetime is thereby underestimated. The bias disappears when the number of channels is increased (for example, for data containing 1024 channels and the same SNR and lifetime values, it is insignificant). For the third pair of lifetimes studied, with $\tau_2 = .2$ ns and $\tau_2 = .5$ ns, it is impossible to determine satisfactory estimates even for data with SNR 15. The very short lifetimes are represented by only a few channels, so that there is not sufficient information.

We found that for the lifetime values examined, for cases in which the contribution of one component was lower than 20% and the SNR was 8, lifetime estimates were not satisfactorily estimable. For SNR 15, lifetimes were not satisfactorily estimable for cases in which the contribution of one component was less than 10%.

This part of the simulation study was also repeated using an IRF measured on a FLIM set-up (as opposed to using a simulated IRF with a Gaussian distribution) to check that noise present in the IRF does not significantly decrease the accuracy of lifetime and amplitude estimates. The obtained lifetime and amplitude estimates were very similar to those reported for the Gaussian IRF

Group	τ_1	τ_2
1	1.14	3.72
2	.6	2.5
3	.2	5

Table 6.1: Parameter values in nanoseconds used in simulation of bi-exponential images. Instances of each group were simulated with contributions from the component with the longer lifetime τ_2 as 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90% of the total intensity.

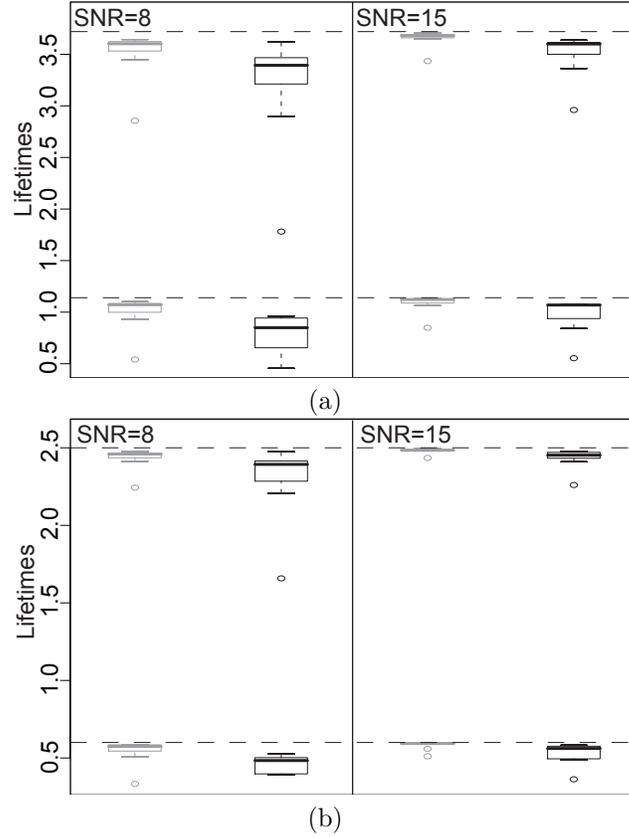


Figure 6.3: Boxplots of lifetime estimates for each of two components given datasets simulated with lifetimes described in the first two rows of Table 6.1. Each boxplot is comprised of lifetime estimates estimated from fitting 9 different images, simulated with different relative amplitudes between the lifetimes. Lifetimes used in simulation are marked as dotted lines. The grey boxes represent results obtained on images in which the decay was represented by 256 channels, whereas the black boxes represent results obtained on images in which the decay was represented by 64 channels. (a) shows results on images simulated with the lifetimes given in row 1 of Table 6.1 and (b) shows results on images simulated with the lifetimes given in row 2 of Table 6.1.

case, validating that the parameter estimation methodology is robust to an experimentally realistic amount of noise in the IRF.

We consider this part of the simulation study to demonstrate some limits of the resolvability of bi-exponential lifetimes on images inspired by measured data, and that for cases of practical interest **TIMP** lifetime estimates returned by **TIMP** are satisfactory.

	value used in simulation	SNR 25	SNR 8
τ_1	.6	.57	.58
τ_2	2.5	2.49	2.38

Table 6.2: Lifetime values in nanoseconds used in simulation and estimated lifetimes for simulated images with smoothly varying contributions from two components.

6.5.2 A simulation study in the estimation of relative amplitudes of bi-exponential decays

A simulation study was made on instances of the image shown in Figure 6.4. The decay curve associated with each pixel is bi-exponential, with the two components having lifetimes of 0.6 and 2.5 ns respectively. The amplitude of the contribution a_1 from the first component varies from 0 to 1 across each column of the image, while the contribution a_2 from the second component varies from 1 to 0. Fitting a bi-exponential model to such images allows examination of whether the software is able to accurately estimate the two amplitude parameters a_1 and a_2 associated with each pixel. This part of the simulation study is inspired by a similar study by Pelet *et al.* (2000). The size of each analyzed image was 64×64 pixels (4096 pixels). The decay of the intensity at each pixel was represented by 256 times equidistant in the interval 0-12.5 ns (this is the 48 ps/channel case described in Section 6.5.1). Images were simulated with both SNR 8 and SNR 25.

Table 6.2 shows that the lifetime estimates well-approximate the values used in simulation of the images in both the SNR 25 and the SNR 8 case. The deviations from the values of the amplitudes a_i used in simulation are small and unbiased, as shown in Figure 6.6 (c) graphically. Furthermore the lifetime

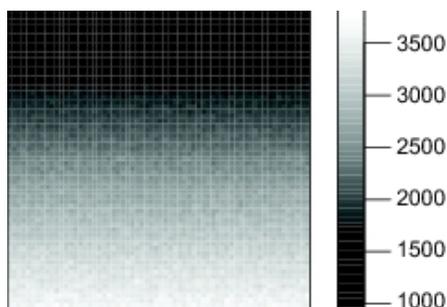


Figure 6.4: A 64×64 pixel simulated image at one timepoint, in which the relative contribution of the first component increases linearly from 0 to 1 and the contribution of the second component decreases linearly from 1 to 0 along each column of the image. Each simulated dataset is comprised of 256 such images, representing the 256 times (channels) simulated.

estimates collated in Table 6.2 are also satisfactory. We conclude that this part of the simulation study demonstrates the ability of **TIMP** to return satisfactory estimates of the amplitude parameters a_i determining the relative contribution of components.

6.6 Case study on measured CFP data

We were interested in investigating the capabilities of **TIMP** for FLIM image analysis of measured data. In cell biology studies FRET-FLIM is often used to demonstrate molecular interactions *in vivo*. For this purpose the fluorescent proteins cyan fluorescent protein (CFP) and yellow fluorescent protein (YFP) are the most widely used as donor-acceptor FRET pairs (Grailhe *et al.*, 2006). However, the fluorescent decay of CFP is bi-exponential, making quantitative analysis of an interacting FRET population challenging (Ruscinova *et al.*, 2004; Peter *et al.*, 2005).

Time-correlated single photon counting experiments with a very high SNR (unattainable in FLIM experiments) described in Borst *et al.* (2005) established the lifetimes of CFP in a solution. We performed an experiment to collect FLIM images of the same sample in a micro-capillary, using the experimental set-up described in Borst *et al.* (2003). Note that FLIM images of proteins in solution are not usually measured (the study of protein conformational dynamics *in situ* being the goal of most FLIM experiments), but that this experiment offers an opportunity to validate the ability of the software to estimate the lifetimes associated with the fluorescence decay of this important donor.

The SNR of the FLIM experiment was approximately 9. The time resolution was 48 ps/channel (over 256 channels). The fluorescence intensity image and region selected for analysis are shown in Figure 6.7.

To convey how the package is used to analyze a FLIM image, we describe

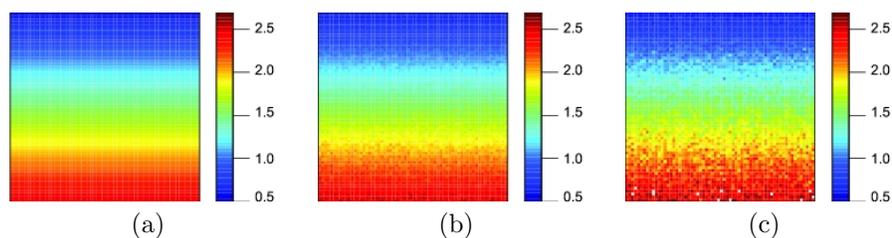


Figure 6.5: Colors above represent the average lifetime estimated with Equation 6.7. (a) Simulated image with $\tau_1 = .6$ ns, $\tau_2 = 2.5$ ns and a linearly varying contribution from two components over time. (b) Estimates of the average lifetime determined with Equation 6.7 for an instance of the image in (a) with SNR = 25. Estimated lifetimes are $\tau_1 = .57$ ns, $\tau_2 = 2.49$ ns. (c) Estimates of the average lifetime determined with Equation 6.7 for an instance of the image in (a) with SNR = 8. Estimated lifetimes are $\tau_1 = .58$ ns, $\tau_2 = 2.38$ ns.

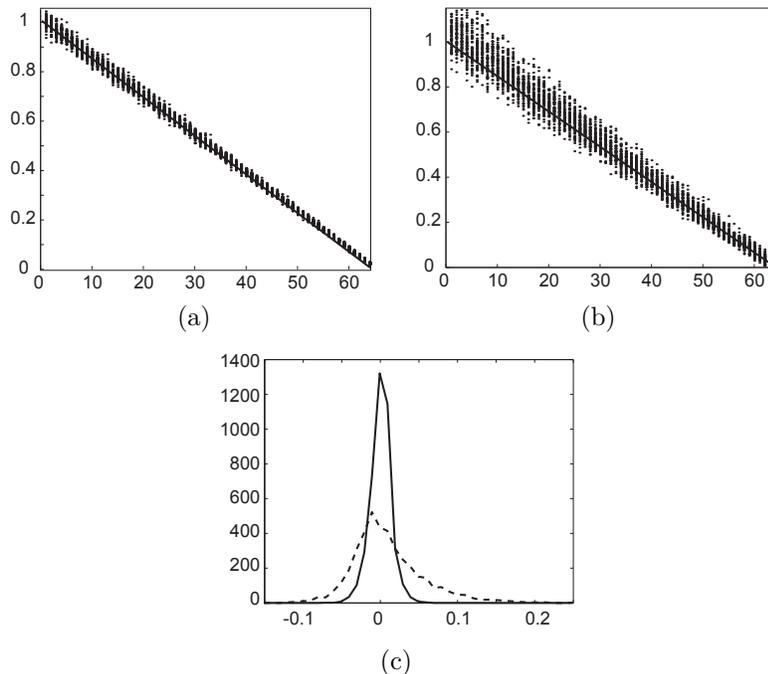


Figure 6.6: A 64×64 image was simulated in which the relative contribution of two exponentially decaying components was made to vary linearly along each column as shown in Figure 6.2 (a). **TIMP** was then used to fit a model for the simulated data, resulting in 64 estimated relative amplitudes that correspond to rows of Figure 6.2 (a) for each of the 64 distinct relative amplitude values used in simulation, under data having both SNR 25 and SNR 8. In the present figure, (a) and (b) show the relative amplitude values used in simulating the data as a line; dashed lines represent the distribution over 64 estimates, i.e., rows in the images in Figure 6.2 (a) and (b). In (c) histograms of deviations from the values used in simulation for (solid line) SNR = 25 (dashed line) SNR = 8 estimates are shown. These deviations are unbiased and small.

all commands used to perform this part of the study.

6.6.1 Reading FLIM data into TIMP and preprocessing

The package is loaded.

```
R> library("TIMP")
```

Data is read into R using the `readData` function of **TIMP**.

```
R> cfp_data <- readData("cfp-13um-256ch-1000s_all.txt")
```

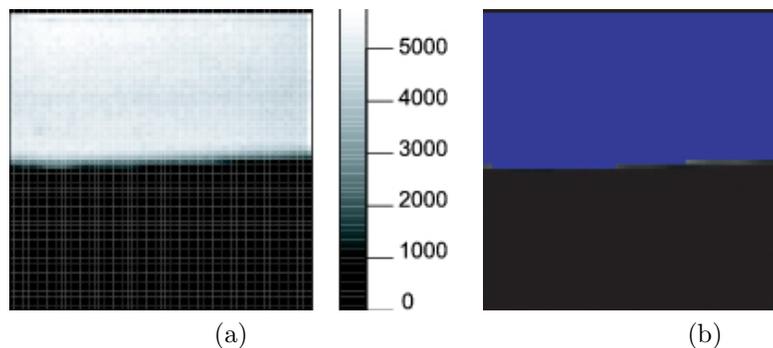


Figure 6.7: (a) Intensity image of a measured image of CFP in solution, where color represents the number of photons detected in a given pixel (b) Intensity image with pixels selected for analysis in blue.

Preprocessing is then performed to select certain times for analysis using the **TIMP** function `preProcess`.

```
R> cfp_data_sel <- preProcess(serT, sel_time=c(33,230))
```

A measured IRF is then read in and the same time points as selected in the data are chosen.

```
R> mea_IRF <- scan("xtetoh_256_060822-bg_int.txt")[33:230]
```

6.6.2 Initial model for CFP in solution: Mono-exponential decay

The first model applied is based on a mono-exponential decay. The starting value for the decay rate given as 0.3, and is constrained positivity. The model is specified using the **TIMP** function `initModel`.

```
R> mono_cfp_model <- initModel(mod_type = "kin",
+ kinpar=c(0.3), convalg = 1, parmu = list(0.01),
+ measured_irf = mea_IRF, fixed = list(parmu=c(1)),
+ seqmod=FALSE, positivepar = c("kinpar"))
```

6.6.3 Fitting and validation of initial mono-exponential model

The **TIMP** function `fitModel` is used to fit the mono-exponential model.

```
R> mono_result <- fitModel(list(cfp_data_sel),
+ list(mono_cfp_model),
+ opt=kinopt(iter=0, linrange = 20,
+ makeps ="cfp_mono", residplot = TRUE,
```

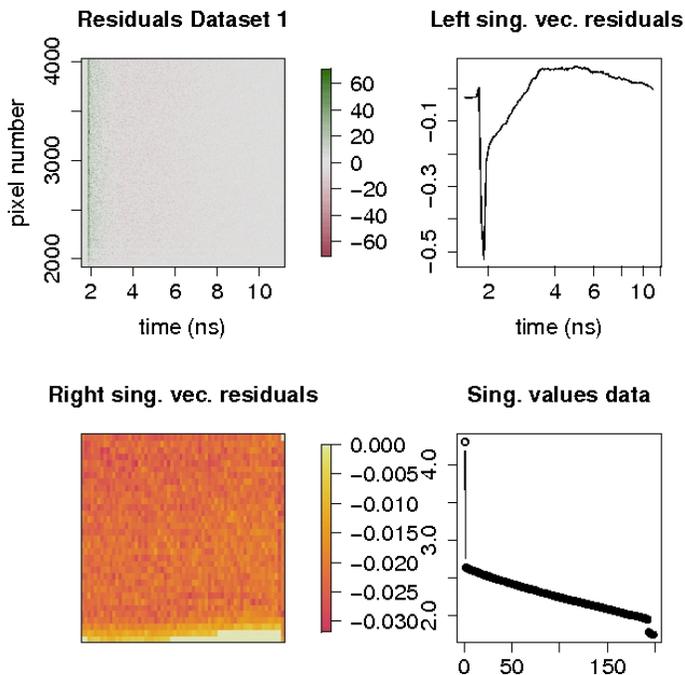


Figure 6.8: An image plot of the residuals under the mono-exponential model fit shows structure in time before 3.5 ns. The first left singular vector resulting from an SVD of the residuals also shows this structure. The first right singular vector of an SVD of the residuals mapped to the associated pixels on the intensity image shows the residuals are relatively homogeneous in space. The RMS error associated with this fit is 5.2.

```
+ notraces = TRUE, xlabel = "time (ns)",
+ ylabel = "pixel number", FLIM=TRUE))
```

The plot of the residuals returned is shown in Figure 6.8. The image plot of the residuals in the upper left hand corner show that there is a pattern of misfit around time 3.5 ns. This pattern of misfit is also indicated in the large upward trend of the left singular vector of the residuals shown in the upper right plot of Figure 6.8, which peaks at 3.5 ns. The root mean square (RMS) error associated with the fit is 5.2. We conclude that a mono-exponential decay model for CFP is not sufficient.

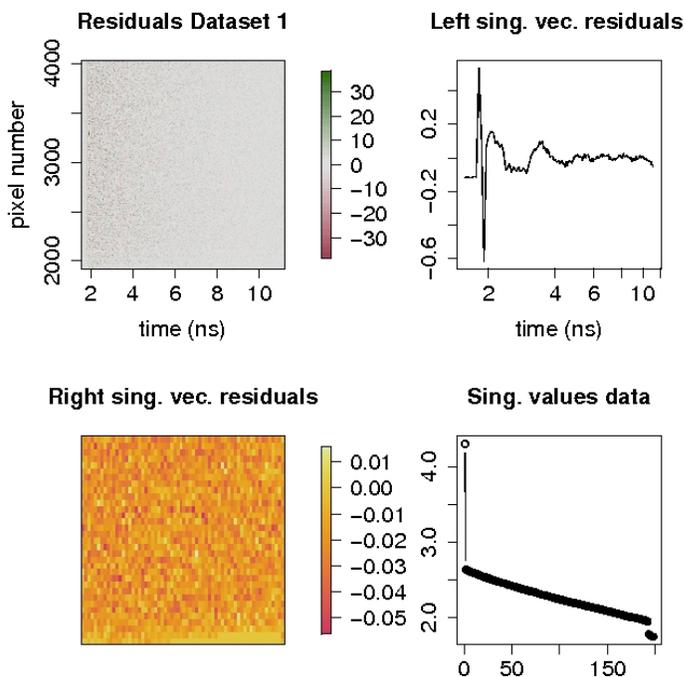


Figure 6.9: An image plot of the residuals under the bi-exponential model fit shows less structure in time as compared to the same plot for the mono-exponential fit in Figure 6.8. The first left singular vector resulting from an SVD of the residuals also shows less structure. The first right singular vector of an SVD of the residuals mapped to the associated pixels on the intensity image shows that the residuals remain homogeneous in space. The RMS error associated with this fit is 4.9, less than for the mono-exponential model fit.

