



**KATHOLIEKE UNIVERSITEIT LEUVEN**  
FACULTEIT INGENIEURSWETENSCHAPPEN  
DEPARTEMENT ELEKTROTECHNIEK  
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

# **REGULARIZATION TECHNIQUES IN MODEL FITTING AND PARAMETER ESTIMATION**

Promotor:  
Prof. dr. ir. S. Van Huffel

Proefschrift voorgedragen tot  
het behalen van het doctoraat  
in de ingenieurswetenschappen

door

**Diana Maria SIMA**

April 2006





KATHOLIEKE UNIVERSITEIT LEUVEN  
FACULTEIT INGENIEURSWETENSCHAPPEN  
DEPARTEMENT ELEKTROTECHNIEK  
Kasteelpark Arenberg 10, 3001 Leuven (Heverlee)

## REGULARIZATION TECHNIQUES IN MODEL FITTING AND PARAMETER ESTIMATION

Jury:

Prof. dr. ir. G. De Roeck, voorzitter  
Prof. dr. ir. S. Van Huffel, promotor  
Prof. dr. ir. P. Van Dooren (UCL)  
Prof. dr. ir. B. De Moor  
Prof. dr. ir. M. Van Barel  
Prof. dr. ir. R. Pintelon (VUB)  
Prof. dr. Z. Strakoš (Czech Acad. Sci.)

Proefschrift voorgedragen tot  
het behalen van het doctoraat  
in de ingenieurswetenschappen  
door

**Diana Maria SIMA**

© Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen

Arenbergkasteel, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2006/7515/25

ISBN 90-5682-691-3

*In the memory of my grandfathers*

Vasile Sima,  
*medical doctor and poet*

Mircea Ulubeanu,  
*civil engineer and inventor*



# Foreword

It is a great pleasure to express my warmest thanks to the people that supported and helped me during the PhD years.

My promoter, Professor Sabine Van Huffel, has constantly been involved in all the aspects that made this work possible. Her concrete help and guidance, as well as her example as an interdisciplinary researcher, inspired my own interests and shaped my scientific development. I thank her!

I thank Professor Paul Van Dooren (Catholic University of Louvain), Professor Marc Van Barel (KUL) and Professor Bart De Moor (KUL) for serving as my advisors, and Professor Zdeněk Strakoš (Czech Academy of Science) and Professor Rik Pintelon (Vrije Universiteit Brussel) for serving as members of the exam committee.

My thanks go to Professor Gene H. Golub at the SCCM division of the Computer Science Department at Stanford University, who in the very beginning became interested in my work and progress, offered good feedback and encouragements, as well as new perspectives during my visit to Stanford in February 2003.

The SISTA group at ESAT, K.U. Leuven, provided an excellent working environment. In particular, I thank all the current and former members of the BioMed subgroup not only for having them around at meetings and seminars, but also for the relaxing informal occasions that we had. My office mates Mieke Schuermans, Ivan Markovsky, and, more recently, Mariya Ishteva, deserve very special thanks for creating a wonderful office atmosphere and for our countless discussions on the most various topics.

My beloved parents, grandparents and my brother, as well as all my dear uncles, aunts, cousins, and friends, helped me all through the years with advice and encouragements. I have always looked forward to the nice holidays in Romania with them.

The constant love of my boyfriend gave me the most concrete support during all these years. It weponed me with confidence and helped me to create a happy home in Leuven, together with him.

With gratitude, I would like to mention the funding organizations that contributed during my PhD years. The work was sponsored by the doctoral scholarships of the Research Council of K.U. Leuven for Central and Eastern European students (OE/03/23, OE/04/40, OE/05/26). In addition, the financial support of the Belgian Programme on Interuniversity Attraction Poles IUAP V-22 (2002-2006), the FWO projects G.0078.01 (structured matrices), G.0270.02 (nonlinear  $L_p$  approximation), research communities (ICCoS, ANMMM), and the EU projects BIOPATTERN (FP6-2002-IST 508803), ETUMOUR (FP6-2002-LIFESCIHEALTH 503094) is greatly appreciated.



# Abstract

We consider fitting data by linear and nonlinear models. The specific problems that we aim at, although they encompass classic formulations, have as common ground the fact that we attack a special situation: the ill-posed problems.

In the linear case, we consider the total least squares problem. There exist special methods to approach the so-called nongeneric cases, but we propose extensions for the more commonly encountered close-to-nongeneric problems. Several methods of introducing regularization in the context of total least squares are analyzed. They are based on truncation methods or on penalty optimization. The obtained problems might not have closed form solutions. We discuss numerical linear algebra and local optimization methods.

Data fitting by nonlinear or nonparametric models is the second subject of the thesis. We extend the nonlinear regression theory to the case when we have to deal with supplementary regularization constraints, and to a semiparametric context, where only part of the model is known and we have to take into account a component with unknown formulation. We apply the developed theory to the biomedical application of quantifying metabolite concentrations in the human brain from nuclear magnetic resonance spectroscopic signals.

x

---

# Notation

## Sets of numbers

$\mathbb{R}$	the set of real numbers
$\mathbb{C}$	the set of complex numbers
$\mathbb{N}$	the set of natural numbers $\{1, 2, \dots\}$

## Matrix operations

$A^\top$	transpose of a matrix
$A^{-1}$	inverse
$A^\dagger$	pseudoinverse
$\text{diag}(v), v \in \mathbb{R}^n$	the diagonal matrix $\text{diag}(v_1, \dots, v_n)$
$\text{Tr} A$	trace of the matrix $A, \sum a_{ii}$
$\otimes$	Kronecker product $A \otimes B := [a_{ij}B]$
$\odot$	element-wise (Hadamard) product $A \odot B := [a_{ij}b_{ij}]$

## Norms and extreme eigenvalues / singular values

$\ x\ $ or $\ x\ _2, x \in \mathbb{R}^n$	2-norm of a vector $\sqrt{\sum_{i=1}^n x_i^2}$
$\ A\ , A \in \mathbb{R}^{m \times n}$	induced 2-norm $\min_{\ x\ =1} \ Ax\ $
$\ A\ _F, A \in \mathbb{R}^{m \times n}$	Frobenius norm $\sqrt{\text{Tr}(AA^\top)}$
$\lambda_{\min}(A), \lambda_{\max}(A)$	minimum, maximum eigenvalue of a symmetric matrix $A$
$\sigma_{\min}(A), \sigma_{\max}(A)$	minimum, maximum singular value of a matrix $A$

## Probability and statistics

$\mathcal{E}$	expectation operator
$\varepsilon \sim \mathcal{N}(\mu, Q)$	the vector $\varepsilon$ is normally distributed with mean $\mu$ and covariance $Q$

## Miscellaneous symbols

$\mathcal{L}$	loss function that measures the error between a model and a data set
$\hat{\theta}$	an estimate computed from available data for an unknown model parameter $\theta$
$\frac{\partial \mathbf{F}}{\partial x}$	partial derivative
$\nabla \mathbf{F}$	gradient or Jacobian of a multivariable (vectorial) function $\mathbf{F}$
$\nabla^2 F$	Hessian of a multivariable scalar function $F$

**Abbreviations**

AIC	Akaike information criterion
AQSES	accurate quantification of short echo-time MRS signals
CV	cross validation
EIV	errors-in-variables
GCV	generalized cross validation
GIC	generalized information criterion
GSVD	generalized singular value decomposition
GUI	graphical user interface
LS	least squares
MRS	magnetic resonance spectroscopy
NLS	nonlinear least squares
NMR	nuclear magnetic resonance
QEP	quadratic eigenvalue problem
RHS	right-hand side
RLS	regularized least squares
RTLS	regularized total least squares
ScTLS	scaled total least squares
SVD	singular value decomposition
VARPRO	variable projection method
TLS	total least squares
TSVD	truncated singular value decomposition
TTLS	truncated total least squares

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Estimation problems . . . . .	1
1.1.1	Estimation in linear problems . . . . .	2
1.1.2	Estimation in nonlinear problems . . . . .	3
1.1.3	When ill-posed problems arise . . . . .	3
1.2	Regularization . . . . .	5
1.2.1	Goals of regularization . . . . .	5
1.2.2	General penalty-type regularization . . . . .	7
1.2.3	Truncation methods . . . . .	8
1.2.4	Deterministic and statistical regularization . . . . .	8
1.2.5	Newer trends in regularization methods . . . . .	9
1.3	Model selection techniques . . . . .	10
1.3.1	The discrepancy principle . . . . .	10
1.3.2	The L-curve . . . . .	11
1.3.3	Cross validation and generalized cross validation . . . . .	12
1.3.4	Information criteria . . . . .	13
1.3.5	Other model selection criteria . . . . .	14
1.4	Contributions in the thesis . . . . .	15
1.5	Chapter-by-chapter overview . . . . .	16
<b>I</b>	<b>Regularization for linear problems</b>	<b>21</b>
<b>2</b>	<b>Truncation methods for core linear systems</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Truncation methods for linear ill-posed problems . . . . .	24
2.2.1	Linear ill-posed problems . . . . .	24
2.2.2	Truncation methods for linear estimation . . . . .	25
2.3	Core reduction with embedded truncation . . . . .	27
2.3.1	The scaled total least squares formulation . . . . .	27
2.3.2	Short description of the core problem within $Ax \approx b$ . . . . .	29
2.3.3	Extension to multiple right-hand sides and to the nullspace formulation . . . . .	31

	2.3.4	Computing optimal solutions and corrections from a core problem . . . . .	39
	2.4	Implementation and numerical examples . . . . .	39
	2.5	Conclusions . . . . .	42
<b>3</b>		<b>Regularized total least squares</b>	<b>43</b>
	3.1	Introduction . . . . .	43
	3.2	Quadratically constrained problem formulations . . . . .	44
	3.2.1	RTLS method of Golub, Hansen and O'Leary: two parameters formulation . . . . .	45
	3.2.2	RTLS method of Sima, Van Huffel and Golub: iterative update of the solution vector, using quadratic eigenvalue problems . . . . .	46
	3.2.3	RTLS method of Renaut and Guo: alternating iteration on a scalar and the solution vector . . . . .	52
	3.2.4	RTLS method of Beck, Ben-Tal and Teboulle: one scalar optimization . . . . .	53
	3.3	Quadratic penalty formulations . . . . .	55
	3.4	Numerical results . . . . .	56
	3.4.1	Test problems description . . . . .	56
	3.4.2	Comparison between regularization solvers . . . . .	57
	3.4.3	Comparison with newer RTLS methods . . . . .	60
	3.4.4	Comparison with optimization solvers . . . . .	61
	3.4.5	Importance of the starting vector . . . . .	62
	3.5	Conclusions . . . . .	62
<b>4</b>		<b>Model selection for regularized errors-in-variables systems</b>	<b>65</b>
	4.1	Introduction . . . . .	65
	4.2	Loss function for errors-in-variables linear models . . . . .	66
	4.2.1	Prediction error vs. generalization error . . . . .	66
	4.2.2	Model selection based on prediction or generalization error . . . . .	67
	4.2.3	Optimal regularization parameter . . . . .	68
	4.3	Consistent cross validation . . . . .	68
	4.3.1	Consistency theorem . . . . .	68
	4.3.2	Computational properties . . . . .	72
	4.3.3	Numerical illustration of the consistent cross validation . . . . .	72
	4.4	Methods for choosing truncation levels . . . . .	73
	4.5	Methods for choosing the regularization parameter in RTLS . . . . .	78
	4.5.1	Numerical results for RTLS . . . . .	81
	4.6	Conclusions . . . . .	82
<b>II</b>		<b>Regularization for nonlinear problems</b>	<b>83</b>
<b>5</b>		<b>Nonparametric regression using template splines</b>	<b>85</b>
	5.1	Introduction . . . . .	85

---

5.2	Template splines on reproducing kernel Hilbert spaces . . . . .	86
5.2.1	Reproducing kernel Hilbert spaces . . . . .	86
5.2.2	Unconstrained smoothing . . . . .	87
5.2.3	Constrained smoothing and template splines . . . . .	87
5.3	Computing template splines . . . . .	88
5.3.1	Transformation to a linear least squares problem . . . . .	88
5.3.2	Data driven spline fitting using generalized cross validation . . . . .	90
5.4	Examples . . . . .	91
5.4.1	Smoothing, regression and penalized splines as template splines . . . . .	91
5.4.2	Other applications of template splines . . . . .	94
5.5	Conclusions . . . . .	96
<b>6</b>	<b>Regularized semiparametric modeling</b> . . . . .	<b>97</b>
6.1	Introduction . . . . .	97
6.2	Semiparametric model with smoothness constraint . . . . .	98
6.2.1	Model formulation . . . . .	98
6.2.2	Spline fitting for the nonparametric part . . . . .	99
6.2.3	Computationally efficient method using the Levenberg-Marquardt algorithm . . . . .	100
6.2.4	Efficient computation of function and Jacobian values . . . . .	100
6.2.5	Choice of regularization parameter . . . . .	101
6.2.6	Efficient computation of the GCV function value . . . . .	105
6.3	Asymptotic properties of semiparametric regression . . . . .	106
6.3.1	Asymptotic normality . . . . .	106
6.3.2	Asymptotic confidence intervals . . . . .	108
6.4	Discussion on identifiability, redundancy and uniqueness . . . . .	110
6.5	Numerical examples . . . . .	111
6.5.1	Description of simulation examples and results . . . . .	111
6.5.2	Comparison between classical confidence intervals and specialized confidence intervals . . . . .	113
6.5.3	Illustration of identifiability problems . . . . .	114
6.6	Conclusions . . . . .	116
<b>7</b>	<b>MRS data quantification and the AQSES software</b> . . . . .	<b>117</b>
7.1	Introduction . . . . .	117
7.1.1	MRS data quantification with unknown macromolecular baseline . . . . .	117
7.1.2	Software for MRS quantification . . . . .	120
7.2	Mathematical formulation . . . . .	122
7.2.1	A semiparametric model for MRS signals . . . . .	122
7.2.2	Using a filter . . . . .	123
7.3	The software package AQSES . . . . .	124
7.3.1	The AQSES GUI framework . . . . .	124
7.3.2	Implementation details . . . . .	125
7.4	Numerical results . . . . .	128

7.4.1	Simulated signals . . . . .	128
7.4.2	Results on simulated data . . . . .	129
7.4.3	Experiments with real data . . . . .	131
7.5	Conclusions . . . . .	132
<b>8</b>	<b>Constrained variable projection implementation</b>	<b>135</b>
8.1	Introduction . . . . .	135
8.2	Separable least squares with constraints on nonlinear variables . . . . .	137
8.2.1	MRS data model without baseline . . . . .	138
8.2.2	MRS data model with baseline . . . . .	140
8.3	Separable least squares with constraints on linear variables . . . . .	141
8.3.1	Motivation: MRS data quantification with equal phases and non-negativity constraint for the amplitudes . . . . .	141
8.3.2	MRS data model without baseline . . . . .	142
8.3.3	MRS data model with baseline . . . . .	146
8.4	Numerical experiments . . . . .	147
8.4.1	Properties of the pseudo-Jacobian . . . . .	147
8.4.2	Improvements of equal phases compared to the non-equal phases version . . . . .	150
8.5	Conclusions . . . . .	151
<b>9</b>	<b>Conclusions and open problems</b>	<b>153</b>
9.1	General conclusions of the thesis . . . . .	153
9.1.1	Regularization for linear problems . . . . .	153
9.1.2	Regularization for nonlinear problems . . . . .	154
9.2	Future work and open problems . . . . .	154
<b>A</b>	<b>More theory on model selection</b>	<b>157</b>
A.1	Derivation of generalized cross validation . . . . .	157
A.1.1	From leave-one-out to generalized cross validation . . . . .	157
A.1.2	Computation of the influence matrix . . . . .	158
A.2	Derivation of information criteria . . . . .	159
<b>B</b>	<b>Fortran implementation of AQSES</b>	<b>163</b>
	<b>Bibliography</b>	<b>167</b>

## Chapter 1

# Introduction

This thesis explores the topic of solving ill-posed problems by using regularization. In the beginning of the 1900's, Jacques Hadamard defined a problem as “ill-posed” if the solution of the problem is not unique, or if it is not a continuous function of the data. He believed that such problems are abstract formulations and that in nature only “well-posed” problems arise. In truth, ill-posed problems arise in many practical situations. Such problems are extremely sensitive to noise in the data; that is, small perturbations can lead to very large changes in the solution.

In this chapter we introduce the estimation problems that will be encountered in the thesis. They encompass linear and nonlinear regression problems, as well as parametric, nonparametric or semiparametric models. We discuss and exemplify the situations when ill-posed problems arise. In Section 1.2 we briefly survey the theory of regularization for ill-posed problems, while in Section 1.3 we discuss model selection techniques that can be used in the context of regularization methods.

This chapter ends with an overview of the original contributions and a summary of the other chapters.

## 1.1 Estimation problems

We discuss first some classical parametrized model formulations and corresponding estimation techniques that range from linear regression through errors-in-variables regression to nonlinear regression.

The parameter estimation problems are customarily divided into linear and nonlinear estimation, depending on whether the parameter of interest appears linearly or nonlinearly into the considered model formulation.

We shall adhere to the classical formulation in which a parametrized model approximates in a certain sense *measured outputs*. Our general model is written as:

$$F(x) \approx y, \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m. \quad (1.1)$$

### 1.1.1 Estimation in linear problems

In linear problems, the parametric model is *linear* in the parameters, thus it can be expressed as a matrix-vector product of the type  $Ax$ , where  $A$  is an  $m \times n$  matrix; we shall consider throughout that  $m \geq n$ .

#### Linear regression: least squares

The classical least squares (LS) technique approaches the problem of estimating  $x$  in the system  $Ax \approx b$  in the following way:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|^2.$$

Statistically, the least squares solution  $x^{\text{LS}}$  is the *maximum likelihood* solution in the case when the data in the matrix  $A$  and the vector  $b$  come from a *true model*  $Ax^{\text{true}} = b + \tilde{b}$ , where  $\tilde{b}$  is normally distributed with zero mean and covariance matrix  $Q$ , and the norm  $\|\cdot\|$  is the weighted Euclidean norm  $\|\cdot\|_{Q^{-1}}$  (i.e.,  $\|v\|_{Q^{-1}}^2 = v^\top Q^{-1}v$ ).

Computationally, the least squares solution admits the closed-form expression  $x^{\text{LS}} = A^\dagger b$ , where  $A^\dagger$  is the Moore-Penrose pseudoinverse,  $A^\dagger = (A^\top A)^{-1}A$ . The least squares solution can also be expressed in terms of the singular value decomposition (SVD) [50] of the matrix  $A$ . Numerical methods for computing  $x^{\text{LS}}$  are ranging from direct methods (based on SVD or the QR factorization) to iterative methods appropriate for large, sparse or structured problems [10].

#### Errors-in-variables: total least squares

The total least squares (TLS) method [49, 132] explicitly allows correction on the matrix  $A$  as well as on the right-hand-side  $b$ . It minimizes the criterion:

$$\min_{x \in \mathbb{R}^n, \Delta A \in \mathbb{R}^{m \times n}, \Delta b \in \mathbb{R}^m} \left\| \begin{bmatrix} \Delta A & \Delta b \end{bmatrix} \right\|_F^2 \quad \text{subject to } (A + \Delta A)x = (b + \Delta b), \quad (1.2)$$

where the norm  $\|\cdot\|_F$  is the Frobenius norm of a matrix, i.e., the square root of the sum of squares of all the elements in the matrix.

The solution  $x^{\text{TLS}}$  of the TLS problem is the maximum likelihood solution in the case when the data in the matrix  $A$  and the vector  $b$  come from a true model (an *errors-in-variables* model)  $(A + \tilde{A})x^{\text{true}} = b + \tilde{b}$ , where all elements in  $\tilde{A}$  and  $\tilde{b}$  are independent and identically distributed with zero mean and equal variance.

Efficient and reliable numerical methods to compute the TLS solution were developed in the literature [49, 132], and they are based on the singular value decomposition. The TLS problem (1.2) admits unique solution (under the condition that the smallest singular value  $\sigma_{n+1}$  of  $\begin{bmatrix} A & b \end{bmatrix}$  is strictly smaller than the smallest singular value  $\sigma'_n$  of  $A$ ) and the solution has a closed-form expression:  $x^{\text{TLS}} = (A^\top A - \sigma_{n+1}^2 I)^{-1} A^\top b$ .

As extensions of the TLS problem, we mention that methods are developed for the problem when only some of the columns of the matrix  $A$  are contaminated by noise and the rest are noise-free [44, 132]. For other noise statistics, there exist specialized weighted versions of the total least squares problem, for which we refer to [85, Chapter 3] and the

references therein. An important special class of problems that appear in practice involves *structured* data matrices (Toeplitz, Hankel, etc). The structured total least squares problem has the same formulation as the TLS problem (1.2), with the additional constraint that  $[\Delta A \quad \Delta b]$  has the same structure as  $[A \quad b]$  [22, 77, 78].

### 1.1.2 Estimation in nonlinear problems

When observed variables exhibit nonlinear relations among each other, estimation methods are provided by the nonlinear regression theory. Nonlinear least squares, maximum likelihood, quasi likelihood, or Bayesian estimation methods are typical techniques that can be defined as well in a nonlinear setting, similarly to the case of linear regression [111]. However, in nonlinear regression, problems related to identifiability, ill-conditioning, convergence of numerical algorithms, design of confidence intervals, etc., are much harder to solve than in the linear case.

Under regularity assumptions [111, Chapter 12], nonlinear regression has also desirable (asymptotic) properties. The result in the following paragraph, adapted from [111, Chapter 1], is related to the nonlinear least squares estimation in the Gaussian noise case:

Given  $m$  observations  $(\mathbf{t}_i, y_i) \in \mathbb{R}^q \times \mathbb{R}$ ,  $i = 1, 2, \dots, m$ , from a nonlinear model with known functional relationship  $F : \mathbb{R}^q \times \mathbb{R}^n \rightarrow \mathbb{R}$ ,

$$y_i = F(\mathbf{t}_i, x^*) + \varepsilon_i \quad (i = 1, 2, \dots, m),$$

where  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$  and  $x^*$  denotes the true value of the unknown parameter  $x \in \mathbb{R}^n$ , then the least squares estimate of  $x^*$ , *i.e.*, the global minimizer  $\hat{x}$  of

$$R(x) := \sum_{i=1}^m (y_i - F(\mathbf{t}_i, x))^2$$

over  $x \in \mathbb{R}^n$ , satisfies:

1.  $\hat{x}$  is a consistent estimate of  $x^*$  with  $\hat{x} - x^* \sim \mathcal{N}(0, \sigma^2(\mathcal{A}^\top \mathcal{A})^{-1})$  where  $\mathcal{A} := \left[ \frac{\partial F}{\partial x}(\mathbf{t}, x^*) \right]$ ;
2.  $s^2 := R(\hat{x})/(m - n)$  is a consistent estimate of the variance  $\sigma^2$ .

Normality of  $\varepsilon$  is not required to prove the consistency of  $\hat{x}$ ; the zero mean condition  $\mathcal{E}(\varepsilon) = 0$  is sufficient. This result gives, thus, the necessary justification of using nonlinear least squares when fitting nonlinear models – under zero-mean noise assumption.

### 1.1.3 When ill-posed problems arise

When we formulate the general model (1.1), we implicitly define an *inverse problem*. The established terminology is: the evaluation of  $F(x)$  from a given  $x$  is called the *forward problem*; computing  $x$  back from measured  $y$  is the *inverse problem*. In ill-posed inverse problems, there exists a well-defined *forward operator*, such that, in general, the forward problem of evaluating the left hand side ( $F(x)$ ) is well-conditioned, but the map  $F$  is either noninvertible, or, if the inverse exists, the computation of  $F^{-1}(y)$  is very badly conditioned.

This means that any small perturbation on  $y$  implies large changes in the estimated solution  $F^{-1}(y)$ .

The particular method used for estimation is not in itself the cause for uncertainties in the solution: inverse problems have an intrinsic uncertainty that depends crucially on the forward operator and on the distribution of the possible observational errors.

In the case of linear ill-posed problems, least squares, total least squares or other classical methods for solving an (overdetermined) linear system  $Ax \approx b$ , when the coefficient matrix  $A$  is ill-conditioned, might provide a solution that is physically meaningless for the given problem. This happens when  $A$  is (nearly) rank-deficient with no significant gap in the singular values. Typical examples are encountered when the system is a discretization of a continuous ill-posed problem [58].

### Examples of linear ill-posed problems

**Example 1.1 (First kind integral equation with a smooth kernel)** We introduce the linear ill-posed problems with a classical example: the solution of linear equations (or linear least squares) that arise from discretizations of integral equations of the first kind [134]. This example gives us the chance to mention the (*discrete*) *Picard condition*, a test that applies to linear ill-posed problems and gives an indication about the existence of acceptable regularized solutions.

A first kind integral equation in one dimension,

$$\int_0^1 K(s,t)f(t)dt = g(s), \quad 0 \leq s \leq 1,$$

gives rise, through discretization, to a (possibly slightly incompatible, due to approximations during numerical discretization) linear system of equations  $Ax \approx b$ . When  $K$  is a *smooth kernel*, then any rough, even discontinuous, function  $f$  is transformed into a smooth  $g$ . The inverse problem of estimating  $f$  from given  $K$  and  $g$  is, therefore, ill-posed.

If  $K$  is a square integrable kernel ( $\int_0^1 \int_0^1 K(s,t)dt ds < \infty$ ), then it admits an (infinite) expansion  $K(s,t) = \sum_{i=1}^{\infty} \lambda_i \phi_i(s) \psi_i(t)$ , where  $\lambda_i$  are scalars and  $\{\phi_i\}_{i \geq 1}$ ,  $\{\psi_i\}_{i \geq 1}$  are families of orthonormal functions. The smoothness of the operator  $K$  is related to the decay rate towards zero of its “singular values”  $\lambda_i$ . (Clearly, they must converge to zero whenever  $K$  is square integrable.) If the  $\lambda_i$ 's decrease exponentially, the kernel is very smooth and the estimation problem is severely ill-posed. In the case of a polynomial decay, a moderately or mildly ill-posed problem is obtained.

The Picard condition says that a square integrable  $f$  can be recovered from  $K$  and  $g$  only if  $\sum_{i=1}^{\infty} (\beta_i / \lambda_i)^2$  is finite, where  $\beta_i$  denotes the scalar coefficient of  $\phi_i$  when  $g$  is expanded in the  $\phi_i$  basis:  $g(s) = \sum_{i=1}^{\infty} \beta_i \phi_i(s)$ .

The link between the expansion of  $K$  and the singular value decomposition of  $A$  is not straightforward (note that  $A$  can be obtained from a variety of discretization schemes). However, the ill-posedness of the original continuous problem is inherited by its discretized version. The degree of ill-posedness of  $Ax \approx b$  can also be quantified in terms of the decay rate of the singular values of  $A$ . Moreover, there exists a *discrete Picard condition* saying that a reasonable solution  $x$  can be obtained only if the singular values of  $A$  decay slower than the coefficients  $\beta_i$ , which in this case are the linear combination coefficients when  $b$  is expanded in the basis of the left singular vectors of  $A$ .

**Example 1.2 (Differentiation, nonsmooth kernels and ill-posed discretizations.)** In the previous example we emphasized that when an integral problem with smooth kernel is ill-posed, then its discretizations are also ill-posed discrete problems. Now we illustrate the situation when a problem is well-posed in the continuous domain, but it becomes ill-posed through a (possibly inappropriate) discretization scheme.

If  $f : [0, 1] \rightarrow \mathbb{R}$  is a given differentiable function, then the computation of its derivative(s) is well-posed. However, if  $f$  is only given through a few discrete points on its graph, then the numerical estimation of its derivative might become an ill-posed problem. Differentiation can be seen as an integral equation with *nonsmooth kernel*. Indeed, the first derivative function  $g = f'$  satisfies a first kind integral equation:  $\int_0^s g(t)dt = f(s) - f(0)$ , or, equivalently,

$$\int_0^1 h(s-t)g(t)dt = f(s) - f(0), \quad (1.3)$$

where  $h$  is the unit step function. Higher order differential operators can also be expressed as integral equations with nonsmooth kernels. Discretization of such a kernel might yield (mildly) ill-posed discrete problems.

Another source of ill-posedness appears in *boundary value differential equations*, if they are discretized on inappropriate grids. Equispaced grid points combined with polynomial interpolation yield catastrophically unstable results, while unevenly spaced grid points that cluster near the boundaries (*e.g.*, Chebyshev points) combined with spectral methods [126] provide much more accurate tools.

### Ill-posed problems in applications

Many application areas give rise to ill-posed problems. Unstructured ill-posed problems can arise from discretizations of integral or differential operators, in applications from medical imaging (electrical impedance tomography, X-ray tomography, optical tomography), bioelectrical inversion problems (inverse electrocardiography, magnetocardiography, electroencephalography [113]), geophysical applications (seismology, radar or sonar imaging [24], atmospheric sciences [62], oceanography), and many others. Regularization can be required for structured problems as well: deconvolution problems in image deblurring [98], in medical applications (renography) [87] and in signal restoration [138].

## 1.2 Regularization

### 1.2.1 Goals of regularization: between stabilization and meaningful information

Instead of attempting a rigorous definition, we introduce *regularization* as any technique of modifying the original ill-posed estimation problem with the goals of *stabilizing the solution* and/or obtaining a *meaningful solution*. We have thus divided the principles of regularization into two categories.

1. *Stabilization*. A goal for the modified regularized problem is that it has a unique solution and a much lower sensitivity than the original ill-posed problem. Various classical regularization methods achieve stabilization by rather simple numerical tricks,

without special concern about the problem at hand. Among these general purpose regularization techniques, we mention *truncation methods* and *Tikhonov-type methods*.

Truncation is used in combination with a certain *decomposition* or *expansion* of the original problem; “high frequency” components (*i.e.*, terms that are more oscillatory and prone to numerical instabilities) are cut out of the original problem, in the hope that the remaining part of the problem is well-conditioned. Various forms of truncation are often used in, *e.g.*, signal processing, as a method for “denoising.”

Tikhonov regularization adds a penalty term, such as the norm of the  $x$  variable. In this way, a trade-off is obtained between the actual model fitting (*e.g.*, in the least squares case, measured by the misfit  $\|F(x) - y\|_2^2$ ), and the variations in the solution  $x$  (measured by the Euclidean norm  $\|x\|_2$  or another (semi)norm  $\|Lx\|_2$ ).

The simple formulation of the Tikhonov regularization problem in the linear least squares case,

$$\min_x \|Ax - y\|_2^2 + \lambda \|x\|_2^2,$$

has the closed-form solution  $x^{\text{Tik}}(\lambda) = (A^\top A + \lambda I)^{-1} A^\top y$ . The ill-conditioning of the original least squares formulation, whose solution has the closed-form expression  $x^{\text{LS}} = A^\dagger y = (A^\top A)^{-1} A^\top y$ , is caused by numerical problems when trying to implicitly invert  $A^\top A$ . This is avoided in the Tikhonov solution for an appropriately chosen value of  $\lambda > 0$  that yields a well-conditioned matrix  $A^\top A + \lambda I$ .

2. *Meaningful solution.* Although in many cases reasonable solutions can be obtained by stabilizing the numerical computations with simple regularization methods, there are still numerous problems that need (or at least that would benefit from) more dedicated regularization techniques. These methods can only be designed if there exist additional pieces of information or a certain *prior knowledge* coming from the physical significance of the solution.

Varah [134] has observed that without prior knowledge we can only infer whether the computed solution is *reasonable* or not by checking the magnitude of the residual. Any regularized solution that gives a residual of about the same magnitude as a prescribed “noise level” can be considered a reasonable solution. However, this does not mean that the regularized solutions are close to the “true solution” (as was shown in the simulations of [134]).

Some forms of prior knowledge can be expressed in such a way that simple penalties (*e.g.*, Tikhonov-type) can be helpful enough. Here are some examples:

- If the solution  $x$  should have elements as small as possible, a penalty of the form  $\|x\|_2$  could be used.
- If  $x$  should be a sparse solution vector, the penalty on the  $l_1$ -norm could be used:  $\|x\|_1 := \sum_i |x_i|$ .
- If the vector  $x$  is actually a discretization of a function  $f$ , and the function  $f$  should have a certain degree of smoothness, penalties of the form  $\|Lx\|_2$  can be used, where  $L$  denotes a discretized differential operator; *e.g.*, the first order

derivative operator is approximated with the matrix  $L_1 = \begin{bmatrix} -1 & 1 & 0 \\ & \ddots & \ddots \\ 0 & & -1 & 1 \end{bmatrix}$ , the discrete second-order differential operator is  $L_2 = \begin{bmatrix} -1 & 2 & -1 & 0 \\ & \ddots & \ddots & \ddots \\ 0 & & -1 & 2 & -1 \end{bmatrix}$ , and so on.

### 1.2.2 General penalty-type regularization

We define a general *penalty-type regularization* as the optimization problem

$$\min_x \mathcal{L}(F(x), y) + \lambda \mathcal{R}(x), \quad (1.4)$$

where  $\mathcal{L}$  is a non-negatively valued *loss function* that measures the distance between the data  $y$  and the model  $F(x)$ ,  $\mathcal{R}$  is a non-negatively valued penalty function that reflects desirable constraints on the solution  $x$ , and the scalar  $\lambda > 0$  is the factor that controls the trade-off between the two objectives.

Depending on the context, the scalar  $\lambda$  is known in the literature as the *penalty parameter*, the *regularization parameter*, or the *smoothing parameter*.

**Remark 1** The penalized problem (1.4) can be seen as a particular scalarization of the double objective (Pareto) minimization  $\min_x (\mathcal{L}(F(x), y), \mathcal{R}(x))$  [13, §6.3.1]. An arbitrary choice of  $\lambda$  in (1.4) gives a Pareto optimal solution for the bi-criterion optimization. In fact, by varying  $\lambda$  over  $(0, \infty)$ , all Pareto-optimal solutions are obtained.

When more than one restriction needs to be imposed on  $x$ , several regularization parameters will be needed as well; the total penalty term will have the form  $\sum_{k=1}^p \lambda_k \mathcal{R}_k(x)$ .

#### Tikhonov regularization for regularized least squares

The most commonly used regularization method for regularized least squares is due to Tikhonov [123, 124]. It amounts to solving the problem:

$$\min_x \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2, \quad (1.5)$$

where  $\lambda > 0$  is a fixed, properly chosen regularization parameter that controls the allowed “size” of the solution vector  $x$ , and  $L$  is a matrix that defines a (semi)norm on the solution through which the “size” is measured.

Consider the singular value decomposition of the coefficient matrix  $A$ :  $A = U'\Sigma'V'^\top = \sum_{i=1}^r \sigma'_i u'_i v'_i{}^\top$  ( $r = \text{rank}(A)$ ). Then, the (unstable) least squares solution is given by

$$x^{\text{LS}} = \sum_{i=1}^r \frac{u'_i{}^\top b}{\sigma'_i} v'_i. \quad (1.6)$$

Tikhonov regularization in the form (1.5) with  $L = I_n$  (called *standard form*) provides a solution

$$x^{\text{Tik}}(\lambda) = \sum_{i=1}^r \frac{\sigma'_i{}^2}{\sigma'_i{}^2 + \lambda} \frac{u'_i{}^\top b}{\sigma'_i} v'_i. \quad (1.7)$$

**Remark 2** Formula (1.7) illustrates a more general characteristic of regularized solutions; many regularization methods compute solutions of the form:

$$x^{\text{FF}} = \sum_{i=1}^r f_i \frac{u_i^\top b}{\sigma_i} v_i, \quad (1.8)$$

where the filter factors  $f_i$  ( $0 \leq f_i \leq 1$ ) are meant to “filter out” the contribution of the noise. For Tikhonov regularization, these factors are computed as  $f_i(\lambda) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$ . See [15] for a regularization method based on a filter function of Gaussian type.

### 1.2.3 Truncation methods

Truncation methods are applicable in the case when an *expansion* of the solution, or at least of the forward operator, is computable. Components corresponding to unwanted subspaces are then completely ignored when computing a solution to a truncated problem.

In linear problems, truncation methods (truncated SVD, truncated generalized SVD, truncated TLS, [33, 57]) involve computing the SVD of a certain matrix and using only the information corresponding to several of the largest singular values. For instance, the truncated SVD solution is obtained taking the sum of the first  $k < r$  terms in (1.6). This corresponds to setting filter factors in (1.8) to either 0 or 1, depending on whether their indices are larger or smaller than  $k$ , respectively.

Similarly, the truncated total least squares (TTLS) solution is computed as the ordinary TLS solution [132], but a lower value  $k$  is used instead of the numerical rank  $r$  of  $[A \ b]$ . Experiments in [33] show that in some cases TTLS outperforms other regularization methods such as Tikhonov regularization, truncated SVD or the LSQR method [93].

Note that the truncation level  $k$  is not obvious when dealing with a discrete ill-posed problem, because the singular values decay smoothly towards zero;  $k$  can be considered as the unknown regularization parameter.

### 1.2.4 Deterministic and statistical regularization

**Deterministic** regularization methods refer to the methods that are designed from a numerical algebra point of view, without explicit underlying statistics. The main streams of methods can, however, be recognised as appearing in both deterministic and statistical settings.

To give a flavour of the deterministic regularization methods’ extent, we remind here the different points of view that can be linked to regularization in the linear problems case.

- One role of regularization is numerical stabilization. In the example of “standard form” Tikhonov regularization, this amounts to solving the better conditioned regularized system  $(A^\top A + \lambda I)x = A^\top b$ , instead of the original ill-conditioned normal system of equations  $A^\top Ax = A^\top b$ .
- A second role of regularization is to filter out the components corresponding to small singular values. Many regularization techniques for linear ill-posed problems can be written in the form of an expansion where filter factors are present.

- The third interpretation of penalty-type regularization is related to multi-objective optimization.

**Statistical** interpretations of regularization include:

- On one hand, the use of appropriate probability distributions models for the possible disturbance components (measurement noise);
- On the other hand, imposing appropriate prior constraints on the unknown variables.

In statistics, shrinkage estimators for linear regression (James-Stein type [66]) were designed in order to obtain a better mean square error than the ordinary linear regression. In fact, the shrinkage estimators are related to Tikhonov regularization in standard form, where a prescribed value of  $\lambda$  depending on the (assumed known) error variance is used:

$$\lambda = \frac{(n-2)\sigma^2}{\|x^{\text{LS}}\|_2^2}.$$

Another term with statistical connotations is *Bayesian regularization*. The Bayesian technique of maximum *a posteriori* (MAP) estimation can be seen, in the linear regression case, as a Tikhonov regularization method, as well. As a simple example, assume that one has a prior distribution for the regression vector  $x$  from the model  $Ax \approx b$ :  $x \sim \mathcal{N}(\mu, C)$ . Then, under the extra assumption that the noise on  $b$  is white Gaussian, the MAP estimate of  $x$  is:  $\hat{x} = (A^\top A + C^{-1})^{-1}(A^\top b + C^{-1}\mu)$ . Note that the inverse of the covariance matrix takes the role of a regularization matrix. (When  $C = \infty$ , we get the least squares solution; it corresponds to the case when no prior about  $x$  is imposed.)

Applying Bayes rule to optimize regularization parameters is studied in [81]. Bayesian regularization is also coined in the neural networks literature [34] as a method for nonlinear regression, solved iteratively with a Levenberg-Marquardt [90] type of algorithm. The idea is that the Levenberg-Marquardt regularization parameter that is used in every iteration to solve a Gauss-Newton system is optimized by examining a posterior distribution, as in [81].

Bayesian regularization seems a powerful tool in what concerns the *inference*, from given data, of optimal (*i.e.*, most probable) parameter values of a model, and then ranking the models within a family of models, based on data.

An even more challenging task to undertake is the choice of the regularization models themselves. Recently, supervised learning was employed for regularization problems [54]. The paper [54] focuses on nonlinear inverse problems, solved with penalty regularization. The choice of a specific penalty is performed through an optimization problem involving a training data set.

### 1.2.5 Newer trends in regularization methods

Classical methods of regularization involve least squares problems and Euclidean norm penalties. Over the years, various problem formulations arose, modifying these elements in order to cope with different goals for the solution.

Specifically, the Euclidean norm in the penalty is replaced by the 1-norm in the method called the *lasso* [122]; in the case of linear ill-posed problems, the lasso becomes:

$$\min_x \|Ax - b\|_2^2 + \lambda \|x\|_1,$$

and its merits are that a *sparse* solution (an  $x$  with many zero elements) is favored.

More general formulations have been analyzed. Consider, for instance, the problem treated (from theoretical and computational points of view) in [20]:

$$\min_x \|Ax - b\|_p + \lambda \|x\|_s^s,$$

where  $p$  and  $s$  can have any value  $\geq 1$ . General norms  $\|Ax - b\|_p$  (with  $p < 2$ ) are useful when robustness against outliers in the data is an issue.

Unlike regularization methods that are used for *smoothing* in, e.g., curve fitting, the field of *image processing* gave rise to regularization techniques that take into account the idea that they must not smoothen some rough features that are important in the image. In this category, we mention the work in [17] aimed at preserving edges in images: this constraint can be imposed by using special nonquadratic penalty functions.

Another popular method in image denoising/deblurring is total variation (TV) regularization, originating in [104]. This technique was developed for continuous ill-posed problems, defined by partial differential equations. The TV function is defined as:

$$\text{TV}(f) = \int_{\Omega} |\nabla f(t)| dt,$$

and its goal, as a penalty function, is to measure an overall variation in the derivatives of (the image)  $f$ , allowing, however, discontinuities.

### 1.3 Model selection techniques

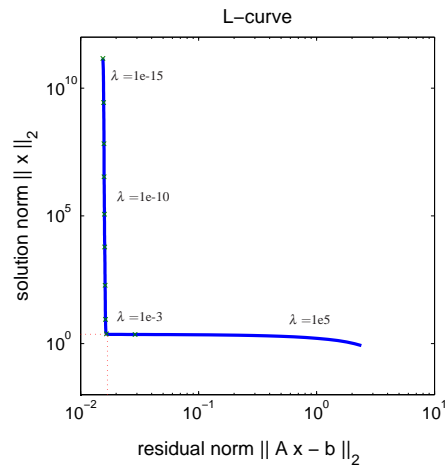
In this section, we discuss a few aspects related to the field of model selection, since model selection techniques will help us tackling the appropriate solutions of the studied ill-posed problems.

We saw in §1.2.4 that regularization can be viewed in deterministic settings as methods for getting rid of ill-conditioning or for adding constraints through penalties. However, solid theoretical techniques for model selection that can be used in regularization for choosing the trade-off between model fitting and stability of solution (via penalties or truncation), are intrinsically *statistic in nature*. Model selection is a very delicate task, because most of the time model selection is combined with estimation of model parameters from the same available data. The aim is to establish reasonable trade-offs between goodness-of-fit and complexity of the model, and between bias and variance.

Statistical model selection methods are often imported in problems with unknown statistical assumptions, but their optimality is no longer guaranteed. Other methods for deterministic regularization are based on heuristic ideas.

#### 1.3.1 The discrepancy principle

The **discrepancy principle** [91] is a method that selects from a family of possible models a model giving a residual that best corresponds to certain *known* statistical properties of the noise. The simplest case is the linear model with white Gaussian measurement noise. Thus, if  $Ax \approx b$  originates from a system  $Ax^{\text{exact}} = b^{\text{exact}}$ , with  $b = b^{\text{exact}} + \varepsilon$ , where  $\varepsilon$  is white noise with known variance  $\sigma^2$ , and  $x_1^{\text{reg}}, \dots, x_q^{\text{reg}}$  are candidate regularized solutions,



**Figure 1.1.** A typical example of an L-curve for a linear ill-posed problem. The norm of the regularized solution  $\|x^{Tik}(\lambda)\|_2$  is plotted against the norm of the residual error  $\|Ax^{Tik}(\lambda) - b\|_2$  in log-log scale. The optimal regularization parameter is the  $\lambda$  corresponding to the corner of the L-curve.

then the discrepancy principle chooses a solution that gives a residual error that minimizes the quantity:

$$\left| \|Ax_i^{reg} - b\|_2^2 - m\sigma^2 \right|.$$

### 1.3.2 The L-curve

The **L-Curve** is a criterion developed for Tikhonov regularization of linear ill-posed problems [56, 60, 58], but suited for other penalty-type regularization methods, as well. It chooses a good trade-off between minimizing the residual norm (*i.e.*,  $\|Ax^{Tik}(\lambda) - b\|_2^2$  in the linear case) and minimizing the value of the penalty (*i.e.*, the (semi)norm of the solution for Tikhonov regularization  $\|Lx^{Tik}(\lambda)\|_2^2$ ). These two quantities are computed for a range of possible regularized models (corresponding to several values of  $\lambda$ ) and then plotted against each other in log-log scale. In many cases, the typical L-shaped plot, as shown in Figure 1.1 for a linear ill-posed system and the Tikhonov method, is obtained. The model (*i.e.*, the  $\lambda$ ) corresponding to the *corner* of the obtained L-curve is chosen as optimum.

This intuitive strategy can be transformed into a minimization problem by using Reginska's modification [102] that rotates the L-curve, such that locating the corner becomes minimizing a function. Another strategy used by Hansen [60] is the maximum curvature criterion. The curvature of the L-curve has a certain computable formula; the corner of the L-curve corresponds to the point of maximum curvature. These methods are developed for Tikhonov regularization (where, in particular, the regularization parameter is continuous), but they can be easily adapted to the choice of a discrete truncation level  $k$ .

### 1.3.3 Cross validation and generalized cross validation

Another family of methods widely used in model selection involves cross validation techniques. Cross validation, in its various forms, ranges from parameter selection to probability density estimation, from classification to stopping criteria in training neural networks. These techniques can also be used in the context of choosing a good regularized model from a set of competing models (in particular, selecting the value of a regularization parameter or a truncation level).

Cross validation relies on repeatedly splitting the available data into estimation and validation parts, computing several models based on the estimation parts, and picking the model that minimizes a certain criterion applied on the validation parts.

In a simplified framework, let  $\{D_1, \dots, D_m\}$  denote given data that comes from an unknown model  $\mathcal{M}^{\text{true}}$ , which is parameterized by an unknown parameter  $\theta^{\text{true}}$ . Let  $\{I_1, I_2, \dots, I_c\}$  be a partition of the set of indices  $\{1, 2, \dots, m\}$ . For a fixed parameter  $\theta$  and each set  $I_j$ , a “partial” model  $\mathcal{M}_{-I_j}(\theta)$  can be estimated using only a subset of data from  $\{D_1, \dots, D_m\}$ , which excludes data with indices in  $I_j$ . Then the performance of the partial model is estimated under a certain error function  $\mathcal{L}$ , and the cross validation function is defined as

$$CV(\theta) := \frac{1}{c} \sum_{j=1}^c \mathcal{L}(\mathcal{M}_{-I_j}(\theta), D_{I_j}). \quad (1.9)$$

The non-negatively valued function  $\mathcal{L}$  must measure the error of assuming that the partial model  $\mathcal{M}_{-I_j}(\theta)$  describes also the samples  $D_{I_j}$  (which are not used in the process of constructing  $\mathcal{M}_{-I_j}(\theta)$ ).

**Remark 3 (on notation)** Intentionally we use the notation  $\mathcal{L}$  for the error measure as for the loss function defined in (1.4). We use it whenever we want to designate a (non-specified) error measure between a (parametrized) model and a data set.

The value of  $\theta$  that minimizes  $CV$  is selected as the cross validation parameter and it is used to construct the cross validated model  $\mathcal{M}(\theta)$ .

**Example 1.3 (Cross validation for regularized least squares)** In Tikhonov-type regularized least squares (see (1.5)), the regularized solution with regularization parameter  $\lambda$  is  $x^{\text{Tik}}(\lambda) = (A^\top A + \lambda L^\top L)^{-1} A^\top b$ . Using the formalism introduced before, the  $D_i$  variables are the rows  $[A_i \quad b_i]$  of the data matrix  $[A \quad b]$  and the model  $\mathcal{M}(\lambda)$  is parameterized by the solution  $x^{\text{Tik}}(\lambda)$ . Similarly, the partial models  $\mathcal{M}_{-I_j}(\lambda)$  are one-to-one with the partial solutions  $x_{-I_j}(\lambda) = (A_{-I_j}^\top A_{-I_j} + \lambda L^\top L)^{-1} A_{-I_j}^\top b_{-I_j}$ . (Here,  $A_{I_j}$ ,  $b_{I_j}$  denote the rows of  $A$ , elements of  $b$ , respectively, with indices in  $I_j$ , and  $A_{-I_j}$ ,  $b_{-I_j}$  denote the rows of  $A$ , elements of  $b$ , respectively, with indices in  $\{1, 2, \dots, m\} \setminus I_j$ .) The error function is the quadratic loss function

$$\mathcal{L}(x, [A_{I_j} \quad b_{I_j}]) = \|A_{I_j} x - b_{I_j}\|_2^2, \quad (1.10)$$

and the classical cross validation function for regularized least squares is:

$$CV_{\text{RLS}}(\lambda) = \frac{1}{c} \sum_{j=1}^c \|A_{I_j} x_{-I_j}(\lambda) - b_{I_j}\|_2^2. \quad (1.11)$$

**Remark 4 (Leave-one-out and generalized cross validation)** The cross validation function (1.11) can be simplified in the *leave-one-out* case to

$$CV_{\text{RLS}}(\lambda) = \frac{1}{m} \left\| \text{diag}\{(1 - A_{ii}(\lambda))^{-1}\}_{i=1}^m (I - A(\lambda))b \right\|_2^2, \quad (1.12)$$

where the matrix  $A(\lambda) := A(A^\top A + \lambda L^\top L)^{-1}A^\top$  satisfies  $Ax^{\text{Tik}}(\lambda) = A(\lambda)b$ .

This equivalent formulation is the core of defining the generalized cross validation (GCV) criterion [43] as a weighted modification of (1.12), which is invariant under rotations of the coordinate system (*i.e.*, orthogonal transformations of  $A$ ):

$$GCV_{\text{RLS}}(\lambda) = \frac{\frac{1}{m} \|(I - A(\lambda))b\|_2^2}{\left[\frac{1}{m} \text{Tr}(I - A(\lambda))\right]^2}.$$

**Remark 5 (The influence matrix and the effective number of parameters)** The matrix  $A(\lambda)$  defined as  $A(A^\top A + \lambda L^\top L)^{-1}A^\top$  for Tikhonov regularized least squares bears the name of *influence matrix* (or, depending on the context, *smoother matrix*, *hat matrix*). The influence matrix (or influence function, in nonlinear models) can be defined for other more general regularization methods, and it is an important concept in model selection techniques. In short, an influence matrix/function  $A(\lambda)$  is a transformation that projects observed data into data *predicted* by a regularized model with regularization parameter  $\lambda$ .

The trace of the influence matrix is an important quantity in several model selection methods; it bears the name of the *effective number of parameters*. We shall denote it with  $p^{\text{eff}}$ .

We refer to Appendix A.1 for the proof that the cross validation function simplifies to a formulation of the type (1.12) and for a complete derivation of GCV, in a general nonlinear setting.

### 1.3.4 Information criteria

In this category, the model selection methods are heavily based on statistical frameworks. Selecting between competing models is interpreted as choosing between probabilistic distributions that can approximately explain given data. Statistical measures, such as the Kullback-Leibler distance between a “true” probability distribution and an arbitrary probability distribution, are of central importance. Information criteria aim at quantifying the loss of information that occurs when an approximate model is used instead of the unknown truth.

In a general form, an information criterion is the minimization of

$$GIC(\mathcal{M}, D) = \mathcal{L}(\mathcal{M}, D) + \mathcal{B}(\mathcal{M}, D)$$

where the loss function  $\mathcal{L}$  is (in classical information criteria) the negative log likelihood of the data using the model  $\mathcal{M}$ , when the model parameters are replaced by their maximum likelihood estimates (over the available data  $D$ ). The second term represents a bias estimate that should correct for the fact that an *averaged* maximized log-likelihood is used instead of the *expected* maximized log-likelihood.

The first crude asymptotic approximation was found by Akaike [1]: bias = number of model parameters. In the bias, a term involving also the dimensions of the data sample is included in the Bayesian information criterion of Schwarz [110]: bias = (number of model parameters)  $\times \log m$ . Finally, in quite general circumstances, the bias formula involves the *number of effective parameters*  $p^{\text{eff}}$  instead of the number of model parameters  $p$  (which equals the dimension of the parameter vector  $\theta$ , describing each model).

We show here a common formula of an information criterion, simplified for the linear regression case when the loss function becomes the residual sum of squares, and for the case when the model choice lies in the choice of a regularization parameter.

**Example 1.4 (Information criterion for regularized least squares)** Consider an ill posed regression problem  $Ax = b + \varepsilon$ , where  $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$  with unknown variance  $\sigma^2$ . Assume that a method ‘reg’ of regularization that depends on a regularization parameter  $\lambda$  is used, and that this method can be seen as a *projection*:  $b^{\text{reg}}(\lambda) := Ax^{\text{reg}}(\lambda) = A(\lambda)b$ . (For the standard Tikhonov regularization,  $A(\lambda) = A(A^\top A + \lambda I)^{-1}A^\top$ .) Then the negative log likelihood function becomes

$$\begin{aligned} \mathcal{L}(x^{\text{reg}}(\lambda); A, b) &= -\log \prod_{i=1}^m \left( \frac{1}{\hat{\sigma} \sqrt{2\pi}} \exp \left( -\frac{(Ax^{\text{reg}}(\lambda) - b)_i^2}{2\hat{\sigma}^2} \right) \right) \\ &= \frac{m}{2} \log 2\pi \hat{\sigma}^2 + \frac{\|Ax^{\text{reg}}(\lambda) - b\|^2}{2\hat{\sigma}^2}, \end{aligned}$$

which simplifies, when  $\hat{\sigma}^2$  is the maximum likelihood estimate  $\|Ax^{\text{reg}}(\lambda) - b\|^2/m$ , to

$$\mathcal{L}(x^{\text{reg}}(\lambda); A, b) = \frac{m}{2} \log \frac{2\pi}{m} + \frac{m}{2} \log \|Ax^{\text{reg}}(\lambda) - b\|^2 + \frac{m}{2}.$$

On the other hand, the bias correction term becomes (cf. Appendix A.2)

$$\mathcal{B}(\lambda; A, b) = p^{\text{eff}}(\lambda) = \text{Tr}A(\lambda).$$

Putting things together and ignoring the terms that do not depend on  $\lambda$ , we get the following generalized Akaike information criterion function for linear regression:

$$GIC(\lambda) = m \log \|Ax^{\text{reg}}(\lambda) - b\| + \text{Tr}A(\lambda).$$

We point to Appendix A.2 for the derivation details of several information criteria, in a general model selection setting.

### 1.3.5 Other model selection criteria

Finally, we shortly enumerate other model selection criteria that can be used in the context of choosing a regularized model (*i.e.*, a regularization parameter).

**Mallows’  $C_p$**  [82, 83] is a statistic similar to AIC and GCV, which chooses a trade-off between the model’s fit and the number of model parameters. It can be considered as a bias-corrected information criterion for the white Gaussian noise case (with variance  $\sigma^2$ ) with a correctly specified model:

$$\min C_p = \frac{\mathcal{L}(\mathcal{M}; D)}{\sigma^2} - m + 2p,$$

where  $m$  is the number of data samples and  $p$  is the number of parameters describing the model  $\mathcal{M}$ . ( $p$  can be replaced by the effective number of parameters  $p_{\text{eff}}$  when needed.)

**Generalized maximum likelihood** (GML) criterion is another related method, which has a better explanation in a probabilistic (Bayesian) setting. Suppose that we aim at selecting a single regularization parameter  $\lambda$  of a regularization method that is characterized by an influence matrix  $A(\lambda)$ . This means that the data  $y$  is linked to the regularized predicted  $\hat{y}_\lambda$  by  $\hat{y}_\lambda = A(\lambda)y$ . In the assumption that  $y$  is a noisy version of an  $y^{\text{exact}}$  satisfying  $y \sim N(y^{\text{exact}}, \sigma^2 I)$ , the maximum likelihood estimate of  $\lambda$  is found by minimizing the GML function [136]

$$GML(\lambda) = \frac{y^\top (I - A(\lambda))y}{\det^+(I - A(\lambda))^{\frac{1}{r}}},$$

where  $\det^+$  denotes the product of positive eigenvalues of a matrix, and  $r$  is the rank of  $I - A(\lambda)$ .

A recent comparative study of  $C_p$  and GML is presented in [27], where emphasis is put on the finite-sample nonasymptotic behavior of the two methods for smoothing scatter data problems.

## 1.4 Contributions in the thesis

In the **first part** of the thesis (Chapters 2–4), we study regularization methods in the context of the total least squares problem formulation. Truncation and penalty-type methods are discussed, and model selection techniques are adapted to the regularized errors-in-variables regression.

The contributions on these topics are:

1. recasting the truncated total least squares problem as a truncated *core problem*, where the term of “core problem,” originating in [95], refers to a reduced, but essential part of a linear system, which can be computed from the SVD or another orthogonal decomposition, such as bidiagonalization;
2. extensions of the core problem concept and of the computational techniques to multiple right-hand sides problems;
3. complete survey of the regularized total least squares problem, with special focus on numerical optimization methods; an original method based on solving iteratively quadratic eigenvalue problems is explained into detail, but more recently published algorithms for the regularized total least squares problem are also presented;
4. development of a consistent cross validation methodology for ill-posed errors-in-variables models;
5. analysis of methods for choosing the truncation level in the truncated total least squares framework and choosing the regularization parameter in the regularized total least squares problem, by adapting several classical methods for model selection.

The **second part** of the thesis is devoted to modeling data that is intrinsically nonlinear. We explore nonparametric and semiparametric modeling, as well as related optimization methods from the nonlinear least squares family. In this field, ill-posedness appears

in the form of having to decide a trade-off between a good fit of the data and some model requirements, such as parsimony or (in the case of curve fitting) smoothness. A biomedical application, namely the *quantification of metabolite concentrations from short echo-time in vivo nuclear magnetic resonance spectroscopy (MRS) signals*, is described. This application has been the motivation for the theory developed in the whole second part of the thesis, since it gave rise to an interesting model formulation, as well as to challenging implementation situations.

Below, we list some of the contributions within this part.

6. Theoretical formulation of a general *template* spline family that encompasses classical spline families, such as smoothing splines, regression splines and penalized splines;
7. statistical and computational analysis of a semiparametric modeling problem, where the parametric part of the model is a given nonlinear function and the additive nonparametric part is modeled with template (or penalized) splines;
8. outline of an algorithm for regularized semiparametric regression, incorporating a generalized cross validation choice of the regularization parameter that controls the importance of the nonparametric part;
9. application of the semiparametric modeling theory to the metabolite quantification problem;
10. implementation of the AQSES<sup>1</sup> software package for the metabolite quantification problem and presentation of results with AQSES on simulations and real data;
11. a theoretical analysis of constrained variable projection optimization for separable nonlinear least squares, with particular emphasis on the model formulation and the specific inequality bound constraints appearing in the AQSES implementation.

## 1.5 Chapter-by-chapter overview

### Chapter 2

We study truncation methods for the regularization of linear discrete ill-posed problems. Among the linear models that are typically obtained in discrete ill-posed problems, we focus on the following: the classical regression model  $A^{\text{exact}}x \approx b^{\text{noisy}}$  and the errors-in-variables model  $A^{\text{noisy}}x \approx b^{\text{noisy}}$ ; their multiple right-hand sides (RHS) counterparts  $A^{\text{exact}}X \approx B^{\text{noisy}}$  and  $A^{\text{noisy}}X \approx B^{\text{noisy}}$ ; the nullspace representation  $C^{\text{noisy}}Y \approx 0$ . Truncation is one of the simplest amongst the methods of regularization that can be used for approximating discrete ill-posed problems [58]. However, a better understanding of truncation methods (such as truncated singular value decomposition (TSVD) and truncated total least squares (TTLS)) is possible in view of the recent results on *core problems* of linear systems [95]. The core reduction of an incompatible linear system is a tool that is able to avoid

<sup>1</sup>AQSES stands for “accurate quantification of short echo-time MRS signals.” It is a method implemented as a FORTRAN 77 module of the AQSES GUI software package developed within BioMed at ESAT, K.U. Leuven.

nonunique and nongeneric solutions of the total least squares problem (and variations). We propose the use of *truncated core problems* in order to avoid close-to-nongenericity in ill-posed linear approximation problems.

We deviate for a while from the regularization path with the goal of generalizing the core problem formulations to multidimensional (*i.e.*, multiple right-hand sides) problems. In the single right-hand side case, a core matrix has the size  $(p + 1) \times p$ , where  $p$  is the number of distinct singular values of the matrix  $A$  corresponding to left singular subspaces that are not orthogonal to the right-hand side vector; the core matrix has as singular values a subset of the distinct nonzero singular values of  $A$ . In the multiple right-hand sides case (with, say,  $d$  right-hand sides), the core matrix has size  $(p + d) \times p$ , and exhibits similar properties. The core subproblem does not suffer from nonuniqueness or nongenericity issues.

For the single right-hand side case, practical computation of a core problem can be done with (direct or Lanczos) bidiagonalization; for multiple right-hand sides, we give a detailed *band Lanczos* diagonalization method.

Finally, we discuss the computation of solutions for the original approximation problem, using the truncated core problem, and we present a Matlab software package for computing core reductions, while monitoring a truncation level.

### Chapter 3

We investigate penalty regularization in the context of the total least squares problem. We focus on two formulations: one is a quadratically constrained total least squares problem, and the second is a quadratically penalized total least squares problem. We refer to them as regularized total least squares (RTLTS). As opposed to the classical regularization methods in the least squares context, the formulations for RTLTS do not have closed-form solutions. Therefore, iterative optimization methods are needed to tackle them. Several computational approaches for solving RTLTS are surveyed. We start with the original RTLTS algorithm of Golub, Hansen and O’Leary [42], we continue with our own iterative quadratic eigenvalue problem solver [117], which we analyze in more detail. Then, we give an overview of more recent algorithms from the literature.

### Chapter 4

We focus on techniques for regularization parameter selection for ill-posed problems in the context of linear errors-in-variables models. There is an important difference between regularization for ordinary linear regression models and linear errors-in-variables models. For the former, a valid error measure is the *prediction error*, *i.e.*, the residual norm between the vector of given noisy regression outputs  $b^{\text{noisy}}$  and its predicted counterpart, given a certain regularization scheme,  $b^{\text{reg}}$ . For errors-in-variables models, this error measure is not appropriate, because the  $A$  data matrix is also noisy, and it is *corrected* by the regularized regression method. Therefore, a *generalization error* is defined to take into account corrections on both  $A$  and  $b$ . This generalization error measure can then be used in various model selection techniques and for various regularization methods (truncation, penalty-based) for ill-posed errors-in-variables models.

## Chapter 5

We begin by studying the role of regularization in the context of curve fitting of nonlinear data, with the problem of nonparametric modeling. In this context, regularization is generally synonym to *smoothing*. A common frame of *template splines* that unifies the definitions of various spline families, such as smoothing splines, regression splines or penalized splines, is introduced. This extension allows an easy incorporation of additional constraints apart from smoothness, such as symmetries, monotonicity, convexity, which is generally not possible in the context of classical spline families.

The nonlinear nonparametric regression problem that defines the template splines can be reduced, for a large class of Hilbert spaces, to a parameterized regularized linear least squares problem, which leads to an important computational advantage.