6.6.4 Refined model for CFP in solution: Bi-exponential decay

Based on the inadequacy of the fit of the mono-exponential model as evidenced by analysis of the residuals, the `initModel` function was used to specify a bi-exponential model for the measured CFP image.

```
R> bi_cfp_model <- initModel(mod_type = "kin",
+ kinpar=c(1, 0.3), convalg = 1, parmu = list(0.01),
+ fixed = list(parmu=c(1)), measured_irf = mea_IRF,
+ seqmod=FALSE, positivepar=c("kinpar"),
+ title="CFP bi-exponential decay")
```

6.6.5 Fitting and validation of initial bi-exponential model

	a_1	τ_1	a_2	τ_2	$\langle \tau \rangle$
TIMP estimate	.373	.95	.627	3.48	2.54
established value	.335	1.14	.665	3.72	2.86

Table 6.3: Parameters estimates obtained using **TIMP** on a measured CFP dataset analyzed with a bi-exponential model, and values in the literature for a dataset collected under similar experimental conditions analyzed using the same bi-exponential model. Note that variability in the experimental set-up, laser power and sample preparation limit the degree to which the results are directly comparable.

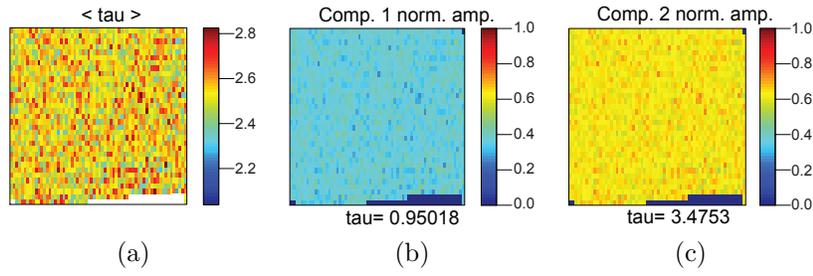


Figure 6.10: (a) Distributions of the average lifetimes per location estimated with Equation 6.7. Normalized amplitudes for component 1 (b) and component 2 (c) as a color on the associated image.

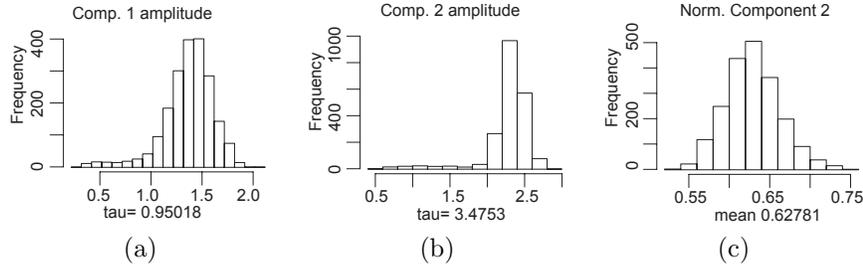


Figure 6.11: Histograms of amplitudes of components (A,B) and normalized amplitude of component 2 calculated with Equation 6.6 (C)

```
R> bi_result <- fitModel(list(cfp_data_sel), list(bi_cfp_model),
+ opt=kinopt(iter=20, linrange = 20,
+ makeps = "cfp_bi",
+ notraces = TRUE, residplot = TRUE,
+ xlabel = "time (ns)",
```

```
+ ylabel = "pixel number", FLIM=TRUE))
```

An image plot of the residuals under the bi-exponential model fit shows less structure around time 3.5 ns as compared to the same plot for the mono-exponential fit in Figure 6.8. The first left singular vector resulting from an SVD of the residuals also shows less structure around 3.5 ns. Note that we are not concerned about the structure in the SVD around 0 ns because misfit at this time results from the large contribution of the IRF and the peak in the amplitude of components at the start of their decay at this timepoint. The first right singular vector of an SVD of the residuals mapped to the associated pixels on the intensity image shows that the residuals remain homogeneous in space. Furthermore, the RMS square error has decreased to 4.9 from the RMS error of 5.2 under the fit of the mono-exponential model.

The lifetime estimates under the bi-exponential model agree well with values published in Borst *et al.* (2005) for analysis of a dataset collected under similar experimental conditions, as tabulated in Table 6.3. Figure 6.10 (a) shows that the estimate for the average lifetime per pixel over the course of the decay (as determined with Equation 6.7) has no spatial structure, as is expected since the measured image represents a homogeneous solution. Figure 6.10 (b) and (c) show that the normalized amplitudes of the components are also spatially homogeneous, also as expected from the homogeneity of the solution.

6.7 Conclusions

A feasibility study has been made to investigate the use of the **TIMP** package of R for the analysis of FLIM data. In the course of the study new options for the fitting and validation of FLIM images with the package were developed.

In a simulation study the package was shown to return satisfactory estimates of both lifetime and amplitude parameters, the latter of which are estimated as conditionally linear parameters. On a real dataset it was possible to resolve the contributions of two components known to exist in terms of lifetime and amplitude estimates known from the literature, which further confirms the applicability of the partitioned variable projection fitting algorithm that **TIMP** implements to modeling FLIM images.

Future work will apply **TIMP** to the analysis of further experimentally collected FLIM data. Energy transfer between components will be modeled using the compartmental modeling options for **TIMP** described in Chapter 5 of this monograph. Implementation of a graphical user interface (GUI) to facilitate interactive model validation is also planned, along with a study to benchmark and optimize the package for speed on problems in FLIM analysis. Furthermore, an analogue of variable projection for the case of Poisson noise is of interest to develop and implement.

Acknowledgments

Sergey Laptinok, an author of the published version of this chapter, is grateful for funding from the Sandwich Programme of Wageningen University. An anonymous reviewer of the published manuscript is thanked for helpful suggestions regarding the presentation of results.

Chapter 7

Global analysis of multiple gas chromatography mass spectrometry (GC/MS) data sets¹

7.1 Introduction

Global analysis methods are widely used in time-resolved spectroscopy to resolve components and to estimate the physico-chemical parameters that describe the system of interest. For review see van Stokkum *et al.* (2004); van Stokkum (2005). In this feasibility study it is investigated whether this methodology can also be applied in hyphenated methods, in particular gas chromatography mass spectrometry (GC/MS). GC/MS is a popular technique in metabolomics, where many samples have to be analyzed and compared with respect to the identity and abundance of metabolites (Tikunov *et al.*, 2005; Ryan and Robards, 2006; Last *et al.*, 2007). A simultaneous analysis of the many data sets is mandatory to extract the most information, and a statistical model is needed to deal with systematic and random errors. In Chapter 8, the results will be compared with alternative non-parametric methods like Multivariate Curve Resolution-Alternating Least Squares (MCR-ALS) (Tauler *et al.*, 1995; Tauler, 1995; de Juan and Tauler, 2003), which is a self-modeling method (Garrido *et al.*, 2008).

Global analysis methodology is based upon a parametrized model of the observed data, including random (and possibly also systematic) errors. The model usefulness is judged by the quality of the fit and by the biophysicochemical interpretability of the estimated parameters (parameters that describe the

¹A version of this chapter will appear as van Stokkum, Mullen, and Mihaleva (2008a) in *Chemometrics and Intelligent Laboratory Systems*.

elution profile and mass spectra of components). Advantages of the method are most evident with multiple data sets and overlapping elution profiles. Differences between data sets are described by alignment parameters and by relative amplitude parameters. The estimated mass spectrum is identical between experiments. Critical is the assumed shape of the elution profile, for which we took an exponentially modified Gaussian (EMG). The quality of the fit is judged by the magnitude of the residuals and their structure and by the precision of the estimated parameters. Outliers and saturation effects can be dealt with systematically.

In Section 7.2, the global analysis method is detailed. In Section 7.3 the results from four case studies using already published data (Jonsson *et al.*, 2005; Tikunov *et al.*, 2005) are presented and discussed. Also the adequacy of the EMG shape is evaluated. Section 7.4 contains conclusions.

7.2 Methods

The aim of global analysis is to obtain a model-based description of the full data set in terms of a model containing a small number of precisely estimated parameters.

7.2.1 Modeling an elution profile

The elution profile is described by an exponentially modified Gaussian (EMG) function which is the convolution of a Gaussian instrument response function with parameters λ and Δ for, respectively, location and full width at half maximum, FWHM, with an exponential decay with rate k as

$$c(t, \lambda, \Delta, k) = \frac{\exp(-kt)}{2} \exp\left(k\left(\lambda + k\frac{\tilde{\Delta}^2}{2}\right)\right) \left\{ 1 + \operatorname{erf}\left[\frac{t - (\lambda + k\tilde{\Delta}^2)}{\sqrt{2}\tilde{\Delta}}\right] \right\} \quad (7.1)$$

where $\tilde{\Delta} = \Delta/(2\sqrt{2\log(2)})$ and t is time.

With a positive decay rate k the elution profile exhibits a tail. To describe fronting, a negative decay rate k can be used. Then the time argument ($t - \lambda$) must be reversed yielding

$$c_{\text{fronting}}(t, \lambda, \Delta, k) = \frac{\exp(-kt)}{2} \exp\left(k\left(-\lambda + k\frac{\tilde{\Delta}^2}{2}\right)\right) \left\{ 1 + \operatorname{erf}\left[\frac{\lambda - t - k\tilde{\Delta}^2}{\sqrt{2}\tilde{\Delta}}\right] \right\} \quad (7.2)$$

The EMG function is the most common peak shape in chromatography (Marco and Bombi, 2001).

7.2.2 Global analysis

The basis of global analysis is the superposition principle, which states that the measured data $\psi(t, \mu)$ (where μ means the mass-to-charge axis) result from a superposition of the mass spectrum $S_i(\mu)$ of the components present in the

system of interest weighted by their concentration $c_l(t)$

$$\psi(t, \mu) = \sum_{l=1}^{n_{\text{comp}}} c_l(t) S_l(\mu) \quad (7.3)$$

where n_{comp} is the number of components. The $c_l(t)$ of each component is described by an EMG function with parameters k_l, λ_l, Δ_l . Thus the model reads

$$\psi(t, \mu) = \sum_{l=1}^{n_{\text{comp}}} c_l(t, \lambda_l, \Delta_l, k_l) S_l(\mu) \quad (7.4)$$

Above the stochastic element of the model is neglected; throughout we assume that it is comprised of additive Gaussian white noise.

For an additional experiment p we assume that the shape parameters k_l and Δ_l are independent of experiment, and introduce alignment parameters λ_p (assumed to be identical for all components), and amplitude parameters $a_{l,p}$, yielding a model parametrized as

$$\psi_p(t, \mu) = \sum_{l=1}^{n_{\text{comp}}} c(t_l, \lambda_l + \lambda_p, \Delta_l, k_l) a_{l,p} S_l(\mu) \quad (7.5)$$

When the shape varies with experiment, or when the alignment varies between components because of interaction, additional parameters can easily be introduced.

7.2.3 Parameter estimation

For each component the $c_l(t)$ is described by an EMG function with three parameters k_l, λ_l and Δ_l . Each additional experiment p requires one time shift parameter λ_p for alignment, and n_{comp} amplitude parameters $a = [a_{1,p}, a_{2,p}, \dots, a_{n_{\text{comp}},p}]$ representing the amplitude of each component, so that we end up with $(2 + n_{\text{exp}})n_{\text{comp}} + n_{\text{exp}} - 1$ intrinsically nonlinear parameters for n_{exp} experiments. The mass spectra S contain $n_{\text{comp}}n_{\mu}$ parameters, where n_{μ} is the number of masses represented by the data, so that the number of parameters represented by S is typically on the order of 10^3 . These parameters, however, are conditionally linear, and can be eliminated analytically from the problem using the variable projection method reviewed by Golub and Pereyra (2003). Note that in this way the model fitting process proceeds much more efficiently. Since negative values of S cannot be interpreted, these parameters are constrained to nonnegative values. The incorporation of nonnegativity constraints on the conditionally linear parameters S within global analysis is described by Sima and Van Huffel (2007) and Chapter 4 of this monograph, and is implemented using a nonnegative least squares (NNLS) algorithm by Lawson and Hanson (1974, 1995). A background contribution may be assumed to be constant in the time window analyzed (and has only a mass spectrum, but no parameters). More complex formulations of the background contribution are possible by introduction of a slope parameter or some other more flexible parametric description.

7.2.4 Residual analysis

The residual matrix is analyzed using a singular value decomposition (SVD):

$$\psi_{res}(t, \mu) = \psi(t, \mu) - \psi_{fit}(t, \mu) = \sum_{i=1}^m u_i(t) s_i w_i^T(\mu) \quad (7.6)$$

where u_i and w_i are the left and right singular vectors, s_i are the sorted singular values, and m is the minimum of the number of rows and columns of the data matrix. When multiple experiments are simultaneously analyzed the residual matrices are concatenated. Extra measures are needed to identify residuals that are caused by detector saturation, and to identify outliers.

A residual caused by detector saturation is defined by $\psi_{res}(t, \mu) > \alpha_1 \psi_{fit}(t, \mu)$ and $\psi(t, \mu) > \alpha_2$, where α_1 and α_2 are values that depend upon the experiment. This definition depends upon ψ_{fit} and thus must be determined iteratively.

To identify outliers all residuals at a particular mass-to-charge (m/z) value of experiment p are sorted, and the Lower and Upper Fourth are determined, and the Fourth Spread. Then the Lower Limit (LL) and Upper Limit (UL) are defined as

$$LL = \text{Lower Fourth} - \alpha \cdot \text{Fourth Spread} \quad (7.7)$$

$$UL = \text{Upper Fourth} + \alpha \cdot \text{Fourth Spread} \quad (7.8)$$

The data points associated with residuals outside these Lower and Upper Limits are identified as outliers (Tukey, 1977). Experimentally we found that a typical value of α useful in the context of GC/MS data is 5. A low weight is applied to huge signals that can saturate the detector e.g. the trimethylsilyl (TMS) peak at m/z 73. Likewise, low weights are applied to outliers and residuals caused by detector saturation (estimated by the above procedures). High weights can be applied to characteristic masses provided the signal-to-noise ratio is sufficient. Instrument nonlinearity is currently neglected, but in principle it could be included in the model function. The global analysis method developed here was applied to two large sets of data, namely tomato data sets described in Tikunov *et al.* (2005) consisting of 37 different tomato fruit varieties, and standard mixture data described in Jonsson *et al.* (2005) comprising samples with up to 101 different chemical compounds. No preprocessing was applied to the data, except for a baseline correction when needed.

7.2.5 Refinement

After a global analysis a further refinement is possible. The estimated mass spectra can be used in a weighted NNLS fit for each time point of each dataset p to estimate the amplitudes of the concentrations, so that c_t in $\psi_{pt}^W = S^W c_t$ has solution

$$c_t = \text{NNLS}(S^W, \psi_{pt}^W) \quad (7.9)$$

where the superscript W is used to indicate that weights are applied, and ψ_{pt}^W is a column vector of all masses at time t of experiment p . S^W is the weighted

matrix of estimated mass spectra. The refinement provides an opportunity to check the assumed peak shape and peak resolution.

7.2.6 Identification

The estimated mass spectra were searched against the NIST05 library (Ausloos *et al.*, 1999) extended with the Golm mass spectral library (Kopka *et al.*, 2005) of derivatized compounds. For each spectrum 10 hits were retrieved. The matching factors reported by the MS Search 2.0 program (Ausloos *et al.*, 1999) are between 0 (no match) and 1000 (perfect match). Two different values for the matching factors, MF and RMF, are reported. In the calculation of MF the experimental spectrum is used as a template, whereas for RMF the template is the library spectrum. Comparable MF and RMF values indicate that there are no additional and/or missing values in either of the spectra. To increase the reliability of the identification we have included the retention index (RI) in the evaluation of the library hits. However, RI data are available for only a small fraction of the compounds in the NIST05 library. Therefore we have used a recently developed quantitative structure-retention index model (Mihaleva *et al.*, 2008) for the estimation of RI based upon the structure of the compounds.

7.3 Results and Discussion

We successfully performed over 100 case studies on selected time windows using different amounts of datasets, from which we present here four illustrative examples. The first and second case study presented in Section 7.3.1 are referred to as Tomato I and Tomato II, respectively, while the first and second case study described in Section 7.3.2 are referred to as Mixture I and Mixture II, respectively.

7.3.1 Tomato data

The case studies here demonstrate the ability of global analysis to resolve two overlapping components and resolve correct mass spectra in the case of saturation effects (case study Tomato I) and the ability to resolve six overlapping peaks (case study Tomato II).

In case study Tomato I, we use six tomato samples (e.g., datasets) and focus on a small part of the chromatogram where two components clearly overlap. At m/z values of 70 and 71 the peak location is below RI 787 and above RI 788, respectively, indicating that these are characteristic masses, whereas at an m/z value of 67 both components contribute almost equally. These six data sets (shown at selected m/z values in Figure 7.1) were simultaneously analyzed, resulting in the mass spectra of Figure 7.4. Both mass spectra were successfully identified as elaborated in Table 7.1. The mass spectra at RI 786.6 was identified as 3(Z)-Hexenal. The first five hits retrieved from the library had very similar MF and RMF values but their RI's differed. Using both the matching factors

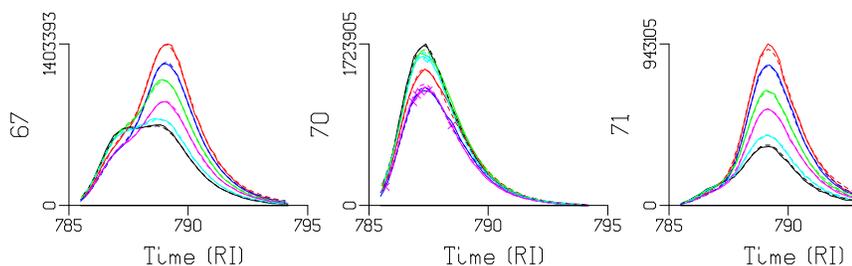


Figure 7.1: **Tomato I.** Representative traces (aligned with estimated shift parameters) below saturation threshold at m/z values of 67, 70 and 71. Colors indicate the six different data sets, solid lines are data and dashed lines indicate fit.

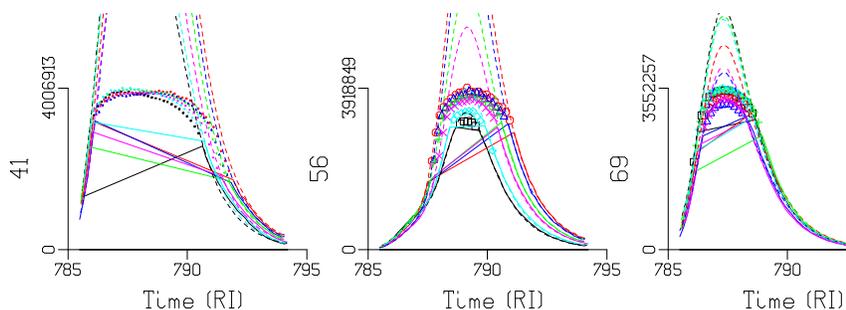


Figure 7.2: **Tomato I.** Representative traces (aligned with estimated shift parameters) above saturation threshold at m/z values of 41, 56 and 69. Colors indicate the six different data sets, solid lines are data and dashed lines indicate fit. Symbols indicate saturation data points. The two tails of the peak are connected by straight lines.

and RI, 3(Z)-Hexenal was selected as the best hit. The component at RI 788.5 was identified as Hexanal. There was a much larger difference between the MF values of Hexanal and the rest of the hits. Also, the predicted RI was close to that estimated in the experiment. These two compounds have been previously identified in tomato samples (Tikunov *et al.*, 2005; Petro-Turza, 1987)

The estimated FWHM parameters are 1.6 and 1.7 RI units, and the locations of these two peaks are somewhat more than one FWHM apart. The estimated decay rate parameters are both about 0.8/RI, indicating appreciable tailing. These components could not be reliably resolved by analysis of individual data sets using e.g. the Automated Mass Spectral Deconvolution and Identification System (AMDIS) program (Stein, 1999). The estimated elution profiles are

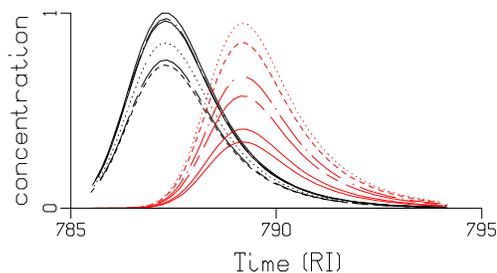


Figure 7.3: **Tomato I.** Estimated elution profiles with colors indicating the two components (corresponding to those in Figure 7.4), and linetypes indicating the different experiments. The contribution of each component to each data set is the product of each elution profile times the associated normalized mass spectrum times the scaling factor 10259207.

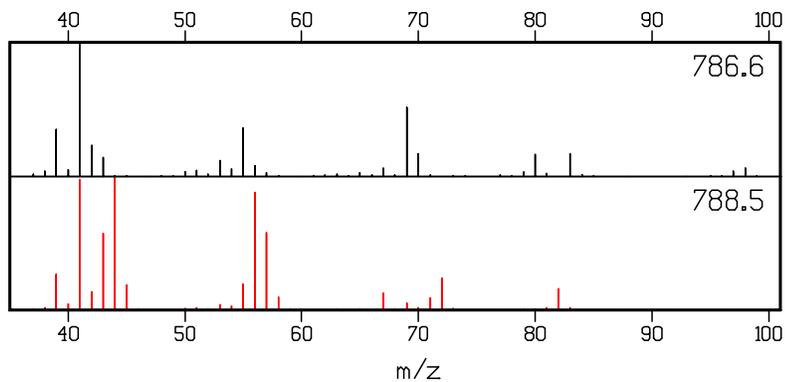


Figure 7.4: **Tomato I.** Normalized mass spectra with location λ of peak maximum indicated in the upper right corner.

depicted in Figure 7.3. Note that the amount of component present in the data sets varies appreciably, as indicated by the different linetypes.

Examples of saturation are depicted in Figure 7.2. Note that at these masses most of the peak is clipped, and it is essential to reconstruct the original shape using the unclipped tails of the elution profile. In this way a reliable estimate

of the mass spectrum can also be made at these critical masses. It turns out that this is essential for adequate identification of the compounds present in the samples. From a further in depth analysis of the residuals (visible as systematic misfit at e.g. the μ value of 41) we infer that a small instrument nonlinearity may be present. In this way, these two overlapping components have been successfully analyzed in 37 different tomato varieties simultaneously.

In the case study Tomato II we present an in-depth analysis of a much smaller peak around RI 1042 in six tomato samples.

Note the large differences in the shapes of the elution profiles in Figure 7.5, and also the differences in the amplitudes of the datasets, as indicated by the different colors. E.g., at m/z values of 58 and 99 the cyan and red are largest, peaking near RI 1043, whereas at 57 green and blue dominate, peaking somewhat earlier, and at 105 blue and black dominate. To describe these differences six components were needed to fit the data, whose mass spectra are shown in Figure 7.7. In this region Tikunov *et al.* (2005) have identified the components at RI 1039.4 and 1042.8 as Benzyl alcohol and 2-Isobutylthiazole. These compounds were found as the best hits also in our analysis (see Table 7.1). The component at RI 1042.0 was identified as 5-Methyl-2(5H)-furanone, a compound recently found to be present in small amounts in tomato (Buttery and Takeoka, 2004). The best hit for the component at RI 1041.2 was that of a chlorinated acid. Both the MF and RMF values were very low which makes the identification doubtful. Also, only a few chlorinated compounds have been identified in tomato (Petro-Turza, 1987). Unsaturated alcohol (3,5-Octadien-2-ol) and ketone (5-Methyl-3-Hepten-2-one) had very similar MF and RMF values with the mass spectra at RI 1040.6. The RI's were also close to the experimental

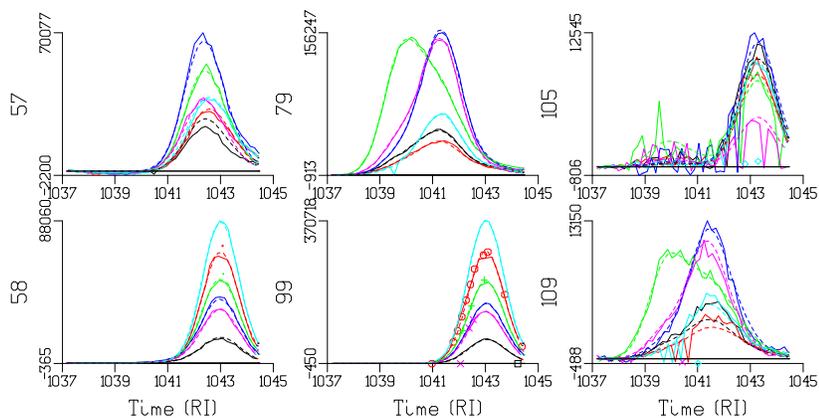


Figure 7.5: **Tomato II.** Representative traces (aligned with estimated shift parameters) at m/z values between 57 and 109. Colors indicate the different data sets, solid lines are data and dashed lines indicate fit.

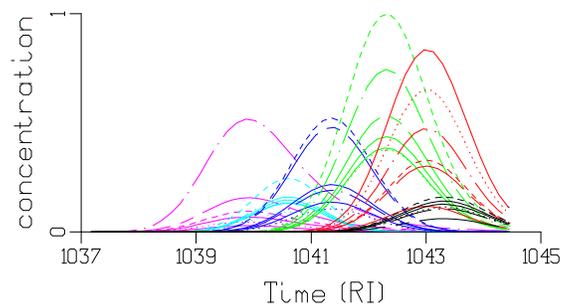


Figure 7.6: **Tomato II.** Estimated elution profiles with colors (corresponding to those in Figure 7.7) indicating the six components, and linetypes indicating the different experiments. The contribution of each component to each data set is the product of each elution profile times the associated normalized mass spectrum times the scaling factor of 519121. In order to improve visibility of the small concentrations, we have used a square root scaling of the maxima. Thus e.g. the black concentration maximum at about 0.1 corresponds to a true maximum of about 0.01.

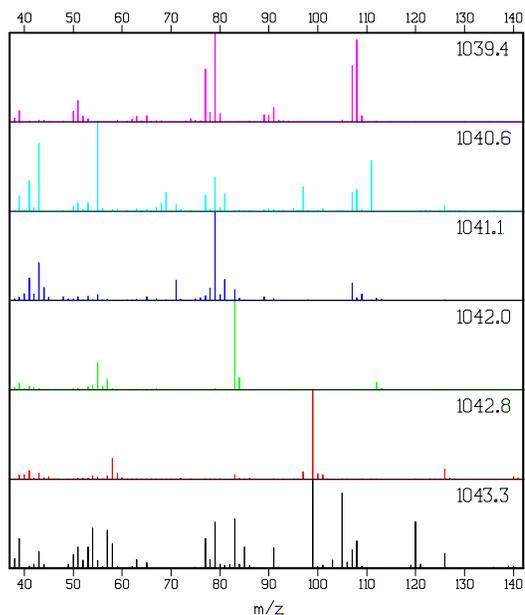


Figure 7.7: **Tomato II.** Normalized mass spectra with location λ of peak maximum indicated in the upper right corner.

name	RI	MF	RMF	RI _{lit}	RI _{pred}
3(Z)-Hexenal	786.6	842	843	799	813
Hexanal	788.5	901	902	784	827
Benzyl Alcohol	1039.4	898	902	1026	1006
3,5-Octadien-2-ol	1040.6	723	738	–	1095
5-Methyl-3-hepten-2-one	1040.6	716	758	–	966
Chloro-2-hydroxy-propanoic acid	1041.1	613	643	–	1081
5-Ethyl-2(5H)-Furanone	1042.0	844	865	952	1004
2-Isobutylthiazole	1042.8	841	861	1013	1122
1-ethyl-4-methyl-Benzene	1043.3	598	711	952	1009

Table 7.1: Identifications for Tomato I and II case studies described in Section 7.3.1.

value. These two compounds have not been identified in tomato but other unsaturated alcohol and ketones are known to be present. There was a large difference between the MF and RMF values for the hits of the mass spectrum at RI 1043.3. This is an indication that the extracted mass spectrum is not pure probably due to the small contribution of the component. The hit list for this component contained compounds with very different RI's. The best hit based on RI was for 1-Ethyl-4-methyl-benzene, a compound known to be present in tomato (Petro-Turza, 1987).

The estimated elution profiles of the components are depicted in Figure 7.6. The estimated FWHM parameters are again around 1.6 RI units, thus the locations of these six components are within a range of 2.5 FWHM.

Note that the contributions of the fifth and sixth component (indicated by cyan, and black) are very small, as evident from Figure 7.6. Nevertheless they can be reliably estimated from these six tomato varieties.

7.3.2 Standard mixture data

Two benefits of global analysis will be illustrated in turn with case studies Mixture I and Mixture II on standard mixture data from Jonsson *et al.* (2005). The data were previously analyzed in Jonsson *et al.* (2005) with a self-modeling method, hierarchical MCR-ALS. The example in Mixture I shows the ability of global analysis to resolve components with small amplitude relative to a large peak nearby. The ability of global analysis to resolve highly overlapping components is illustrated by the example in Mixture II. Both examples will proceed using only a small amount of data (6 or 3 data sets), whereas Jonsson *et al.* (2005) used a large amount of designed data (62 data sets) that are analyzed simultaneously. The data are designed in that the concentration of some components is made to vary between datasets. Four representative traces from six standard mixture datasets are depicted in Figure 8. Note that the peak at $m/z = 327$ is about 200 times smaller than the maximum of the next peak at m/z values of 73 and 219. Furthermore the presence of the baseline aggravates

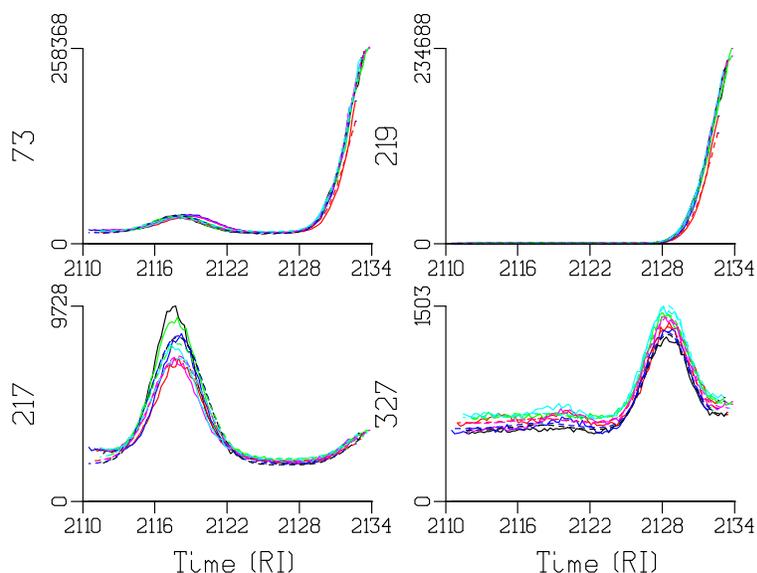


Figure 7.8: **Mixture I.** Representative traces (aligned with estimated shift parameters) at m/z values between 73 and 327. Colors indicate the six different data sets, solid lines are data and dashed lines indicate fit.

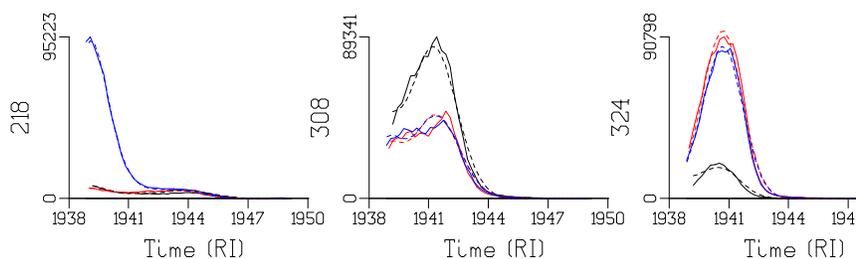


Figure 7.9: **Mixture II.** Representative traces (aligned with estimated shift parameters) at m/z values of 218, 308 and 324. Colors indicate the three different data sets, solid lines are data and dashed lines indicate fit.

this resolution problem.

The red mass spectrum in Figure 7.11 was identified as trans-cafeic acid-3TMS (MF 949, RMF 949), and the green baseline spectrum shows TMS related peaks near 73 and 147. The blue and black mass spectra could not yet be identified. Although hits with relatively good matching factors were retrieved, the library RI's for these compounds did not match the experimental RI. The

global analysis algorithm with nonnegativity constraints on the mass spectra results in estimates for the black mass spectrum that are zero at masses below 217. Apparently it was not possible to estimate its small amplitude (relative to the huge caffeic acid peak) near e.g. $m/z = 73$. The estimated elution profiles of the components depicted in Figure 7.10 highlight the small black component contribution.

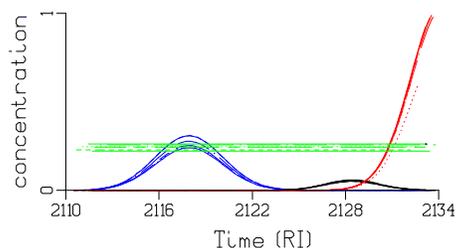


Figure 7.10: **Mixture I**. Estimated elution profiles with colors (corresponding to those in Figure 7.11) indicating the four components, and linetypes indicating the different experiments. The contribution of each component to each data set is the product of each elution profile times the associated normalized mass spectrum times the scaling factor of 243828. In order to improve visibility of the small concentrations, we have used a square root scaling of the maxima. Thus e.g. the black concentration maximum at about 0.05 corresponds to a true maximum of about 0.0025.

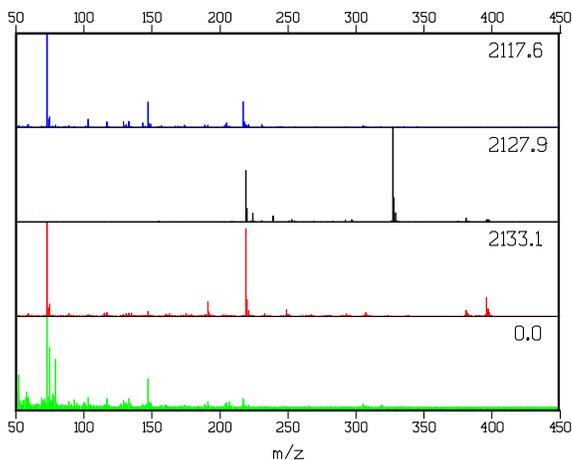


Figure 7.11: **Mixture I**. Normalized mass spectra with location λ of peak maximum indicated in the upper right corner. Zero is the baseline spectrum.

name	RI	MF	RMF	RI _{lit}
L-tyrosine	1938.9	795	882	1939
galacturonic acid	1938.9	755	816	1940
coniferylalcohol	1940.6	911	931	1944
trans-p-coumaric acid	1941.2	887	900	1947
galacturonic acid	1942.8	841	874	1961
trans-caffeic acid (3TMS)	2133.1	949	949	2141

Table 7.2: Identifications for case studies Mixture I and Mixture II described in Section 7.3.2.

The analysis for the Mixture I case study concerns the region from RI 2110-2134. Likewise we analyzed all bands in the region from Retention Index 1902-2433 in small time windows, using only 3 or 6 of the datasets simultaneously. Compared to the hierarchical MCR-ALS methodology described in Jonsson *et al.* (2005), global analysis resolves many more small components using only a very limited amount of data. The Mixture II case study concerns a particularly difficult overlap region from Retention Index 1938-1945. Five different components are present, with two pairs almost completely overlapping. In the three different data sets the amounts of these overlapping components were different by experimental design, thus allowing to resolve them (analogous to the natural component variability with tomato varieties).