Good statistical properties that hold for smoothing splines, such as the optimality of model selection via generalized cross validation, still hold for the template spline extension.

## Chapter 6

In this chapter, we formulate and solve a semiparametric fitting problem with regularization constraints. The model that we focus on is composed of a parametric nonlinear part and a nonparametric part that can be reconstructed using template splines. Regularization is employed in order to impose additional properties, such as a certain degree of smoothness, on the nonparametric part.

Semiparametric regression is presented in this chapter as a generalization of nonlinear regression, and all important differences that arise from the statistical and computational points of view are highlighted. As in nonlinear regression, we can infer asymptotic properties of the semiparametric regression estimates. Under Gaussian noise assumption, normality of the estimates is recovered; however, because of the regularization term, the computed parameters will be biased from the “true” values. We develop detailed bias and covariance formulas, which allow derivation of other statistically relevant information, such as confidence intervals.

We give an algorithmic outline of regularized semiparametric regression, with emphasis on efficient computation. One of the main issues in this context is the choice of the *regularization parameter* that controls the trade-off between nonlinear misfit minimization and effective regularization. We propose an automated iterative selection method that is based on the classical generalized cross validation criterion. The method is data-driven and does not need prior estimates for the noise statistics.

## Chapter 7

In this chapter, we pursue the problem of quantifying metabolite concentrations from short echo-time *in vivo* magnetic resonance spectroscopic (MRS) measurements. The goal of this application is to compute the parameters of a certain model function, which give information about the concentrations of chemical substances in a region of the brain. Along with the contributions of the most relevant metabolites in the brain, the MRS signal also contains a *macromolecular baseline* – for which no model function is available – that must be taken into account in automated quantification methods. For this reason, the semipara-

metric modeling framework developed in Chapter 6 is employed.

We discuss some general background on the MRS quantification problem, we introduce its mathematical formulation, and then we devote some pages on describing its software implementation in the AQSES module, and the results obtained using this software.

## Chapter 8

More computational aspects of the AQSES implementation and comparisons between several of its variants are included in this last chapter.

One important optimization tool implemented in the AQSES software is a specialized variable projection (VARPRO) algorithm for solving separable nonlinear least squares problems. The standard VARPRO implementation (described in [47, 48]) is designed for unconstrained separable nonlinear least squares. The VARPRO implementation in AQSES differs from the standard VARPRO because it allows imposing some constraints on the nonlinear, as well as on the linear model parameters. If only the nonlinear variables are subject to constraints, then the Gauss-Newton or Levenberg-Marquardt minimization algorithm on which VARPRO is classically based should be replaced with a version that can incorporate those constraints. If some of the linear variables are also constrained, then they cannot be projected out in closed-form expression as for the classical VARPRO technique. We show how quadratic programming problems can be solved instead, and we provide details on efficient function and approximate Jacobian evaluations for the inequality bound constrained VARPRO method.

In terms of the AQSES method, we present two versions: one uses a VARPRO algorithm with lower and upper bounds on the nonlinear variables of the MRS parametric model part, and the second has extra nonnegativity constraints on some of the linear parameters. These two versions have a physical motivation in terms of the metabolite signals: they correspond to, on one hand, having *non-equal phases* for all metabolite signals in the MRS model, and, on the other hand, imposing *equal phases* for all the metabolites.



## **Part I**

# **Regularization for linear problems**



## Chapter 2

# Truncation methods for core linear systems

In this chapter, we study one of the simplest types of regularization methods that can be applied to linear discrete ill-posed problems: the truncation methods. These techniques are based on SVD-type expansions of the linear system's solution, and can be applied to a range of estimation methods belonging to the least squares and total least squares families.

Our study views truncation methods as reductions to some essential information about the system; from this perspective, we link truncation to the concept of *core problems* in linear algebraic systems. The chapter also includes extensions of the core reduction to linear approximation problems with multiple right-hand sides.

## 2.1 Introduction

As discussed in the introduction, ill-posed problems are problems where the solution does not depend continuously on the input data, where arbitrarily small perturbations in the input data produce arbitrarily large changes in the solution. Among the linear models that are typically obtained from, *e.g.*, discretizations and approximations of integral or differential models, we shall focus on the following:

- the classical regression model  $A^{\text{exact}}x \approx b^{\text{noisy}}$  and the errors-in-variables model  $A^{\text{noisy}}x \approx b^{\text{noisy}}$ ;
- their multiple right-hand sides (RHS) counterparts  $A^{\text{exact}}X \approx B^{\text{noisy}}$  and  $A^{\text{noisy}}X \approx B^{\text{noisy}}$ ;
- the nullspace representation  $C^{\text{noisy}}Y \approx 0$ .

The considered systems will be square or overdetermined. (We exclude the underdetermined systems because in that case extra information would be needed in order to discriminate a useful solution within the infinite set of possible solutions).

Extensive characterizations of discrete ill-posed problems of the type  $Ax \approx b$  exist in the literature (see the book [58] and the references therein). It is known, for instance, that the properties of such a system can be understood from the singular value decomposition of  $A$ . In an ill-posed system, there is no clear gap in the decay of the singular

values of  $A$ . This makes it difficult to decide the best value that can be used as numerical rank. Moreover, it usually happens that the singular vectors corresponding to smaller and smaller singular values have increasing complexity (meaning that they contain more and more features, oscillations). If the least squares or total least squares solution of  $Ax \approx b$  is computed, then these oscillations are emphasized in the solution by the noise in the data or by the effects of working in finite precision arithmetic.

Among the numerous methods of regularization that can be used for approaching discrete ill-posed problems, we revisit in this chapter one of the simplest: truncation. The aims of our study encompass:

- a better understanding of truncation methods (such as truncated singular value decomposition (TSVD) and truncated total least squares (TTLS)) in view of the recent results on *core* reductions of linear systems [95];
- extensions to multidimensional problems;
- analysis of methods for choosing the truncation level, by adapting several classical methods for model selection.

## 2.2 Truncation methods for linear ill-posed problems

### 2.2.1 Linear ill-posed problems

We illustrate with an example the fact that it is not advisable to use any of the classical estimation methods (least squares, total least squares) directly on an ill-posed linear problem.

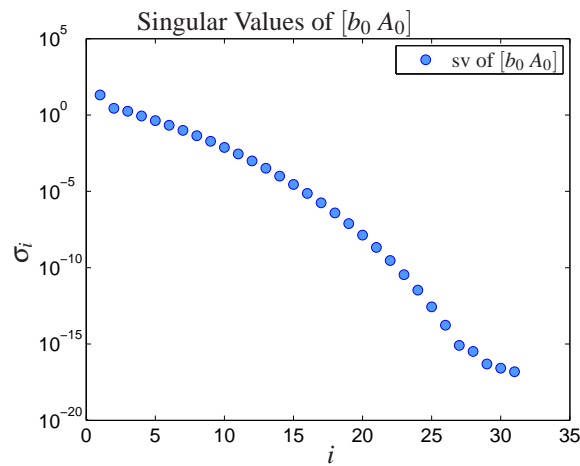
**Example 2.1** A matrix  $A_0$  and a vector  $b_0$  with real valued elements and dimensions  $m = 200$ ,  $n = 30$ , are simulated using the example `ilaplace` from the Regularization Tools [57]. This example that originally comes from [134] constructs  $A_0$  as a discretization of the inverse Laplace transform, using Gauss-Laguerre quadrature for the discretization of the inverse Laplace integral operator. In [134] and in Hansen's toolbox [57], the sampling points where the integral equation is evaluated were as many as the quadrature points, thus yielding a square discretized system. We slightly modify this setting in order to construct rectangular systems, by allowing more sampling points than quadrature points.

Figure 2.1 shows the singular values of  $[b_0 \ A_0]$ , as they decay without noticeable gap towards machine-precision zero.

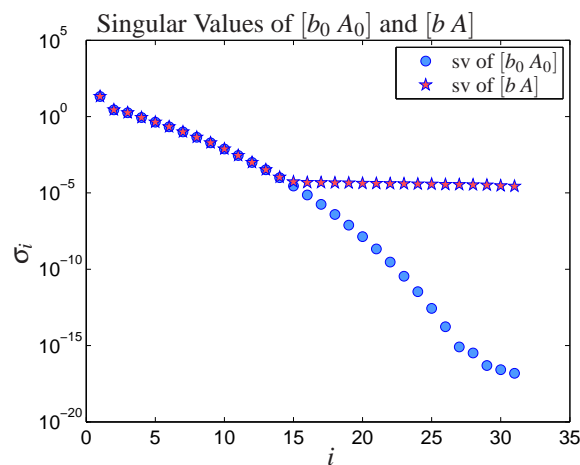
In this simulation,  $A_0x \approx b_0$  has a certain exact solution vector  $x_0$ , satisfying the equation  $A_0x_0 = b_0$ . We construct an incompatible noisy example  $Ax \approx b$ , by adding Gaussian white noise with a standard deviation of  $1e-5$  to all elements of  $A_0$  and  $b_0$ . For comparison, Figure 2.2 shows also the decaying singular values of the noisy  $[b \ A]$ , and it is noticeable that the smallest singular values stagnate at a certain level, determined by the standard deviation of the added noise.

The conditioning of the matrix  $A$  (about  $2e+5$ ) is much better than that of  $A_0$  (about  $3e+17$ ), but this is a misleading improvement. In fact, the added noise has a disastrous effect on the problem of estimating  $x$  using, *e.g.*, the TLS method.

Figure 2.3 shows the true solution  $x_0$  together with the TLS solution computed from the data  $[b \ A]$ .



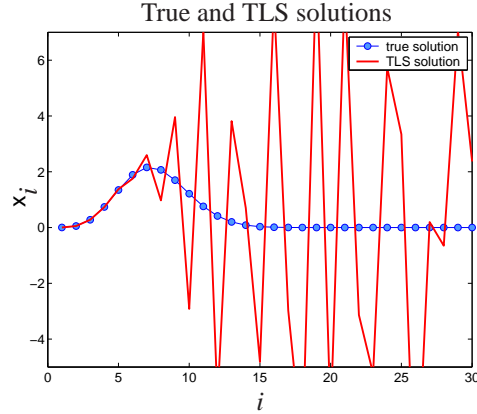
**Figure 2.1.** Singular values of  $[b_0 A_0]$  decaying without gap towards zero, where the data  $[b_0 A_0]$  comes from a discrete ill-posed problem.



**Figure 2.2.** Singular values of  $[b_0 A_0]$  together with the singular values of the noisy data  $[b A]$ .

### 2.2.2 Truncation methods for linear estimation

Truncation methods are an effective way of eliminating unwanted subspaces (such as the noise subspace) from given data sets, using the singular value decomposition [50] or other factorizations. For instance, if for a certain application the noise level is *a priori* known, then it is reasonable to eliminate all information that is below the noise level. It may happen that useful information from the problem is also embedded in that noise; this is an intuitive explanation why *truncation methods* might be too drastic, and thus they are not always



**Figure 2.3.** The true solution  $x_0$  is chosen to be the discretization of a smooth function. However, the TLS solution  $x$  computed from the noisy data  $[b \ A]$  is influenced by the noise and it is far from being a reasonable approximation of the underlying true solution  $x_0$ .

the most appropriate methods for some problem settings. The rich theory and practice of regularization has much more examples of techniques that are better suited for various situations.

Algorithms 2.1 and 2.2 outline the simple procedures of truncation in the LS and, respectively, TLS settings.

---

**Algorithm 2.1** Truncated SVD.

---

- 1: Compute the SVD of  $A = U'\Sigma'V'^\top$ , where  $U'$  is  $m \times n$ ,  $V'$  is  $n \times n$  and  $U'^\top U' = V'^\top V' = V'V'^\top = I_n$ , and  $\Sigma'$  is an  $n \times n$  diagonal matrix with the singular values  $\sigma'_1 \geq \sigma'_2 \geq \dots \geq \sigma'_n \geq 0$  on the diagonal;
- 2: Choose an appropriate truncation level  $k \leq n$ ;
- 3: Compute the truncated SVD solution

$$x_{\text{TSVD},k} = V'\Sigma'_k{}^\dagger U'^\top b, \quad \text{with } \Sigma'_k = \text{diag}\{\sigma'_1, \dots, \sigma'_k, \underbrace{0, \dots, 0}_{n-k}\}.$$


---

These methods are well-defined in the multiple right-hand side case, as well. We need only to replace  $b$  with the  $m \times d$  matrix  $B$ .

In the next example, we aim to illustrate that truncated TLS is a good regularization method for the type of problem described in Example 2.1.

**Example 2.2** We used the SVD of the noisy  $[b \ A]$  and we computed truncated TLS solution ( $x_{\text{TTLS},k}$ ) for all possible values of  $k$ , from 1 to  $n = 30$ . We plot in Figure 2.4 several reconstructed solutions, together with the exact  $x_0$ . Some facts are to be noted:

- the “nontruncated” solution  $x_{\text{TTLS},n}$  is identical to the TLS solution, and the truncated solutions with large  $k$  indices are also very much influenced by the noise and

**Algorithm 2.2** Truncated TLS.

- 1: Compute the SVD of  $\begin{bmatrix} b & A \end{bmatrix} = U\Sigma V^\top$ , where  $U$  is  $m \times (n+1)$ ,  $V$  is  $(n+1) \times (n+1)$  and  $U^\top U = V^\top V = VV^\top = I_{n+1}$ , and  $\Sigma$  is an  $(n+1) \times (n+1)$  diagonal matrix with the singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq \sigma_{n+1} \geq 0$  on the diagonal;
- 2: Choose an appropriate truncation level  $k \leq n$ .
- 3: Build  $\begin{bmatrix} b_k & A_k \end{bmatrix} := U\Sigma_k V^\top$ , where  $\Sigma_k$  is  $\Sigma$  with the smallest  $n-k$  singular values set to zero,  $\Sigma_k = \text{diag}\{\sigma_1, \dots, \sigma_k, \sigma_{k+1}, 0, \dots, 0\}$ ;
- 4: Solve in the TLS sense the truncated problem  $A_k x \approx b_k$  and obtain the truncated TLS solution  $x_{\text{TLS},k}$ .

resemble the TLS solution;

- the truncated solutions with very small  $k$  level are too simplistic to capture the characteristics of the true solution  $x_0$ ;
- there are several values of  $k$ , such as  $k = 10$  in Figure 2.4, for which the truncated TLS solution  $x_{\text{TLS},k}$  is a very good reconstruction of the true solution  $x_0$ .

The aim of regularization by truncation is thus to appropriately identify a good truncation level, and to construct a truncated solution that can capture the essential features of the unknown true solution, without explicit knowledge about the true solution, and even without *a priori* knowledge about the magnitude of the noise in the data.

## 2.3 Core reduction with embedded truncation

### 2.3.1 The scaled total least squares formulation

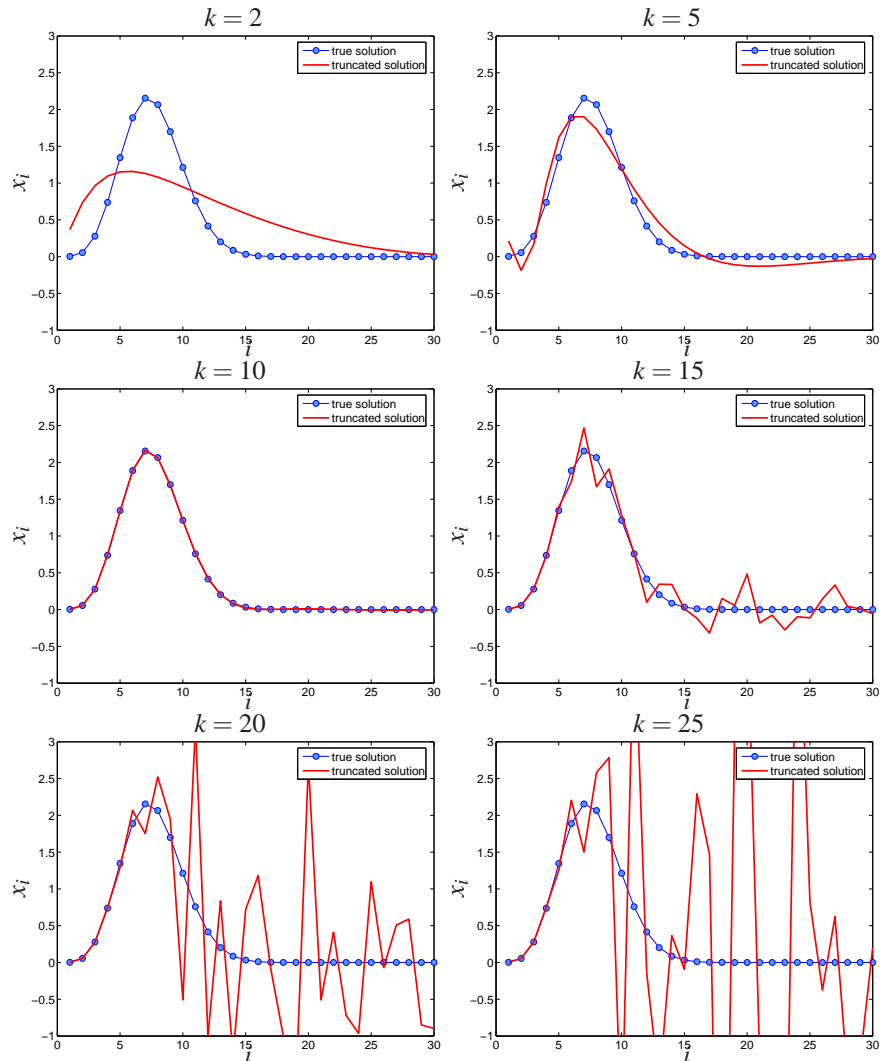
In [94, 95], the notion and properties of core problems in linear algebraic systems are introduced and thoroughly analyzed. The context is solving overdetermined nearly compatible linear systems of equations, written in matrix notation as  $Ax \approx b$ , where  $A \in \mathbb{R}^{m \times n}$  and  $b \in \mathbb{R}^m$  are given ( $m > n$ ), and  $x$  is an unknown vector. In [94, 95], the classical methods of least squares, total least squares and data least squares (DLS) are treated in a unified manner, under the more general *scaled total least squares (ScTLS)* formulation. The ScTLS solution is the optimal  $x$  that solves the minimization

$$\min_{x, \Delta A, \Delta b} \left\| \begin{bmatrix} \Delta b & \Delta A \end{bmatrix} \right\|_F^2, \quad \text{subject to } (A + \Delta A)x\gamma = \gamma b + \Delta b. \quad (2.1)$$

Here,  $\gamma$  is a positive scaling parameter that allows the corrections on the matrix  $A$  to have a different magnitude compared to the corrections on  $b$ . The extreme cases when  $\gamma$  goes to zero or to infinity correspond to the least squares and data least squares problems (*i.e.* corrections *only* on  $b$  or *only* on  $A$ , respectively), while the particular choice  $\gamma = 1$  gives the total least squares problem [132].

#### Unique and minimum norm solution

For finite and nonzero  $\gamma$ , problem (2.1) becomes the nearest approximation of a matrix with a lower rank matrix, and can be solved using the SVD of the matrix  $\begin{bmatrix} \gamma b & A \end{bmatrix}$ . In the



**Figure 2.4.** Several truncated TLS solutions  $x_k$ , together with the true solution  $x_0$ , for the simulation data described in Example 2.1.

usual (generic) case, only the singular vector corresponding to the smallest nonzero singular value is needed, if this last singular value is unique. If the smallest nonzero singular value of  $[\gamma b \ A]$  is multiple, then the minimization (2.1) does not have a unique solution either; then, a convenient and conventional choice is to pick from the corresponding singular subspace a vector that will give the *minimum norm* solution for  $x$  [132].

### Nongeneric and close-to-nongeneric cases

Another cumbersome case happens when problem (2.1) does not have a finite optimal solution at all. Examples can be found in [95] or [132], where the corrections  $\Delta A$  and  $\Delta b$  can get infinitesimally small in magnitude, while the corresponding solution  $x$  has some elements that go to infinity. Such a situation is possible when, for instance, the right-hand side vector  $b$  is orthogonal to the subspace corresponding to the smallest nonzero singular value of  $A$ . This pathological case is a nongeneric case (*i.e.*, it appears in random data with zero probability). However, the interest in these cases is not only theoretical; in realistic ill-posed problems we encounter *close-to-nongeneric* problems, which need specialized methods. The understanding of the nongeneric case provides the background for solving close-to-nongeneric practical problems.

The core problem advertised in [95] avoids the nonuniqueness and nongenericity issues, by transforming the original data in  $A$  and  $b$  to reduced versions  $A_{11}$  and  $b_1$ , which can be smaller in size, but are essential for the approximation problem at hand. All redundant and not useful information that was present in the data  $A$  and  $b$  is eliminated through the reduction to  $A_{11}$  and  $b_1$ .

### 2.3.2 Short description of the core problem within $Ax \approx b$

The suggestion of Paige and Strakoš [95] is to find orthogonal  $P$  and  $Q$  such that

$$P^\top [b \quad AQ] = \left[ \begin{array}{c|c|c} b_1 & A_{11} & 0 \\ \hline 0 & 0 & A_{22} \end{array} \right] \quad (2.2)$$

and  $[b_1 \quad A_{11}]$  has minimal dimension; then, to solve  $A_{11}x_1 \approx b_1$  with the appropriate ScTLS algorithm and to set the final solution as

$$x = Q \begin{bmatrix} x_1 \\ 0 \end{bmatrix},$$

where the zero part comes from setting to zero the solution of the system  $A_{22}x_2 \approx 0$ .

### SVD form of the core decomposition

A core problem  $A_{11}x_1 \approx b_1$  can be obtained from the SVD of  $A = U'\Sigma'V'^\top$ . Indeed, using Householder rotations and some permutations, it is possible to transform

$$U'^\top [b \quad AV'] \quad \text{into} \quad \left[ \begin{array}{c|c|c} c & \Sigma'_1 & 0 \\ \hline \delta & 0 & 0 \\ \hline 0 & 0 & \Sigma'_2 \\ \hline 0 & 0 & 0 \end{array} \right] \begin{array}{l} p \\ 1 \\ n-p \\ m-n-1 \end{array}$$

where all elements of  $c$  are nonzero,  $\Sigma'_1$  contains only distinct nonzero singular values (say  $p$ ), while  $\Sigma'_2$  contains the other  $n-p$  singular values, including possible multiples and all the zero singular values, if any. The scalar  $\delta$  is zero when  $Ax = b$  is compatible and nonzero otherwise.

### Bidiagonal form of the core decomposition

For computational efficiency reasons, a bidiagonal reduction can be used instead of the SVD, *i.e.*,

$$P_1^\top [ b \quad A Q_1 ] = [ b_1 \quad A_{11} ] = \left[ \begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & \cdot & \cdot & \cdot & \\ & & & \beta_p & \alpha_p \\ & & & & (\beta_{p+1}) \end{array} \right]$$

where  $\alpha_i \beta_i \neq 0$  and  $\beta_{p+1}$  is zero only for compatible systems.

We can use direct bidiagonalization (with Householder rotations) for small to moderate problem dimensions, or the Lanczos bidiagonalization, suitable for large scale problems [45]. Note that  $A_{22}$  need not be bidiagonalized. Thus, the bidiagonalization algorithm is run until a (numerically) zero  $\alpha_i$  or  $\beta_i$  is reached.

Paige and Strakoš [95] proved that in exact arithmetic the bidiagonal reduction leads to a minimally dimensioned  $A_{11}$ , where  $A_{11}$  has only distinct and nonzero singular values. Moreover, solving the reduced bidiagonal problem  $A_{11} x_1 \approx b_1$  with the ScTLS algorithm and transforming back to the full solution  $x = Q_1 x_1$  actually gives the ScTLS solution (or the minimum norm ScTLS solution in cases of nonuniqueness, or the typical solution to a constrained ScTLS problem in nongeneric cases) of  $Ax \approx b$ , thus, the same solution that is proposed in [132] can be obtained in a possibly more efficient manner.

As mentioned, in well-posed problems, the bidiagonalization process should stop when a zero value is encountered, either on the main diagonal (corresponding to a compatible linear system), or on the superdiagonal (incompatible system). This stopping criterion is not suitable in the ill-posed problems case, where an exact zero on the (super)diagonal will never be possible, and thus the bidiagonalization would run up to the end. In Chapter 4, we analyze several different rules for deciding when to stop the bidiagonalization of  $A$ . In all cases, we express the truncation rule as a minimization problem over the truncation level  $k$  (where  $k$  is the number of columns of the bidiagonally reduced  $A_{11}$ ), and, moreover, we aim at being able to evaluate the value of these criteria using only the partial bidiagonal reduction obtained after  $k$  steps. We summarize the whole procedure of using the core reduction for truncation in Algorithm 2.3.

---

**Algorithm 2.3** Algorithm for reduction to a core problem, while monitoring a truncation criterion

---

- 1:  $k = 0$
  - 2: **repeat**
  - 3:    $k = k + 1$
  - 4:   compute the  $k^{\text{th}}$  bidiagonalization step of  $[ \gamma b \quad A ]$  using either the Householder or the Lanczos bidiagonalization method
  - 5:   compute value of truncation criterion at  $k$
  - 6: **until** ( $k = n + 1$ ) or truncation criterion cost function starts increasing
- 

**Remark 6** It is interesting to note that the bidiagonalization of  $[ b \quad A ]$ , followed by truncation, was proposed also in [33], a paper that studied the truncated total least squares

problem. At that time, the link with the core formulation and the special properties of this reduction in general were not yet put forward.

**Remark 7** Another remark concerning bidiagonalization and truncation is related to the partial least squares (PLS) method, which is quite popular in application areas such as chemometrics and chemical engineering. PLS is an alternative to principal component regression (= truncated SVD) that projects observed data onto a lower dimensional subspace; it is designed to handle multicollinearities. One computational algorithm designed for PLS is in fact equivalent to the Lanczos bidiagonalization method, as discussed in [29]. The properties of PLS can thus be explained in view of the fact that the bidiagonalization reduction of the data leads to a core problem.

The motivation of stopping prematurely the bidiagonalization process (as opposed to stopping when a zero diagonal or superdiagonal element is encountered, as it happens in the reduction to a core problem) is explained by the following fact. Bidiagonalization of a matrix can be used in large scale problems in order to obtain a lower rank approximation of the original matrix. If one stops the bidiagonalization process at step  $k$ , then a suboptimal lower rank approximation of rank  $k$  can be computed from this partial bidiagonalization. As well known, the best rank  $k$  approximation of a matrix in Frobenius norm or 2-norm can be expressed in terms of its SVD (the Eckart-Young-Mirsky theorem [50, 132]); ideally, the *partial SVD* involving the largest  $k$  singular values and associated singular vectors are the only required elements that can set up the best rank  $k$  approximation.

In the family of direct methods, an algorithm for partial SVD [130] was developed in order to compute efficiently a group of extreme singular values/vectors. However, this method needs to compute first the entire matrix bidiagonalization and it spares computations only at the stage of reducing the bidiagonal matrix to diagonal.

In the family of iterative methods, the Arnoldi/Lanczos methods based on Krylov subspaces are used in the last decades for computing a few (dominant) singular values of large (sparse or structured) matrices. The Lanczos bidiagonalization is a central scheme in this field. The singular values of the partial bidiagonally reduced matrix ‘converge’ to the dominant singular values of the original matrix. However, the computation of singular vectors is generally not accurate in finite precision arithmetic, and needs to be stabilized using reorthogonalization methods.

In *exact* arithmetic, the direct bidiagonalization method based on Householder transformations and the Lanczos bidiagonalization algorithm are equivalent, as they both give identical (partial) decompositions [45]. More precisely, the reduction of  $\begin{bmatrix} b & A \end{bmatrix}$  using Householder reflections corresponds to the Lanczos bidiagonalization of  $A$  with starting vector  $b$ .

In [118], *a priori* and *a posteriori* approximation error bounds are presented for the lower rank approximation computed using the Lanczos bidiagonalization in finite precision, assuming that a reasonable reorthogonalization procedure is also applied.

### 2.3.3 Extension to multiple right-hand sides and to the nullspace formulation

The ScTLS problem can be extended to the multiple right-hand sides problem  $AX \approx B$  (where  $A$  is  $m \times n$ ,  $X$  is  $n \times d$ ,  $B$  is  $m \times d$ , and, for convenience,  $m > n + d$ ), and can be used

in cases when one knows that the corrections on  $B$  and on  $A$  should be scaled at a given ratio with respect to each other:

$$\min_{x, \Delta A, \Delta B} \left\| \begin{bmatrix} \Delta B & \Delta A \end{bmatrix} \right\|_F^2, \quad \text{subject to } (A + \Delta A)x = \gamma B + \Delta B. \quad (2.3)$$

This type of problems can be solved using direct (linear algebraic) methods, such as the SVD. One of the most general TLS-related formulations that can still be solved with SVD-type computations is the *generalized TLS* [131]. It allows the corrections within each row of  $\begin{bmatrix} B & A \end{bmatrix}$  to be differently scaled, but requires that the error on each row is independent from and identically distributed as the error on the other rows; this corresponds to a general (positive semidefinite) covariance matrix – known up to a constant factor – for the errors in each row of  $\begin{bmatrix} B & A \end{bmatrix}$ . Other more general scalings of the corrections require optimization-based methods instead of numerical linear algebra techniques. See, for instance, the methods proposed for element-wise weighted total least squares [85, Chapter 3].

We consider also the (unscaled) approximation problem in nullspace formulation  $CY \approx 0$ , with  $C \in \mathbb{R}^{m \times (n+d)}$  and  $Y \in \mathbb{R}^{(n+d) \times d}$ . Both approximation problems  $AX \approx B$  and  $CY \approx 0$  can benefit from a core reduction that uses partial bidiagonalization, as well. Note that  $AX \approx B$  is a special case of  $CY \approx 0$ , with  $C = \begin{bmatrix} B & A \end{bmatrix}$  and the constraint that  $Y$  has as leading  $d \times d$  block a nonsingular matrix. The nullspace formulation  $CY \approx 0$  is more general, since this constraint is not imposed, but it is necessary to impose a nontriviality condition on  $Y$ , such as the constraint that  $Y$  is full column rank. As a way of estimating  $Y$ , we use the TLS criterion of minimizing  $\|\Delta C\|_F^2$  subject to the constraint that the corrected system is consistent,  $(C + \Delta C)Y = 0$ , and an extra nontriviality constraint on  $Y$ ; for the latter, we set  $Y^\top Y = I_d$ , which does not restrict the generality.

Obviously, the SVD is a possible solution to the nullspace formulation problem, since the problem amounts to finding the nearest low rank approximation of  $C$  (with a rank reduction of at least  $d$ ) and choosing a basis for the nullspace of  $C + \Delta C$  in order to compute  $Y$ .

Using the SVD of  $\begin{bmatrix} B & A \end{bmatrix}$ , or  $C$ , respectively, the optimal corrections  $\begin{bmatrix} \Delta B & \Delta A \end{bmatrix}$ , respectively,  $\Delta C$ , are obtained from the smallest  $d$  singular values and corresponding singular subspaces in these decompositions.

### Unique and minimum norm solution

For both the multiple right-hand sides and the nullspace formulations, an intrinsic source of multiple optimal solutions appears whenever the  $d + 1^{\text{st}}$  smallest singular value  $\sigma_n$  is *multiple* (i.e.,  $\sigma_n = \sigma_{n+1}$ ). In this case, the optimal corrections, as well as the optimal solution  $X$  (or  $Y$ ) are nonunique.

To distinguish between all  $X$  solutions that give the same minimum residual, the *minimum Frobenius norm* solution for  $X$  is typically chosen in the TLS literature [132]. However, the nice closed-form expression for the minimum norm TLS solution  $\hat{X} = (A^\top A - \sigma_{n+1}^2 I)^{-1} A^\top B$  only holds true when the last  $d$  singular values of  $\begin{bmatrix} B & A \end{bmatrix}$  coincide:  $\sigma_{n+1} = \dots = \sigma_{n+d}$  (see the discussion in [132, §3.3.2]).

Note also that  $Y$  is in any case nonunique, since every rotated matrix  $YS$ , with  $S$  orthogonal, is also an optimal solution for any optimal  $Y$ .

### Nongeneric and close-to-nongeneric cases

The nongeneric and close-to-nongeneric cases occur for the  $AX \approx B$  formulation when the singular subspace corresponding to the  $d^{\text{th}}$  smallest singular value ( $\sigma_{n+1}$ ) is orthogonal (or nearly orthogonal) to the columns of the  $B$  matrix.

The nongenericity concept does not exist for the  $CY \approx 0$  formulation, since the source of this type of problems is precisely the partitioning of the data matrix  $C$  into  $A$  and  $B$ .

In the following, we assume that the problem  $AX \approx B$  is such that  $B$  has full column rank.

### Core problem within $AX \approx B$

The material in the rest of Section 2.3.3 contains very recent developments; some work is still in progress.

For the multiple right-hand sides  $AX \approx B$  problem, we consider transformations of the form:

$$P^T [B \quad AQ] = \left[ \begin{array}{c|c|c} B_1 & A_{11} & 0 \\ \hline 0 & 0 & A_{22} \end{array} \right], \quad (2.4)$$

with  $P$  and  $Q$  orthogonal matrices and  $[B_1 \quad A_{11}]$  of minimal dimension. Then, the problem  $AX \approx B$  becomes separable:

$$A_{11}X_1 \approx B_1 \quad \text{and} \quad A_{22}X_2 \approx 0,$$

where the original  $X$  is recovered as  $X = Q [X_1^T \quad X_2^T]^T$ . The second system has as reasonable solution the trivial solution  $X_2 = 0$ . This choice corresponds to a minimum (Frobenius and 2-) norm  $X$  solution.

We argue that a minimally dimensioned  $A_{11}$  has, in the case when  $AX \approx B$  is incompatible, the size  $(p+d) \times p$ , where  $p$  denotes the number of distinct nonzero singular values of  $A$  corresponding to singular subspaces on which  $B$  is not orthogonal. To this end, we construct a reduction of the form (2.4) based on the SVD of  $A$ . Let the full SVD of  $A$  be  $U'\Sigma'V'^T$ , where  $U'$  is  $m \times m$  orthogonal,  $V'$  is  $n \times n$  orthogonal, and  $\Sigma'$  is  $m \times n$  diagonal matrix. Applying  $U'^T$  from the left, we have:

$$U'^T [B \quad AV'] = [U'^T B \quad \Sigma'] =: [B' \quad \Sigma'].$$

Next,  $B'$  should be modified – using orthogonal transformations from the left – in a controlled way that introduces zeros, such that a separable decomposition as in (2.4) is obtained.

- The rectangular  $m \times n$  matrix  $\Sigma'$  ( $m > n$ ) has a zero block below the square diagonal block. The part of  $B'$  corresponding to this zero block can be reduced with a QR factorization:  $B_Q \begin{bmatrix} B_R \\ 0 \end{bmatrix}$ , where  $B_Q$  is an orthogonal matrix and  $B_R$  is an upper triangular  $d \times d$  matrix. The orthogonal matrix from this QR decomposition can be incorporated into the transformations from the left, by multiplying  $U'^T$  with  $\begin{bmatrix} I & 0 \\ 0 & B_Q \end{bmatrix}$ .
- The first column of  $B'$  is reduced (as in the one-dimensional case [95]) to a vector where only one nonzero element remains for each (of the, say  $p'$ ) distinct singular

values in  $\Sigma'$  (including the zero singular value, if  $A$  is rank-deficient). This can be accomplished with a series of Householder transformations on  $B'$  from the left, which will multiply the already existing transformation on the original data  $B$ ; these will be applied from the right to  $A$  as well, such that the diagonal structure of  $\Sigma'$  is kept. For convenience, a permutation of the rows can be performed, such that all nonzero elements corresponding to distinct nonzero singular values of  $A$  are grouped together in the first  $p'$  elements of the first column of the new  $B'$ .

- For the second column of  $B'$  (of the new  $B'$ , in fact), all but  $p' + 1$  elements can be zeroed, using a Householder rotation without influencing the first column. The price is that the second column will have a nonzero element in a new position. This position is chosen to correspond to a row of  $\Sigma'$  where a copy of a singular value of  $A$  resides (for example, a particular choice is the largest multiple singular value for which a nonzero element of the second column of  $B'$  exists).
- The reduction of the other columns of  $B'$  such that they have at most  $p' + i$  nonzero elements (where  $i$  is a column index) continues as long as there are still unprocessed columns in  $B'$  and there are still multiple singular values left in  $\Sigma'$ , which were not already singled out in previous steps.
- At the end, all completely *zero rows* among the first  $p' + d$  rows of  $B'$ , together with the corresponding rows and columns of  $\Sigma'$ , will be permuted to the bottom of the decomposition.

This procedure will compute, finally, a decomposition of the form

$$P^\top [B \quad AQ] = \left[ \begin{array}{c|c|c} B_T & \Sigma'_1 & 0 \\ B_R & 0 & 0 \\ \hline 0 & 0 & \Sigma'_2 \end{array} \right] =: \left[ \begin{array}{c|c|c} B_1 & A_{11} & 0 \\ \hline 0 & 0 & A_{22} \end{array} \right], \quad (2.5)$$

where  $B_T$  is upper trapezoidal with  $d$  columns and, say,  $p$  rows,  $B_R$  is  $d \times d$  upper triangular,  $\Sigma'_1$  contains nonzero singular values of  $A$  with multiplicities at most  $d$ , and  $\Sigma'_2$  contains the zero singular values and the remaining multiple singular values of  $A$ .

The following properties of the decomposition (2.5) are essential for showing that nonuniqueness and nongenericity issues are avoided when solving the core problem  $A_{11}X_1 \approx B_1$ .

**Proposition 2.3.**

1. In the decomposition (2.5),  $A_{11}$  does not have singular values with multiplicity bigger than  $d$ .
2. The number of rows  $p$  of  $B_T$  equals the number of singular subspaces of  $A$  corresponding to distinct singular values, on which  $B$  has nonzero projections.

**Proof.** The first point is obvious by construction. The second point is clear if we look at the particular SVD of  $A$  given in (2.5): the projection of  $B$  onto all singular subspaces corresponding to distinct nonzero singular values appears within  $B_T$  in (2.5). By construction, each row of  $B_T$  contains nonzero elements.  $\square$

As a corollary, we have that the core problem  $A_{11}X_1 \approx B_1$  has unique solution (since nonuniqueness happens when the smallest nonzero singular value has multiplicity larger than the number of right-hand sides). Moreover, the problem  $A_{11}X_1 \approx B_1$  is generic, because in nongeneric problems the right-hand side is orthogonal to the singular subspace corresponding to the smallest nonzero singular value.

### Banded approximations

The explicit partitioning in  $\begin{bmatrix} B & A \end{bmatrix}$  suggests that a *block version* of the Lanczos algorithm can be an appropriate tool; in such a method, the Lanczos process is done with  $d$  starting vectors (the  $d$  columns of  $B$ ) at a time. However, instead of a block-Lanczos, an application of band-Lanczos for the reduction to a core problem in the multiple right-hand sides case was proposed in recent talks by Åke Björck [12, 11]. We resketch the banded decomposition and the band-Lanczos method from there, but then we modify it and make it more rigorous. We discuss afterwards the properties of the obtained decomposition.

A band decomposition of  $\begin{bmatrix} B & A \end{bmatrix}$  has the shape

$$P^\top \begin{bmatrix} B & AQ \end{bmatrix} = \left[ \begin{array}{c|c} R & L \\ \hline 0 & 0 \end{array} \right], \quad (2.6)$$

where  $R$  is  $(\tilde{p} + d) \times d$  upper triangular, and  $L$  is  $(\tilde{p} + d) \times \tilde{p}$  banded lower triangular, for some  $\tilde{p} \leq n$ , as in:

$$\begin{bmatrix} R & | & L \end{bmatrix} = \left[ \begin{array}{ccc|ccc} * & * & * & & & \\ & * & * & * & & \\ & & * & * & * & \\ & & & * & * & * \\ & & & & * & * \\ & & & & & * \\ & & & & & & * \end{array} \right],$$

where the width of the band equals  $d + 1$ . In this decomposition, zero elements might appear at some point on the band of  $L$ ; it is desirable to take advantage of this in order to identify a core subproblem. Björck suggests how to use *deflation* steps whenever a zero appears on the outer diagonals of the band. In that case, the band can shrink with one, and the procedure can be repeated until the whole band dies out, leading to a core problem.

To be more rigorous, let us first show how the band decomposition can be obtained with a band Lanczos iterative method, in the spirit of [105]. Denote by  $u_1, \dots, u_n$  the first  $n$  columns of the matrix  $P$ , by  $v_1, \dots, v_n$  the columns of the matrix  $Q$ , and by  $g_k, h_k$  the  $(d + 1)$ -dimensional vectors<sup>2</sup> that are obtained at the intersection of the  $k^{\text{th}}$  row and, respectively, column of  $\begin{bmatrix} R & L \end{bmatrix}$  with the band of  $\begin{bmatrix} R & L \end{bmatrix}$ . Note that the vectors  $g_k, h_k$  are not independent, since they share the same elements on the band of  $\begin{bmatrix} R & L \end{bmatrix}$ . This is more

<sup>2</sup>For  $k$  smaller than  $d + 1$ ,  $h_k$  has length  $k$ . For  $k$  larger than  $n$ ,  $g_k$  has length  $n - k$ , thus smaller than  $d + 1$ . Note that in this paragraph we let  $\tilde{p} = n$ .





- Finally, from (2.10), we initialize  $\widehat{u}_{k+d_c}$ :

$$\widehat{u}_{k+d_c} \leftarrow Av_{k-i_c} - u_k L_{k,k-i_c}. \quad (2.13)$$

As promised, Algorithm 2.4 shows how deflation is used in order to avoid any divisions by zero that can occur in (2.12) or (2.13). Each of the two possible kinds of deflation steps decrease the bandwidth  $d_c$  by one, and the second one (for (2.13)) increases  $i_c$  by one, as well; this corresponds to cutting a last zero column from the matrix  $L$ .

---

**Algorithm 2.4** Band Lanczos reduction of  $A$  with exact deflation
 

---

- 1:  $k \leftarrow 0, d_c \leftarrow d, i_c \leftarrow 0$
  - 2:   2a: compute the QR factorization of  $B$ :  $B = Q_B R_B$   
       2b: initialize  $\widehat{u}_1, \dots, \widehat{u}_d$  with the first  $d$  columns of the orthogonal matrix  $Q_B$
  - 3: **repeat**
  - 4:    $k \leftarrow k + 1$
  - 5:   **if**  $\|\widehat{u}_k\| = 0$  **then** deflate:
    - 5a:   set  $L_{j,j-d_c-i_c} = 0$  for all  $j > k$
    - 5b:    $d_c \leftarrow d_c - 1$
    - 5c:   **for**  $j = k, \dots, k + d_c - 1$ , set  $\widehat{u}_j \leftarrow \widehat{u}_{j+1}$
    - 5d:   **go to** 5:
  - 6:   6a:  $L_{k,k-d_c-i_c} \leftarrow \|\widehat{u}_k\|$   
       6b:  $u_k \leftarrow \widehat{u}_k / L_{k,k-d_c}$   
       6c: **for**  $j = k + 1, \dots, k + d_c - 1$   
            $L_{k,j-d_c-i_c} \leftarrow u_k^\top \widehat{u}_j, \quad \widehat{u}_j \leftarrow \widehat{u}_j - u_k L_{k,j-d_c-i_c},$
  - 7:   **if**  $v_{k-i_c}$  not computed **then** set  $\widehat{v}_{k-i_c} \leftarrow A^\top u_k$   
       **else** **go to** 11:
  - 8:   **if**  $\|\widehat{v}_{k-i_c}\| = 0$  **then** deflate:
    - 8a:   set  $L_{j,j-i_c} = 0$  for all  $j > k$
    - 8b:    $i_c \leftarrow i_c + 1$  and  $d_c \leftarrow d_c - 1$
    - 8c:   **go to** 11:
  - 9:    $L_{k,k-i_c} \leftarrow \|\widehat{v}_{k-i_c}\|, \quad v_{k-i_c} \leftarrow \widehat{v}_{k-i_c} / L_{k,k-i_c}$
  - 10:    $\widehat{u}_{k+d_c} \leftarrow Av_{k-i_c} - u_k L_{k,k-i_c}$
  - 11: **until**  $k = n + 1$  or  $d_c = 0$ .
- 

In exact arithmetic, Algorithm 2.4 is equivalent to the band Lanczos reduction for square symmetric matrices [35] applied simultaneously to  $A^\top A$  and  $AA^\top$ , with starting vectors the columns of  $A^\top B$  and  $B$ , respectively, or one applied to the larger symmetric  $\begin{bmatrix} 0 & A \\ A^\top & 0 \end{bmatrix}$ , with starting block  $\begin{bmatrix} B \\ 0 \end{bmatrix}$ . For the symmetric version in [35], it is known that after  $d$  exact deflations have occurred, then, as in the case  $d = 1$ , the Lanczos vectors span an invariant subspace and all the eigenvalues of the banded matrix are also eigenvalues of the original matrix. In our case, this translates to the certainty that after  $d$  deflation steps occur, the singular values of the partial banded matrix  $L$  are a subset of the singular values of  $A$ , and the singular values of  $\begin{bmatrix} R & L \end{bmatrix}$  are a subset of the singular values of  $\begin{bmatrix} B & A \end{bmatrix}$ .

It is known that in exact arithmetic, a Krylov subspace generated by a single starting vector can provide only one copy of an eigenvalue. Finding multiplicities requires a block

Krylov subspace. Each deflation step in Algorithm 2.4 corresponds to the identification of a singular value. This means that when the algorithm terminates, the banded reduction has at most  $d$  singular values with multiplicity greater than 1. Denote the final number of columns of  $L$  by  $p$ . Then,  $P$  has the size  $(p + d) \times p$ . If the SVD of  $L$  is  $U_L \Sigma_L V_L^T$ , we conjecture that  $U_L^T R$  does not have any completely zero rows (still to be proven!). This means that in fact  $p$  is the number of singular values of  $A$  with multiplicity smaller than  $d + 1$  for which the corresponding left singular subspaces are not orthogonal onto  $B$ . This reason is an evidence that the band reduction is in fact a core decomposition for multiple right-hand sides problems.

### Bidiagonal approximations

Due to the fact that solving the approximation problems  $AX \approx B$  and  $CY \approx 0$  can be viewed as lower rank approximation problems, we believe that ‘standard’ (direct or Lanczos) bidiagonalization of  $\begin{bmatrix} B & A \end{bmatrix}$  and  $C$  can still be the key to efficient computations in the case of ill-posed problems. The reason is that there are normally no multiple singular values in discrete ill-posed problems (except maybe for the approximately equal singular values settled at noise level). A truncated bidiagonalization keeps the information corresponding to the dominant singular triplets, while the (noise-corrupted) subspaces corresponding to the smaller singular values are discarded.

#### 2.3.4 Computing optimal solutions and corrections from a core problem

We assume that a core reduction (or a truncated core reduction) is readily computed, *e.g.*, from an SVD-based method or from a Lanczos bi- (or banded) diagonalization method. For the single right-hand side problem  $Ax \approx b$ , this decomposition is given in (2.2); for the multiple right-hand side problem  $Ax \approx B$ , it is given in (2.4). In order to compute a ScTLS solution for the original problem, we first need to solve in the ScTLS sense the core problem. Then, it is straightforward to arrive at the minimum norm ScTLS solution of the original problem, projecting back the core solution to the original subspace.

In [95], the authors show how one can efficiently solve a core problem in bidiagonal form in each of the ScTLS special cases. For the multiple right-hand sides case, the essential computations for solving the core problem in banded form are the solution of a linear system or the SVD of a matrix with banded structure. Efficient and stable algorithms are available (see, *e.g.*, [46] for the SVD computation of a banded lower triangular matrix).

## 2.4 Implementation and numerical examples

The core reduction with embedded truncation and several of the methods for choosing the truncation level (see Chapter 4) are implemented in a modular package in Matlab, CoRe, for regularization using truncated core reductions. The CoRe tool can solve least squares, total least squares, data least squares or scaled total least squares, for a given data matrix  $A$  and a single right-hand side vector  $b$ , using the (partial) reduction to bidiagonal form. It is possible to choose between direct bidiagonalization (using Householder transformations) or Lanczos bidiagonalization (relying on the PROPACK package [72]).

Some special functionalities are listed here:

- All variants of the methods can be easily specified through the fields of a single “options” structure. Undefined fields will be assigned default values. For example, if no specific option is given, then the TLS problem will be solved, using direct bidiagonalization and the machine precision tolerance as stopping rule for the core reduction.
- It is possible to save a (partial) bidiagonalization and then to use it as input for continuing the bidiagonalization, or testing other stopping criteria, for solving the problem with another method, or for trying another scale parameter for ScTLS. The execution will then be faster, since the partial bidiagonalization will not be performed twice.

The CoRe tool is composed of the following functions:

1. `core`: main function that solves the specified ScTLS problem using the required bidiagonalization method and the required stopping criterion of truncation.
2. `updatebidiag`: function that computes or updates a matrix bidiagonalization, either by the direct or the Lanczos method.
3. `checktrunc`: function that checks whether the given bidiagonal matrix satisfies a certain specified stopping criterion; the implemented stopping criteria are: fixed tolerance level; fixed truncation level  $k$ ; generalized cross validation; generalized Akaike information criterion; rotated L-curve.
4. `residnorm`: function that computes the residual norm, as well as the total number and the number of effective parameters in an LS, TLS, DLS or ScTLS problem.
5. `solvestls`: function that solves an LS, TLS, DLS, or ScTLS problem with a (small) bidiagonal data matrix.

We tested the implementation on simulation examples, using either well-conditioned problems, or ill-posed problems created with the Regularization Tools [57].

Here are a couple of simple examples for solving well-posed problems with LS or TLS methods. First, the bidiagonalization was performed until the end was reached or until the magnitude of some (super)diagonal element became smaller than machine precision.

The following comparisons between the solution obtained using the implemented bidiagonalization for least squares and total least squares and the classical solvers based on the QR factorization and the SVD, respectively, show that CoRe tool is able to obtain perfect accuracy.

#### **Example 2.4 (Solving least squares problems with CoRe)**

```
% Define dimensions and generate random data
m = 100; n = 5;
A = rand(m,n);
b = rand(m,1);

% Find the LS estimate with Matlab's \
x_ls = A\b;
```

```

% Solve the LS problem using a core reduction
options.meth = 'ls';
options.bidiag = 'direct';
options.trunc = 'none';
options.tol = eps;

xc_ls = core(A,b,options);

[      x_ls          xc_ls      x_ls - xc_ls ]

    0.14070553791492    0.14070553791492    0.000000000000000
    0.07077648882190    0.07077648882190    0.000000000000000
    0.27719471950773    0.27719471950773   -0.000000000000000
    0.49415402722910    0.49415402722910             0
   -0.06076634290543   -0.06076634290543   -0.000000000000000

```

#### Example 2.5 (Solving TLS problems with CoRe)

```

% Solve the TLS problem with a function calling SVD
x_tls = tls(A,b);

% Solve the TLS problem using a core reduction
options.meth = 'tls';
options.bidiag = 'direct';
options.trunc = 'none';
options.tol = eps;

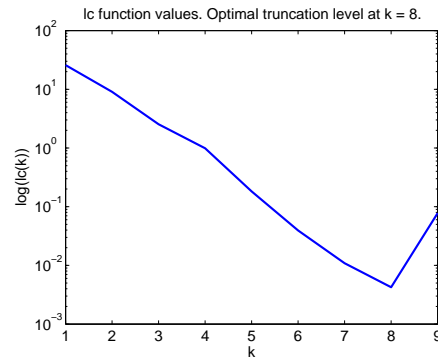
xc_tls = core(A,b,options);

[      x_tls          xc_tls      x_tls - xc_tls ]
-0.06043663155507   -0.06043663155507    0.000000000000000
-0.76899254307091   -0.76899254307091   -0.000000000000000
 0.70429406916636    0.70429406916636    0.000000000000000
 1.68598895989928    1.68598895989928   -0.000000000000000
-0.69587007183368   -0.69587007183368    0.000000000000000

```

For some simulated ill-posed problems such as the problem described in Example 2.1, we have also got good results. However, not all ill-posed problems provided in the Regularization Tools were solved satisfactorily using the truncation methods. We explain this by the fact that not all types of ill-posedness are removable by simple truncation, and more complicated regularization methods should be used in those cases. In general, the studied truncation methods favor solutions of small norms, which might not always be the most appropriate property.

Figure 2.6 shows the rotated L-curve obtained when applying CoRe with the TLS method to the problem in Example 2.1.



**Figure 2.6.** Plot of the rotated L-curve for the problem in Example 2.1. The minimum value of this criterion at  $k = 8$  gives a good reconstruction of the truncated solution compared to the true solution, cf. Figure 2.4.

## 2.5 Conclusions

We addressed the approximate linear modeling problems gathered under the roof of the scaled total least squares formulation, in the case when the data is coming from ill-posed problems. We used truncation methods as a form of regularization for the ScTLS problem. We discussed that a good first step in truncation methods is the partial reduction to core problem.

We extended the core problems to multiple right-hand sides systems. We showed that in the multiple right-hand sides case (with, say,  $d$  right-hand sides), the core matrix  $A_{11}$  has size  $(p+d) \times p$  and the core matrix  $A_{11}$  (and the extended core matrix  $[B_1 \ A_{11}]$ , respectively) has as singular values a subset of the distinct singular values of  $A$  (and of  $[B \ A]$ , respectively), plus some of their multiples (if any); none of the singular values of the core part has multiplicity bigger than  $d$ ; moreover, the singular values of  $A_{11}$  correspond to left singular subspaces of  $A$  that are not orthogonal to the right-hand-side  $B$ . We propose an SVD form and a banded form of the core system. We provide details of a *band Lanczos* diagonalization method.

Finally, we discuss the computation of solutions for the original approximation problem, using the truncated core problem, and we present a modular Matlab software package for computing core reductions, while monitoring a truncation level.

## Chapter 3

# Regularized total least squares

In this chapter we investigate penalty regularization in the context of the total least squares problem. The penalties or constraints that we are focusing on are based on weighted 2-norms of the solution vector, although other type of constraints can be envisaged.

As opposed to the classical regularization methods in the least squares context, the formulations in this chapter do not have closed-form solutions. Therefore, iterative optimization methods will be used to tackle them.

We consider two related problem formulations: the first one is a quadratically constrained total least squares problem, and the second is a penalized total least squares problem. We refer to any of them as regularized total least squares, since their common role is to introduce regularization in the context of total least squares estimation problems.

Although a closed form solution of the regularized total least squares problem does not exist and the iterative optimization methods that were originally proposed can only guarantee local optimality, recent analysis in [7, 6] suggests that both the quadratically constrained and the penalty-type formulations can be recast in a global optimization framework. In fact, scalar problems with unique solution are the key towards equivalent formulations for regularized total least squares (and other quadratically constrained fractional quadratic optimization problems).

## 3.1 Introduction

Throughout this chapter, the estimation of a solution  $x$  to problem  $Ax \approx b$  is considered (with  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $x \in \mathbb{R}^n$ ), assuming that  $A$  and  $[A \ b]$  are ill-conditioned or even rank-deficient with the singular values decreasing to zero without significant gaps. Regularization in the context of the TLS problem aims at decreasing the effect due to the intrinsic ill-conditioning of the problem and due to the noise in the data, and stabilizing the solution.

This chapter presents several computational approaches for solving the regularized total least squares (RTLS) problem. The first regularized total least squares problem formulation considered in this chapter consists in imposing a quadratic constraint on the solution vector  $x$  in the TLS problem (1.2) [42, 53]. In this manner, the elements of the solution vector can be bounded, or a certain degree of smoothness can be imposed on the solution.

This constrained problem cannot be solved with simple and elegant SVD-based methods, as the classical TLS problem does. Therefore, different computational approaches based on iterative methods are proposed. One of the methods, analyzed at large in our paper [117] is based on iteratively solving quadratic eigenvalue problems (QEP). The development of this new algorithm was inspired by the fact that quadratically constrained least squares (also named regularized least squares (RLS) or ridge regression) can be solved by a quadratic eigenvalue problem [40]. Due to the more complicated nature of the RTLS formulation, one QEP cannot solve the problem, but it is shown in this chapter that the solution can be approximated in a few iterations, each consisting in solving a QEP.

Two of the main advantages exhibited by this computational approach are its *robustness* with respect to the initialization of the iterative procedure and its *efficiency* in solving large problems. Experiments showed that the global minimum of the RTLS nonconvex optimization problem can be attained even when using random starting vectors. Quadratic eigenvalue problems equivalent to the RLS formulation [40] can be solved efficiently even for large problem sizes, as it was recently shown in [80]. In the RTLSQEP algorithm presented in this chapter, the same kind of QEPs appears at every iteration; therefore, the efficient solver described in [80] can also be used for RTLS. Moreover, it is not necessary to compute the whole spectrum of the QEP, since only one eigenpair is needed.

We present and discuss other methods proposed more recently in the literature for the same problem. Among these we mention in particular:

- the method of Renaut and Guo [103], which is based on an alternating iterative method with a scalar parameter and the solution vector  $x$ ;
- the ultimate algorithm for quadratically constrained TLS due to Beck, Ben-Tal and Teboulle [7], which only needs to optimize over one scalar variable and has very desirable optimization properties, such as unique attainable global solution, global convergence and fast convergence rate.

We also shortly discuss in Section 3.3 the Tikhonov penalty-type TLS formulation, and the method proposed in [6], in the same spirit as the above-mentioned method from [7].

We end the chapter with the numerical results from [117], based on examples from the *Regularization Tools* [57].

## 3.2 Quadratically constrained problem formulations

Regularization is often introduced by adding a quadratic constraint to the LS or TLS optimization problems [50, §12.1], [42]. In this section, the mathematical form of the problem under study is presented. As in [42], it is important to first stress the differences and the connections between the regularized least squares (RLS) and the regularized total least squares (RTLS) problems.

The RLS problem is defined as follows:

$$\min_x \|Ax - b\|_2^2 \quad \text{subject to } \|Lx\|_2^2 \leq \delta^2, \quad (3.1)$$

where  $L \in \mathbb{R}^{p \times n}$ ,  $p \leq n$ , and  $\delta > 0$ . For  $\delta > 0$  small enough (*i.e.*,  $\delta < \|Lx^{\text{LS}}\|_2$ ), an appropriate value of a parameter  $\lambda > 0$  can be fixed (depending in a nonlinear way on

$\delta$ ), such that the solution of (3.1) coincides with the one of the Tikhonov regularization problem (1.5) and of the normal system of equations

$$(A^\top A + \lambda L^\top L)x = A^\top b. \quad (3.2)$$

The RTLS problem statement is

$$\min_{x, \bar{A}, \bar{b}} \|[A \ b] - [\bar{A} \ \bar{b}]\|_F^2 \quad \text{subject to } \bar{A}x = \bar{b}, \|Lx\|_2^2 \leq \delta^2. \quad (3.3)$$

It is known that the TLS objective function (*i.e.*, the Frobenius norm of the extended correction matrix) can be replaced by the orthogonal distance  $\frac{\|Ax-b\|_2^2}{1+\|x\|_2^2}$  [132, §2.4.2]. Then, the previous formulation can be rewritten as

$$\min_x \frac{\|Ax-b\|_2^2}{1+\|x\|_2^2} \quad \text{subject to } \|Lx\|_2^2 \leq \delta^2. \quad (3.4)$$

**Remark 8** As also noted in [42] and [53], choosing the parameter  $\delta$  smaller than the value  $\|Lx^{\text{TLS}}\|_2$  implies that the quadratic constraint  $\|Lx\|_2 \leq \delta$  is *active* at the solution for optimization problems (3.3) or (3.4). For an ill-conditioned problem  $Ax \approx b$ , the norms  $\|Lx^{\text{LS}}\|_2$  and  $\|Lx^{\text{TLS}}\|_2$  are very big (therefore, the need for regularization); the assumption that  $\delta$  is small enough can be considered as guaranteed in practice.

In view of Remark 8, the regularization problems considered subsequently will be *equality constrained* problems. For further reference, their formulations are stated below:

$$\text{RLS problem:} \quad \min_x \|Ax-b\|_2^2 \quad \text{subject to } \|Lx\|_2^2 = \delta^2, \quad (3.5)$$

$$\text{RTLS problem:} \quad \min_x \frac{\|Ax-b\|_2^2}{1+\|x\|_2^2} \quad \text{subject to } \|Lx\|_2^2 = \delta^2. \quad (3.6)$$

The case when  $L$  is the identity matrix  $I_n$  is called the *standard case* and both RLS and RTLS yield the same solution if  $\delta$  is small enough ( $\delta < \|x^{\text{LS}}\|_2$  and  $\delta < \|x^{\text{TLS}}\|_2$ ), and the inequality constraints are replaced by equalities, as in (3.5) and (3.6). Indeed, in formulation (3.6) with  $L = I_n$ , the denominator  $1 + \|x\|_2^2$  may be replaced by the constant  $1 + \delta^2$  and a problem equivalent to the RLS formulation (3.5) is obtained.

In the *general case*,  $L$  may be rectangular and in practice it is usually chosen as an approximation of the first or second order derivative operators in order to impose a certain degree of smoothness on the solution. In this case, (3.5) and (3.6) are distinct problems.

In the following subsections, we describe in more detail the methods for RTLS, in the chronological order of their appearance.

### 3.2.1 RTLS method of Golub, Hansen and O'Leary: two parameters formulation

In [42], the first steps towards the analysis of the RTLS problem are made. In that paper, the authors are not only exploring computational solutions for the optimization problem that

they define, but they are also motivating the problem and discussing its relation to the classical regularized least squares problem. They show for which cases the RTLS and the RLS problems yield the same solutions and when are these solutions equal to the unconstrained ordinary least squares or total least squares solutions.

Concerning the computation of the RTLS solution, an equivalent two-scalar formulation is employed in [42]. This formulation comes from the first order optimality conditions on the problem (3.6):

$$(A^\top A + \lambda_I I_n + \lambda_L L^\top L)x = A^\top b, \quad (3.7)$$

where

$$\lambda_I = -\frac{\|Ax - b\|_2^2}{1 + \|x\|_2^2} \text{ and } \lambda_L = -\frac{1}{\delta^2} (b^\top (Ax - b) - \lambda_I). \quad (3.8)$$

Stripped-off from the details of how to make the computations efficient (*i.e.*, to solve the system (3.7) efficiently when  $\lambda_I$  and  $\lambda_L$  are fixed – for which we refer to [42]), the method is, essentially, a grid search over possible values of  $\lambda_I$ , when  $\lambda_L$  is considered as regularization parameter instead of  $\delta$ . The optimal RTLS solution for a fixed  $\lambda_L$  is the solution of (3.7) for an  $x$  that corresponds to a  $\lambda_I$  satisfying the first relation in (3.8) and being of minimum absolute value; this corresponds to a minimum value of the TLS criterion.

This method does not solve the quadratically constrained RTLS problem that involves a given  $\delta$ . However, it gives us the flavor of the fact that no regularization method, regardless its original formulation, can be complete without a good method for choosing a certain hyperparameter, which in this case is the  $\lambda_L$  parameter.

### 3.2.2 RTLS method of Sima, Van Huffel and Golub: iterative update of the solution vector, using quadratic eigenvalue problems

In this section it is shown how the RTLS problem can be solved numerically with an iterative method that requires at each iteration the solution of a quadratic eigenvalue problem.