Note the large differences in the shapes of the elution profiles at the three characteristic masses in Figure 7.9, and also the differences in the amplitudes of the datasets, as indicated by the different colors. The estimated mass spectra shown in Figure 7.13 were successfully identified, and the matching factors are shown in Table 7.2. Strikingly, just outside of this region two more mass spectra were identified as galacturonic acid at RI 1930 (MF 808 / RMF 829) and RI 1953 (MF 862 / RMF 875). Jonsson *et al.* (2005) only reported two galacturonic acid peaks at RI 1940.2 and 1956.5. The difference between our RI 1953 and their RI 1956.5 can be well explained by absolute alignment uncertainty. They also report trans-p-coumaric acid at 1943.5 and coniferylalcohol at 1942.8, which is also about 3 RI units higher. Alignments of 3 RI units between their samples are common. We currently have no explanation for the two extra putative galacturonic acid peaks at RI 1930 and RI 1942.8. The estimated elution profiles of the components depicted in Figure 7.12 demonstrate again the importance of amplitude variability for the resolution of overlapping components. Note also that the solid red elution profile is delayed relative to the dotted and dashed one.

7.3.3 Refinement of results

To further check the adequacy of the assumed EMG peak shape, we estimated concentration profiles according to the refinement described in Section 7.2.5. The profiles in Figure 7.14 are very similar to the EMG shapes in Figures 7.3,

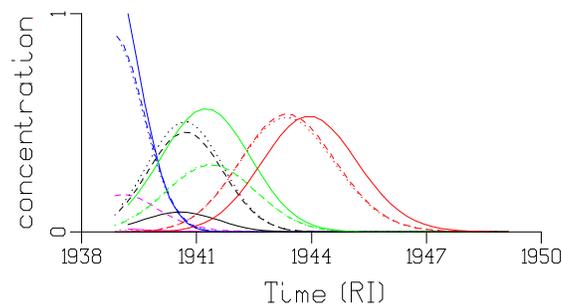


Figure 7.12: **Mixture II.** Estimated elution profiles with colors (corresponding to those in Figure 7.13) indicating the five components, and linetypes indicating the different experiments. The contribution of each component to each data set is the product of each elution profile times the associated normalized mass spectrum times the scaling factor of 538383.

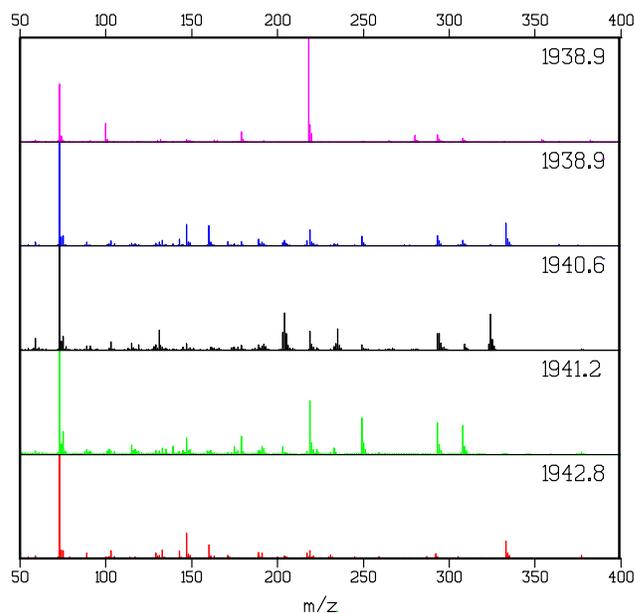


Figure 7.13: **Mixture II.** Normalized mass spectra with location λ of peak maximum indicated in the upper right corner.

7.6, 7.10, and 7.12. The improvement in the weighted root mean square error of the fit was 36% and 15% in case studies Tomato I and II, and 29% and 58% in

case studies Mixture I and II. The green elution profiles of Mixture II deviate most from an EMG peak shape.

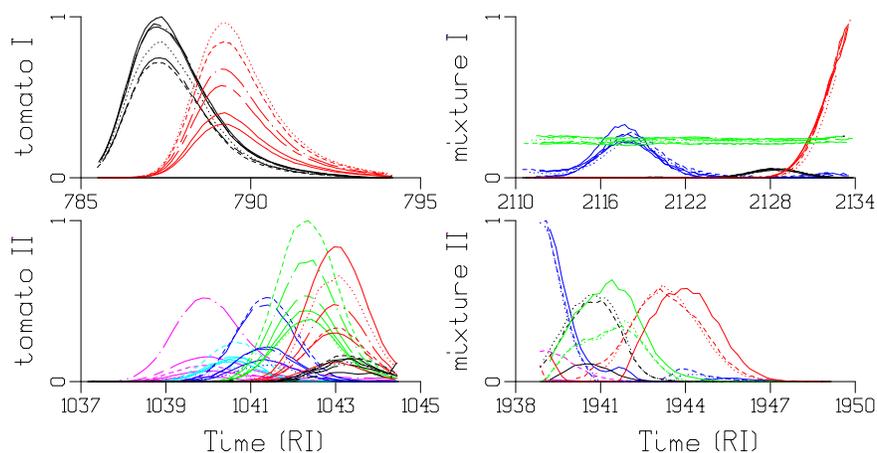


Figure 7.14: Elution profiles estimated for the four case studies considered in Sections 7.3.1 and 7.3.2 after the refinement described in Section 7.2.5, with the scaling conventions as in Figures 7.3, 7.6, 7.10, and 7.12.

7.4 Conclusions

Benefits of global analysis are: (1) direct quantitative estimation of parameters of interest, and (2) the quality of the fit can be judged from residual analysis. The parameters of interest are the mass spectra of the components, and their elution profiles, in particular the Retention Index that can be calculated from the location parameter. The precision of these parameters can be estimated as well. If the singular vectors of the residual matrix show only little structure, this indicates that the data have been fitted up to the noise level, and thus all information has been extracted. If they do show structure, this can indicate that the assumed EMG shape is not fully adequate, or it can indicate that an extra component is needed to fit the data. These benefits apply when analyzing single or multiple datasets. However, with simultaneous global analysis of multiple datasets there are more benefits. Overlapping components can more easily be resolved when they are present in different amounts, which occurs naturally when comparing biological samples. The estimated parameters (in particular

the mass spectra) are more robust against systematic measurement errors. The nuisance parameters (most importantly for alignment) can be directly estimated as well. The model allows the imposition of common shapes for elution profiles across data sets, and common retention time differences between components across data sets.

Outliers and saturation effects can be dealt with systematically.

Compared to hierarchical Multivariate Curve Resolution (a self modeling method described in e.g., Jonsson *et al.* (2005)) it appears that global analysis resolves many more small components using only a very limited amount of data. The strengths and limitations of global analysis as a component resolution methodology will be compared to those of MCR-ALS by means of simulation in Chapter 8.

Acknowledgments

Yury Tikunov and Harrie Verhoeven of Plant Research International kindly provided the tomato data discussed in Section 7.3.1. Thomas Moritz, Pär Jonsson and Krister Lundgren from Umeå Plant Science Center generously provided the standard mixture data described in Section 7.3.2. Francel Verstappen, Egon Willighagen, Arjen Lommen, Hans van Beek, Thomas Binsl, Yury Tikunov and Harrie Verhoeven are thanked for helpful discussions. An author of the version of this chapter submitted for publication, Velitchka V. Mihaleva, was supported by the Netherlands Bioinformatics Centre (NBIC).

Chapter 8

Resolution of co-eluting components in mass spectrometry data via multivariate curve resolution alternating least squares and global analysis¹

8.1 Introduction

The component resolution problem for a $m \times n$ matrix of data D_1 can be stated as the problem of estimating the matrices C_1 and S from D_1 in

$$D_1 = C_1 S^T \quad (8.1)$$

such that the matrix C_1 is $m \times n_{\text{comp}}$, and each column represents the evolution of a component of D_1 in the variable with which the rows of D_1 are resolved, and such that the matrix S is $n \times n_{\text{comp}}$, and each column represents the component of D_1 in the variable with which the columns of D_1 are resolved. For chemistry applications the components often correspond to the different chemical compounds in the sample underlying the measurements, and the problem is sometimes referred to as *deconvolution*.

¹A version of this chapter will appear as van Stokkum, Mullen, and Mihaleva (2008a) in *Chemometrics and Intelligent Laboratory Systems*.

It is often of interest to perform component resolution of many related data matrices D_1, \dots, D_K simultaneously, where the datasets are related in that they represent components with the same properties in the variable with which the columns of D_i are resolved, giving rise to the equation

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_K \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_K \end{bmatrix} S^T \quad (8.2)$$

By analogues of the Beer-Lambert law, the linear relation contained in Equations 8.1 and 8.2 well-describes many varieties of two-way data arising in spectroscopy and spectrometry experiments. The stochastic element of the model is neglected in Equations 8.1 and 8.2, but is often assumed to be comprised of additive Gaussian white noise, as we will assume throughout.

Algorithms to solve the component resolution problems as formulated above may be roughly categorized as those that use constraints but no parametric model, and those that employ a parametric (that is, functional) description for some aspect of the data. The former category of algorithms are often termed self-modeling curve resolution (SMCR) techniques, and have been recently reviewed (Jiang *et al.*, 2004). An important self-modeling curve resolution technique that we will concentrate on here is multivariate curve resolution alternating least squares (Tauler *et al.*, 1995; Tauler, 1995; de Juan and Tauler, 2003), which has a long and successful history of application to a variety of analytical problems (Garrido *et al.*, 2008).

The class of algorithms that employ a parametric model for some aspect of the data includes global analysis algorithms that describe each component of C_i in terms of a nonlinear function, and treat the entries of S as conditionally linear parameters that are not described functionally. Global analysis methods have been widely applied to component resolution problems in spectroscopy (Golub and LeVeque, 1979; Nagle, 1991b; van Stokkum *et al.*, 2004) and microscopy (Pelet *et al.*, 2000; Verveer *et al.*, 2000) applications, where the optimized parameters of the functional description of C_i are interpreted physically, allowing insight into the dynamical processes underlying the data. Applications of global analysis are described more fully in Chapters 5 and 6 of this monograph. Methods that are hybrid in the sense that they use MCR-ALS to iteratively improve estimates for C_1, \dots, C_K and S , but refine the MCR-ALS estimates for C_1, \dots, C_K by fitting them with a parametric model each time C_1, \dots, C_K is updated have also been investigated (de Juan *et al.*, 2001, 2000).

In this study, we will consider MCR-ALS and global analysis for component resolution problems in mass spectrometry data. Whereas MCR-ALS has a well-developed history of successful application in this domain (Jonsson *et al.*, 2005; Peré-Trepat *et al.*, 2007; Mas *et al.*, 2007; Peré-Trepat *et al.*, 2005), software tools for the application of global analysis methods to mass spectrometry data are described here for the first time. Chapter 7 describes via several case studies a proof-of-concept of the utility of global analysis for component resolution

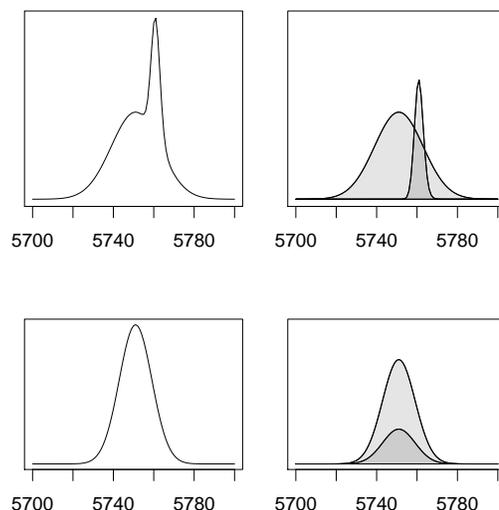


Figure 8.1: The plots on the left depict slices of a chromatogram in which overlapping elution profiles are represented, shown in the plots on the right. The elution profiles in the top plots are overlapping, but not completely so, allowing a variety of algorithms to be applied for component resolution. The plots on the bottom involve completely overlapping components, and only by simultaneously analyzing datasets in which the amplitudes of the components vary is resolution possible, with either MCR-ALS or global analysis.

problems arising in gas chromatography mass spectrometry data.

Mass spectrometry data associated with a particular sample is often stored as a matrix D_i resolved with respect to time and mass-to-charge ratio (m/z), so that each data point is a count of the number of molecular fragments having a given m/z measured at a given time. Then C_i represents elution profiles of components in time and S represents mass spectra. The aim of component resolution in the context of mass spectrometry datasets D_1, \dots, D_K is to determine the mass spectra of components and, possibly, obtain the retention times of components, which is the time at which an elution profile is maximal. Estimated mass spectra and retention times are matched against mass spectra and retention times of known compounds stored in a library (Stein and Scott, 1994), allowing the compounds represented by the estimated spectra to be identified. The relative concentration of a particular component k in the samples represented by datasets D_i and D_j may be estimated by the quotient between the area of the estimated elution profiles of the component, that is, by $C_i[, k]/C_j[, k]$.

In the case that the elution profiles of two or more components are overlapping, the components are said to be *co-eluting*, as in Figure 8.1. If the profiles

overlap exactly and have the same shape, as in the bottom panel of Figure 8.1, only by analyzing several datasets simultaneously in which the amplitude of the profiles varies are the components possible to resolve. Much attention has been paid to the resolution problem in the case of co-eluting components (Tauler, 1995; Peré-Trepat *et al.*, 2005). MCR-ALS is currently considered to be the unique method capable of resolving components in the case of completely overlapping elution profiles (Jonsson *et al.*, 2005). We present global analysis as an alternative component resolution methodology for the case that compounds are completely co-eluting, and make available open source implementations of MRC-ALS and global analysis in the form of packages for the R language and environment (R Development Core Team, 2008), so that others may reproduce the examples discussed here and possibly extend the methodology.

The remainder of this study is organized as follows: Sections 8.2 and 8.3 introduce MCR-ALS and global analysis, respectively. Section 8.4.1 shows the performance of the methods in terms of the matching factor of estimated mass spectra in a simulation study inspired by gas chromatography mass spectrometry (GC/MS) data discussed in Chapter 7. Section 8.6 contains conclusions.

8.2 Multivariate curve resolution alternating least squares (MCR-ALS)

While MCR-ALS has been presented many times in the literature, we present it here again in order to make the rest of discussion of the methodology more concrete. Since it is known in advance that negative values in an elution profile or mass spectra are not physically interpretable, we present the basic algorithm with non-negativity constraints on both C_1, \dots, C_K and S .

Algorithm 1 MCR-ALS($C_1, \dots, C_K, S, D_1, \dots, D_K$)

```

1:  $D_{All} := \text{rbind}(D_1, \dots, D_K)$ 
2: for  $i = 1, 2, \dots$ , until stopping criteria are met do
3:   if  $i$  is even then
4:      $S := \text{getS}(C_1, \dots, C_K, S, D_{All})$ 
5:   else
6:      $C_1, \dots, C_K := \text{getC}(C_1, \dots, C_K, S, D_1, \dots, D_K)$ 
7:   end if
8: end for
9: return  $C_1, \dots, C_K, S$ 

```

Algorithm 1 alternates calls to Algorithms 2 and 3, optimizing C_1, \dots, C_K and S according to non-negative least squares criteria while considering either S or C_1, \dots, C_K as fixed. The stopping criterion is usually based on the change in the residual difference between two iterations falling beneath some threshold. Note that ‘rbind’ refers to binding matrices together along their rows, and ‘nrow’ and ‘ncol’ refer to the number of rows and columns in a given matrix,

respectively. ‘NNLS’ refers to a non-negative least squares algorithm such as that of Lawson and Hanson (1974, 1995), a FORTRAN implementation of which is freely available via the Netlib repository (Browne *et al.*, 1994) and in the R package **nls** (Mullen and van Stokkum, 2007a).

Algorithm 2 `getS($C_1, \dots, C_K, S, D_{All}$)`

```

1:  $C_{All} := \text{rbind}(C_1, \dots, C_K)$ 
2: for  $i = 1, 2, \dots, \text{ncol}(D_{All})$  do
3:    $S[i, ] := \text{NNLS}(C_{All}, D_{All}[i, ])$ 
4: end for
5: return  $S$ 

```

Algorithm 3 `getC($C_1, \dots, C_K, S, D_1, \dots, D_K$)`

```

1: for  $j = 1, 2, \dots, K$  do
2:   for  $i = 1, 2, \dots, \text{nrow}(D_j)$  do
3:      $C_j[i, ] := \text{NNLS}(S^T, D_j[i, ])$ 
4:   end for
5: end for
6: return  $C_1, \dots, C_K$ 

```

In practice, Algorithm 1 is often modified to include other constraints on C_1, \dots, C_K and S , the most important of which are described in Section 8.2.1. Before MCR-ALS based on Algorithm 1 can be applied, it is necessary to decide what constraints to include, what number of components (i.e., columns of C_j and S) to use, and what the starting estimates for either C_1, \dots, C_K or S should be. These prerequisites are described in turn.

8.2.1 Constraints for use in MCR-ALS

While MCR-ALS works in the absence of a parametric model, it allows the application of constraints to the elution profiles and mass spectra. Commonly applied constraints include non-negativity of the elution profiles and mass spectra (Bro and Jong, 1997; Benthem and Keenan, 2004), unimodality of elution profiles (Bro and Sidiropoulos, 1998), selectivity or equality constraints that account for intervals of data or datasets where a component is known to have zero amplitude or known amplitude (Tauler *et al.*, 1995), and normalization or closure constraints that help avoid problems associated with intensity ambiguities (Tauler, 2001).

Unimodality constraints may be desirable in the case that the estimated elution profiles returned by Algorithm 1 have a bi-or-multi-modal shape that is believed to be unlikely given knowledge of the experimental set-up. For the application of unimodality constraints a simple modification of Algorithm 1 suffices; before returning an elution profile, the unimodal vector that is closest to

the estimated elution profile in a least squares sense is determined and returned. The unimodal vector to be returned may be efficiently determined via a method based on isotonic regression (Turner and Wollan, 1997) and implemented in the R package **Iso** (Turner, 2005).

In the case that it is known that a component is not present in a given dataset, a selectivity constraint may fix its elution profile to zero (likewise, if the component has a known elution profile, the profile may be fixed at the known shape). A heuristic that has been successfully applied to the analysis of GC/MS datasets checks if the retention time of an elution profile is within some threshold value of the median retention time for that component over all datasets. If not, the elution profile is set to zero (that is, its contribution is removed) in the deviating dataset (Jonsson *et al.*, 2005).