It is known that the quadratically constrained least squares problem

$$\min_x \|Ax - b\|_2^2 \quad \text{subject to } \|x\|_2^2 = \delta^2$$

can be solved via one QEP [40]:

$$(\lambda^2 I + 2\lambda H + H^2 - \delta^{-2} g g^\top)y = 0, \quad (3.9)$$

where  $H = A^\top A$ , and  $g = A^\top b$ .

For RTLS, the difficulty is that a single QEP cannot solve the problem. Therefore, an iterative procedure to approximate the solution by solving at each step a QEP is proposed here. Numerically, it was observed that very few iterations are needed in order to achieve a desired accuracy of the solution.

#### RTLSQEP algorithm

Consider the RTLS problem (3.6) and write the Lagrangian

$$\mathbf{L}(x, \lambda) = \frac{\|Ax - b\|_2^2}{1 + \|x\|_2^2} + \lambda (\|Lx\|_2^2 - \delta^2).$$

The first order optimality conditions are:

$$B(x)x + \lambda L^\top Lx = d(x), \quad \|Lx\|_2^2 = \delta^2, \quad (3.10)$$

where

$$B(x) = \frac{A^\top A}{1 + \|x\|_2^2} - \frac{\|Ax - b\|_2^2}{(1 + \|x\|_2^2)^2} I_n, \quad d(x) = \frac{A^\top b}{1 + \|x\|_2^2}. \quad (3.11)$$

Algorithm 3.1 shows how to solve system (3.10) iteratively, using a fixed-point method. The focus at every iteration will be on solving system (3.12). In subsection 3.2.2 it is shown

---

**Algorithm 3.1** RTLSQEP algorithm
 

---

1: [Initializations]

Let  $x^0$  be a starting vector. Compute  $B_0 := B(x^0)$  and  $d_0 := d(x^0)$  from (3.11). Set  $k = 0$ .

2: [step  $k$ ]

Find  $x^{k+1}$  and  $\lambda_{k+1}$ , which solve the system in  $x$  and  $\lambda$ :

$$B_k x + \lambda L^\top Lx = d_k, \quad \|Lx\|_2^2 = \delta^2, \quad (3.12)$$

corresponding to the largest  $\lambda$  (using an equivalent quadratic eigenvalue problem).

Compute  $B_{k+1} := B(x^{k+1})$  and  $d_{k+1} := d(x^{k+1})$  from (3.11).

3: [stopping criterion]

If  $\|B_{k+1}x^{k+1} + \lambda_{k+1}L^\top Lx^{k+1} - d_{k+1}\|_2 < \varepsilon$ , where  $\varepsilon$  is a specified tolerance, then STOP; else  $k \leftarrow k + 1$  and go to step  $k$ .

---

how this can be done using a monic QEP.

**Remark 9** Notice that at **step**  $k$ , the solution with largest  $\lambda$  is selected from the set of solutions of system (3.12). This choice is needed for the convergence of the algorithm (see Lemma 3.1 and Theorem 3.3).

### Quadratic eigenvalue problem derivation

Consider the system

$$Bx + \lambda L^\top Lx = d, \quad \|Lx\|_2^2 = \delta^2, \quad (3.13)$$

with  $B$  a symmetric matrix.

#### Case 1: $L$ square and invertible

If  $L$  is invertible, a change of variable,  $z = Lx$ , gives

$$L^{-T}BL^{-1}z + \lambda z = L^{-T}d, \quad z^\top z = \delta^2. \quad (3.14)$$

Therefore, one is led to a system of the form:

$$Wz + \lambda z = h, \quad z^\top z = \delta^2, \quad (3.15)$$

with symmetric matrix  $W = L^{-T}BL^{-1}$ , which can be solved using a QEP [40]. Indeed, assuming  $\lambda > 0$  large enough (such that  $W + \lambda I$  is positive definite) and denoting by  $u = (W + \lambda I)^{-2}h$ , one has  $h^\top u = z^\top z = \delta^2$ ; noticing that  $h = \delta^{-2}hh^\top u$ , the condition

$$(W + \lambda I)^2 u = h$$

can be equivalently written as the QEP

$$(\lambda^2 I + 2\lambda W + W^2 - \delta^{-2}hh^\top)u = 0. \quad (3.16)$$

This QEP is solved in order to find the largest (right-most) eigenvalue  $\lambda$  and a corresponding eigenvector  $u$  (scaled such that  $h^\top u = \delta^2$ ).

**Remark 10** It is known [125] that a QEP having a nonsingular coefficient matrix for the second order term  $\lambda^2$  (in particular, monic QEP) has a full set of *finite* eigenvalues. When all coefficient matrices are real and symmetric, the quadratic eigenvalues are real or come in complex conjugate pairs; moreover, the special structure of the QEP (3.16) will enforce the right-most eigenvalue to be *real* and *positive*. Therefore, expressing system (3.12) as a monic QEP is also a guarantee that a solution corresponding to the largest  $\lambda > 0$  can be found.

The solution of the original problem is recovered by setting first  $z = (W + \lambda I)u$ , then  $x = L^{-1}z$ .

### Case 2: Generalization for nonsquare $L$

If  $L$  has more columns than rows (for example, when  $L$  is an approximation matrix of the first or second order derivative operator), then  $L^\top L$  is singular. Let  $L^\top L = USU^\top$  be the eigenvalue decomposition of  $L^\top L$ . An equivalent form for (3.13) is

$$U^\top BUy + \lambda Sy = U^\top d, \quad y^\top Sy = \delta^2, \quad (3.17)$$

where  $y = U^\top x$ . Let  $r = \text{rank}(S)$  and  $S_1 = S_{1:r,1:r}$ . Partitioning the elements of system (3.17) according to the rank  $r$ :

$$U^\top BU = \begin{bmatrix} T_1 & T_2 \\ T_2^\top & T_4 \end{bmatrix}, \quad S = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad U^\top d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

the system to be solved becomes:

$$\begin{cases} T_1 y_1 + T_2 y_2 + \lambda S_1 y_1 & = d_1, & y_1^\top S_1 y_1 = \delta^2. \\ T_2^\top y_1 + T_4 y_2 & = d_2, \end{cases} \quad (3.18)$$

Under the assumption that  $T_4$  is invertible (otherwise its pseudoinverse may still be used instead of its inverse),

$$y_2 = T_4^{-1}(d_2 - T_2^\top y_1) \quad (3.19)$$

is substituted into the first equation of (3.18):

$$(T_1 - T_2 T_4^{-1} T_2^\top + \lambda S_1) y_1 = (d_1 - T_2 T_4^{-1} d_2).$$

For  $W = S_1^{-\frac{1}{2}}(T_1 - T_2 T_4^{-1} T_2^\top) S_1^{-\frac{1}{2}}$  and  $h = S_1^{-\frac{1}{2}}(d_1 - T_2 T_4^{-1} d_2)$ , a system in the same form as (3.15) for the variable  $z = S_1^{\frac{1}{2}} y_1$  is obtained; it can be solved as described before, in order to find a solution  $(\lambda, z)$ , corresponding to the largest  $\lambda$ .

The solution  $(y_1, y_2)$  of (3.18) is given by

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} S_1^{-\frac{1}{2}} z \\ T_4^{-1}(d_2 - T_2^\top S_1^{-\frac{1}{2}} z) \end{bmatrix}. \quad (3.20)$$

Therefore, the solution for the original problem is  $x = Uy$ .

### Convergence of the method

In this subsection, a convergence theorem is proven. The notation  $f(x) := \frac{\|Ax - b\|_2^2}{1 + \|x\|_2^2}$  will be used.

**Lemma 3.1.** *If  $(\lambda_{k+1}, x^{k+1})$  is a solution of system (3.12) corresponding to the largest  $\lambda_{k+1}$ , then  $x^{k+1}$  is the global minimizer of the quadratically constrained quadratic optimization problem:*

$$\min_x x^\top B_k x - 2d_k^\top x \quad \text{subject to } \|Lx\|_2^2 = \delta^2, \quad (3.21)$$

*if such a global minimum exists.*

**Proof.** The proof follows the ideas in [39, Th. 1]. □

**Lemma 3.2.** *The optimization problem (3.21) admits a global minimum if and only if the vector  $x^k$  satisfies*

$$\min_{x \in \text{Null}(L^\top L), x \neq 0} \frac{x^\top A^\top A x}{x^\top x} \geq f(x^k). \quad (3.22)$$

**Proof.** In the case  $L$  is square and nonsingular, the feasibility region  $\mathcal{F} = \{x \mid \|Lx\|_2^2 = \delta^2\}$  is a nondegenerate ellipsoid, therefore any quadratic function attains its global minimum on  $\mathcal{F}$ . On the other hand,  $\text{Null}(L^\top L) = \{0\}$ , so any vector  $x^k$  satisfy (3.22).

If  $L^\top L$  is singular, any  $x$  can be uniquely written as  $x_1 + x_2$ , with  $x_1^\top x_2 = 0$ ,  $x_1 \in \text{Range}(L^\top L)$ ,  $x_2 \in \text{Null}(L^\top L)$ . It is easy to see that the unboundedness of a quadratic function of  $x$  can only occur from the contribution of the  $x_2$  part. In order to attain a global minimum of the function in (3.21), it is required that  $x_2^\top B_k x_2 \geq 0$ , for any  $x_2 \in \text{Null}(L^\top L)$ . From the definition formula (3.11) of  $B_k = B(x^k)$ , the relation (3.22) is readily obtained. □

**Theorem 3.3.** *For a starting vector  $x^0$  that satisfies (3.22) (with  $k = 0$ ), Algorithm 3.1 provides a sequence of vectors  $\{x^k\}_{k=1,2,\dots}$ , for which the function  $f$  is monotonically decreasing:*

$$0 \leq f(x^{k+1}) \leq f(x^k), \quad \forall k = 0, 1, 2, \dots \quad (3.23)$$

*Any limit point  $(\bar{\lambda}, \bar{x})$  of the sequence  $\{(\lambda_k, x^k)\}_k$  is a solution of the system (3.10).*

**Proof.** For a fixed value of  $k$ , denote by  $g_k(x)$  the objective function in the minimization (3.21):

$$g_k(x) = x^\top B_k x - 2d_k^\top x.$$

Suppose, for now, that the iterate  $x^k$  satisfies the assumption (3.22), thus, by Lemma 3.2,  $g_k$  admits a global minimum.

Taking into account the definition formulas (3.11) for  $B_k = B(x^k)$  and  $d_k = d(x^k)$ , simple algebraic manipulations give

$$g_k(x) = \frac{1}{1 + \|x^k\|_2^2} \left( x^\top A^\top A x - f(x^k) x^\top x - 2b^\top A x \right).$$

Disregarding the constant factor, the minimization of  $g_k(x)$  with respect to  $x$  (subject to the quadratic constraint  $\|Lx\|_2^2 = \delta^2$ ) is equivalent to minimizing

$$\begin{aligned} x^\top A^\top A x - f(x^k) x^\top x - 2b^\top A x &= \|Ax - b\|_2^2 - f(x^k) x^\top x - b^\top b \\ &= (1 + \|x\|_2^2)(f(x) - f(x^k)) + f(x^k) - b^\top b, \quad (\text{s.t. } \|Lx\|_2^2 = \delta^2). \end{aligned}$$

Therefore, the following equivalent problem is derived:

$$\min_x (1 + \|x\|_2^2)(f(x) - f(x^k)) \quad \text{subject to } \|Lx\|_2^2 = \delta^2. \quad (3.24)$$

Let  $\bar{g}_k(x) := (1 + \|x\|_2^2)(f(x) - f(x^k))$ . The iterate  $x^{k+1}$  is a solution for system (3.12), and, by Lemma 3.1, it is the global minimizer of  $g_k(x)$  under the quadratic constraint  $\|Lx\|_2^2 = \delta^2$ . Therefore,  $x^{k+1}$  is also the optimal solution for (3.24). It implies that for any  $x \in \mathbb{R}^n$ ,  $\bar{g}_k(x^{k+1}) \leq \bar{g}_k(x)$ . In particular, for  $x := x^k$ ,

$$\bar{g}_k(x^{k+1}) \leq \bar{g}_k(x^k) \Leftrightarrow (1 + \|x^{k+1}\|_2^2)(f(x^{k+1}) - f(x^k)) \leq 0 \Leftrightarrow f(x^{k+1}) \leq f(x^k).$$

Since  $x^0$  satisfies the assumption (3.22), all the arguments above hold for the case  $k = 0$ . Therefore,  $f(x^1) \leq f(x^0)$ , and  $x^1$  satisfies (3.22), too. By induction, all iterates  $x^k$  satisfy (3.22), and the proof holds for any  $k$ .

The second part of the theorem is trivial: making  $k \rightarrow \infty$  for any convergent subsequence of  $\{(\lambda_k, x^k)\}_k$ , the relation

$$B_{k-1}x^k + \lambda_k L^\top L x^k = d_{k-1},$$

implies  $B(\bar{x})\bar{x} + \bar{\lambda} L^\top L \bar{x} = d(\bar{x})$ . In addition, the quadratic relation  $\|Lx^k\|_2^2 = \delta^2$  is guaranteed at every iteration, and it is preserved for  $\bar{x}$ . Therefore,  $\bar{x}$  is a solution for (3.10).

### Initial vector

Theorem 3.3 says that, when  $L$  is square and nonsingular, any random vector can be used as a starting vector in the RTLSQEP algorithm, whereas if  $L$  is rectangular, the starting vector  $x^0$  should satisfy the condition (3.22). Equivalently, in the latter case, if  $N$  is a matrix whose columns generate  $\text{Null}(L^\top L)$ ,  $x^0$  should satisfy

$$f(x^0) \leq \min_{y \neq 0} \frac{y^\top N^\top A^\top A N y}{y^\top N^\top N y} = \sigma_{\min}^2(AN, N),$$

where  $\sigma_{\min}(M, N)$  denotes the minimum generalized singular value of the matrix pair  $(M, N)$ .

In the common case when  $L$  is taken as the approximation matrix of the first order derivative operator,  $L = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1) \times n}$ , this condition is rewritten as  $f(x^0) \leq \frac{1}{n} \mathbf{1}_n^T A^T A \mathbf{1}_n$ , where  $\mathbf{1}_n$  is the vector of all ones.

### Computational remarks

In the RTLSQEP algorithm, solving a QEP of the form (3.9) will be the most important computation at each iteration. A thorough survey on quadratic eigenvalue problems, their properties and solvers can be found in [125]. For solving quadratic eigenvalue problems at every iteration, it is possible to choose between several computational methods. In particular, one could either *linearize* the QEP and then solve a (generalized) eigenvalue problem, or use a *direct* method for QEPs.

One way to linearize the QEP (3.16) is:

$$\begin{bmatrix} -2W & -W^2 + \delta^{-2}hh^T \\ I & 0 \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = \lambda \begin{bmatrix} v \\ u \end{bmatrix}, \quad (3.25)$$

yielding a standard eigenvalue problem with nonsymmetric matrix. Another possible linearization is:

$$\begin{bmatrix} -W^2 + \delta^{-2}hh^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} 2W & I \\ I & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \quad (3.26)$$

which is a generalized eigenvalue problem with symmetric matrices.

Any of these eigenvalue problem should be solved in order to find the largest real eigenvalue  $\lambda$  and an associated eigenvector. An expensive approach consists in computing the complete eigenvalue decomposition. Such a technique is actually used by Matlab's function `polyeig`, which solves quadratic (or polynomial) eigenvalue problems by linearization. For efficiency, it is preferable to restrict the computations to finding only the rightmost eigenvalue and an associated eigenvector. The rightmost eigenvalue is not necessarily the eigenvalue of largest magnitude, therefore it is not possible to apply the power iteration method directly. Polynomial or rational preconditioning should be used in order to transform the search for the rightmost eigenvalue into a search for the largest magnitude eigenvalue.

For larger dimensions one must avoid even forming matrix  $W$  (and, of course,  $W^2$ ). From the definition of  $W$  (in subsection 3.2.2), it is clear that matrix-vector products with  $W$  can be executed in a fast way for particular forms of nonsingular  $L$  (banded Toeplitz, for instance). For large scale problems, it is advantageous to have  $A$  sparse, but all tests presented later in this chapter have dense  $A$ . Using only matrix-vector products with the matrix in (3.25), one can apply Arnoldi method for computing the largest real eigenvalue and corresponding eigenvector.

Recently, a fast method for solving QEPs was proposed in [80]. It can be successfully applied to monic QEPs of the form  $(\lambda^2 I + \lambda A + B)y = 0$ , where matrices  $A$  and  $B$  satisfy the condition that some linear combination  $\zeta A + \xi B$  is of low rank. The special form of the QEP (3.16) allows the application of a Lanczos process that simultaneously projects

**Table 3.1.** *frtlsqep* and *rtlsqep* implementations of the RTLSQEP algorithm.

<code>rtlsqep</code>	RTLSQEP algorithm 3.1, where at [step $k$ ] the system in $x$ and $\lambda$ is solved by the equivalent QEP using <b>linearization</b> (3.25) and <b>the ARPACK <code>eigs</code> function</b>
<code>frtlsqep</code>	RTLSQEP algorithm 3.1, where at [step $k$ ] the system in $x$ and $\lambda$ is solved by the equivalent QEP using the <b>fast QEP solver of Li and Ye</b> [80]

the matrices  $W$  and  $hh^\top$  into a common subspace. After  $k$  steps of the Lanczos process, the projections of  $W$  and  $hh^\top$  can be approximated by two  $k \times k$  symmetric banded matrices with bandwidth 2. Using the reduced matrices, a lower dimensional QEP that approximates the original one can be solved by standard methods (*i.e.*, linearization and eigenvalue decomposition). Further details are given in [80].

For the numerical tests in Section 3.4, a distinction will be made between two implementations of the RTLSQEP method. The first, which will be referred to as `rtlsqep`, uses the linearization (3.25) and computes the rightmost eigenpair with the ARPACK implementation [76] included in Matlab (namely the function `eigs`)<sup>3</sup>. The second, referred to as `frtlsqep`, uses the fast solver described in [80], which solves lower dimensional approximations of the original QEPs. To be more clear, we summarize the content of our two implementations in Table 3.1.

### 3.2.3 RTLS method of Renaut and Guo: alternating iteration on a scalar and the solution vector

Guo and Renaut [53] had an initial method that solved a parameter dependent eigenvalue problem starting from the 2-parameter formulation of Golub, Hansen and O’Leary (3.7). A shifted inverse power method was used to obtain an eigenpair  $(-\lambda_l, [x^\top \ -1]^\top)$  for the problem

$$D(\lambda_L) \begin{bmatrix} x \\ -1 \end{bmatrix} = -\lambda_l \begin{bmatrix} x \\ -1 \end{bmatrix},$$

with

$$D(\lambda_L) = \begin{bmatrix} A^\top A + \lambda_L L^\top L & A^\top b \\ b^\top A & -\lambda_L \delta^2 + b^\top b \end{bmatrix}, \quad (3.27)$$

where  $\lambda_l$  and  $\lambda_L$  are also given by (3.8). However, this method does not always converge, and the cases when convergence occurs or does not occur were not effectively analyzed.

Numerical results in Section 3.4 and in [117] compare, among others, the QEP method with the method in [53] and these comparisons show that only when the initial starting vector for the method in [53] belongs to a very narrow neighborhood around the optimal solution, the method converges successfully.

<sup>3</sup>We chose for the nonsymmetric linearization (3.25) instead of the symmetric linearization (3.26) because the function `eigs` can only solve standard eigenvalue problems  $Ax = \lambda x$  or generalized eigenvalue problems of the type  $Ax = \lambda Bx$  with  $B$  symmetric and positive definite. However, the  $B$  matrix in the linearization (3.26) is *indefinite*.

Renaut and Guo [103] carefully reworked their idea and arrived at another iterative method based on finding a zero of the constraint function  $g(x) = (\|Lx\|^2 - \delta^2)/(\|x\|^2 + 1)$ . The zero-finding refers to finding a scalar value for the parameter  $\lambda_L$  (see (3.8)) such that  $g(x_{\lambda_L}) = 0$ , where  $([x_{\lambda_L}^\top \ -1]^\top)^\top$  is the eigenvector corresponding to the smallest eigenvalue of the matrix  $D(\lambda_L)$  in (3.27). In [103] it is proven that  $\lambda_L \mapsto g(x_{\lambda_L})$  is a decreasing function of  $\lambda_L$ , and it has a unique zero.

In summary, the main iterative procedure proposed in [103] is presented in Algorithm 3.2 below.

---

**Algorithm 3.2** Alternating iteration algorithm from [103]

---

1: [Initializations]

Given  $\delta^2 > 0$  and  $\lambda_L^{(0)} > 0$ , calculate eigenpair  $(\rho_{n+1}^{(0)}, [x^{(0)\top} \ -1]^\top)^\top$  of  $D(\lambda_L^{(0)})$ .

Set  $k = 0, \iota = 1$ .

2: [Step  $k$ ]

If  $k > 0$  and  $g(x^{(k)}) \cdot g(x^{(k+1)}) < 0$ , then  $\iota = \iota/2$  else  $\iota = 1, k = k + 1$ .

3: [Update  $\lambda_L$ ]

$\lambda_L^{(k)} = \lambda_L^{(k-1)} \left(1 + \frac{\iota}{\delta^2} g(x^{(k-1)})\right)$ .

4: [Update  $x$ ]

estimate the smallest eigenvalue  $\rho_{n+1}^{(k+1)}$  and a corresponding eigenvector  $[x^{(k+1)\top} \ -1]^\top$  of the matrix  $D(\lambda_L^{(k)})$ .

5: [Stopping criterion]

If  $|g(x^{(k)})| < \varepsilon$ , where  $\varepsilon$  is a specified tolerance, then STOP; else  $k \leftarrow k + 1$  and go to step  $k$ .

---

### 3.2.4 RTLS method of Beck, Ben-Tal and Teboulle: one scalar optimization

A more general theory for the optimization of quadratically constrained fractional quadratic problems is the starting point that is taken in [7]. The authors prove that, under reasonable assumptions (which do not even require positive semidefiniteness of the quadratic forms involved), a quadratically constrained fractional quadratic problem can be equivalently solved using a scalar optimization problem.

For the RTLS problem (3.4)<sup>4</sup>, the essential remark leading to the development in [7] is that the following two statements are equivalent:

1.  $\min_{\{x: \|Lx\|^2 \leq \delta^2\}} \frac{\|Ax - b\|^2}{\|x\|^2 + 1} \leq \alpha,$
2.  $\min_{\{x: \|Lx\|^2 \leq \delta^2\}} \{ \|Ax - b\|^2 - \alpha(\|x\|^2 + 1) \} \leq 0.$

Note that the second problem is a quadratically constrained quadratic minimization; although it might still be a nonconvex problem, it is clear that this problem (for a fixed value

---

<sup>4</sup>Note that (3.4) is the inequality constrained RTLS. We consider this formulation only to keep in line with [7]. The results hold for the equality constrained formulation as well.

of  $\alpha$ ) is much simpler than the RTLS problem. Obviously, what we are looking for is  $\alpha^* := \min_{\{x: \|Lx\|^2 \leq \delta^2\}} \frac{\|Ax-b\|^2}{\|x\|^2+1}$ . The above equivalence suggests that a search for the optimal  $\alpha^*$  can be performed within the framework of the second problem as a search for a zero of the function

$$\phi(\alpha) = \min_{\{x: \|Lx\|^2 \leq \delta^2\}} \{\|Ax-b\|^2 - \alpha(\|x\|^2 + 1)\}.$$

In fact,  $\phi$  is clearly a decreasing function of  $\alpha$  and it is easy to find upper and lower initial values for  $\alpha$  such that  $\phi(\alpha^{\text{low}}) \geq 0$  and  $\phi(\alpha^{\text{up}}) \leq 0$  (e.g.,  $\alpha^{\text{low}} = 0$  and  $\alpha^{\text{up}} = \|b\|^2$ ). This leads to the observation that one can use a bisection search for  $\alpha$  to find the unique zero of  $\phi$ , and, simultaneously, an optimal solution of the RTLS problem. We provide in Algorithm 3.3 the procedure proposed for this purpose in [7].

---

**Algorithm 3.3** RTLSC algorithm from [7]

---

1: [Initializations]

Given  $\delta^2 > 0$ ,  $\alpha^{\text{up}} > 0$  – an upper bound on the optimal function value,  $\varepsilon$  – a tolerance value,

set  $k = 0$ ,  $\alpha_0^{\text{low}} = 0$ ,  $\alpha_0^{\text{up}} = \alpha^{\text{up}}$ .

2: [Step  $k$ ]

2a:  $\alpha_k \leftarrow \frac{1}{2}(\alpha_k^{\text{low}} + \alpha_k^{\text{up}})$

2b: Solve the subproblem

$$\min_{\{x: \|Lx\|^2 \leq \delta\}} \{x^\top (A^\top A - \alpha_k I)x - 2b^\top Ax\}$$

and denote the optimal solution with  $x^k$  and the optimal objective function value with  $\beta_k$ .

2c:  $f_k \leftarrow f(x^k)$

2d: **if**  $\beta_k + \|b\|^2 > \alpha_k$  **then**

$$\alpha_{k+1}^{\text{low}} \leftarrow \alpha_k, \quad \alpha_{k+1}^{\text{up}} \leftarrow \min\{\alpha_k^{\text{up}}, f_k\},$$

**else**

$$\alpha_{k+1}^{\text{low}} \leftarrow \alpha_k^{\text{low}}, \quad \alpha_{k+1}^{\text{up}} \leftarrow \min\{\alpha_k, f_k\}.$$

2e:  $k \leftarrow k + 1$

3: [Stopping criterion]

**If**  $\alpha_k^{\text{up}} - \alpha_k^{\text{low}} < \varepsilon$ , **then STOP**; **else** go to Step  $k$ .

---

Some comments are in order. The subproblem in step 2b: is another way of writing the evaluation of  $\phi(\alpha_k)$ , such that it is more obvious that we deal with a quadratically constrained (possibly indefinite) quadratic problem. We remark the similarity between the subproblem in step 2b: and the subproblem obtained for the RTLSQEP algorithm: see equation (3.21). The computational method chosen in [7] involves solving a secular equation in one scalar parameter by means of a Newton method with global quadratic convergence rate, due to Melman [88].

Theoretical analysis in [7] provides conditions for existence of finite optimal solution to the problem in step 2b:. Not surprisingly, their condition resemble our condition (3.22); indeed, Proposition 4.2 in [7] says that the subproblem to be solved at step  $k$  of Algorithm 3.3 item 2b: admits finite minimum if

$$\lambda_{\min} \left( (N^{\top} N)^{-1/2} (N^{\top} A^{\top} A N) (N^{\top} N)^{-1/2} \right) > \alpha_k,$$

where  $\lambda_{\min}$  denotes the smallest eigenvalue of a matrix, and  $N$  is, as before, a matrix whose columns span the nullspace of  $L^{\top} L$ . Since the left-hand side above is nothing else than  $\sigma_{\min}^2(AN, N)$  from §3.2.2, it means that the existence condition from [7] is the same as for RTLSQEP, with the only difference that  $\alpha_k$  replaces  $f(x^k)$ .

As for RTLSQEP (see §3.2.2), it is sufficient to have a guarantee that the first subproblem solved at iteration  $k = 0$  has finite global minimum. This property is automatically satisfied at further iterations. For RTLSQEP, this implies a guarantee that the algorithm converges to a local minimum. However, for the formulation in this section, this implies also that the converged solution is a global solution as well, because of the fact that the reformulation  $\phi(\alpha) = 0$  admits a unique solution  $\alpha^*$ . Note from Algorithm 3.3 that the choice of the first  $\alpha_0$  is replaced by the choice of upper and lower bounds,  $\alpha^{\text{up}}$  and  $\alpha^{\text{low}}$ . We refer for more details on these choices to [7].

It is remarkable that in [7] these results are in fact proven for the more general problem where the objective function  $f$  is a ratio of two arbitrary quadratic forms; the RTLS problem is only a special case.

### 3.3 Quadratic penalty formulations

When a Tikhonov-type quadratic penalty term  $\|Lx\|^2$  is added to the TLS objective function, we obtain a problem of the form

$$\min_x \frac{\|Ax - b\|_2^2}{1 + \|x\|_2^2} + \lambda \|Lx\|_2^2. \quad (3.28)$$

For  $\delta$  small enough (*i.e.*,  $\delta < \|Lx^{\text{TLS}}\|_2$ ), there exists a value of the parameter  $\lambda > 0$  such that the solution of (3.4) coincides with the solution of (3.28).

For this quadratic penalty formulation, a complete analysis is recently presented in [6] and an optimization method is proposed there. We shortly provide an account of the main conclusions.

- A sufficient condition for attainability of the minimum of (3.28) is given. This condition is:

$$\lambda_{\min} \left( \begin{bmatrix} N^{\top} A^{\top} A N & N^{\top} A^{\top} b \\ b^{\top} A N & b^{\top} b \end{bmatrix} \right) < \lambda_{\min}(N^{\top} A^{\top} A N),$$

or, in terms of SVDs (which are more common in the TLS literature):  $\sigma_{\min}([AN \ b]) < \sigma_{\min}(AN)$ . Here, as before,  $N$  is a basis for the nullspace of  $L^{\top} L$ . Remark the similarity of this condition with the condition for existence of the TLS solution:  $\sigma_{\min}([A \ b]) < \sigma_{\min}(A)$ .

- A simple reformulation makes the problem more tractable. It is replaced with the minimization of the one-variable function

$$\mathcal{G}(\alpha) := \min_{\|x\|^2 = \alpha - 1} \left\{ \frac{\|Ax - b\|^2}{\alpha} + \lambda \|Lx\|^2 \right\}. \quad (3.29)$$

- Properties of  $\mathcal{G}$  include: continuity and differentiability, and, in many practical cases, unimodality.
- The evaluation of  $\mathcal{G}$  involves solving a quadratically constrained quadratic optimization; it can be, thus, computed from an equivalent secular equation.
- Lower and upper bounds for  $\alpha$  can be computed in terms of the data  $A$ ,  $b$ ,  $L$  and  $\lambda$ . This makes the minimization of  $\mathcal{G}$  possible using a bisection search.

### 3.4 Numerical results

We mainly give in this section the numerical results from [117]. We comment at the end about how RTLSQEP compares with the newer methods described in §3.2.3 and §3.2.4.

#### 3.4.1 Test problems description

In order to test the performance of the proposed RTLSQEP method, several problems from the “Regularization Tools” [57] were employed. All of them are discretizations of continuous ill-posed problems of the Fredholm integral type [59, §2], constructed by quadrature. For completeness, we provide in Table 3.2 the elements of the integral equations used in the considered examples. Further explanation about the precise discretization schemes used can be found in the manual file accompanying the Regularization Tools.

**Table 3.2.** Elements  $K$ ,  $f$  and  $g$  of the Fredholm integral equations  $\int_a^b K(s,t)f(t)dt = g(s)$  in several examples from the Regularization Tools.

Name	baart	deriv2	ilaplace	shaw
$K(s,t)$	$\exp(s \cos t)$	$\begin{cases} s(t-1), & s < t \\ t(s-1), & s \geq t \end{cases}$	$\exp(-st)$	$(\cos s + \cos t)^2 \left( \frac{\sin(\pi(\sin s + \sin t))}{\pi(\sin s + \sin t)} \right)^2$
$f(t)$	$\sin t$	$\exp t$	$1 - \exp(-\frac{t}{2})$	$2 \exp(-6(t-.8)^2) + \exp(-2(t+.5)^2)$
$g(s)$	$2 \frac{\sin s}{s}$	$\exp s + (1-e)s - 1$	$\frac{1}{s} - \frac{1}{s+1/2}$	-
$s \in$	$[0, \frac{\pi}{2}]$	$[0, 1]$	$[0, \infty)$	$[-\pi/2, \pi/2]$
$t \in$	$[0, \pi]$	$[0, 1]$	$[0, \infty)$	$[-\pi/2, \pi/2]$

The example functions from the Regularization Tools return the elements of a square system  $A^{\text{true}} x^{\text{true}} \approx b^{\text{true}}$ , with matrix  $A^{\text{true}}$  singular or very ill-conditioned. In the classical context (*i.e.*, without additional regularization), it is known that TLS gives more accurate results than LS when increasing the degree of overdetermination, provided entries of  $[A \ b]$  are affected by independent identically distributed errors of zero mean and equal

variance [132, §8]. For this reason, the RTLS approach is also tested for rectangular systems. Some example functions from “Regularization Tools” were easily modified to construct rectangular problems (by using a rectangular discretization grid instead of a square one or by “partitioning” a square system in order to form an overdetermined system).

In the following, denote by  $A^{\text{true}}, x^{\text{true}} = b^{\text{true}}$  a (square or rectangular) ill-conditioned example system and let  $\sigma$  be a noise level. By adding white noise to the data:

$$A = A^{\text{true}} + \sigma E, \quad b = b^{\text{true}} + \sigma e,$$

with  $E = \text{randn}(m, n)$ ,  $e = \text{randn}(n, 1)$ , the problem to be solved becomes  $Ax \approx b$ .

The matrix  $L \in \mathbb{R}^{(n-1) \times n}$  is set to approximate the first order derivative operator. For the simulations below, the exact solutions are known; it is straightforward to consider as regularization condition the equality  $\|Lx\|_2 = \delta := \|Lx^{\text{true}}\|_2$ . With this strong prior knowledge imposed, it is expected to obtain a regularized solution  $x^{\text{reg}}$  of the original ill-conditioned problem which is close to the exact solution  $x^{\text{true}}$ .

### 3.4.2 Comparison between regularization solvers

The purpose of these tests is to numerically validate the RTLSQEP method, but also to compare its performance with other existing methods. The solvers employed in the tests are described in Table 3.3. Results were obtained in Matlab 6 on an i686 PC.

**Table 3.3.** *Solvers for regularized least squares and regularized TLS*

Solver	Description
tikhonov	Tikhonov regularization (from Hansen’s “Regularization Tools” [57])
rllsqep	RLS solved by a QEP (with Matlab’s <code>polyeig</code> )
frllsqep	RLS solved by a QEP (with fast Lanczos method [80])
ttls	truncated total least squares (from Hansen’s “Regularization Tools” [57])
rtlseig1	Guo and Renault’s eigenvalue method for RTLS [53] (random starting vector)
rtlseig2	Guo and Renault’s eigenvalue method for RTLS [53] (starting vector - the <code>frllsqep</code> solution)
rtllsqep	RTLSQEP (each QEP solved by linearization, with Matlab’s <code>eigs</code> )
frtllsqep	RTLSQEP (each QEP solved with fast Lanczos method [80])

For several noise levels  $\sigma$ , relative errors  $\|x^{\text{reg}} - x^{\text{true}}\| / \|x^{\text{true}}\|$  are averaged in 200 random simulations. Table 3.4 shows results for the square problem ( $m = n = 20$ ) and Table 3.5, results concerning the rectangular problem ( $m = 200, n = 20$ ).

The relative errors (averaged over 200 random simulations) are illustrated also in Figures 3.1 and 3.2 for two of the problems.

For the square problem or for small noise levels, there is no significant improvement of the RTLS solutions in comparison with the ‘traditional’ methods; in the overdetermined case and for increasing noise level, as expected, `rtllsqep` or `frtllsqep` gave the most

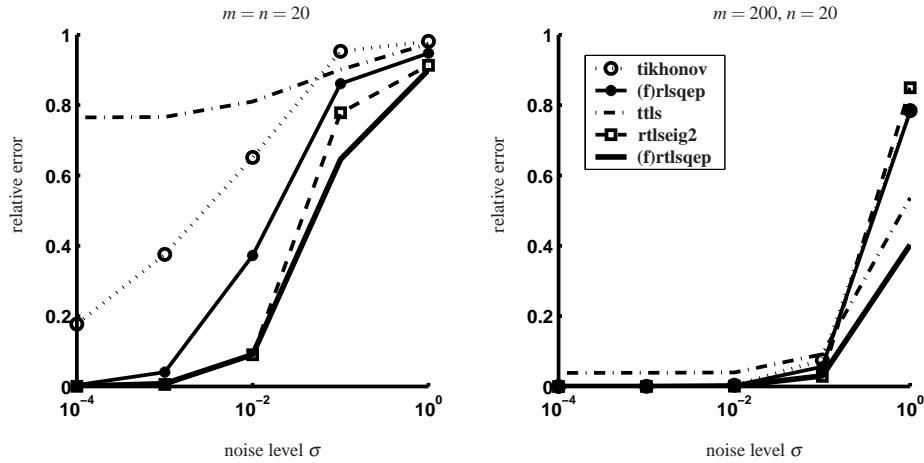


Figure 3.1. Average relative errors for example *ilaplace*

Table 3.4. The relative errors  $\|x^{reg} - x^{true}\|/\|x^{true}\|$  in several example problems, for all methods and several noise levels  $\sigma$ ; square case, with  $m = n = 20$ . The smallest errors for each problem set are indicated in underlined bold numbers.

<i>ilaplace</i>	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	1.8e-1	4.1e-3	7.6e-1	1.0e+0	<b><u>5.8e-4</u></b>	8.5e-4
$\sigma = 1e-3$	3.8e-1	4.1e-2	7.7e-1	1.0e+0	<b><u>5.6e-3</u></b>	8.0e-3
$\sigma = 1e-2$	6.5e-1	3.7e-1	8.1e-1	1.0e+0	<b><u>9.0e-2</u></b>	9.1e-2
$\sigma = 1e-1$	9.5e-1	8.6e-1	9.0e-1	9.0e-1	7.8e-1	<b><u>6.5e-1</u></b>
$\sigma = 1e+0$	9.8e-1	9.5e-1	9.7e-1	9.3e-1	9.1e-1	<b><u>9.0e-1</u></b>
<i>baart</i>	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	1.6e-2	2.1e-2	<b><u>1.3e-2</u></b>	1.0e+0	1.7e-2	1.6e-2
$\sigma = 1e-3$	2.9e-2	2.5e-2	2.7e-2	1.0e+0	<b><u>2.4e-2</u></b>	<b><u>2.4e-2</u></b>
$\sigma = 1e-2$	3.0e-1	<b><u>8.1e-2</u></b>	8.3e-2	1.0e+0	8.2e-2	8.2e-2
$\sigma = 1e-1$	9.5e-1	<b><u>2.6e-1</u></b>	3.1e-1	8.1e-1	3.2e-1	3.2e-1
$\sigma = 1e+0$	9.9e-1	7.4e-1	8.7e-1	8.0e-1	7.4e-1	<b><u>7.3e-1</u></b>
<i>shaw</i>	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	1.1e-2	<b><u>2.3e-3</u></b>	2.2e-2	1.0e+0	3.2e-3	<b><u>2.3e-3</u></b>
$\sigma = 1e-3$	1.2e-1	9.1e-3	2.2e-2	1.0e+0	<b><u>7.3e-3</u></b>	9.2e-3
$\sigma = 1e-2$	3.5e-1	1.1e-1	<b><u>3.0e-2</u></b>	1.0e+0	2.0e-1	2.0e-1
$\sigma = 1e-1$	6.2e-1	1.9e-1	<b><u>1.4e-1</u></b>	9.9e-1	4.1e-1	4.1e-1
$\sigma = 1e+0$	9.6e-1	7.5e-1	8.0e-1	8.9e-1	7.5e-1	<b><u>7.2e-1</u></b>
<i>deriv2</i>	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	6.1e-4	<b><u>1.5e-4</u></b>	5.5e-2	3.2e-1	<b><u>1.5e-4</u></b>	<b><u>1.5e-4</u></b>
$\sigma = 1e-3$	1.0e-2	<b><u>8.8e-4</u></b>	5.6e-2	3.2e-1	8.9e-4	8.9e-4
$\sigma = 1e-2$	8.1e-2	7.4e-3	8.7e-2	3.2e-1	<b><u>7.0e-3</u></b>	<b><u>7.0e-3</u></b>
$\sigma = 1e-1$	9.1e-1	<b><u>6.8e-2</u></b>	2.6e-1	2.2e-1	6.9e-2	6.9e-2
$\sigma = 1e+0$	9.7e-1	7.1e-1	8.8e-1	5.4e-1	6.4e-1	<b><u>4.9e-1</u></b>

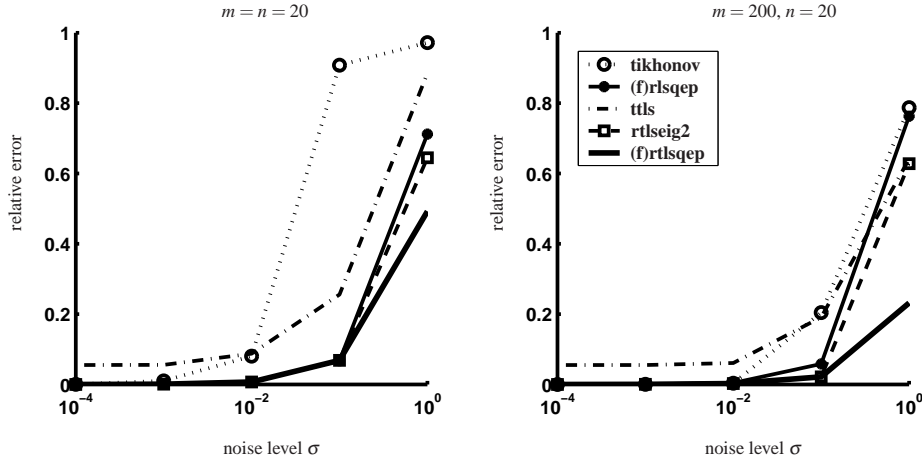


Figure 3.2. Average relative errors for example `deriv2`

Table 3.5. The relative errors  $\|x^{reg} - x^{true}\|/\|x^{true}\|$  in several example problems, for all methods and several noise levels  $\sigma$ ; overdetermined problems, with  $m = 200, n = 20$ . The smallest errors for each problem set are indicated in underlined bold numbers.

ilaplace	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	2.2e-4	<b><u>3.0e-6</u></b>	3.9e-2	3.2e-3	3.5e-5	3.1e-6
$\sigma = 1e-3$	4.8e-4	8.3e-5	3.9e-2	3.2e-3	<b><u>4.4e-5</u></b>	8.1e-5
$\sigma = 1e-2$	3.4e-3	2.8e-3	4.0e-2	3.3e-3	<b><u>6.9e-4</u></b>	1.3e-3
$\sigma = 1e-1$	7.5e-2	5.6e-2	9.2e-2	3.6e-2	<b><u>2.9e-2</u></b>	3.0e-2
$\sigma = 1e+0$	7.8e-1	7.8e-1	5.4e-1	8.5e-1	8.5e-1	<b><u>4.0e-1</u></b>
baart	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	2.5e-2	2.2e-2	<b><u>2.1e-2</u></b>	1.0e+0	<b><u>2.1e-2</u></b>	<b><u>2.1e-2</u></b>
$\sigma = 1e-3$	1.8e-1	4.8e-2	<b><u>3.5e-2</u></b>	1.0e+0	4.0e-2	4.0e-2
$\sigma = 1e-2$	2.3e-1	1.8e-1	<b><u>1.4e-1</u></b>	9.9e-1	2.7e-1	2.7e-1
$\sigma = 1e-1$	7.3e-1	7.1e-1	6.2e-1	6.5e-1	7.1e-1	<b><u>5.3e-1</u></b>
$\sigma = 1e+0$	9.6e-1	9.5e-1	9.5e-1	8.9e-1	9.0e-1	<b><u>8.1e-1</u></b>
shaw	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	1.8e-3	2.8e-3	<b><u>2.0e-4</u></b>	4.3e-1	3.6e-3	3.6e-3
$\sigma = 1e-3$	7.9e-2	1.6e-2	<b><u>4.2e-3</u></b>	3.5e-1	3.9e-2	3.9e-2
$\sigma = 1e-2$	1.6e-1	7.3e-2	<b><u>3.9e-2</u></b>	1.7e-1	1.6e-1	1.5e-1
$\sigma = 1e-1$	9.3e-1	8.9e-1	9.4e-1	7.0e-1	7.4e-1	<b><u>6.7e-1</u></b>
$\sigma = 1e+0$	9.6e-1	9.4e-1	9.6e-1	7.3e-1	7.8e-1	<b><u>7.0e-1</u></b>
deriv2	tikhonov	(f)rlsqep	ttls	rtlseig1	rtlseig2	(f)rtlsqep
$\sigma = 1e-4$	1.2e-4	<b><u>7.5e-5</u></b>	5.5e-2	3.2e-1	8.3e-5	8.9e-5
$\sigma = 1e-3$	<b><u>3.2e-4</u></b>	<b><u>3.2e-4</u></b>	5.5e-2	3.2e-1	3.7e-4	3.8e-4
$\sigma = 1e-2$	3.5e-3	4.4e-3	6.1e-2	3.2e-1	<b><u>2.7e-3</u></b>	<b><u>2.7e-3</u></b>
$\sigma = 1e-1$	2.0e-1	5.8e-2	1.9e-1	3.2e-1	<b><u>2.2e-2</u></b>	<b><u>2.2e-2</u></b>
$\sigma = 1e+0$	7.9e-1	7.6e-1	6.4e-1	2.8e-1	6.3e-1	<b><u>2.3e-1</u></b>

accurate results (or as accurate as the other solvers) for many of the experiments. For the iterative RTLS methods `rtlseig1`, `rtlsqep` and `frtlsqep`, random starting vectors were used for each one of the problems. For the `rtlseig` method in [53], starting from the same random vector gave bad results (as shown in the column corresponding to `rtlseig1`); nevertheless, using the regularized least squares solution given by `frlsqep` as starting vector improved the results of the same solver (shown in the `rtlseig2` column).

In Table 3.6, average timings and number of iterations for several problem dimensions are reported. Two representative examples from the Regularization Tools are used for this purpose: `baart`, which is a severely ill-posed problem, and `deriv2`, which is a mildly ill-posed problem.

In order to have a fair comparison, the same starting vector and the same termination criterion were used for all implementations. Namely, the initial vector was set to the regularized least squares solution, and the convergence test was:  $\|x^{k+1} - x^k\| / \|x^k\| < 1e-4$ .

For `rtlseig2`, each iteration involves solving an  $n \times n$  linear system, requiring, in general,  $\mathcal{O}(n^3)$  operations (or  $\mathcal{O}(n^2)$ , if  $A$  and  $L$  are first transformed via generalized singular value decomposition; however, this preprocessing is of cubic complexity).

The `rtlsqep` implementation solves at each iteration a quadratic eigenproblem via linearization. The total number of matrix-vector products with matrix  $W$  in (3.25) is also shown. This implementation is in general the fastest and allows solving large problems. We explain why `rtlsqep` is faster than `frtlsqep` by the fact that the former uses `eigs`, which calls the Fortran-written ARPACK library, while the latter is entirely implemented in Matlab code.<sup>5</sup>

Results in Table 3.6 are obtained with  $L$  set to the  $n \times n$  approximation matrix for the first order derivative operator. In this situation, one takes advantage of the fact that the number of *flops* for solving a system with  $L$  is linear in  $n$ . Therefore, each matrix-vector product with  $W = L^{-T} B_k L^{-1}$  is of order  $\mathcal{O}(2mn + 3n)$  operations.

Note that for both `rtlsqep` and `frtlsqep`, a very small number of iterations (under 5, which means that at most 5 quadratic eigenvalue problems were solved) is required for each example, regardless of the fact that one example is severely ill-posed and the other is mildly ill-posed. Moreover, for various problem sizes, almost the same number of matrix-vector products were performed (for the `rtlsqep` approach). This number is actually linked to the convergence of the Arnoldi method (ARPACK's `eigs`) and shows a stability of the number of Arnoldi steps with respect to problem dimensions.

### 3.4.3 Comparison with newer RTLS methods

In what concerns the comparison of RTLSQEP with the newer methods from [103] and [7], described, respectively, in §3.2.3 and §3.2.4, we summarize here the results reported by the authors of [103] and [7].

The simulations in [103] are favorable to the alternating iteration algorithm 3.2 when compared to RTLSQEP. However, we note a detail in the simulation scenario that makes RTLSQEP to produce bias results. The quadratic constraint chosen for RTLSQEP was  $\delta :=$

<sup>5</sup>We would like to thank Ren-Cang Li and Qiang Ye for providing the Matlab code of their fast quadratic eigenvalue problem method.

**Table 3.6.** Average timings in two example problems – one severely ill-posed problem (*baart*) and one mildly ill-posed problem (*deriv2*), for several RTLS methods, and for various problem dimensions. ‘Iter’ = number of iterations; ‘CPU’ = CPU time (in seconds); ‘ $W \times v$ ’ = number of matrix-vector products (with matrix  $W$ ); ‘–’ denotes problems that were not solved in comparable time ( $> 30$  min).

		rtlseig2		rtlsqep			frtlsqep	
		Iter	CPU	Iter	CPU	$W \times v$	Iter	CPU
	<i>baart</i>							
$m = n$	$n = 50$	406.8	0.62	4.0	0.17	135	4.0	0.15
$m = 2n$	$n = 50$	241.7	0.41	4.1	0.18	140	4.1	0.17
$m = n$	$n = 500$	777.0	364.67	4.0	1.56	135	4.0	19.00
$m = 2n$	$n = 500$	–	–	4.0	3.14	135	4.0	19.53
$m = 2n$	$n = 1000$	–	–	4.0	18.17	135	–	–
$m = 2n$	$n = 2500$	–	–	4.0	108.21	135	–	–
$m = n$	$n = 5000$	–	–	4.0	132.77	135	–	–
$m = 2n$	$n = 5000$	–	–	4.0	255.27	135	–	–
	<i>deriv2</i>							
$m = n$	$n = 50$	35.9	0.06	4.0	0.14	135	4.1	0.16
$m = 2n$	$n = 50$	23.0	0.04	4.0	0.15	135	4.0	0.15
$m = n$	$n = 500$	34.3	19.18	4.0	1.38	135	4.0	18.30
$m = 2n$	$n = 500$	34.6	20.03	4.3	3.26	148	4.0	18.86
$m = 2n$	$n = 1000$	–	–	4.0	15.03	135	–	–
$m = 2n$	$n = 2500$	–	–	4.0	105.10	135	–	–

$0.9\|Lx^{\text{true}}\|_2$  (remember that in our simulations we chose  $\delta := \|Lx^{\text{true}}\|_2$ , which makes it reasonable to compare, then, the solution  $x^{\text{reg}}$  with  $x^{\text{true}}$ ). The alternating iteration algorithm was used without a strict equality constraint, and gave better approximations of the true solution.

The extensive experiments in [7] show that RTLSQEP gives identical results with the RTLSC algorithm 3.3. This sustains the fact that RTLSQEP also finds the global optimum, although our theory in [117] or §3.2.2 only proves local convergence.

### 3.4.4 Comparison with optimization solvers

The RTLS methods discussed in §3.2 are, in fact, methods for solving a quadratically constrained nonconvex optimization problem. Numerical experiments confirm that classical optimization methods seem not as suited for the RTLS problem as the tailored RTLS methods described herein. For the quasi-Newton method (function `fmincon` from Matlab), a very good initial approximation must be used in order to have convergence. Decreasing the default tolerance values (even with a ‘good’ initial vector) has also the effect of non-convergence (after  $10^5$  iterations).

In Table 3.7, the average over 200 random simulations of the objective function value  $f(\cdot)$  is reported for several examples and methods. For reference, the function  $f$  is also evaluated in the exact solution  $x^{\text{true}}$  (which is the exact solution of the unperturbed example system  $A^{\text{true}}x^{\text{true}} = b^{\text{true}}$ , and not the optimal solution of the RTLS problem!). The vector  $x^{\text{true}}$  is actually used as the initial approximation for `fmincon`. Note that the solution provided by `fmincon` improves just a little bit the value at the initial approximation. In

**Table 3.7.** Average of the function value in 200 random simulations ( $m = 200$ ,  $n = 20$ ,  $\sigma = 0.001$ ).  $f(x^{true})$  is given just for reference.

	ilaplace	baart	shaw	deriv2
$f(x^{true})$	1.983e-4	2.005e-4	2.003e-4	2.018e-4
$f(x^{rtlseig2})$	1.932e-4	1.902e-4	1.944e-4	1.936e-4
$f(x^{frtlsqep})$	1.920e-4	1.902e-4	1.944e-4	1.932e-4
$f(x^{fmincon})$	1.961e-4	1.991e-4	1.982e-4	1.983e-4

**Table 3.8.** Average of the quadratic constraint violation in 200 random simulations ( $m = 200$ ,  $n = 20$ ,  $\sigma = 0.001$ ).

	ilaplace	baart	shaw	deriv2
$x^{rtlseig2}$	8.51e-04	1.99e-07	6.47e-09	1.76e-04
$x^{frtlsqep}$	3.08e-13	1.58e-13	2.24e-17	4.30e-14
$x^{fmincon}$	2.53e-07	2.48e-08	1.88e-08	7.46e-03

contrast, the objective function value at the `frtlsqep` solution is the smallest, for all examples.

Another remark on the RTLSQEP method is that the quadratic constraint  $\|Lx\|_2^2 = \delta^2$  is preserved at each iteration (at least in exact arithmetic). In Table 3.8, the constraint violation  $|\|Lx\|_2^2 - \delta^2|$  is averaged in 200 simulations for the RTLS solutions computed by three methods (`rtlseig2`, `frtlsqep`, and `fmincon`, the latter with initial approximation  $x^{true}$ ). Again, `frtlsqep` is much more accurate than the other two solvers. The results in this section show the good numerical performances of the RTLSQEP algorithm as a specialized nonlinear optimization solver. In practice, however, the merit that the constraint is satisfied with high accuracy is not so important, because the parameter  $\delta$  might not be known exactly.

### 3.4.5 Importance of the starting vector

The implementations of the RTLS method using either the fast QEP solver [80] or the Arnoldi method for the linearized eigenvalue problem are quite robust with respect to the chosen starting vector. For many of the problems, there is no need to seek for a certain initial vector, because random vectors satisfy the condition required for convergence of the method. To a certain extent, the convergence depends also on the problem dimensions (*i.e.*, square or rectangular) and on the noise level. In Table 3.9, percentages of ‘good’ solutions for the test problem `ilaplace` using the solver `frtlsqep` are shown. The tolerances to which the relative errors were compared are also shown.

## 3.5 Conclusions

From both theoretical and practical points of view, the regularized total least squares problem is a necessary extension of the regularized least squares formulation. At present, there are several proposed methods for solving the RTLS problem. In this chapter, we surveyed them in the chronological order of publishing. At the beginning, the problem was analyzed more from the computational point of view, and it took a few years until it was shown that

**Table 3.9.** (a) Percentage of `firtlsqep` solutions close to (within a given tolerance) the exact solution in example `ilaplace` with  $n = 30$ , for several noise levels  $\sigma$  and dimensions  $m$ , in 1000 runs with different random starting vectors `rand(n, 1)`.  
 (b) Tolerances for the relative errors between `firtlsqep` solutions and the exact solution.

$\sigma \setminus \frac{m}{n}$	1	5	10	20
0.01	100%	99.9%	98.5%	96.6%
0.1	100%	99.5%	98.1%	96.8%
1	0%	99.7%	98.8%	96.1%
2	0%	99.5%	99.1%	96.6%

$\sigma \setminus \frac{m}{n}$	1	5	10	20
0.01	5e-4	1e-4	1e-4	5e-5
0.1	5e-4	1e-4	1e-4	1e-4
1	1e+0	5e-2	5e-3	5e-4
2	1e+0	5e-2	1e-2	1e-3

what was thought to be a complicated nonconvex optimization problem can be equivalently solved with optimization techniques that are guaranteed to reach the global minimum.

We spent more details to explain our own RTLSQEP method, which involves solving iteratively quadratic eigenvalue problems. An advantage is that either standard or fast methods can be used for solving the specific QEPs. All options were numerically tested and validated in Matlab implementations of the method. Dense problems with dimensions up to several thousands of rows and columns could be solved using the Arnoldi method applied to the linearized QEPs. An important remark drawn from numerical experiments is that the RTLSQEP method is *robust*: the initial vector of the iterative algorithm can be arbitrarily chosen in most of the cases. This remark has, in fact, a solid theoretical ground, in view of the analysis in [7]. The RTLSQEP method is equivalent to the one-scalar minimization problem of [7] (see §3.2.4).

The drawback of most of these methods is that they require an exact specification of the constraint parameter  $\delta$  or of the regularization parameter  $\lambda$ . In real-life problems, such a parameter is rarely *a priori* available, therefore it should be estimated from given data, using, for instance, cross validation. Discussion around optimal choices of regularized models is the topic of the next chapter, where we focus our attention on both truncation and penalty regularization techniques for the total least squares problem.



## Chapter 4

# Model selection for regularized errors-in-variables linear systems

In the context of errors-in-variables ill-conditioned linear models, regularization techniques where a hyperparameter is present, are often employed. In this chapter it is shown that the error function used by the model selection criteria for choosing a good regularization/truncation parameter must be based on the *generalization error* instead of the *prediction error*, which is used in ordinary linear regression. This observation leads to new model selection criteria that are based on orthogonal distances. For the case of the cross validation method, a consistency theorem is also proven. We discuss model selection methods in the context of choosing a truncation level for truncated total least squares and choosing a regularization parameter for regularized total least squares. Numerical experiments sustain the superiority of the generalization error approaches in comparison with classical methods for selecting regularization parameters.

### 4.1 Introduction

We consider again the slightly incompatible ill-conditioned linear system,

$$Ax \approx b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m, \quad x \in \mathbb{R}^n.$$

When a certain regularization method (either truncation or penalty-based) is used for this problem, we need methods for choosing a good hyperparameter (truncation level or regularization parameter, respectively). These selection methods should be based on the goal that the regularized solution is an appropriate solution for  $Ax \approx b$ , but the size of the penalty term  $\|Lx\|^2$  or the complexity of the model are also kept under control.

In this chapter, new techniques for hyperparameter selection in the context of the errors-in-variables model are proposed. They are based on the classical model selection criteria. Section 4.2 highlights that classical criteria based on a *prediction error* loss function are not appropriate for errors-in-variables models. Instead, modified criteria based on a *generalization error* loss function are proposed. It is then proven that the latter criterion can be defined based on orthogonal distances.

For the simple case of the cross validation criterion, we prove in Section 4.3 an important property, similar to the consistency property of the total least squares solution.

That is: for a growing degree of overdetermination (*i.e.*,  $m$  increasingly large) this criterion provides a consistent estimator of the *best regularized solution*.

Section 4.4 is devoted to discussing methods for truncation level selection, whereas Section 4.5 proposes methods for choosing the regularization parameter in RTLS.

## 4.2 Loss function for errors-in-variables linear models

In this section, it will be assumed that the linear model generating the incompatible system  $Ax \approx b$  is an *errors-in-variables model*. Specifically, it is assumed that there exists an exact linear relation  $A^0 x^0 = b^0$  and that  $A$  and  $b$  are perturbed measurements of  $A^0$  and  $b^0$ ,  $A = A^0 + \tilde{A}$ ,  $b = b^0 + \tilde{b}$ ; moreover, the elements of  $[\tilde{A} \ \tilde{b}]$  are independent, identically distributed, with zero mean.

The case of interest is when  $A^0$  is ill-conditioned, and the noisy data matrix  $A$  (although it is *better conditioned* than  $A^0$ ) is also ill-conditioned, with no significant gap in the singular values.

### 4.2.1 Prediction error vs. generalization error

Let  $[A \ b]$  be noisy measured data from an errors-in-variables model and let  $\hat{x}$  be an arbitrary estimator of  $x^0$ . In this context, it is important to make a distinction between *prediction* and *generalization errors*. Let  $[C \ d]$  denote a row (or several rows) of measured data from the same errors-in-variables model. In the context of prediction, the estimator  $\hat{x}$  is used to compute  $\hat{d} := C\hat{x}$ , which is a predicted value for  $d$ .

**Definition 4.1.**  $\|d - \hat{d}\|_2^2$  is called the *prediction error*.

Let  $\Delta d := \hat{d} - d$ . Due to noise in  $[C \ d]$  it is probable that  $C\hat{x} = d$  is not readily satisfied. Using  $\Delta d$ , a compatible system is constructed:

$$C\hat{x} = d + \Delta d.$$

Note that this system corrects only the right-hand side, whereas the noisy matrix  $C$  remains unaltered. A redundant way of defining  $\Delta d$  is via the trivial (feasible set is a singleton) optimization problem:

$$\min_{\Delta d} \|\Delta d\|_2^2 \quad \text{subject to } C\hat{x} = d + \Delta d.$$

This optimization problem is introduced for comparison with the following similar problem, which is related to the *generalization error*:

$$\min_{\Delta C, \Delta d} \|\begin{bmatrix} \Delta C & \Delta d \end{bmatrix}\|_F^2 \quad \text{subject to } (C + \Delta C)\hat{x} = d + \Delta d. \quad (4.1)$$

In words, the optimal solution of problem (4.1) consists of the smallest corrections that must be added to  $[C \ d]$  in order to make the equation  $C\hat{x} \approx d$  compatible. Let  $\begin{bmatrix} \widehat{\Delta C} & \widehat{\Delta d} \end{bmatrix}$

be the optimal solution of (4.1) and define  $\widehat{C} := C + \Delta C$  and  $\widehat{d} := d + \Delta d$ . Then  $\widehat{C}\widehat{x} = \widehat{d}$  is satisfied.

**Definition 4.2.**  $\| [C \ d] - [\widehat{C} \ \widehat{d}] \|_F^2$  is called the *generalization error*.

### 4.2.2 Model selection based on prediction or generalization error

The model selection criteria in §1.3 use a certain error function  $\mathcal{L}$  in order to assess the performance of the regularized models. Applied to the errors-in-variables context, two error functions are compared in the following. They are based on the prediction error and the generalization error, respectively.

**Definition 4.3.** The *prediction error function* is defined as

$$\mathcal{L}^{pred}(\widehat{x}, [C \ d]) := \|C\widehat{x} - d\|_2^2;$$

the *generalization error function* is

$$\mathcal{L}^{gen}(\widehat{x}, [C \ d]) := \frac{\|C\widehat{x} - d\|_2^2}{\|\widehat{x}\|_2^2 + 1}.$$

The justification of the previous definition is straightforward in the case of  $\mathcal{L}^{pred}$ ; for  $\mathcal{L}^{gen}$ , it is clarified by the following lemma (see also [132, Thm. 6.5]):

**Lemma 4.4.** The optimal solution of the optimization problem (4.1) is given by

$$\widehat{\Delta C} = -\frac{(C\widehat{x} - d)\widehat{x}^\top}{\|\widehat{x}\|_2^2 + 1}, \quad \widehat{\Delta d} = \frac{C\widehat{x} - d}{\|\widehat{x}\|_2^2 + 1}, \quad (4.2)$$

and the optimal value of the generalization error is the sum of orthogonal distances,  $\frac{\|C\widehat{x} - d\|_2^2}{\|\widehat{x}\|_2^2 + 1}$ .

**Proof.** Defining the Lagrangian of (4.1) as

$$\mathbf{L}(\Delta C, \Delta d, v) = \| [\Delta C \ \Delta d] \|_F^2 + 2v^\top ((C + \Delta C)\widehat{x} - d - \Delta d)$$

(with  $v$  - the vector of Lagrange multipliers), the formulas (4.2) are easily derived from the first order optimality conditions:

$$\Delta C = -v\widehat{x}^\top, \quad \Delta d = v, \quad (C + \Delta C)\widehat{x} = d + \Delta d.$$

□

The two error functions,  $\mathcal{L}^{pred}$  and  $\mathcal{L}^{gen}$ , lead to different definitions of the model selection criteria introduced in §1.3. In particular, if we let  $\lambda$  denote the regularization parameter

(or truncation level) that characterizes  $\widehat{x}$  as a regularized model depending on  $\lambda$ , then we can write the following two cross validation functions:

$$CV^{pred}(\lambda) = \frac{1}{c} \sum_{j=1}^c \|A_{I_j} \widehat{x}_{-I_j}(\lambda) - b_{I_j}\|_2^2, \quad CV^{gen}(\lambda) = \frac{1}{c} \sum_{j=1}^c \frac{\|A_{I_j} \widehat{x}_{-I_j}(\lambda) - b_{I_j}\|_2^2}{\|\widehat{x}_{-I_j}(\lambda)\|_2^2 + 1}, \quad (4.3)$$

where, as in §1.3,  $\{I_1, \dots, I_c\}$  is a partition of the set of indices  $\{1, \dots, m\}$  into  $c$  disjoint sets, and  $A_{I_j}, b_{I_j}$  denote the rows corresponding to indices in  $I_j$ ,  $A_{-I_j}, b_{-I_j}$  denote the rows corresponding to indices in  $\{1, \dots, m\} \setminus I_j$ , and  $\widehat{x}_{-I_j}(\lambda)$  is a regularized solution computed only with the data  $A_{-I_j}, b_{-I_j}$ .

Note that  $CV^{pred}$  is identical to the function  $CV_{RLS}$  in (1.11) on page 12.

### 4.2.3 Optimal regularization parameter

It should be noted at this point that, depending on the definition of a regularized model  $\widehat{x}(\lambda)$  (which means, e.g.,  $\widehat{x}(\lambda) := x^{\text{Tik}}(\lambda) = (A^\top A + \lambda L^\top L)^{-1} A^\top b$  in the Tikhonov regularization case), there may be several definitions of the “optimal” regularized model (or, respectively, optimal regularization parameter  $\lambda$ ). For the numerical experiments in §4.3.3, the optimal  $\lambda$  was defined as the minimizer of  $\|\widehat{x}(\lambda) - x^0\|_2$ . Another choice might be, for instance, the minimizer of the angle between  $\widehat{x}(\lambda)$  and  $x^0$ .

Assuming that minimizing  $\|\widehat{x}(\lambda) - x^0\|_2$  is a good criterion for obtaining a meaningful solution, let  $\lambda^{opt}$  be the optimal regularization parameter. (Note that  $\lambda^{opt}$  can be computed effectively *only* in simulation examples, when  $x^0$  is known.) Clearly, any method for choosing  $\lambda$  cannot give a better regularized solution under this criterion. Therefore, the aim is to find a  $\lambda$  that gives an  $\widehat{x}(\lambda)$  as close to  $\widehat{x}(\lambda^{opt})$  as possible.

## 4.3 Consistent cross validation based on the generalization error

### 4.3.1 Consistency theorem

In the case of the cross validation method, we prove an interesting consistency result. It shows that it is appropriate to use the criterion of minimizing  $CV^{gen}$ , instead of  $CV^{pred}$ , whenever the true model is an errors-in-variables model. Theorem 4.9 below is closely related to the consistency discussion for the least squares and total least squares solutions [132, Chapter 8].

Let  $[A \ b] \in \mathbb{R}^{m \times (n+1)}$  be noisy data from an errors-in-variables model, for which  $[A \ b] = [A^0 \ b^0] + [\tilde{A} \ \tilde{b}]$ , with all elements in  $[\tilde{A} \ \tilde{b}]$  independent identically distributed, with zero mean and variance  $\sigma^2$ , and assume there exists  $x^0 \in \mathbb{R}^n$  such that  $A^0 x^0 = b^0$ . As before, consider a partition of the set of indices  $\{1, \dots, m\}$  into  $c$  disjoint sets  $\{I_1, \dots, I_c\}$ , each of size  $p$ . (This implies the condition  $m = pc$ , which is not a necessary restriction, but it is used to simplify notation.)

Before presenting the actual theorem, we state a couple of auxiliary results. For the first auxiliary result, the definition of the cross validation function for an arbitrary error function  $\mathcal{L}$  must be contrasted with the definition of the *conditional risk* function [26]:

**Definition 4.5.**

i. The cross validation function is

$$CV(\lambda) = \frac{1}{c} \sum_{j=1}^c \mathcal{L}(\hat{x}_{-I_j}(\lambda), [A_{I_j} \quad b_{I_j}]); \quad (4.4)$$

the optimal cross validation parameter is denoted by  $\hat{\lambda} = \arg \min CV(\lambda)$ .

ii. The conditional risk function is

$$\tilde{V}(\lambda) = \frac{1}{c} \sum_{j=1}^c \mathcal{E}_{[\tilde{C} \quad \tilde{d}]} \left[ \mathcal{L}(\hat{x}_{-I_j}(\lambda), [A_{I_j}^0 + \tilde{C} \quad b_{I_j}^0 + \tilde{d}]) \right], \quad (4.5)$$

where  $\mathcal{E}_{[\tilde{C} \quad \tilde{d}]}$  denotes the expectation taken with respect to the common density function of the elements of  $[\tilde{C} \quad \tilde{d}] \in \mathbb{R}^{p \times n}$ , which have the same characteristics as the noise  $[\tilde{A} \quad \tilde{b}]$ ; the optimal conditional risk parameter is denoted by  $\tilde{\lambda} = \arg \min \tilde{V}(\lambda)$ .

Note that the two formulations differ in the fact that the cross validation function uses a particular noise realization  $[\tilde{A}_{I_j} \quad \tilde{b}_{I_j}]$  that is added to the true data  $[A_{I_j}^0 \quad b_{I_j}^0]$ , whereas  $\tilde{V}$  takes the expectation of any possible added noise.  $\tilde{V}$  is uncomputable (since the exact  $[A^0 \quad b^0]$  is unknown), but it is used in the proof, because of the following property proven (in a different context and for a different purpose, however) in [26]:

**Lemma 4.6.** Under certain assumptions (see [26, Thm. 1]),<sup>6</sup>

$$\lim_{m \rightarrow \infty} |\tilde{V}(\tilde{\lambda}) - \tilde{V}(\hat{\lambda})| = 0.$$

In other words, the cross validation parameter  $\hat{\lambda}$  is asymptotically optimal for  $\tilde{V}$ . This will allow to replace (at the limit  $m \rightarrow \infty$ , i.e., when the row dimension of the data matrix  $[A \quad b]$  grows to infinity) the minimization of  $V$  with the minimization of  $\tilde{V}$ , in order to prove the properties of the cross validation parameter.

Another replacement that may be done in the limit is:  $\hat{x}_{-I_j}(\lambda)$  by  $\hat{x}(\lambda)$ , where  $\hat{x}(\lambda)$  is computed from all the  $m$  rows of the given data  $[A \quad b]$ .

**Lemma 4.7.** If

$$\lim_{m \rightarrow \infty} \frac{\sigma_{\min}(A^T A)}{\sigma_{\max}(A_{I_j})} = \infty, \quad \forall j \in \{1, \dots, c\}, \quad (4.6)$$

then  $\lim_{m \rightarrow \infty} \|\hat{x}(\lambda) - \hat{x}_{-I_j}(\lambda)\|_2 = 0, \quad \forall j \in \{1, \dots, c\}$ .

<sup>6</sup>The technical assumptions are not listed here, to save space. Among these assumptions, the most troublesome is that [26] allows only a finite number of models to select from. In the present context, this is fine for choosing truncation levels, but for regularization parameters it implies that  $\lambda$  should be constrained to belong to a discrete set.

We show the proof in the case of Tikhonov solutions. It follows from the expansion

$$\begin{aligned}\widehat{x}_{-I_j}(\lambda) &= (A^\top A - A_{I_j}^\top A_{I_j} + \lambda L^\top L)^{-1} (A^\top b - A_{I_j}^\top b_{I_j}) \\ &= \widehat{x}(\lambda) + (A^\top A + \lambda L^\top L)^{-1} A_{I_j}^\top (A_{I_j} \widehat{x}_{-I_j}(\lambda) - b_{I_j}),\end{aligned}$$

by bounding from above the norm  $\|\widehat{x}(\lambda) - \widehat{x}_{-I_j}(\lambda)\|_2$  with a term proportional to  $\frac{\sigma_{\max}(A_{I_j})}{\sigma_{\min}^2(A)}$ . From (4.6), this ratio goes to zero as  $m$  goes to infinity.

The last auxiliary result is:

**Lemma 4.8.** *If  $C = C^0 + \tilde{C} \in \mathbb{R}^{p \times n}$  and  $d = d^0 + \tilde{d} \in \mathbb{R}^p$ , and all elements of  $[\tilde{C} \ \tilde{d}]$  are i.i.d., have zero mean and variance  $\sigma^2$ , then, for any  $x \in \mathbb{R}^n$ ,*

$$\mathcal{E}_{[\tilde{C} \ \tilde{d}]} [\|Cx - d\|_2^2] = \|C^0 x - d^0\|_2^2 + p\sigma^2(\|x\|_2^2 + 1). \quad (4.7)$$

This follows clearly:

$$\mathcal{E} [\|Cx - d\|_2^2] = \|C^0 x - d^0\|_2^2 + \mathcal{E} [\|\tilde{C}x - \tilde{d}\|_2^2] = \|C^0 x - d^0\|_2^2 + p\sigma^2(\|x\|_2^2 + 1).$$

The consistency theorem is:

**Theorem 4.9.** *Let  $\lambda^{pred}$  and  $\lambda^{gen}$  be the minimizers of the cross validation functions  $CV^{pred}$  and  $CV^{gen}$ , respectively, and let  $\lambda^{opt}$  be the optimal regularization parameter, i.e., the minimizer of  $\|\widehat{x}(\lambda) - x^0\|_2$ . If*

$$\lim_{m \rightarrow \infty} \sigma_{\min}(A^\top A) = \infty, \quad \text{and} \quad \lim_{m \rightarrow \infty} \frac{\sigma_{\min}(A^\top A)}{\sigma_{\max}(A_{I_j})} = \infty, \quad \forall j \in \{1, \dots, c\}, \quad (4.8)$$

and

$$\exists \lim_{m \rightarrow \infty} \frac{A^0{}^\top A^0}{m} =: F \in \mathbb{R}^{n \times n}, \quad (4.9)$$

then, as  $m \rightarrow \infty$ ,

- a.  $\|\widehat{x}(\lambda^{pred}) - \widehat{x}(\lambda^{opt})\|_2$  is asymptotically biased away from zero;
- b.  $\|\widehat{x}(\lambda^{gen}) - \widehat{x}(\lambda^{opt})\|_2 \rightarrow 0$ .

**Proof.** *Proof of point a:* From Lemma 4.6, the optimal  $\lambda^{pred}$  can be obtained (when

$m \rightarrow \infty$ ) by minimizing  $\tilde{V}^{pred}$ . This is written as

$$\begin{aligned} \min_{\lambda} \tilde{V}^{pred}(\lambda) &= \min_{\lambda} \frac{1}{c} \sum_{j=1}^c \mathcal{E}_{[\tilde{c} \ \tilde{d}]} \left[ \|(A_{I_j}^0 + \tilde{C})\hat{x}_{-I_j}(\lambda) - (b_{I_j}^0 + \tilde{d})\|_2^2 \right] \\ &\approx \min_{\lambda} \frac{1}{c} \sum_{j=1}^c \left[ \|A_{I_j}^0 \hat{x}(\lambda) - b_{I_j}^0\|_2^2 + p\sigma^2(\|\hat{x}(\lambda)\|_2^2 + 1) \right] \end{aligned} \quad (4.10)$$

$$\begin{aligned} &= \min_{\lambda} \frac{1}{c} \|A^0 \hat{x}(\lambda) - b^0\|_2^2 + p\sigma^2(\|\hat{x}(\lambda)\|_2^2 + 1) \\ &= \frac{1}{c} \min_{\lambda} \left\| \begin{bmatrix} A^0 \\ \sqrt{m\sigma^2} I_n \\ 0 \end{bmatrix} \hat{x}(\lambda) - \begin{bmatrix} b^0 \\ 0 \\ \sqrt{m\sigma^2} \end{bmatrix} \right\|_2^2. \end{aligned} \quad (4.11)$$

Line (4.10) follows from Lemma 4.8, but the approximation sign is used because  $\hat{x}_{-I_j}$  is replaced by  $\hat{x}(\lambda)$  (Lemma 4.7). The minimization (4.11) is a constrained least squares problem (the constraint being represented by the parameterization  $\hat{x}(\lambda)$ ). In the unconstrained situation (*i.e.*,  $\hat{x}(\lambda)$  replaced by a free  $x$ ) the least squares solution is given by  $x_m^{LS} = (A^{0\top} A^0 + m\sigma^2 I_n)^{-1} A^{0\top} b^0$ , which is a biased estimator of  $x^0$ . It is easy to show that the solution  $\hat{x}(\lambda^{pred})$  of (4.11) equals also

$$\arg \min_{\hat{x}(\lambda)} \left\| \begin{bmatrix} A^0 \\ \sqrt{m\sigma^2} I_n \end{bmatrix} (x_m^{LS} - \hat{x}(\lambda)) \right\|_2^2 = \arg \min_{\hat{x}(\lambda)} \left\{ \|A^0 (x_m^{LS} - \hat{x}(\lambda))\|_2^2 + m\sigma^2 \|x_m^{LS} - \hat{x}(\lambda)\|_2^2 \right\}.$$

As  $m \rightarrow \infty$ , the second term in this function becomes dominant and it implies that  $\hat{x}(\lambda^{pred})$  will tend to be as close as possible to  $x_m^{LS}$ . Note that  $x_m^{LS} = x^0 + \left( \frac{A^{0\top} A^0}{m} + \sigma^2 I_n \right)^{-1} x^0$ ; therefore, as  $m \rightarrow \infty$ ,  $\lambda^{pred}$  is the minimizer of

$$\|\hat{x}(\lambda) - x_{\infty}^{LS}\|_2 = \|\hat{x}(\lambda) - x^0 - (F + \sigma^2 I_n)^{-1} x^0\|_2$$

with  $F$  defined in (4.9).

On the other hand, we have that  $\lambda^{opt}$  as the solution of the minimization of  $\|\hat{x}(\lambda) - x^0\|$ . It follows that  $\lambda^{opt}$  cannot be, at the limit, equal to  $\lambda^{pred}$ , and the bias between the prediction error model and the optimal regularized model is therefore proven.

*Proof of point b:* Lemmas 4.6–4.8 help writing the minimization of  $\tilde{V}^{gen}$  as

$$\min_{\lambda} \frac{1}{c} \frac{\|A^0 \hat{x}(\lambda) - b^0\|_2^2}{\|\hat{x}(\lambda)\|_2^2 + 1} + p\sigma^2 \quad \iff \quad \min_{\lambda} \frac{\|A^0 \hat{x}(\lambda) - b^0\|_2^2}{\|\hat{x}(\lambda)\|_2^2 + 1}.$$

In the unconstrained case ( $\hat{x}(\lambda)$  replaced by a free  $x$ ), this problem is a trivial (noiseless) TLS problem, which yields the exact solution  $x^0$ . In the singular value decomposition of  $\begin{bmatrix} A^0 & b^0 \end{bmatrix}$ , the largest  $n$  singular values go to infinity in the limit (see (4.8)); the smallest one is 0 and corresponds to the right singular vector  $\begin{bmatrix} x^0 \\ -1 \end{bmatrix} / \left\| \begin{bmatrix} x^0 \\ -1 \end{bmatrix} \right\|_2$ . This implies that the optimal solution  $\hat{x}(\lambda^{gen})$  should be as close as possible to  $x^0$ . From the definition of  $\lambda^{opt}$ , it follows  $\lim_{m \rightarrow \infty} \|\hat{x}(\lambda^{gen}) - \hat{x}(\lambda^{opt})\|_2 = 0$ .  $\square$

### 4.3.2 Computational properties

In what follows, the cross validation function based on the *generalization error*,  $CV^{gen}(\lambda)$ , will be denoted by  $CV(\lambda)$  and will be referred to as the “new cross validation” function.

From a computational point of view, the minimization of  $CV(\lambda)$  might pose problems sometimes, such as multiple local minima or a global minimum at 0. Nevertheless, our examples show that this behavior is less frequent than in the case when we use criteria based on the prediction error, such as the classical generalized cross validation function, applied to errors-in-variables problems.

#### Comparison between partitioning methods

In a statistical framework for the cross validation method, there are several established ways of choosing the partition. Among them: leave-one-out CV, V-fold (or leave-many-out) CV, Monte-Carlo CV, bootstrap CV. Here not all these methods will be considered; we refer to [26] for recent results on an unified methodology for cross validation.

The way in which the partition is designed might be influenced by knowledge of the problem at hand. In general, choosing the partition involves also a *bias-variance trade-off*.

In the following, two extreme partitioning cases will be used: the leave-one-out and the 2-fold cross validation; they will be denoted by **LOO** and **2-fold** partitions.

The leave-one-out cross validation is the most general (and classical) choice; it uses singleton subsets,  $I_j = \{j\}$ , for any index  $j$ . This method might be too computationally expensive and it might suffer from over-fitting.

The other simple choice, in the other “extreme”, is to split the data only in two parts. For instance  $I_1 = \{1, 3, 5, \dots\}$ ,  $I_2 = \{2, 4, 6, \dots\}$  proves to be an excellent and cheap choice when the data comes from a continuous problem.

### 4.3.3 Numerical illustration of the consistent cross validation

The numerical examples in this section are using ill-posed problems from the Regularization Tools [57].<sup>7</sup> Each test problem provides the exact data  $[A^0 \ b^0]$  as well as exact solution  $x^0$ . For several noise levels  $\sigma$ , white noise is added to the exact data as:

$$A = A^0 + \sigma E \frac{\|A^0\|_F}{\|E\|_F}, \quad b = b^0 + \sigma e \frac{\|b^0\|_2}{\|e\|_2},$$

where the matrix  $E$  and the vector  $e$  have independent normally distributed random elements with zero mean and variance 1. The regularization matrix  $L$  is set to the approximation of the first order derivative operator,  $L \in \mathbb{R}^{(n-1) \times n}$ , which is a bidiagonal matrix with -1 on the diagonal and 1 on the first superdiagonal.

The “new cross validation” method for errors-in-variables models (that is, the minimization of  $CV^{gen}(\lambda)$  from (4.3)) is compared with two of the most popular regularization parameter selection methods: L-curve and generalized cross validation (both implemented in the Regularization Tools [57]). The chosen regularization method that we use on these problems is Tikhonov regularization (in the least squares, not the total least squares sense).

<sup>7</sup>Some of the problems are modified in order to construct rectangular data matrices instead of only square ones.

**Table 4.1.** Comparison of average relative errors between Tikhonov-regularized solutions and exact solution  $x^0$  ( $\|x^{\text{Tik}} - x^0\|/\|x^0\|$ ) for three methods for computing the regularization parameter. Several problems from the Regularization Tools, each with 100 noisy realizations (with noise level  $\sigma = 0.1$ ), are used.

Problem	$m, n$	L-curve	GCV	new CV
ilaplace	[100, 20]	0.089	0.420	<b>0.058</b>
baart	[100, 20]	0.433	<b>0.304</b>	0.385
shaw	[40, 40]	1.038	0.659	<b>0.496</b>
phillips	[40, 40]	<b>0.120</b>	0.408	0.197
foxgood	[40, 40]	1.354	0.473	<b>0.172</b>
deriv2	[100, 20]	0.165	0.378	<b>0.147</b>
regutm	[100, 20]	0.431	0.517	<b>0.420</b>

Note that the L-curve and GCV implementations are designed for Tikhonov regularization of the least squares problem, and they do not assume that  $A$  is also noisy. Results are reported in Table 4.1. It can be noted that the relative errors obtained with the new cross validation method are comparable with, and usually smaller than, those for the other methods.

Another experiment illustrates the *consistency* property of the new cross validation criterion. Random problems of growing size  $m$  are used. The matrix  $A^0$  is generated with the function `regutm` from the Regularization Tools; thus,  $A^0$  is ill-conditioned, with exponentially decaying singular values, and random (left and right) singular vectors. The exact solution is set to  $x^{0\top} = \left( \left(\frac{1}{n}\right)^2, \left(\frac{2}{n}\right)^2, \dots, \left(\frac{n}{n}\right)^2 \right)$ , and  $b^0$  is computed as  $b^0 = A^0 x^0$ . White noise is added to  $[A^0 \ b^0]$  in order to obtain  $[A \ b]$ ; 100 different noise realizations are then used to compute average relative errors.

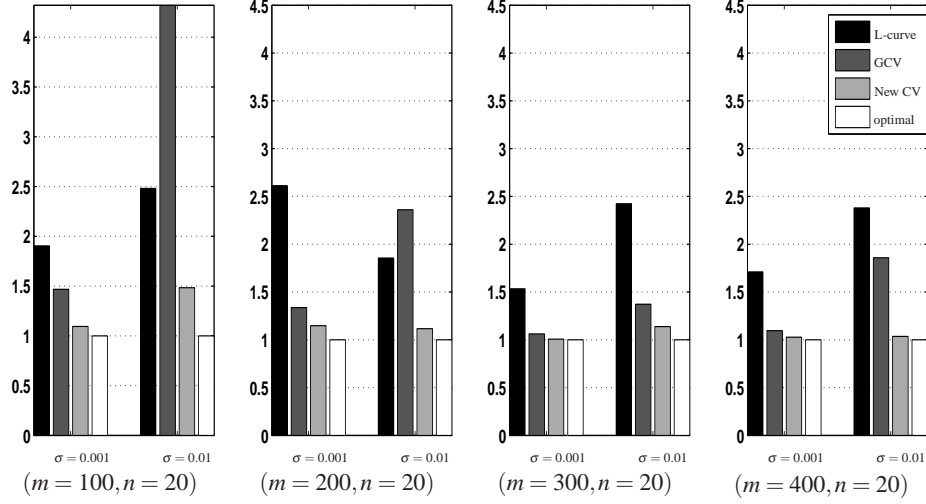
Figure 4.1 shows the behavior of the average relative errors obtained with three methods for computing the regularization parameter. For reference, the best possible error (obtained for the  $\lambda$  that minimizes  $\|x^{\text{Tik}}(\lambda) - x^0\|_2$ ) is shown. The experiment demonstrates that with increasing  $m$  the new cross validation estimator performs better and better, while the other estimators don't have this consistency property.

## 4.4 Methods for choosing truncation levels

In this section, we discuss several methods for choosing the truncation level. For the truncated SVD formulation, many classical model selection methods are easily adapted. Our contribution is to specialize some of the classical methods for truncation parameter selection to the other classes of ScTLS problems; in particular, we focus on the TLS formulation.

In the following, we denote by  $\hat{x}_{\text{TSVD},k}$  the truncated SVD solution of the problem  $Ax \approx b$  with truncation level  $k$ , and by  $\hat{x}_{\text{TTLS},k}$  the truncated TLS solution with truncation level  $k$ . (See §2.2.2.) These  $n$ -dimensional vectors satisfy exactly the following truncated systems:

$$A\hat{x}_{\text{TSVD},k} = \hat{b}'_k, \quad \hat{A}_k \hat{x}_{\text{TTLS},k} = \hat{b}_k,$$



**Figure 4.1.** Average relative errors between Tikhonov-regularized solutions and exact solution  $x^0$  when the regularization parameter is computed using the L-curve, the generalized cross validation criterion, the new cross validation for errors-in-variables and the optimal regularization parameter. The latter is computed by minimizing the Euclidean distance between the regularized solution and the exact solution  $x^0$ . All values are scaled by dividing to the corresponding minimal average relative error (the fourth bar). This means that the bars that approach the value 1 indicate nearly optimal regularized solution.

where  $\hat{b}'_k := U'_k U'^{\top}_k b$  (the  $m \times k$  matrix  $U'_k$  contains the first  $k$  left singular vectors of  $A$ ), and  $\begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix}$  is a certain rank  $k$  approximation of  $\begin{bmatrix} b & A \end{bmatrix}$ .

Let  $r_k$  be the residual error obtained for the truncation level  $k$ . The expression of  $r_k$  depends on the formulation that we use:

$$r_k^{\text{LS}} = \|b - \hat{b}'_k\|_2, \quad \text{for the LS case,}$$

$$r_k^{\text{TLS}} = \left\| \begin{bmatrix} b & A \end{bmatrix} - \begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix} \right\|_F, \quad \text{for the TLS case.}$$

Another variable of interest is  $N$  – the total number of elements in  $\begin{bmatrix} b & A \end{bmatrix}$  that are assumed noisy,

$$N^{\text{LS}} = m, \quad \text{for the LS case,}$$

$$N^{\text{TLS}} = m(n+1), \quad \text{for the TLS case.}$$

Computing the residual error  $r_k$  is needed in all model selection methods. For the LS case, this is straightforward. We focus now on the TLS case, since we did not yet specify what  $\hat{A}_k$  and  $\hat{b}_k$  are.

We assume that the truncated solution  $\hat{x}_{\text{TLS},k}$  is computed using the partial reduction to a core problem, as described in Chapter 2, Algorithm 2.3. Thus,  $\begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix}$  is not nec-

essarily the best rank  $k$  approximation of  $[b \ A]$ . This happens in the case when some of the largest  $k$  singular values of  $A$  are multiple or when  $b$  is orthogonal onto some of the left singular subspaces of  $A$ . Consider the truncated core reduction (see (2.2) on page 29)

$$[b_1^k \ A_{11}^k] = P_1^{k\top} [b \ A Q_1^k] \quad (4.12)$$

with  $[b_1^k \ A_{11}^k] \in \mathbb{R}^{(k+1) \times (k+1)}$  and  $P_1^k \in \mathbb{R}^{m \times (k+1)}$ ,  $Q_1^k \in \mathbb{R}^{n \times k}$  having orthonormal columns; let  $(u'', \sigma'', v'')$  be the smallest (that is, the  $k+1$ st) singular triplet of  $[b_1^k \ A_{11}^k]$ . Then we have [95]

$$\begin{bmatrix} \widehat{b}_k & \widehat{A}_k \end{bmatrix} = [b \ A] - P_1^k u'' \sigma'' v_1'' \begin{bmatrix} -1 & \widehat{x}_{\text{TTLs},k}^\top \end{bmatrix},$$

where  $v_1''$  is the first element of the vector  $v''$ . It is not difficult to reformulate this expression and to obtain the corrections in their typical formula (4.2) as:

$$\begin{bmatrix} \widehat{b}_k & \widehat{A}_k \end{bmatrix} = [b \ A] - \frac{(A \widehat{x}_{\text{TTLs},k} - b) \begin{bmatrix} -1 & \widehat{x}_{\text{TTLs},k}^\top \end{bmatrix}}{\|\widehat{x}_{\text{TTLs},k}\|^2 + 1}.$$

The proof involves the relation (4.12) and point (c) of the following lemma:

**Lemma 4.10.** *The core problem  $A_{11}^k y \approx b_1^k$  with optimal TLS solution  $x_1^k$ , and the smallest singular triplet of  $[b_1^k \ A_{11}^k]$ , denoted by  $(u'', \sigma'', v'')$ , satisfy:*

(a) $\frac{\ A_{11}^k x_1^k - b_1^k\ _2^2}{\ x_1^k\ _2^2 + 1} = (\sigma'')^2$	(b) $\ x_1^k\ _2^2 + 1 = (v_1'')^{-2}$
(c) $\frac{(A_{11}^k x_1^k - b_1^k)}{\ x_1^k\ _2^2 + 1} = u'' \sigma'' v_1''$	(d) $\frac{A_{11}^{k\top} (A_{11}^k x_1^k - b_1^k)}{\ x_1^k\ _2^2 + 1} = -(\sigma'')^2 (v_1'')^2 x_1^k$

Moreover, the  $n$ -dimensional vector  $\widehat{x}_{\text{TTLs},k}$  is linked to the  $k$ -dimensional  $x_1^k$  by the relation:  $\widehat{x}_{\text{TTLs},k} = Q_1^k x_1^k$ .

If the reduction in (4.12) is a partial bidiagonalization, then we have a way of computing the residual norm efficiently at each bidiagonalization step. The following identity can be used:

$$r_k^{\text{TLS}} = \left\| [b \ A] - \begin{bmatrix} \widehat{b}_k & \widehat{A}_k \end{bmatrix} \right\|_F^2 = \| [b \ A] \|_F^2 - \| [\beta_1 e_1 \ A_{11}^k] \|_F^2 + (\sigma'')^2$$

where  $[b_1^k \ A_{11}^k] = [\beta_1 e_1 \ A_{11}^k]$  is the bidiagonal matrix obtained after  $k$  steps. This identity is proven in [33].

We discuss next how the model selection methods from Chapter 1, §1.3, can be used in the context of choosing a truncation level in TSVD or TTLs.

### The discrepancy principle

This method requires knowledge about the statistical properties of the noise that perturbs the data. Specifically, let's assume that the noise is i.i.d. with zero mean and variance  $\sigma^2$ . The norm of the residual error is computed for the candidate truncation level  $k$  and the following criterion is monitored:

$$\min_k \left| \|r_k\|^2 - \sigma^2 N \right|,$$

thus, a  $k$  is sought that provides a residual error of the same magnitude as the noise in the data.

### The L-curve

The norm of the truncated solution  $\|\hat{x}_k\|_2$  is plotted against the residual error norm  $r_k$  in log-log scale for various  $k$ 's, and the *corner* in log-log scale is chosen. A version of the *rotated L-curve* [102] can also be designed.

### Generalized cross validation

Generalized cross validation can be written as

$$\min_k \frac{\|r_k\|_F^2}{(N - p_k^{\text{eff}})^2}, \quad (4.13)$$

where the *effective number of parameters*  $p_k^{\text{eff}}$  is the trace of the generalized information matrix (see §1.3.3 and the Appendix A.1), which is the derivative of the reconstructed data model with respect to the noisy data.

**In the LS case**, only  $b$  is considered noisy; therefore, the generalized influence matrix shows how perturbations in the measured output  $b$  are reflected as perturbations in the recomputed model  $\hat{b}'_k$ . It becomes

$$S^{\text{LS}}(k) = \frac{\partial \hat{b}'_k}{\partial b} = \frac{\partial U'_k U_k{}^\top b}{\partial b} = U'_k U_k{}^\top,$$

giving the obvious statement that  $p_k^{\text{eff}} = \text{Tr}(U'_k U_k{}^\top) = k$ . Indeed, in a linear regression model, the number of effective parameters is exactly the number of *free* model parameters. This is easily explained in terms of the truncated core problem as follows: in truncated SVD, the core problem has size  $(k+1) \times k$ ; we need to solve it in least squares sense and we obtain a  $k$ -dimensional solution vector  $x_1^k$ . The final solution  $\hat{x}_{\text{TSVD},k}$ , though of length  $n$ , is obtained directly from  $x_1^k$  by multiplication with an orthogonal transformation matrix (which does not depend on  $b$ ). Thus, the number of effective parameters is  $k$ .

We recover the GCV criterion as the minimization with respect to  $k$  of the function:

$$GCV^{\text{TSVD}}(k) = \frac{\|Ax_{\text{TSVD},k} - b\|^2}{m-k} = \frac{\|(I - U'_k U_k{}^\top)b\|^2}{m-k} = \frac{(r_k^{\text{LS}})^2}{m-k}.$$

This criterion is well-known, and implemented, for instance, in the Regularization Tools [57].

**In the TLS case**,  $A$  and  $b$  are both considered noisy. The residual error norm in this case is given by

$$r_k^{\text{TLS}} = \left\| \begin{bmatrix} b & A \end{bmatrix} - \begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix} \right\|_F^2 = \frac{\|A\hat{x}_{\text{TLS},k} - b\|_2^2}{\|\hat{x}_{\text{TLS},k}\|_2^2 + 1}.$$

Due to the nonlinearity (in  $\hat{x}_{\text{TLS},k}$ ) of this expression, we need to use the nonlinear generalized influence matrix definition [89] (see also Remark 5 on page 13 and Appendix A.1), which in our case becomes the Jacobian of the estimated model  $\text{vec}[\hat{b}_k \ \hat{A}_k]$  with respect to the noisy data  $\text{vec}[b \ A]$ . It is, thus, an  $m(n+1) \times m(n+1)$  matrix.

We prove in Appendix A.1 that, if this direct differentiation is not easy, it is possible to compute the influence matrix using partial derivatives with respect to some auxiliary model parameter, say a vector  $y$ . Then, the influence matrix is written as  $S^{\text{TLS}}(k) = JH^{-1}J^\top$ , where  $H$  is the Hessian of the objective function in  $y$  (*i.e.*, the loss function that should be minimized in order to obtain optimal model parameters), and  $J$  is the derivative of the final “model” with respect to the model parameter  $y$ .

In the truncated core problem with truncation level  $k$ , we solve the core system  $A_{11}^k y \approx b_1^k$  in TLS sense (see §2.3.2). This means that we minimize the criterion:  $\|A_{11}^k y - b_1^k\|_2^2 / (\|y\|^2 + 1)$  with respect to  $y$ . The optimal solution for  $y$  is denoted by  $x_1^k$ . Therefore, we choose as auxiliary variable the  $k$ -dimensional vector  $y$ , and we evaluate the Hessian at its minimizer  $x_1^k$ .  $J$  is the derivative of the final model  $\begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix}$  with respect to  $y$ , computed also at  $x_1^k$ . Thus,

$$\begin{aligned} S^{\text{TLS}}(k) &:= JH^{-1}J^\top \\ &:= \frac{\partial}{\partial y} \text{vec} \begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix} \Big|_{y=x_1^k} \left( \frac{\partial^2}{\partial y \partial y^\top} \frac{\|A_{11}^k y - b_1^k\|_2^2}{\|y\|_2^2 + 1} \Big|_{y=x_1^k} \right)^{-1} \frac{\partial}{\partial y} \text{vec} \begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix}^\top \Big|_{y=x_1^k}. \end{aligned}$$

Let us first compute  $H$ :

$$H = \frac{2}{\|x_1^k\|_2^2 + 1} \left( A_{11}^{k \top} A_{11}^k - \frac{\|A_{11}^k x_1^k - b_1^k\|_2^2}{\|x_1^k\|_2^2 + 1} I_k - 2 \frac{A_{11}^{k \top} (A_{11}^k x_1^k - b_1^k) x_1^{k \top}}{\|x_1^k\|_2^2 + 1} - 2 \frac{x_1^k (A_{11}^k x_1^k - b_1^k)^\top A_{11}^k}{\|x_1^k\|_2^2 + 1} + 4 \frac{\|A_{11}^k x_1^k - b_1^k\|_2^2}{(\|x_1^k\|_2^2 + 1)^2} x_1^k x_1^{k \top} \right).$$

The relations given in Lemma 4.10 help simplifying this expression and make the computation of  $S^{\text{TLS}}(k)$  efficient. The matrix  $H$  becomes:

$$H = 2(v_1'')^2 \left( A_{11}^{k \top} A_{11}^k - (\sigma'')^2 I_k + 8(\sigma'')^2 (v_1'')^2 x_1^k x_1^{k \top} \right).$$

In the case we used a bidiagonalization method,  $H$  is a small ( $k \times k$ ) tridiagonal plus rank-one matrix. Its inversion is easy! Note that the inverse exists: indeed,  $A_{11}^{k \top} A_{11}^k - (\sigma'')^2 I_k$  is positive definite, because  $\sigma'' < \sigma_{\min}(A_{11}^k)$  (because the truncated core problem has unique generic solution).

Now, we compute the  $m(n+1) \times k$  matrix  $J$ :

$$\begin{aligned} J &= \frac{\partial}{\partial y} \text{vec} \begin{bmatrix} \hat{b}_k & \hat{A}_k \end{bmatrix} \Big|_{y=x_1^k} = \frac{\partial}{\partial y} \text{vec} \left( \begin{bmatrix} b & A \end{bmatrix} - P_1^k u'' \sigma'' v_1'' \begin{bmatrix} -1 & y^\top Q_1^{k \top} \end{bmatrix} \right) \Big|_{y=x_1^k} \\ &= - \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ P_1^k u'' \sigma'' v_1'' \end{bmatrix}}_k \begin{matrix} m \\ mn \end{matrix} \end{aligned}$$

Remember that our goal was to evaluate the number of effective parameters, which is the trace of the influence matrix:  $p_k^{\text{eff}} = \text{Tr}(JH^{-1}J^\top) = \text{Tr}(J^\top JH^{-1})$ . We note that  $J^\top J$  simplifies even more:

$$J^\top J = \left( (P_1^k u'' \sigma'' v_1'')^\top (P_1^k u'' \sigma'' v_1'') \right) \otimes \left( Q_1^k \right) = (\sigma'' v_1'')^2 I_k.$$

Thus, the formula for the effective number of parameters in the truncated core TLS problem becomes:

$$p_k^{\text{eff}} = \frac{1}{2} \text{Tr} \left\{ \left( \frac{A_{11}^k \top A_{11}^k}{(\sigma'')^2} - I_k + 8(v_1'')^2 x_1^k x_1^{k \top} \right)^{-1} \right\}.$$

Finally, we are able to plug in computable formulas for  $r_k^{\text{TLS}}$  and  $p_k^{\text{eff}}$  into the GCV function (4.13).

### Generalized information criteria

A generalized information criterion is based on the same elements that we have already discussed for GCV: the residual norm and the number of effective parameters. (See again §1.3.4 and Appendix A.2.) For some forms of information criteria, the (estimated) noise variance is also required. Note that an estimate for the variance (in the case of Gaussian noise) is  $\frac{1}{N-p_k^{\text{eff}}} r_k^2$ .

We can write GIC as

$$\min_k \log(r_k) + \frac{p_k^{\text{eff}}}{N}. \quad (4.14)$$

For TSVD, the particular  $p_k^{\text{eff}}$  is just the number  $k$  of free parameters in the model, and GIC gives a form of the classical Akaike information criterion,

$$\min_k \log(\|b - A\hat{x}_{\text{TSVD},k}\|_2) + \frac{k}{N}.$$

**Concluding** the model selection techniques for choosing truncation levels, we emphasize the fact that in both the truncated SVD and the truncated TLS problems we obtained closed-forms expressions for each of the classical criteria. For TSVD, these are well-known, but for TTLS these results are new. Their implementation in the CoRe software (see §2.4) still waits for more extensive testing.

## 4.5 Methods for choosing the regularization parameter in RTLS

We consider the Tikhonov formulation of the RTLS problem:

$$\min_x \left( \frac{\|Ax - b\|_2^2}{\|x\|_2^2 + 1} + \lambda \|Lx\|_2^2 \right), \quad \text{for a fixed parameter } \lambda. \quad (4.15)$$

The first order optimality condition for this problem reads

$$\left( A^T A + \lambda (\|x\|_2^2 + 1) L^T L - \frac{\|Ax - b\|_2^2}{\|x\|_2^2 + 1} I \right) x = A^T b. \quad (4.16)$$

As we have discussed in Chapter 3, the RTLS problem does not have closed form solution. Model selection criteria, applied to choosing the regularization parameter in RTLS, will not have closed form expressions, either. This means that in order to compare several regularization parameters, we have to solve several RTLS problems.

A discrepancy principle or an L-curve criterion can be easily designed using the generalization error (*i.e.*, the orthogonal distance) as residual measure. We note that an L-curve criterion for RTLS was already proposed and successfully applied in [103].

In this section, we propose a new technique that is based on choosing a regularization parameter of a regularized least squares problem as first step towards choosing the regularization parameter in the RTLS problem. Interesting enough, we believe that the 2-parameter formulation originally proposed in the first paper on RTLS [42] is a better formulation for getting us started on discussing the regularization parameter selection than the other methods from Chapter 3. We remind that formulation:

$$(A^T A + \lambda_L L^T L + \lambda_I I) x = A^T b, \quad (4.17)$$

where we can identify the  $x$ -dependent formulas of  $\lambda_L$  and  $\lambda_I$  from (4.16):

$$\lambda_L = \lambda (\|x\|_2^2 + 1), \quad \text{and} \quad \lambda_I = -\frac{\|Ax - b\|_2^2}{\|x\|_2^2 + 1}. \quad (4.18)$$

Clearly, it is a requirement that  $\lambda_L$  and  $\lambda_I$  are in such a way that the system (4.17) is well-conditioned. For a fixed  $\lambda_L$ , this implies a restriction on  $\lambda_I$ :

$$|\lambda_I| \ll \sigma_{\min}(A^T A + \lambda_L L^T L). \quad (4.19)$$

Since it can be assumed that  $A$  is (nearly) rank-deficient, a sufficient simple condition is to discard all  $(\lambda_L, \lambda_I)$  for which  $|\lambda_I| \ll \lambda_L \sigma_{\min}^2(L)$ . We argue that due to the fact that  $\lambda_I$  should be relatively small, we can use a criterion for choosing  $\lambda_L$  in the assumption that  $\lambda_I = 0$ .

Let  $x(\lambda_L, \lambda_I)$  be the solution of  $(A^T A + \lambda_L L^T L - \lambda_I I) x = A^T b$ . For  $|\lambda_I| \ll \sigma_{\min}(A^T A + \lambda_L L^T L)$ , note that  $(A^T A + \lambda_L L^T L - \lambda_I I)$  can be seen as a perturbation (with  $-\lambda_I I$ ) of the matrix  $(A^T A + \lambda_L L^T L)$ . From well-known sensitivity results [50, §2.7], one concludes that

$$\|x(\lambda_L, 0) - x(\lambda_L, \lambda_I)\| \leq \frac{\lambda_I}{\sigma_{\min}(A^T A + \lambda_L L^T L) - \lambda_I} \|x(\lambda_L, 0)\|.$$

Since  $|\lambda_I| \ll \sigma_{\min}(A^T A + \lambda_L L^T L)$ , and assuming a reasonable value for  $\|x(\lambda_L, 0)\|$  (because  $x(\lambda_L, 0)$  is a solution of a regularized problem), it is clear that  $x(\lambda_L, 0)$  and  $x(\lambda_L, \lambda_I)$  are quite close to each other.

By fixing  $\lambda_I = 0$ , we obtain a simplified framework for regularization parameter selection for RTLS, by using the model selection methods that are well-known for regularized least squares (Tikhonov regularization). Only one variable,  $\lambda_L$ , is optimized.

---

**Algorithm 4.1** Iterative refinement for the RTLS solution.

---

- 1:  $\lambda_L^0 = \lambda_L^{\min}$ ,  $x^0 = (A^T A + \lambda_L^0 L^T L)^{-1} A^T b$ ;  $k \leftarrow 1$
  - 2:  $\lambda_I^{k-1} = -\frac{\|Ax^{k-1} - b\|_2^2}{\|x^{k-1}\|_2^2 + 1}$  (cf. (4.18));
  - 3:  $x^k = (A^T A + \lambda_L^{k-1} L^T L + \lambda_I^{k-1} I)^{-1} A^T b$  (cf. (4.17));
  - 4: leave  $\lambda_L^k = \lambda_L^{k-1} = \lambda_L^{\min}$  unmodified, or update  $\lambda_L^k = \lambda_L^{k-1} \frac{\|x^k\|_2^2 + 1}{\|x^{k-1}\|_2^2 + 1}$ .
  - 5: (stopping criterion) **if**  $\|(A^T A + \lambda_L^k L^T L + \lambda_I^k I)x^k - A^T b\| < \varepsilon$  ( $\varepsilon$  is a given tolerance),  
**then STOP; else**  $k \leftarrow k + 1$ ; go to 2:
- 

Suppose that  $\lambda_L^{\min}$  is the optimal argument that minimizes a certain criterion  $V(\lambda_L)$  (which can be cross validation, generalized cross validation, an information criterion, etc). A few iterations of updating  $\lambda_L$ ,  $\lambda_I$  and  $x(\lambda_L, \lambda_I)$  can improve the approximation of the RTLS solution, as shown in Algorithm 4.1. Note that an estimate for the original regularization parameter  $\lambda$  is  $\lambda^* = \lambda_L^*/(\|x^*\|_2^2 + 1)$  (where we denote with \* the corresponding converged values). In step 4., one can choose between two methods: either keep  $\lambda_L^k = \lambda_L^{\min}$  unchanged, or update  $\lambda_L$ , by taking into account that in (4.16) the coefficient of  $L^T L$  is of the form  $\lambda(\|x\|_2^2 + 1)$ .

Algorithm 4.1 is in fact a fixed point iteration for computing the RTLS solution, when the original regularization parameter  $\lambda$  is – more or less – *a priori* decided by the value of the Tikhonov parameter  $\lambda_L^{\min}$ .

More freedom in the choice of  $\lambda$  can be obtained if we use an algorithm in the spirit of Wahba [136], who proposes the following strategy:

for nonlinear regularization problems that are solved iteratively, the regularization parameter is chosen at each iteration, using a classical selection method for the (linearized) subproblems.

In the case of RTLS, we propose, to this end, to modify the algorithm in [6] (see §3.3). We remind that it was based on a scalar bisection minimization of the function  $\mathcal{G}(\alpha)$  (see (3.29) on page 56). Each evaluation of  $\mathcal{G}$  for a fixed  $\alpha$  involves solving a quadratically constrained quadratic problem:

$$\min_x \{ \|Ax - b\|^2 + \lambda \alpha \|Lx\|^2 : \|x\|^2 = \alpha - 1 \}.$$

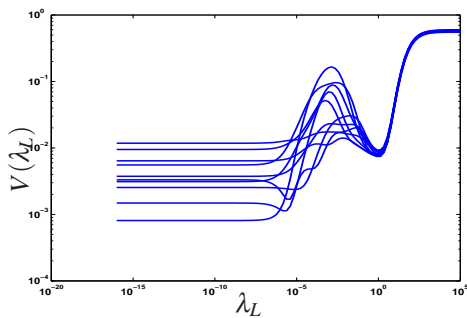
A model selection technique can be used in order to find an appropriate  $\lambda^\alpha$  for this subproblem; then, this should be done at every function evaluation  $\mathcal{G}(\alpha)$ , for each new value of  $\alpha$ . The behavior of such a method, as well as efficient computational methods are still a matter of future investigation.

### 4.5.1 Numerical results for RTLS

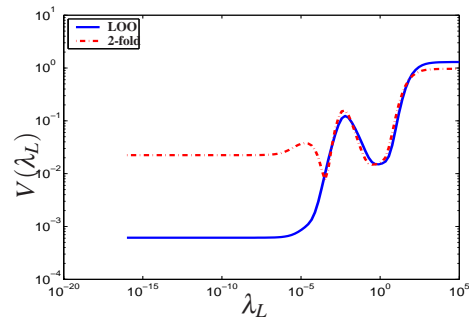
#### Interval of interest for $\lambda_L$ in RTLS

For relatively small values of  $\lambda_L$ , the noise in  $A$  and  $b$  is amplified in a similar manner as when computing the unregularized LS solution. Each  $x(\lambda_L, 0)$  carries its noise amplification into the value of the model selection criterion  $V(\lambda_L)$ . Therefore, for too small  $\lambda_L$ 's,  $V(\lambda_L)$  reflects mainly the contribution of the random noise realization.

Figure 4.2 illustrates the shape of  $V(\lambda_L)$  (when  $V$  is the 2-fold cross validation function) for many noise realizations of the `ilaplace` problem from the Regularization Tools [57], with dimensions  $m = 100$ ,  $n = 20$ . Obviously, one is interested in discarding the



**Figure 4.2.** Cross validation functions (2-fold partitioning) for many noise realizations of the same problem. The right-most minimum occurs around  $\lambda_L = 1$  in all experiments.



**Figure 4.3.** Comparison between LOO and 2-fold partitions. In the region of interest  $\lambda \in (10^{-3}, 10^1)$ , the two functions are very similar.

effects due to the particular noise realization. Figure 4.2 suggests that for  $\lambda_L$  large enough all trajectories of  $V(\lambda_L)$  are similar. In this case, the noise effect is damped, while for small  $\lambda_L$ , the noise influences tremendously the shape of the  $V$  function.

Moreover, the global minimum of  $V(\lambda_L)$  might occur in the “unreliable”, noisy part of the CV function. This is the case in the simulation of Figure 4.2: for the majority of the noise realizations, the global minimum of  $V$  is found for  $\lambda_L \rightarrow 0$ . This is unrealistic, since the unregularized LS solution ( $\lambda_L = 0$ ) is an inappropriate and highly noise-contaminated solution for example `ilaplace`.

These comments suggest defining the optimal regularization parameter away from the part dominated by the noise effects, even if it will not necessarily be at the global minimum of  $V(\lambda_L)$ . Although  $V$  might have several local minima, it can be conjectured that the optimal value of interest is the *right-most local minimum*.

Figure 4.3 is an illustration (on a noisy sample from the same `ilaplace` problem) of the fact that, in the *region of interest*, the leave-one-out and 2-fold partition methods of the cross validation criterion provide basically the same solution, since their right-most local minima are very close to each other.

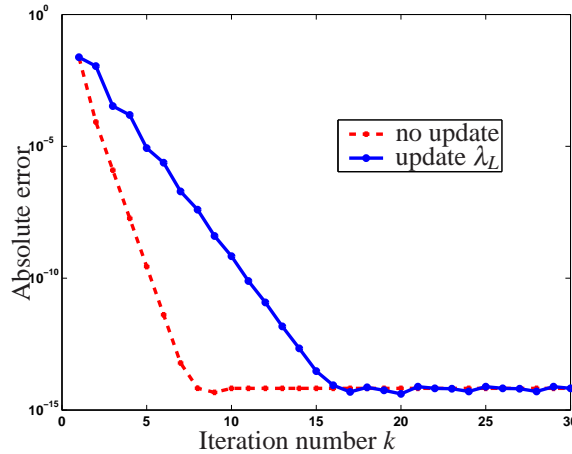
### Iterative refinement of RTLS solution

Algorithm 4.1 is applied to the same `ilaplace` example problem by starting with  $\lambda_L^0 = \lambda_L^{\min} = 0.8111$  and  $\lambda_I^0 = 0$ . The method converges in a few iterations to an approximate RTLS solution:

	$\lambda_L^*$	$\lambda_I^*$	iterations
no update $\lambda_L^k$	0.8111	0.0055	7
update $\lambda_L^k$	0.8846	0.0056	18

Keeping  $\lambda_L$  constant during the iterations seems preferable; this is probably due to the fact that, in this case, the regularization parameter  $\lambda$  of the original Tikhonov RTLS formulation (4.15) is implicitly updated at each iteration of Algorithm 4.1, while the case when we iteratively update  $\lambda_L$  corresponds to keeping  $\lambda$  fixed at a value involving the initial  $\lambda_L$  estimate.

Figure 4.4 shows how the absolute error of solving the RTLS equation (4.16) decreases.



**Figure 4.4.** The absolute error in solving RTLS with Algorithm 4.1, computed as  $\|(A^T A + \lambda_L^k L^T L + \lambda_I^k I)x^k - A^T b\|_2^2$ .

## 4.6 Conclusions

Estimating regularization parameters in the context of linear errors-in-variables models was discussed. The advantages of using a statistically correct model selection criterion were illustrated by studying a cross validation procedure based on the generalization error, instead of the classical prediction error.

Extensions of classical model selection techniques to the computation of optimal truncation levels in truncated total least squares and optimal regularization parameter for regularized total least squares were also proposed.

## **Part II**

# **Regularization for nonlinear problems**



## Chapter 5

# Nonparametric regression using template splines

We study the role of regularization in the context of curve fitting of nonlinear data and we begin with the problem of nonparametric modeling. In this context, regularization is generally synonym to *smoothing*. We define a very general conceptual family of *template splines* that unifies the definitions of various spline families, such as smoothing splines, regression splines or penalized splines. This extension allows an easy incorporation of additional constraints apart from smoothness, such as symmetries, monotonicity, convexity, which is generally not possible in the context of classical spline families.

The nonlinear nonparametric regression problem that defines the template splines can be reduced, for a large class of Hilbert spaces, to a parameterized regularized linear least squares problem, which leads to an important computational advantage.

### 5.1 Introduction

We consider nonlinear nonparametric models of the form  $y_i = f(t_i) + \varepsilon_i$ , where  $y_i$  denotes a real-valued measured observation,  $t_i$  is a regression abscissa from a certain given bounded real interval  $\mathcal{I}$ ,  $\varepsilon_i$  is a zero-mean additive noise term, and the function  $f : \mathcal{I} \rightarrow \mathbb{R}$  is an unknown nonlinearity to be determined.

The problem of estimating the unknown nonlinearity  $f$ , given measured data  $(t_1, y_1), \dots, (t_m, y_m)$ , is a nonparametric regression problem. The range of applications for this type of problems is very wide: from density function estimations to econometric predictions, from biology to sociology. The literature on nonparametric regression is also quite rich (see the monographs [31, 52, 61, 136]). But one tool that is almost always mentioned in connection with nonparametric regression problems is *splines*.

In this chapter, we design a new and more general formulation, which encompasses many of the commonly used types of splines. We partly motivate this work by the need to link the family of penalized splines [28] to the smoothing splines family [136], whose theory is much more rigorous. In a recent article [55], the authors prove some basic statistical properties such as consistency and give expressions for mean squared errors for penalized splines. The extension that we propose is a super-family for both the smoothing splines and for penalized splines, among others. Good statistical properties that hold for smooth-

ing splines, such as the optimality of model selection via generalized cross validation, still hold for the template spline extension. Moreover, with this general formulation we are able to solve not only smoothing problems, but also various constrained smoothing problems.

In section 5.2, we define template splines by generalizing the approach of [136], which embedded the smoothing splines into the theory of reproducing kernel Hilbert spaces. We allow more freedom in choosing some essential elements of the template splines (compared to the smoothing splines definition), and this fact will play a role in widening the splines family.

In section 5.3 we show that by allowing these generalizations, we do not overly restrict the computational properties, since the template spline solution will also be computable using a linear parameterization and solving a regularized linear least squares problem.

Derivations of the well-known types of splines from the more general template spline formulation and some illustrative examples are detailed in section 5.4.

## 5.2 Template splines on reproducing kernel Hilbert spaces

Given the measured data  $(t_1, y_1), \dots, (t_m, y_m)$ , the goal is to fit this data using a nonparametric model  $y_i = f(t_i) + \varepsilon_i$ . Depending on the application, we have to define more specifically some properties to which a good  $f$  should comply. The first step is to choose a function space where our search for a good  $f$  can be restricted to. A general framework for function spaces is described next; it is a framework that has attractive theoretical properties, but which also proves to be advantageous from a computational point of view, as it will be shown in Section 5.3.

### 5.2.1 Reproducing kernel Hilbert spaces

The theory of reproducing kernel Hilbert spaces is a functional analysis tool that gives a sound foundation to the use of splines and other concepts in curve fitting, function estimation, model description or model building applications [137].

A reproducing kernel Hilbert space (RKHS)  $\mathcal{H}$  is a linear function space, with an embedded inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and an induced norm  $\| \cdot \|_{\mathcal{H}}$ . It is a Hilbert space on some domain  $\mathcal{I}$ , thus it is complete with respect to the induced distance topology, but the most important property of a RKHS is that all pointwise evaluations can be written as bounded linear functionals. That is, for every  $t \in \mathcal{I}$  the linear functional  $L_t$  defined by the relation  $L_t f = f(t)$  is bounded ( $\|L_t f\|_{\mathcal{H}} \leq \text{const}_t \|f\|_{\mathcal{H}}$ ). As a consequence of Riesz representation theorem [135] (which says that every bounded linear functional has a *representer* in the Hilbert space), for each  $t \in \mathcal{I}$  there exists an element  $\eta_t \in \mathcal{H}$  such that

$$f(t) = \langle \eta_t, f \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}.$$

In other words, each  $t \in \mathcal{I}$  has a corresponding  $\eta_t \in \mathcal{H}$  that “describes” all possible values of the functions in  $\mathcal{H}$  at  $t$  through the inner product of  $\mathcal{H}$ .

The function  $K : \mathcal{I} \otimes \mathcal{I} \rightarrow \mathbb{R}$  defined as  $K(s, t) := \langle \eta_s, \eta_t \rangle_{\mathcal{H}}$  for all  $s, t \in \mathcal{I}$  is the *reproducing kernel* for  $\mathcal{H}$ . It is symmetric, positive definite and satisfies  $\langle K(s, \cdot), K(t, \cdot) \rangle_{\mathcal{H}} =$

$K(s, t)$ .

There is a one-to-one relationship between the set of reproducing kernel Hilbert function spaces defined on an interval  $\mathcal{I}$  and the set of positive definite functions defined on  $\mathcal{I} \times \mathcal{I}$ . Moreover, under general circumstances (*e.g.*, continuity and square-integrability), any positive definite function  $K$  admits an eigenfunction-eigenvalue decomposition:

$$K(s, t) = \sum_{v=1}^{\infty} \sigma_v \Phi_v(s) \Phi_v(t),$$

where  $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$ , with  $\sum_{v=1}^{\infty} \sigma_v^2 < \infty$ , are the eigenvalues of  $K$ , and  $\Phi_1, \Phi_2, \dots$ , an orthonormal sequence of square-integrable functions on  $\mathcal{I}$ , are called the eigenfunctions of  $K$ .  $\Phi_1, \Phi_2, \dots$ , form a Hilbert basis for the RKHS that corresponds to the kernel  $K$ .

These properties imply good news from the practical point of view. Choosing the RKHS  $\mathcal{H}$  can either be replaced by choosing a desired kernel  $K$  or by choosing a sequence of orthonormal functions that act as a basis or a generator family for  $\mathcal{H}$ . For this reason, in practical applications, the theoretical formulation of  $\mathcal{H}$  is often omitted; typically, it is replaced by the definition of a certain sequence of generators for the “splines”, such as truncated polynomials, piecewise polynomials or B-spline bases [21].

## 5.2.2 Unconstrained smoothing

When a space  $\mathcal{H}$  is specified, we formulate the following least squares problem:

**Problem 1 ( $\mathcal{H}$ -smoothing)** Solving the least squares problem

$$\min_{f \in \mathcal{H}} \sum_{i=1}^m (y_i - f(t_i))^2 \quad (5.1)$$

for the function  $f$  is called the  $\mathcal{H}$ -smoothing problem.

Through  $\mathcal{H}$ -smoothing, the measured data  $\mathbf{y} \in \mathbb{R}^m$  is *orthogonally projected* onto the space  $\mathcal{H}(\mathbf{t})$ , which is a vector space in  $\mathbb{R}^m$  of the discretizations of all the functions in  $\mathcal{H}$  at the abscissas  $t_1, \dots, t_m$ . In other words,  $\mathcal{H}$ -smoothing finds the function  $\hat{f}$  in  $\mathcal{H}$  that best approximates the data in  $\mathbf{y}$  at given abscissas  $\mathbf{t}$ . For the values of  $t \in \mathcal{I}$  that are not among the elements of  $\mathbf{t}$ ,  $\hat{f}$  can be used as predictor.

## 5.2.3 Constrained smoothing and template splines

To impose additional constraints such as smoothness, monotonicity, convexity, equality or inequality relations, on the estimated function  $f$ , it is possible in some cases to revise the definition of the space  $\mathcal{H}$ , such that any function in  $\mathcal{H}$  satisfies the extra constraints. However, the design of  $\mathcal{H}$  in this case can become tedious. An alternative method is based on the projection framework for constrained smoothing developed by [84]; basically, the additional constraints are imposed in a sequential manner: “smooth then constrain,” which might be suboptimal.

Here, we propose a one-step solution that is feasible and efficient for a wide class of constraints that can be written as (semi)norm conditions.

Let  $\mathcal{P}$  denote a linear operator from  $\mathcal{H}$  to  $\mathcal{H}'$ , where  $\mathcal{H}'$  is a certain normed space.

**Problem 2 ( $\mathcal{P}$ -constrained  $\mathcal{H}$ -smoothing)** Solving the constrained least squares problem

$$\min_{f \in \mathcal{H}} \sum_{i=1}^m (y_i - f(t_i))^2, \quad \text{such that } \|\mathcal{P}f\|_{\mathcal{H}'} \leq \delta \quad \text{for a fixed } \delta \geq 0, \quad (5.2)$$

for the function  $f$  is called the  $\mathcal{P}$ -constrained  $\mathcal{H}$ -smoothing problem.

The value  $\|\mathcal{P}f\|_{\mathcal{H}'}$  is the measure of the quality of an arbitrary  $f$  with respect to the desired constraints, or to the prior knowledge on properties that good estimators  $f$  should satisfy. Sometimes, if no such prior knowledge is available,  $\|\mathcal{P}f\|_{\mathcal{H}'}$  could also be a measure of parsimony for the model  $f$ .

Using a Lagrange multiplier argument, we arrive at the following:

**Problem 3 (Template spline)** The following minimization is the *template spline* minimization problem:

$$\min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m (y_i - f(t_i))^2 + \lambda \|\mathcal{P}f\|_{\mathcal{H}'}^2. \quad (5.3)$$

This formulation is equivalent to the  $\mathcal{P}$ -constrained  $\mathcal{H}$ -smoothing problem, for a certain relation between  $\lambda \geq 0$  and  $\delta \geq 0$ . The penalty formulation of the template spline problem (5.3) shows that a trade-off between closeness of the model  $f$  to the observations and “quality” of the model – via the  $\mathcal{P}$ -constraint – can be ensured by choosing an appropriate  $\lambda \geq 0$ . In Section 5.3.2 we focus on the generalized cross validation (GCV) criterion that can be used for choosing the so-called *smoothing* (or *penalty* or *regularization*) parameter  $\lambda$ , given the measured data.

**Remark 11 (More general formulations)** Sometimes in spline literature, the function evaluation operator that appears in the sum of squares of problems (5.1, 5.2, 5.3),  $t_i \rightarrow f(t_i)$ , is replaced by a more general bounded linear operator  $L_i$ , i.e.,  $t_i \rightarrow L_i f$ . All results in this chapter are straightforwardly generalizable to this setting.

Moreover, if the zero-mean noise term  $\varepsilon_i$  is assumed to have a known covariance matrix other than the identity matrix, the least squares term in all minimization problems (5.1, 5.2, 5.3) can be easily transformed into *weighted least squares* formulations, using the inverse of the noise covariance matrix as weight matrix.

## 5.3 Computing template splines

### 5.3.1 Transformation to a linear least squares problem

The following theorem says that when  $\mathcal{H}$  is a RKHS and  $\mathcal{P}$  has some general properties (see below), the solution of the template problem (5.3) has a representation in terms of the reproducing kernel and/or its eigenfunctions, and it can be easily computed by solving linear least squares or linear equations. In particular, problem (5.1) can also be efficiently solved as a linear least squares problem.

Let  $\mathcal{H}_{\mathcal{P}}$  denote the null space of  $\mathcal{P}$  in  $\mathcal{H}$  (that is,  $\mathcal{H}_{\mathcal{P}} := \ker \mathcal{P} := \{f \in \mathcal{H} : \mathcal{P}f = 0\}$ ), and assume it is finite dimensional, of dimension  $d$ . For instance, if  $\mathcal{P}$  is the  $d^{\text{th}}$  order

derivative operator on some space  $\mathcal{H}$  of (infinitely) continuously differentiable functions, then  $\ker \mathcal{P}$  is spanned by the finite basis  $\{1, t, t^2, \dots, t^{d-1}\}$ .

Assume that  $\mathcal{P}$  preserves orthogonality relations from  $\mathcal{H}$  to  $\mathcal{H}'$ : thus, if  $\langle f, g \rangle_{\mathcal{H}} = 0$ , then  $\langle \mathcal{P}f, \mathcal{P}g \rangle_{\mathcal{H}'} = 0$ .