Normalization and closure constraints, which are discussed in more detail elsewhere (Tauler, 2001), may be of use to deal with the problem of intensity ambiguity. For any scalar m and elution profile c_i and mass spectrum s_i^T , $c_i m (1/m) s_i^T = c_i s_i^T$, meaning that increasing the estimate of c_i by a factor of m gives a model that fits the data equally well, as long as the mass spectrum s_i^T is multiplied by a factor of $1/m$. The normalization constraint is typically applied to the spectra, and may constrain $\|s_i^T\|_2 = 1$, or constrain the maximum of s_i^T to be one. A closure constraint is usually implemented on the rows C_1, \dots, C_K , and constrains the sum of the elements of each row of matrix C_j to be equal to a known constant, as for instance is desirable for reaction-based systems, where the elution profiles obey a mass balance equation. Typically application of either normalization or closure is desirable, but not both types of constraint simultaneously.

8.2.2 Data selection and pre-processing

In the case that the mass spectrometry data contain on the order of 1-10 components, it is possible to treat all data at once. Otherwise, for the case of complex samples containing hundreds or thousands of different components, the resolution problem is approached by dividing the datasets into time intervals, so that data D_1, \dots, D_K represent time windows of the full data that contain on the order of 1-10 components. Typically the time-windows are demarcated by points of low intensity, so that each set of data matrices D_1, \dots, D_K contains the entire elution profile of the represented components (Jonsson *et al.*, 2005).

In addition to time-window selection, baseline subtraction, alignment of datasets by peak matching and data smoothing are commonly performed prior to application of MCR-ALS. These pre-processing techniques are something of an art in themselves, and the optimal set of techniques is highly dependent on the particular experimental conditions (Jonsson *et al.*, 2005).

8.2.3 Number of components

While many heuristics exist in the literature for the determination of the number of components present in a dataset D_i , methods based on principal component

analysis (PCA) (Wold *et al.*, 1987) or a singular value decomposition seem most popular (Golub and van Loan, 1996), and are based on the assumption that any components to be resolved contribute more to the data than the noise term. For application to spectroscopy data, a heuristic has been suggested that performs MCR-ALS with an increasing number of components until the greatest number of components is found such that the retention times of the estimated elution profiles are in the same order in each dataset (Jonsson *et al.*, 2005). If it is known in advance that there is a baseline term in the data, it may be modeled with an extra component not subject to unimodality constraints (Peré-Trepat *et al.*, 2005).

8.2.4 Starting estimates

Most heuristics for obtaining starting values derive estimates for C_1, \dots, C_K as opposed to S . Popular methods to obtain starting values for C_1, \dots, C_K involve the use of evolving factor analysis (Maeder, 1987) and the SIMPLISMA algorithm (Jonsson *et al.*, 2005; Windig and Guilment, 1991). Algorithm 1 is sensitive to starting values in that both the final estimates for C_1, \dots, C_K and S and the number of iterations required to meet stopping criteria depend on the initial values of C_1, \dots, C_K or S .

8.3 Parametric model-based global analysis

The recovery of C_1, \dots, C_K and S from D_1, \dots, D_K can be approached via global analysis methods based on a parametric model, as is described in Chapter 7. In the mass spectrometry context, a parametric model is applied to the description of the elution profiles C_1, \dots, C_K , so that each elution profile in each dataset is described by a peak function such as the exponentially modified Gaussian (EMG). The process of model fitting optimizes the parameters describing the elution profiles and treats the mass spectra as conditionally linear on their estimates. Usually the criterion optimized is the sum of squared differences between the model evaluated at the parameter estimates and the data. In this case the model fitting problem is an instance of separable non-linear least squares, which is possible to address with the variable projection algorithm (Golub and Pereyra, 2003). In mass spectrometry applications the data represent counts and can be expected to behave as a Poisson distributed stochastic process. Since the number of counts is usually large, minimization of the actual log-likelihood function would offer very little improvement over optimization of the sum of squared deviations (Maus *et al.*, 2001). Negative values of mass spectra have no physical interpretation so that it is desirable to constrain their estimates to non-negative values, which is possible to accomplish via the modification of the variable projection algorithm that adds constraints to the conditionally linear parameters, as described in Chapter 4.

8.3.1 Functions for the description of elution profiles

Many functions are possible to apply to the description of chromatographic peaks (Marco and Bombi, 2001). The exponentially modified Gaussian (EMG) function is the most popular in practice and can be used to describe a variety of peak shapes with a relatively small number of parameters. It employs four parameters, location λ , a full width at half-maximum (FWHM) Δ , rate k and amplitude a to determine each elution profile c over a vector of times t as

$$c(t, \lambda, \Delta, k, a) = a \frac{\exp(-kt)}{2} \exp(k(\lambda + k \frac{\tilde{\Delta}^2}{2})) \left\{ 1 + \operatorname{erf} \left[\frac{t - (\lambda + k \tilde{\Delta}^2)}{\sqrt{2} \tilde{\Delta}} \right] \right\} \quad (8.3)$$

where $\tilde{\Delta} = \Delta / (2\sqrt{2 \log(2)})$.

It is usually desirable to keep the total number of parameters describing the elution profiles as low as possible, which renders the model better determined and faster to optimize. This may often be accomplished by assuming that the shape of the elution profile corresponding to a given component is the same in C_1, \dots, C_K , but has an amplitude parameter that varies per-dataset. To address the problem of intensity ambiguity, one of the amplitude parameters is fixed for each component. The amplitude of each component is fixed in one dataset, so that the other amplitudes describing the component are relative to the fixed value and the model is well-determined. Provided all components are present in the D_1 , it is often convenient to fix the amplitude parameters of all components in this dataset to 1, while allowing all amplitude parameters in datasets D_2, \dots, D_K to remain free.

8.3.2 Data selection and pre-processing

As in MCR-ALS (Jonsson *et al.*, 2005), for global analysis the data is cut into time windows so that D_1, \dots, D_K represent on the order of 1-10 components, as described in Section 8.2.2. Instead of correcting for a baseline in the data, which can be problematic in the case that its contribution is changing in time, global analysis allows the possibility of modeling the baseline term as a component, assigned its own mass spectrum. It is also possible to allow the parameter describing the location of each elution profile to vary per-dataset, meaning that pre-alignment is not a prerequisite for data analysis.

8.3.3 Number of components

The estimation of the number of components proceeds along the same lines as for MCR-ALS, as described in Section 8.2.3. A baseline in the data may be described as a time-invariant component.

8.3.4 Starting estimates

Starting estimates for global analysis may be estimated by obtaining starting estimates as for MCR-ALS as described in Section 8.2.4, and then fitting the desired functional description of the elution profiles to these estimates. The

decision regarding which parameters to make common between elution profiles in different datasets is at present made by hand, though automation would be desirable. For numerical reasons, the starting values should not result in peaks associated with precisely the same parameter values (shifting the location of one peak slightly resolves any problems).

8.4 Comparison of the methods

MCR-ALS proceeds by iteratively solving constrained least squares problems, whereas global analysis requires the solution of a nonlinear optimization problem that usually takes the form of a separable nonlinear regression. For typical problems, MCR-ALS will be much faster than global analysis (requiring on the order of seconds as opposed to minutes). Furthermore, MCR-ALS requires significantly less hand-work in determining the model form.

Global analysis describes the data using a small number of free parameters as compared to MCR-ALS. For example, for the analysis of two datasets where each dataset represents two components having elution profiles described by an EMG with the same shapes but different amplitudes per-dataset, only 8 intrinsically nonlinear parameters completely determine the model. In contrast, the number of free parameters in each iteration of MCR-ALS is equal to the number of entries in C_1, \dots, C_K or S . Whether the large number of free parameters employed by MCR-ALS is an advantage or a liability depends on the specifics of the component resolution problem. For datasets in which the elution profiles are difficult to describe in terms of a model with a small number of parameters, MCR-ALS may provide a significantly better fit, due to its flexibility. However, as we will show in the remainder of this section, the relatively large number of parameters used by MCR-ALS may preclude the resolution of components in certain situations.

It is well-established that MCR-ALS is a useful and powerful component resolution tool. Therefore we will not concentrate on examples in which MCR-ALS is successful, which are abundant in the literature, but rather on those problems in which it performs poorly in terms of the matching factor of the estimated mass spectra as compared to global analysis. Many properties of the data affect the performance of both algorithms, such as the presence or absence of shape differences in the elution profiles, overlap/distance in time between locations of the elution profiles, amplitude differences of components between datasets, relative abundances of components, similarity of the underlying mass spectra, signal-to-noise ratio of the data, starting values, and number of datasets available for simultaneous analysis. The present study necessarily examines only a subset of the problem instances possible to encounter as these properties vary, though indicates some strengths and weaknesses of the algorithms that may be extrapolated to many other cases.

	component 1	component 2
location	5754	5755
FWHM	7	7
rate	1	1

Table 8.1: Tabulated are the parameters for the two EMGs used to describe the elution profiles in both datasets. The elution profiles have amplitudes 1 and 2, respectively, in dataset 1, and amplitudes 1.5 and 2 respectively, in dataset 2. The resulting EMGs are shown graphically in Figure 8.2.

8.4.1 A basic example

We consider an example in which the data consists of two simulated GC/MS datasets that each represent two co-eluting components, shown in Figure 8.2, with associated mass spectra shown in the top row of Figure 8.4. This simulated data (shown in Figure 6.2) is inspired by the Mixture I case study described in Chapter 7.3.2. The parameters for the EMGs used to simulate the elution profiles are given in Table 8.1. The datasets represent integer time points 5720-5800 and integer masses 50-449.

In order to introduce a stochastic element into the data (that is, noise) we let a deterministic data point represent the parameter λ of a Poisson distribution. The Poisson distribution has density

$$p(x) = \lambda^x \exp(-\lambda)/x! \quad (8.4)$$

for $x = 0, 1, 2, \dots$. The mean and variance are $E(X) = Var(X) = \lambda$. The signal-to-noise ratio (SNR) of a matrix of data with Poisson distributed noise is defined to be the square root of the maximum deterministic data point. To obtain data with SNR 1000 (which is encountered in GC/MS experiments) we scale the deterministic simulated dataset to contain 10^6 counts in the maximum data point, and then consider each data point to represent the λ in a Poisson process via the R function `rpois`.

In order to judge how well an estimated mass spectrum s resembles the mass spectrum u used in simulating the data, the normalized dot product matching factor function

$$\cos(u, s) = \frac{u \cdot s}{\|u\| \|s\|} \quad (8.5)$$

is useful (Stein and Scott, 1994; Alfassi, 2004). The normalized dot product matching factor of the spectra shown in the top row of Figure 8.4 is almost 0, as the spectra are very dissimilar. If the abundance ratio of components p and q is defined as

$$ab(p, q) = \frac{\text{sum}(C_j[, p]S[, p]^T)}{\text{sum}(C_j[, q]S[, q]^T)} \quad (8.6)$$

where $C_j[, p]$ and $C_j[, q]$ are the elution profiles p and q over all times, $S[, p]$ and $S[, q]$ are the mass spectra p and q over all masses, and the `sum` function takes the sum of all matrix elements. In this way Equation 8.6 represents the number

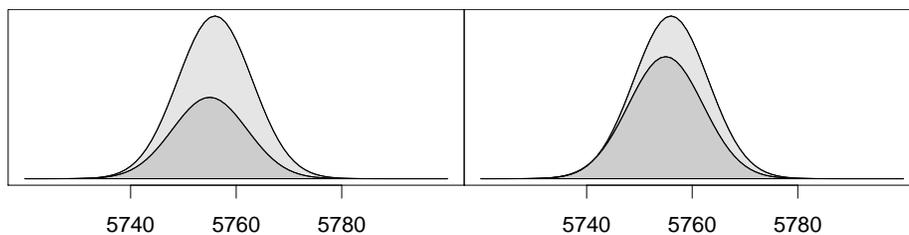


Figure 8.2: The elution profiles represented in the two simulated datasets. In dataset 1 (left) the profiles have slightly different amplitude compared to the profiles in dataset 2 (right). Both components are represented by EMG functions, with the location parameter of the first component (dark grey) 1 time unit before that of the second component (light grey). Parameter values used to generate these profiles over the vector of integer times 5720-5800 are given in Table 8.1.

of counts contributed to the data by the component p divided by the number of counts contributed by component q . Under this definition the abundance ratio of component 1 to component 2 is ≈ 2.1 in dataset 1 and ≈ 3.1 in dataset 2; the difference in abundance ratios between datasets is due to the difference in the amplitude of the first elution profile.

In applying MCR-ALS to resolving components in the simulated data, we apply constraints for non-negativity of C_1, \dots, C_K and S , unimodality of C_1, \dots, C_K , and normalization of each mass spectrum such that the maximum value is one. For global analysis, the mass spectra and amplitudes of the elution profiles are constrained to non-negative values. The starting values for both MCR-ALS and global analysis are taken to be the elution profiles used in simulation, but with location parameters shifted to 5757 and 5753. The stopping criterion used for MCR-ALS is reduction of the residual difference by no more than .001 between iterations or completion of 100 iterations. The stopping criteria for global analysis were the defaults used by the Levenberg-Marquardt nonlinear regression algorithm implemented in the R package `minpack.lm`, or completion of 50 iterations. In all problems considered in this study, allowing the algorithms to run for more iterations results in further reductions in the sum of squared errors but does not change the matching factor of the estimated mass spectra with the mass spectra used in simulation by more than $\approx 5\%$.

Figure 8.4 shows that for this problem, MCR-ALS does not resolve the spectral signature of both components, whereas global analysis estimates the mass spectra well. The normalized dot product matching factor of the estimated spectra and the spectra used in simulation over 100 noise realizations of the problem were calculated. Global analysis results in matching factors for both components that are always $> .99$, whereas MCR-ALS does not well-resolve both spectra. By examination of many stochastic realizations of the data, it is clear that the results of both algorithms are stable with respect to this level of noise. We turn to a more thorough simulation study to further investigate the

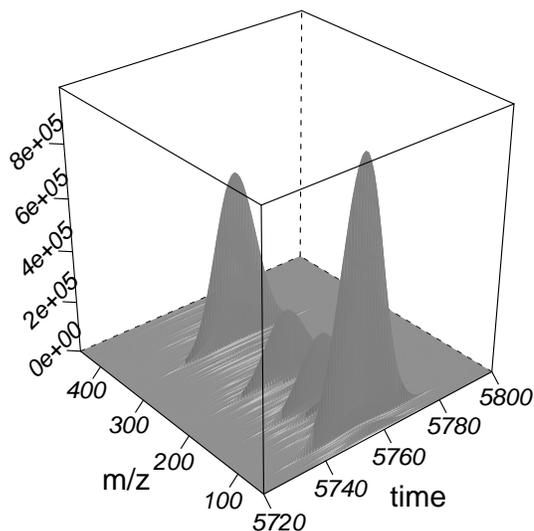


Figure 8.3: An overview of the simulated data. Only dataset 1 is shown. The signal-to-noise ratio in both datasets is 1000.

resolution power of the techniques.

8.4.2 Varying distance between components

The example put forth in Section 8.4.1 can be studied as the location of component 1 with respect to the location of component 2 is made to vary. We simulate pairs of datasets with different distances between the locations of the components, as collated in Table 8.2 and shown graphically in Figure 8.5. All other experimental parameters are as described previously in Section 8.4.1.

For each pair of datasets, 25 different stochastic realizations are generated. The left plots in Figure 8.6 show the average matching factor of the mass spectra estimated by MCR-ALS and global analysis with the mass spectra used in simulation over 25 stochastic realizations of each pair of datasets. As the distance between the components increases, MCR-ALS is better able to solve the problem, but does not resolve components when the distance between the locations of the components is small. Global analysis is able to estimate both mass spectra well in all cases. When the distance between the components is reduced to zero, both global analysis and MCR-ALS do not resolve the underlying mass spectra. Both methods also fail in the case that only one of the two datasets is analyzed.

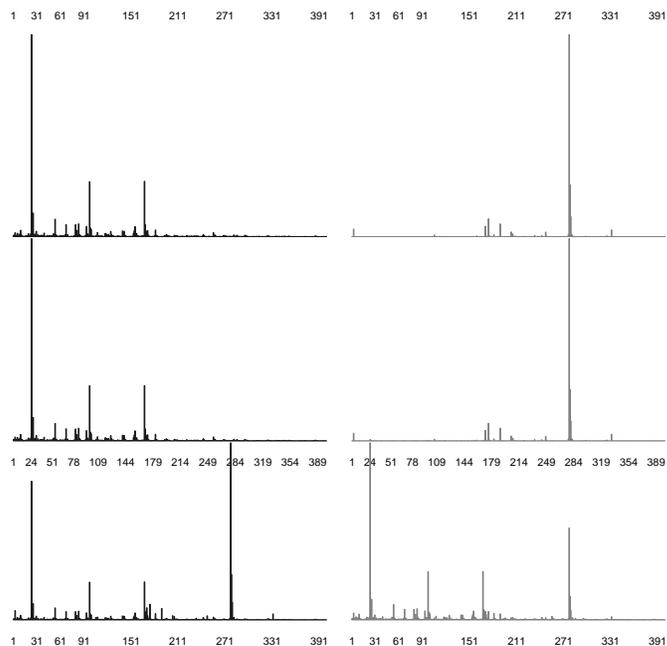
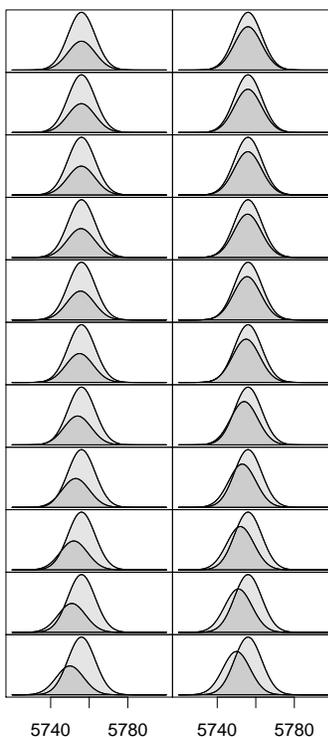


Figure 8.4: The spectra on the top row are those used in simulating the data. The second row contains representative spectra estimated by global analysis. The third row contains representative spectra estimated by MCR-ALS, in which the spectra are not well resolved.

dataset pair	1	2	3	4	5	6	7	8	9	10	11
location separation	0.01	.05	.1	.25	.5	1	2	3	4	5	6

Table 8.2: Eleven pairs of datasets are simulated for the studies in Sections 8.4.2 and 8.4.3, each of which represents elution profiles over integer times 5720-5800 with different distances between the location of the components. The resulting EMGs are shown graphically in Figure 8.5.

Figure 8.5: Elution profiles used in simulating the pairs of datasets used in Sections 8.4.2 and 8.4.3; each row represents a pair of datasets with a different location of the first component. Each elution profile is represented by an EMG having the parameter values given in Table 8.1, except for the location of the first component, which is less than the location of the second component by the values given in Table 8.2.



8.4.3 Mass spectra with increased matching factor

We have performed the same simulation study as in Section 8.4.2 but using mass spectra with a larger matching factor (.31 as opposed to 0), again taken from a case study described in Chapter 7, which are shown in Figure 8.7. We again simulate pairs of datasets with varying locations of the first component, as collated in Table 8.2 and shown graphically in Figure 8.5. The right plots in Figure 8.6 show the average matching factor of the mass spectra estimated by MCR-ALS and global analysis with the mass spectra used in simulation, again as averages over 25 stochastic realizations of each pair of datasets. As in Section 8.4.2, global analysis is better able to estimate the underlying mass spectra when the separation between the location of the components is small.

8.4.4 Varying SNR

Returning to the study in Section 8.4.2, we choose a distance between the components for which both global analysis and MCR-ALS estimated the mass spectra of both components well, namely the problem instance with elution profiles separated by 6 time units described in Section 8.4.2, and a distance for which global analysis succeeds in estimating the mass spectra well, but MCR-ALS does not, namely the problem instance with the elution profiles separated by 1 time unit

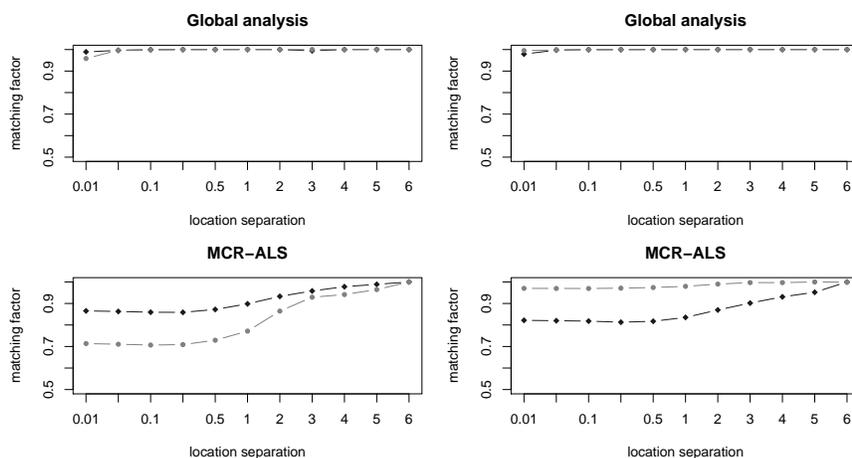


Figure 8.6: The normalized dot product matching factor of the estimated spectra and the spectra used in simulation for global analysis (top) and MCR-ALS (bottom). Results for component 1 are shown as black diamonds, and for component 2 as grey circles. As the separation of the components is increased, the component resolution problem is rendered progressively easier. The left plots are for the data in Section 8.4.2, whereas the right plots are for the data in Section 8.4.3. MCR-ALS well-resolves the spectra only at a separation of 6.

described in Section 8.4.2.

We simulated these problem instances for SNR 500, 200, 100 and 25, to examine the robustness of the results to noise, examining the performance of the algorithm in terms of the matching factor of the estimated spectra with the mass spectra used in simulating the data. For all these SNRs, for both problems, the algorithms performed qualitatively the same as under SNR 1000, with differences of less than $< 5\%$ in the resulting average matching factors of the estimated spectra compared to the values used in simulation over 25 noise realizations for each SNR considered. Hence the results described in previous sections are stable with respect to the noise level of the data.

8.4.5 Sensitivity to starting values and stopping criteria

For the problem considered in Section 8.4.1, using starting values for the elution profiles that are slightly shifted (e.g., by three time units) from the values used to simulate the elution profiles, MCR-ALS continues to reduce the sum of squared errors for many iterations (> 10000) provided the stopping criterion is altered to allow continuation even if the differences in the residuals between iterations are very small. Eventually MCR-ALS reaches the same minimum in sum of squared errors it finds in only a few iterations when given perfect starting values for the elution profiles. However, the estimated spectra do not well-represent the

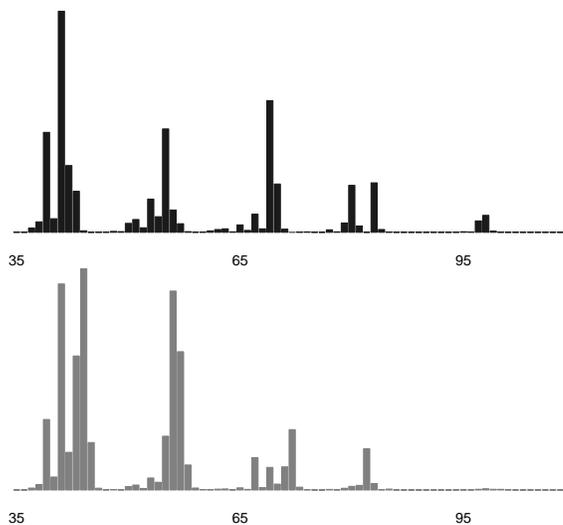


Figure 8.7: Mass spectra associated with the two components underlying the data in Section 8.4.3. The normalized dot product matching factor of the spectra is .31.

spectra used in simulation, and are associated with a matching factor of $> .9$. There are therefore a range of values for the elution profiles and mass spectra that result in the same sum of squared errors under MCR-ALS. That is, there are bands of feasible solutions, not a unique description of the elution profiles and mass spectra that result in the best model fit. This is due to the relatively large number of parameters used for the description of the elution profiles under MCR-ALS, and is an issue that has been studied in detail elsewhere (Tauler, 2001; Garrido *et al.*, 2005). Figure 8.8 shows that shifting the starting values for the elution profiles from the values used in simulation for the problem described in Section 8.4.1 results in varying estimates for the mass spectra under MCR-ALS. This is in contrast to the estimates for the mass spectra returned by global analysis, which always well approximate the mass spectra used in simulation.

8.4.6 Sensitivity to abundance ratio of components

By increasing the magnitude of the first mass spectrum relative to the magnitude of the second mass spectrum used in simulating the data, it is possible to vary the number of counts contributed to the data by each of the two components, so that the abundance ratio defined in Equation 8.6 varies. For the problem described in Section 8.4.2 where the distance between the location of the components is 6 time units, and both MCR-ALS and global analysis estimate the components

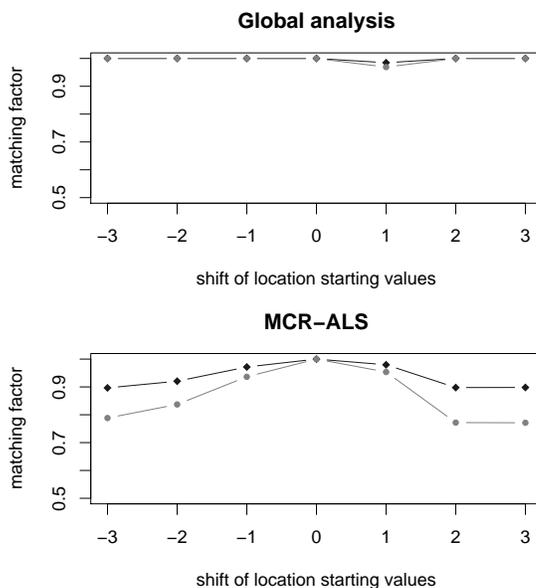


Figure 8.8: The normalized dot product matching factor of the estimated spectra and the spectra used in simulation for global analysis (top) and MCR-ALS (bottom) as the starting value for the location of the elution profiles is shifted in time units from the values used in simulation. Results for component 1 are shown as black diamonds, and for component 2 as grey circles. MCR-ALS converges to mass spectra estimates that are different from the actual underlying mass spectra when the starting values are less than perfect. The results shown are average matching factors over 10 stochastic realizations of the data.

well, we examined how the performance varies in terms of the estimated mass spectra as the abundance ratio of component 1 to component 2 is made to vary. We also examined this same problem but using the spectra shown in Figure 8.7. The breakdown of the resolution power of the techniques was qualitatively similar as the abundance of component 1 to component 2 increases, as shown in Figure 8.9.

8.5 Implementation and future work

The MCR-ALS and global analysis algorithms discussed here are publicly available under the terms of the GNU General Public License in the form of the R packages **ALS** (Mullen and van Stokkum, 2008a) and **TIMP** (described in Chapter 5), respectively, from the Comprehensive R Archive Network (CRAN) (R Development Core Team, 2008). The package includes scripts to repeat the results in this paper.

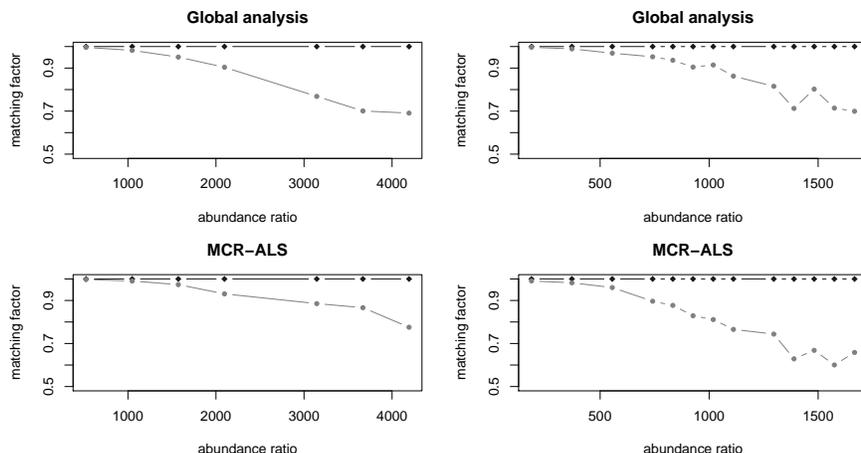


Figure 8.9: The normalized dot product matching factor of the estimated spectra and the spectra used in simulation for global analysis (top plots) and MCR-ALS (bottom plots) as the abundance of the first component relative to the second is made to increase. Matching factors shown are averages over 25 stochastic realizations of the data. The left and right plots are for data with the underlying mass spectra shown in the top row of Figure 8.4 and in Figure 8.7, respectively.

While global analysis is a powerful component resolution tool, much additional work is necessary to render it a suitable tool for high-throughput applications. Of particular importance are the development of heuristics for model specification.

Other tools for component resolution of mass spectrometry data such as the “Automated Mass Spectral Deconvolution and Identification System” (AMDIS) by the National Institute of Standards and Technology (Stein, 1999) have the great advantage of being much more fully automatic, and are integrated with libraries of mass spectra so as to best facilitate the end goal of compound identification. However they are not able to deal with the problem of completely overlapping components or multiple datasets. The ideal software would combine the best features of all available methodologies. Then when one of the faster methods based on peak-finding in the chromatogram or MCR-ALS fail to resolve components, global analysis could be applied. This would require the formulation of heuristics to decide when application of global analysis is likely to offer improved estimates of the underlying spectra. Possible heuristics could be based on rank analysis of the data, or prior knowledge indicating the expected number of co-eluting components present.

8.6 Conclusions

This study has described the global analysis and MCR-ALS methodologies for component resolution in mass spectrometry data. The methods have been explored in simulation studies in which the elution profiles used to generate the data are completely overlapping. Problem instances have been presented in which global analysis is able to estimate the mass spectra of all components well, whereas MCR-ALS fails to do so. Such problem instances arise when elution profiles are completely overlapping, making the component resolution task difficult. Application of global analysis to experimental data in which component resolution is significantly complicated by other issues such as instrument saturation, outliers and baseline drift is considered in Chapter 7, where indications are found that global analysis is better able than MCR-ALS to resolve small components given these complications.

While the global analysis methodology is a powerful component resolution method, especially for data representing co-eluting compounds, it requires more run-time, and at present is less automated than other techniques based on peak-finding and MCR-ALS. Possible improvements in accuracy offered by the method must therefore be carefully weighed against losses in efficiency. Further study and development of the method for mass spectrometry applications is warranted.

Chapter 9

Summary

The studies described in this monograph are concerned with separable nonlinear models that have the form of a linear combination of nonlinear functions with a stochastic component comprised of additive Gaussian white noise. The free parameters of such models are possible to estimate using the variable projection algorithm due to Golub and Pereyra. Variable projection is introduced in Chapter 2 with special emphasis on applications to modeling matrix data. Separable nonlinear models for data arising in time-resolved spectroscopy, microscopy and mass spectrometry experiments are also introduced in Chapter 2. These application domains have motivated many of the investigations in this monograph. In them, and in other application areas in physics and chemistry, the use of a unified separable nonlinear model to describe all measurements collected over multiple independent variables, possibly over the course of many experiments, is a powerful tool for extracting information on a system of interest. The use of separable nonlinear models in this way is often referred to as *global analysis*.

Chapter 3 describes the properties of variable projection variants and the NIPALS algorithm for alternating least squares for treatment of the problem of fitting a sum of exponential functions under least squares criteria. A new derivation of the Fisher information matrix under the full Golub-Pereyra variable projection algorithm allows a numerical comparison of parameter precision under variable projection variants. It is found that the analytical Kaufman approximation to the full variable projection functional is appropriate for a sum-of-exponentials fitting problem that is typical of problems that arise often in time-resolved spectroscopy. It was also shown for this problem that the linear approximation standard error estimates for parameters under variable projection algorithms are an acceptable approximation for likelihood-based standard error estimates.

Chapter 4 describes the modification of the variable projection algorithm to accommodate constraints on the linear coefficients in a sum of nonlinear functions. Such constraints are often desirable if only certain values of these linear parameters are physically interpretable. For instance, non-negativity constraints are often desirable on linear parameters that represent fluorescence spectra or

mass spectra, since negative values of such parameters are not interpretable. For the case that non-negativity constraints are applied to the linear parameters, we show that active constraints result in an increase in precision of the intrinsically nonlinear parameter estimates relative to estimates obtained with classical variable projection.

Chapter 5 presents a framework for fitting separable nonlinear models in physics and chemistry applications in the form of the package **TIMP** for the R language and environment for computational statistics. The package facilitates flexible specification of models, efficient optimization of free parameters, and validation of the fit. The design allows the definition of new model types with minimal effort. Implementation in R facilitates easy integration with other statistical routines and allows the package to be used on all major operating systems. Both R and **TIMP** are freely available in source code form under the terms of the GNU General Public License.

The use of separable nonlinear models to describe fluorescence lifetime imaging microscopy (FLIM) data is investigated in Chapter 6, with emphasis on testing the utility of the package **TIMP** to estimate free model parameters. It is found via simulation studies and a control experiment that the methodology is indeed of great promise for modeling FLIM data. Applications to mapping protein-protein interactions are described in brief in Chapter 2 and further studies using separable nonlinear models for this purpose are currently in progress.

Chapters 7 and 8 describe the use of separable nonlinear models for component resolution problems in time-resolved mass spectrometry data. This global analysis methodology is an alternative to the nonparametric (self-modeling) component resolution techniques that are usually applied in the problem domain. The proof-of-concept application of global analysis to the solution of component resolution problems in gas chromatography mass spectrometry (GC/MS) data is described in Chapter 7. Then by way of simulation studies, it is shown in Chapter 8 that global analysis is better able than a widely applied self-modeling method to estimate the underlying mass spectra from GC/MS data in certain cases in which the elution profiles are very overlapping.

9.1 Future work

Chapter 2.6 sketches some ideas for further development of algorithms for separable nonlinear optimization problems. These topics are currently under investigation.

Many extensions of the package **TIMP** are currently in progress or of interest. Options for modeling reaction kinetics as described by kinetic theory models are in development in collaboration with David Nicolaidis. As discussed in Chapter 2.3, separable nonlinear models have proved useful for the analysis of Nuclear Magnetic Resonance (NMR) spectroscopy data, and extensions of **TIMP** for modeling this data are under investigation. Furthermore, options to model data arising in ultra-fast pump-dump-probe spectroscopy experiments are in development.

In collaboration with Joris Snellenburg and Sergey Laptенок, a java-based graphical user interface (GUI) to **TIMP** has been developed, **TIMPGUI**. This provides possibilities for interactive exploration of data and model fit, and is in preparation for public release.

9.2 Samenvatting: Scheidbare niet-lineaire modellen: theorie, implementatie en toepassingen in natuur- en scheikunde

De studies beschreven in dit proefschrift hebben betrekking op scheidbare niet-lineaire modellen die de vorm hebben van een lineaire combinatie van niet-lineaire functies plus een stochastische component samengesteld uit Gaussische witte ruis. In deze modellen kunnen twee soorten parameters onderscheiden worden: intrinsiek niet-lineaire, en conditioneel lineaire. De laatste vormen vaak een grote groep en kunnen automatisch geschat worden met behulp van het variabele projectie algoritme geïntroduceerd door Golub and Pereyra (1972). Variabele projectie wordt beschreven in Hoofdstuk 2 met nadruk op het gebruik voor matrix data. Ook de scheidbare niet-lineaire modellen voor data die bij tijdsopgeloste spectroscopie, microscopie en massaspectrometrie experimenten voorkomen wordt in dit hoofdstuk beschreven. Deze toepassingsgebieden zijn de belangrijkste motivatie voor het onderzoek in dit proefschrift. In deze gebieden is het gebruik van een gemeenschappelijk model voor alle metingen verzameld over meerdere onafhankelijke variabelen, mogelijk zelfs een veelheid aan experimenten, een krachtige methode voor het extraheren van informatie over het te bestuderen systeem. Het gebruik van scheidbare niet-lineaire modellen op deze manier wordt vaak *globale analyse* genoemd.

Hoofdstuk 3 beschrijft de eigenschappen van variabele projectie varianten en het NIPALS algoritme voor “alternating least squares” om het fit-probleem voor een som van exponentiële functies onder een kleinste kwadraten conditie te behandelen. Een nieuwe afleiding van de Fisher informatie matrix binnen het Golub-Pereyra variabele projectie algoritme staat een numerieke vergelijking toe van parameter precisie bij verschillende variabele projectie varianten. Er is geconstateerd dat de analytische Kaufman benadering van de volle variabele projectie functionaal geschikt is voor het som-van-exponenten fit probleem dat typisch is voor problemen die voorkomen bij tijdsopgeloste spectroscopie. Ook is voor dit probleem aangetoond dat de lineaire benadering voor de standaardfout in de parameters onder variabele projectie algoritmen een acceptabele benadering is voor de standaard fout schatting.

Hoofdstuk 4 beschrijft de aanpassing van het variabele projectie algoritme om eisen op te leggen aan de lineaire coëfficiënten in een som van niet-lineaire functies. Zulke eisen zijn vaak nodig als slechts enkele waarden van de lineaire parameters fysisch interpreteerbaar zijn. Bijvoorbeeld van lineaire parameters die fluorescentiespectra of massaspectra beschrijven, wordt geëist dat ze niet negatief zijn, aangezien negatieve waarden voor deze parameters niet

interpreteerbaar zijn. In het geval dat deze eisen worden toegepast, wordt er aangetoond dat de actieve eisen resulteren in een toename van de precisie van de intrinsieke niet-lineaire parameterschatting, vergeleken met de schatting verkregen met klassieke variabele projectie.

Hoofdstuk 5 presenteert een kader voor het fitten van scheidbare niet-lineaire modellen in de fysische en chemische toepassingen in de vorm van een pakket **TIMP** voor de R programmeertaal en omgeving voor computationele statistiek. Het pakket bevat flexibele specificatie van modellen, efficiënte optimalisatie van de vrije parameters, en een validatie van de fit. Het ontwerp geeft de mogelijkheid om gemakkelijk nieuwe modeltypen te definiëren. Implementatie in R bevordert eenvoudige integratie van andere statistische routines en zorgt dat het op alle veelgebruikte besturingssystemen kan draaien. De broncode van R en **TIMP** is gratis verkrijgbaar onder de voorwaarden van de GNU General Public License.

Het gebruik van **TIMP** voor het schatten van vrije parameters van scheidbare niet-lineaire modellen om fluorescence lifetime imaging microscopy (FLIM) te beschrijven wordt onderzocht in Hoofdstuk 6. Met behulp van simulatie en een controle experiment is aangetoond dat deze methode van grote waarde is voor het modelleren van FLIM data. Toepassingen om eiwit-eiwit interacties te beschrijven worden kort beschreven in hoofdstuk 2. Verder zijn er momenteel studies gaande om scheidbare niet-lineaire modellen voor dit soort interacties te gebruiken.

Hoofdstuk 7 en 8 beschrijven het gebruik van scheidbare niet-lineaire modellen voor het oplossen van componenten uit tijdsopgeloste massaspectrometrie data. Deze globale analyse methode is een alternatief voor de niet-parametrische (zelf-modellerende) technieken, die in dit probleemdomen gebruikelijk zijn. De “proof-of-concept” toepassing van globale analyse voor het oplossen van component elutieprofielen in gaschromatografie–massaspectrometrie data wordt beschreven in Hoofdstuk 7. Vervolgens wordt door simulatie studies in Hoofdstuk 8 aangetoond dat globale analyse beter werkt dan de algemeen toegepaste zelf-modellerings methode om de onderliggende massa spectra van data te schatten in gevallen waarbij de elutieprofielen erg overlappen.

Acknowledgements

I would like to thank my advisor Ivo van Stokkum first and foremost, for the possibility to pursue this research, for all of his feedback and guidance along the way, and for the freedom I had, especially with regard to software development. His insight has enhanced every aspect of this thesis. I will always be grateful for the chance to have worked here with him for these four years.

Sergey Laptanok provided my introduction into the world of FLIM, and moreover was always ready to discuss any scientific (or non-scientific) idea. He contributed many improvements to **TIMP**, and a lot of time to designing methods for FLIM data visualization. Via Sergey I came to know others working on FLIM at Wageningen University, including Ton Visser, Herbert van Amerongen, Jan Willem Borst and Bart van Oort, all of whom have been a pleasure to interact with.

I am grateful to the biophysics group, the staff of which includes Jan Dekker, Rienk van Grondelle, Marloes Groot, Sandrine d'Haene, John Kennis, Henny van Roon, Jos Thieme, and Silvia Volker, for including me in the group and in their projects. Cosimo Bonetti, Stephane Calmat, Magdalena Gauden, Alessandro Marin, Elisabet Romero, Mikas Vengris and Luuk van Wilderen all involved me in the analysis of their spectroscopy data. This provided a very interesting window on experimental work. Other PhD students and postdocs from the group that I did not work with directly, but who I would like to thank for creating a good atmosphere and help in various forms include Maxime Alexandre, Rudi Bererra, Toh Kee Chua, Mariangela di Donato, Janne Ihalainen, Cristian Ilioaia, Jason Key, Tjaart Krueger, Natalia Pawlowicz, Alisa Rupenyan, Emmanouil Papagiannakis, Danielis Rutkauskas, Andreas Stahl, Thomas Stuart Cohen, Olga Sytina, Kinga Sznee, and Stefan Witte.

Thomas Cohen Stuart kindly provided the translation of the summary into Dutch.

Joris Snellenburg contributed a lot of effort toward the graphical user interface **TIMPGUI**, especially during the summer of 2008, which included some memorable weeks of programming in collaboration with Sergey Laptanok.

Thomas Binsl and Hans van Beek introduced me to something of metabolomics research, and were excellent to work with.

Velitchka Mihaleva and her colleagues in Wageningen at Plant Research International were very helpful in the development of mass spectroscopy applications. Filipe Lopes and Linda Switzar were also very helpful resources regarding

mass spectrometry.

Sabine Van Huffel and Fetsja Bijma suggested many improvements to the chapters herein, and I am thankful for the time and attention they devoted to this work.

Mathisca de Gunst provided me with a taste of the perspective of her statistics group via an invitation to speak at her seminar, for which I am grateful.

Henri Bal is the promotor of this thesis, and I am very glad for his help and supportive presence.

Jan de Leeuw gave me the chance to contribute to the organization of the special volume of the *Journal of Statistical Software* devoted to “Spectroscopy and Chemometrics in R”. I thank him, his co-editor Achim Zeileis and all of the authors of issues in this volume for the experience.

The research described herein has benefited greatly from the availability of R and the Comprehensive R Archive Network of internet sites, and from interactions with people in the R community via email, the mailing lists and the useR! conferences. I am grateful to everyone that makes up this community for their efforts and cooperative spirit.

Finally, I want to thank my friends and family for their support during this period. Special thanks are due to JEB, MAUT, the 3rd ARM, Happy and my late mother, whose encouragement has been with me ever since I can remember.

Katharine Mullen

Amsterdam, September 2008

Bibliography

- Al-Khayyal FA (1990). “Jointly constrained bilinear programs and related problems: An overview.” *Computers and Mathematics with Applications*, **19**(11), 53–62.
- Alfassi ZB (2004). “On the normalization of a mass spectrum for comparison of two spectra.” *Journal of the American Society for Mass Spectrometry*, **15**, 385–387.
- Ameloot M, Boens N, Andriessen R, van den Bergh V, de Schryver FC (1992). “Global analysis of biochemical and biophysical data.” In L Brand, ML Johnson (eds.), “Methods in Enzymology,” volume 210, pp. 314–340. Academic Press.
- Audet C, Hansen P, Jaumard B, Savard G (2000). “A branch and cut algorithm for non-convex quadratically constrained quadratic programming.” *Mathematical Programming*, **87**(1), 131–152.
- Ausloos P, Clifton C, Lias S, Mikaya A, Stein S, Tchekhovskoi D, Sparkman O, Zaikin V, Zhu D (1999). “The critical evaluation of a comprehensive mass spectral library.” *Journal of the American Society for Mass Spectrometry*, **10**(4), 287–299.
- Bajzer Ž, Zelić A, Prendergast FG (1995). “Analytical Approach to the Recovery of Short Fluorescence Lifetimes from Fluorescence Decay Curves.” *Biophysical Journal*, **69**(3), 1148–1161.
- Barber PR, Ameer-Beg SM, Gilbey JD, Edens RJ, Ezike I, Vojnovic B (2005). “Global and Pixel Kinetic Data Analysis for FRET Detection by Multi-photon Time-domain FLIM.” In A Periasamy, PT So (eds.), “Multiphoton Microscopy in the Biomedical Sciences V, Proceedings of the SPIE,” volume 5700, pp. 171–181.
- Basu S, Bresler Y (2000). “Stability of Nonlinear Least Squares Problems and the Cramer-Rao Bound.” *IEEE Transactions on Signal Processing*, **48**(12), 3426–3436.
- Bates DM, DebRoy S (2003). “Converting a Large R Package to S4 Classes and Methods.” In K Hornik, F Leisch, A Zeileis (eds.), “Proceedings of the 3rd

- International Workshop on Distributed Statistical Computing, Vienna, Austria,” ISSN 1609-395X, URL <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/>.
- Bates DM, Lindstrom MJ (1986). “Nonlinear least squares with conditionally linear parameters.” In “Proceedings of the Statistical Computing Section,” pp. 152–157. American Statistical Association, New York.
- Bates DM, Watts DG (1988). *Nonlinear regression analysis and its applications*. John Wiley & Sons, New York.
- Becker W, Bergmann A (2003). “Lifetime Imaging Techniques for Optical Microscopy.” *Technical report*, Becker & Hickl GmbH. URL www.becker-hickl.de/pdf/tcvgbh1.pdf.
- Becker W, Bergmann A, Biskup C, Zimmer T, Kloecker N, Benndorf K (2002). “Multiwavelength TCSPC Lifetime Imaging.” In A Periasamy, PTC So (eds.), “Multiphoton Microscopy in the Biomedical Sciences II, Proceedings of the SPIE,” volume 4620, pp. 79–84.
- Becker W, Bergmann A, Wabnitz H, Grosenick D, Liebert A (2001). “High-count-rate Multichannel TCSPC for Optical Tomography.” In S Andersson-Engels, MF Kaschke (eds.), “Photon Migration, Optical Coherence Tomography, and Microscopy, Proceedings of the SPIE,” volume 4431, pp. 249–254.
- Beechem J (1989). “A second generation global analysis program for the recovery of complex inhomogeneous fluorescence decay kinetics.” *Chemistry and Physics of Lipids*, **50**, 237–51.
- Beechem J (1992). “Global analysis of biochemical and biophysical data.” In L Brand, ML Johnson (eds.), “Methods in Enzymology,” volume 210, pp. 37–54. Academic Press.
- Beechem J, Ameloot M, Brand L (1985). “Global and target analysis of complex decay phenomena.” *Analytical Instrumentation*, **14**, 379–402.
- Benthem MHV, Keenan MR (2004). “Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems.” *Journal of Chemometrics*, **18**(10), 441–450.
- Bijlsma S, Boelens HFM, Hoefsloot HCJ, Smilde AK (2000). “Estimating reaction rate constants: comparison between traditional curve fitting and curve resolution.” *Analytica Chimica Acta*, **419**, 197–2007.
- Bijlsma S, Boelens HFM, Hoefsloot HCJ, Smilde AK (2002). “Constrained least squares methods for estimating reaction rate constants from spectroscopic data.” *Journal of Chemometrics*, **16**(1), 28–40.
- Black PE (2004). “Dictionary of Algorithms and Data Structures.” *Technical report*, U.S. National Institute of Standards and Technology.

- Böckmann C (1995). "A modification of the trust-region Gauss-Newton method to solve separable nonlinear least squares problems." *Journal of Mathematical Systems, Estimation and Control*, **5**, 111–115.
- Borst J, Hink MA, van Hoek A, Visser AJWG (2003). "Multiphoton Microspectroscopy in Living Plant Cells." In A Periasamy, PTC So (eds.), "Multiphoton Microscopy in the Biomedical Sciences III, Proceedings of the SPIE," volume 4963, pp. 231–238.
- Borst JW, Hink MA, van Hoek A, Visser AJWG (2005). "Effects of Refractive Index and Viscosity on Fluorescence and Anisotropy Decays of Enhanced Cyan and Yellow Fluorescent Proteins." *Journal of Fluorescence*, **15**(2), 153–160.
- Brimberg J, Hansen P, Mladenović N (2002). "A note on reduction of quadratic and bilinear programs with equality constraints." *Journal of Global Optimization*, **22**(1-4), 39–47.
- Bro R, Jong SD (1997). "A fast non-negativity-constrained least squares algorithm." *Journal of Chemometrics*, **11**(5), 393–401.
- Bro R, Sidiropoulos ND (1998). "Least squares algorithms under unimodality and non-negativity constraints." *Journal of Chemometrics*, **12**, 223–247.
- Browne SV, Dongarra JJ, Green SC, Moore K, Rowan TH, Wade RC (1994). "Netlib Services and Resources." *Technical Report CS-94-222*, University of Tennessee.
- Buttery R, Takeoka G (2004). "Some Unusual Minor Volatile Components of Tomato." *Journal of Agricultural and Food Chemistry*, **52**(20), 6264–6266.
- Byrd RH, Lu P, Nocedal J, Zhu CY (1995). "A Limited Memory Algorithm for Bound Constrained Optimization." *SIAM Journal on Scientific Computing*, **16**(6), 1190–1208.
- Cantor CR, Schimmel PR (1980). *Biophysical Chemistry, Part II: Techniques for the Study of Biological Structure and Function*. W. H. Freeman.
- Chambers JM (1998). *Programming with Data, A Guide to the S Language*. Springer-Verlag, New York.
- de Hoffmann E, Stroobant V (2007). *Mass Spectrometry: Principles and Applications*. Wiley, 3rd edition.
- de Juan A, Maeder M, Martínez M, Tauler R (2000). "Combining hard- and soft-modelling to solve kinetic problems." *Chemometrics and Intelligent Laboratory Systems*, **54**, 123–141.

- de Juan A, Maeder M, Martínez M, Tauler R (2001). "Application of a novel resolution approach combining soft- and hard-modelling features to investigate temperature-dependent kinetic processes." *Analytica Chimica Acta*, **442**, 337–350.
- de Juan A, Tauler R (2003). "Chemometrics applied to unravel multicomponent processes and mixtures; Revisiting latest trends in multivariate resolution." *Analytica Chimica Acta*, **500**, 195–210.
- Dioumaev AK (1997). "Evaluation of intrinsic chemical kinetics and transient product spectra from time-resolved spectroscopic data." *Biophysical Chemistry*, **67**(1), 1–25.
- Elzhov TV, Mullen KM (2008). **minpack.lm**: *R interface to the Levenberg-Marquardt nonlinear least squares algorithm found in MINPACK*. R package version 1.1-1.
- Fisz J (2006). "Combined Genetic Algorithm and Multiple Linear Regression (GA-MLR) Optimizer: Application to Multi-exponential Fluorescence Decay Surface." *Journal of Physical Chemistry A*, **110**(48), 12977–12985.
- Garrido M, Larrechi M, Rius F, Tauler R (2005). "Calculation of band boundaries of feasible solutions obtained by Multivariate Curve Resolution-Alternating Least Squares of multiple runs of a reaction monitored by NIR spectroscopy." *Chemometrics and Intelligent Laboratory Systems*, **76**, 111–120.
- Garrido M, Rius FX, Larrechi MS (2008). "Multivariate curve resolution-alternating least squares (MCR-ALS) applied to spectroscopic data from monitoring chemical reactions processes." *Analytical and Bioanalytical Chemistry*, **390**, 2059–2066.
- Gay D, Kaufman L (1991). "Tradeoffs in Algorithms for Separable and Block Separable Nonlinear Least Squares." In R Vichnevetsky, JJH Miller (eds.), "IMACS '91, Proceedings of the 13th World Congress on Computational and Applied Mathematics," pp. 157–158. Criterion Press, Dublin.
- Gobets B, van Stokkum IHM, Rogner M, Kruij J, Schlodder E, Karapetyan NV, Dekker JP, van Grondelle R (2001). "Time-Resolved Fluorescence Emission Measurements of Photosystem I Particles of Various Cyanobacteria: A Unified Compartmental Model." *Biophysical Journal*, **81**(1), 407–424.
- Godfrey K (1983). *Compartmental Models and Their Application*. Academic Press, London.
- Golub GH, LeVeque RJ (1979). "Extensions and Uses of the Variable Projection Algorithm for Solving Nonlinear Least Squares Problems." In "Proceedings of the 1979 Army Numerical Analysis and Computers Conference," pp. 1–12.

- Golub GH, Pereyra V (1972). "The Differentiation of Pseudo-inverses and Non-linear Least Squares Problems whose Variables Separate." *Technical report*, Stanford University, Department of Computer Science.
- Golub GH, Pereyra V (1973). "The Differentiation of Pseudoinverses and Non-linear Least Squares Problems whose Variables Separate." *SIAM Journal on Numerical Analysis*, **10**, 413–432.
- Golub GH, Pereyra V (2003). "Separable Nonlinear Least Squares: the Variable Projection Method and its Applications." *Inverse Problems*, **19**, R1–R26.
- Golub GH, van Loan CF (1996). *Matrix Computations* (3rd edn). The Johns Hopkins University Press, Baltimore, MD, USA.
- Grailhe R, Merola F, Ridard J, Couvignou S, Poupon CL, Changeux J, Laguitton-Pasquier H (2006). "Monitoring Protein Interactions in the Living Cell Through the Fluorescence Decays of the Cyan Fluorescent Protein." *ChemPhysChem*, **7**(7), 1442–1454.
- Gratton E, Breusegem S, Sutin J, Ruan Q, Barry N (2003). "Fluorescence Lifetime Imaging for the Two-photon Microscope: Time-domain and Frequency-Domain Methods." *Journal of Biomedical Optics*, **8**(3), 381–390.
- Grinvald A, Steinberg IZ (1974). "On the Analysis of Fluorescence Decay Kinetics by the Method of Least-squares." *Analytical Biochemistry*, **59**(2), 583–598.
- Henry ER (1997). "The Use of Matrix Methods in the Modeling of Spectroscopic Data Sets." *Biophysical Journal*, **72**(2), 652–673.
- Holzwarth AR (1996). "Data analysis of time-resolved measurements." In J Amesz, A Hoff (eds.), "Biophysical Techniques in Photosynthesis," volume I, chapter 5, pp. 75–92. Kluwer Academic Publishers.
- Horst R, Tuy H (1996). *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin, 3rd edition.
- Istratov AA, Vyvenko OF (1999). "Exponential Analysis in Physical Phenomena." *Review of Scientific Instruments*, **70**(2), 1233–1257.
- Jandanklang P, Maeder M, Whitson AC (2001). "Target transform fitting: a new method for the non-linear fitting of multivariate data with separable parameters." *Journal of Chemometrics*, **15**(6), 886–9383.
- Jiang J, Liang Y, Ozaki Y (2004). "Principles and methodologies in self-modeling curve resolution." *Chemometrics and Intelligent Laboratory Systems*, **71**, 1–12.
- Jonsson P, Johansson A, Gullberg J, Trygg J, A J, Grung B, Marklund S, Sjöström M, Antti H, Moritz T (2005). "High-Throughput Data Analysis for Detecting and Identifying Differences between Samples in GC/MS-Based Metabolomic Analyses." *Analytical Chemistry*, **77**(17), 5635–5642.

- Kaufman L (1975). "A variable projection method for solving separable nonlinear least squares problems." *BIT*, **15**, 49–57.
- Kaufman L, Pereyra V (1978). "A method for separable nonlinear least squares problems with separable nonlinear equality constraints." *SIAM Journal of Numerical Analysis*, **15**(1), 12–20.
- Kopka J, Schauer N, Krueger S, Birkemeyer C, Usadel B, Bergmuller E, Dormann P, Weckwerth W, Gibon Y, Stitt M, Willmitzer L, Fernie AR, Steinhauser D (2005). "GMD@CSB.DB: the Golm Metabolome Database." *Bioinformatics*, **21**(8), 1635–1638.
- Krogh FT (1974). "Efficient implementation of a variable projection algorithm for nonlinear least squares problems." *Communications for the Association for Computing Machinery*, **17**(3), 167–169.
- Laptenok S, Mullen KM, Borst JW, van Stokkum IHM, Apanasovich VV, Visser AJWG (2007). "Fluorescence Lifetime Imaging Microscopy (FLIM) data analysis with **TIMP**." *Journal of Statistical Software*, **18**(8), 1–20. URL <http://www.jstatsoft.org/v18/i08/>.
- Last RL, Jones AD, Shachar-Hill Y (2007). "Towards the plant metabolome and beyond." *Nature Reviews Molecular Cell Biology*, **8**, 167–174.
- Lawson CL, Hanson RJ (1974). *Solving Least Squares Problems*. Prentice Hall, Englewood Cliffs, NJ.
- Lawson CL, Hanson RJ (1995). *Solving Least Squares Problems*. Classics in Applied Mathematics. SIAM, Philadelphia.
- Lawton WH, Sylvestre EA (1971). "Elimination of Linear Parameters in Nonlinear Regression." *Technometrics*, **13**, 461–467.
- Lukeman GG (1999). *Application of the Shen-Ypma algorithm for Separable Overdetermined Nonlinear Systems*. Master's thesis, Department of Mathematics and Statistics, Dalhousie University, Canada.
- Maeder M (1987). "Evolving factor analysis for the resolution of overlapping chromatographic peaks." *Analytical Chemistry*, **59**(3), 527–530.
- Marco VBD, Bombi GG (2001). "Mathematical functions for the representation of chromatographic peaks." *Journal of Chromatography A*, **931**, 1–30.
- Mas S, Fonrodona G, Tauler R, Barbosa J (2007). "Determination of phenolic acids in strawberry samples by means of fast liquid chromatography and multivariate curve resolution methods." *Talanta*, **71**, 1455–1463.
- Maus M, Cotlet M, Hofkens J, Gensch T, de Schryver FC, Schaffer J, Seidel CAM (2001). "An Experimental Comparison of the Maximum Likelihood Estimation and Nonlinear Least-Squares Fluorescence Lifetime Analysis of Single Molecules." *Analytical Chemistry*, **73**(9), 2078–2086.

- Meyer D, Zeileis A, Hornik K (2006). **vcd**: *Visualizing Categorical Data*. R package version 1.0-2.
- Mihaleva VV, Verhoeven HA, de Vos RC, Hall RD, van Ham RC (2008). “Automated procedure for candidate compound selection in GC-MS metabolomics based on prediction of Kovats retention index.” *Submitted*.
- Moré JJ (1978). “The Levenberg-Marquardt Algorithm, Implementation and Theory.” In GA Watson (ed.), “Numerical Analysis,” volume 630 of *Lecture Notes in Mathematics*, pp. 105–116. Springer-Verlag.
- Moshier SL (1992). *Cephes Mathematical Library*. URL <http://www.moshier.net/>.
- Mullen KM, van Stokkum IHM (2007a). **npls**: *The Lawson-Hanson algorithm for non-negative least squares (NNLS)*. R package version 1.2.
- Mullen KM, van Stokkum IHM (2007b). “**TIMP**: an R package for modeling multi-way spectroscopic measurements.” *Journal of Statistical Software*, **18**(3), 1–46. URL <http://www.jstatsoft.org/v18/i03/>.
- Mullen KM, van Stokkum IHM (2008a). **ALS**: *multivariate curve resolution alternating least squares (MCR-ALS)*. R package version 0.0.2.
- Mullen KM, van Stokkum IHM (2008b). **TIMP**: *a problem solving environment for fitting separable nonlinear models in physics and chemistry applications*. R package version 1.7, URL <http://timp.r-forge.r-project.org/>.
- Mullen KM, van Stokkum IHM (2008c). “Separable nonlinear least squares with constraints on the conditionally linear parameters.” *submitted*.
- Mullen KM, van Stokkum IHM (2008d). “The variable projection algorithm in time-resolved spectroscopy, microscopy and mass spectrometry applications.” *Numerical Algorithms, in press*.
- Mullen KM, Vengris M, van Stokkum IHM (2007). “Algorithms for Separable Nonlinear Least Squares with Application to Modelling Time-resolved Spectra.” *Journal of Global Optimization*, **38**(2), 201–213.
- Nagle JF (1991a). “Photocycle kinetics: analysis of Raman data from bacteriorhodopsin.” *Photochemistry and Photobiology*, **54**, 897–903.
- Nagle JF (1991b). “Solving Complex Photocycle Kinetics - Theory and Direct Method.” *Biophysical Journal*, **59**, 476–487.
- Nagle JF, Parodi LA, Lozier RH (1982). “Procedure for testing kinetic models of the photocycle of bacteriorhodopsin.” *Biophysical Journal*, **38**(2), 161–174.
- Nagle JF, Zimanyi L, Lanyi JK (1995). “Testing BR photocycle kinetics.” *Biophysical Journal*, **68**(4), 1490–1499.

- Osborne MR (2007). “Separable least squares, variable projection, and the Gauss-Newton algorithm.” *Electronic Transactions on Numerical Analysis*, **28**, 1–15.
- Osborne MR, Smyth GK (1995). “A modified Prony algorithm for exponential function fitting.” *SIAM Journal on Scientific Computing*, **16**(1), 119–138.
- Pandit A, Visschers RW, van Stokkum IHM, Kraayenhof R, van Grondelle R (2001). “Oligomerization of Light-Harvesting I Antenna Peptides of *Rhodospirillum rubrum*.” *Biochemistry*, **40**, 12913–12924.
- Parks TA (1985). *Reducible Nonlinear Programming Problems*. Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, USA.
- Pelet S, Previte MJR, Laiho LH, So PTC (2000). “A Fast Global Fitting Algorithm for Fluorescence Lifetime Imaging Microscopy Based on Image Segmentation.” *Biophysical Journal*, **87**(4), 2807–2817.
- Peré-Trepat E, Lacorte S, Tauler R (2005). “Solving liquid chromatography mass spectrometry coelution problems in the analysis of environmental samples by multivariate curve resolution.” *Journal of Chromatography A*, **1096**, 111–122.
- Peré-Trepat E, Lacorte S, Tauler R (2007). “Alternative calibration approaches for LC–MS quantitative determination of coeluted compounds in complex environmental mixtures using multivariate curve resolution.” *Analytica Chimica Acta*, **595**, 228–237.
- Peter M, Ameer-Beg SM, Hughes MKY, Keppler MD, Prag S, Marsh M, Vojnovic B, Ng T (2005). “Multiphoton-FLIM Quantification of the EGFP-mRFP1 FRET Pair for Localization of Membrane Receptor-Kinase Interactions.” *Biophysical Journal*, **88**(2), 1224–1237.
- Petersson J, Holmström K (1997). “A review of the parameter estimation problem of fitting positive exponential sums to empirical data.” *Technical Report Technical Report IMA-TOM-1997-08*, Department of Mathematics and Physics, Mälardalen University, Sweden.
- Petersson J, Holmström K (1998). “Initial values for the exponential sum least squares fitting problem.” *Technical Report Technical Report IMA-TOM-1998-01*, Department of Mathematics and Physics, Mälardalen University, Sweden.
- Petro-Turza M (1987). “Flavour of tomato and tomato products.” *Food Reviews International*, **2**(3), 309–351.
- Pribić R (1996). *Parameter estimation in mathematical models for spectroscopic observations*. Ph.D. thesis, Department of Physics and Astronomy, Vrije Universiteit Amsterdam, The Netherlands.

- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Ruhe A, Wedin PA (1980). "Algorithms for Separable Nonlinear Least Squares Problems." *SIAM Review*, **22**(3), 318–337.
- Russinova E, Borst J, Kwaaitaal M, Cano-Delgado A, Yin Y, Chory J, de Vries SC (2004). "Heterodimerization and Endocytosis of Arabidopsis Brassinosteroid Receptors BRI1 and AtSERK3 (BAK1)." *Plant Cell*, **16**(12), 3216–3229.
- Rust BW, Leventhal M, McCall SL (1976). "Evidence for a radioactive decay hypothesis for supernova luminosity." *Nature*, **262**, 118–120.
- Ryan D, Robards K (2006). "Metabolomics: The Greatest Omics of Them All?" *Analytical Chemistry*, **78**(23), 7954–7958.
- Seber GAF, Wild CJ (2003). *Nonlinear regression*. John Wiley & Sons, New Jersey.
- Shen Y, Ypma TJ (2007). "Solving rank-deficient separable nonlinear equations." *Applied Numerical Mathematics*, **57**(5-7), 609–615.
- Sima DM, Van Huffel S (2006). "Regularized semiparametric model identification with application to nuclear magnetic resonance signal quantification with unknown macromolecular base-line." *Journal of the Royal Statistical Society: Series B*, **68**(3), 383–409.
- Sima DM, Van Huffel S (2007). "Separable nonlinear least squares fitting with linear bound constraints and its application in magnetic resonance spectroscopy data quantification." *Journal of Computational and Applied Mathematics*, **203**, 264–278.
- Sjöberg J, Viberg M (1997). "Separable non-linear least-squares minimization-possible improvements for neural net fitting." In "Proceedings of the IEEE Workshop Neural Networks for Signal Processing VII (Amelia Island, FL, USA)," pp. 345–354.
- Smyth GK (1996). "Partitioned algorithms for maximum likelihood and other nonlinear estimation." *Statistics and Computing*, **6**, 201–216.
- Soo Y, Bates DM (1992). "Loosely coupled nonlinear least squares." *Computational Statistics & Data Analysis*, **14**(2), 249–259.
- Stark PB, Parker RL (1995). "Bounded-variable least-squares: an algorithm and applications." *Computational Statistics*, **10**, 129–141.
- Stein SE (1999). "An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data." *Journal of the American Society for Mass Spectrometry*, **10**(8), 770–781.

- Stein SE, Scott DR (1994). "Optimization and testing of mass spectral library search algorithms for compound identification." *Journal of the American Society for Mass Spectrometry*, **5**, 859–866.
- Suhling K, French PMW, Phillips D (2005). "Time-resolved Fluorescence Microscopy." *Photochemical and Photobiological Sciences*, **4**, 13–22.
- Tauler R (1995). "Multivariate curve resolution applied to second order data." *Chemometrics and Intelligent Laboratory Systems*, **30**, 133–146.
- Tauler R (2001). "Calculation of maximum and minimum band boundaries of feasible solutions for species profiles obtained by multivariate curve resolution." *Journal of Chemometrics*, **15**, 627–646.
- Tauler R, Smilde A, Kowalski B (1995). "Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution." *Journal of Chemometrics*, **9**, 31–58.
- Tikunov Y, Lommen A, de Vos CR, Verhoeven HA, Bino RJ, Hall RD, Bovy AG (2005). "A Novel Approach for Nontargeted Data Analysis for Metabolomics. Large-Scale Profiling of Tomato Fruit Volatiles." *Plant Physiology*, **139**(3), 1125–1137.
- Tsien RY (1998). "The Green Fluorescent Protein." *Annual Review of Biochemistry*, **67**, 509–544.
- Tukey JW (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, Massachusetts.
- Turner R (2005). **Iso**: *Functions to perform isotonic regression*. R package version 0.0-2.
- Turner TR, Wollan PC (1997). "Locating a maximum using isotonic regression." *Computational Statistics & Data Analysis*, **25**(3), 305–320.
- van den Bos A (2007). *Parameter estimation for scientists and engineers*. Wiley-Interscience, Hoboken, NJ.
- van den Bos A, Swarte JH (1993). "Resolvability of the parameters of multiexponentials and other sum models." *IEEE Transactions on Signal Processing*, **41**(1), 313–322.
- van der Veen JWC, de Beer R, Luyten PR, van Ormondt D (1988). "Accurate Quantification of in Vivo 31P NMR Signals Using the Variable Projection Method and Prior Knowledge." *Magnetic Resonance in Medicine*, **6**, 92–98.
- van Stokkum IHM (1997). "Parameter Precision in Global Analysis of Time-Resolved Spectra." *IEEE Transactions on Instrumentation and Measurement*, **46**(4), 764–768.

- van Stokkum IHM (2005). “Global and Target Analysis of Time-resolved Spectra, Lecture notes for the Troisième Cycle de la Physique en Suisse Romande.” *Technical report*, Department of Physics and Astronomy, Faculty of Sciences, Vrije Universiteit, Amsterdam, The Netherlands. URL <http://www.nat.vu.nl/~ivo/lecturenotes.pdf>.
- van Stokkum IHM, Bal HE (2006). “A Problem Solving Environment for interactive modelling of multiway data.” *Concurrency and computation: practice and experience*, **18**, 263–269.
- van Stokkum IHM, Brouwer AM, van Ramesdonk H, Scherer T (1993). “Multiresponse parameter estimation and compartmental analysis of time resolved fluorescence spectra: Application to conformational dynamics of charge-separated species in solution.” *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen*, **96**, 43–68.
- van Stokkum IHM, Gobets B, Gensch T, van Mourik F, Hellingwerf KJ, van Grondelle R, Kennis JTM (2006). “(Sub)-Picosecond Spectral Evolution of Fluorescence in Photoactive Proteins Studied with a Synchroscan Streak Camera System.” *Photochemistry and Photobiology*, **82**, 380–388.
- van Stokkum IHM, Larsen DS, van Grondelle R (2004). “Global and Target Analysis of Time-resolved Spectra.” *Biochimica et Biophysica Acta*, **1657**, 82–104, and erratum, 1658, 262.
- van Stokkum IHM, Linsdell H, Hadden JM, Haris PI, Chapman D, Bloemendal M (1995). “Temperature-Induced Changes in Protein Structures Studied by Fourier Transform Infrared Spectroscopy and Global Analysis.” *Biochemistry*, **34**, 10508–10518.
- van Stokkum IHM, Lozier RH (2002). “Target Analysis of the Bacteriorhodopsin Photocycle Using a Spectrotemporal Model.” *Journal of Physical Chemistry B*, **106**(13), 3477–3485.
- van Stokkum IHM, Mullen KM, Mihaleva VV (2008a). “Global analysis of multiple gas chromatography-mass spectrometry (GC/MS) data sets: a method for resolution of co-eluting components with comparison to MCR-ALS.” *Chemometrics and Intelligent Laboratory Systems*, *in press*.
- van Stokkum IHM, van Oort B, van Mourik F, Gobets B, van Amerongen H (2008b). “(Sub)-Picosecond Spectral Evolution of Fluorescence Studied with a Synchroscan Streak-Camera System and Target Analysis.” In TJ Aartsma, J Matysik (eds.), “Biophysical Techniques in Photosynthesis, Volume II, Series Advances in Photosynthesis and Respiration,” volume 26, pp. 223–240. Springer, Dordrecht.
- Varah J (1985). “On Fitting Exponentials by Nonlinear Least Squares.” *SIAM Journal on Scientific and Statistical Computing*, **1**, 30–44.

- Verveer PJ, Squire A, Bastiaens PIH (2000). "Global Analysis of Fluorescence Lifetime Imaging Microscopy Data." *Biophysical Journal*, **78**(4), 2127–2137.
- Windig W, Guilment J (1991). "Interactive self-modeling mixture analysis." *Analytical Chemistry*, **63**, 1425–1432.
- Wohlleben W, Buckup T, Herek JL, Cogdell RJ, Motzkus M (2003). "Multichannel Carotenoid Deactivation in Photosynthetic Light Harvesting as Identified by an Evolutionary Target Analysis." *Biophysical Journal*, **85**, 442–450.
- Wold H, Lyttkens E (1969). "Nonlinear iterative partial least squares (NIPALS) estimation procedures." *Bulletin of the International Statistical Institute*, **43**, 29–51.
- Wold S, Esbensen K, Geladi P (1987). "Principal component analysis." *Chemometrics and Intelligent Laboratory Systems*, **2**, 37–52.