**Theorem 5.1 (generalized from [136], Th. 1.3.1).** *The solution of the template spline problem (5.3), for a fixed  $\lambda \geq 0$ , has a closed-form expression given by*

$$\hat{f}_\lambda := \sum_{k=1}^d a_k \phi_k + \sum_{i=1}^m b_i \mu_i, \quad (5.4)$$

where

- $\phi_1, \dots, \phi_d$  are a basis for  $\mathcal{H}_{\mathcal{P}}$ , ( $d := \dim \mathcal{H}_{\mathcal{P}}$ ),
- $\mu_i = K(t_i, \cdot)$ , for  $i = 1, \dots, m$ ,
- the coefficients  $\mathbf{a} := (a_1, \dots, a_d)^\top$  and  $\mathbf{b} := (b_1, \dots, b_m)^\top$  are given by

$$\begin{aligned} \mathbf{a} &= \left( A^\top A - A^\top B(B^2 + m\lambda C)^{-1} B A \right)^{-1} A^\top (I_m - B(B^2 + m\lambda C)^{-1} B) \mathbf{y}, \\ \mathbf{b} &= (B^2 + m\lambda C)^{-1} B(\mathbf{y} - A\mathbf{a}), \end{aligned} \quad (5.5)$$

where the matrices  $A$  ( $m \times d$ ),  $B$  ( $m \times m$ ) and  $C$  ( $m \times m$ ) are computed as

$$\begin{aligned} A &= \{(\phi_k(t_l))\}_{(l=1, \dots, m; k=1, \dots, d)}, \\ B &= \{K(t_i, t_k)\}_{(i, j=1, \dots, m)} = \{\langle \mu_i, \mu_j \rangle_{\mathcal{H}}\}_{(i, j=1, \dots, m)}, \\ C &= \{\langle \mathcal{P}\mu_i, \mathcal{P}\mu_j \rangle_{\mathcal{H}'}\}_{(i, j=1, \dots, m)}. \end{aligned} \quad (5.6)$$

**Proof.** Since the set  $\{\phi_1, \dots, \phi_d\}$  is a basis for  $\mathcal{H}_{\mathcal{P}}$ , any  $f \in \mathcal{H}$  can be written as  $f = \sum_{k=1}^d a_k \phi_k + f^\perp$ , for some  $a_1, \dots, a_d \in \mathbb{R}$  and an  $f^\perp \in \mathcal{H}_{\mathcal{P}}^\perp$ . It will be shown that  $f^\perp$  can be further decomposed such that the optimal solution of the minimization (5.3) has a finite decomposition in  $\mathcal{H}$ . To this end, write  $f^\perp = \sum_{i=1}^m b_i \mu_i + \rho$ , where  $\mu_i := K(t_i, \cdot)$  and  $\rho \in \mathcal{H}$  is orthogonal to  $\phi_1, \dots, \phi_d$  and to  $\mu_1, \dots, \mu_m$ .

The least squares objective to be minimized becomes:

$$\begin{aligned} \mathcal{F}(f) &:= \frac{1}{m} \sum_{i=1}^m (y_i - f(t_i))^2 + \lambda \|\mathcal{P}f\|_{\mathcal{H}'}^2 \\ &= \frac{1}{m} \sum_{i=1}^m \left( y_i - \sum_{k=1}^d a_k \phi_k(t_i) - \sum_{j=1}^m b_j \mu_j(t_i) - \rho(t_i) \right)^2 \\ &\quad + \lambda \left\| \mathcal{P} \left( \sum_{k=1}^d a_k \phi_k + \sum_{j=1}^m b_j \mu_j + \rho \right) \right\|_{\mathcal{H}'}^2 \\ &= \frac{1}{m} \|\mathbf{y} - A\mathbf{a} - B\mathbf{b}\|_2^2 + \lambda \mathbf{b}^\top C \mathbf{b} + \lambda \|\mathcal{P}\rho\|_{\mathcal{H}'}^2, \end{aligned}$$

where  $A$ ,  $B$  and  $C$  are given in (5.6).

We used the facts that  $\rho(t_i) = \langle \rho, K(t_i, \cdot) \rangle_{\mathcal{H}} = \langle \rho, \mu_i \rangle_{\mathcal{H}} = 0$  from the choice of  $\rho$  orthogonal to  $\mu_1, \dots, \mu_m$ ; that  $\phi_1, \dots, \phi_d$  are in the null space of  $\mathcal{P}$ ; and that  $\rho$  being orthogonal to  $\sum_{j=1}^m b_j \mu_j$  in  $\mathcal{H}$  implies  $\langle \sum_{j=1}^m \mathcal{P} \mu_j, \mathcal{P} \rho \rangle_{\mathcal{H}'} = 0$ , by the ‘‘preservation of orthogonality’’ property of  $\mathcal{P}$ .

For  $f$  to be a minimizer of  $\mathcal{F}$ , we must have  $\|\mathcal{P} \rho\|_{\mathcal{H}'} = 0$ . This implies  $\rho \in \ker \mathcal{P}$ , which contradicts the choice of  $\rho$  orthogonal to the basis of  $\ker \mathcal{P}$ ; thus, the only possibility is  $\rho = 0$ .

The normal system of equations for the linear least squares problem that we obtained is:

$$\begin{cases} A^\top A \mathbf{a} + A^\top B \mathbf{b} = A^\top \mathbf{y}, \\ B^\top A \mathbf{a} + (B^\top B + m\lambda C) \mathbf{b} = B^\top \mathbf{y}. \end{cases} \quad (5.7)$$

Since  $B$  is symmetric positive definite, the closed-form expressions (5.5) are easily obtained.  $\square$

The theorem gives us a simple way to express the minimization (5.3) as a problem that is linearly parameterized by the vectors  $\mathbf{a}$  and  $\mathbf{b}$ , more precisely, as a *regularized linear least squares problem*:

$$\min_{\mathbf{a} \in \mathbb{R}^d, \mathbf{b} \in \mathbb{R}^m} \frac{1}{m} \|\mathbf{y} - A\mathbf{a} - B\mathbf{b}\|_2^2 + \lambda \mathbf{b}^\top C \mathbf{b}. \quad (5.8)$$

The matrices  $A$ ,  $B$  and  $C$  defined in (5.6) are characteristic for the RKHS  $\mathcal{H}$ , the penalty operator  $\mathcal{P}$ , and the set of given abscissas  $\mathbf{t}$ . The columns of  $A$  are discretized versions of the functions that span the null space of  $\mathcal{P}$ , and the columns of  $B$  are those that contribute in the directions outside the null space of  $\mathcal{P}$ . For this reason, only the coefficients  $\mathbf{b}$  are subject to the constraint implied by the penalty term  $\lambda \mathbf{b}^\top C \mathbf{b}$ .

In practical cases where a spline basis is given instead of the RKHS  $\mathcal{H}$ , the extended matrix  $\begin{bmatrix} A & B \end{bmatrix}$  could be replaced by a matrix whose columns are discretized splines from the given basis set.

### 5.3.2 Data driven spline fitting using generalized cross validation

The smoothing parameter  $\lambda$  will usually not be known *a priori*; the methods for choosing a good  $\lambda$  often rely on optimizing some criterion and involves many evaluations of spline solutions (from their coefficients  $\mathbf{a}$ ,  $\mathbf{b}$ ), for various values of  $\lambda$ . For this reason, it is required to have efficient algorithms for the computation of coefficients  $\mathbf{a}$  and  $\mathbf{b}$ , instead of the expressions in (5.5).

The regularized least squares problem of minimizing

$$\frac{1}{m} \|\mathbf{y} - A\mathbf{a} - B\mathbf{b}\|_2^2 + \lambda \mathbf{b}^\top C \mathbf{b} = \frac{1}{m} \left\| \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} A & B \\ 0 & \sqrt{m\lambda} D \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \right\|_2^2,$$

where  $D$  is a Cholesky factor of the symmetric positive definite matrix  $C$  ( $D^\top D = C$ ), can be efficiently solved if we compute in advance the generalized singular value decomposition (GSVD) [50] of the matrix pair  $(\begin{bmatrix} A & B \\ 0 & D \end{bmatrix}, \begin{bmatrix} 0 & D \end{bmatrix})$ .

Many methods for choosing an “optimal” regularization parameter  $\lambda$  can be employed in the context of template splines. A recent simulation study [75] compares several parameter selection procedures in the case of *smoothing splines* (leave-one-out cross validation, generalized cross validation, Mallows’  $C_p$ , Akaike’s information criterion, and risk estimation methods); the conclusion is that none of the methods can be overall the best.

Here we present the application of generalized cross validation [19] to template splines. The GCV criterion aims at minimizing the function

$$G(\lambda) := \frac{\|\mathbf{y} - \mathbf{A}\mathbf{a}_\lambda - \mathbf{B}\mathbf{b}_\lambda\|_2^2}{[\text{Tr}(I_m - H(\lambda))]^2},$$

where  $(\mathbf{a}_\lambda, \mathbf{b}_\lambda)$  denote the optimal solution vectors that are obtained from solving problem (5.8) holding the regularization parameter at the fixed value  $\lambda$ , and  $H(\lambda)$  is the *influence matrix* (or *hat matrix*, or *smoother matrix*) defined as the  $m \times m$  matrix that makes the equality  $H(\lambda)\mathbf{y} = \mathbf{A}\mathbf{a}_\lambda + \mathbf{B}\mathbf{b}_\lambda$  hold. Its formula reads:

$$H(\lambda) = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{A}^\top \mathbf{A} & \mathbf{A}^\top \mathbf{B} \\ \mathbf{B}^\top \mathbf{A} & \mathbf{B}^\top \mathbf{B} + m\lambda \mathbf{C} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A}^\top \\ \mathbf{B}^\top \end{bmatrix}.$$

GCV was originally introduced for problems such as ridge regression [43] and smoothing splines [19], which belong to the same category of regularized linear least squares problems as the template splines do. For these problems, it is possible to show that the GCV criterion is a *predictive mean squared error* criterion. In the context of template splines this will mean that, under reasonable assumptions and in the asymptotic situation, the values chosen for  $\lambda$  by minimizing the GCV function, when the sample size grows to infinity, converge to the optimal  $\lambda$  that corresponds to the spline model that is closest to the *true* underlying model.

For efficient computations, the GSVD of the matrix pair  $(\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{D} \end{bmatrix})$  can be used, because in this case, the matrix that must be inverted in the formula of  $H(\lambda)$  becomes a diagonal matrix, and, thus, computing this inverse is very fast, for any  $\lambda$ . (See [58].) Interesting techniques for making the optimization of the GCV function fast for large-scale problems are described by [51], making use of a randomization method to compute the trace of a large matrix.

## 5.4 Examples

In this section, we exemplify the described theory, by first showing that some classical spline families belong to our generalization. Afterwards, we design some examples that are atypical for the classical splines, but which can however be successfully tackled by the template splines.

### 5.4.1 Smoothing, regression and penalized splines as template splines

Smoothing splines and penalized splines start from the same idea of using a weighted penalty term in order to obtain a desired degree of smoothness of the reconstructed non-

parametric model. Regression splines rely on a different technique to obtain a smoother or a rougher approximation: tuning the number of basis splines.

Despite their similarities, the penalized splines cannot be seen as a subset of smoothing splines (or *vice-versa*), mainly because of the fact that their penalty terms are of totally different nature.

But with template splines, we generalize these approaches by allowing the operator  $\mathcal{P}$  to be more general than just an orthogonal projection (as for smoothing splines) and to take value in other normed spaces (such as  $\mathbb{R}^n$ , for penalized splines). In the meantime, the good properties of smoothing splines, such as the asymptotic optimality of using GCV for choosing the penalty parameter, are still kept.

### Smoothing splines as template splines

Smoothing splines [136] are a subset of the template splines family, obtained when the space  $\mathcal{H}^l$  is the same as the space  $\mathcal{H}$ , and the operator  $\mathcal{P}$  is an orthogonal projection.

A classical example of a smoothing spline is the *natural spline*. The space  $\mathcal{H}$  is defined as the Sobolev space,

$$\mathcal{H} := \{f : [0, 1] \rightarrow \mathbb{R} : f, f', \dots, f^{(d-1)} \text{ are absolutely continuous and } f^{(d)} \in \mathcal{L}_2\},$$

where  $f^{(k)}$  denotes the  $k^{\text{th}}$  derivative and  $\mathcal{L}_2$  denotes the space of square integrable functions.  $\mathcal{H}$  is endowed with the square seminorm:

$$\|f\|_{\mathcal{H}}^2 := \sum_{k=1}^d \left[ f^{(k)}(0) \right]^2 + \int_0^1 \left[ f^{(d)}(u) \right]^2 du.$$

The corresponding reproducing kernel is

$$K(s, t) := \sum_{k=1}^d \phi_k(s)\phi_k(t) + \int_0^1 G_d(s, u)G_d(t, u)dt,$$

where each  $\phi_k$  (for  $k = 1, \dots, d$ ) denotes the monomial  $\phi_k(t) = \frac{t^{k-1}}{(k-1)!}$  and  $G_d$  is the Green function  $G_d(t, u) = \frac{(t-u)_+^{d-1}}{(d-1)!}$ , with  $(x)_+ = x$  for  $x \geq 0$  and 0 otherwise.

The operator  $\mathcal{P}$  is simply defined as the orthogonal projection onto the subspace

$$\mathcal{H}_{\mathcal{P}} := \{f \in \mathcal{H} : f(0) = f(1) = f'(0) = f'(1) = \dots = f^{(d-1)}(0) = f^{(d-1)}(1) = 0\}.$$

The optimal natural spline solution solves the problem:

$$\min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m (y_i - f(t_i))^2 + \lambda \int_0^1 \left[ f^{(d)}(u) \right]^2 du.$$

Thus, the natural spline reconstruction from noisy measurements aims at  $d$ -order smoothness, with no boundary conditions.

The smoothing spline solution can still be obtained by solving a regularized linear least squares problem (details in [136, Chapter 1]). But  $\mathcal{H}^l = \mathcal{H}$  implies that for smoothing splines we can redefine the matrices that appear in our Theorem 5.1 and have  $B = C$ , which gives a simpler numerical solution for the smoothing spline, since a QR decomposition can be used instead of a GSVD.

### Regression splines as template splines

A regression spline [36] is a piecewise polynomial function that is smooth at the jointure points (knots) up to the degree of the polynomials minus 1. Typical polynomial functions that are used with regression splines are truncated polynomials or B-splines [21].

The space  $\mathcal{H}$  can be defined as the span of the considered polynomial set. An example is the span of the truncated polynomials of degree  $q$ ,  $\text{span}\{t, t^2, \dots, t^q, (t - \tau_1)_+, (t - \tau_2)_+, \dots, (t - \tau_N)_+\}$ , where  $\tau_1, \dots, \tau_N$  are  $N$  given knots, and  $(x)_+$  denotes, as before,  $x$  or 0, depending whether  $x$  is nonnegative or negative. Another example is the span of the B-spline basis. The widely used B-splines are defined recursively such that, at the given knots  $\tau_1, \dots, \tau_N$ , each of the functions in the B-spline basis set is  $q$ -times continuously differentiable. The recursion formula reads:

$$v_i^{\{1\}}(t) = \begin{cases} 1, & \text{if } \tau_i \leq t \leq \tau_{i+1} \\ 0, & \text{otherwise,} \end{cases}; \quad v_i^{\{k\}}(t) = \frac{v_i^{\{k-1\}}(t) \cdot (t - \tau_i)}{\tau_{i+k-1} - \tau_i} + \frac{v_{i+1}^{\{k-1\}}(t) \cdot (\tau_{i+k} - t)}{\tau_{i+k} - \tau_{i+1}}, \quad (5.9)$$

and the  $q$ -order basis that defines  $\mathcal{H}$  as  $\text{span}\{v_1^{\{q\}}, \dots, v_{N-q}^{\{q\}}\}$ , contains thus  $N - q$  functions.

For fitting with regression splines, we can set the operator  $\mathcal{P}$  as the identically zero mapping (and  $\mathcal{H}' = \{0\}$ ). Thus the penalty part disappears from the problem (5.3), and we are left with a simple least squares problem.

Sometimes, the number of knots  $N$  and the knot positions must be optimized before performing regression. In the situation when we consider equally-spaced knots (*i.e.*, uniform regression splines), only the integer parameter  $N$  has to be chosen. Therefore, we consider as hyperparameter  $\lambda$  the number of knots  $N$  and the generalized cross validation (or other typical method for model order selection) can then be used in order to choose an appropriate value for  $N$ . The parameter  $N$  controls in this case the bias-variance trade-off. A large  $N$  means that the measured  $\mathbf{y}$  can be modeled with more fidelity, and a smaller  $N$  means that the reconstructed spline is smooth.

### Penalized splines as template splines

The simple smoothing method of *penalized splines* was introduced by [28] (and in very similar terms, by [108]). It is based on (uniform) regression splines, but with an added penalty term, weighted by a certain regularization parameter  $\lambda$ . An interesting detail is found in [107], namely that the number of knots can be set to an arbitrary value (a number large enough, but always less than the number of regression points); the technique of penalized splines controls the degree of smoothness only through the parameter  $\lambda$ .

In formal notation, let  $v_1, \dots, v_n$  denote a basis family of truncated polynomials or of B-splines of degree  $q$ . Then, given the measurements  $y_1, \dots, y_m$ , a function  $f$  can be modeled as a linear combination of  $v_1, \dots, v_n$ , *i.e.*,  $f = \sum_{k=1}^n c_k v_k$ , by minimizing the penalized least squares function

$$\min_{\mathbf{c} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^m \left( y_i - \sum_{k=1}^n c_k v_k(t_i) \right)^2 + \lambda \|\Delta \mathbf{c}\|^2, \quad (5.10)$$

where  $\Delta$  is a finite difference operator (of a certain chosen order) and  $\lambda$  is a positive scalar

parameter that controls the degree of smoothness.

From the formulation (5.10) it is clear that the penalized splines are also a particular example of template splines. We define the space  $\mathcal{H}$  as the span of the function family  $\{v_1, \dots, v_n\}$ , and the space  $\mathcal{H}'$  as the  $n$ -dimensional Euclidean space  $\mathbb{R}^n$ . The operator  $\mathcal{P}$  can be easily set using the canonical decomposition of any element  $f$  in  $\mathcal{H}$ , as  $\mathcal{P}(f) = \mathcal{P}(\sum_{k=1}^n c_k v_k) = \Delta \mathbf{c}$ , where  $\Delta$  is the same approximation matrix for a finite difference operator that is used in the penalized splines problem (5.10).

## 5.4.2 Other applications of template splines

### Smoothing in transformed spaces

We consider the abscissa interval  $\mathcal{I}$  as the “time axis” (and therefore the measurements  $y_1, \dots, y_m$  are given in the “time-domain”), but we are interested in imposing smoothness to the *Fourier transform* of the solution  $f$ .

Let  $\mathcal{F}$  be the Fourier transform and  $\mathcal{F}^{-1}$  denote the inverse Fourier transform. Consider a RKHS  $\mathcal{H}$  of functions defined on the time-domain  $\mathcal{I}$ . Then the image space is  $\mathcal{H}' := \mathcal{F}\mathcal{H} = \{\mathcal{F}(f) : f \in \mathcal{H}\}$ . We can measure smoothness in  $\mathcal{H}'$  with the aid of a projection  $\mathcal{Q}$  (see, for instance, the natural spline type of smoothing in section 5.4.1), *i.e.*, the norm  $\|\mathcal{Q}g\|_{\mathcal{H}'}$  (for any  $g$  in  $\mathcal{H}'$ ) is the measure that we use for smoothness. It is then possible to define the mapping  $\mathcal{P} : \mathcal{H} \rightarrow \mathcal{H}'$  by the composition

$$\mathcal{P}(f) = \mathcal{Q}(\mathcal{F}(f)).$$

Solving the template spline problem (5.3) in this case is equivalent to solving

$$\min_{g \in \mathcal{H}'} \frac{1}{m} \sum_{i=1}^m (y_i - (\mathcal{F}^{-1} \circ g)(t_i))^2 + \lambda \|\mathcal{Q}g\|_{\mathcal{H}'}^2,$$

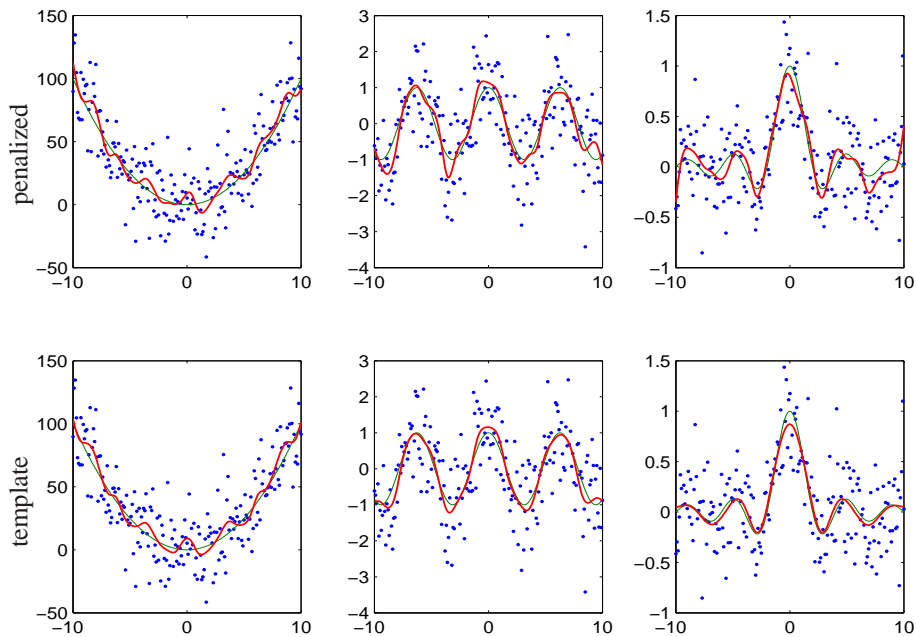
which is a template spline problem in the space  $\mathcal{H}'$ .

### Constrained smoothing

Classical smoothing splines or regression splines are designed to smooth scattered data in order to obtain meaningful approximations. However, additional constraints are usually hard to impose. (See for instance [71], where linear programming methods are used for several constrained smoothing problems.) With templates splines, constraints can be imposed through the penalty term. Tuning the penalty coefficient allows to slightly break the constraints, which is advantageous for identifying situations when the data does not obey the constraints.

In this paragraph, we give two numerical examples from an application where template splines are designed for constrained smoothing problems. In the first example, we show that imposing a symmetry constraint helps finding a reliable model for noisy data that comes from a symmetric function. The second example illustrates that when using a wrong constraint, the GCV value for  $\lambda$  gives a model that clearly breaks the constraint.

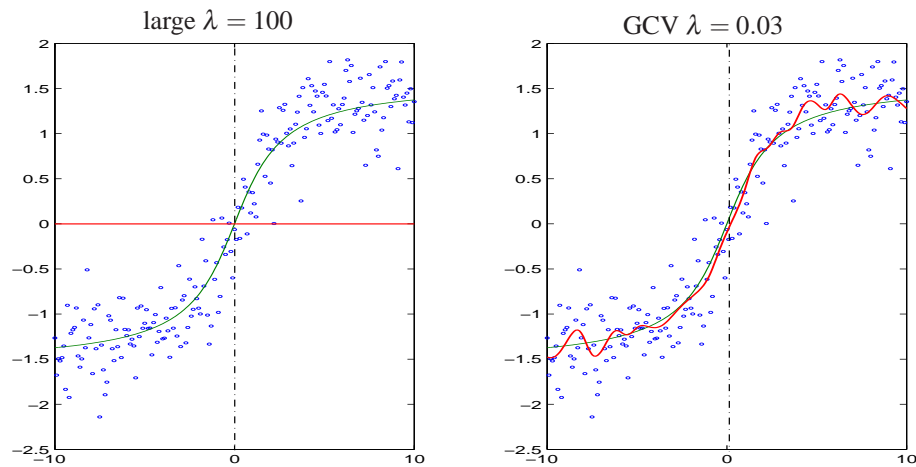
**Example 5.2** We consider several scalar functions on an interval  $[-\alpha, \alpha]$  that are symmetric with respect to the vertical axis through zero. We generate discretizations and add



**Figure 5.1.** In all plots, the thin solid line is the simulated symmetric function, respectively:  $x^2$ ,  $\cos(x)$  and  $\text{sinc}(x)$ ; the dots are the noise corrupted data. The top row of plots shows as thick red lines the penalized spline reconstructions, while in the bottom row these thick red lines represent the template splines with symmetry constraint. The latter exhibit better reconstructions of the symmetric models.

Gaussian noise. We then compute from the noisy data two models, both based on a B-spline basis: a penalized spline where a second order derivative operator is used to obtain smoothness, and a template spline where the penalty contains the second order derivative for smoothness, but also a term for symmetry, which is governed by a matrix  $[I \ -J]$ , where  $I$  is the identity matrix and  $J$  is its mirror reflection. We choose the penalty coefficient by GCV in both models, but we observe that using the symmetry constraint gives models that have smaller mean squared error with respect to the original simulated functions, than when no constraint is imposed. See Figure 5.1 for illustration.

**Example 5.3** We consider now as original function a function that is not symmetric with respect to the vertical axis through zero, *e.g.*, the arctangent function. We compute however a template spline with symmetry penalty. As illustrated in Figure 5.2, the fit is symmetric, but does not reflect the data, when we choose a penalty parameter  $\lambda$  which is quite big. On the other hand, the GCV criterion chooses a  $\lambda$  that corresponds to a spline reconstruction that is reasonable for the given data, and ignores the inappropriate symmetry constraint.



**Figure 5.2.** In both plots, the thin solid line is the simulated non-symmetric function  $\arctan(x)$ ; the dots are the noise corrupted data. On the left, the spline corresponds to a very large value of the penalty parameter, thus the symmetry is strongly imposed, yielding an almost zero solution (thick red horizontal line). On the right, the penalty parameter is computed with GCV, and the symmetry constraint is completely ignored, but the data is reasonably fitted by the template spline (thick red line).

## 5.5 Conclusions

The template spline introduced in this chapter generalizes some of the most used families of splines from literature. An issue that we find important is that, due to this common framework, the family of penalized splines can be put in its own right beside the classical family of smoothing splines.

Template splines solutions are quite easily computable, since a regularized linear least squares optimization can be solved instead.

Template splines can constitute a solid basis for nonparametric regression problems where the “data acquisition space” is different from the space where interesting properties appear, or where strong or weak constraints should be imposed.

## Chapter 6

# Regularized semiparametric modeling

In this chapter, we formulate and solve a semiparametric fitting problem with regularization constraints. The model that we focus on is composed of a parametric nonlinear part and a nonparametric part that can be reconstructed using template splines. Regularization is employed in order to impose additional properties, such as a certain degree of smoothness, on the nonparametric part.

Semiparametric regression is presented in this chapter as a generalization of nonlinear regression, and all important differences that arise from the statistical and computational points of view are highlighted.

### 6.1 Introduction

This chapter is a survey dedicated to a semiparametric fitting problem with regularization constraints. We consider nonlinear models that are partially known, partially unknown. From given noisy measurements, we are interested in estimating regression parameters of the known nonlinear part, as well as estimating the nuisance as a nonparametric part.

Our motivation for studying this problem comes from an application in nuclear magnetic resonance (NMR) spectroscopy, which will be thoroughly described in Chapter 7.

In the context of semiparametric modeling, much work has been devoted to partially linear stochastic models of the form

$$y_i = F(t_i)^\top \theta + g(t_i) + \varepsilon_i, \quad (i = 1, 2, \dots, m),$$

where  $y_1, \dots, y_m \in \mathbb{R}$ , as well as  $F(t_1), \dots, F(t_m) \in \mathbb{R}^p$  are measured quantities,  $\varepsilon_i$  denotes the measurement noise;  $\theta \in \mathbb{R}^p$  is the linear regression vector to be estimated, and  $g : \mathbb{R} \rightarrow \mathbb{R}$  is a nonlinear nonparametric part. Usually, the goal is to find a consistent estimate of  $\theta$ , while considering the nonparametric part  $g(\cdot)$  as nuisance (see, *e.g.*, a very recent comprehensive study of semiparametric models presented in the book [109]; see also [9]). The nonlinearity  $g(\cdot)$  is often modeled by smoothing splines [136]. Smoothing splines are sometimes employed together with a nonlinear model as a way of testing if the nonlinear model is adequate [136, Chapter 9]: an almost zero reconstructed spline means that the nonlinear model is the right one. Another example of semiparametric modeling is presented

in [16], which employs a Bayesian framework in order to discriminate nonparametrically between a deterministic regression function and a noise term with smooth spectral density. In [69], a very general spline framework is developed such that nonlinear relationships are allowed between several nonparametric functions.

The semiparametric regression problem treated in this chapter combines additively parametric nonlinear regression with nonparametric spline smoothing. It is shown that it is adequate to solve such a semiparametric fitting problem using nonlinear least squares plus a penalty on the spline smoothness. As in nonlinear regression, we can infer asymptotic properties of the semiparametric regression estimates. Under Gaussian noise assumption, normality of the estimates is recovered; however, because of the regularization term, the computed parameters will be biased from the “true” values. We develop detailed bias and covariance formulas, which allow derivation of other statistically relevant information, such as confidence intervals.

We give an algorithmic outline of regularized semiparametric regression, with emphasis on efficient computation. One of the main issues in this context is the choice of the *regularization parameter* that controls the trade-off between nonlinear misfit minimization and effective regularization. We propose an automated iterative selection method that is based on the classical generalized cross validation criterion. The method is data-driven and does not need prior estimates for the noise statistics.

The outline of this chapter is as follows. The semiparametric problem formulation that we propose, its theoretical solution, as well as computational issues are discussed in Section 6.2. In Section 6.3, its statistical properties are developed, and in Section 6.4 we discuss to which types of problems we can safely apply this framework without having to deal with identifiability and non-uniqueness issues. Finally, in Section 6.5 we show through simulation examples the performance as well as the limitations of the method.

## 6.2 Semiparametric model with smoothness constraint

### 6.2.1 Model formulation

Let  $\mathcal{H}$  be a Hilbert space of functions defined on an interval  $\mathcal{I} \subset \mathbb{R}$ , endowed with an inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and an induced norm  $\|\cdot\|_{\mathcal{H}}$ . Let  $\mathcal{P}$  denote an operator from  $\mathcal{H}$  to a space  $\mathcal{H}'$  (which could be  $\mathcal{H}$  itself,  $\mathbb{R}^q$  for some  $q \in \mathbb{N}$ , etc); assuming that  $\mathcal{P}$  has a finite dimensional null space is sufficient in general, but here, for simplicity, we assume that even the space  $\mathcal{H}$  is finite dimensional.

We consider the following semiparametric model:

$$y_i = F(t_i, \theta^*) + g^*(t_i) + \varepsilon_i, \quad (i = 1, 2, \dots, m), \quad (6.1)$$

where  $y_1, \dots, y_m$  are scalar measurements,  $F : \mathcal{I} \times \Theta \rightarrow \mathbb{R}$  is a known nonlinear model function, parameterized by a vector  $\theta \in \Theta \subset \mathbb{R}^p$ ,  $\theta^*$  is the *true* but unknown value of  $\theta$ , and  $g^* \in \mathcal{H}$  is a true but unknown nonlinearity.

Given the measurements  $y_1, \dots, y_m$ , we are interested in finding optimal approxima-

tion  $\hat{\theta}$  of  $\theta^*$  and of the function  $g^* \in \mathcal{H}$ , under the criterion:

$$\min_{\theta \in \Theta, g \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m (y_i - F(t_i, \theta) - g(t_i))^2 \quad \text{such that } g \in \mathcal{H}^\delta, \quad (6.2)$$

where  $\mathcal{H}^\delta := \{g \in \mathcal{H} : \|\mathcal{P}g\|_{\mathcal{H}'} \leq \delta\}$ . The constraint  $g \in \mathcal{H}^\delta$  will be referred to as the *smoothing constraint*, since in general  $\mathcal{H}^\delta$  will be defined so that it contains smooth functions. The smoothing constraint can be easily imposed by adding a penalty term to the nonlinear least squares objective function. The constrained least squares problem (6.2) becomes:

$$\min_{\theta \in \Theta, g \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m (y_i - F(t_i, \theta) - g(t_i))^2 + \lambda \|\mathcal{P}g\|_{\mathcal{H}'}^2. \quad (6.3)$$

### 6.2.2 Spline fitting for the nonparametric part

The regularized nonlinear least squares criterion (6.3) can be rewritten as a double minimization

$$\min_{\theta \in \Theta} \left( \min_{g \in \mathcal{H}} \sum_{i=1}^m (y_i - F(t_i, \theta) - g(t_i))^2 + \lambda \|\mathcal{P}g\|_{\mathcal{H}'}^2 \right),$$

thus it is possible to apply the theory of spline fitting for the nonparametric part  $g$ , in order to solve the inner minimization problem.

We use a parameterization of  $g$  in terms of a family of generators for  $\mathcal{H}$ , i.e.,  $g = \sum_{k=1}^n a_k \phi_k$ , where  $a_1, \dots, a_n$  are free coefficients, and  $\phi_1, \dots, \phi_n$  are basis functions or generators that span the space  $\mathcal{H}$ , assumed to be finite dimensional. Thus, the nonparametric part of the model is simply reduced to a submodel, which is linearly parameterized over  $a_1, \dots, a_n$ , and subject to a regularization constraint. For further reference, here is the *completely parameterized* formulation of the model (6.1):

$$y_i = F(t_i, \theta^*) + \sum_{j=1}^n A_{ij} a_j + \varepsilon_i, \quad (i = 1, 2, \dots, m), \quad (6.4)$$

where  $A$  is the  $m \times n$  matrix that has as elements  $\phi_k(t_i)$ , for  $k$  from 1 to  $n$  and  $i$  from 1 to  $m$ . Using vector notation, the model in problem (6.4) is written as  $\mathbf{y} := \hat{\mathbf{y}} + \boldsymbol{\varepsilon} := \mathbf{F}(\theta) + A\mathbf{a} + \boldsymbol{\varepsilon}$ . This semiparametric model will be fitted using the following regularized nonlinear least squares formulation:

$$\min_{\theta \in \Theta, \mathbf{a} \in \mathbb{R}^n} \frac{1}{m} \|\mathbf{y} - \mathbf{F}(\theta) - A\mathbf{a}\|^2 + \lambda \mathbf{a}^\top C \mathbf{a}, \quad (6.5)$$

with  $C$  the  $n \times n$  matrix having the elements  $\langle \mathcal{P}\phi_k, \mathcal{P}\phi_l \rangle_{\mathcal{H}'}$ , for  $k, l$  from 1 to  $n$ , as in section 5.3.

For a fixed value of  $\lambda \geq 0$ , we denote by  $\hat{\theta}_\lambda$  and  $\hat{\mathbf{a}}_\lambda$  the globally optimal solution of the minimization (6.5). Moreover, we denote with  $\hat{\mathbf{y}}_\lambda$  the optimal model obtained for a fixed  $\lambda$ , that is  $\hat{\mathbf{y}}_\lambda := \mathbf{F}(\hat{\theta}_\lambda) + A\hat{\mathbf{a}}_\lambda$ .

### 6.2.3 Computationally efficient method using the Levenberg-Marquardt algorithm

Denote by  $\mathbf{y}(\theta)$  the  $m$ -dimensional vector with elements  $y_i - F(t_i, \theta)$ , for  $i = 1, \dots, m$ . Thus,  $\mathbf{y}(\theta) = \mathbf{y} - \mathbf{F}(\theta)$ .

For a fixed value of the parameter  $\theta$ , the original nonlinear minimization (6.5) becomes a regularized linear least squares problem only in  $\mathbf{a} \in \mathbb{R}^n$ . Its closed-form solution is  $\mathbf{a}(\theta, \lambda) = (A^\top A + m\lambda C)^{-1} A^\top \mathbf{y}(\theta)$ . Plugging-in this formula in the original optimization problem, we get a nonlinear least squares problem in the variable  $\theta$  alone:

$$\begin{aligned} & \min_{\theta \in \Theta} \frac{1}{m} \|\mathbf{y}(\theta) - A\mathbf{a}(\theta, \lambda)\|_2^2 + \lambda \mathbf{a}(\theta, \lambda)^\top C \mathbf{a}(\theta, \lambda) \Leftrightarrow \\ & \min_{\theta \in \Theta} \frac{1}{m} \left\| \left( I_m - A(A^\top A + m\lambda C)^{-1} A^\top \right) \mathbf{y}(\theta) \right\|_2^2 + \lambda \left\| D(A^\top A + m\lambda C)^{-1} A^\top \mathbf{y}(\theta) \right\|_2^2 \Leftrightarrow \\ & \min_{\theta \in \Theta} \left\| \begin{bmatrix} I_m - A(A^\top A + m\lambda C)^{-1} A^\top \\ \sqrt{m\lambda} D(A^\top A + m\lambda C)^{-1} A^\top \end{bmatrix} \mathbf{y}(\theta) \right\|_2^2, \end{aligned} \quad (6.6)$$

where  $D$  denotes a Cholesky factor of  $C$ , *i.e.*,  $D^\top D = C$ . Denote the coefficient matrix under the norm in (6.6) by  $B(\lambda)$ .

The minimization problem (6.6) can be solved using a nonlinear least squares solver, such as the Levenberg-Marquardt (LM) algorithm. We consider the cases when the parameter set  $\Theta$  is either the full  $\mathbb{R}^p$  or it is defined by linear constraints, for which good implementations of the (modified) LM algorithm are available [90, 18].

A nonlinear least squares solver requires at each new  $\theta$  the evaluation of the function  $f(\theta, \lambda) := B(\lambda)\mathbf{y}(\theta)$  and of the Jacobian  $J(\theta, \lambda) := B(\lambda)\nabla\mathbf{y}(\theta)$ . Efficient computations of these two ingredients are essential for the overall computational time. Subsection 6.2.4 shows more details on how to use the generalized singular value decomposition [50] as a preprocessing step and to increase the efficiency of the computations in every iteration.

### 6.2.4 Efficient computation of function and Jacobian values

If the evaluation of the nonlinear function  $F(t, \theta)$  (as a function of  $\theta$ ) and of its gradient (also with respect to  $\theta$ ) are not very computationally demanding, then the evaluation of  $f(\theta, \lambda)$  and  $J(\theta, \lambda)$ , defined at the end of Section 6.2.3, for any values of the parameters  $\theta \in \Theta$  and  $\lambda > 0$ , can be achieved with linear computational complexity in the problem dimensions  $m$  (number of regression points),  $n$  (number of basis functions) and  $p$  (length of the vector  $\theta$ ).

In the computation of the function and Jacobian values, the inverse of the matrix  $A^\top A + m\lambda C$  appears. Dealing with this inverse would be much more efficient if  $A$  and  $C$  were diagonal matrices, instead of full matrices: then the “inversion” would only involve a diagonal matrix, for any possible value of  $\lambda$ . Algorithm 6.1 shows the preprocessing operations that should be executed beforehand, *i.e.*, the simultaneous diagonalization of the matrices  $A^\top A$  and  $C$ , achieved using the generalized singular value decomposition.<sup>8</sup>

<sup>8</sup>In practical implementations, we use the “economic” generalized singular value decomposition, but here, for a simplified presentation, we only give formulas using the full decomposition, *i.e.*, with square orthogonal matrix of generalized singular vectors.

---

**Algorithm 6.1** Preprocessing. complexity


---

**Input:** matrices  $A \in \mathbb{R}^{m \times n}$  and  $C \in \mathbb{R}^{n \times n}$ .

- 1: Compute Cholesky decomposition of  $C$ ,  $C = D^\top D$ ;  $n^3/3$
- 2: Compute GSVD of the pair  $(A, D)$ ;  $\mathcal{O}(mn^2) + \mathcal{O}(n^3)$

$$A = U \Sigma_A X^{-1}, \quad D = V \Sigma_D X^{-1},$$

with  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$  orthogonal,  $X \in \mathbb{R}^{n \times n}$  invertible,  
and  $\Sigma_A \in \mathbb{R}^{m \times n}$ ,  $\Sigma_D \in \mathbb{R}^{n \times n}$  positive diagonal matrices.

**Output:** Elements of the GSVD  $U, V, \Sigma_A, \Sigma_D, X$ . total:  $\mathcal{O}(mn^2) + \mathcal{O}(n^3)$ .


---

Algorithm 6.2 shows how to employ the preprocessing step in order to compute the function and Jacobian values in a fast way, for any parameters  $\theta$  and  $\lambda$ . Note that the non-linear least squares minimization (6.6) involves the 2-norm of  $f(\theta, \lambda)$ ; therefore, multiplication of  $f(\theta, \lambda)$  from the left with an orthogonal matrix does not change the optimization criterion. Algorithm 6.2 exploits this fact in order to avoid unnecessary matrix multiplications with the orthogonal matrices  $U$  and  $V$  from Algorithm 6.1, and provides orthogonally transformed function and Jacobian values.

---

**Algorithm 6.2** Compute function value  $f(\theta, \lambda)$ , and Jacobian  $J(\theta, \lambda)$ . complexity


---

$$f(\theta, \lambda) := B(\lambda) \mathbf{y}(\theta), \quad J(\theta, \lambda) := B(\lambda) \nabla \mathbf{y}(\theta),$$

$$\text{where } B(\lambda) = \begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix} \begin{bmatrix} I_m - A(A^\top A + m\lambda C)^{-1} A^\top \\ \sqrt{m\lambda} D(A^\top A + m\lambda C)^{-1} A^\top \end{bmatrix}.$$

**Input:** matrices  $U, V, \Sigma_A, \Sigma_D$ , computed from  $A$  and  $C$  via Algorithm 6.1,  
parameters  $\lambda$  and  $\theta$ .

- 1: Evaluate nonlinear expressions depends on nonlinear  $\mathbf{y}$   
 $v \leftarrow \mathbf{y}(\theta) \in \mathbb{R}^m$  and  $G \leftarrow \nabla \mathbf{y}(\theta) \in \mathbb{R}^{m \times p}$ .  $\mathcal{O}(mn)$ , resp.  $\mathcal{O}(mnp)$
- 2:  $v \leftarrow U^\top v$ ,  $G \leftarrow U^\top G$ .  $\mathcal{O}(n)$
- 3:  $c_1 \leftarrow \text{diag}(I_m - \Sigma_A (\Sigma_A^\top \Sigma_A + m\lambda \Sigma_D^2)^{-1} \Sigma_A^\top)$   $\mathcal{O}(n)$
- 4:  $c_2 \leftarrow \sqrt{m\lambda} \text{diag} \Sigma_D (\Sigma_A^\top \Sigma_A + m\lambda \Sigma_D^2)^{-1} \Sigma_A^\top$ .  $\mathcal{O}(n)$
- 5: Set  $f(\theta, \lambda) \leftarrow \begin{bmatrix} c_1 \odot v \\ c_2 \odot v_{1:n} \end{bmatrix}$ , where  $\odot$  denotes element-wise product.  $\mathcal{O}(m+n)$
- 6: **for**  $j = 1, \dots, p$  **do**
- 7:  $\text{col}_j(J(\theta, \lambda)) \leftarrow \begin{bmatrix} c_1 \odot G_{:,j} \\ c_2 \odot G_{1:n,j} \end{bmatrix}$ ,  $\mathcal{O}(m+n)$
- 8: **end for**

**Output:**  $f(\theta, \lambda)$  and  $J(\theta, \lambda)$ . total:  $\geq \mathcal{O}(mn + m + 3n)$ , resp.  $\geq \mathcal{O}(mnp + np + mp)$ 


---

### 6.2.5 Choice of regularization parameter

Although many model selection methods can be adapted to our setting, we concentrate herein only on the derivation of the generalized cross validation criterion.

To explain the GCV criterion in this context, we particularize the more general derivation from Appendix A.1. We start as in [136] from the leave-one-out cross validation, which chooses a  $\lambda$  that minimizes the function

$$CV(\lambda) := \frac{1}{m} \sum_{i=1}^m \left( y_i - F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda^{[-i]} \right) - A_i \widehat{\mathbf{a}}_\lambda^{[-i]} \right)^2,$$

where  $A_i$  is the  $i^{\text{th}}$  row of  $A$  and where  $\widehat{\boldsymbol{\theta}}_\lambda^{[-i]}$  and  $\widehat{\mathbf{a}}_\lambda^{[-i]}$  are the solution of problem (6.5), when the data point  $(t_i, y_i)$  is omitted from  $(\mathbf{t}, \mathbf{y})$ . This formulation is inconvenient since it involves solving  $m$  problems of the type (6.5), one for each deleted data point. We show that also in the nonlinear semiparametric setting it is possible, as in the classical smoothing splines case, to simplify this formulation and to only need the solution of the total problem (6.5) for evaluating the cross validation function.

Firstly, we emphasize the influence that the  $i^{\text{th}}$  measured output  $y_i$  has on the optimal solution of (6.5), for a fixed value of  $\lambda$  and for fixed values of the other data points in  $y_1, \dots, y_m$ . We denote the direct link between  $y_i$  and the corresponding component of the optimal model  $\widehat{\mathbf{y}}_\lambda$  by a function  $h$  such that  $h(y_i) = (\widehat{\mathbf{y}}_\lambda)_i = F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda \right) + A_i \widehat{\mathbf{a}}_\lambda$ .

Secondly, we note that the leaving-one-out lemma that was proven in the context of smoothing splines [19] still holds trivially for our semiparametric problem. It ensures that if the measured  $y_i$  was by any chance equal to the function value predicted by the solution computed without the  $i^{\text{th}}$  measurement, *i.e.*,  $y_i = (\widehat{\mathbf{y}}_\lambda^{[-i]})_i := F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda^{[-i]} \right) + A_i \widehat{\mathbf{a}}_\lambda^{[-i]}$ , then the vectors  $\widehat{\boldsymbol{\theta}}_\lambda^{[-i]}$  and  $\widehat{\mathbf{a}}_\lambda^{[-i]}$  would be the optimal solution for the complete problem (6.5). We can write this observation in terms of the function  $h$  as  $h((\widehat{\mathbf{y}}_\lambda^{[-i]})_i) = (\widehat{\mathbf{y}}_\lambda^{[-i]})_i$ .

Using a similar trick as in [136] for smoothing splines and in [69] for nonlinear nonparametric regression, we have:

$$y_i - F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda^{[-i]} \right) - A_i \widehat{\mathbf{a}}_\lambda^{[-i]} = \frac{y_i - F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda \right) - A_i \widehat{\mathbf{a}}_\lambda}{1 - \Delta_i(\lambda)},$$

where

$$\Delta_i(\lambda) := \frac{\left( F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda \right) + A_i \widehat{\mathbf{a}}_\lambda \right) - \left( F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda^{[-i]} \right) + A_i \widehat{\mathbf{a}}_\lambda^{[-i]} \right)}{y_i - F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda^{[-i]} \right) - A_i \widehat{\mathbf{a}}_\lambda^{[-i]}} = \frac{h(y_i) - h((\widehat{\mathbf{y}}_\lambda^{[-i]})_i)}{y_i - (\widehat{\mathbf{y}}_\lambda^{[-i]})_i},$$

which holds whenever  $y_i \neq (\widehat{\mathbf{y}}_\lambda^{[-i]})_i$ .  $\Delta_i(\lambda)$  is a divided difference for the function  $h$ , which can be approximated with the derivative of  $h$ . This leads to the definition of the following generalized influence (or smoother or hat) matrix  $S(\lambda)$ , which agrees with the definition given by [89] for the generalized influence matrix in a nonlinear context:

$$S(\lambda)_{ij} := \frac{\partial \left( F \left( t_i, \widehat{\boldsymbol{\theta}}_\lambda \right) + A_i \widehat{\mathbf{a}}_\lambda \right)}{\partial y_j} = \frac{\partial (\widehat{\mathbf{y}}_\lambda)_i}{\partial y_j},$$

for which  $S(\lambda)_{ii} \approx \Delta_i(\lambda)$ , as a first order approximation.

The ordinary cross validation cost function can be approximated by

$$CV(\lambda) \approx \frac{1}{m} \sum_{i=1}^m \left( y_i - F(t_i, \hat{\theta}_\lambda) - A_i \hat{\mathbf{a}}_\lambda \right)^2 / (1 - S(\lambda)_{ii})^2,$$

and the generalized cross validation is its “rotation-invariant” version:

$$G(\lambda) = \frac{1}{m} \sum_{i=1}^m \left( y_i - F(t_i, \hat{\theta}_\lambda) - A_i \hat{\mathbf{a}}_\lambda \right)^2 / \left[ \frac{1}{m} \text{Tr}(I_m - S(\lambda)) \right]^2 = \frac{m \|\hat{\mathbf{y}}_\lambda - \mathbf{y}\|_2^2}{[\text{Tr}(I_m - S(\lambda))]^2}. \quad (6.7)$$

In Chapter 5, Section 5.3.2, we have seen that for spline smoothing (which is a linear regularization problem), the influence matrix is  $A(\lambda) = A(A^\top A + m\lambda C)^{-1} A^\top$ .

For our semiparametric model, we derive the following result.

**Lemma 6.1.** *The generalized influence matrix for the semiparametric model (6.4) is given by*

$$S(\lambda) \approx \begin{bmatrix} \nabla \mathbf{F}(\hat{\theta}_\lambda) & A \end{bmatrix} \begin{bmatrix} \nabla \mathbf{F}(\hat{\theta}_\lambda)^\top \nabla \mathbf{F}(\hat{\theta}_\lambda) & \nabla \mathbf{F}(\hat{\theta}_\lambda)^\top A \\ A^\top \nabla \mathbf{F}(\hat{\theta}_\lambda) & A^\top A + m\lambda C \end{bmatrix}^{-1} \begin{bmatrix} \nabla \mathbf{F}(\hat{\theta}_\lambda)^\top \\ A^\top \end{bmatrix}, \quad (6.8)$$

where the approximation sign indicates that some high order terms are ignored.

*Proof.* We compute explicitly

$$S(\lambda) = \frac{\partial(\hat{\mathbf{y}}_\lambda)}{\partial \mathbf{y}} = \frac{\partial(\mathbf{F}(\hat{\theta}_\lambda) + A\hat{\mathbf{a}}_\lambda)}{\partial \mathbf{y}} = \frac{\partial(\mathbf{F}(\hat{\theta}_\lambda) + A\hat{\mathbf{a}}_\lambda)}{\partial(\theta, \mathbf{a})} \frac{\partial(\hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda)}{\partial \mathbf{y}}. \quad (6.9)$$

Denote by  $\mathbf{E}$  the regularized least squares objective function, but with the dependence on the measured  $\mathbf{y}$  explicitly marked:

$$\mathbf{E}(\mathbf{y}, \theta, \mathbf{a}; \lambda) := \|\mathbf{y} - \mathbf{F}(\theta) - A\mathbf{a}\|_2^2 + m\lambda \mathbf{a}^\top C \mathbf{a}.$$

Since  $\hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda$  are optimal for given data  $\mathbf{y}$ , it means that  $\partial \mathbf{E}(\mathbf{y}, \hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda; \lambda) / \partial(\theta, \mathbf{a}) = 0$ . On the other hand, if  $\mathbf{y}$  is ‘perturbed’ and the data becomes  $\mathbf{y} + d\mathbf{y}$ , the optimum also changes. We denote the new optimal solution by  $(\hat{\theta}_\lambda + d\theta, \hat{\mathbf{a}}_\lambda + d\mathbf{a})$ ; it satisfies

$$\partial \mathbf{E}(\mathbf{y} + d\mathbf{y}, \hat{\theta}_\lambda + d\theta, \hat{\mathbf{a}}_\lambda + d\mathbf{a}; \lambda) / \partial(\theta, \mathbf{a}) = 0.$$

Using a first order Taylor approximation, it implies that

$$\frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda; \lambda)}{\partial(\theta, \mathbf{a}) \partial \mathbf{y}^\top} d\mathbf{y} + \frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda; \lambda)}{\partial(\theta, \mathbf{a}) \partial(\theta, \mathbf{a})^\top} d(\theta, \mathbf{a}) \approx 0. \quad (6.10)$$

It is easy to see from the formula of  $\mathbf{E}$  that

$$\frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda; \lambda)}{\partial(\theta, \mathbf{a}) \partial \mathbf{y}^\top} = -2 \frac{\partial(\mathbf{F}(\hat{\theta}_\lambda) + A\hat{\mathbf{a}}_\lambda)^\top}{\partial(\theta, \mathbf{a})} = -2 \begin{bmatrix} \nabla \mathbf{F}(\hat{\theta}_\lambda) & A \end{bmatrix}^\top$$

and the block of the Hessian of  $\mathbf{E}$  corresponding to  $(\boldsymbol{\theta}, \mathbf{a})$  is

$$\begin{aligned} 2\widehat{H}(\lambda) &= \begin{bmatrix} \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} E(\mathbf{y}, \widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda; \lambda) & \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \mathbf{a}^\top} E(\mathbf{y}, \widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda; \lambda) \\ \frac{\partial^2}{\partial \mathbf{a} \partial \boldsymbol{\theta}^\top} E(\mathbf{y}, \widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda; \lambda) & \frac{\partial^2}{\partial \mathbf{a} \partial \mathbf{a}^\top} E(\mathbf{y}, \widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda; \lambda) \end{bmatrix} \\ &\approx 2 \begin{bmatrix} \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda)^\top \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) & \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda)^\top A \\ A^\top \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) & A^\top A + m\lambda C \end{bmatrix}, \end{aligned}$$

where the approximation appears only in the (1,1) block of the matrix  $\widehat{H}(\lambda)$  and is related to ignoring a term containing the second order differential of  $\mathbf{F}$  (which is a tensor) and the residual vector  $\mathbf{y} - \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) - A\widehat{\mathbf{a}}_\lambda$ . It is customary to ignore such a difficult-to-compute term, and thus use a *pseudo-Hessian*, since this approximation is made at converged solutions, where the residual is usually small.

Since  $d\mathbf{y}$  and  $d(\boldsymbol{\theta}, \mathbf{a})$  represent small perturbations on  $\mathbf{y}$  and  $(\widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda)$ , we can use the relation (6.10) and set the matrix  $\widehat{H}(\lambda)^{-1} \begin{bmatrix} \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) & A \end{bmatrix}^\top$  as approximation for the differential  $\frac{\partial(\widehat{\boldsymbol{\theta}}_\lambda, \widehat{\mathbf{a}}_\lambda)}{\partial \mathbf{y}}$  in (6.9). The formula (6.8) of  $S(\lambda)$  now follows.  $\square$

The evaluation and minimization of the GCV function (6.7) is more difficult than in classical linear spline fitting, because there is no closed form expression for the optimal regularized estimate  $\widehat{\boldsymbol{\theta}}_\lambda$ , which enters implicitly into the computation of  $G(\lambda)$  through the formula of  $\widehat{\mathbf{y}}_\lambda$  and of  $S(\lambda)$ .

In Algorithm 6.3, a scalar minimization algorithm (*e.g.*, a golden section method) is used for minimizing the GCV function, while at each iteration a nonlinear least squares minimization is carried out for estimating  $\widehat{\boldsymbol{\theta}}_\lambda$ , at the current value of  $\lambda$ . (Remember from subsection 6.2.3 that once we have an estimated  $\boldsymbol{\theta}$  we have a corresponding  $\mathbf{a}$  in closed form.) Often in applications, the GCV function is unimodal thus it is natural to search for a globally optimal  $\lambda$  via a scalar minimization method. However, there is no proof for such an observation. Any scalar optimization method could be used (Algorithm 6.3 exemplifies a variant with golden section search method), or improved combinations, *e.g.*, steps of a golden section search can be alternated with a parabolic search. All methods require one GCV function evaluation per iteration, and correspondingly, one nonlinear least squares solution. Both the (nonlinear least squares) inner minimization and the outer scalar minimization over  $\lambda$  in Algorithm 6.3 are globally convergent; but different initial values of the parameters might give different converged local solutions, since we deal with nonlinear (nonconvex) optimization.

Since at every iteration a nonlinear optimization problems is solved, this operation should be made as efficient as possible. For the optimization on  $\boldsymbol{\theta}$  (step 4:), one should use efficient function and Jacobian computations as described in subsection 6.2.4. For the evaluation of the GCV criterion (step 5:), efficient computations are described in subsection 6.2.6.

**Remark 12** We have also experimented with the procedure shortly outlined in §9.1 of [136], which suggested to employ an iterative nonlinear minimization method where at each iteration a new  $\lambda$ , optimal for the GCV of the linearized nonlinear function at the current iterates, is used. However, this method didn't exhibit a good convergence behavior compared to Algorithm 6.3. This suggests that applying the linearization in early iterations,

---

**Algorithm 6.3** Algorithm for regularized nonlinear least squares with adaptive choice of the regularization parameter; golden section variant

---

**Input:** Measurements vector  $\mathbf{y}$ , deterministic function  $\mathbf{F}$ , matrix  $A$  for spline basis and matrix  $C$  for smoothing constraint. Initial approximation  $\theta_0 \in \Theta$  and bounds  $0 \leq \lambda_{\min} < \lambda_{\max}$ . Convergence tolerance  $tol$ . Scalar  $\gamma \in (0, 0.5)$ , e.g.,  $\gamma \cong 0.3819$  defined by the golden section

- 1: Initialize:  $\theta \leftarrow \theta_0$ ,  $\lambda_{\text{left}} \leftarrow \lambda_{\min}$ ,  $\lambda_{\text{right}} \leftarrow \lambda_{\max}$
- 2: **repeat**
- 3:    $\lambda_1 \leftarrow \lambda_{\text{left}}(1 - \gamma) + \lambda_{\text{right}}\gamma$ ,  $\lambda_2 \leftarrow \lambda_{\text{left}}\gamma + \lambda_{\text{right}}(1 - \gamma)$
- 4:   compute / update  $\theta_{\lambda_1}$  and  $\theta_{\lambda_2}$  as the optimal solutions of the NLS criterion (6.6), for the fixed values  $\lambda_1$  and  $\lambda_2$ , respectively
- 5:   evaluate / update the GCV function values at  $\lambda_{\text{left}}$ ,  $\lambda_{\text{right}}$ ,  $\lambda_1$  and  $\lambda_2$ , using formula (6.7)
- 6:   **if**  $G(\lambda_1) > G(\lambda_2)$  **then**
- 7:      $\lambda_{\text{left}} \leftarrow \lambda_1$
- 8:   **else**
- 9:      $\lambda_{\text{right}} \leftarrow \lambda_2$
- 10:   **end if**
- 11: **until**  $\lambda_2 - \lambda_1 < tol$
- 12:  $\hat{\lambda} \leftarrow \lambda_1$ ,  $\hat{\theta} \leftarrow \theta_{\hat{\lambda}}$ ,  $\hat{\mathbf{a}} \leftarrow \mathbf{a}(\hat{\theta}, \hat{\lambda})$ .

**Output:** Solution  $(\hat{\theta}, \hat{\mathbf{a}})$  and smoothing parameter  $\hat{\lambda}$ .

---

where we might be far from the optimal solution, is not appropriate for highly nonlinear functions.

### 6.2.6 Efficient computation of the GCV function value

The evaluation of the GCV function defined in (6.7) involves computing the misfit between the data  $\mathbf{y}$  and the current model  $\hat{\mathbf{y}}_{\lambda}$ , for the numerator, and computing the trace of the influence matrix  $S(\lambda)$  in (6.8), for the denominator. The estimated model  $\hat{\mathbf{y}}_{\lambda}$  is easily available from the nonlinear least squares step, thus the numerator is computed in  $\mathcal{O}(m)$ .

Since the Jacobian matrix  $\nabla \mathbf{F}(\hat{\theta}_{\lambda})$  changes at each new evaluation (together with the current iterate  $\hat{\theta}_{\lambda} = \theta_{\lambda_k}$ ), it is not possible to compute the influence matrix as efficiently as in the classical linear setting [51]. However, we propose to use also the preprocessing step described in Algorithm 6.1 in order to create an easily computable diagonal part in the influence matrix formula. The term  $G := U^{\top} \nabla \mathbf{F}(\hat{\theta}_{\lambda})$  is already available from the nonlinear least squares function and Jacobian evaluation, see step 2: in Algorithm 6.2. The influence matrix becomes (up to an orthogonal similarity transformation, which leaves the trace invariant):

$$S(\lambda) = \begin{bmatrix} G & \Sigma_A \end{bmatrix} \begin{bmatrix} G^{\top} G & G^{\top} \Sigma_A \\ \Sigma_A^{\top} G & \Sigma_A^{\top} \Sigma_A + m\lambda \Sigma_D^2 \end{bmatrix}^{-1} \begin{bmatrix} G^{\top} \\ \Sigma_A^{\top} \end{bmatrix},$$

where  $U$ ,  $\Sigma_A$  and  $\Sigma_D$  are matrices provided by Algorithm 6.1. After some manipulations

we get:

$$S(\lambda) = A(\lambda) + (I_m - A(\lambda))G \left[ G^\top (I_m - A(\lambda))G \right]^{-1} G^\top (I_m - A(\lambda)),$$

where  $A(\lambda) = \Sigma_A(\Sigma_A^\top \Sigma_A + m\lambda \Sigma_D^2)^{-1} \Sigma_A^\top$ . Finally, using a QR decomposition of  $(I_m - A(\lambda))^{1/2}G = QR$  yields the simplified formula

$$\begin{aligned} \text{Tr}S(\lambda) &= \text{Tr}A(\lambda) + \text{Tr}R(R^\top R)^{-1}R^\top Q^\top (I_m - A(\lambda))Q \\ &= \text{Tr}A(\lambda) + p - \text{Tr}R(R^\top R)^{-1}R^\top Q^\top A(\lambda)Q. \end{aligned}$$

Note that  $\text{Tr}S(\lambda)$  is also used in the computation of confidence intervals (see subsection 6.3.2), since it represents the number of effective parameters.

## 6.3 Asymptotic properties of semiparametric regression

### 6.3.1 Asymptotic normality

Although we do not impose additional regularization constraints on  $\theta$ , the smoothness constraint that is imposed on the nonparametric part  $g$  also influences the computation of the parameter of interest,  $\theta$ . We describe precisely what this influence is on the bias and variance of the estimator  $\hat{\theta}_\lambda$  in the following theorem.

**Theorem 6.2.** *Let  $(t_i, y_i)$ ,  $i = 1, 2, \dots, m$ , denote  $m$  observations from a semiparametric model with partially known functional relationship,*

$$y_i = F(t_i, \theta^*) + g^*(t_i) + \varepsilon_i, \quad (i = 1, 2, \dots, m),$$

where  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ ,  $t_1, \dots, t_m$  are regression abscissas from a real interval  $\mathcal{I}$ ,  $\theta^*$  denotes the true value of the unknown parameter  $\theta \in \Theta \subset \mathbb{R}^p$ , and  $g^*$  is the true 'baseline' function. Assume that:

- (A1).  $g^* \in \mathcal{H}$ , for a certain given finite dimensional Hilbert space  $\mathcal{H}$ ;
- (A2).  $\phi_1, \dots, \phi_n \in \mathcal{H}$  is a family of generators in  $\mathcal{H}$ ,  $A$  denotes the  $m \times n$  matrix that has as elements  $\phi_k(t_i)$ , for  $k$  from 1 to  $n$  and  $i$  from 1 to  $m$ , and  $C$  denotes the  $n \times n$  matrix having the elements  $\langle \mathcal{P}\phi_k, \mathcal{P}\phi_l \rangle_{\mathcal{H}'}$ , for  $k, l$  from 1 to  $n$ , where  $\mathcal{P}$  and  $\mathcal{H}'$  have been introduced in subsection 6.2.1.

Then, for any  $\lambda \geq 0$ , the estimates of  $\theta^*$  and  $\mathbf{a}^*$  (where  $\mathbf{a}^*$  satisfies  $A\mathbf{a}^* = g^*$ ) defined as the global minimizer  $(\hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda)$  of (6.5) over  $\theta \in \Theta$ ,  $\mathbf{a} \in \mathbb{R}^n$ , satisfy, when  $m \rightarrow \infty$ :

$$\begin{aligned} & \begin{bmatrix} \hat{\theta}_\lambda \\ \hat{\mathbf{a}}_\lambda \end{bmatrix} \sim \\ & \mathcal{N} \left( \begin{bmatrix} \theta^* \\ \mathbf{a}^* \end{bmatrix} - H(\lambda)^{-1} \begin{bmatrix} 0 \\ m\lambda C\mathbf{a}^* \end{bmatrix}, \sigma^2 H(\lambda)^{-1} \left( H(\lambda) - \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right) H(\lambda)^{-1} \right), \end{aligned} \quad (6.11)$$

$$\text{where } H(\lambda) := \begin{bmatrix} \nabla \mathbf{F}(\theta^*)^\top \nabla \mathbf{F}(\theta^*) & \nabla \mathbf{F}(\theta^*)^\top A \\ A^\top \nabla \mathbf{F}(\theta^*) & A^\top A + m\lambda C \end{bmatrix}.$$

**Proof.** When  $m \rightarrow \infty$ , we can use a first order approximation of the nonlinear function  $\mathbf{F}$ :

$$\mathbf{F}(\theta) \approx \mathbf{F}(\theta^*) + \nabla \mathbf{F}(\theta^*)(\theta - \theta^*),$$

since we expect that at convergence the bias between  $\hat{\theta}$  and  $\theta^*$  is not so big.

Replacing  $\mathbf{y} = \mathbf{F}(\theta^*) + A\mathbf{a}^* + \varepsilon$  into the minimization (6.5) leads to:

$$\min_{\theta \in \Theta, \mathbf{a} \in \mathbb{R}^n} \frac{1}{m} \|J\theta^* + A\mathbf{a}^* + \varepsilon - J\theta - A\mathbf{a}\|^2 + \lambda \mathbf{a}^\top C \mathbf{a}, \quad (6.12)$$

where we used the short-hand notation  $J := \nabla \mathbf{F}(\theta^*)$ . From the first order optimality conditions, we get

$$\begin{aligned} \begin{bmatrix} \hat{\theta}_\lambda \\ \hat{\mathbf{a}}_\lambda \end{bmatrix} &= \begin{bmatrix} J & A \\ 0 & \sqrt{m\lambda}D \end{bmatrix}^\dagger \begin{bmatrix} J\theta^* + A\mathbf{a}^* + \varepsilon \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \theta^* \\ \mathbf{a}^* \end{bmatrix} - H(\lambda)^{-1} \begin{bmatrix} 0 \\ m\lambda C \mathbf{a}^* \end{bmatrix} + H(\lambda)^{-1} \begin{bmatrix} J^\top \\ A^\top \end{bmatrix} \varepsilon, \end{aligned}$$

where  $D$  is a Cholesky factor of  $C$ , i.e.,  $D^\top D = C$ .

The assumptions on the noise term  $\varepsilon$  imply the normality of the estimator, with the bias from the true values and the covariance matrix given in the conclusion of the theorem.  $\square$

**Remark 13** The assumptions (A1-2) can be relaxed, at the expense of a more technical proof. More specifically, the true nonlinearity  $g^*$  might violate (A1) by not being in  $\mathcal{H}$ , but then we should consider an asymptotic case with respect to the number of spline basis functions  $n$  such that when  $n$  goes to infinity  $g^*$  should become arbitrarily close to a member of  $\mathcal{H}$ . (A2) can be replaced by the condition that the null space of the operator  $\mathcal{P}$  in  $\mathcal{H}$  is finite dimensional, in the case when  $\mathcal{H}$  itself is infinite dimensional.

Theorem 6.2 gives, for the case  $\lambda = 0$ , similar conclusions as the results on nuisance parameter analysis of [120]: the estimator is unbiased and its covariance depends on the nuisance term (in our case, the baseline parameter  $\mathbf{a}^*$ ). The difference in the  $\lambda \neq 0$  case is, however, that we must also incorporate the contribution of the regularization term; thus, there will be a bias, but the covariance can be of smaller magnitude. An adequate choice of  $\lambda$  should provide an optimal bias-variance trade-off. The GCV criterion provides such a trade-off. We observe next that the bias and covariance formulas are very much linked to the formula of the generalized smoother matrix  $S(\lambda)$  in (6.9), where instead of the estimated  $(\hat{\theta}_\lambda, \hat{\mathbf{a}}_\lambda)$  we plug in the values of the true parameters  $(\theta^*, \mathbf{a}^*)$ .

Denote by  $T$  the gradient  $\begin{bmatrix} J & A \end{bmatrix}$ , where as before  $J = \nabla \mathbf{F}(\theta^*)$ . Then the smoother matrix at the true values becomes  $S(\lambda) = TH(\lambda)^{-1}T^\top$ .

Whenever the linear approximation  $\mathbf{F}(\theta) \approx \mathbf{F}(\theta^*) + J(\theta - \theta^*)$  holds, we have as in the proof of the Theorem 6.2 that

$$\hat{\mathbf{y}}_\lambda := \mathbf{F}(\hat{\theta}_\lambda) + A\hat{\mathbf{a}}_\lambda \approx T \begin{bmatrix} J & A \\ 0 & \sqrt{m\lambda}D \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} = TH(\lambda)^{-1}T^\top \mathbf{y} = S(\lambda)\mathbf{y}.$$

Thus, we obtain the approximation of a classical relation,  $\mathbf{y} - \widehat{\mathbf{y}}_\lambda \approx (I_m - S(\lambda))\mathbf{y}$ . This means that the GCV criterion for the regularized nonlinear problem can be approximated by the GCV criterion for the regularized linear problem (6.12), where the linear approximation is plugged in. In the asymptotic linear case (see [136, §4.4]), the optimal  $\lambda$  given by GCV approaches the value of  $\lambda$  minimizing the *predictive mean square error* criterion

$$\|J\theta^* + A\mathbf{a}^* - J\widehat{\theta}_\lambda - A\widehat{\mathbf{a}}_\lambda\|_2^2.$$

This implies bounds on the forward error  $\|\theta^* - \widehat{\theta}_\lambda\|_2^2 + \|\mathbf{a}^* - \widehat{\mathbf{a}}_\lambda\|_2^2$ , as well.

Moreover, note that the bias in (6.11) can be rewritten as

$$\begin{aligned} -H(\lambda)^{-1} \begin{bmatrix} 0 \\ m\lambda C\mathbf{a}^* \end{bmatrix} &= -H(\lambda)^{-1}(H(\lambda) - T^\top T) \begin{bmatrix} \theta^* \\ \mathbf{a}^* \end{bmatrix} \\ &= -(I_{p+n} - H(\lambda)^{-1}T^\top T) \begin{bmatrix} \theta^* \\ \mathbf{a}^* \end{bmatrix} \end{aligned}$$

and the covariance matrix in (6.11) satisfies

$$\mathcal{C} = \sigma^2 H(\lambda)^{-1} T^\top T H(\lambda)^{-1}, \quad \text{and} \quad T^\top \mathcal{C} T = \sigma^2 S(\lambda)^2.$$

The magnitude of the matrix  $I_{p+n} - H(\lambda)^{-1}T^\top T$  influences the bias term. One measure of this magnitude could be the trace, and  $\text{Tr}(I_{p+n} - H(\lambda)^{-1}T^\top T) = p+n - \text{Tr}S(\lambda)$ . In nonlinear or regularized models, the quantity  $p_{\text{eff}} := \text{Tr}S(\lambda)$ , the trace of the influence matrix, has the meaning of the *effective number of parameters* [89]; in other words, the number  $p_{\text{eff}}$  replaces  $p+n$ , the number of components in the regression variable  $\theta$  and  $\mathbf{a}$ . This quantity comes into play when we estimate the noise variance in the semiparametric model (generalizing [136, Equation (5.1.3)]):

$$\widehat{\sigma}^2 := \frac{\|\mathbf{y} - \mathbf{F}(\widehat{\theta}_\lambda) - A\widehat{\mathbf{a}}_\lambda\|_2^2}{m - p_{\text{eff}}} \approx \frac{\|(I_m - S(\lambda))\mathbf{y}\|_2^2}{\text{Tr}(I_m - S(\lambda))}. \quad (6.13)$$

### 6.3.2 Asymptotic confidence intervals

In this section, we derive practical statistical information, as corollaries of Theorem 6.2. In essence, one is interested in finding confidence intervals for the parameter of interest  $\theta$ . As usual when dealing with Fisher information matrices, one cannot compute the exact covariance matrix (since it depends on the true unknown parameter); thus, it is common to replace the true parameters (in our case,  $\theta^*$  and  $\mathbf{a}^*$ ) by their estimated values (*i.e.*,  $\widehat{\theta}_\lambda$  and  $\widehat{\mathbf{a}}_\lambda$ ). Due to the same reason, we can only perform an approximate bias correction, which follows by replacing  $H(\lambda)$  with  $\widehat{H}(\lambda)$  in the following corollary.

**Corollary 6.3.** *With the notation of Theorem 6.2, an unbiased estimate of  $[\theta^{*\top}, \mathbf{a}^{*\top}]^\top$  is given by*

$$\begin{bmatrix} \widehat{\theta}_\lambda \\ \widehat{\mathbf{a}}_\lambda \end{bmatrix} := \begin{bmatrix} \widehat{\theta}_\lambda \\ \widehat{\mathbf{a}}_\lambda \end{bmatrix} + \left( H(\lambda) - \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ m\lambda C\widehat{\mathbf{a}}_\lambda \end{bmatrix}.$$

**Proof.** From Theorem 6.2, we see that  $[\widehat{\boldsymbol{\theta}}_\lambda^\top, \widehat{\mathbf{a}}_\lambda^\top]^\top$  is centered around

$$\begin{bmatrix} \boldsymbol{\theta}^* \\ \mathbf{a}^* \end{bmatrix} - H(\lambda)^{-1} \begin{bmatrix} 0 \\ m\lambda C \mathbf{a}^* \end{bmatrix} = \left( I - H(\lambda)^{-1} \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\theta}^* \\ \mathbf{a}^* \end{bmatrix}.$$

Thus, using the matrix inversion lemma,

$$\begin{aligned} \left( I - H(\lambda)^{-1} \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right)^{-1} \begin{bmatrix} \widehat{\boldsymbol{\theta}}_\lambda \\ \widehat{\mathbf{a}}_\lambda \end{bmatrix} &= \left( I + \left( H(\lambda) - \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right)^{-1} \right. \\ &\quad \left. \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right) \begin{bmatrix} \widehat{\boldsymbol{\theta}}_\lambda \\ \widehat{\mathbf{a}}_\lambda \end{bmatrix} \end{aligned}$$

is an unbiased estimate of  $[\boldsymbol{\theta}^{*\top}, \mathbf{a}^{*\top}]^\top$ .  $\square$

At this point, we are ready to give formulas for  $100(1 - \alpha)\%$  confidence intervals<sup>9</sup> for each of the parameters  $\theta_i, i = 1, \dots, p$ .

**Corollary 6.4.** *An approximate  $100(1 - \alpha)\%$  confidence interval for the parameter  $\theta_i$  is given by*

$$(\widehat{\boldsymbol{\theta}}_\lambda)_i \pm t_{m-p_{\text{eff}}}^{\alpha/2} \cdot (\widehat{\boldsymbol{\sigma}}^2 \widehat{\mathcal{C}}_{ii}^\theta)^{1/2},$$

where  $t_k^\alpha$  denotes the  $\alpha$  quantile of the Student  $t$  distribution with  $k$  degrees of freedom,  $\widehat{\boldsymbol{\sigma}}^2$  is given in (6.13), and  $\widehat{\mathcal{C}}_{ii}^\theta$  is the  $i^{\text{th}}$  diagonal element of the covariance matrix of  $\widehat{\boldsymbol{\theta}}_\lambda$ ,

$$\begin{aligned} \widehat{\mathcal{C}}^\theta &:= \widehat{\mathcal{G}}^\top \widehat{\mathcal{G}}, \quad \text{with } \widehat{\mathcal{G}} := (I_m - A(\lambda)) \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) \left( \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda)^\top (I_m - A(\lambda)) \nabla \mathbf{F}(\widehat{\boldsymbol{\theta}}_\lambda) \right)^{-1} \\ &\text{and } A(\lambda) = A(A^\top A + m\lambda C)^{-1} A^\top. \end{aligned}$$

**Proof.** We further simplify the covariance formula in Theorem 6.2,

$$\mathcal{C} = H(\lambda)^{-1} \left( H(\lambda) - \begin{bmatrix} 0 & 0 \\ 0 & m\lambda C \end{bmatrix} \right) H(\lambda)^{-1} = H(\lambda)^{-1} \begin{bmatrix} \nabla \mathbf{F}^\top \\ A^\top \end{bmatrix} \begin{bmatrix} \nabla \mathbf{F} & A \end{bmatrix} H(\lambda)^{-1},$$

focusing only on  $\mathcal{C}^\theta$ , the upper-left  $p \times p$  block of  $\mathcal{C}$ , which gives the covariance information for the parameter of interest,  $\boldsymbol{\theta}$ . We use a block matrix inversion formula involving the Schur complement of the (2,2) block, and partition

$$H(\lambda)^{-1} = \begin{bmatrix} \nabla \mathbf{F}^\top \nabla \mathbf{F} & \nabla \mathbf{F}^\top A \\ A^\top \nabla \mathbf{F} & A^\top A + m\lambda C \end{bmatrix}^{-1} \quad \text{as} \quad \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^\top & \star \end{bmatrix},$$

where  $H_{11} = (\nabla \mathbf{F}^\top (I_m - A(\lambda)) \nabla \mathbf{F})^{-1}$ , and  $H_{12} = -H_{11} \nabla \mathbf{F}^\top A (A^\top A + m\lambda C)^{-1}$ . We see that

$$\begin{aligned} \mathcal{C}^\theta &= \mathcal{G}^\top \mathcal{G}, \quad \text{with } \mathcal{G} = \nabla \mathbf{F} H_{11} + A H_{12}^\top = (I_m - A(\lambda)) \nabla \mathbf{F} H_{11} \\ &= (I_m - A(\lambda)) \nabla \mathbf{F} \left( \nabla \mathbf{F}^\top (I_m - A(\lambda)) \nabla \mathbf{F} \right)^{-1}. \end{aligned}$$

<sup>9</sup>In a straightforward manner, we can design *confidence regions* (in the  $\mathbb{R}^p$  space) instead of individual confidence intervals for each parameter; see [111, Chapter 5] for more details.

(For simplicity, we left out the argument of  $\nabla\mathbf{F}$ , which in exact formulas should be  $\theta^*$ , but can be replaced by  $\hat{\theta}_\lambda$ , yielding the computable “hat” formulas.)

Since the estimator  $\hat{\theta}_\lambda$  is unbiased and normally distributed, we plug-in the approximate estimated variance (6.13) and we get that, approximately and for large  $m$ ,

$$\frac{(\hat{\theta}_\lambda)_i - \theta_i^*}{\sqrt{\hat{\sigma}^2 \hat{\mathcal{C}}_{ii}^\theta}} \sim t_{m-p_{\text{eff}}},$$

which gives the announced confidence interval.  $\square$

Taking a careful look at the specialized confidence intervals that we obtained, and observing in particular the formula of  $\hat{\mathcal{C}}^\theta$ , the covariance matrix of the estimated parameter, we note the following differences from the classical confidence intervals in nonlinear regression [111]:

- we need to use a bias-corrected estimate;
- the number of regression parameters  $p$  is replaced by the effective number of parameters  $p_{\text{eff}}$ , both in the variance formula (6.13) and in the degrees of freedom of the Student  $t$  distribution;
- if we ignore the ‘linear smoother matrix’  $A(\lambda)$  from the covariance matrix  $\hat{\mathcal{C}}^\theta$ , we get the classical inverse of the Fisher information matrix as covariance matrix for the estimate of  $\theta$ ; indeed:

$$\text{if } \hat{\mathcal{G}} = I_m \nabla\mathbf{F} (\nabla\mathbf{F}^\top I_m \nabla\mathbf{F})^{-1} \quad \text{then} \quad \hat{\mathcal{C}}^\theta = \hat{\mathcal{G}}^\top \hat{\mathcal{G}} = (\nabla\mathbf{F}^\top \nabla\mathbf{F})^{-1}.$$

## 6.4 Discussion on identifiability, redundancy and uniqueness

The *identifiability* concept involves proving or disproving that the parameters of the exact model can be exactly recovered from noiseless measurements. Some parameters might be *redundant*; this happens when the same (noisy) data can be explained by at least two different parameter values. Even for an identifiable model, ill-conditioning and noise can make some parameters non-uniquely determined. Other *uniqueness* issues are related to the optimization objective function (*e.g.*, nonlinear least squares) that might not be unimodal, thus several locally optimal solutions could be computed, depending on the starting point of the iterative optimization algorithm.

In linear regression, the uniqueness of the least squares solution, the identifiability and the parameter redundancy questions are all related to the rank deficiency (or nearly rank deficiency) of the coefficient matrix  $X$  of the linear model  $\mathbf{y} = X\theta + \varepsilon$ .

For nonlinear regression, one often translates the results of linear regression by using an estimate of the matrix  $\mathcal{A} = \left[ \frac{\partial F}{\partial \theta}(\mathbf{t}, \theta^*) \right]$  instead of the linear coefficient  $X$ . However, the asymptotic linearization is not always appropriate, thus the problems become much more involved. A whole detailed discussion on these issues for nonlinear regression is found in Chapter 3 of [111].

In this section, we focus on the combination of a nonlinear parametric part (which we assume identifiable and not prone to non-uniqueness and parameter redundancy problems) and the nonparametric part (the part that is actually linearly *parameterized* by splines).

In order to avoid identifiability problems that come from partly modeling the nonlinear  $F$  with splines, one can first think of a strategy of designing an appropriate spline space  $\mathcal{H}$  such that it does not interfere with the nonlinear function of the parametric part. If such an  $\mathcal{H}$  is possible to construct, then a pure semiparametric regression setting without any penalty term can be used. A sufficient identifiability condition would simply require that the given nonlinear function  $F(\cdot, \theta)$  is **not** in the space  $\mathcal{H}$  for any  $\theta$ . This restriction forbids the possibility that the output data  $\mathbf{y}^*$  can be interpolated in  $\mathcal{H}$ . Thus, the nonlinear  $F$  and the nonparametric part would be completely disentangled.

In practice, when the space  $\mathcal{H}$  is generated by splines, it can be quite rich. Even if the nonlinear  $F$  is not a member of  $\mathcal{H}$ , it might happen that (parts of)  $F$  can be very well approximated in  $\mathcal{H}$ . In this situation, we would like to ensure that the problem is still identifiable, at least for an appropriate penalty term and for an appropriate value of the regularization parameter  $\lambda$ .

The semiparametric regression problem is not identifiable if the parametric part doesn't have at least one *distinctive feature* that can separate it from what could be considered as background. Since we are focusing on the case when we require that the nonparametric part is fitted with a certain degree of smoothness, it means that our condition for identifiability requires that the nonlinear  $F$  should contain a certain non-smooth feature. This will definitely exclude the cases when  $F$  is a linear function or a low order polynomial.

The NMR spectroscopy data quantification problem (to be described in Chapter 7) is a good example for applying our semiparametric modeling method because in that context we have a parametric model containing specific Lorentzian peaks, which will not significantly interfere with the nonparametric part – a smooth baseline. Moreover, in this application, we have in general good convergence of the numerical optimization, because good starting values for the nonlinear parameters of the model are available.

Indicative information about the quality of the estimates can be obtained by looking at the estimated confidence intervals.

## 6.5 Numerical examples

In this section, we illustrate with a few examples the use of Algorithm 6.3, and of the statistical information extracted with the procedures of Sections 6.3.1 and 6.3.2.

First, we choose a simulation scenario that approaches in a simplified way the design of the NMR spectroscopy data quantification problem, as a prelude to the application in the next Chapter. Then, in subsection 6.5.2, we give other examples that illustrate the identifiability issues.

### 6.5.1 Description of simulation examples and results

Inspired by the shape of the frequency domain transformation of metabolite signals in the NMR experiment (*i.e.*, Lorentzian lineshapes), we consider as a simplification the rational function

$$F(t, (\theta_1^{(1)}, \dots, \theta_1^{(K)}, \theta_2^{(1)}, \dots, \theta_2^{(K)})) := \sum_{k=1}^K \frac{\theta_1^{(k)}}{(t - \theta_2^{(k)})^2 + 0.1},$$

where  $t$  represents a scalar abscissa in the interval  $\mathcal{I} = [-10, 10]$ . We use as true values for the parameter  $\theta^*$  the following data:

$$\theta_1^{(1,\dots,5)} = [5, 12, 8, 20, 15], \quad \theta_2^{(1,\dots,5)} = [-7, -4, -3, 1, 5].$$

Note that  $\theta_1^{(1,\dots,5)}$ , which appear linearly in the model function  $F$ , determine the amplitude of the Lorentzian shapes, while  $\theta_2^{(1,\dots,5)}$ , appearing in the denominator, determine the positions for each peak. (See middle plot in Figure 6.1.)

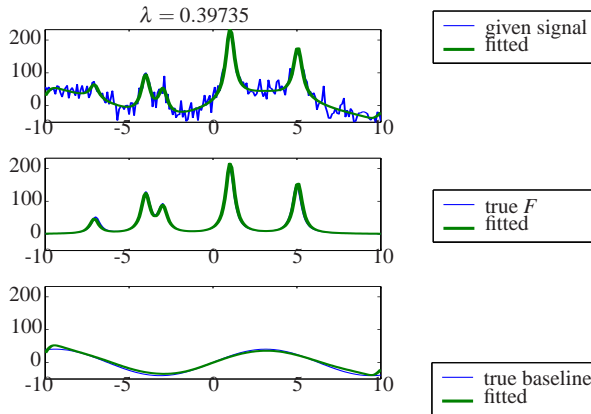
We consider the following simple baseline function:

$$g(t) := \sin(t/2).$$

Then we simulate  $m = 200$  measured outputs  $y_1, \dots, y_m$  at equidistant abscissas  $t_1, \dots, t_m$  in  $\mathcal{I}$ , by adding up the simulation function values (at fixed parameter  $\theta^*$ ), the values of the baseline  $g$ , and Gaussian noise with given variance  $\sigma^2$ , which represents a noise-to-signal ratio of approximately 30%, taking into account the magnitude of the considered simulated function.

For computing the nonparametric part, we use penalized splines [28]. Thus, the matrix  $A$  is formed from a B-spline basis [21] of degree 3; we set the size of  $A$  to  $m \times n = 200 \times 40$ . Moreover, instead of the regularization operator  $\mathcal{P}$ , we compute the matrix  $C$  as  $C = D^\top D$ , where  $D$  is the second order derivative matrix, *i.e.*, the  $(n-2) \times n$  tridiagonal Toeplitz matrix, with 2 on the main diagonal and  $-1$  on the first super- and subdiagonal.

Figure 6.1 shows the signal  $y$  obtained as described above, together with the fit returned by our Matlab implementation of Algorithm 6.3. As initial values for the parameters we choose random values with the constraint that  $\theta_1^{(1,\dots,5)}$  are positive and scaled to reflect the magnitude of the simulated signal, and  $\theta_2^{(1,\dots,5)}$ , appearing in the denominators, are taken from non-overlapping intervals in  $[-10, 10]$  that contain the true values.



**Figure 6.1.** Simulated signal  $y$ : the bigger peaks come from the function of interest  $F$ , the baseline gives a smooth trend, and the Gaussian noise has a noise-to-signal ratio of 30%. With a GCV choice for  $\lambda$  and with reasonably good initial values for the nonlinear parameters of  $F$ , we obtain excellent fit of the model and of the baseline.

We performed a Monte Carlo simulation study to assess statistical information on the converged estimates given by Algorithm 6.3. Table 6.1 presents averaged results after 100 simulations for several noise levels. The averaged relative errors for the regression

**Table 6.1.** Monte Carlo simulation results for the example problem. The noise-to-signal ratio ( $N/S$ ) is varied from 10% to 50%, while the problem dimensions  $m = 200$ ,  $n = 40$  and  $p = 10$  remain constant. Averaged values of the true and estimated noise variances, and averaged relative errors in the regression parameter  $\theta$  (i.e.,  $\|\hat{\theta}_\lambda - \theta^*\|_2 / \|\theta^*\|_2$ ), in the function fit (i.e.,  $\|\mathbf{F}(\hat{\theta}_\lambda) - \mathbf{F}(\theta^*)\|_2 / \|\mathbf{F}(\theta^*)\|_2$ ), and, respectively, in the fit of the baseline (i.e.,  $\|A\hat{\mathbf{a}}_\lambda - \mathbf{g}^*\|_2 / \|\mathbf{g}^*\|_2$ ) are presented.

N/S	true variance	variance estimate	error in $\theta$	error in $\mathbf{F}$	error in baseline
10 %	32.61	33.18	0.024	0.028	0.052
20 %	131.4	133.1	0.044	0.052	0.095
30 %	292.2	296.1	0.060	0.074	0.123
40 %	530.5	526.5	0.065	0.086	0.151
50 %	826.5	823.2	0.071	0.098	0.176

parameter  $\theta$ , as well as for the reconstructed nonlinear function  $\mathbf{F}(\theta)$  and baseline  $A\mathbf{a}$  are reported in the last three columns of Table 6.1. These small errors confirm that the regularized nonlinear least squares algorithm, with the GCV choice for the smoothing parameter  $\lambda$ , performs well. Moreover, in the left part of Table 6.1, next to the noise-to-signal ratio, we have the corresponding noise variance used in the simulations, as well as the estimated noise variance using formula (6.13). The values agree well in all experiments, thus confirming the validity of formula (6.13) in practice.

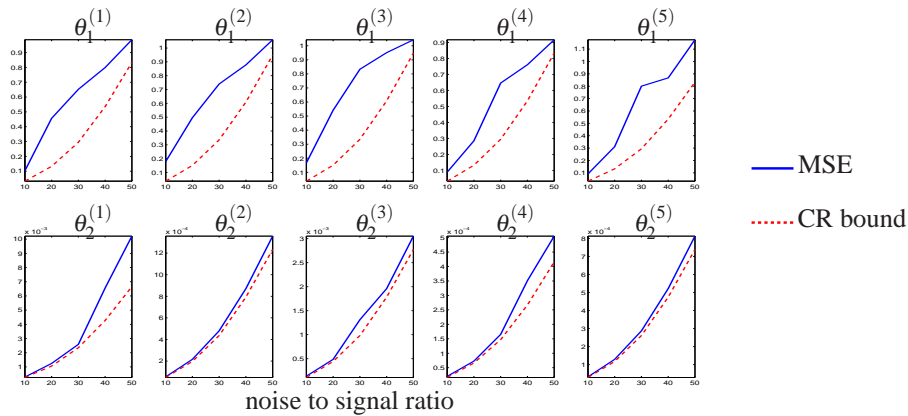
An illustration of the estimation results for each individual parameter in  $\theta$  is shown in Figure 6.2. The mean squared errors obtained in the Monte Carlo study are plotted together with the Cramer-Rao lower bounds computed from the Fisher information matrix corresponding to the true parameter values of the nonlinear function  $F$ . Note that these Cramer-Rao bounds ignore the incorporation of a possible baseline in the model. As a result, we see that the mean squared error for the estimate of the linear parameters  $\theta_1^{(1,\dots,5)}$  are further away from the Cramer-Rao bounds than it is the case for the nonlinear parameters  $\theta_2^{(1,\dots,5)}$ . This means that the latter group of parameters is less affected by the nuisance part of the semiparametric model.

### 6.5.2 Comparison between classical confidence intervals and specialized confidence intervals

At the end of Section 6.3.2 we observed that the confidence intervals taking into account the presence of the baseline term (see Corollary 6.4) are a generalization of the classical confidence intervals from nonlinear regression. Here we compare experimentally the new specialized confidence intervals with the classical ones.

With the same example setting as above, we use a noise-to-signal ratio of 30% and generate 100 random simulations (i.e., 100 noise realizations); we set the required confidence level to 95% and we count the number of successful outcomes for the condition: “ $\theta_i^*$  is inside the computed confidence interval”, for every individual parameter in  $\theta^*$ .

Table 6.2 gives the percentages of successes for both the classical confidence intervals and the new specialized confidence intervals. Note that the specialized method gives



**Figure 6.2.** Each plot corresponds to an individual variable in  $\theta$ ; the horizontal axis corresponds to several noise levels, and the vertical axis corresponds to squared errors in each parameter. In every plot, the full (blue) line shows the mean squared errors of the estimated parameter to its true value, and the interrupted (red) line is the Cramer-Rao lower bound.

**Table 6.2.** Percentages of confidence intervals that contain the true parameter value.

Method	$\theta_1^{(1)}$	$\theta_1^{(2)}$	$\theta_1^{(3)}$	$\theta_1^{(4)}$	$\theta_1^{(5)}$	$\theta_2^{(1)}$	$\theta_2^{(2)}$	$\theta_2^{(3)}$	$\theta_2^{(4)}$	$\theta_2^{(5)}$
Classical	79%	77%	75%	76%	80%	91%	96%	91%	93%	93%
Specialized	97%	94%	93%	99%	97%	94%	99%	96%	97%	96%

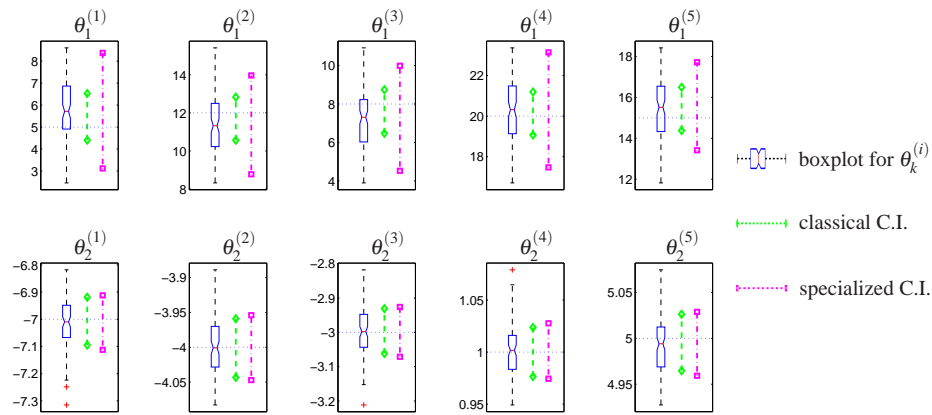
percentages closer to the required confidence level of 95%, while the classical method, which ignores some aspects of the problem, can give as low as 75% successful counts, for the same required level of 95%. This means that in up to 20% of the outcomes, the true value  $\theta_i^*$  is inside the specialized confidence interval, but outside the classical one.

The new confidence intervals are a bit wider than the classical ones. In Figure 6.3 we plot the *averaged* confidence intervals in the 100 simulations next to the boxplots of the computed  $\theta$  parameters, for the noise level of 30%. We note that the length difference is just between 5 to 50%, and thus the specialized confidence intervals are still tight enough for giving useful information on the estimated parameters.

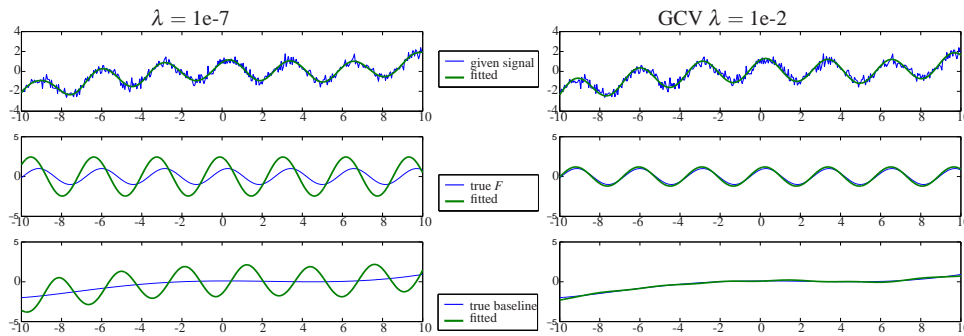
### 6.5.3 Illustration of identifiability problems

In this subsection we show the extents of the identifiability problem for our semiparametric model. First, we give an example where the parametric and nonparametric parts are not strongly disentangled, but where the baseline component is however much smoother than the parametric  $F$ .

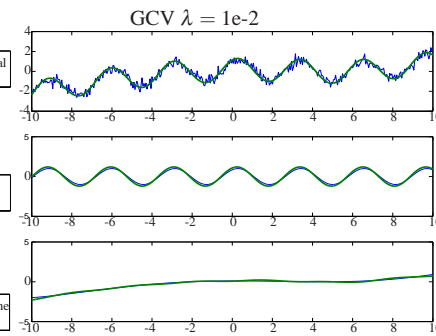
Let  $F$  be a sine function in  $t$ , and let the unknown nonlinear parameter  $\theta$  give the amplitude and the phase of this sinusoid. We construct a true baseline as a polynomial of degree 5. We generate data using a discretization with 400 points in the interval  $[-10,10]$



**Figure 6.3.** Boxplots for the  $\theta$  parameters in 100 simulations; averaged confidence intervals, where the shorter are the classical and the wider are the specialized confidence intervals. Horizontal dotted line marks the true value of the parameters.



**Figure 6.4.** With a too small value of the smoothing parameter  $\lambda$ , the spline reconstruction models also part of the sine model function.

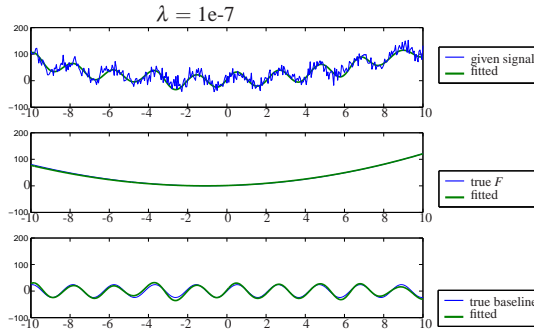


**Figure 6.5.** With the value of  $\lambda$  computed using GCV, the spline reconstruction doesn't interfere with the sine model function.

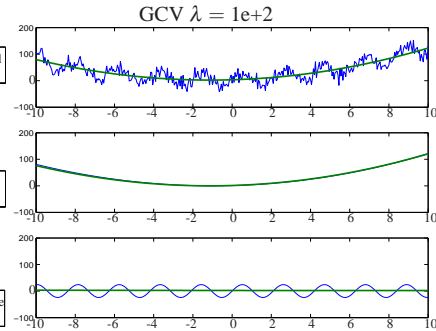
and we add Gaussian noise corresponding to a 15% noise-to-signal ratio. We use a dense space of B-splines and a penalty of the second order derivative type.

Figure 6.4 shows that when we give a very small value to the smoothing parameter, the spline baseline tends to reconstruct part of the sine behaviour, and thus the amplitude and phase parameters of the modeled sinusoid are clearly biased from the exact values. However, as we see in Figure 6.5, the choice made by GCV gives excellent results. The explanation lies in the fact that in this case GCV favours a more parsimonious model for the baseline.

We performed also an “opposite” experiment where we considered as parametric part a simple polynomial function and as baseline a sine function. The GCV choice in



**Figure 6.6.** A small value of the smoothing parameter  $\lambda$  favors the spline reconstruction of the sinusoidal baseline.



**Figure 6.7.** With the value of  $\lambda$  computed using GCV, the spline reconstruction is too smooth.

this unidentifiable problem didn't give a good result; although the parametric part was reasonably recovered, the baseline was over-smoothed, as if the sine waves were treated as noise and smoothed out. If we manually selected a very small  $\lambda$  value, close to zero, then the smoothness constraint was ignored and a good fit of both model and baseline was obtained. (See Figures 6.6 and 6.7.)

## 6.6 Conclusions

In this chapter, we have presented a semiparametric problem formulation, its statistical properties, its computational solution, and simulation examples. We reserved the room for explaining a real life application that in fact motivated this semiparametric modeling framework in Chapter 7.

The main points of the study in this chapter reveal that it is relatively easy to include a nonparametric part into a nonlinear regression problem, in the case when the nonparametric part can be restricted through a weighted norm (e.g., by imposing smoothness of the baseline term). The nonparametric part can then be modeled by using a spline basis, and thus only a linear term is actually added in the nonlinear least squares optimization criterion.

However, special care should be taken when statistical information is retrieved from a regularized semiparametric regression problem, since in general we obtain biased estimates (but the bias can be corrected), and the classical nonlinear regression confidence bounds that use the Fisher information matrix must be adapted to take into account the regularization term.

The simulations from this chapter (and the application in the next chapter) use penalized splines for modeling the nonparametric part. Future work involves using in applications other types of splines from the template splines family. For example, *regression splines* [36] could be used in the semiparametric modeling context and the number and position of the knots should then be optimized instead of the regularization parameter  $\lambda$ .

## Chapter 7

# Application to MRS data quantification and its software implementation in AQSES

In this chapter, we pursue the problem of quantifying metabolite concentrations from short-echo time *in vivo* Magnetic Resonance Spectroscopic (MRS) measurements. The goal of this application is to compute the parameters of a certain model function, which give information about the concentrations of chemical substances in a region of the brain. Nowadays, the nuclear magnetic resonance scanners record *short echo-time* signals that are richer and richer in information because they display the responses of significantly more metabolites (chemical substances). This means that not only the most relevant metabolites in the brain, but also a *macromolecular baseline* for which no model function is available must be taken into account in automated quantification methods. For this reason, the semiparametric modeling framework developed in the previous chapter will be employed.

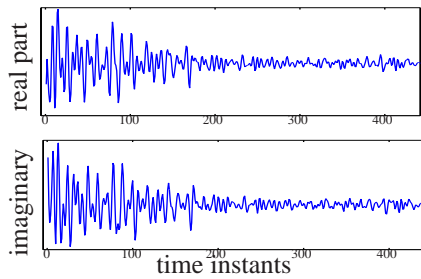
## 7.1 Introduction

Magnetic Resonance Spectroscopy (MRS) is a non-invasive technique that is used in a wide range of medical applications, *e.g.*, for brain tumors diagnosis [38, 63]. Measured MRS proton spectra from the human brain can provide essential information about the chemical substances present in each specified voxel (small volume) of the brain.

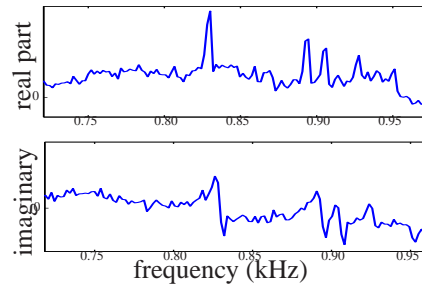
### 7.1.1 MRS data quantification with unknown macromolecular baseline

The nuclear magnetic resonance spectrometer outputs complex-valued time-domain signals that have a decaying pattern. Figure 7.1 shows a typical preprocessed time-domain signal obtained *in vivo* from a healthy selected small volume in a human brain. For visual interpretation, the Fourier transformed signal is usually plotted in the frequency domain (see Figure 7.2), because the position and magnitude of each peak gives information about the chemicals present in the sample.

Each pure metabolite (chemical substance) has a peculiar time response that depends on the number and position of hydrogen protons in the molecular structure; in theory, the time response is a sum of complex damped exponentials, which yield the typical Lorentzian

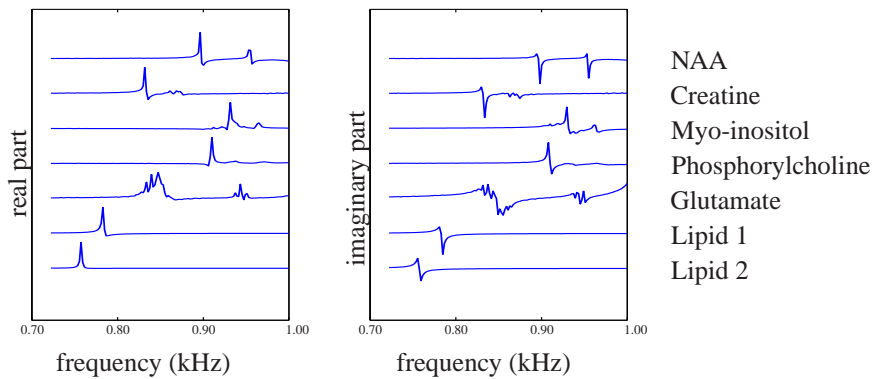


**Figure 7.1.** The time domain MRS signal measured from a selected volume in the human brain.



**Figure 7.2.** The frequency domain spectrum measured from a selected volume in the human brain.

peaks in the frequency domain signals. Spectra of metabolites that are known to be present in the human brain can also be measured *in vitro*. Such measurements can be grouped together in a database of metabolite signals, see Figure 7.3. An *in vivo* signal can be modeled



**Figure 7.3.** The frequency domain spectrum profiles for several typical metabolites in the human brain.

as a combination of metabolites in the database. In this way, the weighting coefficients (amplitudes) in the combination yield the *concentrations* of the metabolites. Estimating these concentrations is the main goal of the MR spectroscopy data quantification problem. However, this combination cannot simply be a linear combination; one should allow small corrections in other spectral parameters (frequency shifts, damping corrections, phase shifts, etc), since these parameters may slightly vary from measurement to measurement [101].

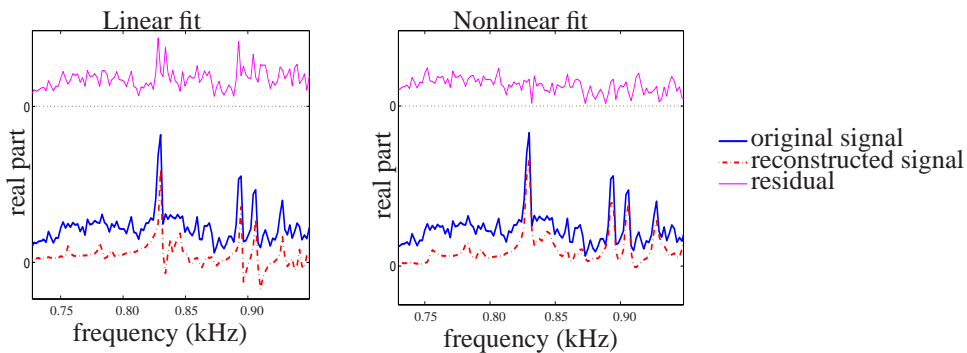
Denote by  $\{v_k, \text{ for } k = 1, \dots, K\}$  the  $K$  given complex-valued time series of length  $T$ , representing *in vitro* measured metabolites, and by  $w$  the *in vivo* measured MRS signal.

The model that allows spectral corrections in the combination of metabolites is:

$$w(t) = \hat{w}(t) + \varepsilon_t := \sum_{k=1}^K \alpha_k (\zeta_k)^t v_k(t) + \varepsilon_t, \quad t = 0, \dots, T-1, \quad (7.1)$$

where  $\alpha_k, \zeta_k \in \mathbb{C}$  are the model parameters (their meaning will be described in Section 7.2, where a more general model formulation will also be allowed).  $\varepsilon_t$  is an unknown noise perturbation with zero mean, for all  $t$ 's from 0 to  $T-1$ .

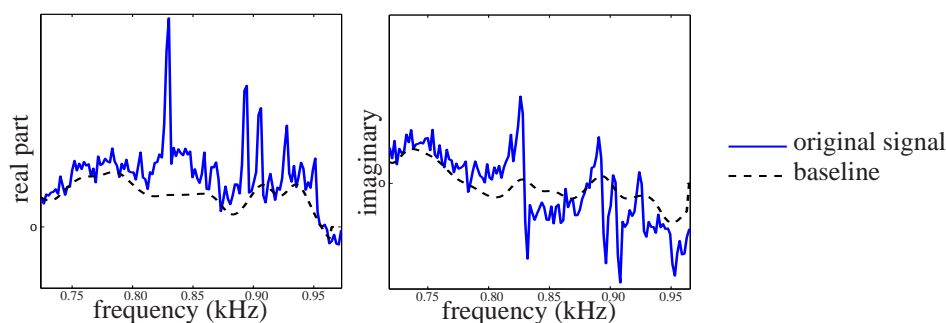
Figure 7.4 shows two fits: one is the best linear combination of the signals from the metabolite database, and the second is the optimal fit with the nonlinear model (7.1) that uses spectral corrections, of the same *in vivo* signal as plotted before. (In practice, we might have up to 25 metabolites in the database, but here, for illustration, we consider only the 7 metabolites given in Figure 7.3.) It is noticeable that the residuals are highly biased from the zero line, in both fits. This happens because the pure metabolites frequency-domain profiles are zero outside the range of resonant frequencies, but this is different in what concerns an *in vivo* signal. The database of metabolite profiles only contains the most prominent and



**Figure 7.4.** Real part of the original frequency domain spectrum from an *in vivo* signal, together with the optimal reconstructed spectrum using a combination of metabolites from the database. Left: linear combination. Right: linear combination with nonlinear spectral corrections.

relevant metabolites, but the *in vivo* signal also holds information that originates from some unknown ‘macromolecules’. This nuisance part is known as the *baseline*. The baseline is a highly damped component; seen in the frequency domain, it is a smooth, broadband, low amplitude spectrum, that gives an underlying trend to the *in vivo* signal. (See Figure 7.5.) Its shape can vary and it is in general unpredictable, especially in pathological cases. In mathematical terms, we can say that the baseline is characterized by the fact that its Fourier transformation should be a smooth function<sup>10</sup>. For this reason, a good choice is found in identifying it as a smooth curve using splines in the frequency domain. We denote the

<sup>10</sup>There is an abuse of terms here: actually, the Fourier transform of the unknown *continuous-time* baseline should be a smooth function. In practice, we work with discrete time instants and with the discrete Fourier transform.



**Figure 7.5.** The original frequency domain spectrum from an in vivo signal, together with a baseline constructed with splines.

baseline with  $b(\cdot)$  and the model becomes

$$w(t) = \hat{w}(t) + \varepsilon_t := \sum_{k=1}^K \alpha_k (\zeta_k)^t v_k(t) + b(t) + \varepsilon_t, \quad t = 0, \dots, T-1.$$

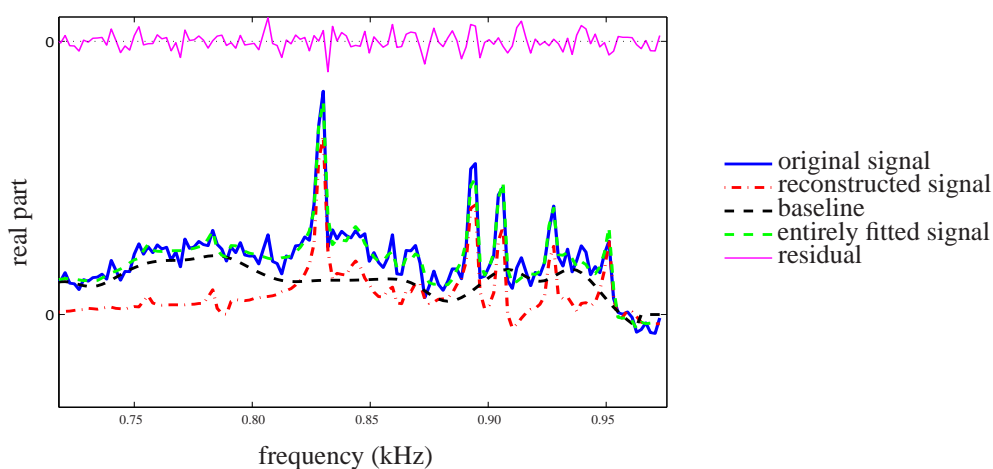
We fit the exemplified data using this semiparametric model and we apply the method described in Section 6.2. The result is depicted in Figure 7.6 (only the real part, for more clarity). Observe that the bias that was visible in the residual of Figure 7.4 is now removed.

Introducing the baseline term into the model proves to be an adequate solution for fitting the given data. But does this also imply that a more reliable estimation of the parameters of interest (and, finally, of the metabolite concentrations) can be obtained? Under the assumption that the reconstructed baseline doesn't model the part of the data that is characteristic for the metabolites of interest, the answer is yes. An important problem is deciding how many features of the data could be allowed to be fitted with the baseline. The trade-off between the parametric part of the model and the baseline can be viewed, in this application, as a trade-off in the smoothness of the frequency domain baseline.

## 7.1.2 Software for MRS quantification

In Section 7.3 we shall discuss the AQSES software module for quantification of short-echo time MRS signals. The mathematics behind AQSES are based on the semiparametric modeling theory presented in Chapter 6. Particular details about designing a semiparametric model for MRS quantification with unknown macromolecular baseline are presented in Section 7.2. Moreover, we devote Chapter 8 to computational aspects of the AQSES implementation and comparisons between several of its variants.

Other important contributions in the field of fitting short-echo time MRS signals include [97] and [100]. The differences and similarities between the technique implemented in AQSES and the procedure used in LCMoel [97] are:



**Figure 7.6.** Real part of the original frequency domain spectrum from an in vivo signal, together with the optimal reconstructed spectrum using the metabolites from the database, the optimal baseline constructed with penalized splines and the entirely reconstructed signal (metabolites plus baseline).

- AQSES performs the fitting in the time-domain, which is the data acquisition domain; LCMoel fits the real part of the frequency domain signals;
- in both methods, the baseline is nonparametrically modeled using penalized splines in the frequency domain; the fact that AQSES fits in the time-domain is not a problem, because a smoothing criterion that involves an inverse Fourier transformed spline basis can be used;
- the criterion for choosing the amount of smoothing via the regularization parameter in AQSES is based on nonlinear model selection theory, while the method in [97] seems more heuristic in nature.

Comparing AQSES to the recent contribution in [100] (QUEST), we highlight the following differences:

- a nonparametric baseline is recovered in QUEST using heuristic methods, where several steps are involved: truncation, partial fitting, subtraction, and final fitting. Its performance is sensitive to the choice of the number of truncated data points and the model order for the baseline fit. The algorithm in AQSES uses only one common optimization problem for the fitting of both the model and the baseline. It is thus less prone to accumulated errors.
- In reference [100], an augmented Fisher information matrix (inspired by [120]) is used. However, it is not clear how to choose the value of the number of effective parameters, involved in the computation of confidence bounds. In this respect, the

discussion in Chapter 6, Section 6.3.2 (see also [115]) clarifies the way the confidence bounds can be automatically estimated for the procedure in AQSES.

## 7.2 Mathematical formulation

For the quantification of short echo-time MRS signals, we assume that we are given a “metabolite database”, which is a set  $\{v_k, \text{ for } k = 1, \dots, K\}$  of complex-valued time series of length  $m$ , representing *in vitro* measured MRS responses [38]. An *in vivo* measured MRS signal  $y$  is also a complex-valued time series of length  $m$  that will satisfy the model

$$y(t) = \hat{y}(t) + \varepsilon_t := \sum_{k=1}^K \alpha_k (\zeta_k)^t (\eta_k)^{t^2} v_k(t) + b(t) + \varepsilon_t, \quad t = t_0, \dots, t_{m-1}, \quad (7.2)$$

where  $\alpha_k, \zeta_k, \eta_k \in \mathbb{C}$  are unknown parameters that account for concentrations of the metabolites in the database and for the necessary corrections of the database signals, due to inherent differences in the data acquisition techniques [97, 101, 79]. In fact the complex amplitudes  $\alpha_k$  and the complex  $\zeta_k$  and  $\eta_k$  can be written in terms of a parametrization with real-valued variables as (with  $j = \sqrt{-1}$ ):

$$\begin{aligned} \alpha_k &= a_k \exp(j\phi_k), \\ \zeta_k &= \begin{cases} \exp(-d_k + jf_k), & \text{for Lorentzian and Voigt lineshapes;} \\ \exp(jf_k), & \text{for Gaussian lineshapes;} \end{cases} \\ \eta_k &= \begin{cases} \exp(je_k), & \text{for Lorentzian lineshapes;} \\ \exp(-g_k + je_k), & \text{for Gaussian and Voigt lineshapes;} \end{cases} \end{aligned}$$

where  $a_k$  are the real amplitudes,  $\phi_k$  are the phase shifts,  $d_k$  are damping corrections,  $g_k$  are Gaussian damping corrections,  $f_k$  are frequency shifts, and  $e_k$  are Eddy current correction terms [64, 86].

Moreover,  $b(t)$  represents the chemical part that is not modeled, which is the response of the substances that are not included in the database, and  $\varepsilon_t$  is an unknown noise perturbation with zero mean, for all  $t$ 's with indices from 0 to  $m - 1$ .

The identification of complex amplitudes  $\alpha_k$ , and complex  $\zeta_k$ 's and  $\eta_k$ 's, for  $k = 1, \dots, K$ , can be accomplished by minimizing the least squares criterion:  $\sum_{t=t_0, \dots, t_{m-1}} |y(t) - \hat{y}(t)|^2$ .

### 7.2.1 A semiparametric model for MRS signals

A semiparametric model for MRS quantification can be designed, which takes into account the parametric part of the model, but treats nonparametrically the baseline, with its constraint on smoothness [109, 97, 100, 101, 30, 115]. For the nonparametric reconstruction of the baseline, we construct a basis of splines [21, 28] and put the discretized splines as columns in a matrix  $A$  of size  $m \times n$ , with  $n$  smaller than the number of data points  $m$ . Any nonlinear function can be approximated as a linear combination of spline functions. The coefficients in this linear combination are the unknowns that must be identified. We denote these linear coefficients by  $c_1, \dots, c_n$  (or by  $\mathbf{c} \in \mathbb{C}^n$ , when stacked in a column vector).

Thus the discretization of a nonlinear function approximated with splines can be written in matrix notation as  $\mathbf{Ac}$ .

A regularization operator  $D$  (that gives a matrix  $B = D^T D$ ) is defined to measure the smoothness of the baseline in the frequency domain. We can take  $D$  as the discrete second-order differential operator,  $D = \begin{bmatrix} -1 & 2 & -1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & -1 & 2 & -1 \end{bmatrix}$ , first-order differential operator,  $D = \begin{bmatrix} -1 & 1 & & 0 \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix}$ , zero-order differential operator, the identity  $D = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$ , or some combination.

Since the goal is to reconstruct a smooth baseline in the frequency domain, while still fitting in the time-domain, we transform the basis matrix  $A$  to the time domain, with the discrete inverse Fourier transform. Thus  $\mathcal{A} := \mathcal{F}^{-1}(A)$ , where the operator  $\mathcal{F}^{-1}$  denotes the discrete inverse Fourier transform, applied on each column of a matrix. Finally, we consider the regularized nonlinear least squares criterion

$$\min_{\substack{\alpha_1, \dots, \alpha_K \in \mathbb{C}, \mathbf{c} \in \mathbb{C}^n \\ (\zeta_1, \dots, \zeta_K, \eta_1, \dots, \eta_K) \in \Omega}} \frac{1}{m} \sum_{t=0}^{t_{m-1}} \left| y(t) - \sum_{k=1}^K \alpha_k (\zeta_k)^t (\eta_k)^{t^2} v_k(t) - (\mathcal{A}\mathbf{c})(t) \right|^2 + \lambda \mathbf{c}^H \mathbf{B} \mathbf{c}, \quad (7.3)$$

where  $\Omega$  denotes some constrained parameter space.  $\Omega$  usually involves only linear equality and inequality constraints on the real-valued parameters appearing in  $\zeta_k, \eta_k$ . The role of possible equality constraints is to impose *prior knowledge* relationships between corresponding parameters of related metabolites. The inequality constraints are, in fact, simple bound constraints on the real-valued parameters. We expect that these corrections will be small, since all these parameters are used to *slightly correct* the metabolite signals. For instance, it is expected that the damping and frequency shifts are bounded within a small interval around zero, since otherwise some metabolite patterns can become interchangeable and the chemical meaning of the corrected metabolite profiles can be lost.

In (7.3),  $\lambda$  is a fixed regularization (penalty) parameter, and the whole term  $\lambda \mathbf{c}^H \mathbf{B} \mathbf{c}$  is responsible for ensuring a certain degree of smoothness to the baseline  $b$ . The value that we give to  $\lambda$  controls also the degree of smoothness.

Note that in (7.3) the complex amplitudes  $\alpha_k$  are free variables. This translates into the fact that, when we look at their representation in terms of the real amplitudes and the spectral phases,  $\alpha_k = a_k \exp(j\phi_k)$ , the phases  $\phi_1, \dots, \phi_K$  are free between  $-\pi$  and  $\pi$ , and the amplitudes  $a_1, \dots, a_K$  have *positive* values. In fact, the real amplitudes are the most representative parameters for the MRS model, since they are the weights with which each metabolite appears into the quantified signal; they yield, thus, the metabolite concentrations in the given brain region, and these concentrations are indicative for the health status or tumor degree in that region [92, 97].

### 7.2.2 Using a filter

A filter can be used to remove irrelevant information from the *in vivo* MRS signal. For instance, a pass-band FIR filter can be used to select only the frequency region of interest. The FIR filter in AQSES is taken from the implementation of [121]; it is a maximum phase

FIR filter that is automatically optimized in order to remove the water component from an MRS *in vivo* signal.

Applying a FIR filter to a vector (a discrete signal) involves a convolution operation. This is however a simple and fast computation. When such an operation is applied to the measured signal, it must also be taken into account by the fitting model. In other words, a filtered measured signal will be fitted with a filtered model plus a filtered baseline.

The design of the filter is performed outside the actual fitting method of AQSES. Such a filter consists of a vector of coefficients; the length of the filter and the coefficients are optimized during the automatic filter design [121]. Let  $h_1, \dots, h_p$  denote the filter coefficients. Then a discrete-time signal  $v$ , *i.e.*, a vector of length  $m$ , can be convolved with the filter giving the filtered signal  $w$  (of length  $m - p + 1$ ), according to

$$w_i = \sum_{l=1}^p h_l v_{i+l-1}, \quad i = 1, \dots, m - p + 1.$$

A FIR filter commutes with the sum, but this doesn't mean that it commutes also with the *modified* sum of metabolites, where there are shifts and corrections to the spectral parameters; thus we cannot apply the same methodology as for unfiltered signals, only by using a filtered database. In AQSES minimization, we replace the original signal, the reconstructed signal and the reconstructed baseline with their filtered versions. The changes that are involved in the function and Jacobian evaluation are only the following three:

- the *in vivo* signal  $\mathbf{y}$  is replaced throughout with its filtered version;
- each corrected metabolite signal (having the elements  $\alpha_k (\zeta_k)^{t_i} (\eta_k)^{t_i^2} v_k(t_i)$ , for  $i = 0$  to  $m - 1$ ) is replaced with its filtered version whenever a new function/Jacobian evaluation is required;
- the spline matrix  $\mathcal{A}$  is replaced throughout with its filtered version (each column is thus separately filtered).

## 7.3 The software package AQSES

### 7.3.1 The AQSES GUI framework

AQSES GUI is an open source Java-based graphical user interface (GUI) for processing and displaying short-echo-time magnetic resonance spectroscopy signals [23]. Among the tasks that AQSES GUI can perform, we enumerate:

- loading MRS signals from a multitude of file formats;
- visualizing MRS signals (time-domain or frequency-domain) in a 3D environment;
- preprocessing signals using phase or frequency corrections, filtering (HLSVD-PRO [73]), and others;
- creating metabolite databases by grouping signals together and, possibly, applying preprocessing steps;

- processing signals using the AQSES method (accurate quantification of short echo-time MRS signals);
- displaying *metabolic images* obtained from quantifying MRS signals coming from a 2D grid of voxels from a slice of the human brain (see Figure 7.11 on page 133).

Figure 7.7 shows the main window of AQSES GUI where a project with many spectroscopic signals is open and the options of the AQSES quantification method can be set.

More details about the various capabilities of AQSES GUI can be found in [119] and in the user's manual of AQSES GUI.

### 7.3.2 Implementation details

#### Programming language and dependencies

AQSES is implemented in FORTRAN 77. The main optimization part is carried out using an extension of the Levenberg-Marquardt algorithm, which accepts linear bounds constraints, in the DN2GB implementation written by David M. Gay [41]. DN2GB is an extension of the NL2SOL package of the same author [25], which deals with nonlinear least squares minimization for real data. AQSES is performing nonlinear least squares with a complex-valued function, *i.e.*, a complex norm

$$\|f(\theta)\|^2 = \|\Re f(\theta)\|_2^2 + \|\Im f(\theta)\|_2^2$$

is minimized, where  $f(\theta) = \Re f(\theta) + i\Im f(\theta) \in \mathbb{C}^m$ . Thus, instead of solving an optimization problem with complex data, it is possible to transform all computations to real by doubling the problem dimension.

We have chosen for a reimplement of the Variable Projection method [47, 48], taking the guidelines from the VARPRO implementation written in FORTRAN by John Bolstad [127]. Our new code for nonlinear least squares with variable projection embedded in AQSES works directly with complex function computations. That is, function and Jacobian evaluations are performed in the complex domain. The evaluated values are transformed to real in order to apply the optimization module DN2GB. Note that in this way we are able to impose linear bounds on the nonlinear parameters involved in the minimization, a fact not implemented by the classical VARPRO code. Details about the VARPRO implementation are presented in the next chapter.

AQSES uses several routines from BLAS and LAPACK [2] for some basic linear algebra computation with (double) real as well as (double) complex data. A few routines from SLICOT [8] and from the FFTW package [37] are also used.

#### Multiple round fitting

AQSES is designed with an option of multiple round minimization: fitting a new signal with the signals from a database can be done in one or several steps. In each *round*, a database larger than in the previous round (a superset of the previous one) is used. The goal of this multiple round procedure is to have good estimates of (some of) the variables as starting values for the nonlinear optimization.



**Figure 7.7.** Main frame of AQSES GUI. A project with many MRS signals is loaded. The settings of the AQSES quantification method can be modified in the bottom part of the view.

Although this heuristic approach was quite helpful in previous implementations of AQSES (see the results in [96, Chapter 4]) that didn't distinguish between linear and nonlinear types of variables in the model (and, thus, good starting values are needed for the amplitudes and phases of each metabolite), it became obsolete in the latest versions, where the VARPRO method was used. In this case, a single round of fitting, with zero starting values for all nonlinear parameters, leads usually to the same convergence behavior as multiple round fitting. A notable exception is the need for a good starting value for the common phase correction (when equal phases for all metabolites are imposed, see Chapter 8). In this case, the common phase becomes a nonlinear parameter whose initial value in the optimization algorithm is set using a preliminary round of fitting with the non-equal phases option. More precisely, the initial value for the phase is taken as the converged value of the phase of the metabolite with highest concentration in the preliminary round.

### Testing the validity of databases

Two conditions are provided in the code in order to test the numerical validity of the database and to warn if numerical sensitivity of the database might influence the correctness of the fitting results. Both tests act in the same way, but the first one is done for the original database, while the second is done for the filtered database, in case a filter is used.

Numerical problems during the nonlinear least squares fitting procedure might occur in the case when there exist nearly linearly dependent columns in the (filtered) database. In this situation, the Jacobian computed for each step of the Levenberg-Marquardt minimization algorithm gets nearly rank deficient and causes computational problems.<sup>11</sup>

### Automatic baseline tuning

The generalized cross validation criterion for semiparametric models (see Chapter 6 for the definition formula (6.7) on page 103 and for the computational remarks in §6.2.6) is implemented as a method for choosing a good degree of smoothness for the macromolecular baseline. The  $\lambda$  factor that multiplies the penalty term in (7.3) is tuned in such a way that the generalized cross validation criterion is (approximately) minimized. A large value for  $\lambda$  implies a smooth frequency-domain baseline, while a small  $\lambda$  allows rougher baselines. When the automatic method is used, several fitting problems for several values of  $\lambda$  are solved, until a penalty parameter  $\lambda$  that leads to a good enough trade-off between the fitting quality and the smoothness of the baseline is found.

### Error measures for the quantification

Approximate Cramer-Rao bounds are also computed at the end of the quantification procedure. These error bounds are using the confidence intervals definition that was specially adapted for semiparametric nonlinear regression in Chapter 6, Section 6.3.2 (or [115]). The bounds correspond to all the spectral parameters for the metabolites of interest (linear and

---

<sup>11</sup>It is true that Levenberg-Marquardt has guards against rank-deficient Jacobian, but this typical rank-deficiency is usually caused in nonlinear least squares minimization by nearly compatible nonlinear equation fitting. The case of linearly dependent columns in the database is a degenerate situation yielding rank-deficient Jacobian, and should be avoided.

nonlinear parameters, as well). They give an indication about the uncertainty of the final quantified parameters. If the given bounds are small enough relative to the corresponding parameter value, then it means that the computed value is reliable. If a large bound is found for a certain component, then the computed parameters might be unreliable. This might be due to a poor signal to noise ratio, or to an incomplete database of metabolites.

## 7.4 Numerical results

Experiments using simulation signals, as well as real *in vitro* and *in vivo* signals have been designed to test the AQSES software package. These experiments are presented in detail in the paper [119]. In this section, we present an overview of these results, focusing on the simulations, because these can be easily interpreted in a parameter estimation context, while the results on real data require a deeper biomedical and chemical background that falls outside the scope of this thesis.

### 7.4.1 Simulated signals

A large number of simulated signals were created for the purpose of testing the robustness and accuracy of the optimization methods in AQSES. Each simulated signal consisted of a linear combination of 8 spectrally perturbed metabolite profiles in the basis set (containing Myo-inositol (Myo), Creatine (Cr), Phosphorylcholine (Pch), Glutamate (Glu), NAA, Lactate (lac), and two lipid signals (Lip1 and Lip2)).

In a **first** data set of 200 signals, no nuisance such as baseline, noise or water peaks, was added to the simulated signals. The simulated parameters for amplitude, damping, phase and frequency shifts for each simulated signal were randomly chosen, with values in meaningful intervals. This means that the perturbations on dampings and frequencies were small enough such that the characteristics of each metabolite signal were not lost.

For each simulated signal, the true amplitudes,  $a_k$ , were compared to the ones estimated by AQSES (using the same 8 metabolite profiles in the basis set) by means of a performance measure defined as

$$PM_k = 100 \sqrt{\frac{\sum_{l=1}^{200} (a_{k,l}^{\text{estimated}} - a_{k,l}^{\text{true}})^2}{\sum_{l=1}^{200} (a_{k,l}^{\text{true}})^2}}, \quad (7.4)$$

where  $a_{k,l}^{\text{estimated}}$  (resp.,  $a_{k,l}^{\text{true}}$ ) is the estimated amplitude (resp., the simulated amplitude) for metabolite  $k$  in the simulation number  $l$ . A low value of the performance measure reflects a high accuracy, since it is a percentage measure of the difference between estimated and true amplitudes.

The **second** experiment extends the results of the first one for larger databases, *i.e.*, with more metabolite profiles. We took the same simulated spectra (set 1) as in the first experiment, but three more metabolite profiles were added to the basis set: Taurine (Tau), Alanine (Ala) and Glucose (Glc). We chose these metabolites because they are known to be important metabolites that have no strong correlation with the metabolites that were already inside the basis set.

The **third** experiment shows the influence of water, baseline and noise on the estimated amplitudes. We used the same basis set of 8 metabolites as in the first experiment. Five more sets of simulated signals were constructed:

**set 2** = set 1 with a pronounced water resonance added,

**set 3** = set 1 with low white noise (SNR = 25),

**set 4** = set 1 with high noise (SNR = 7),

**set 5** = set 1 with baseline distortion

**set 6** = set 1 with water, baseline and high noise.

The simulated baselines were computed as sum of Gaussians centered at specific frequencies, as prescribed in the reference [112]. The water signal (appearing in set 2 and set 6) has been extracted from an *in vivo* spectrum by means of the HLSVD PRO method [73]. The additive noise is a circular Gaussian white noise with a standard deviation  $\sigma$  defined as the ratio of a certain reference peak height and the SNR, in the frequency domain.

For illustration, we plot one signal from each of the six datasets in Figure 7.8.

The influence of water, baseline and noise was also evaluated by means of the performance measure (7.4).

### 7.4.2 Results on simulated data

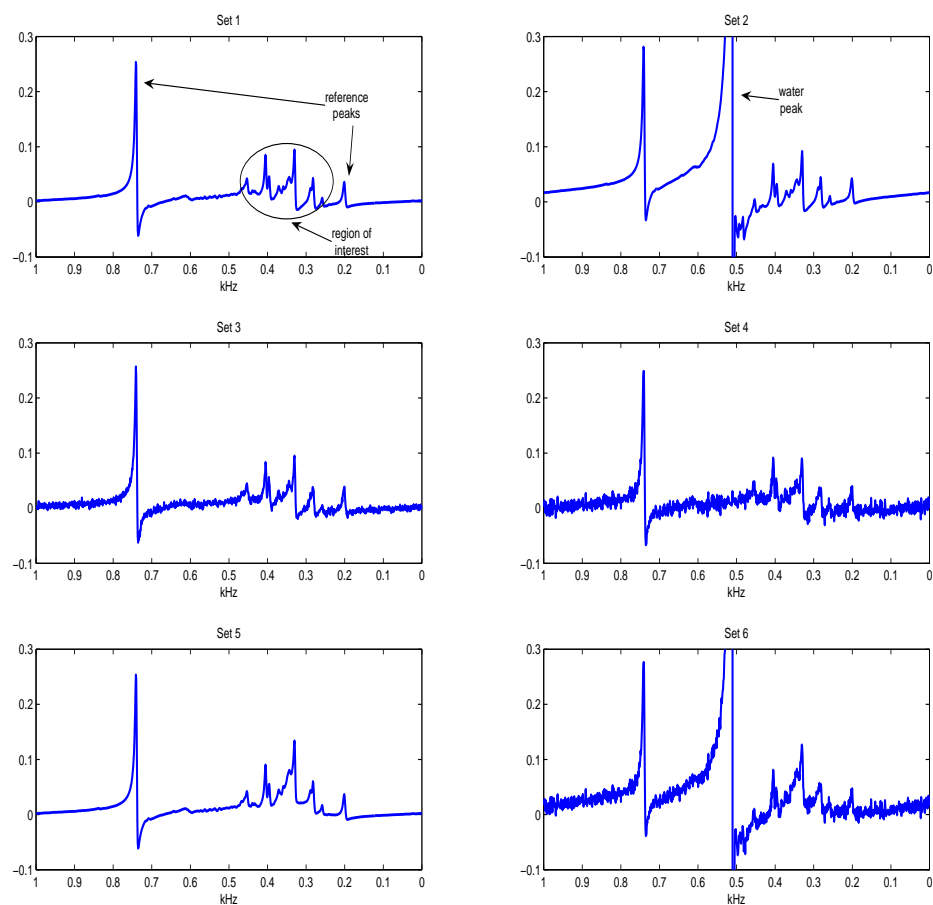
Results of AQSES on all data sets are presented in Table 7.1 and in Figure 7.9.

**Table 7.1.** Performance measure values for each metabolite and each simulation set (in percentage, cf. Eq. 7.4).

	NAA	Myo	Cr	Pch	Glu	Lac	Lip1	Lip2
exp. 1, Set 1	0.21	3.67	1.31	1.74	0.61	6.08	2.51	0.54
exp. 2, Set 1	0.58	13.54	1.25	15.18	1.18	8.47	2.74	0.56
exp. 3, Set 2	0.50	1.39	0.42	1.56	0.82	12.09	8.56	1.67
exp. 3, Set 3	3.13	7.37	4.13	5.94	4.86	20.23	19.88	7.98
exp. 3, Set 4	5.76	11.12	6.09	9.83	8.04	23.84	34.13	15.55
exp. 3, Set 5	7.90	11.67	4.31	15.43	27.91	29.14	30.55	7.75
exp. 3, Set 6	11.09	13.77	8.01	16.84	30.83	26.43	32.33	18.52

The first row of Table 7.1 shows the results of the **first** experiment. The PM values are relatively low, in the range of a few percentages. The biggest error of 6% pertains to lactate. A closer inspection of the relative errors between the simulated and estimated values for Set 1 (sorted and averaged between all metabolites and depicted as the lower line in Figure 7.9) shows that in 80% of the cases the estimation is almost perfect, with average relative errors under 0.001%.

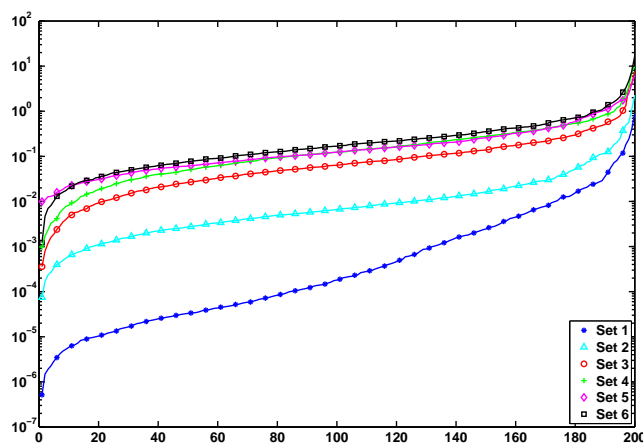
The **second** simulated example shows that the results are slightly affected if the numbers of metabolites in the basis set and in the simulated spectra are not equal. The PM is, for the majority of metabolites, of the same order of magnitude as in the first experiment



**Figure 7.8.** The upper-left plot shows the real part of the Fourier transform of an arbitrary noiseless signal from Set 1. The other plots show the corresponding signals when different nuisance components are added to the noiseless signal.

(second row of Table 7.1). The estimates for the amplitudes for Tau, Ala and Glc were generally small (while their true value is zero); the standard deviations were, respectively, 39.42, 18.12 and 48.45. These values are influenced by a few outliers, which correspond to cases when several metabolites were misfitted. In these cases, extreme values are also present in some overlapping peaks, such as Lac and Lip1. We noticed that Lac and Lip1 were often in opposite phase, canceling each other out, in order to fit the relatively flat signal in their frequency region. This problem will be dealt with in the following chapter.

The **third** experiment investigates the robustness of AQSES against the additional



**Figure 7.9.** For each of the six data sets of 200 simulated signals, the individual relative errors between the true and estimated amplitudes for each simulation and for each metabolite were sorted in increasing order and then averaged for the eight metabolites within each set. The trend is that when the complexity of the simulation signals increases (set 1 towards set 6) the relative errors increase as well, but they stay below 1% for more than 90% of the cases.

nuisance components such as noise, baseline and water resonances. The performance measure values are reported in the last five rows of Table 7.1 for each simulation set. Inspection of the results of the data sets 2 to 6 shows that the overall performance degrades when the complexity of the nuisance components in the simulations increases. However, Figure 7.9 confirms that in more than 90% of the simulations, the relative errors stay below 1%.

The maximum-phase filter removes satisfactorily the water component, as illustrated if we observe the differences between rows 1 and 3 of Table 7.1, corresponding to set 1 and set 2. Indeed, the water resonance does not significantly affect the relative square error.

The performance measure for different noise levels confirms the stability of AQSES against noise. At low noise values, the errors do not increase dramatically for any metabolites, except for Lac and Lip1. At high noise, the errors of most metabolites increase by a factor of about two, compared to the low noise level case.

The baseline affects each component but mainly Myo and Glu. These components are wider and therefore are more likely to be fitted by the baseline. Cr seems to be less affected. We notice that the PM of NAA and Cr remain under 12% in all cases.

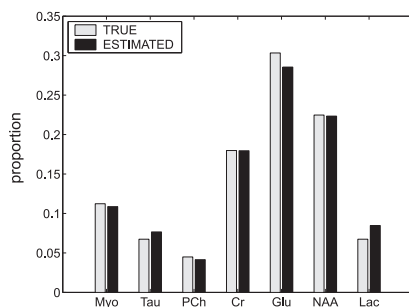
### 7.4.3 Experiments with real data

#### *In vitro* data

In a first experiment with real data, AQSES was validated using an *in vitro* sample. This means that the signal measured from a test chemical solution was quantified; the test so-

lution contained metabolites in known concentrations. The true and estimated proportions of metabolites have been compared (the proportion of metabolite  $k$  being the ratio of the concentration of metabolite  $k$  and the total concentration (all metabolites)).

The results show that the estimation errors are below an acceptable threshold of 25%, with lower errors obtained for the metabolites that had higher concentrations (below 8%). Figure 7.10 shows graphically the comparison between the true proportions inside the *in vitro* test solution and the proportions computed with AQSES.



**Figure 7.10.** *In vitro* test sample results. True and computed metabolite proportions are in close agreement.

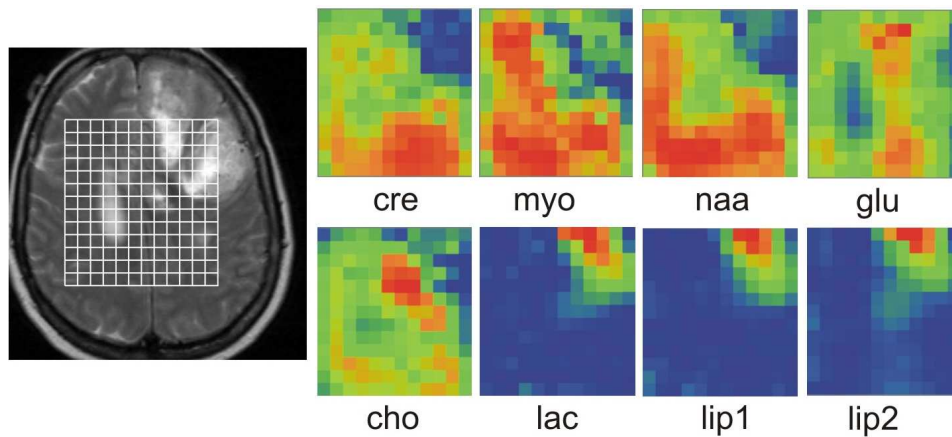
### ***In vivo* data**

In a second experiment with real data, *in vivo* MRS signals from a database containing spectra from normal tissue, and two types of brain tumor tissues were processed. This experiment is meant to show that AQSES can provide useful results for diagnosis and classification of brain tumors. The detailed explanation of these results in [119] can be summarized in the statement that the results were in accordance to the literature in the field, showing for each tissue type the presence of each important metabolite in a characteristic level of concentration with respect to the other chemicals.

An illustration of the results of AQSES on *in vivo* MRS signals is given in Figure 7.11. This is an example of using AQSES on a grid of MRS signals. The MR image on the left displays a slice of brain from a patient suffering from a glioblastoma tumor. The metabolic images on the right of the figure show concentration values corresponding to each metabolite from an 8-components database. The obtained values help the clinician to identify the location and grade of the tumor, since some of the metabolites have more elevated values and other metabolites have smaller values in the tumor region.

## **7.5 Conclusions**

This chapter presented the MRS data quantification problem and its implementation in the AQSES software. This practical application was, in fact, the driving motor that led to the development of the theory in Chapters 5 and 6.



**Figure 7.11.** *Metabolic images obtained with AQSES GUI. MRS signals in a grid are processed with AQSES and the resulting metabolite concentrations are plotted in colors ranging from blue (small concentration value) to red (high concentration value). The tumor region is clearly identified in the upper right corner, where some of the metabolites have concentrations higher than normal, and others have concentrations lower than in normal tissue.*

The AQSES framework provides a flexible and easy-to-use environment for quantification of short echo proton MRS spectra. The described experiments testify for the level of accuracy, robustness and reliability that AQSES shows in a variety of nuisance conditions.

One important remark is that the FIR filter included in AQSES removes suitably well the noise and the water resonances in the frequency region with no relevant metabolites, reducing partially (almost totally, for water) the effects of these nuisance components.

The experiments also showed a problematic situation: some metabolites that have common resonant frequencies were often in opposite phase, cancelling each other. In theory, the phase of the metabolites could be considered as equal; this change in the model is detailed in the next chapter.



## Chapter 8

# Constrained variable projection implementation

One important optimization tool implemented in the AQSES software for quantification of metabolite concentrations in MRS signals is a specialized variable projection (VARPRO) algorithm for solving separable nonlinear least squares problems. The standard VARPRO implementation for unconstrained separable nonlinear least squares is described in [47, 48] and implemented by John Bolstad [127]. The VARPRO implementation in AQSES is, however, nonstandard, for two main reasons:

1. AQSES allows *lower and upper bounds* on the nonlinear parameters and other special constraints on the linear parameters (to be described later in this chapter).
2. AQSES' optimization problem is a fitting problem involving *complex data*, while the standard implementation of VARPRO and other nonlinear least squares solvers are designed for real data; in AQSES, a *hybrid* method, which combines complex computations of function and Jacobian evaluations with an optimization over reals, is used.

## 8.1 Introduction

Separable nonlinear least squares fitting problems are minimization problems of the form

$$\min_{\mathbf{x}, \mathbf{a}} \|\mathbf{y} - \Phi(\mathbf{x})\mathbf{a}\|_2^2, \quad (8.1)$$

where  $\mathbf{y}$  is an  $m$ -dimensional given noisy data vector,  $\mathbf{x}$  denotes an  $n$ -dimensional vector of nonlinear parameters,  $\mathbf{a}$  is a  $p$ -dimensional vector of linear parameters, and  $\Phi$  is a nonlinear function that maps a vector  $\mathbf{x}$  into an  $m \times p$  matrix.

In the beginning of the seventies, the VARPRO method for solving separable least squares problems appeared [47]. During the past 30 years, VARPRO received attention from theoreticians, as well as practitioners. A recent review of VARPRO and its applications is given in the paper [48]. VARPRO uses the fact that the variable  $\mathbf{a}$  that appears linearly in the model function  $\Phi(\mathbf{x})\mathbf{a}$  can be optimally expressed as a linear least squares solution depending on the variable  $\mathbf{x}$ :  $\mathbf{a}^{\text{ls}}(\mathbf{x}) = \Phi(\mathbf{x})^\dagger \mathbf{y}$ , where  $^\dagger$  denotes the Moore-Penrose

pseudoinverse of a matrix. Therefore, this closed formula of  $\mathbf{a}$  can be plugged in into the original minimization problem, yielding the equivalent problem only in  $\mathbf{x}$ :

$$\min_{\mathbf{x}} \|(I_m - \Phi(\mathbf{x})\Phi(\mathbf{x})^\dagger)\mathbf{y}\|_2^2.$$

This problem can be solved with classical nonlinear (least squares) optimization methods, such as the Gauss-Newton or the Levenberg-Marquardt algorithms. These methods require evaluation of the Jacobian with respect to  $\mathbf{x}$  of the functional inside the norm. An essential idea is found in [67] and bears the name of Kaufman's simplification: it involves computing an approximate Jacobian instead of the true Jacobian, trading-off a negligible loss of accuracy in the Jacobian for a rather important computational time saving.

An interesting extension towards constrained variable projection is the case when *separable equality constraints* appear in the problem [68]. Another important related problem deals with having two separable classes of variables, without the requirement that some of them appear linearly [106].

In this chapter, we consider the classical linear/nonlinear variable separation and we analyze some extensions of the separable nonlinear least squares problem and of the VARPRO technique when (inequality) constraints appear within one or both classes of separable variables. These extensions are motivated by the biomedical application of quantifying metabolite concentrations from magnetic resonance spectroscopic signals. The extensions from the classical VARPRO presented in this chapter and needed in our application are shortly enumerated here, in increasing degree of difficulty:

**Complex data.** The VARPRO technique can be extended to work with complex variables and data. This observation helps us to address the computation of the residual and of the approximate Jacobian explicitly in the complex domain. We transform to real data when solving the resulting minimization problem (only in the nonlinear variables), since all classical nonlinear minimization implementations work with real data.

**Constraints on the nonlinear variables.** These constraints acting only on the nonlinear variables do not affect the VARPRO idea of projecting out the linear variables. Thus, these constraints can be simply imported to the resulting minimization problem only in  $\mathbf{x}$ . However, the classical Gauss-Newton or Levenberg-Marquardt method that we used in the unconstrained case must be replaced with a method that can take into account constraints. In our application, we focus only on lower and upper bounds for the variables; for this case, efficient methods are available (see the Netlib repository [www.netlib.org/opt/](http://www.netlib.org/opt/)).

**Constraints on the linear variables.** Imposing general constraints to the linear parameters takes away the possibility of projecting them out via a closed-form expression. Nevertheless, we would still like to keep the idea of solving an outer minimization problem only in the nonlinear variables  $\mathbf{x}$ , and solve the constrained linear least squares problem in  $\mathbf{a}$  in an efficient manner. In our application, we have non-negativity restrictions for some of the linear variables (but lower and upper bounds can be treated similarly). The inner problem in  $\mathbf{a}$  can then be efficiently solved with a quadratic programming type of method. In any case, we expect computational advantages and possibly faster convergence rate for this type of constrained VARPRO,

compared to solving the initial problem (8.1) with additional constraints, using a general nonlinear solver that does not distinguish between linear and nonlinear variables.

In Chapter 7, we gave an overview of the quantification of signals in magnetic resonance spectroscopy (MRS) [38, 63], and in Section 7.2 we emphasized the nonlinear least squares optimization problems that are obtained as mathematical formulations for this application. The VARPRO method was already used in MRS problems [129, 133]; a historical note on the application of VARPRO to MRS data quantification can be read in Section 17 of the review paper [48]. The previous usage of VARPRO in the mentioned papers was restricted to models of the type “sum of complex damped exponentials.” These models are appropriate for fitting the so-called *long echo-time* MRS signals. Nowadays, the nuclear magnetic resonance scanners record *short echo-time* signals that are richer in information because they display the responses of significantly more metabolites (chemical substances). The model given by a sum of complex exponentials no longer holds, since the response of each metabolite is spread out over the whole spectrum. A signal measured *in vivo* (from a region in the human brain) can be modeled as a combination of individual metabolite signals in the metabolite database. A baseline signal that accounts for the presence of some non-predominant unknown macromolecules must also be added to the model.

In Sections 8.2 and 8.3, the VARPRO extensions are presented in relation with the MRS application, and we summarize the studied cases in the following table:

Linear variables	Nonlinear variables	MRS data model	Section
complex unconstrained	real/complex (un)constrained	without/with baseline non-equal phases	8.2
real constrained	real/complex (un)constrained	without baseline equal phases	8.3.2
real partially constrained/ partially unconstrained	real/complex (un)constrained	with baseline equal phases	8.3.3

Finally, the numerical experiments in Section 8.4 aim to illustrate that an approximate Jacobian formula proposed for the constrained linear variables case yields accurate results for the MRS data quantification problem with equal phases.

## 8.2 Separable least squares with constraints on nonlinear variables

The motivation for the formulation in this section comes from the problem already described in Chapter 7, namely the MRS data quantification with non-equal phases. We show that problem (7.3) can be easily turned into a separable nonlinear least squares problem. The differences with respect to the classical separable problems solved by VARPRO is that the linear parameters, as well as the residual under the norm, will be complex-valued; moreover, simple equality or inequality constraints are allowed in the MRS data quantification problem formulation (7.3), but they may only affect the nonlinear (real-valued) variables.

We divide the analysis into two parts: quantification with or without a baseline. We start with the simple case when we ignore the baseline. This case gives us the opportu-

nity to revisit the original ideas behind VARPRO, including computational issues such as Kaufman's simplification for Jacobian computation.

### 8.2.1 MRS data model without baseline

The nonlinear least squares problem formulation (7.3) simply becomes

$$\min_{\substack{\alpha_1, \dots, \alpha_K \in \mathbb{C} \\ (\zeta_1, \dots, \zeta_K, \eta_1, \dots, \eta_K) \in \Omega}} \frac{1}{m} \sum_{t=t_0}^{t_{m-1}} \left| y(t) - \sum_{k=1}^K \alpha_k (\zeta_k)^t (\eta_k)^{t^2} v_k(t) \right|^2. \quad (8.2)$$

Problem (8.2) is a separable problem, where linear parameters  $\alpha_k$  can be projected out of the least squares problem, and only a smaller sized nonlinear least squares problem remains to be solved for the nonlinear variables  $\zeta_k, \eta_k$ . For the optimization over the (possibly constrained) set of parameter values for  $\zeta_k, \eta_k$ , we choose for an iterative minimization algorithm of the Levenberg-Marquardt type [90]. A trust-region implementation that allows imposing bounds on the real variables is described in [41]. Without entering into the details of such an algorithm, we continue our exposition by providing its necessary inputs: initial starting values, as well as procedures to evaluate the function value and the corresponding Jacobian, at each arbitrary set of parameter values.

We set all initial values for the real-valued nonlinear parameters to zero. This is a reasonable starting point, since it means that we start the optimization with *no spectral corrections* to the signals in the database, which corresponds to an ideal situation.

#### Function evaluation

We rewrite (8.2) as

$$\min_{\alpha, \mathbf{x} \in \Omega} \frac{1}{m} \|\mathbf{y} - \Phi(\mathbf{x})\alpha\|_2^2, \quad (8.3)$$

where  $\mathbf{y}$  is a column vector containing  $y(t_0), \dots, y(t_{m-1})$ ,  $\alpha$  is a complex  $K$ -dimensional column vector containing the complex amplitudes,  $\mathbf{x}$  is a vector formed from all nonlinear variables (preferably, the real-valued  $d_k, f_k, g_k, e_k$ ), and the  $m \times K$  complex-valued matrix  $\Phi(\mathbf{x})$  has elements of the form:

$$\begin{aligned} \Phi_{ik} &= (\zeta_k)^{t_i} (\eta_k)^{t_i^2} v_k(t_i) \\ &= \begin{cases} \exp((-d_k + jf_k)t_i + je_k t_i^2) v_k(t_i), & \text{for Lorentzian lineshapes;} \\ \exp(jf_k t_i + (-g_k + je_k)t_i^2) v_k(t_i), & \text{for Gaussian lineshapes;} \\ \exp((-d_k + jf_k)t_i + (-g_k + je_k)t_i^2) v_k(t_i), & \text{for Voigt lineshapes.} \end{cases} \end{aligned} \quad (8.4)$$

As mentioned in the introduction of this chapter, the optimal linear coefficients  $\alpha^{\text{ls}}(\mathbf{x})$ , for some fixed values of the nonlinear coefficients,  $\mathbf{x}$ , can be plugged-in such that the residual that we need to compute is the following *variable projection functional*

$$\mathbf{y} - \Phi(\mathbf{x})\alpha^{\text{ls}}(\mathbf{x}) = (I - \Phi(\mathbf{x})\Phi(\mathbf{x})^\dagger)\mathbf{y}.$$

Clearly, we only need a basis for the column space of the matrix  $\Phi(\mathbf{x})$  in order to evaluate the projection matrix  $I - \Phi(\mathbf{x})\Phi(\mathbf{x})^\dagger$ . This basis can be obtained from the QR decomposition of  $\Phi(\mathbf{x})$ :

$$\Phi(\mathbf{x}) = QR = [Q_1 \quad Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where  $R_1 \in \mathbb{C}^{K \times K}$ ,  $Q_1 \in \mathbb{C}^{m \times K}$ , and  $Q_2 \in \mathbb{C}^{m \times (m-K)}$ . Then, the residual  $(I - \Phi(\mathbf{x})\Phi(\mathbf{x})^\dagger)\mathbf{y}$  becomes  $Q_2 Q_2^H \mathbf{y}$ . Since only the norm of this residual is in fact needed in the optimization algorithm, we can further simplify the definition of our residual by ignoring the multiplication with  $Q_2$  (which has orthonormal columns), and simply compute  $Q_2^H \mathbf{y}$  at each function evaluation.

### Jacobian evaluation

The gradient of the residual  $Q_2^H \mathbf{y}$  is also needed by the Levenberg-Marquardt type nonlinear least squares solver. Note that in the residual  $Q_2^H \mathbf{y}$ , the nonlinear parameters appear implicitly through  $Q_2$ .

Consider again the original variable projection functional  $f(\mathbf{x}) = (I - \Phi(\mathbf{x})\Phi(\mathbf{x})^\dagger)\mathbf{y}$ . The Jacobian of  $f$  with respect to any of the scalar variables  $x_k$  (here denoted by  $\nabla_k f$ ) can be derived with the following manipulations:

$$\begin{aligned} \nabla_k f &= -(\nabla_k \Phi)\Phi^\dagger \mathbf{y} - \Phi(\nabla_k(\Phi^\dagger))\mathbf{y} = -(\nabla_k \Phi)\Phi^\dagger \mathbf{y} - \Phi \nabla_k \left( (\Phi^H \Phi)^{-1} \Phi^H \right) \mathbf{y} \\ &= -(\nabla_k \Phi)\Phi^\dagger \mathbf{y} + \Phi (\Phi^H \Phi)^{-1} \left[ (\nabla_k \Phi)^H \Phi + \Phi^H (\nabla_k \Phi) \right] (\Phi^H \Phi)^{-1} \Phi^H \mathbf{y} \\ &= -\left( \nabla_k \Phi - (\Phi^\dagger)^H (\nabla_k \Phi)^H \Phi - \Phi \Phi^\dagger (\nabla_k \Phi) \right) \Phi^\dagger \mathbf{y}. \end{aligned}$$

Kaufman's simplification [67] proposes that only the part

$$-(\nabla_k \Phi - \Phi \Phi^\dagger (\nabla_k \Phi)) \Phi^\dagger \mathbf{y} = -(I - \Phi \Phi^\dagger) (\nabla_k \Phi) \Phi^\dagger \mathbf{y}$$

should be used to compute an approximate Jacobian, yielding a computational saving that is more important than the loss of accuracy in the Jacobian, which is negligible.

If we take into account the definition (8.4) of  $\Phi$  for the MRS data model, the matrix  $\nabla_k \Phi$  can be computed using the formulas:

$$\frac{\partial \Phi_{ik}}{\partial d_k} = -t_i \Phi_{ik}, \quad \frac{\partial \Phi_{ik}}{\partial f_k} = jt_i \Phi_{ik}, \quad \frac{\partial \Phi_{ik}}{\partial g_k} = -t_i^2 \Phi_{ik}, \quad \frac{\partial \Phi_{ik}}{\partial e_k} = jt_i^2 \Phi_{ik}. \quad (8.5)$$

Note here that the variable  $d_k$  (or  $f_k$ , or  $g_k$ , or  $e_k$ ) only appears in the column  $k$  of  $\Phi$ . Therefore, all other columns of  $\nabla_k \Phi$  different from the column  $k$  are identically zero:

$$\frac{\partial \Phi_{il}}{\partial d_k} = 0, \quad \frac{\partial \Phi_{il}}{\partial f_k} = 0, \quad \frac{\partial \Phi_{il}}{\partial g_k} = 0, \quad \frac{\partial \Phi_{il}}{\partial e_k} = 0, \quad \text{for } l \neq k.$$

To fix the ideas, the matrix  $\nabla_k \Phi$ , which represents the derivative of the matrix  $\Phi$  with respect to the  $k^{\text{th}}$  variable  $x_k$  (that is either  $d_k$ , or  $f_k$ , or  $g_k$ , or  $e_k$ ), is an  $m \times K$  matrix that

has the following structure

$$\nabla_k \Phi = \begin{bmatrix} 0 & 0 & \frac{\partial \Phi_{1k}}{\partial x_k} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \frac{\partial \Phi_{mk}}{\partial x_k} & 0 & 0 \end{bmatrix}$$

Thus, the column corresponding to  $x_k$  in the approximate Jacobian equals

$$\tilde{\nabla}_k f = -(I - \Phi\Phi^\dagger)(\nabla_k \Phi)\Phi^\dagger \mathbf{y} = -(I - \Phi\Phi^\dagger) \begin{bmatrix} \frac{\partial \Phi_{1k}}{\partial x_k} \\ \vdots \\ \frac{\partial \Phi_{mk}}{\partial x_k} \end{bmatrix} \alpha_k^{\text{ls}}, \quad (8.6)$$

where we used the fact that  $\alpha^{\text{ls}} = \Phi^\dagger \mathbf{y}$ . The complete approximate Jacobian  $\tilde{\nabla} f$  is obtained by putting next to each other all columns of type (8.6), one column for each nonlinear variable in our optimization.

For stable and efficient computation of the Jacobian, we make use of the QR decomposition of  $\Phi$ , as introduced before. Thus,  $\alpha^{\text{ls}} = R_1^{-1} Q_1^H \mathbf{y}$  and  $I - \Phi\Phi^\dagger = Q_2 Q_2^H$ . Since we ignore the factor  $Q_2$  in the function evaluation, we must also do the same thing in what concerns the Jacobian.

In the end, the approximate Jacobian using Kaufman's simplification is:

$$\tilde{\nabla} f = -Q_2^H \Delta \Phi, \quad (8.7)$$

where  $\Delta \Phi$  for the MRS data model in its most general formulation, is

$$\left[ \begin{array}{c|c|c|c} \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{1k}}{\partial d_k} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots \\ \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{mk}}{\partial d_k} & \cdot & \cdot \end{array} \right] \left[ \begin{array}{c|c|c|c} \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{1k}}{\partial g_k} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots \\ \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{mk}}{\partial g_k} & \cdot & \cdot \end{array} \right] \left[ \begin{array}{c|c|c|c} \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{1k}}{\partial f_k} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots \\ \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{mk}}{\partial f_k} & \cdot & \cdot \end{array} \right] \left[ \begin{array}{c|c|c|c} \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{1k}}{\partial e_k} & \cdot & \cdot \\ \vdots & \vdots & \vdots & \vdots \\ \cdot & \alpha_k^{\text{ls}} \frac{\partial \Phi_{mk}}{\partial e_k} & \cdot & \cdot \end{array} \right],$$

with  $k$  going from 1 to  $K$ . In cases when not all variables among  $d_k, g_k, f_k, e_k$ , are estimated in the model, or when we impose *prior knowledge* in the form of linear equalities between some variables of the same sort (leading to eliminations), the formula above simplifies by deleting the not-needed columns.

## 8.2.2 MRS data model with baseline

Little is changed in the variable projection implementation, when we augment the optimization criterion (8.2) to the regularized version (7.3). In fact, using the notation from (8.3), the minimization (7.3) can be written as

$$\min_{\alpha \in \mathbb{C}^K, \mathbf{x} \in \Omega, \mathbf{c} \in \mathbb{C}^n} \frac{1}{m} \left\| \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{x})\alpha + \mathcal{A}\mathbf{c} \\ \sqrt{m\lambda} D\mathbf{c} \end{bmatrix} \right\|_2^2,$$

which is also a separable nonlinear least squares problem, where the linear variables are  $\alpha$  and  $\mathbf{c}$ , and the nonlinear ones are  $\mathbf{x}$ .

For the function evaluation, we use the QR decomposition of the matrix

$$\begin{bmatrix} \mathcal{A} & \Phi(\mathbf{x}) \\ \sqrt{m\lambda D} & 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

with  $R_1 \in \mathbb{C}^{(n+K) \times (n+K)}$ , and  $Q_1, Q_2$  of appropriate sizes. The function value is  $Q_2^H \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}$ , and the projected linear variables have the expression

$$\begin{bmatrix} \mathbf{c}^{\text{ls}} \\ \alpha^{\text{ls}} \end{bmatrix} = R_1^{-1} Q_1^H \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}.$$

For approximate Jacobian evaluation, the only difference comes from the fact that we have augmented  $\Phi(\mathbf{x})$  with some blocks that *do not* depend on the nonlinear parameters. This translates into the fact that the new Jacobian is also extended with zero blocks of corresponding dimensions. All completely zero columns can be ignored in the implementation.

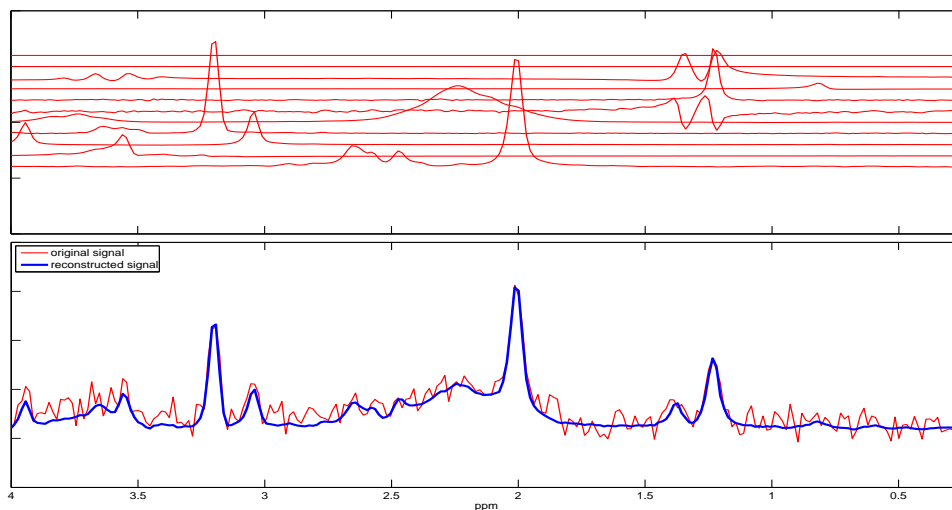
## 8.3 Separable least squares with constraints on linear variables

### 8.3.1 Motivation: MRS data quantification with equal phases and non-negativity constraint for the amplitudes

In this section, we are concentrating on the problems introduced by requiring equal phase corrections for all metabolites ( $\phi_1 = \dots = \phi_K =: \phi_0$ ). Remember that the phases entered into the problem in the previous sections through the *complex amplitudes*  $\alpha_k = a_k \exp(j\phi_k)$ , which were the complex linear parameters in the VARPRO method.

Requiring equal phase corrections is a reasonable approximation, since the phase distortions between different metabolites within an *in vivo* signal are negligible. Moreover, it was noticed in experiments with the non-equal phases version presented in Section 8.2 that in some cases (when the metabolite database contains signals with overlapping resonant frequency regions) there is a tendency for overlapping metabolites to compensate for each other by having opposite phases. In other words, these metabolites partially cancel each other, and thus their amplitudes are unreliably computed, although the residual is small. As an illustration, see the reconstructed signal, together with the database of corrected metabolite profiles, in Figure 8.1. Anticipating the method in this section, Figure 8.2 shows the fitting results when the equal phase constraint is used. The reconstructed signals are very similar in the two figures, but, noticeably, there are no longer artifacts from interchangeable metabolites in Figure 8.2. All the plots show real parts of the signals in the frequency domain.

Conceptually, imposing equal phases makes the model simpler, but practically, the method for solving the problem becomes a bit more complicated. In the next subsection, we shall see that the fact that we want to have equal phases implies that we can no longer use the VARPRO technique with complex linear variables. We must switch to a version where



**Figure 8.1.** *Fit with non-equal phases. Upper plot shows all the corrected metabolites spectra, which should be summed up in order to yield the reconstructed signal; bottom plot shows the noisy spectrum and the reconstructed spectrum (the less noisy thick line).*

the linear variables are only the real amplitudes, and introduce the unique phase variable  $\phi_0$  among the nonlinear variables. However, the real amplitudes must be non-negative in order to be meaningful.

The method developed here can be easily applied in the case when other constraints for the linear variables are imposed. One simple generalization involves lower and upper bounds for each individual linear variable; the non-negativity condition is a particular case thereof.

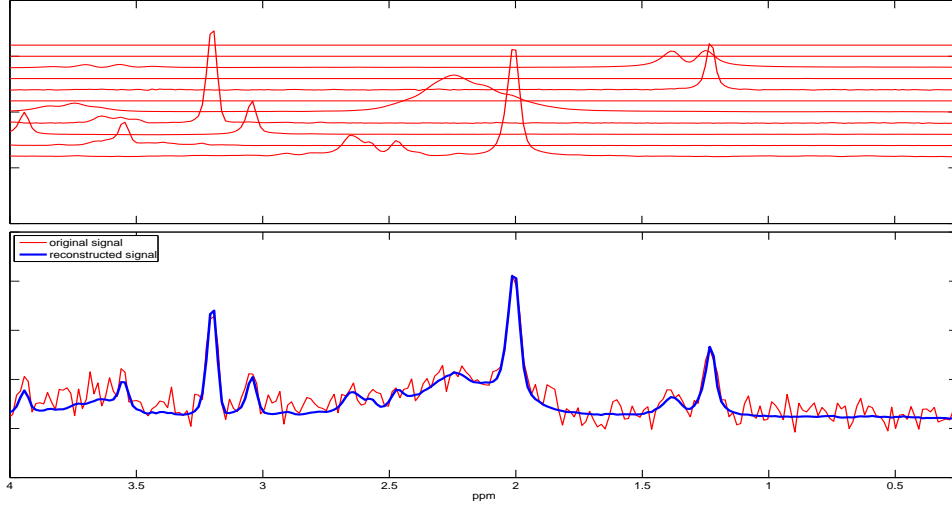
### 8.3.2 MRS data model without baseline

**The problem formulation is a separable nonlinear least squares with non-negative linear variables**

The nonlinear least squares problem formulation (7.3) becomes

$$\min_{\substack{a_1, \dots, a_K \in [0, \infty), \phi_0 \in (-\pi, \pi), \\ \zeta_1, \dots, \zeta_K, \eta_1, \dots, \eta_K \in \Omega}} \frac{1}{m} \sum_{t=t_0}^{t_{m-1}} \left| y(t) - \sum_{k=1}^K a_k \exp(j\phi_0) (\zeta_k)^t (\eta_k)^{t^2} v_k(t) \right|^2. \quad (8.8)$$

Problem (8.8) involves a mixture of real and complex variables. For practical optimization, we need to transform everything to real and to optimize only with respect to real param-



**Figure 8.2.** Fit with equal phases. The upper and bottom plots contain the same elements as in Figure 8.1. Note that the canceling effect around the value 1.3 ppm is not present, as opposed to Figure 8.1.

ters:

$$\min_{\substack{a_1, \dots, a_K \in [0, \infty), m \\ \phi_0 \in (-\pi, \pi), \\ (d_k, f_k, g_k, e_k) \in \Omega}} \frac{1}{m} \sum_{t=t_0}^{t_{m-1}} \left\{ \left( \text{real}(y(t)) - \sum_{k=1}^K a_k \text{real} \left( \exp(j\phi_0) (\zeta_k)^t (\eta_k)^{t^2} v_k(t) \right) \right)^2 \right. \\ \left. + \left( \text{imag}(y(t)) - \sum_{k=1}^K a_k \text{imag} \left( \exp(j\phi_0) (\zeta_k)^t (\eta_k)^{t^2} v_k(t) \right) \right)^2 \right\},$$

where  $\zeta_k$  and  $\eta_k$  will be substituted with their formulas depending on  $d_k$ ,  $f_k$ ,  $g_k$  and  $e_k$ . The set  $\Omega$  is used to impose simple linear (in)equality constraints on the nonlinear parameters  $d_k$ ,  $f_k$ ,  $g_k$  and  $e_k$ .

We rewrite the minimization problem in the following compact form, which emphasizes the fact that the parameters  $a_1, \dots, a_K$  appear linearly in the objective function:

$$\min_{\mathbf{a} \geq 0, \mathbf{x} \in \Omega} \frac{1}{m} \|\mathbf{y} - \Phi(\mathbf{x})\mathbf{a}\|_2^2, \quad (8.9)$$

where  $\mathbf{y}$  is a column vector containing

$$\text{real}(y(t_0)), \text{imag}(y(t_0)), \dots, \text{real}(y(t_{m-1})), \text{imag}(y(t_{m-1})),$$

$\mathbf{a}$  is the  $K$ -dimensional column vector containing the positive amplitudes  $a_1, \dots, a_K$ ,  $\mathbf{x}$  denotes the vector obtained from all the rest of the real parameters  $d_k$ ,  $f_k$ ,  $g_k$ ,  $e_k$  and  $\phi_0$ , and

the  $2m \times K$  matrix  $\Phi(\mathbf{x})$  has elements of the form:

$$\begin{aligned}\Phi_{(2i-1),k} &= \text{real} \left( \exp(j\phi_0) (\zeta_k)^{t_i} (\eta_k)^{t_i^2} v_k(t_i) \right), \\ \Phi_{(2i),k} &= \text{imag} \left( \exp(j\phi_0) (\zeta_k)^{t_i} (\eta_k)^{t_i^2} v_k(t_i) \right),\end{aligned}\quad (8.10)$$

$$\begin{aligned}\text{with } \exp(j\phi_0) (\zeta_k)^{t_i} (\eta_k)^{t_i^2} v_k(t_i) &= \\ \begin{cases} \exp(j\phi_0 + (-d_k + jf_k)t_i + je_k t_i^2) v_k(t_i), & \text{for Lorentzian lineshapes;} \\ \exp(j\phi_0 + jf_k t_i + (-g_k + je_k)t_i^2) v_k(t_i), & \text{for Gaussian lineshapes;} \\ \exp(j\phi_0 + (-d_k + jf_k)t_i + (-g_k + je_k)t_i^2) v_k(t_i), & \text{for Voigt lineshapes.} \end{cases}\end{aligned}$$

Note that without the non-negativity constraint on  $\mathbf{a}$ , problem (8.9) is a separable least squares problem, where the optimal linear coefficients  $\mathbf{a}^{\text{ls}}(\mathbf{x})$ , have a closed-form expression  $\mathbf{a}^{\text{ls}}(\mathbf{x}) = \Phi(\mathbf{x})^\dagger \mathbf{y}$ .

However, the non-negative least squares problem (8.9), with  $\mathbf{x}$  fixed, does not have closed-form solution for  $\mathbf{a}$ . An effective method for solving non-negative least squares (NNLS) problems is given in [74, Chapter 23]. The algorithm is very much linked to linear-quadratic programming theory and it is an iterative active set primal-dual method where convergence occurs when all elements of the dual vector become negative. Starting with a set of possible primal solutions (basis vectors), the algorithm computes an associated dual vector, and selects at each iteration the worst basis vector solution to be exchanged from the basis set, corresponding to the maximum (positive) element of the dual vector.

If, in other applications, bounds or other types of simple constraints on the linear parameters need to be imposed, then the non-negative least squares solver discussed above must be replaced by an appropriate method for the corresponding *constrained linear least squares* problem. For many practical cases (such as linear or quadratic inequality constraints), efficient methods and software implementations are available in the literature.

We denote the optimal non-negative solution of (8.9) for a given  $\mathbf{x}$  by  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$ . It seems preferable to optimize (8.9) only over the variables in  $\mathbf{x}$ , while estimating  $\mathbf{a}$  at every iteration as  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$ . A conceptual advantage of this approach must be emphasized: if problem (8.9) is solved with a nonlinear least squares solver that treats  $\mathbf{a}$  and  $\mathbf{x}$  as nonlinear variables, without distinction, then any value close-to-zero in  $\mathbf{a}$  would cause an almost zero column in the Jacobian of the objective function, leading to possibly unreliable numerical computations. The implementation that optimizes only on  $\mathbf{x}$  and uses  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$  at each iteration does not suffer from this numerical problem.

The case of almost zero amplitudes is important in our application, since a practitioner might want to use a database of metabolites that has more elements than the actual number of chemicals significantly present in the signal to be quantified. Thus, identifying almost zero metabolite concentrations is an important case that should not be affected by numerical errors.

## Function and pseudo-Jacobian evaluation

The nonlinear least squares solver needs implementations for the specific objective function and Jacobian evaluations.

As explained before, the function evaluation is performed by computing the non-negative least squares solution at the current value of  $\mathbf{x}$ ,  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$ ; then, the residual vector is computed simply as  $\mathbf{y} - \Phi(\mathbf{x})\mathbf{a}^{\text{nnls}}(\mathbf{x})$ .

Unfortunately, the lack of closed-form expressions for  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$  implies that we do not have an expression for the true Jacobian of the residual with respect to  $\mathbf{x}$ . For computational efficiency, we propose to use an adequate pseudo-Jacobian, instead of numerical differentiation methods.

Consider the function  $f(\mathbf{x}) = \mathbf{y} - \Phi(\mathbf{x})\mathbf{a}^{\text{nnls}}(\mathbf{x})$ . The Jacobian of  $f$  with respect to any of the scalar variables  $x_k$  (here denoted  $\nabla_k f$ ) has the formula:

$$\nabla_k f = -(\nabla_k \Phi(\mathbf{x}))\mathbf{a}^{\text{nnls}}(\mathbf{x}) - \Phi(\mathbf{x})(\nabla_k \mathbf{a}^{\text{nnls}}(\mathbf{x})). \quad (8.11)$$

While the matrix  $\nabla_k \Phi(\mathbf{x})$  is easily computable, the gradient  $\nabla_k \mathbf{a}^{\text{nnls}}(\mathbf{x})$  cannot be computed explicitly. At this point, a numerical differentiation technique can be used in order to estimate the matrix  $\nabla \mathbf{a}^{\text{nnls}}(\mathbf{x})$ . In Section 8.4, we show experimental results that are obtained with this approach for Jacobian computation, as well as with the approach described next, using another approximate Jacobian that is cheaper to compute.

Since  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$  is in many cases close to the least squares solution  $\mathbf{a}^{\text{ls}}(\mathbf{x}) = \Phi(\mathbf{x})^\dagger \mathbf{y}$ , we can approximate the gradient of  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$  with the one of  $\mathbf{a}^{\text{ls}}(\mathbf{x})$ , yielding:

$$\begin{aligned} \nabla_k f &\approx -(\nabla_k \Phi)\mathbf{a}^{\text{nnls}} - \Phi \nabla_k \left( (\Phi^\top \Phi)^{-1} \Phi^\top \right) \mathbf{y} \\ &= -(\nabla_k \Phi)\mathbf{a}^{\text{nnls}} + \Phi (\Phi^\top \Phi)^{-1} \left[ (\nabla_k \Phi)^\top \Phi + \Phi^\top (\nabla_k \Phi) \right] (\Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y} \\ &\approx -\left( \nabla_k \Phi - (\Phi^\dagger)^\top (\nabla_k \Phi)^\top \Phi - \Phi \Phi^\dagger (\nabla_k \Phi) \right) \mathbf{a}^{\text{nnls}}. \end{aligned}$$

Moreover, Kaufman's simplification [67] proposes to avoid the complicated computation of  $(\Phi^\dagger)^\top (\nabla_k \Phi)^\top \Phi$  and thus only the part

$$-(\nabla_k \Phi - \Phi \Phi^\dagger (\nabla_k \Phi)) \mathbf{a}^{\text{nnls}} = -(I - \Phi \Phi^\dagger) (\nabla_k \Phi) \mathbf{a}^{\text{nnls}}$$

can be used to compute an approximate Jacobian.

For more details on how to compute the elements of  $\nabla_k \Phi$  we refer to Section 8.2 or [116]. The only addition is that we have a new column in the Jacobian, corresponding to the variable  $\phi_0$ , which is equally treated as a nonlinear parameter. The gradient with respect to  $\phi_0$  is easily computable, since  $\phi_0$  only appears in the factor  $\exp(j\phi_0)$ .

In order to obtain the Jacobian matrix needed in the optimization process, all columns of the type  $(\nabla_k \Phi)\mathbf{a}^{\text{nnls}}$  should first be stacked into a matrix  $\Delta\Phi$ . To complete the Jacobian computation, the product  $(I - \Phi \Phi^\dagger) \cdot \Delta\Phi$  should be evaluated. For stable and efficient computation, we make use of the QR decomposition of  $\Phi$ ,

$$\Phi = QR = [Q_1 \quad Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where  $R$  is upper triangular,  $Q$  is an orthogonal matrix,  $R_1 \in \mathbb{R}^{K \times K}$ ,  $Q_1 \in \mathbb{R}^{2m \times K}$ , and  $Q_2 \in \mathbb{R}^{2m \times (2m-K)}$ . Thus,  $I - \Phi \Phi^\dagger = I - Q_1 Q_1^\top = Q_2 Q_2^\top$ , and then  $(I - \Phi \Phi^\dagger) \Delta\Phi = Q_2 Q_2^\top \Delta\Phi$ .

### 8.3.3 MRS data model with baseline

#### The minimization problem formulation

In this case, we augment the optimization criterion (8.8) to the regularized version that takes into account a smooth baseline reconstructed by penalized splines:

$$\min_{\substack{a_1, \dots, a_K \in [0, \infty), \phi_0 \in (-\pi, \pi), \\ \zeta_1, \dots, \zeta_K, \eta_1, \dots, \eta_K \in \Omega, \mathbf{c} \in \mathbb{C}^n}} \frac{1}{m} \sum_{t=t_0}^{t_{m-1}} \left| y(t) - \sum_{k=1}^K a_k \exp(j\phi_0) (\zeta_k)^t (\eta_k)^{t^2} v_k(t) - (\mathcal{A}\mathbf{c})(t) \right|^2 + \lambda \mathbf{c}^H D^H D \mathbf{c}, \quad (8.12)$$

where  $\mathcal{A} \in \mathbb{C}^{m \times n}$  is an inverse Fourier transformed spline matrix, the vector  $\mathbf{c} \in \mathbb{C}^n$  denotes the spline coefficients,  $\lambda$  is a fixed regularization (penalty) parameter, and the whole penalty term  $\lambda \mathbf{c}^H D^H D \mathbf{c}$  is responsible for ensuring a certain degree of smoothness to the frequency-domain baseline.

Using the notation from (8.9) and the transformation to real of all the complex elements (subscripted here with an  $r$ ), this minimization can be written as

$$\min_{\mathbf{a} \geq 0, \mathbf{x} \in \Omega, \mathbf{c}_r \in \mathbb{R}^{2n}} \frac{1}{m} \left\| \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \Phi(\mathbf{x})\mathbf{a} + \mathcal{A}_r \mathbf{c}_r \\ \sqrt{m\lambda} D_r \mathbf{c}_r \end{bmatrix} \right\|_2^2, \quad (8.13)$$

where  $\mathcal{A}_r \in \mathbb{R}^{2m \times 2n}$  is obtained from  $\mathcal{A}$  by unfolding all elements into real and imaginary parts and shuffling them such that each odd/even row corresponds to the real/imaginary part of an element of  $y(t)$ , and each odd/even column corresponds to the real/imaginary part of an element of the spline coefficient vector  $\mathbf{c}$ , which is also unfolded into the vector of real elements  $\mathbf{c}_r \in \mathbb{R}^{2n}$ . The matrix  $D_r$  is obtained from  $D$ , shuffled in the same manner.

The problem (8.13) is also a separable nonlinear least squares problem, where the linear variables are  $\mathbf{a}$  and  $\mathbf{c}_r$ , and the nonlinear ones are grouped in the vector  $\mathbf{x}$ . However, the linear amplitudes in  $\mathbf{a}$  are non-negatively constrained, while the spline parameters  $\mathbf{c}_r$  are free. Moreover, the coefficient matrices that multiply  $\mathbf{c}_r$  are independent of  $\mathbf{x}$ . This allows us to use an explicit optimal solution for  $\mathbf{c}_r$ , while still using a non-negative least squares solver to optimize  $\mathbf{a}$  at each new  $\mathbf{x}$ .

Solving the optimization problem (8.13) is done using a nonlinear least squares minimization over  $\mathbf{x}$ , where for each fixed  $\mathbf{x}$  (thus, at every function evaluation), the linear parameters are the optimal solutions of a minimization problem of the type:

$$\min_{\mathbf{a} \geq 0, \mathbf{c}_r \in \mathbb{R}^{2n}} \frac{1}{m} \left\| \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathcal{A}_r & \Phi \\ \sqrt{m\lambda} D_r & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_r \\ \mathbf{a} \end{bmatrix} \right\|_2^2. \quad (8.14)$$

Using a QR factorization of the matrix  $\begin{bmatrix} \mathcal{A}_r \\ \sqrt{m\lambda} D_r \end{bmatrix} = ST = [S_1 \ S_2] \begin{bmatrix} T_1 \\ 0 \end{bmatrix}$ , with  $T_1 \in \mathbb{C}^{2n \times 2n}$ , and  $S_1, S_2$  of appropriate sizes, the optimal  $\mathbf{c}_r$  is expressed as

$$\mathbf{c}_r^{\text{ls}} = T^\dagger S^\top \left( \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \Phi \\ 0 \end{bmatrix} \mathbf{a} \right) = T_1^{-1} S_1^\top \left( \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} - \begin{bmatrix} \Phi \\ 0 \end{bmatrix} \mathbf{a} \right),$$

which can be plugged in into (8.14) such that a non-negative least squares problem in the variable  $\mathbf{a}$  only remains to be solved:

$$\min_{\mathbf{a} \geq 0} \frac{1}{m} \left\| S(I - TT^\dagger)S^\top \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \Phi \\ 0 \end{pmatrix} \mathbf{a} \right\|_2^2 \iff \min_{\mathbf{a} \geq 0} \frac{1}{m} \left\| S_2^\top \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \Phi \\ 0 \end{pmatrix} \mathbf{a} \right\|_2^2,$$

where we used the fact that  $(I - TT^\dagger)S^\top = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} S^\top = \begin{bmatrix} 0 \\ S_2^\top \end{bmatrix}$ , and we ignored the multiplication with the orthogonal matrix  $S$  (since the norm is invariant to such an operation).

### Function and pseudo-Jacobian evaluation

To evaluate the residual needed in the nonlinear least squares algorithm, it is thus possible to ignore the orthogonal matrix  $S$  and to compute instead of

$$f(\mathbf{x}) = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{bmatrix} \mathcal{A}_r & \Phi(\mathbf{x}) \\ \sqrt{m\lambda}D_r & 0 \end{bmatrix} \begin{pmatrix} \mathbf{c}_r^{\text{ls}} \\ \mathbf{a}^{\text{nnls}} \end{pmatrix},$$

directly the residual  $\tilde{f}(\mathbf{x}) = S_2^\top \left( \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \Phi \\ 0 \end{pmatrix} \mathbf{a}^{\text{nnls}} \right)$ .

The Jacobian that we need to compute is

$$\nabla_{\mathbf{x}} \tilde{f} = -S_2^\top \nabla_{\mathbf{x}} \left( \begin{pmatrix} \Phi(\mathbf{x}) \\ 0 \end{pmatrix} \mathbf{a}^{\text{nnls}}(\mathbf{x}) \right) = -S_2^\top \begin{bmatrix} \nabla_{\mathbf{x}} (\Phi(\mathbf{x}) \mathbf{a}^{\text{nnls}}(\mathbf{x})) \\ 0 \end{bmatrix}.$$

For its approximate evaluation, we need the same tricks as in the case when there is no baseline. One difference is that the new Jacobian is extended with a zero block, which comes from the fact that we augmented  $\Phi(\mathbf{x})$  with some blocks that *do not* depend on the nonlinear parameters.

## 8.4 Numerical experiments

### 8.4.1 Properties of the pseudo-Jacobian

In this subsection, we focus on illustrating the fact that the pseudo-Jacobian introduced in Section 8.3.2 performs as good in our optimization problems as the alternative approach of using an approximate Jacobian with numerical differentiation. The latter Jacobian combines an analytical formula with the numerical differentiation of the part that does not have a closed-form expression; this is the part involving the vector  $\mathbf{a}^{\text{nnls}}(\mathbf{x})$ . For numerical differentiation, we choose a simple forward difference approximation, which involves computing the NNLS solution in as many vectors (neighboring the current  $\mathbf{x}$ ) as there are elements in the vector  $\mathbf{x}$ .

The computation of  $\nabla \mathbf{a}^{\text{nnls}}(\mathbf{x})$  using numerical differentiation and its use within the analytical formula of the full Jacobian (see (8.11)) is more efficient than using numerical differentiation for the full Jacobian itself. However, the pseudo-Jacobian from Section 8.3.2 is much more computationally efficient, since the NNLS solution is computed only once (at  $\mathbf{x}$ ).

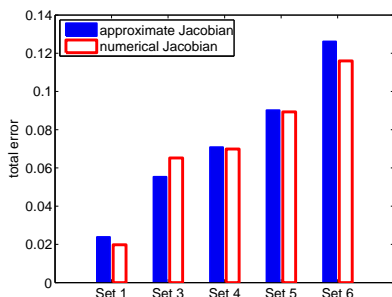
We use five data sets described in more detail in Chapter 7, §7.4.1, page 128 (set 1, 3-6). We mention again the characteristics of these simulation data sets:

- Set 1 consists of signals obtained from a metabolite database of 8 components. The model (7.2) with random values (but extracted from meaningful intervals) for the parameters of interest (amplitudes, damping corrections, frequency shifts, and equal phase correction  $\phi_0$ ). No baseline term and no noise are added to the simulation signals in this set.

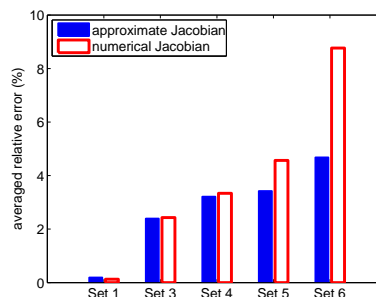
This set corresponds to a *zero-residual* nonlinear least squares problem.

- Set 2 from §7.4.1 is omitted as irrelevant for the comparisons herein.
- Set 3 is obtained from Set 1 by adding noise terms with a signal-to-noise ratio of 25.
- Set 4 is obtained from Set 1 by adding noise terms with a signal-to-noise ratio of 7.
- Set 5 is obtained from Set 1 by adding simulated (smooth in the frequency domain) baseline terms. This set corresponds to a zero-residual nonlinear least squares problem only in the case when the simulated baseline can be perfectly reconstructed by penalized splines.
- Set 6 is obtained from Set 5 by adding also noise terms.

In Figure 8.3, we show in a condensed manner the relative errors for all the simulation scenarios described above. The errors were computed with respect to the true values used for building up the simulation sets. The relative error formula is  $\|\mathbf{X}^{\text{estimated}} - \mathbf{X}^{\text{true}}\|_F / \|\mathbf{X}^{\text{true}}\|_F$ , where we stacked in the matrix  $\mathbf{X}^{\text{true}}$  (respectively,  $\mathbf{X}^{\text{estimated}}$ ) all the 100 vectors of true (respectively, estimated) parameters for the 100 simulation examples in each set. The norm  $\|\cdot\|_F$  is the Frobenius norm.



**Figure 8.3.** Comparison of total estimation errors for the two Jacobian variants and the 5 sets of simulations.



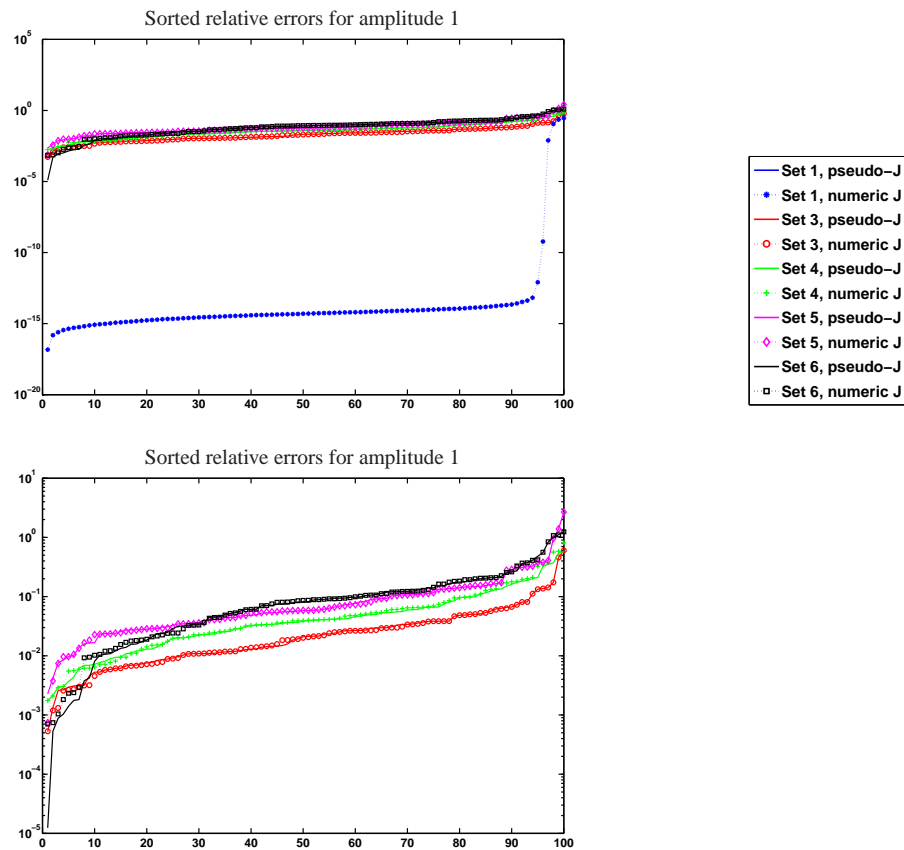
**Figure 8.4.** Comparison of averaged relative estimation errors for the two Jacobian variants and the 5 sets of simulations.

Moreover, Figure 8.4 views the same results under a different error measure: the averaged relative error, computed as the mean (over all variables in each set of 100 simulations) of individual relative square errors of the form  $(x_k^{\text{estimated}} - x_k^{\text{true}})^2 / (x_k^{\text{true}})^2$ .

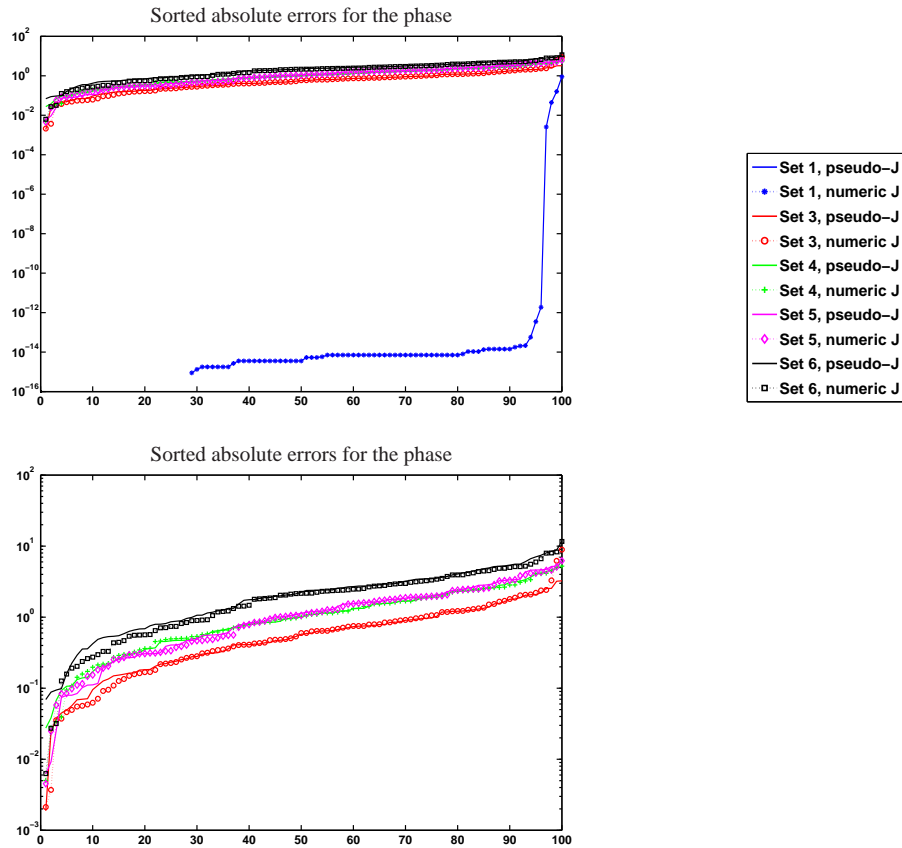
Obviously, there is no loss of accuracy related to the pseudo-Jacobian approach compared with the numerical differentiation scheme.

We plot the relative errors  $|x_k^{\text{estimated}} - x_k^{\text{true}}| / |x_k^{\text{true}}|$  for two individual variables (the amplitude  $a_1$  and the phase  $\phi_0$ ) in Figures 8.5 and 8.6. The scale is logarithmic and the relative errors (in percentages) are sorted for each of the five simulation sets and the two methods under investigation. The solid lines pertain to the relative errors obtained with the pseudo-Jacobian, and the lines with different markers are the corresponding errors for the numerical differentiation-based Jacobian. The two related curves for each set are very similar. Note also that the estimation errors for the zero-residual problems in Set 1 are at machine precision level for 95% of the simulations. The errors deteriorate when the noise level is increased or when the baseline term is added. However, the errors stay under a reasonable threshold of about 1 – 5%.

These numerical results show that the optimization approaches proposed in Section 8.3 are performing well, and that we can safely make use of the easily computable pseudo-Jacobian.



**Figure 8.5.** Relative errors for the variable  $a_1$  in 100 simulations for each of the five testing scenarios. The two plots contain the same information, but set 1 is removed from the plot below. The two proposed approximate Jacobians perform equally good.



**Figure 8.6.** Absolute errors for the variable  $\phi_0$  in 100 simulations for each of the five testing scenarios. For Set 1, 30% of the simulations give 0 residual, and 95% give machine precision residual. The two plots contain the same information, but set 1 is removed from the plot below. The two proposed approximate Jacobians perform equally good.

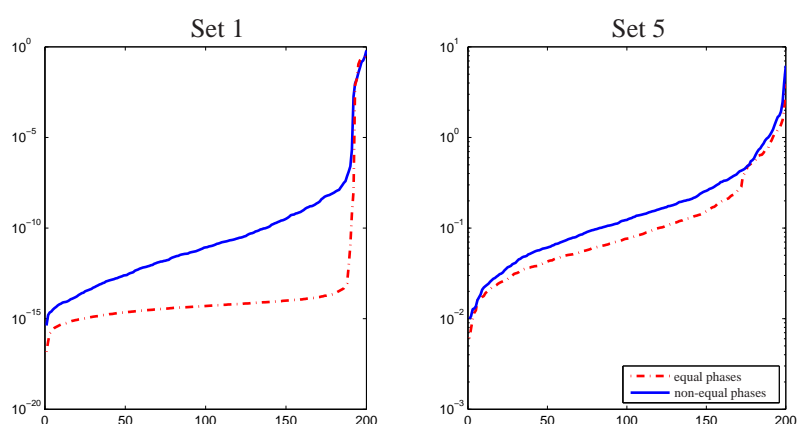
#### 8.4.2 Improvements of equal phases compared to the non-equal phases version

Finally, we provide the results that were obtained on all simulation data sets, as a way of illustrating the improvements of the equal-phases version of AQSES compared with the non-equal phases version, described in Chapter 7.

Table 8.1 is the equivalent of Table 7.1 on page 129. The biggest improvements are obtained for sets 1 and 5. To have a graphical illustration, we plot in Figure 8.7 a comparison between the relative errors in sets 1 and 5 using the non-equal and the equal phases versions.

**Table 8.1.** Performance measure values for each metabolite and each simulation set (in percentage, cf. equation 7.4 on page 128).

	NAA	Myo	Cr	Pch	Glu	Lac	Lip1	Lip2
exp. 1, Set 1	0.06	0.34	0.33	0.23	0.68	4.03	4.74	0.69
exp. 2, Set 1	0.80	5.32	0.63	12.92	0.75	2.23	4.97	0.67
exp. 3, Set 2	0.20	1.23	0.53	0.94	0.94	8.51	10.05	1.28
exp. 3, Set 3	2.52	5.83	2.78	4.11	4.40	15.90	20.13	7.01
exp. 3, Set 4	5.76	11.12	6.09	9.83	8.04	23.84	34.13	15.55
exp. 3, Set 5	7.13	8.83	3.17	4.40	22.72	17.42	24.43	7.45
exp. 3, Set 6	10.90	13.51	6.24	8.14	26.19	21.68	28.49	18.60

**Figure 8.7.** Comparison between the non-equal and the equal phases versions for sets 1 and 5. The relative errors for the amplitudes of all metabolites are averaged between all 8 metabolites in each of the 200 simulations. The equal phases constraint gives better estimated metabolite parameters.

## 8.5 Conclusions

We have described computational details related to the implementation of several variants of non-standard variable projection algorithms for separable nonlinear least squares. The main issues that we encountered are related to the introduction of constraints on the linear or nonlinear variables. In order to take into account the separability of the minimization criterion, the constrained (or unconstrained) linear subproblem must be solved efficiently and independently at each function evaluation of the outer nonlinear (constrained or unconstrained) minimization.

The described extensions were motivated by optimization problem formulations for the quantification of metabolites from short echo-time magnetic resonance spectroscopic signals. All the methods are implemented in the AQSES processing module of the AQSES GUI software package [3].



## Chapter 9

# Conclusions and open problems

## 9.1 General conclusions of the thesis

### 9.1.1 Regularization for linear problems

In the first part of the thesis, we mainly focused on the application of truncation and penalty-type regularization methods to the total least squares formulation. We also explored model selection techniques for these types of regularization methods.

We encountered and solved the following issues:

- In computing truncated total least squares solutions, there is a (theoretical) danger of ending up with a truncated problem where nonuniqueness and nongenericity of the solution can still be present (although this is hardly probable, when the data is noisy). However, to be on the safe side, we proposed to replace the truncated (total) least squares problem formulation with a *truncated core reduction* that can be used as a first step when we want to compute truncated SVD, truncated total least squares or other truncated scaled total least squares solutions.
- Regularized total least squares does not have closed-form solution as the regularized least squares (Tikhonov regularization) solution does. Therefore, local nonlinear optimization should be used in order to numerically compute the RTLS solution. We surveyed all the currently available methods from the literature, comparing them with our own method, based on iteratively solving quadratic eigenvalue problems.
- The error measures that are used in the context of model selection techniques such as cross validation, generalized cross validation, information criteria or even the L-curve, are not appropriate for consistently providing good estimates of regularization parameters or truncation levels. We define the *generalization error* that is an appropriate error measure for linear errors-in-variables models. Using this measure, we are able to adapt the classical model selection methods to the truncated total least squares and the regularized total least squares problems.

We also made an incursion into the generalization of the core problem concept to multiple right-hand sides systems. We found that an SVD form and a band diagonal form

for such core problems are computable. The core system that is obtained in this way has desirable properties – uniqueness and genericity of the solution – which otherwise are a major source of complications in the total least squares family of algorithms.

### 9.1.2 Regularization for nonlinear problems

In the second part of the thesis, we explored modeling nonlinear data. The main issues connected to regularization appeared in the context of nonparametric modeling. In this field, ill-posedness appears in the form of having to decide a trade-off between a good fit of the data and some model requirements. For instance, the semiparametric model involved in the biomedical application from Chapter 7 requires *smoothness* of the Fourier transform of an additive baseline function.

We drew the following conclusions:

- The template splines family unifies several classical spline formulations. In particular, this shows that the family of penalized splines can be put in its own right beside the classical family of smoothing splines. Template splines are, in many situations, easily computable using regularized linear least squares.
- It is relatively easy to include a nonparametric part into a nonlinear regression problem, in the case when the nonparametric part can be restricted through a weighted norm (as it is the case when imposing smoothness of the baseline term). The nonparametric part can then be modeled by using a spline basis, and thus only a linear term is actually added in the nonlinear least squares optimization criterion. However, special care should be taken when statistical information is retrieved from a regularized semiparametric regression problem, since, in general, we obtain biased estimates (but the bias can be corrected), and the classical nonlinear regression confidence bounds that use the Fisher information matrix must be adapted to take into account the penalty regularization term.
- We applied the semiparametric framework to the quantification problem of metabolite concentration from short echo-time MRS signals.
- An analysis of constrained variable projection optimization for separable nonlinear least squares showed that separable inequality constraints can be incorporated. However, the constraints on the linear variables must be simple enough to make the inner problem worth solving at each function evaluation of the outer nonlinear minimization. To maintain computational efficiency, we propose the use of a pseudo-Jacobian instead of an approximate Jacobian obtained through numerical differentiation.

## 9.2 Future work and open problems

### Nonlinear regularization

As a new trend, future work should shift towards truly nonlinear regularization. What we studied so far in the context of semiparametric nonlinear optimization has been based, in fact, on the assumption that the part that needs regularization (the nonparametric part)

appears *additively* in the model; moreover, the regularization that we used in the template splines context reduced everything to a linear parameterization in terms of a spline basis. We, thus, avoided nonlinear regularization altogether.

A general regularization problem is a problem where the parameters of interest appear nonlinearly in the model and in the regularization criterion. In fact, we had a glance at a nonlinear optimization problem when we studied regularized total least squares, since that formulation involved the orthogonal distance  $\|Ax - b\|^2 / (\|x\|^2 + 1)$  objective function. We have categorized RTLS within the framework of regularization for linear models only because it deals with a linear model  $Ax \approx b$ .

### Multidimensional ill-posed problems

Ill-posed problems with multiple right hand sides would benefit from a further analysis. For linear models, we discussed truncated core reductions, but we did not have the chance yet to thoroughly analyze how effective this kind of reduction helps in practical problems. (As a side remark, we are in fact not aware of a benchmark of ill-posed problems with multiple right-hand sides.)

For the regularized total least squares problem, extensions to multiple right-hand sides have not yet been attempted. A simple idea to approach multiple right-hand sides linear problems, regardless of the regularization formulation, is to *vectorize* the problem and, thus, transform it to a large sparse single right hand-side problem. This idea is already in use in image deblurring regularization techniques, for instance, because in that case the obtained large matrix has an embedded structure that can be exploited.

### Bidiagonal approximation

We speculated at the end of Section 2.3.3 that for the ill-posed system with multiple right-hand sides or for the nullspace formulation, it is also possible to use simple bidiagonalization, instead of block or band diagonalization. This conjecture should be further analyzed.

### Multiple penalties regularization

We mentioned in the introduction that it is possible to have more than one penalty term in a penalty-type regularization criterion. The problem of estimating multiple regularization parameters appears. Although in theory model selection techniques could be easily adapted to this demand, further study is needed in order to make it solvable in practice, even for quite simple problem formulations.

### More validations and improvements of AQSES

For the AQSES software package, we need a more extensive test suite for the evaluation of the baseline reconstruction in *in vivo* signals. The problem is that the baseline is unknown in real signals. We have preliminary results indicating that AQSES is more accurate in simulations than other software packages (mentioned in §7.1.2), but on *in vivo* data more experiments and comparisons should be done.



## Appendix A

# More theory on model selection

## A.1 Derivation of generalized cross validation

### A.1.1 From leave-one-out to generalized cross validation

Let  $F(t, x) \approx y$  be a (possibly nonlinear) model, where  $F : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$  is a known function,  $x$  is the model parameter vector and  $t_1, \dots, t_m, y_1, \dots, y_m$  are given data from the model. Assume that in order to estimate  $x$  we use a regularization scheme that depends on a regularization parameter  $\lambda$ . We denote  $\hat{x}_\lambda$  the regularized solution obtained when the regularization parameter has a fixed value  $\lambda$  and when all  $m$  available data instances are used for estimation. Let also  $\hat{x}_\lambda^{[-i]}$  denote the solution vector obtained with the same regularization method, when the  $i^{\text{th}}$  point  $(t_i, y_i)$  is omitted from the data set.

The leave-one-out cross validation chooses a  $\lambda$  that minimizes the function

$$CV(\lambda) := \frac{1}{m} \sum_{i=1}^m \left( y_i - F \left( t_i, \hat{x}_\lambda^{[-i]} \right) \right)^2.$$

This formulation is inconvenient since it involves solving  $m$  estimation problems, one for each deleted data point. We show that it is possible, as in the classical linear case, to simplify this formulation and to require only the solution of the total problem in order to evaluate the cross validation function.

First we emphasize the influence that the  $i^{\text{th}}$  measured output  $y_i$  has on the optimal solution, for a fixed value of  $\lambda$  and for fixed values of the other data points in  $y_1, \dots, y_m$ . We denote the direct link between  $y_i$  and the corresponding component of the optimal model  $\hat{y}_\lambda$  by a function  $h$  such that  $h(y_i) = (\hat{y}_\lambda)_i = F(t_i, \hat{x}_\lambda)$ .

Secondly, we discuss the leaving-one-out lemma that was proved in the context of smoothing splines [19], which still holds trivially for our nonlinear problem. It ensures that if the measured  $y_i$  was by any chance equal to the function value predicted by the solution computed without the  $i^{\text{th}}$  measurement, *i.e.*,

$$y_i = (\hat{y}_\lambda^{[-i]})_i := F \left( t_i, \hat{x}_\lambda^{[-i]} \right),$$

then the vector  $\hat{x}_\lambda^{[-i]}$  would be the optimal solution for the complete regularized estimation

problem as well. We can write this observation in terms of the function  $h$  as  $h((\hat{y}_\lambda^{[-i]})_i) = (\hat{y}_\lambda^{[-i]})_i$ .

Using a similar trick as in [136] for smoothing splines and in [69] for nonlinear nonparametric regression, we have:

$$y_i - F(t_i, \hat{x}_\lambda^{[-i]}) = \frac{y_i - F(t_i, \hat{x}_\lambda)}{1 - \Delta_i(\lambda)},$$

where

$$\Delta_i(\lambda) := \frac{(F(t_i, \hat{x}_\lambda) - F(t_i, \hat{x}_\lambda^{[-i]}))}{y_i - F(t_i, \hat{x}_\lambda^{[-i]})} = \frac{h(y_i) - h((\hat{y}_\lambda^{[-i]})_i)}{y_i - (\hat{y}_\lambda^{[-i]})_i},$$

which holds whenever  $y_i \neq (\hat{y}_\lambda^{[-i]})_i$ .  $\Delta_i(\lambda)$  is a divided difference for the function  $h$ , which can be approximated with the derivative of  $h$ . This leads to the definition of the following generalized influence (or smoother or hat) matrix  $S(\lambda)$ , which agrees with the definition given by [89] for the generalized influence matrix in a nonlinear context:

$$S(\lambda)_{ij} := \frac{\partial (F(t_i, \hat{x}_\lambda))}{\partial y_j} = \frac{\partial (\hat{y}_\lambda)_i}{\partial y_j}, \quad (\text{A.1})$$

for which  $S(\lambda)_{ii} \approx \Delta_i(\lambda)$ , as a first order approximation.

The ordinary cross validation cost function can be approximated by

$$CV(\lambda) \approx \frac{1}{m} \sum_{i=1}^m \frac{(y_i - F(t_i, \hat{x}_\lambda))^2}{(1 - S(\lambda)_{ii})^2},$$

and the generalized cross validation is its “rotation-invariant” version:

$$GCV(\lambda) = \frac{1}{m} \sum_{i=1}^m (y_i - F(t_i, \hat{x}_\lambda))^2 / \left[ \frac{1}{m} \text{Tr}(I_m - S(\lambda)) \right]^2 = \frac{m \|\hat{\mathbf{y}}_\lambda - \mathbf{y}\|_2^2}{[\text{Tr}(I_m - S(\lambda))]^2}, \quad (\text{A.2})$$

where  $\mathbf{y}$  is the vector  $[y_1, \dots, y_m]^\top$ , and  $\hat{\mathbf{y}}_\lambda$  is the vector  $[F(t_1, \hat{x}_\lambda), \dots, F(t_m, \hat{x}_\lambda)]^\top$ .

### A.1.2 Computation of the influence matrix

The definition formula of the generalized influence matrix (A.1) might not be computationally friendly. It depends on the specific regularization scheme on how  $S(\lambda)$  is computed.

Assume that the regularization estimation method involves the minimization of a function  $\mathbf{E}(\mathbf{y}; x; \lambda)$ .

We compute explicitly

$$S(\lambda) = \frac{\partial \hat{\mathbf{y}}_\lambda}{\partial \mathbf{y}} = \frac{\partial \mathbf{F}(\hat{x}_\lambda)}{\partial \mathbf{y}} = \frac{\partial \mathbf{F}(\hat{x}_\lambda)}{\partial x} \frac{\partial (\hat{x}_\lambda)}{\partial \mathbf{y}}. \quad (\text{A.3})$$

Since  $\hat{x}_\lambda$  is optimal for given data  $\mathbf{y}$ , it means that

$$\frac{\partial \mathbf{E}(\mathbf{y}, \hat{x}_\lambda; \lambda)}{\partial x} = 0.$$

On the other hand, if  $\mathbf{y}$  is ‘perturbed’ and the data becomes  $\mathbf{y} + d\mathbf{y}$ , the optimum also changes. We denote the new optimal solution by  $\hat{x}_\lambda + dx$ ; it satisfies

$$\frac{\partial \mathbf{E}(\mathbf{y} + d\mathbf{y}, \hat{x}_\lambda + dx; \lambda)}{\partial x} = 0.$$

Using a first order Taylor approximation, it implies that

$$\frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{x}_\lambda; \lambda)}{\partial x \partial \mathbf{y}^\top} d\mathbf{y} + \frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{x}_\lambda; \lambda)}{\partial x \partial x^\top} dx \approx 0. \quad (\text{A.4})$$

Denote the block of the Hessian of  $\mathbf{E}$  corresponding to  $x$  by  $\hat{H}(\lambda)$ :

$$\hat{H}(\lambda) := \frac{\partial^2}{\partial x \partial x^\top} \mathbf{E}(\mathbf{y}, \hat{x}_\lambda; \lambda),$$

and the block  $\frac{\partial^2 \mathbf{E}(\mathbf{y}, \hat{x}_\lambda; \lambda)}{\partial x \partial \mathbf{y}^\top}$  by  $\hat{J}(\lambda)$ .

Since  $d\mathbf{y}$  and  $dx$  represent small perturbations on  $\mathbf{y}$  and  $\hat{x}_\lambda$ , we can use the relation (A.4) and set the matrix  $-\hat{H}(\lambda)^{-1} \hat{J}(\lambda)^\top$  as approximation for the differential  $\frac{\partial \hat{x}_\lambda}{\partial \mathbf{y}}$  in (A.3). In conclusion, the generalized influence matrix can be computed as:

$$S(\lambda) = -\frac{\partial \mathbf{F}(\hat{x}_\lambda)}{\partial x} \hat{H}(\lambda)^{-1} \hat{J}(\lambda)^\top.$$

Note that in the case when  $\mathbf{y}$  appears in the function  $\mathbf{E}$  in a square error term of the form

$$\frac{1}{2} \|\mathbf{y} - \hat{\mathbf{y}}_\lambda\|_2^2,$$

then  $\hat{J}(\lambda)$  becomes

$$\hat{J}(\lambda) = -\frac{\partial \mathbf{F}(\hat{x}_\lambda)}{\partial x} = -\nabla \mathbf{F}(\hat{x}_\lambda),$$

and, thus,  $S(\lambda) = \nabla \mathbf{F}(\hat{x}_\lambda) \hat{H}(\lambda)^{-1} \nabla \mathbf{F}(\hat{x}_\lambda)^\top$ .

## A.2 Derivation of information criteria

In this section, we consider a probabilistic setting, in the sense that what we call ‘‘the data’’ is now a realization (or a set of realizations) of a random variable.

### The Kullback-Leibler measure

If  $f$  denotes a density function of the ‘‘true’’ distribution and  $g$  the density function of another distribution, then the Kullback-Leibler divergence is:

$$\text{KL}(f, g) = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} dx = \int_{-\infty}^{\infty} f(x) \log f(x) dx - \int_{-\infty}^{\infty} f(x) \log g(x) dx.$$

Properties of KL include: nonnegativity ( $\text{KL}(f, g) \geq 0$ ) and the equivalence  $\text{KL}(f, g) = 0$  if and only if  $f \equiv g$  almost everywhere. In fact,  $\text{KL}(f, g)$  quantifies the loss when model  $g$  is used to approximate the ‘‘reality’’  $f$ .

Suppose one has several statistical models that try to explain a random variable/vector whose true distribution has a density function  $f$ ; suppose that these models are expressed in terms of probability densities (say,  $g_1, \dots, g_r$ ). Then, ideally, one could compute KL for each of the  $r$  models and select the model that loses the least information with respect to the “truth”. However, these computations require knowledge of the full reality and complete description of the approximating models (not estimates plugged in for parameters, for instance). Obviously, one cannot compute KL.

### Akaike’s information criterion

Most of the information criteria are trying to approximate the KL measure in a computable way. Akaike [1] was the first to show how KL can be *estimated* from data, based on the maximized empirical log-likelihood (or maximum entropy). However, his formula was derived under some restrictive assumptions. Other authors improved his idea, giving rise to new and more complicated information criteria.

Note that minimizing  $\text{KL}(f, g)$  amounts to maximizing the term

$$\int_{-\infty}^{\infty} f(x) \log g(x) dx = \mathcal{E}[\log g(x)],$$

where  $\mathcal{E}$  denotes expectation (with respect to the true – and unknown – statistical distribution  $f$ ).

From given data (instances of  $x$ , denoted by  $z_1, \dots, z_m$ ), one can compute the *log-likelihood*  $\ell := \sum_{i=1}^m \log g(z_i)$ . To take  $g$  out of anonymity, we assume that  $g$  is a parametric density function that depends on an unknown  $d$ -dimensional vector of parameters  $\theta_0$ . The log-likelihood becomes, then, a function of  $\theta$ :

$$\ell(\theta) := \sum_{i=1}^m \log g(z_i; \theta).$$

The maximum likelihood estimate of  $\theta_0$  is consistent and asymptotically normal:

$$\hat{\theta} \rightarrow \theta_0 \quad \text{and} \quad \sqrt{m}(\hat{\theta} - \theta_0) \sim \mathcal{N}(0, (\mathcal{F}(\theta_0))^{-1}) \quad \text{when } m \rightarrow \infty,$$

where  $\mathcal{F}(\theta_0) := -\mathcal{E} \left[ \frac{\partial^2}{\partial \theta \partial \theta^\top} \log g(x; \theta_0) \right]$  is the Fisher information matrix at  $\theta_0$ . Moreover,

$$m(\hat{\theta} - \theta_0)^\top \mathcal{F}(\theta_0)(\hat{\theta} - \theta_0) \sim \chi^2(d),$$

where  $\chi^2(d)$  is the chi-square distribution with  $d$  degrees of freedom. From a second order expansion, using the properties of  $\hat{\theta}$  as a maximum likelihood estimator (thus,  $\hat{\theta}$  is also unbiased:  $\mathcal{E}[\hat{\theta} - \theta_0] = 0$ ),

$$\mathcal{E} [m \log g(x; \hat{\theta}) - m \log g(x; \theta_0)] \approx \mathcal{E} \left[ m \frac{1}{2} (\hat{\theta} - \theta_0)^\top \mathcal{F}(\theta_0) (\hat{\theta} - \theta_0) \right] = \frac{1}{2} \mathcal{E} [\chi^2(d)] = \frac{d}{2}.$$

Remember that what we want to minimize is the KL distance, which amounts to maximizing  $\mathcal{E}[\log g(x; \theta_0)]$ . Since  $\theta_0$  is not available, it can be replaced by its maximum likelihood estimate  $\hat{\theta}$ , using the observed data  $z_1, \dots, z_m$ . The idea is to use the log-likelihood

$\frac{1}{m}\ell(\hat{\theta}) = \frac{1}{m}\sum_{i=1}^m \log g(z_i; \hat{\theta})$  as an estimate of  $\mathcal{E}[\log g(x; \hat{\theta})]$ . The asymptotic expected value of the bias between what we want to compute and what we can compute is:

$$\begin{aligned} \text{bias} &= \lim_{m \rightarrow \infty} \mathcal{E} \left\{ m\mathcal{E}[\log g(x; \theta_0)] - \sum_{i=1}^m \log g(z_i; \hat{\theta}) \right\} \\ &= \lim_{m \rightarrow \infty} \mathcal{E} \left\{ \underbrace{m\mathcal{E}[\log g(x; \theta_0)] - m\mathcal{E}[\log g(x; \hat{\theta})]}_{\rightarrow -d/2} + \underbrace{m\mathcal{E}[\log g(x; \hat{\theta})] - \sum_{i=1}^m \log g(z_i; \hat{\theta})}_{\rightarrow 0 \text{ (law of large numbers)}} \right\} \end{aligned}$$

This asymptotic bias correction estimate is what Akaike used to define “An Information Criterion” (AIC) as the minimization of:

$$\text{AIC} = -2\ell(\hat{\theta}) + 2d$$

### Modified Akaike’s information criterion

Akaike’s idea to correct for the bias term between the *averaged* maximized log-likelihood and the *expected* maximized log-likelihood is taken over by other authors who improved his bias estimate. The bias provided in the classical AIC (equal to the number of free parameters  $d$ ) is a crude first-order estimate of the bias, obtained in asymptotics and under the assumptions that the unknown model parameters are estimated by the *maximum likelihood* method and that the true distribution  $f$  is a member of the model distribution family  $g(\cdot, \theta)$ .

The bias can be better estimated by the formula (replacing  $d$ ) [65]: bias =  $\frac{m(d+1)}{m-d-1}$ , yielding the corrected AIC:

$$\text{AIC}_c = -2\ell(\hat{\theta}) + 2\frac{m(d+1)}{m-d-1}.$$

### Bayesian information criterion

The Bayesian information criterion (BIC) introduced by Schwarz [110] gives a very similar criterion starting from seemingly different concepts. The Bayesian framework averages the model’s likelihood over all possible  $\theta$ , given a prior distribution for  $\theta$  (that is, the likelihood is integrated over  $\theta$ , instead of being maximized). For given data  $z_1, \dots, z_m$ , we arrive at another bias in the KL divergence computation. (For more details, see, *e.g.*, [99].) The Bayesian information criterion is:

$$\text{BIC} = -2\ell(\hat{\theta}) + \frac{d}{2} \log m.$$

### Generalized information criterion

When no assumptions about the true model’s distribution are made, a more general information criterion can be defined [70]. In this case, the bias between the average of the maximized log-likelihood and the expected maximized log-likelihood is approximated by the expression

$$\frac{1}{m} \text{Tr}(\mathcal{F}^{-1} \mathcal{R}),$$

where  $\mathcal{F}$  is the Fisher information matrix  $-\left[\frac{\partial^2}{\partial\theta\partial\theta^\top}\log g(x;\hat{\theta})\right]$  computed at the maximum likelihood parameter  $\hat{\theta}$ , and  $\mathcal{R}$  is the product of the log likelihood gradients, that is,  $\mathcal{R} := \left[\frac{\partial}{\partial\theta}\log g(x;\hat{\theta})^\top \frac{\partial}{\partial\theta}\log g(x;\hat{\theta})\right]$ .

The quantity  $\text{Tr}(\mathcal{F}^{-1}\mathcal{R})$  is the *effective number of parameters*  $p_k^{\text{eff}}$  that we already encountered.

The generalized information criterion is defined as

$$\min_k -2\ell(\hat{\theta}) + 2p_k^{\text{eff}}. \quad (\text{A.5})$$

In linear estimation, the particular case when  $\mathcal{E}$  is the negative log-likelihood and  $p_k^{\text{eff}}$  is just the number of free parameters in the model, gives the classical AIC.

See also [14, 128] for related methods based on information complexity.

**Remark 14 (Link between generalized cross validation and information criteria)** We note here that generalized cross validation, in the general formulation

$$\min_\lambda \frac{\frac{1}{m}\|r_\lambda\|^2}{(1 - p^{\text{eff}}(\lambda)/m)^2}.$$

where  $r_\lambda$  denotes the model's misfit measure to the data  $D$  (e.g., the residual norm  $\|Ax^{\text{reg}}(\lambda) - b\|$ ), is linked to GIC, in the asymptotic situation ( $m \rightarrow \infty$ ). Using the approximation  $\frac{1}{(1-\varepsilon)^2} \approx 1 + 2\varepsilon$ , we obtain (setting  $\varepsilon = p^{\text{eff}}/m$ )

$$\frac{\frac{1}{m}\|r_\lambda\|^2}{(1 - p^{\text{eff}}(\lambda)/m)^2} \approx \frac{\|r_\lambda\|^2}{m} + 2 \frac{\|r_\lambda\|^2}{m} \cdot \frac{p^{\text{eff}}(\lambda)}{m}.$$

If we take  $\mathcal{L}(x^{\text{reg}}(\lambda); D) := \|r_\lambda\|^2$  and, in the second term,  $\frac{\|r_\lambda\|^2}{m}$  as an estimate for the variance  $\hat{\sigma}^2$ , we obtain a particular GIC criterion:

$$\min_\lambda \mathcal{L}(x^{\text{reg}}(\lambda); D) + 2\hat{\sigma}^2 p^{\text{eff}}(\lambda).$$

## Appendix B

# Fortran implementation of AQSES

In this appendix, we provide an overview of the software implementation structure inside the FORTRAN 77 module AQSES.

### Driver routine: `drivervp.f`

The `drivervp.f` function has the role of coordinating the main computational flow. The steps performed by `drivervp.f` are:

- initializations
- filtering, if required
- for each round, except last round
  - fit signal using current database (`mainround.f`)
- in the last round
  - fit signal using current database (`mainround.f`) and baseline if required (`setsplines.f` and `initsvspace.f`)
  - optimize baseline smoothness parameter by performing previous step repeatedly and evaluating generalized cross validation function
- compute estimated error bounds (`crerrorvp.f`)

### Computations for each round: `mainround.f`

Function `mainround.f` performs the whole optimization process of a single round and deals with both cases of considering a baseline in the fit or not). The steps follow:

1. initialize optimization parameters (`inivp.f`):
  - put all free (nonlinear) variables in the vector `alf`, in the order: dampings, Gaussian factors, frequencies, Eddy current factors;

- count the number of variables and set `nrv` to this value;
  - set corresponding linear bounds (that were given in the input variables ending in `-low` and `-upp`, e.g., `freqlow` and `frequpp`) for all variables into a  $2 \times nrv$  matrix of lower and upper bounds `B`.
2. compute number of linear parameters, number of constraints, and some index values for referencing different blocks within the complex workspace
  3. if baseline is used, then the baseline spline coefficient matrix and the smoothness penalty matrix should be copied to the “saved space”, operation done in the function `initsvspace.f`. The part containing the penalty matrix  $\sqrt{m\lambda}D$  is updated in `updsvspace.f` by using the newest value of the smoothness parameter `lambda`. The call to `updsvspace.f` provides also the QR decomposition of the blocks containing the baseline coefficient matrix and penalty matrix. This way, the QR factorization of the matrix  $\begin{bmatrix} \mathcal{A} & \Phi(\zeta, \eta) \\ \sqrt{m\lambda}D & 0 \end{bmatrix}$  that is needed at each function and Jacobian evaluations will be updated only in the block column that contains  $\Phi(\zeta, \eta)$ , thus saving computations.
  4. main optimization takes place, using the (slightly modified) DN2GB module implemented in `dn2gbvp.f`. Function and Jacobian evaluations are made through the `funcvp.f` and `jacovp.f` functions, where all operations are performed with complex data (e.g., evaluation of the linear parameters), but only at the end the complex data is transformed to real and fed to the DN2GB optimization solver.
  5. optimal nonlinear parameters stored in `alf` are transformed back to meaningful spectral variables `damp`, `gauss`, `freq`, `eddy`, `t0` in the function `updatevp.f`.
  6. optimal linear parameters corresponding to the optimal nonlinear variables in `alf` and computed as linear least squares solution are transformed to meaningful real spectral parameters `amp1` and `phas`, and to the baseline (if computed), in the routine `updatelin.f`.

### Function evaluation: `funcvp.f`

The following steps are taken in `funcvp.f`:

- The matrix  $\Phi$  from (8.4) is efficiently computed, by first evaluating the complex exponentials for each index  $k$ , but for a fixed value of  $t_i = 1$ , and then filling in each row of  $\Phi$  with appropriate values for each  $t_i$ . This means that instead of computing  $mK$  complex exponentials, we compute only  $K$  complex exponentials and we raise  $2mK$  complex numbers to real powers.

The matrix  $\Phi$  is stored in the array `expon` and it is kept for possible use in the Jacobian evaluation. We rely on the fact that there is always a function evaluation before any Jacobian evaluation at the current parameter values.

- In the case when a FIR filter is required, each column of  $\Phi$  is filtered.

- The filtered version of  $\Phi$  is stored in the reserved block of `svSpace`, which is either the upper `nrfp × effbasis`, when no baseline is used, or in the upper right block next to the filtered spline matrix `film`, when also the baseline is used. Thus, the `svSpace` workspace contains in fact the matrix 
$$\begin{bmatrix} \mathcal{A} & \Phi(\zeta, \eta) \\ \sqrt{m\lambda}D & 0 \end{bmatrix},$$
 but with filtered columns corresponding to  $\mathcal{A}$  and  $\Phi(\zeta, \eta)$ , and with a partial QR factorization in the left block column.
- The complete QR factorization of `svSpace` is computed, and a compact representation is saved in the subdiagonal part of the first `nrlinpar` columns of `svSpace`, and in the vector `tau`.
- The residual  $Q^H y$  is computed and stored in the  $(\text{nrlinpar} + 1)^{\text{st}}$  column of the matrix `svSpace`. Then, this complex residual is transformed to real, by unfolding real and imaginary parts into odd and even locations of the vector `fvecf`.

### Jacobian evaluation: `jacovp.f`

$\nabla_k \Phi$  is stored in a compact way, by keeping only nonzero columns. In other words, if a certain column of  $\Phi$  does not depend on a certain variable  $x_k$ , then the all-zero column will not be stored.

The following steps are taken in `funcvp.f`:

- Linear variables are computed from the linear least squares problem that involves the current  $\Phi$ . Since we already have the QR decomposition of the coefficient matrix in the least squares problem, the complex vector of linear parameters is easily computed by solving a triangular linear system of equations. (The same computation is performed in the routine `updatelin.f`, for extracting the linear parameters at the end of the whole optimization process.)

The linear coefficients are stored in the  $(\text{nrlinpar} + 1)^{\text{th}}$  column of `svSpace`, with the baseline coefficients in the first `noslpars` positions (note that when the baseline is not required, `noslpars` is zero), and the complex amplitudes in the next `effbasis` positions.

- For each of the active nonlinear variables (dampings, Gaussian dampings, frequencies, Eddy current corrections), a corresponding column is added to the Jacobian. The implemented formulas are found in Subsection 8.2.1.
- If a filter is required, then each stored column is also passed through the FIR filter, and the columns from `nrlinpar + 2` to `nrlinpar + nrvar + 2` of `svSpace` are used for storage.
- A zero block corresponding to the penalty part of the optimization is added under the  $\Delta\Phi$  block, since the penalty term does not involve the nonlinear parameters.
- Kaufman's simplification is used; it implies that the above mentioned columns of `svSpace` are multiplied with the block  $-Q_2$  of the unitary matrix  $Q$  from the already computed QR factorization (see (8.7)). Notice that  $Q_2$  is compactly stored in

the lower trapezoidal part of the first `nrlinpar` columns of `svSpace` and in the vector `tau`, according to LAPACK's ZGEQRF routine conventions.

- The complex approximate Jacobian is transformed to real, by unfolding real and imaginary parts into odd and even rows of the matrix `fjacf`.

# Bibliography

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorenson. *LAPACK Users' Guide, 2nd Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995.
- [3] AqsesGUI: Software package for accurate quantification of short-echo time NMR signals. [www.esat.kuleuven.be/sista/members/biomed/new/](http://www.esat.kuleuven.be/sista/members/biomed/new/), 2005. K.U. Leuven, E.E. Dept. (ESAT-SISTA), Belgium.
- [4] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*. Software, Environments, and Tools. SIAM, Philadelphia, PA, 2000.
- [5] Z. Bai and R. W. Freund. A band symmetric Lanczos process based on coupled recurrences with applications. Technical report numerical analysis manuscript, Bell Laboratories, Murray Hill, NJ, USA, 1998.
- [6] A. Beck and A. Ben-Tal. On the solution of the Tikhonov regularization of the regularized total least squares problem. *SIAM Journal on Optimization*, 2006. To appear.
- [7] A. Beck, A. Ben-Tal, and M. Teboulle. Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. *SIAM Journal on Matrix Analysis and Applications*, 2006. To appear.
- [8] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT – a subroutine library in systems and control theory. *Applied and Computational Control, Signals and Circuits*, pages 499–539, 1998.
- [9] P. Bhattacharya and P.-L. Zhao. Semiparametric inference in a partial linear model. *The Annals of Statistics*, 25(1):244–262, 1997.
- [10] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA., 1996.

- [11] Å. Björck. A band-Lanczos generalization of bidiagonal decomposition. <http://www.mai.liu.se/~akbjo/umea05.pdf>, November 2005.
- [12] Å. Björck. Bidiagonal decomposition and least squares. <http://www.maths.anu.edu.au/events/sy2005/odataalks/canb05.pdf>, September 2005.
- [13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [14] H. Bozdogan. Akaike's information criterion and recent developments in information complexity. *Journal of Mathematical Psychology*, 44:61–91, 2000.
- [15] D. Calvetti and L. Reichel. Lanczos-based exponential filtering for discrete ill-posed problems. *Numerical Algorithms*, 29:45–65, 2002.
- [16] C. K. Carter and R. Kohn. Semiparametric Bayesian inference for time series with mixed spectra. *Journal of the Royal Statistical Society: Series B*, 59(1):255–268, 1997.
- [17] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.
- [18] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.
- [19] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.
- [20] A. Dax. On regularized least norm problems. *SIAM Journal on Optimization*, 2(4):602–618, 1992.
- [21] C. de Boor. *A Practical Guide to Splines*. Springer, Berlin, 1978.
- [22] B. De Moor. Structured total least squares and  $L_2$  approximation problems. *Linear Algebra and its Applications*, 188–189:163–207, 1993.
- [23] B. De Neuter, L. Vanhamme, L. P., and S. Van Huffel. Java-based framework for processing and displaying short-echo-time magnetic resonance spectroscopy signals. Technical Report 04-227, K.U. Leuven, E.E. Dept. (ESAT-SISTA), 2004.
- [24] Y.-H. De Roeck. Sparse linear algebra and geophysical migration: a review of direct and iterative methods. *Numerical Algorithms*, 29:283–322, 2002.
- [25] J. E. J. Dennis, D. M. Gay, and R. E. Welsch. Algorithm 573: NL2SOL – adaptive nonlinear least-squares algorithm [E4]. *ACM Trans. Math. Softw.*, 7(3):369–383, 1981.
- [26] S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in model selection and performance assessment. Technical Report Working Paper 126, U.C. Berkeley Division of Biostatistics, February 2003.

- [27] B. Efron. Selection criteria for scatterplot smoothers. *The Annals of Statistics*, 29(2):470–504, 2001.
- [28] P. H. C. Eilers and B. D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996.
- [29] L. Elden. Partial least squares vs. Lanczos bidiagonalization I: Analysis of a projection method for multiple regression. *Computational Statistics & Data Analysis*, 46:11–31, 2004.
- [30] C. Elster, F. Schubert, A. Link, M. Walzel, F. Seifert, and H. Rinneberg. Quantitative magnetic resonance spectroscopy: semiparametric modeling and determination of uncertainties. *Magnetic Resonance in Medicine*, 53(6):1288–1296, 2005.
- [31] R. L. Eubank. *Nonparametric Regression and Spline Smoothing*. Marcel Dekker, New York, 1999.
- [32] P. Feldmann and R. W. Freund. Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm. In *Proceedings of the 32nd Design Automation Conference*, New York, 1995. ACM.
- [33] R. D. Fierro, G. H. Golub, P. C. Hansen, and D. P. O’Leary. Regularization by truncated total least squares. *SIAM Journal on Scientific Computing*, 18(1):1223–1241, 1997.
- [34] D. Foresee and M. Hagan. Gauss-Newton approximation to Bayesian learning. In *Proceedings of the 1997 International Joint Conference on Neural Networks*, pages 1930–1935, 1997.
- [35] R. W. Freund and P. Feldmann. Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Padé approximation. In *Proceedings of the Design, Automation and Test in Europe Conference*, pages 530–537, Los Alamitos, CA, 1998. IEEE Computer Society Press.
- [36] J. H. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19(1), 1991.
- [37] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".
- [38] D. Gadian. *NMR and its applications to living systems*. Oxford Science publishers, 2nd edition, 1995.
- [39] W. Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36:291–307, 1981.
- [40] W. Gander, G. H. Golub, and U. von Matt. A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114/115:815–839, 1989.

- [41] D. M. Gay. A trust-region approach to linearly constrained optimization. In D. F. Griffiths, editor, *Numerical Analysis Proceedings*, pages 72–105, Dundee, 1983. Springer-Verlag.
- [42] G. H. Golub, P. C. Hansen, and D. P. O’Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.
- [43] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.
- [44] G. H. Golub, A. Hoffman, and G. W. Stewart. A generalization of the Eckhart-Young-Mirsky matrix approximation theorem. *Linear Algebra and its Applications*, 88/89:317–327, 1987.
- [45] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM Journal on Numerical Analysis*, 2(2):205–224, 1965.
- [46] G. H. Golub, F. T. Luk, and M. L. Overton. A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Transactions on Mathematical Software (TOMS)*, 7(2):149–169, 1981.
- [47] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10:413–432, 1973.
- [48] G. H. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19(2):1–26, 2003.
- [49] G. H. Golub and C. F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17:883–893, 1980.
- [50] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.
- [51] G. H. Golub and U. von Matt. Generalized cross-validation for large scale problems. In S. Van Huffel, editor, *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling*, pages 139–148. SIAM, 1997.
- [52] P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman and Hall, London, 1994.
- [53] H. Guo and R. Renaut. A regularized total least squares algorithm. In S. Van Huffel and P. Lemmerling, editors, *Total Least Squares and Errors-in-Variables Modeling*, pages 57–66. Kluwer, 2002.
- [54] E. Haber and L. Tenorio. Learning regularization functionals – a supervised training approach. *Inverse Problems*, 19(3):611–626, 2003.
- [55] P. Hall and J. D. Opsomer. Theory for penalised spline regression. *Biometrika*, 92:105–118, 2005.

- [56] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 32:561–580, 1992.
- [57] P. C. Hansen. Regularization Tools, a Matlab package for analysis of discrete regularization problems. *Numerical Algorithms*, 6:1–35, 1994.
- [58] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.
- [59] P. C. Hansen. Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29:323–378, 2002.
- [60] P. C. Hansen and D. P. O’Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computations*, 14:1487–1503, 1993.
- [61] W. Härdle. *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press, 1990.
- [62] O. P. Hasekamp and J. Landgraf. Ozone profile retrieval from backscattered ultraviolet radiances: The inverse problem solved by regularization. *Journal of Geophysical Research*, 106(D8):8077–8088, 2001.
- [63] J. C. Hoch and A. Stern. *NMR Data Processing*. John Wiley & Sons, 1996.
- [64] L. Hofmann, J. Slotboom, B. Jung, P. Maloca, A. Boesch, and R. Kreis. Quantitative  $^1\text{H}$ -magnetic resonance spectroscopy of human brain: Influence of composition and parameterization of the basis set in linear combination model-fitting. *Magnetic Resonance in Medicine*, 48:440–453, 2002.
- [65] C. M. Hurvich, J. S. Simonoff, and C.-L. Tsai. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society: Series B*, 60(2):271–293, 1998.
- [66] W. James and C. Stein. Estimation with quadratic loss. In J. Neyman, editor, *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics*, volume 1, pages 361–379, Univ. of California Press, Berkeley, 1961.
- [67] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT*, 15:49–57, 1975.
- [68] L. Kaufman and V. Pereyra. A method for separable nonlinear least squares problems with separable equality constraints. *SIAM Journal on Numerical Analysis*, 15:12–20, 1978.
- [69] C. Ke and Y. Wang. Smoothing spline nonlinear nonparametric regression models. *Journal of the American Statistical Association*, 99(468):1166–1175, 2004.
- [70] S. Konishi and G. Kitagawa. Generalised information criteria in model selection. *Biometrika*, 83(4):875–890, 1996.

- [71] F. Kuijt and R. van Damme. A linear approach to shape preserving spline approximation. *Advances in Computational Mathematics*, 14:25–48, 2001.
- [72] R. M. Larsen. *Lanczos bidiagonalization with partial reorthogonalization*. PhD thesis, Dept. Computer Science, University of Aarhus, DK-8000 Aarhus C, Denmark, October 1998.
- [73] T. Laudadio, N. Mastronardi, L. Vanhamme, P. Van Hecke, and S. Van Huffel. Improved Lanczos algorithms for blackbox MRS data quantitation. *Journal of Magnetic Resonance*, 157:292–297, 2002.
- [74] C. Lawson and R. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974. Republished by SIAM, Classics in Applied Mathematics Series 15, 1994.
- [75] T. C. M. Lee. Smoothing parameter selection for smoothing splines: A simulation study. *Computational Statistics and Data Analysis*, 42(1-2):139–148, 2003.
- [76] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM Publications, Philadelphia, PA, 1998.
- [77] P. Lemmerling. *Structured Total Least Squares: Analysis, algorithms and applications*. PhD thesis, K.U. Leuven, Electrical Engineering Department, 1999.
- [78] P. Lemmerling, N. Mastronardi, and S. Van Huffel. Fast algorithm for solving Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms*, 21:371–392, 2000.
- [79] P. Lemmerling, L. Vanhamme, H. J. A. in't Zandt, S. Van Huffel, and P. Van Hecke. Time-domain quantification of short-echo-time in-vivo proton MRS. *MAGMA*, 15:178–179, 2002.
- [80] R.-C. Li and Q. Ye. A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM Journal on Matrix Analysis and Applications*, 25(2):405–428, 2003.
- [81] D. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.
- [82] C. L. Mallows. Some comments on Cp. *Technometrics*, 15:661–667, 1973.
- [83] C. L. Mallows. More comments on Cp. *Technometrics*, 37:362–372, 1995.
- [84] E. Mammen, J. S. Marron, B. A. Turlach, and M. P. Wand. A general projection framework for constrained smoothing. *Statistical Science*, 16(3):232–248, 2001.
- [85] I. Markovsky, J. C. Willems, B. De Moor, and S. Van Huffel. *Exact and Approximate Modeling of Linear Systems: A Behavioral Approach*. SIAM, 2006.
- [86] I. Marshall, J. Higinbotham, S. Bruce, and A. Freise. Use of Voigt lineshape for quantification of *in vivo*  $^1\text{H}$  spectra. *Magnetic Resonance in Medicine*, 37:449–459, 1997.

- [87] N. Mastronardi, P. Lemmerling, and S. Van Huffel. Fast regularized structured total least squares problems for solving the basic deconvolution problem. *Numerical linear algebra with applications*, 12(2–3):201–209, 2005.
- [88] A. Melman. A unifying convergence analysis of second-order methods for secular equations. *Mathematics of Computation*, 66(217):333–344, 1997.
- [89] J. E. Moody. The *effective* number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In Moody, Hanson, and Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 847–854. Morgan Kaufmann, Palo Alto, 1992.
- [90] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis: Proceedings of the Biennial Conference held at Dundee, June 28-July 1, 1977 (Lecture Notes in Mathematics #630)*, pages 104–116. Springer Verlag, 1978.
- [91] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet. Math. Dokl.*, 7:414–417, 1966.
- [92] K. Opstad, M. Murphy, P. Wilkins, B. Bell, J. Griffiths, and F. Howe. Differentiation of metastases from high-grade gliomas using short echo time  $^1\text{H}$  spectroscopy. *Journal of Magnetic Resonance Imaging*, 20:187–192, 2004.
- [93] C. C. Paige and M. A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.
- [94] C. C. Paige and Z. Strakoš. Scaled total least squares fundamentals. *Numerische Mathematik*, 91:117–146, 2002.
- [95] C. C. Paige and Z. Strakoš. Core problems in linear algebraic systems. *SIAM Journal on Matrix Analysis and Applications*, 27(3):861–875, 2006.
- [96] P. Pels. *Analysis and Improvement of Quantification Algorithms for Magnetic Resonance Spectroscopy*. PhD thesis, Faculty of Engineering, K.U.Leuven, Leuven, Belgium, January 2005.
- [97] S. W. Provencher. Estimation of metabolite concentrations from localized *in vivo* proton NMR spectra. *Magnetic Resonance in Medicine*, 30(6), 1993.
- [98] A. Pruessner and D. P. O’Leary. Blind deconvolution using a regularized structured total least norm algorithm. Technical Report CS-TR-4287, University of Maryland, Computer Science Department, 2001.
- [99] A. E. Raftery. Bayesian model selection in social research. *Sociological Methodology*, 25:111–163, 1995.
- [100] H. Ratiney, Y. Coenradie, S. Cavassila, D. van Ormondt, and D. Graveron-Demilly. Time-domain quantitation of  $^1\text{H}$  short echo-time signals: Background accommodation. *MAGMA*, 16:284–296, 2004.

- [101] H. Ratiney, M. Sdika, Y. Coenradie, S. Cavassila, D. van Ormondt, and D. Graveron-Demilly. Time-domain semi-parametric estimation based on a metabolite basis set. *NMR in Biomedicine*, 17:1–13, 2004.
- [102] T. Reginska. A regularization parameter in discrete ill-posed problems. *SIAM J. Sci. Comput.*, 17(3):740–749, 1996.
- [103] R. Renaut and H. Guo. Efficient algorithms for solution of regularized total least squares. *SIAM Journal on Matrix Analysis and Applications*, 26(2):457–476, 2005.
- [104] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [105] A. Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Mathematics of Computation*, 1979.
- [106] A. Ruhe and P.-Å. Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22:318–337, 1980.
- [107] D. Ruppert. Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11:735–757, 2002.
- [108] D. Ruppert and R. J. Carroll. A simple roughness penalty approach to regression spline estimation. Technical Report 1167, School of OR&IE, Cornell University, August 1996.
- [109] D. Ruppert, M. P. Wand, and R. J. Carroll. *Semiparametric Regression*. Cambridge University Press, 2003.
- [110] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6, 1978.
- [111] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. Probability and Mathematical Statistics. John Wiley & Sons, 1989.
- [112] U. Seeger, U. Klose, I. Mader, W. Grodd, and T. Nagele. Parameterized evaluation of macromolecules and lipids in proton MR spectroscopy of brain diseases. *Magnetic Resonance in Medicine*, 49(1):19–28, 2003.
- [113] Y. Serinağaoğlu, D. H. Brooks, and R. S. MacLeod. Bayesian solutions and performance analysis in bioelectric inverse problems. *IEEE Transactions on Biomedical Engineering*, 52(6):1009–1020, 2005.
- [114] D. M. Sima and S. Van Huffel. Appropriate cross-validation for regularized errors-in-variables linear models. In *Proceedings of the COMPSTAT 2004 Symposium*, pages 1815–1822, Prague, Czech Republic, August 2004. Physica-Verlag/Springer.
- [115] D. M. Sima and S. Van Huffel. Regularized semiparametric model identification with application to NMR signal quantification with unknown macromolecular baseline. Technical Report 04-229, K.U. Leuven, E.E. Dept. (ESAT-SISTA), December 2004. To appear in *Journal of the Royal Statistical Society: Series B* 68(3).

- [116] D. M. Sima and S. Van Huffel. AQSES<sub>VP</sub> – description of a variable projection implementation for nonlinear least squares with linear bounds constraints, applied to accurate quantification of short-echo time magnetic resonance spectroscopic signals. Technical Report 05-120, K.U. Leuven, E.E. Dept. (ESAT-SISTA), 2005. Available from <ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/dsima/abstracts/05-120.html>.
- [117] D. M. Sima, S. Van Huffel, and G. H. Golub. Regularized total least squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, 44(4):793–812, December 2004.
- [118] H. D. Simmon and H. Zha. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(8):2257–2274, 2000.
- [119] A. Simonetti, J.-B. Poulet, D. M. Sima, B. De Neuter, L. Vanhamme, P. Lemmerling, and S. Van Huffel. An open source short echo time MR quantitation software solution: AQSES. Technical Report 05-168, K.U. Leuven, E.E. Dept. (ESAT-SISTA), 2005. Available from <ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/dsima/abstracts/05-168.html>.
- [120] J. C. Spall and J. P. Garner. Parameter identification of state-space models with nuisance parameters. *IEEE Transactions on Aerospace and Electronic Systems*, 26(6):992–998, 1990.
- [121] T. Sundin, L. Vanhamme, P. Van Hecke, I. Dologlou, and S. Van Huffel. Accurate quantification of <sup>1</sup>H spectra: from FIR filter design for solvent suppression to parameter estimation. *Journal of Magnetic Resonance*, 139:189–204, 1999.
- [122] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
- [123] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet. Math. Dokl.*, 4:1035–1038, 1963.
- [124] A. N. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston & Sons, Washington, DC, USA, 1977.
- [125] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001.
- [126] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.
- [127] <http://www.netlib.org/opt/varpro>.
- [128] A. M. Urmanov, A. V. Gribok, H. Bozdogan, J. W. Hines, and R. E. Uhrig. Information complexity-based regularization parameter selection for solution of ill conditioned inverse problems. *Inverse Problems*, 18, 2002. Institute of Physics Publishing.

- [129] J. W. C. van der Veen, R. de Beer, P. R. Luyten, and D. van Ormondt. Accurate quantification of *in vivo* PNMR signals using the variable projection method and prior knowledge. *Magnetic Resonance in Medicine*, 6:92–98, 1988.
- [130] S. Van Huffel and J. Vandewalle. The partial total least squares algorithm. *Journal of Computational and Applied Mathematics*, 21:333–342, 1988.
- [131] S. Van Huffel and J. Vandewalle. Analysis and properties of the generalized total least squares problem  $AX \approx B$  when some or all columns in  $A$  are subject to error. *SIAM Journal on Matrix Analysis and Applications*, 10(3):294–315, 1989.
- [132] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*, volume 9 of *Frontiers in Applied Mathematics*. SIAM, Philadelphia, 1991.
- [133] L. Vanhamme, T. Sundin, P. Van Hecke, and S. Van Huffel. MR spectroscopic quantitation: a review of time domain methods. *NMR in Biomedicine*, 14:233–246, 2001.
- [134] J. M. Varah. Pitfalls in the numerical solution of linear ill-posed problems. *SIAM Journal on Scientific and Statistical Computing*, 4(2):164–176, 1983.
- [135] E. M. Vestrup. *The Theory of Measures and Integration*. Wiley Series on Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 2003.
- [136] G. Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics 59. SIAM, 1990.
- [137] G. Wahba. An introduction to reproducing kernel Hilbert spaces and why are they so useful. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID 2003)*, Rotterdam, August 2003.
- [138] N. H. Younan and X. Fan. Signal restoration via the regularized constrained total least squares. *Signal Processing*, 71:85–93, 1998.

# Publication list

## Papers in international journals

1. **D. M. Sima** and S. Van Huffel. Separable nonlinear least squares fitting with linear bound constraints and its application in magnetic resonance spectroscopy data quantification. Technical Report 05-236, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2005. To appear in *Journal of Computational and Applied Mathematics*.
2. **D. M. Sima** and S. Van Huffel. Regularized semiparametric model identification with application to NMR signal quantification with unknown macromolecular baseline. Technical Report 04-229, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2004. To appear in *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(3).
3. **D. M. Sima** and S. Van Huffel. A class of template splines. Technical Report 04-228, K.U. Leuven, E.E. Dept. (ESAT-SISTA), 2004. To appear in *Computational Statistics & Data Analysis*.
4. **D. M. Sima**, S. Van Huffel, and G. H. Golub. Regularized total least squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, 44(4):793–812, December 2004.
5. V. Sima, **D. M. Sima** and S. Van Huffel. High-performance numerical algorithms and software for subspace-based linear multivariable system identification. *Journal of Computational and Applied Mathematics*, 170(2):371–397, Sep. 2004.
6. **D. M. Sima**. Nonlinear optimization with linear matrix inequalities constraints. *Studies in Informatics and Control*, 10(2):99–108, June 2001.

## Papers in proceedings of international conferences

7. **D. M. Sima** and S. Van Huffel. Using core formulations for ill-posed linear systems. In *PAMM, Proceedings of the 76th GAMM Annual Meeting, Luxembourg, March 28 – April 1, 2005*, 5(1):795–796, 2005.
8. **D. M. Sima** and S. Van Huffel. Appropriate cross-validation for regularized errors-in-variables linear models. In *Proceedings of the COMPSTAT 2004 Symposium*, pages 1815–1822, Prague, Czech Republic, August 2004. Physica-Verlag/Springer.

9. **D. M. Sima** and S. Van Huffel. Minor Component Analysis by incremental inverse iteration. Proceedings of the *Sixteenth International Symposium on Mathematical Theory of Networks and Systems, MTNS 2004*, July 5-9, 2004, K.U. Leuven, Belgium.
10. V. Sima, **D. M. Sima** and S. Van Huffel. SLICOT system identification software and applications. In Proceedings of the *IEEE International Symposium on Computer Aided Control System Design*, Sept. 2002, Glasgow, Scotland, U.K., pages 45–50.
11. **D. M. Sima**. Linear matrix inequalities for discrete-time linear systems. In Proceedings of the *7<sup>th</sup> International Symposium on Automatic Control and Computer Science*, Oct. 2001, Iași, Romania.
12. **D. M. Sima**. LMI Optimization in connection with quadratic and nonlinear functions. In Proceedings of the *Third Niconet Workshop on Numerical Software in Control and Engineering*, Jan. 2001, pages 91-96.

### Internal reports (including submitted papers)

13. **D. M. Sima** and S. Van Huffel. Level choice in truncation methods for linear models. Technical Report 06-51, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2006.
14. **D. M. Sima** and S. Van Huffel. Core problems in  $AX \approx B$ . Technical Report 06-50, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2006.
15. A. Simonetti, J.-B. Pouillet, **D. M. Sima**, B. De Neuter, L. Vanhamme, P. Lemmerling, and S. Van Huffel. An open source short echo time MR quantitation software solution: AQSES. Technical Report 05-168, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2005. Submitted to *NMR in Biomedicine*.
16. **D. M. Sima** and S. Van Huffel. AQSES<sub>VP</sub> – Description of a variable projection implementation for nonlinear least squares with linear bounds constraints, applied to accurate quantification of short-echo time magnetic resonance spectroscopic signals. Technical Report 05-120, K.U. Leuven, E.E. Dept. (ESAT-SISTA). 2005.

# Curriculum vitae

Diana M. Sima was born in Bucharest, Romania, on March 16, 1978. She studied at the Mathematics Department, University of Bucharest, and obtained in June 2000 a Bachelor degree in *Computer Science*. She finalized a Master in *Applied Statistics and Optimization* at the same department. The topics of her Bachelor and Master theses involved Linear Matrix Inequalities in optimization problems and applications in systems and control.

In February 2002, she started PhD research at ESAT-SCD-SISTA, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium.