

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

WEIGHTED LOW RANK APPROXIMATION: ALGORITHMS AND APPLICATIONS

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

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Mieke Schuermans

April 2006



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

WEIGHTED LOW RANK APPROXIMATION: ALGORITHMS AND APPLICATIONS

Jury:
Prof. dr. ir. P. Van Houtte, voorzitter
Prof. dr. ir. S. Van Huffel, promotor
Prof. dr. ir. J. Vandewalle
Prof. dr. ir. B. De Moor
Prof. dr. ir. M. Van Barel
Prof. dr. A. Cuyt (UA)
Prof. dr. ir. L. De Lathauwer (ETIS,Cergy-Pontoise)
Dr. N. Mastronardi (IAC-CNR, Bari)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Mieke Schuermans

U.D.C. 681.3*G12 April 2006

An engineer thinks that his equations are an approximation to reality. A physicist thinks reality is an approximation to his equations. A mathematician doesn't care.

© Katholieke Universiteit Leuven – Faculteit Ingenieurswetenschappen
Arenbergkasteel, B-3001 Heverlee (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/2006/7515/18

ISBN 90-5682-684-0

Voorwoord

Aan het einde van mijn doctoraat zou ik graag een aantal mensen willen bedanken. Dankzij hun steun en hulp is het mogelijk om dit werk voor te stellen.

In de eerste plaats wil ik uiteraard mijn promotor, Prof. Sabine Van Huffel, bedanken. Sabine, je onuitputtelijke bron van energie en je blijvend enthousiasme hebben me dikwijls geholpen om verder te blijven doen. Ik wil je bedanken voor de kans die je me gegeven hebt, de tijd die je steeds vrijmaakte voor me en de aanmoedigingen wanneer ik ze nodig had. Je hebt ervoor gezorgd dat Biomed is uitgegroeid tot een transparante, interactieve onderzoeksgroep. Bedankt om me de kans te hebben gegeven om deel uit te maken van zo'n stimulerende werkomgeving.

Prof. Joos Vandewalle zou ik willen bedanken omdat hij me enkele jaren geleden gestimuleerd heeft om aan een doctoraat te beginnen. Joos, heel fel bedankt voor de fijne tijd die ik als monitor van algebra gehad heb. Het heeft me deugd gedaan dat ik, naast onderzoekswerk, me ook heb kunnen uitleven in het geven van oefenzittingen. Mijn taken als monitor waren geregeld welgekomen afleidingen van knelpunten tijdens mijn onderzoek.

Prof. Marc Smet zou ik willen bedanken voor de fijne tijd die ik kon doorbrengen op het monitoraat. Marc, bedankt dat je me de vrijheid hebt gegeven om veel tijd te spenderen aan mijn onderzoek. Het is mede door jouw flexibiliteit dat ik binnen een redelijke tijdspanne dit werk heb kunnen voltooien.

Naast Prof. Joos Vandewalle zou ik ook graag Prof. Bart De Moor en Prof. Marc Van Barel willen bedanken voor het beoordelen van mijn proefschrift en het geven van waardevolle opmerkingen en suggesties.

Mijn dank gaat ook uit naar de voorzitter van de jury Prof. Paul Van Houtte, alsook naar de overige leden van de jury, Prof. Annie Cuyt, Prof. Lieven De Lathauwer en dr. Nicola Mastronardi, voor hun bereidwillige medewerking.

Het is een plezier om dr. Philippe Lemmerling te kunnen bedanken voor zijn uitstekende begeleiding. Bedankt, Philippe, voor je vele ideeën, je constante

paraatheid om op mijn vragen te antwoorden en je hulp bij het zoeken naar fouten in de algoritmen. Ook Prof. Lieven De Lathauwer zou ik willen bedanken om me wat wijzer te maken in de wereld van tensoren.

Ivan en Diana, bedankt voor jullie uitleg over talloze algoritmen en voor jullie hulp bij mijn computerperikelen. Onze gezamenlijke koffiepauzen droegen bij tot een aangename sfeer in ons bureau.

Voorts wil ik de vele SCD-leden bedanken voor de fijne werksfeer en dankzij Ida, Pela, Ilse en Lut werden alle administratieve en financiële zaken vlot afgehandeld. Bedankt hiervoor.

Naast het ESAT was ik ook soms te vinden op het monitoraat. Bedankt, collega's monitoren, voor de ontspannen werksfeer en de aangename babbel.

Uiteraard mogen ook mijn vrienden en vriendinnen niet ontbreken in dit woordje van dank. Jullie hebben ervoor gezorgd dat er naast onderzoek ook nog plaats was voor ontspannende momenten en die had ik nodig.

Als laatsten zou ik mijn ouders, mijn broer Wim en mijn vriend Jan willen bedanken. Jullie zijn de allerbelangrijkste mensen in mijn leven en hebben ervoor gezorgd dat ik sta waar ik nu sta. Een dikke kus voor alle vier!

Abstract

In order to find more sophisticated trends in data, potential correlations between larger and larger groups of variables must be considered. Unfortunately, the number of such correlations generally increases exponentially with the number of input variables and, as a result, brute force approaches become unfeasible. So, the data needs to be simplified sufficiently. Yet, the data may not be oversimplified. A method that is widely used for this purpose is to first cast the data as a matrix and then compute a low rank matrix approximation.

The tight equivalences between the Weighted Low Rank Approximation(WLRA) problem and the Total Least Squares (TLS) problem are explored. Despite the seemingly different problem formulations of WLRA and TLS, it is shown that both methods can be reduced to the same mathematical kernel problem, i.e. finding the closest (in a certain sense) weighted low rank matrix approximation where the weight is derived from the distribution of the errors in the data. Different solution approaches, as used in WLRA and TLS, are discussed. In particular, we discuss the Null space parameterized WLRA (NullWLRA), the Maximum Likelihood Principal Component Analysis (MLPCA), the Element-wise Weighted-TLS (EW-TLS) and the Generalized TLS (GTLS) methods. It is shown that these four approaches tackle an equivalent weighted low rank approximation problem, but different algorithms are used to come up with the best approximation matrix.

The WLRA problem is studied in the application field of chemometrics. In chemometrics, existing approaches to come up with the best weighted low rank matrix approximation are the well-known Principal Component Analysis (PCA) method and the MLPCA method. The use of TLS-like algorithms in chemometrics is discussed. An adapted version of the EW-TLS algorithm is developed to solve the WLRA problem for data matrices in chemometrics with more columns than rows and with only row-wise correlated measurement errors. It is shown that this new algorithm is a good alternative for the existing methods used in chemometrics.

The WLRA approach is extended towards linearly structured matrices. The

Hankel structure is investigated since this is one of the most frequently occurring structures in signal processing applications. We study the cases of scalar Hankel matrices and block-row Hankel matrices. For both cases, new algorithms are presented in order to solve these structured WLRA problems. These algorithms can handle structured WLRA problems with rank reductions larger than one, while most existing algorithms from the literature can only handle rank-one reduction problems. By means of simulation experiments the improved statistical accuracy of the proposed algorithms compared to known algorithms from the literature is confirmed.

The structured WLRA problem is studied in the field of recovering the vertices of a planar polygon from its measured complex moments. In the literature, the use of the structured TLS approach to solve this shape-from-moments problem has not been discussed. Therefore, the potential and limitations of the structured TLS algorithm and existing algorithms to solve this reconstruction problem are discussed.

Korte Inhoud

Om verfijndere tendensen in gegevens te vinden, moet er nagedacht worden over belangrijke correlaties tussen steeds grotere groepen variabelen. Jammer genoeg stijgt het aantal potentiële correlaties over het algemeen exponentieel met het aantal ingevoerde variabelen. Bijgevolg worden brutekrachtmethoden onbruikbaar. De gegevens moeten dus voldoende worden vereenvoudigd. Nochtans mogen de gegevens niet te eenvoudig voorgesteld worden of, anders gezegd, informatie die essentieel is om belangrijke correlaties te kunnen achterhalen, moet weerhouden worden. Een methode die hier wereldwijd voor gebruikt wordt, is de methode waarbij de gegevens eerst in een matrix gegoten worden om vervolgens een lagere rangbenadering ervan te zoeken.

De equivalentie tussen het Gewogen Lagere Rangbenaderingsprobleem (GLRB) enerzijds en het Totale Kleinste Kwadraten (TKK)-probleem anderzijds wordt onderzocht. Ondanks de schijnbaar verschillende probleemformuleringen van het GLRB-probleem en het TKK-probleem, wordt er aangetoond dat beide methoden tot hetzelfde wiskundig kernprobleem herleid kunnen worden, namelijk het vinden van de dichtst mogelijke gewogen lagere rangbenadering waarbij de wegingsfactoren afgeleid worden uit de statistische verdelingen van de meetfouten. De verschillende oplossingsmethoden, zoals die gebruikt worden voor de GLRB-problemen en de TKK-problemen, worden besproken. Wij bespreken in het bijzonder de nulruimte-geparametrizeerde GLRB (NullGLRB)-methode, de Maximum Likelihood Principaal Component Analyse (MLPCA), de element-gewijze gewogen TKK (EG-TKK)-methode en de veralgemeende TKK (VTKK)-methode. Er wordt aangetoond dat deze vier methoden een equivalent gewogen lagere rangbenaderingsprobleem aanpakken, maar dat ze verschillende algoritmen gebruiken om tot een goede benaderingsmatrix te komen.

Het GLRB-probleem wordt in het toepassingsgebied van de chemometrie bestudeerd. In de chemometrie worden er reeds welgekende methoden, zoals de (ML)PCA, gebruikt om een goede lagere rangbenaderingsmatrix te bepalen. Het nut van TKK-algoritmen voor problemen uit de chemometrie wordt besproken. Er wordt een nieuwe versie ontwikkeld van de reeds bestaande EG-

TKK-methode om lagere rangbenaderingsproblemen uit de chemometrie op te lossen voor matrices met meer kolommen dan rijen en waarvoor de meetfouten enkel gecorreleerd zijn binnen de rijen. Er wordt aangetoond dat dit nieuwe algoritme een goed alternatief is voor de reeds bestaande methoden die op dit moment in de chemometrie gebruikt worden.

Het GLRB-probleem wordt uitgebreid voor lineair gestructureerde matrices. We bestuderen de Hankel-structuur omdat deze structuur één van de meest voorkomende structuren in signaalverwerking is. Wij bestuderen de scalaire Hankel-structuur en de blok-rij Hankel-structuur. Voor beide structuren worden er nieuwe algoritmen voorgesteld om deze gestructureerde GLRB-problemen op te lossen. Deze algoritmen kunnen gestructureerde GLRB-problemen oplossen waarbij de rang gereduceerd kan worden met meer dan één. De meeste algoritmen uit de literatuur kunnen slechts GLRB-problemen oplossen waarbij de rang slechts met eentje verlaagd wordt. Aan de hand van simulatievoorbeelden wordt er aangetoond dat de ontwikkelde algoritmen een hogere statistische nauwkeurigheid hebben dan de algoritmen uit de literatuur.

Het gestructureerde GLRB-probleem wordt bestudeerd om de hoekpunten van een vlakke veelhoek te kunnen reconstrueren aan de hand van zijn gegeven complexe momenten. De gestructureerde TKK-methode is tot nu toe niet aan bod gekomen in de literatuur als mogelijke methode om dit reconstructieprobleem aan te pakken. Wij bespreken de mogelijkheden en beperkingen van deze alternatieve TKK-oplossingsmethode in vergelijking met methoden die wel in de literatuur besproken worden om het reconstructieprobleem op te lossen.

Glossary

The current section lists the symbols and acronyms that occur frequently in this thesis.

List of symbols

$A^\top$	transpose of A
A^H	Hermitean transpose of A
A^{-1}	inverse of A
$A^\dagger$	Moore-Penrose pseudo-inverse of A
a_{ij}	the entry in the jth column of the ith row of A
$A \otimes B$	Kronecker product of A and B, defined as $[a_{ij}B]_{i,j}$
$\mathbb{R}^{n \times m}$	set of $n \times m$ matrices
$\mathbb{R}^n$	set of n dimensional vectors
$\mathbb{C}^{n \times m}$	set of $n \times m$ complex matrices
$\mathbb{C}^n$	set of n dimensional complex vectors
i.i.d.	independently and identically distributed
I_n	$n \times n$ identity matrix
$vec(A)$	column wise vectorization of a matrix A
$vec_2(A)$	minimal vector representation of a linearly structured matrix A

List of acronyms

ALS	Alternating Least Squares
EIV	Errors-In-Variables
EVD	Eigen Value Decomposition
EWTLS	Element-wise Weighted Total Least Squares
GPOF	Generalized Pencil Of Function
HOSVD	Higher Order Singular Value Decomposition

HOVDMD	Higher Order Vandermonde Decomposition
HTLS	Hankel Total Least Squares
LRA	Low Rank Approximation
LRR	Latent Root Regression
LS	Least Squares
LTI	Linear Time Invariant
MIMO	Multiple-Input Multiple-Output
MLLRR	Maximum Likelihood Latent Root Regression
MLPCA	Maximum Likelihood Principal Component Analysis
MLPCR	Maximum Likelihood Principal Component Regression
MRS	Magnetic Resonance Spectroscopy
NLLS	Non Linear Least Squares
NullWLRA	Null space parameterized WLRA
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLS	Partial Least Squares
QP	Quadratic Programming
RMSE	Root Mean Square Error
STLN	Structured Total Least Norm
STLS	Structured Total Least Squares
SVD	Singular Value Decomposition
SWLRA	Structured Weighted Low Rank Approximation
TLS	Total Least Squares
VDMD	Vandermonde Decomposition
WLRA	Weighted Low Rank Approximation

Contents

Voorwoord **i**

Abstract **iii**

Korte Inhoud **v**

Glossary **vii**

Contents **ix**

Nederlandse samenvatting **xiii**

1 Introduction **1**

1.1 Introduction 1

1.2 WLRA in applications 5

1.2.1 Chemometrics 5

1.2.2 MRS data quantification 7

1.2.3 Speech enhancement 10

1.2.4 Web search 11

1.3 Chapter by chapter overview 12

1.4 Conclusion 15

2 Introduction to Weighted Low Rank Approximation 17

2.1 Problem formulation 18

2.2 Link with Total Least Squares 18

2.2.1 TLS: problem formulation, algorithms and extensions . 18

2.2.2 Link between the WLRA and the TLS problem 23

2.3 Different approaches for the WLRA problem 23

2.4 Different algorithms for the WLRA problem 26

2.4.1 NullWLRA algorithm 27

2.4.2 EW-TLS algorithm of Premoli–Rastello 30

2.4.3 Classical EW-TLS algorithm 33

2.4.4 Adapted EW-TLS algorithm 35

2.4.5 GTLS algorithm 37

2.5 Conclusion 37

3 WLRA problem in chemometrics 41

3.1 Existing approaches 41

3.1.1 PCA/PCR 42

3.1.2 MLPCA 44

3.2 TLS in chemometrics: state-of-the-art 49

3.3 The use of TLS and extensions in multivariate (ML)PCR . . . 50

3.4 Performance comparison of MLPCA versus EW-TLS 51

3.4.1 Uncorrelated measurement errors 51

3.4.2 Row-wise correlated measurement errors 54

3.5 Conclusion 60

4 Structured WLRA problem 63

4.1 Introduction 63

4.2 Problem formulation 64
4.3 SWLRA problem for scalar Hankel matrices 66
4.3.1 An equivalent unconstrained optimization problem . . . 66
4.3.2 Algorithm 67
4.3.3 Numerical experiments 70
4.4 SWLRA problem for block-row Hankel matrices 74
4.4.1 An equivalent unconstrained optimization problem . . . 75
4.4.2 Algorithm 78
4.4.3 Numerical results 79
4.5 Conclusion 83

5 The shape from moments problem 85
5.1 Introduction 85
5.2 Problem formulation 86
5.3 Methods to solve the shape-from-moments problem 88
5.3.1 GPOF 88
5.3.2 HTLS 90
5.3.3 STLS 92
5.3.4 The tensor approach 95
5.4 Related work 102
5.5 Performance comparison between the methods 104
5.6 Conclusion 109

6 Conclusions and further research 111
6.1 Conclusions 111
6.2 Further research 112

Appendix 115

A Proof of Theorem 3 115
B Proof of Theorem 4 117
C Mathematical background to PCA 118
D STLNB algorithm 121
E HTLSstack algorithm 124

Bibliography **129**

List of publications **139**

Curriculum Vitae **141**

Nederlandse samenvatting

Gewogen lagererang-benaderingsproblemen: algoritmen en toepassingen

Hoofdstuk 1: Inleiding

Om verfijndere tendensen in gegevens te vinden, moet er nagedacht worden over belangrijke correlaties tussen steeds grotere groepen variabelen. Jammer genoeg stijgt het aantal potentiële correlaties over het algemeen exponentieel met het aantal ingevoerde variabelen. Bijgevolg worden brutekrachtmethoden onbruikbaar. De gegevens moeten dus voldoende worden vereenvoudigd. Nochtans mogen de gegevens niet te eenvoudig voorgesteld worden of, anders gezegd, informatie die essentieel is om belangrijke correlaties te kunnen achterhalen, moet weerhouden worden. Een methode die hier wereldwijd voor gebruikt wordt, is de methode waarbij de gegevens eerst in een matrix gegoten worden om vervolgens een lagererangbenadering (LRB) ervan te zoeken.

De exacte definities van lagererangbenaderingsproblemen worden geformuleerd en er wordt ook een overzicht gegeven van de verschillende oplossingsmethoden die door de jaren heen gepubliceerd werden. De voor- en nadelen van de bestaande methoden worden besproken en nieuwe, verbeterde methoden worden voorgesteld.

Naast het lagererangbenaderingsprobleem worden er ook uitbreidingen naar gewogen LRB (GLRB)-problemen en gestructureerd GLRB (GGLRB)-problemen geformuleerd. Het LRB-probleem bestaat erin om voor een gegeven $m \times n$ da-

tamatrix D van bepaalde rang k een zo dicht mogelijke lagererangbenaderingsmatrix $\widehat{D}$ te zoeken die dezelfde dimensies als D heeft, maar van lagere rang $r < k$ is. Wiskundig vertaalt zich dat dan in het volgende:

$$\min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n} \\ rang\,\widehat{D}\leq r<k}} \| D - \widehat{D} \|_F^2,$$

waarbij $\| \cdot \|_F$ de Frobenius-norm voorstelt en voor een willekeurige $s \times t$ matrix C gedefinieerd wordt als $\| C \|_F^2 = \sum_{i=1}^{s} \sum_{j=1}^{t} c_{ij}^2$.

Een nadeel van de Frobenius-norm in bovenstaande probleemformulering is dat alle elementen van de datamatrix D evenwichtig in rekening worden gebracht. Nochtans kan het in toepassingen belangrijk zijn om een onderscheid te maken tussen belangrijke en minder belangrijke elementen uit de datamatrix. In het GLRB-probleem kan er een verschillend gewicht aan elk element worden meegegeven. Het GLRB-probleem bestaat er dan ook in om een zo goed mogelijke lagererangbenaderingsmatrix te zoeken voor een gegeven datamatrix ten opzichte van een welbepaalde gewogen norm. Door zo een gewogen norm te gebruiken, kunnen belangrijke elementen uit de datamatrix meer in rekening worden gebracht dan andere door hen een hogere wegingsfactor toe te kennen. Er wordt getoond hoe het GLRB-probleem in feite een veralgemening is van het LRB-probleem.

In veel toepassingen op het gebied van signaalverwerking, zoals magnetische resonantiespectroscopie en spraakcompressie, hebben de bijhorende datamatrices een specifieke lineaire structuur. Voor deze toepassingen is het niet alleen belangrijk om het LRB-probleem uit te breiden door wegingsfactoren in rekening te brengen, maar door bovendien ook rekening te houden met de specifieke structuur van de gegeven datamatrix. Het GGLRB-probleem houdt zowel rekening met de structuur van de matrix als met het onderscheid in gewicht van zijn afzonderlijke elementen.

In het verleden werden er reeds veel methoden gepubliceerd om LRB-problemen op te lossen voor lagererangbenaderingen waarbij de rang slechts met één gereduceerd werd. LRB-problemen waarbij er gezocht wordt naar hogere rangreducties komen echter niet zoveel voor. Naast een overzicht van bestaande oplossingsmethoden, worden er ook nieuwe methoden voorgesteld om G(G)LRB-problemen aan te pakken waarbij de rangverlaging $\Delta rang \equiv k - r$ groter dan één kan zijn.

Tevens worden er vier toepassingsgebieden vermeld om een idee te geven over de vele toepassingen die er zijn waarin GLRB-problemen voorkomen. Zo worden chemometrie, magnetische resonantiespectroscopie, spraakversterking en websurfen in het kort besproken.

Hoofdstuk 2: Inleiding tot Gewogen-Lagere-Rang-benaderingsproblemen

Zoals reeds in het voorgaande hoofdstuk vermeld werd, is het GLRB-probleem een uitbreiding van het LRB-probleem. Het GLRB-probleem wordt wiskundig als volgt voorgesteld:

$$\min_{\widehat{D}\in\mathbb{R}^{m\times n}} vec^{\top}(D-\widehat{D})W^{-1}vec(D-\widehat{D}) \quad \text{zodat rang } (\widehat{D}) \leq r, \tag{0.1}$$

waarbij $D \in \mathbb{R}^{m\times n}, m \geq n$ de ruizige datamatrix is van rang k met $r < k$, $vec(M)$ de gevectorizeerde vorm van een willekeurige matrix M die verkregen wordt door de opeenvolgende kolommen van M onder elkaar in een vector te plaatsen en waarbij W een positief definiete, symmetrische wegingsmatrix is. Als we het speciale geval $W = I$ bekijken, waarbij I de eenheidsmatrix voorstelt, dan komt het hierboven beschreven probleem overeen met het LRB-probleem.

In dit hoofdstuk onderzoeken we de gelijkenissen tussen het GLRB-probleem en het Totale-Kleinste-Kwadraten (TKK)-probleem. Hiertoe bespreken we eerst in het kort het TKK-probleem en herleiden de TKK-probleemformulering tot een GLRB-probleem. Het TKK-probleem bestaat erin om volgend overgedetermineerd stelsel van vergelijkingen (er wordt verondersteld dat $m > n$) consistent te maken:

$$XB \approx Y,$$

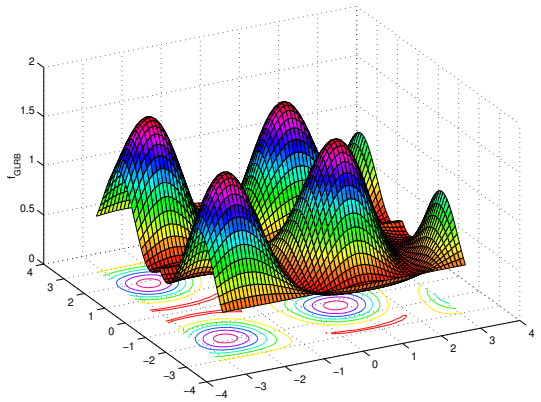
waarbij $[X\, Y]$ de gegeven $m\times n$ datamatrix is met $X \in \mathbb{R}^{m\times(n-d)}$ en $Y \in \mathbb{R}^{m\times d}$, en B de te schatten parametermatrix. Het TKK-probleem zoekt dus naar een oplossing voor

$$\min_{\widehat{\Delta X},\widehat{\Delta Y},\widehat{B}} \| [\widehat{\Delta X}\ \widehat{\Delta Y}] \|_F \quad \text{zodat } (X-\widehat{\Delta X})\widehat{B} = Y - \widehat{\Delta Y}. \tag{0.2}$$

$\widehat{B}$ wordt de TKK-oplossing genoemd en $[\widehat{\Delta X}\ \widehat{\Delta Y}]$ is de bijhorende TKK-correctie. Van bovenstaand TKK-probleem worden er veralgemeningen vermeld waarin er andere normen aan bod komen dan de Frobenius norm.

In plaats van een beste gewogen lagererangbenaderingsmatrix $\widehat{D}$ van rang r voor een gegeven datamatrix $D \in \mathbb{R}^{m\times n}$ te zoeken, kan men dus ook een TKK-oplossing zoeken voor het stelsel $XB \approx Y$, waarbij de datamatrix D opgesplitst wordt in twee delen $D = [X\, Y]$ zodat $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{m\times(n-r)}$. Als $W = I$, dan kan formulering (0.2) gebruikt worden en is de beste lagererangbenaderingsmatrix van rang r gelijk aan $\widehat{D} = [X - \widehat{\Delta X}\ Y - \widehat{\Delta Y}]$. Voor een meer algemene wegingsmatrix W moet er een veralgemeende versie van het TKK-probleem gebruikt worden.

Hoewel het GLRB-probleem (0.1) en het TKK-probleem (0.2) op een verschillende manier geformuleerd worden, tonen we aan dat beide problemen herleid



Figuur 0.1: Functiewaarden van de kostfunctie f_{GLRB} die in het GLRB-probleem geminimalizeerd moet worden voor een 3×3 datamatrix, een diagonale wegingsmatrix en $r = 2$.

kunnen worden tot eenzelfde wiskundig kernprobleem, namelijk het vinden van een zo goed mogelijke gewogen lagererangbenadering waarbij de wegingsfactoren bepaald worden door de statistische verdelingen van de meetfouten van de gegeven data.

Omdat het zoeken naar een gewogen lagererangbenadering geen convex probleem is, bestaan er geen efficiënte methoden waarvan we zeker kunnen zijn dat ze naar een globale, optimale oplossing convergeren. In plaats van globale convergentie bestuderen we dan ook lokale convergentie in de buurt van een startwaarde. In Figuur 0.1 worden de functiewaarden gegeven van de kostfunctie uit (0.1) voor een klein voorbeeldje waarbij $D \in \mathbb{R}^{3\times 3}$, W een diagonaalmatrix is en $r = 2$. Het valt meteen op dat er meerdere lokale minima zijn.

Door aan te tonen dat het GLRB- en het TKK-probleem nauw verwant zijn met elkaar, kunnen we hun verschillende oplossingsmethoden met elkaar vergelijken. We geven een overzicht van bestaande methoden om het gewogen lagererangbenaderingsprobleem op te lossen en stellen een nieuwe oplossingsmethode voor. De methoden verschillen elk in de manier waarop ze de parametervoorstelling van de rangbeperking $\text{rang}(\widehat{D}) \leq r$ opstellen, alsook in de techniek die ze gebruiken om het resulterende geparameterizeerde optimalizatieprobleem op te lossen. Hoewel de methoden een equivalent gewogen lagererangbenaderingsprobleem aanpakken, maken ze gebruik van verschillende algoritmen om uiteindelijk een goede benaderingsmatrix te bepalen.

Hoofdstuk 3: GLRB-problemen in chemometrie

Dit hoofdstuk begint met een overzicht van 2 vaak gebruikte algoritmen voor het oplossen van GLRB-problemen in chemometrie: het PCA-algoritme, Algoritme 7, en het MLPCA-algoritme, Algoritme 9.

In chemometrie wordt de PCA-methode gebruikt om bepaalde relaties tussen de multivariate variabelen te beschrijven, vooral in gevallen waar er een hoge graad van collineariteit onder de variabelen bestaat. In deze context is het voordeel van de PCA-methode dat multivariate data door een kleiner aantal variabelen kunnen voorgesteld worden die de hoofdcomponenten genoemd worden. In toepassingen zoals mengselanalyse, is het bijvoorbeeld belangrijk om een lager-dimensionaal lineair model te ontwikkelen waarbinnen de gegevens kunnen beschreven worden op een experimentele fout na.

PCA kan worden beschouwd als een Maximum-Likelihoodsmethode wanneer alle meetfouten onafhankelijk zijn van elkaar en dezelfde normaalverdeling hebben. Wanneer er kleine afwijkingen worden waargenomen van de veronderstellingen voor Maximum-Likelihoodsschatting, kan PCA nog nuttig zijn, maar wanneer de afwijkingen te groot worden, wordt de methode nutteloos. Het is in dit opzicht dat de Maximum Likelihood PCA (MLPCA)-methode ontwikkeld werd. MLPCA houdt rekening met de statistische verdeling van de meetfouten bij het schatten van de parameters.

Om gelijkaardige redenen werd de TKK-methode ontwikkeld in het gebied van de computerwetenschappen om consistente parameterschatters te bepalen van lineaire modellen waarbij de verdeling van de meetfouten gekend is.

In dit hoofdstuk worden algoritmen die gebruikt worden om TKK-problemen op te lossen, vergeleken met algoritmen die in de chemometrie gebruikt worden om MLPC's te bepalen. Hiervoor gebruiken we simulaties die data bevatten waarbij de meetfouten zowel ongecorreleerd kunnen zijn als waarbij er wel correlaties tussen de meetfouten kunnen optreden binnen de kolommen van de datamatrix of enkel binnen de rijen ervan. Aan de hand van die simulaties laten we zien dat de TKK-algoritmen nuttig zijn om te gebruiken bij chemometrische GLRB-problemen.

Hoofdstuk 4: Gestructureerde GLRB-problemen

In veel toepassingen op het gebied van signaalverwerking, zoals magnetische resonantiespectroscopie en spraakcompressie, hebben de bijhorende datamatrices een specifieke lineaire structuur. Het GLRB-probleem houdt echter geen rekening met de eventuele structuur van de gegeven datamatrix.

In dit hoofdstuk breiden we het GLRB-probleem dan ook uit naar een Gestructureerd GLRB (GGLRB)-probleem. We beschouwen lineair gestructureerde datamatrices. Het GGLRB-probleem bestaat erin om voor een gegeven gestructureerde datamatrix D van rang k een lagererangbenaderingsmatrix $\widehat{D}$ te zoeken van rang $r < k$ die dezelfde structuur heeft als de originele datamatrix D. Om $\widehat{D}$ te bepalen, dient het volgende probleem opgelost te worden:

$$\min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n}\\ rang(\widehat{D})\leq r<k\\ \widehat{D}\in\Omega}} \| D-\widehat{D} \|_W^2, \tag{0.3}$$

waarbij $D \in \mathbb{R}^{m\times n}$, rang$(D)\equiv k$, $m \geq n$ (als $m < n$, verander D dan door $D^\top$), Ω is de verzameling van alle matrices met dezelfde structuur als D en $\|C\|_W^2 \equiv vec(C)^\top W vec(C)$ waarbij $vec(C)$ de vector is die bekomen wordt door de opeenvolgende kolommen van C boven elkaar te plaatsen en W een positief definiete, symmetrische wegingsmatrix is. Voor specifieke keuzen van W en Ω herleidt het bovenstaande probleem (0.3) zich tot eerder besproken problemen: voor $\Omega = \mathbb{R}^{m\times n}$ herleidt probleem (0.3) zich tot het ongestructureerde GLRB-probleem en als bovendien W gelijk is aan de eenheidsmatrix I, komt probleem (0.3) overeen met het LRB-probleem.

Statistisch gezien, is het zinvol om elementen uit de datamatrix die gelijk zijn aan elkaar, ook op eenzelfde manier te behandelen. Voor een Hankel-matrix, bijvoorbeeld, moeten elementen uit dezelfde antidiagonaal gelijkwaardig behandeld worden. Hiervoor dient probleem (0.3) geherformuleerd te worden:

$$\min_{\substack{vec_2(\widehat{D})\\ rang(\widehat{D})\leq r<k}} \|D-\widehat{D}\|_V^2, \tag{0.4}$$

waarbij $\|C\|_V^2 \equiv vec_2(C)^\top V vec_2(C)$ met $vec_2(C)$ de minimale vectorvoorstelling van een lineair gestructureerde matrix C en waarbij V een positief definiete, symmetrische matrix is. Bemerk dat de voorwaarde $\widehat{D} \in \Omega$ niet langer voorkomt in (0.4) wegens de eenduidige relatie tussen C en $vec_2(C)$. Probleem (0.4) wordt het GGLRB-probleem genoemd.

Als eerste stap in de richting van een oplossingsmethode voor het GGLRB-probleem wordt probleem (0.4) in het volgende equivalent dubbel-minimalizatieprobleem geherformuleerd:

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^\top N=I}} \left(\min_{\substack{vec_2(\widehat{D})\\ \widehat{D}N=0}} \|D-\widehat{D}\|_V^2 \right). \tag{0.5}$$

In een tweede stap wordt er gezocht naar een uitdrukking voor de oplossing van het binnenste minimalizatieprobleem $f(N) \equiv \min_{\substack{vec_2(\widehat{D})\\ \widehat{D}N=0}} \|D-\widehat{D}\|_V^2$. Hiertoe wordt het binnenste minimalizatieprobleem in woorden geformuleerd: "Zoek voor een gegeven matrix N waarvan de kolomruimte de nulruimte van een

niet-triviale gestructureerde matrix is, de matrix $\widehat{D}$ die het dichtste bij matrix D ligt en dezelfde structuur als D heeft zodat de nulruimte van $\widehat{D}$ opgespannen wordt door de kolomruimte van N." Uit deze formulering in woorden blijkt dat er gezocht moet worden naar een parameterizatie van de nulruimte van een gestructureerde matrix. Omdat deze parameterizatie afhangt van de structuur van de matrix, kan er geen algemeen geldend algoritme gevonden worden dat toegepast kan worden op eender welke lineair gestructureerde matrix D. Vandaar dat we ons in dit hoofdstuk beperken tot de Hankel-structuur. We ontwikkelen algoritmen om het GGLRB-probleem op te lossen voor scalaire Hankel-matrices en blok Hankel-matrices waarvan de blokken uit rijvectoren bestaan. Deze algoritmen worden ontwikkeld door eerst een parameterizatie op te stellen voor de gegeven structuur alvorens een uitdrukking te bepalen voor de kostfunctie $f(N)$ die geminimalizeerd dient te worden.

De ontwikkelde algoritmen worden vergeleken met bestaande algoritmen die in de literatuur al eerder werden voorgesteld om Hankel-gestructureerde GLRB-problemen op te lossen. Voor het scalair Hankel-gestructureerde geval vergelijken we de statistische nauwkeurigheid en de numerieke efficiëntie van het nieuw ontwikkelde algoritme met het bestaande STLNB-algoritme. In het geval van blok Hankel-gestructureerde datamatrices maken we de vergelijking met het bestaande HTLSstack-algoritme.

Hoofdstuk 5: Veelhoekreconstructie o.b.v. complexe momenten

In dit hoofdstuk bespreken we een voorbeeld van een GGLRB-probleem. Het veelhoekreconstructieprobleem bestaat erin om de hoekpunten van een vlakke veelhoek te bepalen aan de hand van zijn gegeven complexe momenten.

Volgens een alombekende stelling uit 1964 weten we dat er coëfficiënten a_n, $n = 1, \ldots, N$ bestaan, die enkel afhankelijk zijn van de hoekpunten van een veelhoek P, zodat de volgende vergelijking opgaat voor elke analytische functie $f(z)$ die gedefinieerd is binnen het veelhoekgebied:

$$\int\int_P f''(z)dxdy = \sum_{n=1}^{N} a_n f(z_n).$$

Voor de analytische functie $f(z) = z^k$ wordt de bovenstaande vergelijking:

$$\begin{aligned} \int\int_P f''(z)dxdy &= k(k-1)\int\int_P z^{k-2}dxdy \\ &= \sum_{n=1}^{N} a_n f(z_n) = \sum_{n=1}^{N} a_n z_n^k. \end{aligned}$$

De uitdrukking $k(k-1)\int\int_P z^{k-2}dxdy$ wordt het $k$de complex moment τ_k genoemd. Per definitie geldt er bovendien dat $\tau_0 = \tau_1 = 0$.

We veronderstellen dat er $M+1$ complexe momenten τ_k voor $k = 0, \ldots, M$ voorhanden zijn. We willen de hoekpunten z_n voor $n = 1, \ldots, N$ bepalen door gebruik te maken van het volgende verband tussen momenten en hoekpunten van een veelhoek:

$$\tau_k = \sum_{n=1}^{N} a_n z_n^k, \tag{0.6}$$

waarbij de coëfficiënten a_n voor $n = 1, \ldots, N$ enkel van de hoekpunten afhankelijk zijn. Er wordt verondersteld dat het aantal hoekpunten N gekend is. Als de gemeten momenten ruizig zijn, dan is het stelsel (0.6) niet exact oplosbaar. In dit hoofdstuk veronderstellen we dat de gemeten complexe momenten Gaussische ruis bevatten. Bijgevolg leidt vergelijking (0.6) tot een benaderingsprobleem.

In de literatuur komen er veel algoritmen aan bod om zo'n benaderingsprobleem op te lossen. In dit hoofdstuk beperken we ons tot de TKK-benaderingsmethoden HTLS en STLS, de Generalized Pencil of Function (GPOF)-methode en een tensormethode. We bespreken het verband tussen de HTLS-benadering en de GPOF-benadering en gebruiken de HTLS-benadering om een initiële waarde voor de STLS-benadering te bepalen. We vergelijken de statistische nauwkeurigheid van de vier methoden aan de hand van simulatievoorbeelden. De datamatrix die uit de gegeven rij complexe momenten opgebouwd werd, is Hankel-gestructureerd. We laten zien dat de hoekpunten die via de STLS-benadering bepaald worden nauwkeuriger zijn dan de hoekpunten bepaald via de GPOF-benadering. De reden hiervoor is dat de STLS-benadering de structuur van de datamatrix behoudt. Bovendien blijkt uit simulaties dat het soms nuttig kan zijn om de gegeven complexe momenten in een tensor te schikken. De hoekpunten kunnen in deze gevallen nauwkeuriger geschat worden door de tensorbenadering toe te passen op de datatensor in plaats van te werken met de matrixbenaderingen.

Hoofdstuk 6: Besluit

In dit proefschrift werden algoritmen en toepassingen van GLRB-problemen behandeld. We geven nu kort de belangrijkste besluiten en eigen bijdragen weer.

- De gelijkenissen tussen het Gewogen Lagererangbenaderingsprobleem (GLRB) en het Totale-Kleinste-Kwadraten (TKK)-probleem werden onderzocht. Er werd aangetoond dat, ondanks hun schijnbaar verschillende

probleemformuleringen, beide methoden herleid kunnen worden tot hetzelfde wiskundige kernprobleem, namelijk het vinden van een zo goed mogelijke gewogen lagererangbenadering waarbij de wegingsfactoren bepaald worden door de statistische verdelingen van de meetfouten van de gegeven data. Verschillende oplossingsmethoden, die gebruikt worden om GLRB- en TKK-problemen op te lossen, werden besproken. In het bijzonder werden de NullWLRA-methode, de MLPCA-methode, de EW-TLS-methode en de GTLS-methode besproken. We hebben aangetoond dat deze vier methoden een equivalent gewogen lagererangbenaderingsprobleem behandelen, maar dat ze verschillende algoritmen gebruiken om de beste benaderingsmatrix te bepalen. Al deze bevindingen werden gepubliceerd in [81].

- Het GLRB-probleem werd bestudeerd binnen het toepassingsgebied van de chemometrie. In de chemometrie zijn de PCA-methode en de MLPCA-methode de meest gebruikte methoden om GLRB-problemen op te lossen. Het nut om in de chemometrie algoritmen te gebruiken die oorspronkelijk ontwikkeld werden om TKK-problemen op te lossen, werd besproken. Bovendien werd er een aangepaste versie van het EW-TLS-algoritme ontwikkeld om GLRB-problemen op te lossen voor datamatrices uit de chemometrie die meer kolommen hebben dan rijen en waarbij er enkel correlaties tussen de meetfouten kunnen optreden binnen de rijen. Aan de hand van simulatievoorbeelden hebben we laten zien dat het nieuw ontwikkelde algoritme een beter alternatief kan zijn dan het MLPCA-algoritme voor GLRB-problemen in de chemometrie. Deze resultaten werden in [81] gepubliceerd.

- Het GLRB-probleem werd uitgebreid naar lineair gestructureerde matrices. De Hankel-structuur werd onderzocht aangezien dit één van de meest voorkomende structuren is in toepassingen op het gebied van signaalverwerking. We ontwikkelden algoritmen om het gestructureerd GLRB-probleem op te lossen voor scalaire Hankel-matrices en blok Hankel-matrices waarvan de blokken uit rijvectoren bestaan. De ontwikkelde algoritmen werden vergeleken met bestaande algoritmen die in de literatuur al eerder werden voorgesteld om Hankel-gestructureerde GLRB-problemen op te lossen. Voor het scalair Hankel-gestructureerde geval vergeleken we de statistische nauwkeurigheid en de numerieke efficiëntie van het nieuw ontwikkelde algoritme met het bestaande STLNB-algoritme. In het geval van blok Hankel-gestructureerde datamatrices hebben we de vergelijking gemaakt met het bestaande HTLSstack-algoritme. De resultaten werden in [79, 80] gepubliceerd.

- Als voorbeeld van een gestructureerd GLRB-probleem werd het veelhoekreconstructieprobleem bestudeerd. Het veelhoekreconstructieprobleem bestaat erin om de hoekpunten van een vlakke veelhoek te bepalen aan de hand van zijn gegeven complexe momenten. Tot hiertoe werd het

gebruik van een gestructureerd TKK-algoritme om het reconstructieprobleem op te lossen nog niet besproken in de literatuur. Nochtans is de datamatrix die uit de gegeven rij complexe momenten opgebouwd wordt, Hankel-gestructureerd. Door middel van simulaties hebben we dan ook aangetoond dat de hoekpunten die bepaald worden met behulp van een gestructureerd TKK-algoritme, nauwkeuriger zijn dan de hoekpunten bepaald via methoden die geen rekening houden met de structuur van de datamatrix. Deze bevindingen werden gepubliceerd in [82].

Chapter 1

Introduction

In this chapter, we will situate the weighted low rank approximation problem. In a first section, a short description is given of how the low rank approximation problem and its variations have been tackled throughout many years. In addition, we will also describe the problems which have not been studied extensively in the past, but which will be covered in this thesis. To motivate our study of the weighted low rank approximation problem, the presence of this problem in real life is illustrated in section 1.2 with several examples. A chapter by chapter overview of this thesis is given in section 1.3.

1.1 Introduction

In this section, we will give exact problem formulations of the Low Rank Approximation (LRA) problem and its generalizations. We will also discuss the different methods which have been developed in the past, in order to solve the LRA problem and we will situate the methods that we have worked on and that are presented in this thesis. Before starting with an overview of the work done on LRA problems through history, first the LRA itself needs to be clarified. How can one get a feeling of the meaning and usefulness of an LRA? Probably, by projecting the LRA into daily life.

LRA plays an important role in machine learning and data mining, for instance. In order to find more sophisticated trends in data, potential correlations between larger and larger groups of variables must be considered. Unfortunately, the number of such correlations generally increases exponentially with the number of input variables and, as a result, brute force approaches become unfeasible. What often makes this all the more frustrating is the realization that

human beings have no difficulty identifying all salient features of a high resolution image. How is it possible that humans are able to process such images almost instantly? This discrepancy is generally attributed to the observation that meaningful images do not exploit their full allotment of dimensionality. There are hundreds of thousands of pixels in an image of a face, yet a face has only so many muscles and there are only so many understood expressions (e.g. the facial expression during anger is different from that when one is happy). Humans are not so concerned with the nature of individual pixels, but rather with the implications of a particular trend, the upturning of the mouth to form a smile. This structure inherent in the domain of facial movements suggests that there are just a few important axes for understanding a facial expression: Is the mouth open? What emotion does the face exhibit? Is the person confused? Each of these questions represents a meaningful dimension for the data. Understanding an image could be as simple as discovering the right questions to ask; the proper dimension to consider. So, in order to apply traditional machine learning techniques, the data need to be simplified sufficiently. Yet, the data may not be oversimplified or, to put it in another way, information crucial to understanding should be kept. A method that is widely used for this purpose is to first cast the data as a matrix D and then compute a low rank matrix approximation $\widehat{D}$.

In what follows, exact problem formulations are given of the LRA problem and its extensions. Moreover, methods developed through the years to solve LRA problems are presented. The LRA problem can be formulated as follows: for a given data matrix D of a certain rank k, look for the nearest lower rank matrix $\widehat{D}$ of rank $r < k$ with the same dimensions as D. Mathematically, one has to solve:

$$\min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n}\\ rank\,\widehat{D}\leq r}} \| D - \widehat{D} \|_F^2, \tag{1.1}$$

with D a given $m \times n$ data matrix and $\| \cdot \|_F$ the Frobenius norm, defined as $\| C \|_F^2 = \sum_{i=1}^{s}\sum_{j=1}^{t} c_{ij}^2$ for an arbitrary $s \times t$ matrix C. The low rank approximation problem with respect to the Frobenius norm was first studied by Eckart and Young [25]. They proved that the matrix $\widehat{D}$ can be readily obtained by computing the Singular Value decomposition (SVD) of data matrix D, as stated in the following theorem:

Theorem 1 (Truncated SVD solution) *An important theorem of matrix algebra, called the Singular Value decomposition (SVD), states that any $m \times n$ matrix D can be written as the matrix product of three factors U, Σ and V given by*

$$D = U\Sigma V^\top,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal, Σ is a matrix with the only non-zero elements $\sigma_1, \sigma_2, \ldots, \sigma_k$ on its diagonal, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$ and $rank(D) = k$. The columns of U are called the left singular vectors, the columns

of V are called the right singular vectors and the elements $\sigma_1, \sigma_2, \ldots, \sigma_k$ are called the singular values. In [38], it was proved that such an SVD decomposition always exists and that the decomposition is unique, except for the algebraic signs of the columns of U and V.

Now, let $D = U\Sigma V^\top$ be the SVD decomposition of D. Then, the best rank r approximation of D, for $1 \leq r < k$, is equal to

$$\widehat{D} = U_r \Sigma_r V_r^\top,$$

where Σ_r is obtained from Σ by setting all but the first r singular values to zero and U_r and V_r are the matrices formed by the first r columns of U and V, respectively.

This result is commonly referred to as the Eckart-Young-Mirsky Theorem. Mirsky [68] proved that the result also holds under the 2-norm. An alternative to perform an SVD is to first over-parameterize the problem to remove the rank constraint and then apply an alternating projection algorithm [59]. Specifically, the algorithm in [59] works as follows. Replace $\widehat{D}$ in equation (1.1) with the matrix product AB, where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$. Fix a value for A and minimize $\| D - AB \|_F^2$ over B, then fix B and minimize $\| D - AB \|_F^2$ over A, and repeat this until the product AB converges. The potential disadvantage of this approach is that the decomposition $\widehat{D} = AB$ is not unique, or equivalently, too many parameters are used to represent rank r matrices.

A weak point of the LRA problem is that it treats all entries of the sampled matrix D equally. In order to discriminate between the important and unimportant elements of the matrix, we seek to find an LRA in the *weighted* norm sense.

The weighted LRA (WLRA) problem is to search for the lower rank matrix approximation $\widehat{D}$ of a given matrix D that comes as close as possible with respect to a certain weighted norm. To solve the WLRA problem, one has to compute:

$$\min_{\substack{\widehat{D} \in \mathbb{R}^{m \times n} \\ rank\, \widehat{D} \leq r}} \| D - \widehat{D} \|_W^2, \tag{1.2}$$

with D a given $m \times n$ data matrix of a certain rank k, $r < k$, W a positive-definite, symmetric weighting matrix and $\| C \|_W^2 = vec^\top(C) W vec(C)$, where C is an arbitrary matrix and $vec(C)$ stands for the vectorized form of C, i.e., a vector constructed by stacking the consecutive columns of C in one vector. It is important to note that the norm $\| \cdot \|_W$, which we will use throughout the thesis, is more general than the usual weighted norm $\sum_{i=1}^m \sum_{j=1}^n w_{ij} (c_{ij})^2$. Note, moreover, that for $W = I$, with I the identity matrix, problem (1.2) reduces to the unweighted LRA problem (1.1). Unlike its unweighted version, the WLRA problem does not admit a closed-form solution in general. Presumably, it is because of this that the WLRA problem has received less attention

in the literature than the unweighted LRA problem. The nature of locally optimal solutions arises in this context. So, in general, existing algorithms for the weighted case only converge to a local minimum of (1.2). Because the truncated SVD treats all entries of D equally, this method is no longer appropriate to solve the WLRA problem. The only algorithms we are aware of for solving the WLRA problem are the alternating projection algorithm presented in [59] and algorithms using gradient optimization techniques presented in [61].

Not only the generalization of LRA problems by including weights can be useful in practice. In many signal processing applications, such as Magnetic Resonance Spectroscopy, Speech Compression and Filter Design, the given data matrix D has a specific *linear structure*. To retrieve a structured WLRA (SWLRA) matrix $\widehat{D}$, the following problem needs to be solved:

$$\min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n}\\ rank\,\widehat{D}\leq r}} \parallel D-\widehat{D}\parallel_W^2 \text{ such that } D \text{ and } \widehat{D} \text{ have the same structure,} \qquad (1.3)$$

with D a given $m \times n$ structured data matrix of certain rank k, $r < k$, W a positive-definite, symmetric weighting matrix and $\parallel C \parallel_W^2 = vec^\top(C)Wvec(C)$ for an arbitrary matrix C. The rank condition in (1.3) has to be less than or equal to r, but not necessarily exactly equal to r. It is possible that a given target matrix D does not have a nearest rank r structured matrix approximation, but does have a nearest rank $r-1$ or lower structured matrix approximation. Since the truncated SVD does not preserve structure, other methods need to be used to solve the SWLRA problem. In the literature, many papers can be found that deal with a rank reduction by one. The case when the rank needs to be reduced by more than one, however, has not been studied so intensively. To our knowledge, only references [10, 13] and [95] touch this problem. The so-called Cadzow method [10, 13] is a popular method used by engineers to retrieve the structured LRA matrix $\widehat{D}$ in the unweighted case, for $W = I$. First, this method obtains a rank deficient approximation by truncating the SVD of the given matrix. This approximation will typically have no structure at all. Secondly, the best least squares fit can be computed to the obtained rank deficient matrix that has the required structure. This new matrix will be no longer rank deficient. One could again use a truncation of the SVD in order to find a rank deficient approximation etc. Although this procedure indeed converges to a rank deficient structured matrix, it is shown in [52, 54] that it does not converge to the optimal solution. In [95], an optimal method is presented. However, the method uses over-parameterization. Over-parameterizing the SWLRA problem leads to convergence problems and algorithms which are computationally not very efficient. Moreover, the weighting matrix W is assumed to be diagonal.

Table 1.1 gives an overview of the LRA problems mentioned above. Each cell contains references that address that particular problem. From the table, it is clear that there is still a need for methods to solve the S(W)LRA problem

	$\Delta rank = 1$	$\Delta rank > 1$
LRA	[38]	[4, 25, 30, 85, 92, 99, 98, 107]
WLRA	[59, 61]	[59, 61]
SLRA	[10, 13]	[10, 13]
SWLRA	[1, 2, 23, 77]	[95]

Table 1.1: Overview of tackled Low Rank Approximation problems.

with an arbitrary rank reduction, $\Delta rank > 1$, where $\Delta rank \equiv k - r$. The references mentioned in the corresponding cells present methods that are either non-optimal or implemented in a non-efficient way. In this thesis, algorithms are provided to solve the SWLRA problem for $\Delta rank > 1$ in an optimal and efficient way.

1.2 WLRA in applications

To give an idea of the large amount of fields and related applications in which the WLRA problem occurs, four examples are described in the next four subsections. We will illustrate the occurrence of the WLRA problem in Chemometrics, in Magnetic Resonance Spectroscopy, in Speech Enhancement and in Web Searching.

1.2.1 Chemometrics

In general the primary goal of chemometrics is to develop and utilize models for chemical measurements. Techniques for the treatment of multivariate chemical data need to have the ability to incorporate knowledge of measurement uncertainty into the analysis of experimental data. The measurement error covariance matrix describes the relationships among all of the errors in a series of measurements. In practice, the error covariance matrix is seldom known exactly, but approximations to it can be very useful. Despite the fact that measurement errors are almost universal for multivariate measurements, in chemometrics there have been very few techniques that have taken this into account. As far as we know, only the MLPCA method, explained in chapter 3, provides a framework from which the importance of correlated measurement errors can be assessed.

Over the past several decades, advances in chemometrics have led to the development of multivariate calibration methods for the analysis of chemical mixtures [101]. Calibration allows the user to relate instrumental measurements to the sample of interest. Multivariate calibration allows the analysis of several

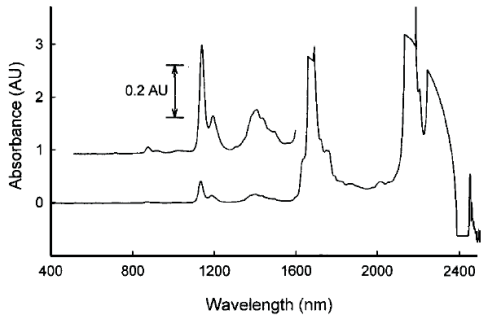


Figure 1.1: Typical spectrum for a three-component mixture containing toluene, chlorobenzene and heptane.

measurements from several samples. This compares to univariate calibration, which involves the use of a single instrumental measurement to determine a single analyte. Either method may contribute to the two-step procedure where (i) data is calibrated and (ii) predictions based on the calibration are made. In calibration, indirect measurements are made from samples where the amount of the analyte has been pre-determined, usually by an independent technique. These measurements, along with the pre-determined analyte levels, comprise a group known as the calibration set. This set is used to develop a model that relates the amount of sample to the measurements by the instrument. Once the model is constructed, it can predict analyte levels based on measurements of new samples. The prediction step for new sample levels uses a model that provides the basis for the evaluation of a linear combination of the measurements.

A multivariate calibration method begins with a set of calibration samples for which the concentrations of the analytes have been obtained by some independent means. The first step in the procedure is to apply an LRA to the spectra of the calibration samples. In applications such as mixture analysis the object is to develop an r-dimensional linear model to describe the data within experimental error. In this case, r is sometimes called the chemical rank of the data set to distinguish it from the mathematical rank, which is nearly always maximized owing to the presence of experimental error. The chemical rank is typically related to the number of underlying chemical components present. For example, consider an $m \times n$ data matrix D which contains the near-infrared spectra for three-component mixtures containing toluene, chlorobenzene and heptane. The number of samples in the calibration set is equal to m and the number of wavelengths is n. Figure 1.1 shows a typical spectrum over the full range. By applying a calibration method to D, an LRA of rank $r = 3$ will be computed in order to construct a calibration model. Well known techniques such as PCR, described in chapter 3, suffer from the fact that they rely on

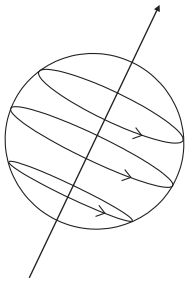


Figure 1.2: The nucleus can be visualized spinning around its own axis.

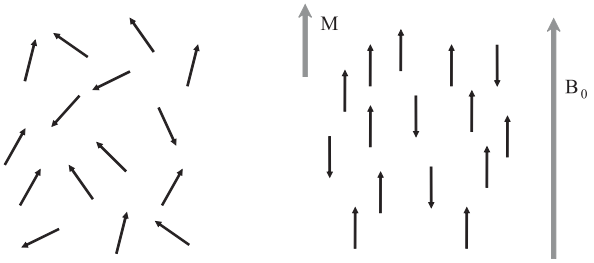


Figure 1.3: Spins aligning along/reversely to the static magnetic field B_0.

SVD to obtain a reliable estimation of the r-dimensional subspace that contains the component spectra. As long as the measurement errors in all of the calibration spectra are all independently and identically distributed (i.i.d.), the r-dimensional hyperplane determined by SVD will be an optimal model for the data. However, if the measurement errors are not independent with uniform variance, this will no longer be true and the estimation of the subspace will be suboptimal. What is needed is a modelling method which accounts for spectral measurement errors in the estimation of the spectral subspace. Such a method is MLPCR [101]. The second step in the procedure of a calibration method is called the prediction step. Based on the calibration, unknown concentrations of the chemical components in new samples can be estimated.

1.2.2 MRS data quantification

The second example concerns a medical application called Magnetic Resonance Spectroscopy (MRS). MRS is a technique used in fundamental research and in clinical environments. During recent years, clinical application of MRS has gained importance, especially as a non-invasive tool for diagnosis and therapy monitoring of brain and prostate tumours. The most important asset of MRS is its ability to determine the concentration of chemical substances non-invasively. Briefly (for a comprehensive treatment of MRS see [32]), the basic principles

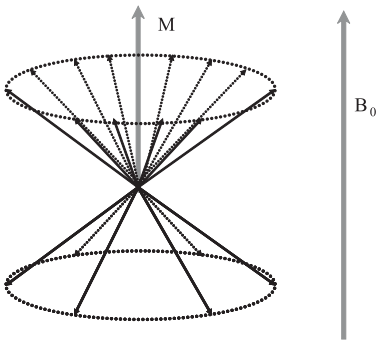


Figure 1.4: Spins precessing around the static magnetic field B_0.

of MRS can be explained as follows: atomic nuclei in a static magnetic field can be in two different spins or energy states: spin up or spin down. This can be understood intuitively by imagining the nucleus spinning around its own axis creating a tiny bar magnet along the axis of rotation, as illustrated in Figure 1.2. When brought in a static magnetic field, the spins will no longer be oriented randomly, but will start to precess around the static magnetic field B_0. However, the different spins will align differently: the magnetic moment of the protons with spin up will align along the magnetic field, whereas the spin down protons will align their magnetic moment inversely to the magnetic field, as illustrated in a simplified way in Figure 1.3. The global created bar magnet is aligned along B_0. Actually, the spins precess around B_0, as illustrated in Figure 1.4.

When an oscillating magnetic field is applied to the system of nuclei, spins will transit to the high-energy state, or from spin down to a spin up state. After removing the oscillating magnetic field, the population of nuclei returns to its equilibrium state, thereby sending out the previously absorbed energy as magnetic waves. Those magnetic waves are detected and generate the MR signal. This signal can be modelled (in the time-domain) as a sum of K exponentially damped sinusoids, where K is the number of expected sinusoids which is assumed to be known, perturbed by i.i.d. complex Gaussian noise:

$$y(n) = \sum_{k=1}^{K} a_k e^{j\phi_k} e^{(d_k + j2\pi f_k)n\Delta t} + \varepsilon_n, \quad n = 0, \ldots, N-1, \tag{1.4}$$

with a_k the amplitude, f_k the frequency, ϕ_k the phase and d_k the damping of the kth component, Δt the sampling interval and $\varepsilon \in \mathbb{C}^{N\times 1}$ a vector of zero-mean i.i.d. Gaussian noise samples.

Every spectral component or term in (1.4) corresponds to a different chemical substance or metabolite present in the volume element from which the signal was obtained. The amplitude a_k is the most important parameter since it is

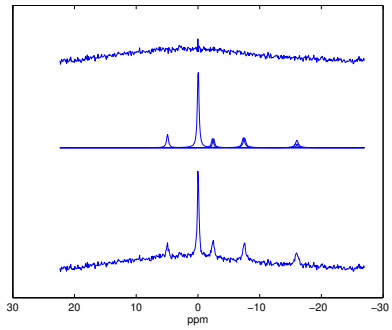


Figure 1.5: SWLRA applied to an MRS signal. From bottom to top, the original noisy signal, the fit by an SWLRA method and the residue are shown.

directly proportional to the concentration of the corresponding metabolite (i.e. the metabolite associated to the resonance frequency f_k) in the volume element under investigation. Therefore, if signals are obtained from several adjacent volume elements in a cross section of e.g. a patient's head, it is possible to produce an image that visualizes the concentration of a specific metabolite over the cross section. In order to remove the noise and get good estimates of the parameters a_k, f_k, ϕ_k, d_k in equation (1.4), the noisy data $y_n, n = 0, \ldots, N-1$ are ordered in a so-called Hankel matrix as follows:

$$H = \begin{bmatrix} y_0 & y_1 & \cdots & y_M \\ y_1 & y_2 & \cdots & y_{M+1} \\ \vdots & & & \\ y_{N-M-1} & y_{N-M} & \cdots & y_{N-1} \end{bmatrix},$$

with $M \geq K$ and $N - M \geq K$. In a Hankel matrix, the entries on the same antidiagonal are all equal to each other and the matrix is fully specified by its first column and last row. Due to the i.i.d. complex Gaussian noise, the rank of matrix H is larger than K. In order to get good estimates of the parameters in (1.4), a low rank K approximation matrix $\widehat{H}$ needs to be computed. It is intuitively obvious that this approximation matrix should have the same Hankel structure. If not, it could mean e.g. that the entries on the fourth anti-diagonal of $\widehat{H}$ are different although they all represent the 'noise-free' estimation of one and the same element y_4. In Figure 1.5, an example is given in which an SWLRA method has been applied to a noisy MRS signal. From bottom to top, the figure contains the real part of the original ^{31}P signal spectrum, the real part of the estimated spectra and the real part of the residual signal spectrum after estimation.

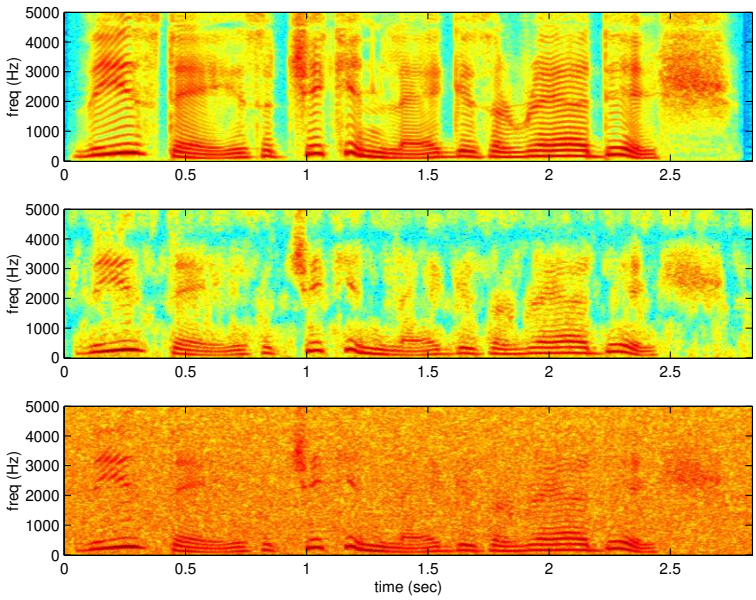


Figure 1.6: Enhancement is applied to a noisy speech signal. From bottom to top, spectrograms of the noisy signal, the enhanced signal and the clean signal are shown.

1.2.3 Speech enhancement

The goal of speech enhancement [27, 46, 44, 22] is to facilitate the understanding, communication and processing of speech signals by suppressing the noise distortion from noisy signals. The speech signal can be distorted by any kind of background noise. The level of the interfering signal can range from very low (e.g. in the office) to very strong (e.g. in a car or on a factory floor). As such, noise reduction can increase the comfort of the listener in a variety of applications like e.g. teleconferencing, mobile phones and hearing aids. It is also useful to improve the quality of valuable old recordings that were typically stored on magnetic media. It is well known that the quality of these recordings tends to deteriorate with time which explains their bad condition. After the original signal has been extracted and enhanced, it can be stored in digital format to preserve it for future generations.

Noise reduction is performed under the assumption of a low-rank signal model that is often attributed to speech signals and that is similar to the model mentioned in the previous example. If a clean (i.e. noise-free) speech vector consists of a sum of p complex damped exponentials, then its $m \times q$ signal observation Hankel matrix H, with $m \geq p$ and $q \geq p$, will be of rank p and will hence be rank deficient. On the other hand, the noise signal that is considered

additive to the speech, will mostly be full rank. Consequently, the observation matrix H of the noisy speech will also be full rank. Speech enhancement is then performed by reducing the rank of the noisy signal observation matrix to rank p, thereby removing the noise subspace. Note that after the rank reduction, the enhanced matrix $\widehat{H}_p$ will in general not be Hankel anymore. Therefore, the enhanced signal cannot be extracted from the first column and the last row of $\widehat{H}_p$. A straightforward and often used solution is to perform an averaging operation along the anti-diagonals of $\widehat{H}_p$ to obtain the enhanced Hankel matrix $\widehat{H}$. Not surprisingly, the averaging operation usually results in an enhanced signal matrix that has full rank again. The enhanced signal can now be extracted from the first column and the last row of $\widehat{H}$. Figure 1.6 shows an example for an utterance of the phonetically balanced speech sentence "These days a chicken leg is a rare dish." The figure shows narrow-band spectrograms of the noisy signal, the enhanced signal and the clean signal for the specific speech sentence mentioned.

1.2.4 Web search

The exponentially growing amount of electronically available documents creates the need for efficient and reliable techniques to retrieve information from large databases. Basic search engines retrieve information by literally matching terms in a user's query with terms extracted from the documents in the database. In a standard web search program such as Google or Yahoo, the results for a given query are ranked in a linear order. Although suitable for some queries, the linear order fails to show the inherent clustered structure of the results for queries with multiple meanings. The same word can have different meanings depending on the context in which it appears. This is called polysemy and results in the retrieval of documents that are of no interest to the user but nevertheless lexically match the query. For instance, consider the query "JAVA". The query can refer to the programming language JAVA, but also to the Island or even to the coffee etc. How to glean the most relevant documents from the (usually large) set of documents returned by a standard web search, for query keywords?

A document-term matrix representation $D \in \mathbb{R}^{m \times n}$ of the results can be constructed [3] which describes the occurrences of terms (i.e. keywords) in documents, $D(i, j)$ is the frequency that term (or keyword) j occurs in document i. It is a sparse matrix whose rows correspond to the documents and whose columns correspond to the terms. The element of the matrix is proportional to the number of times the terms appear in each document, where rare terms are upweighted to reflect their relative importance. After the construction of the occurrence matrix, an LRA to the document-term matrix needs to be computed. The necessity of an approximation can be explained as follows: the original document-term matrix is supposed to be noisy. For instance, anecdot-

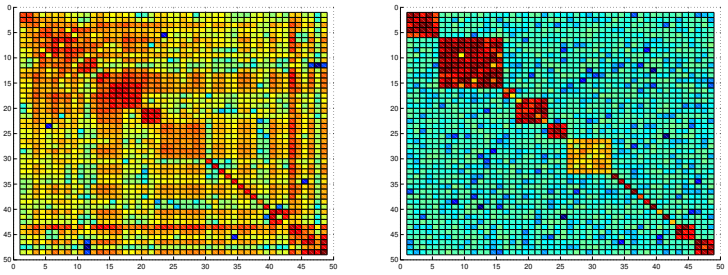


Figure 1.7: A document-term matrix representation of results arranged in the order received from Google (on the left) and from a spectral method (on the right).

ical instances of terms need to be eliminated. As the rank lowering is expected to merge the dimensions associated to terms of similar meanings, it mitigates polysemy. An example is shown in Figure 1.7 for the query "mickey" [12]. The query can refer to multiple people (e.g. Mickey Rourke and Mickey Knox) or even a fictional character (e.g. Mickey Mouse). The user enters the query in Google to find 50 results. In the left part of Figure 1.7, the results for the given query are arranged in a document-term matrix representation in the order received from Google. Each result is a document and the words in its title make up its terms. The right part contains the results arranged according to the cuts made by a spectral algorithm. The cluster structure is apparent. Furthermore, many results are correctly labelled as singletons. By forcing a low-dimensional representation, only the usage patterns that correspond to strong trends are maintained.

1.3 Chapter by chapter overview

This thesis deals with several methods to solve the Weighted Low Rank Approximation problem and with some applications in which the problem occurs.

Chapter 2 introduces the Weighted Low Rank Approximation (WLRA) problem. An exact problem formulation of the WLRA problem is given. The purpose of this chapter is to explore the tight equivalences between WLRA and Total Least Squares (TLS). Despite the seemingly different problem formulations of WLRA and TLS, it is shown that both methods can be reduced to the same mathematical kernel problem, i.e. finding the closest (in a certain sense) weighted low rank matrix approximation where the weight is derived from the distribution of the errors in the data. Different solution approaches, as used in WLRA and TLS, are discussed. In particular, we will discuss the Null

space parameterized WLRA (NullWLRA), the Maximum Likelihood Principal Component Analysis (MLPCA), the Element-wise Weighted-TLS (EW-TLS) and the Generalized TLS (GTLS) problems. These four approaches tackle an equivalent weighted low rank approximation problem, but different algorithms are used to come up with the best approximation matrix $\widehat{D}$. These methods use different representations of the rank constraint $\text{rank}(\widehat{D}) \leq r$ and different optimization techniques in order to solve the WLRA problem. Any of the representations for the rank constraint is bilinear. To handle such a bilinear problem, the MLPCA method makes use of an alternating least squares algorithm, while the other three approaches use the optimization technique of eliminating one of the variables and obtaining an equivalent problem in the other variable. In addition to the mathematical descriptions of the four problem formulations and their equalities and differences, also their algorithms are presented to come up with the best approximation matrix. These findings have been published in [81].

Chapter 3 concerns the WLRA problem in chemometrics. In chemometrics, existing approaches to come up with the best weighted low rank matrix approximation are the well-known Principal Component Analysis (PCA) method and the MLPCA method. Initially employed by statisticians to describe the variance and covariance of random variables, PCA is more commonly used in chemometrics to describe deterministic relationships among variables, especially in cases where a high degree of collinearity exists. In this context, the advantage of PCA is that it allows multivariate data to be represented by a smaller number of variables, called principal components. PCA can be considered to be a maximum likelihood method if all measurement error standard deviations have the same normal distribution (i.e. i.i.d.). The MLPCA method has been devised in chemometrics as a generalization of the well-known Principal Component Analysis (PCA) method in order to derive consistent estimators in the presence of errors with known error distribution. In addition to these two approaches, we will discuss the use of TLS-like algorithms in chemometrics. In the previous chapter, we have shown that the PCA-like approaches and the TLS-like approaches tackle an equivalent weighted low rank approximation problem, but that they use different algorithms. So, it makes sense to compare the performances of the different algorithms on simulated as well as real-life chemical data. For the comparison, several simulated and experimental data sets from chemical measurements are used. The data sets that are used differ in the structure of the covariance matrix W of the measurement noise. We will use data sets with uncorrelated measurement errors as well as data sets where correlations between the measurement errors exist only along the rows. We will discuss the individual weak and strong points of the different algorithms. These results are published in [81].

The fourth chapter extends the WLRA approach towards linearly structured matrices. First of all, we will give an exact problem formulation of the struc-

tured WLRA (SWLRA) problem. It will be shown that it is not possible to derive a general algorithm that can deal with any type of linearly structured matrices. In this chapter, the Hankel structure will be investigated since this is one of the most frequently occurring structures in signal processing applications. As a result Toeplitz matrices are also dealt with since they can be converted into Hankel matrices with a simple permutation of the rows, whereas the solution of the corresponding SWLRA problem does not change. The first part of this chapter tackles the SWLRA problem in the case of scalar Hankel matrices, while the second part deals with block-row Hankel matrices. For both structures an equivalent unconstrained optimization problem will be derived by means of the method of Lagrange multipliers. Algorithms will be presented in order to solve these two unconstrained optimization problems. The correctness of the proposed algorithm for scalar Hankel matrices will be verified on a benchmark problem. Its statistical accuracy and numerical efficiency will be compared with a previously proposed algorithm in the literature for solving Hankel WLRA problems. Moreover, simulation experiments will confirm the improved statistical accuracy of the proposed algorithm for block-row Hankel matrices compared to known algorithms from the literature. These results are published in [79, 80].

In chapter 5, the shape-from-moments problem will be discussed. The formulation of this problem of recovering the vertices of a planar polygon from its measured complex moments is very similar to several other diverse applications found in the literature, such as decomposing a signal built as a linear mixture of complex exponentials. All these applications lead to the very same formulation and, therefore, to the same estimation problem when noise is involved. The literature offers many algorithms for solving such an estimation problem. In this chapter, we will restrict our discussion to the Total Least Squares (TLS) data fitting models HTLS and STLS, the matrix pencil method GPOF and a tensor-based algorithm. A description of the methods will be given and they will be compared on simulated data to discuss their accuracy. The matrix formed by the given data sequence of moments will be Hankel structured. We will show that the vertices estimated via the structured TLS algorithm are more accurate than the ones estimated via the pencil method because the STLS method preserves the Hankel structure of the matrix. Through experiments it will become clear that the parameter accuracy may improve by arranging the data sequence in a higher-order tensor and estimating the model parameters via a multilinear generalization of the SVD. Nevertheless, for the shape-from-moments problem the tensor approach, which is a natural higher-dimensional generalization of the matrix approach, will only be more accurate in a small area. These findings have been published in [82].

Finally, chapter 6 summarizes the conclusions and outlines suggestions for further research.

1.4 Conclusion

In this chapter, we have situated the weighted low rank approximation problem. A short description has been given of how the low rank approximation problem and its variations have been tackled throughout the years. Also problems have been described which have not been studied extensively in the past, but which will be covered in this thesis. To motivate our study of the weighted low rank approximation problem, the occurrence of this problem in real life has been illustrated with several examples. Finally, a chapter by chapter overview of this thesis is given.

Chapter 2

Introduction to Weighted Low Rank Approximation

First, we will give an exact problem formulation of the Weighted Low Rank Approximation (WLRA) problem. The purpose of this chapter is to explore the tight equivalences between WLRA and Total Least Squares (TLS). To do so, we give an overview of the well-studied TLS problem in section 2.2. The basic motivation for TLS is the following. Let a set of multidimensional data points (vectors) be given. How can one obtain a linear model that explains these data? The idea is to modify all data points in such a way that some norm of the modification is minimized subject to the constraint that the modified vectors satisfy a linear relation. Although the name "total least squares"appeared in the literature only 25 years ago, this method of fitting is certainly not new and has a long history in the statistical literature, where the method is known as "orthogonal regression", "errors-in-variables regression"or "measurement error modelling". Despite the seemingly different problem formulations of WLRA and TLS, it is shown that both methods can be reduced to the same mathematical kernel problem, i.e. finding the closest (in a certain sense) weighted low rank matrix approximation where the weight is derived from the distribution of the errors in the data. Different solution approaches, as used in WLRA and TLS, are discussed. In particular, we will discuss the Null space parameterized WLRA (NullWLRA), the Maximum Likelihood Principal Component Analysis (MLPCA), the Element-wise Weighted-TLS (EW-TLS) and the Generalized TLS (GTLS) methods. These four approaches tackle an equivalent weighted low rank approximation problem, but different algorithms are used to come up with the best approximation matrix.

2.1 Problem formulation

As already mentioned in the introductory Chapter 1, the WLRA problem is an extension of the LRA problem. In words, the WLRA problem searches for a lower rank matrix approximation $\widehat{D}$ of a given data matrix D that comes as close as possible with respect to a certain weighted norm. Mathematically, we will consider the following weighted low rank matrix approximation problem:

$$\min_{\widehat{D}} vec^\top(D-\widehat{D})W^{-1}vec(D-\widehat{D}) \quad \text{s.t. } \operatorname{rank}(\widehat{D}) \leq r, \tag{2.1}$$

where $D \in \mathbb{R}^{m\times n}, m \geq n$ is the noisy data matrix (if $m < n$, simply replace D by $D^\top$ and adjust W in (2.1) accordingly), $\operatorname{rank}(D) = k$, $r < k$, $vec(M)$ stands for the vectorized form of a matrix M, i.e., a vector constructed by stacking the consecutive columns of M in one vector and W is a positive-definite symmetric weighting matrix.

For example, if $\widehat{\Delta D} = D - \widehat{D}$ is the estimated measurement noise, then W is equal to the covariance matrix of $vec(\widehat{\Delta D})$. So, the weighting is related to the noise covariance matrix. For a definition of the covariance matrix the reader is referred to Appendix C. When the measurement noise is centered, normal and independent identically distributed, $W = I$ and the optimal closeness norm is the Frobenius norm, $\|\cdot\|_F$. This is used in the well-studied unweighted LRA problem. In this case, problem (2.1) has an analytic solution in terms of the SVD of D. Nevertheless, when the measurement errors are not identically distributed the Frobenius norm is no longer optimal and a weighted norm $\|\cdot\|_W = vec^\top(\cdot)W^{-1}vec(\cdot)$ is needed instead. In this case a closed-form solution is no longer available and is only implicitly defined via a non-convex optimization problem, which is the case in this thesis.

2.2 Link with Total Least Squares

In order to clarify the link between the WLRA problem and the TLS problem we will give an overview of the TLS problem in a first subsection. Subsequently, the link between both problems will become clear and is explained in subsection 2.2.2. Because of the close relationship between the problems, it is obvious that methods created to solve TLS-like problems may also be useful to solve WLRA problems. This will be discussed in sections 2.3 and 2.4.

2.2.1 TLS: problem formulation, algorithms and extensions

The Total Least Squares (TLS) method is one of several linear parameter estimation techniques that have been devised to compensate for data errors. The

univariate line fitting problem has already been discussed since 1877 [5]. More recently, the TLS approach to fitting has also stimulated interests outside statistics. One of the main reasons for its popularity is the availability of efficient and numerically robust algorithms in which the Singular Value Decomposition (SVD) plays a prominent role [36]. Another reason is the fact that TLS is an application-oriented procedure. It is suited for situations in which all data are corrupted by noise, which is almost always the case in engineering applications. In this sense, TLS and errors-in-variables (EIV) modelling are a powerful extension of classical least squares and ordinary regression, which corresponds only to a partial modification of the data. A comprehensive description of the state of the art on TLS from its conception up to the summer of 1990 and its use in parameter estimation has been presented in [92]. While the latter book is entirely devoted to TLS, a second [96] and third book [97] present the progress in TLS and in the broader field of errors-in-variables modelling from 1990 till 1996 and from 1996 till 2001, respectively.

The problem of *linear parameter estimation* arises in a broad class of scientific disciplines such as signal processing, automatic control, system theory and in general engineering, statistics, physics, economics, biology, medicine, etc. It starts from a model described by a linear equation:

$$\xi_1 b_1 + \cdots + \xi_p b_p = \eta \tag{2.2}$$

where $\xi_1, \ldots, \xi_p$ and η denote the variables and $b = [b_1, \ldots, b_p]^\top \in \mathbb{R}^p$ plays the role of a parameter vector that characterizes the specific system. A basic problem of applied mathematics is to determine an estimate of the *true* but *unknown* parameters from given measurements of the variables. In general, η can be a vector of dimension $d > 1$. In this case, the parameters form a matrix B and more than one (independent) linear relationship between η and the variables ξ_i has to be estimated. This gives rise to an *overdetermined* set of m linear equations ($m > p$) :

$$XB \approx Y \tag{2.3}$$

where the ith row of data matrix $X \in \mathbb{R}^{m \times p}$ and vector $Y \in \mathbb{R}^{m \times d}$ contain the measurements of the variables $\xi_1, \ldots, \xi_p$ and η, respectively.

In the classical least squares approach, as commonly used in ordinary regression, the measurements X of the variables ξ_i are assumed to be free of error and hence, all errors are confined to the observations Y. However, this assumption is frequently unrealistic: sampling errors, human errors, modelling errors and instrument errors may imply inaccuracies of the data matrix X as well. One way to take errors in X into account is to introduce *perturbations also in* X. Therefore, the following TLS problem was introduced in the field of computational mathematics [36, 35] ($R(X)$ denotes the range of X and $\|X\|_F$ its Frobenius norm [38]):

Definition 1 (Total Least Squares problem) *Given an overdetermined set of m linear equations $XB \approx Y$ in $p \times d$ unknowns B. The total least squares problem seeks to*

$$\min_{\widehat{\Delta X}, \widehat{\Delta Y}, \widehat{B}} \| [\widehat{\Delta X} \ \widehat{\Delta Y}] \|_F \quad \text{subject to } (X - \widehat{\Delta X})\widehat{B} = Y - \widehat{\Delta Y} \tag{2.4}$$

$\widehat{B}$ is called a TLS solution and $[\widehat{\Delta X} \ \widehat{\Delta Y}]$ the corresponding TLS correction.

In most multivariate problems, the TLS problem (2.4) has a unique solution which can be obtained from a simple scaling of the d right singular vectors of $[X \ Y]$ corresponding to its d smallest singular values.

Theorem 2 (Solution of the TLS problem) *To solve problem (2.4), $XB \approx Y$ will be rewritten into the following form:*

$$[X \ Y][B^\top \ -I_d]^\top \approx 0.$$

Let $[X \ Y] = U\Sigma V^\top$ be the SVD of $[X \ Y]$. Partition the matrix V as follows:

$$V = \begin{matrix} p & d \\ \end{matrix} \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{matrix} p \\ d \end{matrix}.$$

If V_{22} is nonsingular, the TLS solution $\widehat{B}$ is equal to:

$$\widehat{B} = -V_{12}V_{22}^{-1}.$$

If V_{22} is singular, then the TLS problem (2.4) fails to have a solution. These problems are called non-generic.

Extensions of the basic TLS problem (2.4) to problems in which the TLS solution is no longer unique (called minimum norm TLS since the solution with minimal norm is selected) or fails to have a solution altogether (called nongeneric TLS) and to mixed LS-TLS problems that assume some of the columns of X to be error-free, are considered in detail in [92]. In addition, it is shown how to speed up the TLS computations directly by computing the SVD only partially or iteratively if a good starting vector is available. More recent advances, such as recursive TLS algorithms, neural based TLS algorithms, rank-revealing TLS algorithms, regularized TLS algorithms, TLS algorithms for large scale problems, the core problem, etc., are reviewed in [70, 96, 97].

Under specific conditions, the TLS solution, as introduced in numerical analysis, computes optimal parameter estimates in models with *only measurement error*, referred to as classical *EIV* models. These models are characterized by the fact that the true values of the observed variables satisfy one or more unknown but exact linear relations of the form (2.2). In particular, we define [11, 31, 33]:

Definition 2 (Multivariate EIV regression model) *Assume that the m measurements in X, Y are related to $p \times d$ unknowns B by:*

$$\Xi B = \eta \qquad\qquad X = \Xi + \Delta X \;\; \textit{and} \;\; Y = \eta + \Delta Y \tag{2.5}$$

where $\Delta X, \Delta Y$ represent the measurement errors and all rows of $[\Delta X \; \Delta Y]$ are i.i.d. with zero mean and covariance matrix $\mathcal{C}$, known up to a scalar multiple σ_ν^2.

If additionally $\mathcal{C} = \sigma_\nu^2 I$ is assumed with I the identity matrix (i.e. Δx_{ij} and Δy_{ij} are uncorrelated random variables with equal variance) and $\lim_{m\to\infty} \frac{1}{m}\Xi^\top \Xi$ exists and is positive definite, then it can be proven [31, 35] that the TLS solution $\widehat{B}$ of $XB \approx Y$ estimates the *true* parameter values B, given by $(\Xi^\top \Xi)^{-1}\Xi^\top \eta$, *consistently*, i.e. $\widehat{B}$ converges to B as $m \to \infty$. This TLS property does not depend on any assumed distributional form of the errors. If additionally the errors are normal, then TLS computes the Maximum Likelihood estimate. It should be noted that the TLS correction $[\widehat{\Delta X} \; \widehat{\Delta Y}]$, being of rank $\leq d$, cannot be considered as an appropriate estimator for the true measurement errors ΔX and ΔY, added to the data [36, 92]. Note also that the LS estimates are inconsistent in this case. In these cases, TLS gives better estimates than LS, as confirmed by simulations [92]. This situation may occur far more often in practice than is recognized. It is very common in agricultural, medical and economic science, in humanities, business and many other data analysis situations. Hence TLS should be a quite useful tool to data analysts. In fact, the key role and importance of LS in regression analysis is the same as that of TLS in EIV regression. Nevertheless, a lot of confusion exists in the fields of numerical analysis and statistics about the principle of TLS and its relation to EIV modelling. Roughly speaking, TLS is a special case of EIV estimation and, as such, TLS is reduced to a method in statistics but, on the other hand, TLS appears in many other fields, where mainly the data modification idea is used and explained from a geometric point of view, independently from its statistical interpretation.

The statistical model that corresponds to the basic TLS approach is the no-equation-error EIV regression model with the restrictive condition that the measurement errors on the data are i.i.d. with zero mean and common error covariance matrix, equal to the identity matrix up to an unknown scalar. Most published TLS algorithms just handle this case while other more useful EIV regression estimators did not receive enough attention in computational mathematics. To relax these restrictions, several extensions of the TLS problem have been investigated. In particular, the *mixed LS-TLS* problem formulation [33, 37, 91, 108] allows extension of the consistency of the TLS estimator in EIV models, where some of the variables ξ_i are measured without error. The *data least squares* problem refers to the special case in which all variables except η are measured with error and was introduced in the field of signal processing

by Degroat and Dowling [17] in the mid-nineties. Whenever the errors are independent but *unequally* sized, *weighted TLS* problems should be considered using appropriate diagonal scaling matrices in order to maintain consistency. If, additionally, the errors are also correlated, then the *generalized TLS* (GTLS) [91] problem formulation (defined in the next section) allows extension of the consistency of the TLS estimator in EIV models, provided the corresponding error covariance matrix $\mathcal{C}$ is known up to a factor of proportionality (see Definition 2).

More general problem formulations, such as *restricted TLS*, which also allow the incorporation of equality constraints, have been proposed, as well as equivalent problem formulations using other $\mathrm{L_p}$ norms and known as the so-called *Total* $\mathrm{L_p}$ *approximations* (see [92] for references). The latter problems proved to be useful in the presence of outliers. Robustness of the TLS solution is also improved by adding regularization, resulting in the *regularized TLS* methods [97, 28, 84]. In addition, various types of bounded uncertainties have been proposed in order to improve robustness of the estimators under various noise conditions and algorithms are outlined [96, 97].

Furthermore, *constrained TLS* problems have been formulated. Arun [6] addressed the unitarily constrained TLS problem, i.e., $XB \approx Y$, subject to the constraint that the solution matrix B should be unitary. He proved that this solution is the same as the solution to the orthogonal Procrustes problem [38, p.582]. Abatzoglou et al [2] considered yet another constrained TLS problem, which extends the classical TLS problem (2.4) to the case where the errors $[\Delta X \ \Delta Y]$ in the data $[X \ Y]$ are algebraically related. However, if there is a linear dependence among the error entries in $[\Delta X \, \Delta Y]$, then the TLS solution no longer has optimal statistical properties (e.g. maximum likelihood in case of normality). This happens, for instance, in dynamic system modelling when we try to estimate the impulse response of a system from its input and output by discrete deconvolution. In these so-called *structured TLS* problems, the data matrix $[X \ Y]$ is structured, typically block Toeplitz or Hankel. In order to preserve maximum likelihood properties and consistency of the solution [2, 50], the TLS problem formulation, given in Definition 1, must be extended with the additional constraint that any (affine) structure of X or $[X \ Y]$ must be preserved in $\widehat{\Delta X}$ or $[\widehat{\Delta X} \ \widehat{\Delta Y}]$, where $\widehat{\Delta X}$ and $\widehat{\Delta Y}$ are chosen to minimize the error in the discrete L_1, L_2 and L_∞ norm. For L_2 norm minimization, various computational algorithms have been presented, as surveyed in [96, 97], and shown to reduce the computation time by exploiting the matrix structure in the computations. In addition, it is shown how to extend the problem and solve it, if latency or equation errors are included. Recently, robustness of the structured TLS solution has been improved by adding regularization, see e.g. [66], and the structured TLS problem has been related to the approximate fitting problem of data by linear static and dynamic models in the field of system identification [64].

Yet, another important extension is the *Element-wise Weighted-TLS* (EW-TLS) estimator [75, 62] (see definition in the next section). EW-TLS computes consistent estimates in linear EIV models, where the measurement errors are element-wise differently sized or, more generally, where the corresponding error covariance matrices may differ from row to row, provided the total covariance structure of the errors is known up to a scalar factor. Some of the variables are allowed to be exactly known (observable) [97, 51]. Mild conditions for weak consistency of the EW-TLS estimator have been given and an iterative procedure to compute it has been proposed.

2.2.2 Link between the WLRA and the TLS problem

Despite the seemingly different problem formulations of WLRA (2.1) and TLS (2.4), in this subsection, it is shown that both methods can be reduced to the same mathematical kernel problem, i.e. finding the closest (in a certain sense) weighted low rank matrix approximation where the weight is derived from the distribution of the errors in the data.

Instead of computing the best lower rank r approximation $\widehat{D}$ of a data matrix $D \in \mathbb{R}^{m\times n}$ according to a weighted norm $\|\cdot\|_W$, the TLS solution of the problem $XB \approx Y$ can be computed, where D is split up in two parts $D = [X\, Y]$ such that $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{m\times(n-r)}$. When $W = I$, then the basic TLS solution (2.4) can be computed with TLS correction $[\widehat{\Delta X}\ \widehat{\Delta Y}\,] = D - \widehat{D}$, while an extended version of the TLS problem should be used in the case of a more general W. For the other way around, after solving the TLS problem $XB \approx Y$ with $X \in \mathbb{R}^{m\times p}$ and $Y \in \mathbb{R}^{m\times d}$, the best lower rank $r = p$ approximation matrix $\widehat{D}$ of the augmented data matrix $D = [X\, Y]$ is given by $\widehat{D} = [X\, Y] - [\widehat{\Delta X}\ \widehat{\Delta Y}\,]$, where $[\widehat{\Delta X}\ \widehat{\Delta Y}\,]$ is the corresponding TLS correction. In the case the weighting matrix W in problem (2.1) is equal to the identity matrix I, the TLS correction is defined in Definition 1.

Because of the close relationship between the WLRA problem and the TLS problem, it is obvious that methods created to solve TLS-like problems may also be useful to solve WLRA problems. This will be the topic of the next sections.

2.3 Different approaches for the WLRA problem

Since the matrix weighted low rank approximation problem is non-convex, there are no efficient optimization methods that are guaranteed to find a globally optimal solution. What is aimed at instead, is a locally optimal solution nearby

a given initial approximation. Optimization methods to tackle the WLRA problem (2.1) have been considered in the literature under different names. We will discuss the Null space parameterized WLRA (NullWLRA) method, the Maximum Likelihood Principal Component Analysis (MLPCA) method, and the EW-TLS and the GTLS methods from the TLS family. These methods differ in the parameter representation of the rank constraint $\text{rank}(\widehat{D}) \leq r$ in problem (2.1) and in the optimization technique applied in order to solve the resulting parameter optimization problem. In this section a short description is given of the different problem formulations and the different approaches for these methods. In the next section their different algorithms will be discussed.

The different representations of the rank constraint $\text{rank}(\widehat{D}) \leq r$ in a parametric form are the following:

$\mathcal{C}1$: $\widehat{D} = TP^\top$, where $T \in \mathbb{R}^{m \times r}$ and $P \in \mathbb{R}^{n \times r}$

$\mathcal{C}2$: $\widehat{D}N = 0$, where $N \in \mathbb{R}^{n \times (n-r)}$ and $N^\top N = I$

$\mathcal{C}3$: $\widehat{D} \begin{bmatrix} B \\ -I_{n-r} \end{bmatrix} = 0$, where $B \in \mathbb{R}^{r \times (n-r)}$.

$\mathcal{C}1$ and $\mathcal{C}2$ are equivalent. Representations $\mathcal{C}2$ and $\mathcal{C}3$ only differ in the way N is restricted to be full rank. Representation $\mathcal{C}2$ imposes the additional constraint $N^\top N = I$, while representation $\mathcal{C}3$ sets the lower $(n-r) \times (n-r)$ block of N to $-I$. The constraint $N^\top N = I$ has the advantage that it avoids the so-called non-generic cases, but it is more difficult to deal with. Indeed, by using $\mathcal{C}3$ for representing the rank constraint, the condensed problem is an unconstrained optimization problem, while using $\mathcal{C}2$ the condensed problem remains a constrained optimization problem since the constraint $N^\top N = I$ cannot be substituted in the object function.
Any of the representations $\mathcal{C}1$ – $\mathcal{C}3$ for the rank constraint is bilinear. To handle such a bilinear problem, different optimization techniques can be used:

$\mathcal{T}1$: applying an alternating least squares algorithm

$\mathcal{T}2$: eliminating one of the variables and obtaining an equivalent problem in the other variable

$\mathcal{T}3$: linearizing the constraint and solving iteratively equality-constrained least squares problems.

Thus, by combining different rank representations with different local optimization methods, different solution methods are obtained for the WLRA problem. Now, we can formulate the NullWLRA [61], the MLPCA [102], the EW-TLS [75] and the GTLS [91] problems:

approach	NullWLRA	MLPCA	EW-TLS
rank constraint	$\mathcal{C}2$	$\mathcal{C}1$	$\mathcal{C}3$
optimization technique	$\mathcal{T}2$	$\mathcal{T}1$	$\mathcal{T}2$

Table 2.1: Different approaches of the WLRA, the MLPCA and the EW-TLS problem

NullWLRA: Employing representation $\mathcal{C}2$ for the rank constraint the WLRA problem becomes

$$\min_{N^\top N = I} \left(\min_{\widehat{D}} vec^\top (D - \widehat{D}) W^{-1} vec(D - \widehat{D}) \quad \text{s.t. } \widehat{D}N = 0 \right),$$

with $N \in \mathbb{R}^{n \times (n-r)}$.

MLPCA: Employing representation $\mathcal{C}1$ for the rank constraint the WLRA problem becomes

$$\min_{T} \left(\min_{\widehat{D},P} vec^\top (D - \widehat{D}) W^{-1} vec(D - \widehat{D}) \quad \text{s.t. } \widehat{D} = TP^\top \right),$$

with $T \in \mathbb{R}^{m \times r}$ and $P \in \mathbb{R}^{n \times r}$.

EW-TLS: Employing representation $\mathcal{C}3$ for the rank constraint and block diagonal matrix W with blocks $W_1, \ldots, W_m$ the WLRA problem becomes

$$\min_{\widehat{B}} \left(\min_{\widehat{D}} \sum_{i=1}^{m} (d_i - \widehat{d}_i) W_i^{-1} (d_i - \widehat{d}_i)^\top \quad \text{s.t. } \widehat{D} \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} = 0 \right),$$

with $d_i, \widehat{d}_i \in \mathbb{R}^n$ the i-th row of D and $\widehat{D}$, respectively, and W_i the i-th weighting matrix defined as the covariance matrix of the errors in d_i.

GTLS: Employing representation $\mathcal{C}3$ for the rank constraint and block diagonal matrix W with equal blocks W_f the WLRA problem becomes

$$\min_{\widehat{B}} \left(\min_{\widehat{D}_2} \sum_{i=1}^{m} (d_{2,i} - \widehat{d}_{2,i}) W_f^{-1} (d_{2,i} - \widehat{d}_{2,i})^\top \quad \text{s.t. } \widehat{D} \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} = 0 \right),$$

with $\widehat{d}_{2,i} \in \mathbb{R}^{n_2}$ the i-th row of $\widehat{D}_2$, $\widehat{D} = \begin{bmatrix} D_1 & \widehat{D}_2 \end{bmatrix} \in \mathbb{R}^{m \times n}$ and the weighting matrix W_f is the covariance matrix of the errors, assumed to be equal for all rows, in the i-th row of the noisy data matrix D_2. $D_2 \in \mathbb{R}^{m \times n_2}$ is the noisy part of the data matrix $D \in \mathbb{R}^{m \times n}$. Some columns of D can be known exactly, so $D = [D_1\, D_2] = \begin{bmatrix} D_1 & \widehat{D}_2 + \widehat{\Delta D}_2 \end{bmatrix}$ with $D_1 \in \mathbb{R}^{m \times n_1}$ noise-free and $n_1 + n_2 = n$.

The NullWLRA and the MLPCA problems are the most general problem formulations. The weighting matrix W can be any possible positive-definite covariance matrix of the measurement noise. So, for all data sets, no matter what kind of correlation exists among their measurement errors, the NullWLRA and the MLPCA method can be applied. For the EW-TLS and the GTLS problem the covariance matrix W has a specific block diagonal structure, $W = blkdiag(W_1, \ldots, W_m)$ and $W = blkdiag(W_f, \ldots, W_f)$ respectively. In terms of correlation, it means that there should be no correlation at all among the measurement errors along the columns of the data matrix. Only the rows of the measurement noise matrix can be correlated. In calibration problems [103], the rows of the data matrix are formed by individual spectra. Hence, it is reasonable to discuss the case where correlations between the measurement errors exist only along the rows. The GTLS problem is restrictive in the way that the covariance matrices of the rows of the measurement noise matrix should be the same for all rows. In this thesis, we consider $n_1 = 0$. Thus, all the columns of the data matrix D contain measurement noise. For real life data this is mostly the case.
Another difference between the NullWLRA-/MLPCA problem and the two TLS problem formulations is the approximation variable. The first two problems are matrix approximation problems to find the approximation matrix $\widehat{D}$, while the latter ones are looking for a solution $\widehat{B}$ of a linear model. Nevertheless, one can easily find the corresponding $\widehat{B}$ from $\widehat{D}$ such that $\widehat{D} \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} = 0$ and at any optimal solution $\widehat{B}$ from the TLS problem formulations, $\widehat{D}$ can be obtained from $\widehat{B}$ and the weighting matrices via a closed form expression [62, eq. 14]. Thus, except for the above mentioned differences, the four problems formulated are equivalent. Their numerical algorithms, however, are different. This will be discussed in the next section.

In Table 2.1 the rank constraint representations and the optimization approaches of the four problems mentioned are summarized. Because the GTLS problem is a special case of the EW-TLS problem with $W_i = W_f$ for $i = 1, \ldots, m$, the GTLS problem is not included in the table. Figure 2.1, at the end of this chapter, explains the hierarchy of the four problem formulations.

2.4 Different algorithms for the WLRA problem

In this section, the different algorithms that are used in the NullWLRA approach, the EW-TLS approach and the GTLS approach, are described. The algorithm used in the MLPCA approach is presented in the next chapter.

2.4.1 NullWLRA algorithm

The underlying idea in this subsection is to reformulate the WLRA problem (2.1) as the following double-minimization problem:

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^\top N=I}} \left(\min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n}\\ \widehat{D}N=0}} \| D-\widehat{D} \|_W^2 \right), \tag{2.6}$$

with $\| \cdot \|_W^2 = vec^\top(\cdot)W^{-1}vec(\cdot)$. Close inspection shows that if N and $\widehat{D}$ are the minimizing arguments of the two minimizations in (2.6), then $\widehat{D}$ is the solution of the WLRA problem (2.1): the restriction $\widehat{D}N = 0$ enforces the constraint $\text{rank}(\widehat{D}) \leq r$ since every column of N must belong to the null space of $\widehat{D}$.

The first step, now, in the derivation of an algorithm to solve problem (2.6) consists of finding a closed form expression, $f(N)$, for the solution of the inner minimization of (2.6).

Theorem 3 *Let $f(N)$ be defined as*

$$f(N) \quad = \quad \min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n}\\ \widehat{D}N=0}} \| D-\widehat{D} \|_W^2,$$

with $D \in \mathbb{R}^{m\times n}$ a given data matrix and $W \in \mathbb{R}^{mn\times mn}$ a positive-definite, symmetric weighting matrix. Then a closed-form expression for $f(N)$ is given by

$$f(N) = vec^\top(D)(N\otimes I_m)[(N\otimes I_m)^\top W(N\otimes I_m)]^{-1}(N\otimes I_m)^\top vec(D), \tag{2.7}$$

where $\otimes$ denotes the Kronecker product. $f(N)$ depends only on the column space of N: for any invertible matrix S, $f(NS) = f(N)$. The minimizing $\widehat{D}$ is given by

$$\begin{aligned} vec(\widehat{D}) = (I - W(N\otimes I_m)[(N\otimes I_m)^\top W(N\otimes I_m)]^{-1} \\ (N\otimes I_m)^\top)\, vec(D). \end{aligned} \tag{2.8}$$

Proof: The reader is referred to Appendix A for a complete derivation of equations (2.7) and (2.8). □

As a result of the theorem, the double-minimization problem (2.6) can be written as the following optimization problem:

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^\top N=I}} vec^\top(D)(N\otimes I_m)[(N\otimes I_m)^\top W(N\otimes I_m)]^{-1}(N\otimes I_m)^\top vec(D). \tag{2.9}$$

Secondly, problem (2.9) needs to be solved. Due to the non-convexity of the optimization problem (2.9), algorithms that solve this problem will be typically locally convergent. In this subsection, a linearly convergent, steepest descent algorithm is presented. Because of the presence of local minima, first an expression for the local cost function $g(K)$ is derived. Henceforth, $N_\perp$ is used to denote the orthogonal complement of N, that is, $N_\perp \in \mathbb{R}^{n\times r}$ is any full column rank matrix satisfying $N^\top N_\perp = 0$. Since $N_\perp$ is not uniquely defined, implicit in any statement involving $N_\perp$ is that the statement holds for any fixed choice of $N_\perp$. Use the local parameterization from $K \in \mathbb{R}^{r\times(n-r)}$ into $\mathbb{R}^{n\times(n-r)}$ defined by

$$\phi(K) \quad = \quad N + N_\perp K$$

to form the local cost function

$$g(K) \quad = \quad f(\phi(K)) = f(N + N_\perp K). \tag{2.10}$$

Secondly, an expression for the steepest descent direction of the local cost function $g(K)$ can be derived. This expression will be used to derive a steepest descent algorithm for solving the WLRA problem (2.1).

Theorem 4 *Define $f(N)$ as in (2.7) and define $g(K)$ as in (2.10) with fixed $N \in \mathbb{R}^{n\times(n-r)}$ and $N_\perp \in \mathbb{R}^{n\times r}$. Then, the gradient of $g(K)$ about $K = 0$ is*

$$grad\ g = 2N_\perp^\top (D - B)^\top A, \tag{2.11}$$

where $A \in \mathbb{R}^{m\times(n-r)}$ and $B \in \mathbb{R}^{m\times n}$ are the unique matrices that satisfy

$$\begin{aligned} vec(A) &= [(N \otimes I_m)^\top W (N \otimes I_m)]^{-1} vec(DN) \\ vec(B) &= W vec(AN^\top). \end{aligned} \tag{2.12}$$

Proof : The reader is referred to Appendix B for a complete derivation of equation (2.11). □

Now that an expression for the local cost function $g(K)$ and an expression of its steepest descent direction have been derived, the algorithm can be presented. An outline of the NullWLRA algorithm is given in Algorithm 1.

It is important to note that this steepest descent algorithm is not a standard descent algorithm, because the cost function changes at each iteration. The disadvantage of Algorithm 1 is its computational complexity. Evaluating the cost function f requires many computations. In the unweighted case, with $W = I$, the algorithm can be made more efficient by optimizing over $f(N_\perp)$ rather than over $f(N)$. If $W = I$ and $N^\top N = I$, then equation (2.7) becomes $f(N) = tr(N^\top D^\top D N)$. Hence, the following holds:

$$f(N) + f(N_\perp) \quad = \quad tr(N^\top D^\top D N + N_\perp^\top D^\top D N_\perp)$$

$$\begin{aligned} &= tr([N\, N_\perp]^\top D^\top D[N\, N_\perp]) \\ &= tr(D^\top D) > 0. \end{aligned}$$

Thus, performing steepest *descent* on $f(N)$ is identical to performing steepest *ascent* on $f(N_\perp)$. Maximizing $f(N_\perp)$ makes sense in the case when $r < (n/2)$. When $r < (n/2)$, it is computationally more efficient to maximize $f(N_\perp)$ rather than minimize $f(N)$.

In [61], quadratically convergent algorithms for solving problem (2.9) are presented, but for these algorithms both the gradient and the hessian of the local cost function $g(K)$ are needed. We will not explain these algorithms in the thesis. Not the algorithm, but the reformulation of the original WLRA problem (2.1) into the double-minimization problem (2.6) is the most important part of this subsection.

Algorithm 1 NullWLRA algorithm

1: Input: data matrix $D \in \mathbb{R}^{m\times n}$, covariance matrix $W \in \mathbb{R}^{mn\times mn}$, rank specification r and relative convergence tolerance ε.
2: Choose starting position $N \in \mathbb{R}^{n\times(n-r)}$ and $N_\perp \in \mathbb{R}^{n\times r}$ satisfying $[N\, N_\perp]^\top [N\, N_\perp] = I$. Set step size $\lambda := 1$.
3: **repeat**
4: Compute the cost function

$$f(N) = vec^\top(D)(N\otimes I_m)[(N\otimes I_m)^\top W(N\otimes I_m)]^{-1}(N\otimes I_m)^\top vec(D).$$

5: Compute descent direction $K = -2N_\perp^\top (D-B)^\top A$, where A and B are defined in (2.12).
6: Evaluate $f(N + 2\lambda N_\perp K)$. If $f(N) - f(N + 2\lambda N_\perp K) \geq \lambda \parallel K \parallel^2$, then set $\lambda := 2\lambda$ and repeat this algorithmic step.
7: Evaluate $f(N + \lambda N_\perp K)$. If $f(N) - f(N + \lambda N_\perp K) < 1/2\,\lambda \parallel K \parallel^2$, then set $\lambda := 1/2\,\lambda$ and repeat this algorithmic step.
8: Set $N := N + \lambda N_\perp K$. Renormalize $[N\, N_\perp]$ by computing the QR factorization of N, $N = QR$ and setting $[N\, N_\perp] = Q$.
9: **until** $\parallel K \parallel < \varepsilon$.
10: Compute

$$\begin{aligned} vec(\widehat{D}) = {} & (I - W(N\otimes I_m)[(N\otimes I_m)^\top W(N\otimes I_m)]^{-1} \\ & (N\otimes I_m)^\top)\, vec(D). \end{aligned}$$

11: Output: approximation matrix $\widehat{D}$.

Note 1 (Initial starting point) *We need to choose the initial conditions N and $N_\perp$ very carefully, because the convergence of the algorithm depends on them due to the appearance of local minima. Initialize N with the $(n-r)$ right*

singular vectors of D corresponding to its $(n-r)$ smallest singular values. So, if the SVD of D is equal to $D = U\Sigma V^\top$, then take $N = V_2$ with $V = [V_1\, V_2]$ and $V_2 \in \mathbb{R}^{n\times(n-r)}$. Initialize $N_\perp$ with the r right singular vectors of D corresponding to its r largest singular values. So, take $N_\perp = V_1$.

2.4.2 EW-TLS algorithm of Premoli–Rastello

The algorithm of Premoli–Rastello, see [75, 62], is derived for the special case when

$$W = \operatorname{blk}\operatorname{diag}(W_1, \ldots, W_n), \quad \text{where} \quad W_i \in \mathbb{R}^{m\times m}$$

and when D has size $m \times n$ with $m \leq n$. The following input/output representation of the model is used

$$\operatorname{rank}(\widehat{D}) \leq r \quad \Longleftrightarrow \quad \widehat{X}\widehat{B} = \widehat{Y}, \quad \text{where} \quad \widehat{D}^\top = \begin{bmatrix} \widehat{X} & \widehat{Y} \end{bmatrix},$$

with $\operatorname{col}\dim(\widehat{X}) = r$, $\operatorname{col}\dim(\widehat{Y}) = d$, and $m = r + d$. So, the WLRA problem to solve can be written as follows:

$$\min_{\widehat{B}} \left(\min_{\widehat{D}} \sum_{i=1}^{n} (d_i - \widehat{d}_i)^\top W_i^{-1} (d_i - \widehat{d}_i) \quad \text{s.t. } \widehat{D}^\top \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} = 0 \right), \tag{2.13}$$

with $d_i, \widehat{d}_i \in \mathbb{R}^m$ the i-th column of D and $\widehat{D}$, respectively, and W_i the i-th weighting matrix defined as the covariance matrix of the errors in d_i.

The WLRA problem (2.13) can be solved partially by minimizing analytically with respect to $\widehat{D}$. In this way the following equivalent unconstrained optimization problem is derived

$$\widehat{B}^* = \arg\min_{\widehat{B}} f(\widehat{B}), \tag{2.14}$$

where

$$f(\widehat{B}) = \sum_{i=1}^{n} d_i^\top R^\top (R W_i R^\top)^{-1} R\, d_i, \qquad R = \begin{bmatrix} \widehat{B}^\top & -I \end{bmatrix}. \tag{2.15}$$

Define the residual matrix

$$E(\widehat{B}) = (RD)^\top = X\widehat{B} - Y, \qquad E^\top(\widehat{B}) = \begin{bmatrix} e_1(\widehat{B}) & \cdots & e_n(\widehat{B}) \end{bmatrix}$$

and partition d_i and W_i as follows:

$$d_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \begin{matrix} r \\ d \end{matrix}, \qquad W_i = \begin{bmatrix} W_{x,i} & W_{xy,i} \\ W_{yx,i} & W_{y,i} \end{bmatrix} \begin{matrix} r \\ d \end{matrix}.$$

The first order optimality condition $f'(\widehat{B}) = 0$ of (2.14) is

$$2\sum_{i=1}^{n}\Big(x_i e_i^\top(\widehat{B})\Gamma_i^{-1}(\widehat{B}) - (W_{x,i}\widehat{B} - W_{xy,i})\Gamma_i^{-1}(\widehat{B})e_i(\widehat{B})e_i^\top(\widehat{B})\Gamma_i^{-1}(\widehat{B})\Big) = 0,$$

where

$$\Gamma_i(\widehat{B}) = RW_iR^\top.$$

We aim to find a solution of $f'(\widehat{B}) = 0$ that corresponds to a solution of the low rank approximation problem, *i.e.*, to a global minimum point of f.

The algorithm proposed in [75] uses an iterative procedure starting from an initial approximation $B^{(0)}$ and generating a sequence of approximations $B^{(k)}$, $k = 0, 1, 2, \ldots$, that approaches a solution of $f'(\widehat{B}) = 0$. The iteration is implicitly defined by the equation

$$F(B^{(k+1)}, B^{(k)}) = 0, \tag{2.16}$$

where

$$\begin{aligned} F(B^{(k+1)}, B^{(k)}) = 2\sum_{i=1}^{n}\Big(&x_i\big(B^{(k+1)\top}x_i - y_i\big)^\top\Gamma_i^{-1}(B^{(k)}) \\ &- (W_{x,i}B^{(k+1)} - W_{xy,i})\Gamma_i^{-1}(B^{(k)})e_i(B^{(k)})e_i^\top(B^{(k)})\Gamma_i^{-1}(B^{(k)})\Big). \end{aligned}$$

Note that $F(B^{(k+1)}, B^{(k)})$ is linear in $B^{(k+1)}$, so that $B^{(k+1)}$ can be computed in a closed form as a function of $B^{(k)}$. Equation (2.16) with $B^{(k)}$ fixed can be viewed as a linear relaxation of the first order optimality condition of (2.14), which is a highly nonlinear equation.

An outline of the Premoli–Rastello algorithm is given in Algorithm 2. In general, solving the equation (2.16) for $B^{(k+1)}$ requires vectorization. The *vec* operator vectorizes a matrix column-wise and $\otimes$ denotes the Kronecker product. The identity $vec(XBY) = (Y^\top \otimes X)vec(B)$ is used in order to transform (2.16) to the classical system of equations $G(B^{(k)})vec(B^{(k+1)}) = h(B^{(k)})$, where G and h are given in the algorithm. The mapping $b = vec(B) \mapsto B$ is denoted by vec^{-1}.

Note 2 *In the special cases of rank reduction by one,* i.e., $r = n - 1$, *equation (2.16) becomes particularly simple. In this case,* $\Gamma_i(b^{(k)})$, $e_i(b^{(k)})$, *and* y_i *are scalars, so that (2.16) can be written as*

$$\begin{aligned} \textstyle\sum_{i=1}^{n} \quad &\Big(x_i\big(x_i^\top b^{(k+1)} - y_i\big)\Gamma_i^{-1}(b^{(k)}) - \big(W_{x,i}b^{(k+1)} - W_{xy,i}\big)\Gamma_i^{-1}(b^{(k)})e_i(b^{(k)}) \\ &e_i(b^{(k)})\Gamma_i^{-1}(b^{(k)})\Big) = 0, \end{aligned}$$

Algorithm 2 Algorithm of Premoli–Rastello.

1: Input: the data matrix $D \in \mathbb{R}^{m\times n}$, the covariance matrices $\{W_i\}_{i=1}^n$, a rank specification r, and a convergence tolerance ε.
2: Initial approximation: compute a GTLS solution $\widehat{B}_{\text{gtls}}$ of $XB \approx Y$ where $D^\top = [X\, Y]$ with $W_f = \frac{1}{n}\sum_{i=1}^n W_i$ where W_i is the submatrix of W at the intersection of the rows $(i-1)m+1$ to $i\cdot m$ and the columns $(i-1)m+1$ to $i\cdot m$ and let $B^{(0)} = \widehat{B}_{\text{gtls}}$ (see Algorithm 6).
3: Define: $D = \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix} \begin{matrix} r \\ d \end{matrix}$, where $d = m - r$.
4: $k = 0$.
5: **repeat**
6: Let $G = 0_{rd\times rd}$ and $h = 0_{rd\times 1}$.
7: **for** $i = 1, \ldots, n$ **do**
8: $e_i = B^{(k)\top} x_i - y_i$.
9: $N_i = \left(\begin{bmatrix} B^{(k)} \\ -I \end{bmatrix}^\top W_i \begin{bmatrix} B^{(k)} \\ -I \end{bmatrix} \right)^{-1}$.
10: $n_i = N_i e_i$.
11: $G = G + N_i \otimes (x_i x_i^\top) - (n_i n_i^\top) \otimes W_{x,i}$.
12: $h = h + vec(x_i y_i^\top N_i - W_{xy,i} n_i n_i^\top)$.
13: **end for**
14: Solve the system $Gb = h$ and let $B^{(k+1)} = vec^{-1}(b)$.
15: $k = k + 1$.
16: **until** $\|B^{(k)} - B^{(k-1)}\|_{\text{F}} / \|B^{(k)}\|_{\text{F}} < \varepsilon$
17: Output: $\widehat{B}^* = B^{(k)}$.

which is equivalent to a standard linear system of equations $G(b^{(k)})b^{(k+1)} = h(b^{(k)})$,

$$\underbrace{\sum_{i=1}^n \left(\frac{x_i x_i^\top}{\Gamma_i(b^{(k)})} - W_{x,i} \frac{e_i^2(b^{(k)})}{\Gamma_i^2(b^{(k)})} \right)}_{G(b^{(k)})} b^{k+1} = \underbrace{\sum_{i=1}^n \left(\frac{x_i y_i}{\Gamma_i(b^{(k)})} - W_{xy,i} \frac{e_i^2(b^{(k)})}{\Gamma_i^2(b^{(k)})} \right)}_{h(b^{(k)})}.$$

Note 3 (Relation to Gauss–Newton type algorithms) *Algorithm 2 is* not *a Gauss –Newton type algorithm for solving the first order optimality condition because the approximation F is not the first order truncated Taylor series of f'; it is a different linear approximation. The choice makes the derivation of the algorithm simpler but complicates the convergence analysis.*

Note 4 (Convergence properties) *Algorithm 2 is proven to be locally convergent with a super linear convergence rate, see [62, Sec. 5.3]. The algorithm, however, is not globally convergent and simulation results suggest that the region of convergence to a minimum point could be rather small. Hence, a good initial approximation is required for convergence.*

Note 5 (Computing the approximation matrix $\widehat{D}$) *At any optimal solution $\widehat{B}^*$, $\widehat{D}$ can be obtained from $\widehat{B}^*$ and the weighting matrices $\{W_i\}_{i=1}^n$ via a closed form expression as follows:*

$$\widehat{D} = D + \widehat{\Delta D}, \tag{2.17}$$

where an expression for $\widehat{\Delta D}$ is given by

$$\widehat{\Delta D}^\top = - \begin{bmatrix} d_1^\top R^\top (RW_1R^\top)^{-1} RW_1 \\ \vdots \\ d_n^\top R^\top (RW_nR^\top)^{-1} RW_n \end{bmatrix}.$$

2.4.3 Classical EW-TLS algorithm

The Premoli–Rastello algorithm is a heuristic optimization method. Next, we describe an algorithm for the WLRA problem based on classical local optimization methods. The classical local optimization methods have reached a high level of maturity by now. In particular, their convergence properties are well understood, while the convergence properties of the alternative methods still are not.

In order to apply classical optimization algorithms for the solution of problem (2.1), first we have to choose a parameterization of the model. A possible parameterization is given by the input/output representation, so the problem considered is (2.14).

A quasi-Newton type method requires an evaluation of the cost function $f(\widehat{B})$ and its first derivative $f'(\widehat{B})$. Both the cost function and its first derivative are available in closed form, so that their evaluation is a matter of numerical implementation of the operations involved. The computational steps are summarized in Algorithm 3. The proposed algorithm, based on a classical optimization method, is outlined in Algorithm 4.

The optimization problem (2.14) is a nonlinear least squares problem, *i.e.*,

$$f(\widehat{B}) = F^\top(\widehat{B})F(\widehat{B})$$

for certain $F : \mathbb{R}^{r\times d} \to \mathbb{R}^{nd}$. Therefore, the use of special optimization methods like the Levenberg–Marquardt method is preferable. The vector $F(\widehat{B})$, however, is computed numerically, so that the Jacobian $J(\widehat{B}) = \left[\delta F_i/\delta \hat{b}_j\right]$, where $\hat{b} = vec(\widehat{B})$, cannot be found in closed form. A possible workaround for this problem is proposed in [40], where an approximation, called quasi-Jacobian is used instead. The quasi-Jacobian can be evaluated in a similar way to the one for the gradient, which allows use of the Levenberg–Marquardt method for the solution of the WLRA problem.

Algorithm 3 Cost function and first derivative evaluation.

1: Input: $D \in \mathbb{R}^{m\times n}$, $\{W_i\}_{i=1}^n$, r, and $\widehat{B}$.
2: Define: $D = \begin{bmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix} \begin{matrix} r \\ d \end{matrix}$, where $d = m - r$.
3: Let $f = 0_{1\times 1}$ and $f' = 0_{r\times d}$.
4: **for** $i = 1, \ldots, n$ **do**
5: $\quad e_i = \widehat{B}^\top x_i - y_i$.
6: $\quad$ Solve the system $\left(\begin{bmatrix} \widehat{B} \\ -I \end{bmatrix}^\top W_i \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} \right) n_i = e_i$.
7: $\quad f = f + e_i^\top n_i$.
8: $\quad f' = f' + x_i n_i^\top - (W_{x,i}\widehat{B} - W_{xy,i}) n_i n_i^\top$.
9: **end for**
10: Output: the cost function f and its first derivative f' at $\widehat{B}$.

Algorithm 4 EW-TLS algorithm based on classical local optimization.

1: Input: the data matrix $D \in \mathbb{R}^{m\times n}$, the covariance matrices W_i, $i = 1, \ldots, n$, a rank specification r, and a convergence tolerance ε.
2: Initial approximation: compute a GTLS approximation $\widehat{B}_{\text{gtls}}$ of $D^\top$, and let $B^{(0)} = \widehat{B}_{\text{gtls}}$. (See Step 2 of Algorithm 2.)
3: Apply a standard optimization algorithm, *e.g.*, the BFGS (Broyden, Fletcher, Goldfarb, and Shanno) quasi-Newton method, for the minimization of f over $\widehat{B}$ with initial approximation $B^{(0)}$ and with cost function and first derivative evaluation performed via Algorithm 3. Let $\widehat{B}^*$ be the approximation found by the optimization algorithm upon convergence.
4: Output: $\widehat{B}^*$.

Note 6 (Row versus column error covariances) *Algorithms 3 and 5 are designed for the case when the data matrix D has size $m \times n$ with $m \leq n$ and when D has column-wise correlated measurement errors. When the data matrix has size $m \times n$ with $m \geq n$ and the measurement errors are uncorrelated or row-wise correlated, Algorithms 3 and 5 can be applied to the transposed data matrix. However, in some scientific disciplines, e.g. chemometrics, the data matrix usually has size $m \times n$ with $m \leq n$ and the measurement errors are row-wise correlated. Moreover, it makes sense to study the case where correlations between the measurement errors exist only along the rows, because in calibration problems [102] for example, the rows of the data matrix D are formed by individual spectra. For this case of measurement error correlation, the algorithms need to be optimized by considering the left kernel of $\widehat{D}$, i.e., the following modifications of $\mathcal{C}2$ and $\mathcal{C}3$ should be used:*

$\mathcal{C}2'$: *$N^\top \widehat{D} = 0$, where $N \in \mathbb{R}^{m\times(m-r)}$ and $N^\top N = I$.*

$\mathcal{C}3'$: *$\begin{bmatrix} B^\top & -I_{m-r} \end{bmatrix} \widehat{D} = 0$, where $B \in \mathbb{R}^{r\times(m-r)}$.*

In the next subsection, an algorithm will be designed for the case when the data matrix has size $m \times n$ with $m \leq n$ and row-wise correlated measurement errors.

Note 7 (Computing the approximation matrix $\widehat{D}$) *A closed form expression for $\widehat{D}$ has been given in the previous subsection, in equation (2.17).*

2.4.4 Adapted EW-TLS algorithm

The adapted EW-TLS algorithm is derived for the special case when the data matrix D has size $m \times n$ with $m \leq n$ and row-wise correlated measurement errors. The adapted EW-TLS problem, with the modified constraint $\mathcal{C}3'$, can be formulated as follows:

$$\min_{\widehat{B}_2} \left(\min_{\widehat{D}} vec^\top (D^\top - \widehat{D}^\top)\, W^{-1}\, vec(D^\top - \widehat{D}^\top) \text{ s.t. } [\widehat{B}_2^\top \; - I_{m-r}]\widehat{D} = 0 \right), \tag{2.18}$$

where the weighting matrix W is block diagonal, because the measurements are uncorrelated among the columns:

$$W = \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_m \end{bmatrix}. \tag{2.19}$$

Note that the blocks $W_i \in \mathbb{R}^{n \times n}$, for $i = 1, \ldots, m$, have larger dimensions than the non-zero blocks of the matrix W in the previous subsection.

By defining $R := [\widehat{B}_2^\top \; - I_{m-r}] \in \mathbb{R}^{(m-r)\times m}$, the rank constraint $[\widehat{B}_2^\top \; - I_{m-r}]\widehat{D} = 0$ can now be written as $R\widehat{D} = 0$ or $\widehat{D}^\top R^\top = 0$. So, problem (2.18) can be written as the following optimization problem:

$$\min_{\widehat{B}_2} \min_{\substack{\widehat{D} \in \mathbb{R}^{m\times n} \\ \widehat{D}^\top R^\top = 0}} vec^\top (D^\top - \widehat{D}^\top)\, W^{-1}\, vec(D^\top - \widehat{D}^\top). \tag{2.20}$$

Solving the inner minimization of problem (2.20) via Lagrange multipliers gives:

$$\psi(L, \widehat{D}) \;\; = \;\; vec^\top (D^\top - \widehat{D}^\top) W^{-1} vec(D^\top - \widehat{D}^\top) - tr(L^\top \widehat{D}^\top R^\top),$$

where L is the Lagrange multiplier. By minimizing analytically with respect to $\widehat{D}$, an expression for the cost function $g(\widehat{B}_2)$ of the inner minimization of (2.20) can be found and denoted as follows:

$$g(\widehat{B}_2) = vec^\top (D^\top)(R^\top \otimes I_n)[(R \otimes I_n) W (R^\top \otimes I_n)]^{-1} (R \otimes I_n) vec(D^\top). \tag{2.21}$$

Problem (2.20) can now be written as the following equivalent non-convex unconstrained optimization problem:

$$\widehat{B}_2^* = \arg\min_{\widehat{B}_2} g(\widehat{B}_2). \tag{2.22}$$

Due to the non-convexity of problem (2.22), we consider a standard method for *local* optimization: the Levenberg-Marquardt algorithm (Matlab's `lsqnonlin`), which is a nonlinear least squares optimization algorithm. By applying such an optimization method, the efficiency of the cost function evaluation is of great importance. In the previous subsection, an efficient algorithm to solve the classical EW-TLS problem was described in Algorithm 4 by minimizing over the simplified expression (2.15) of the cost function $f(\widehat{B})$. For the adapted EW-TLS problem, however, it is not so straightforward to rewrite the expression (2.21) for the cost function $g(\widehat{B}_2)$ in a simpler way. Still, computational savings can be achieved as follows. By denoting R_i as the ith column of matrix R, expression $(R \otimes I_n)W(R^\top \otimes I_n)$ in equation (2.21) can be rewritten as

$$[R_1 \otimes I_n \ \dots \ R_m \otimes I_n] \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_m \end{bmatrix} \begin{bmatrix} R_1^\top \otimes I_n \\ \vdots \\ R_m^\top \otimes I_n \end{bmatrix} = \sum_{i=1}^{m} R_i R_i^\top \otimes W_i.$$

Hence, a simplified expression for the cost function $g(\widehat{B}_2)$ is given as follows:

$$g(\widehat{B}_2) = (\sum_{i=1}^{m} R_i^\top \otimes d_i^\top)(\sum_{i=1}^{m} R_i R_i^\top \otimes W_i)^{-1}(\sum_{i=1}^{m} R_i \otimes d_i). \tag{2.23}$$

So, the evaluation of the cost function g is a matter of numerical implementation of the operations involved. The proposed algorithm, based on a classical optimization method, to solve problem (2.22) is outlined in Algorithm 5.

Algorithm 5 Adapted EW-TLS algorithm.

1: Input: the data matrix $D \in \mathbb{R}^{m \times n}$, the covariance matrices W_i, $i = 1, \dots, m$, a rank specification r, and a convergence tolerance ε.

2: Initial approximation $B_2^{(0)}$: compute the TLS solution (see Theorem 2) $B_2^{(0)} = -U_{12}U_{22}^{-1}$, where $D = U\Sigma V^\top$ is the SVD of D and the matrix U is partitioned as follows:

$$U = \begin{array}{c} \begin{matrix} r & m-r \end{matrix} \\ \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix} \end{array}.$$

3: Apply a standard optimization algorithm, *e.g.*, the BFGS (Broyden, Fletcher, Goldfarb, and Shanno) quasi-Newton method, for the minimization of g over $\widehat{B}_2$ with initial approximation $B_2^{(0)}$ and with cost function evaluation performed via implementation of expression (2.23). Let $\widehat{B}_2^*$ be the approximation found by the optimization algorithm upon convergence.

4: Output: $\widehat{B}_2^*$.

2.4.5 GTLS algorithm

The GTLS algorithm is derived for the special case when

$$W = \text{blkdiag}(W_f, \ldots, W_f) \text{ with } W_f \in \mathbb{R}^{n \times n}$$

and when the data matrix D has size $m \times n$ with $m \geq n$. The following input/output representation of the model is used:

$$\text{rank}(\widehat{D}) \leq r \Longleftrightarrow \widehat{X}\widehat{B} = \widehat{Y}, \text{ where } \widehat{D} = [\widehat{X}\,\widehat{Y}],$$

with col $\dim(\widehat{X}) = r$, $\widehat{X} \in \mathbb{R}^{m \times r}$ and $\widehat{Y} \in \mathbb{R}^{m \times (n-r)}$. Hence, the WLRA problem (2.1) can be written as

$$\min_{\widehat{B}} \left(\min_{\widehat{D}} \sum_{i=1}^{m} (d_i - \widehat{d}_i) W_f^{-1} (d_i - \widehat{d}_i)^\top \quad \text{s.t. } \widehat{D} \begin{bmatrix} \widehat{B} \\ -I \end{bmatrix} = 0 \right), \tag{2.24}$$

with $d_i, \widehat{d}_i \in \mathbb{R}^n$ the i-th row of D and $\widehat{D}$, respectively, and W_f the weighting matrix defined as the covariance matrix of the measurement errors in each row d_i. To compute a solution of problem (2.24) the data matrix $D = [X\,Y]$ is scaled such that the error covariance matrix W_f^* of the transformed data $D^* = [X^*\,Y^*]$ is diagonal with equal variances, i.e. $W_f^* \sim I$. The classical TLS algorithm [92] is used to solve this transformed set of equations $X^*B^* \approx Y^*$ and finally the solution of the transformed set is converted into a solution of the original set of equations $XB \approx Y$. An outline of the GTLS algorithm is given in Algorithm 6. Alternatively, the GTLS problem can also be solved by making use of the Generalized SVD [38]. This approach is recommended when W_f is close to rank deficiency. For more details, the interested reader is referred to [91].

Note 8 (Special cases: ordinary TLS and weighted TLS) *It is worth noting that when $W \sim I$ (i.e., the errors of the data matrix D are uncorrelated and equally sized), the GTLS solution reduces to the ordinary TLS estimate. Whenever the errors are uncorrelated but unequally sized (W is diagonal), the GTLS problem formulation is reduced to the weighted TLS problem.*

2.5 Conclusion

In this chapter, we have given an exact problem formulation of the WLRA problem. We have shown that the WLRA problem is closely related to the well-studied TLS problem. In order to solve the WLRA problem, several equivalent approaches have been proposed, namely the NullWLRA method, the MLPCA method, the EWTLS method and the GTLS method. Moreover, the different algorithms that are used in these approaches, have been presented.

Algorithm 6 GTLS algorithm via rescaling the data.

1: Input: the data matrix $D \in \mathbb{R}^{m\times n}$, the error covariance matrix $W_f \in \mathbb{R}^{n\times n}$ which is equal for all rows of D and a rank specification r.

2: Define $D = [X\, Y]$, with $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{m\times(n-r)}$.

3: Compute the Cholesky factorization of W_f: $W_f = C^\top C$.

4: Rescale the data matrix D :

$$D^* = [X^*\, Y^*] = [X\, Y]C^{-1} = [X\, Y]\begin{array}{c} \begin{matrix} r & n-r \end{matrix} \\ \begin{bmatrix} C_{11} & C_{12} \\ 0 & C_{22} \end{bmatrix} \end{array} \begin{matrix} r \\ n-r \end{matrix}$$

5: Apply the classical TLS algorithm to the data D^*:
compute a rank r truncated SVD approximation $\widehat{D^*}_{\text{tls}} = U_r S_r V_r^\top$ of D^*, with the SVD of the matrix D^* equal to $USV^\top$, U_r the truncation of the matrix U to $m \times r$, S_r the truncation of S to $r \times r$ and V_r the truncation of V to $n \times r$, $U_r^\top U_r = I$, $V_r^\top V_r = I$;
compute the solution $\widehat{B^*}_{\text{tls}}$ of $X^*B^* \approx Y^*$:

$$\text{if } r \geq n-r \quad \text{then } \widehat{B^*}_{\text{tls}} = -V_{12}V_{22}^{-1}, \quad \text{where } \begin{array}{c} \begin{matrix} r & n-r \end{matrix} \\ \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \end{array} \begin{matrix} r \\ n-r \end{matrix}$$

$$\text{else } \widehat{B^*}_{\text{tls}} = (V_{11}^\top)^{-1}V_{21}^\top.$$

6: Output:
$\widehat{B}_{\text{gtls}} = (C_{11}\widehat{B^*}_{\text{tls}} - C_{12})C_{22}^{-1}$;
$\widehat{D}_{\text{gtls}} = \widehat{D^*}_{\text{tls}}\, C$.

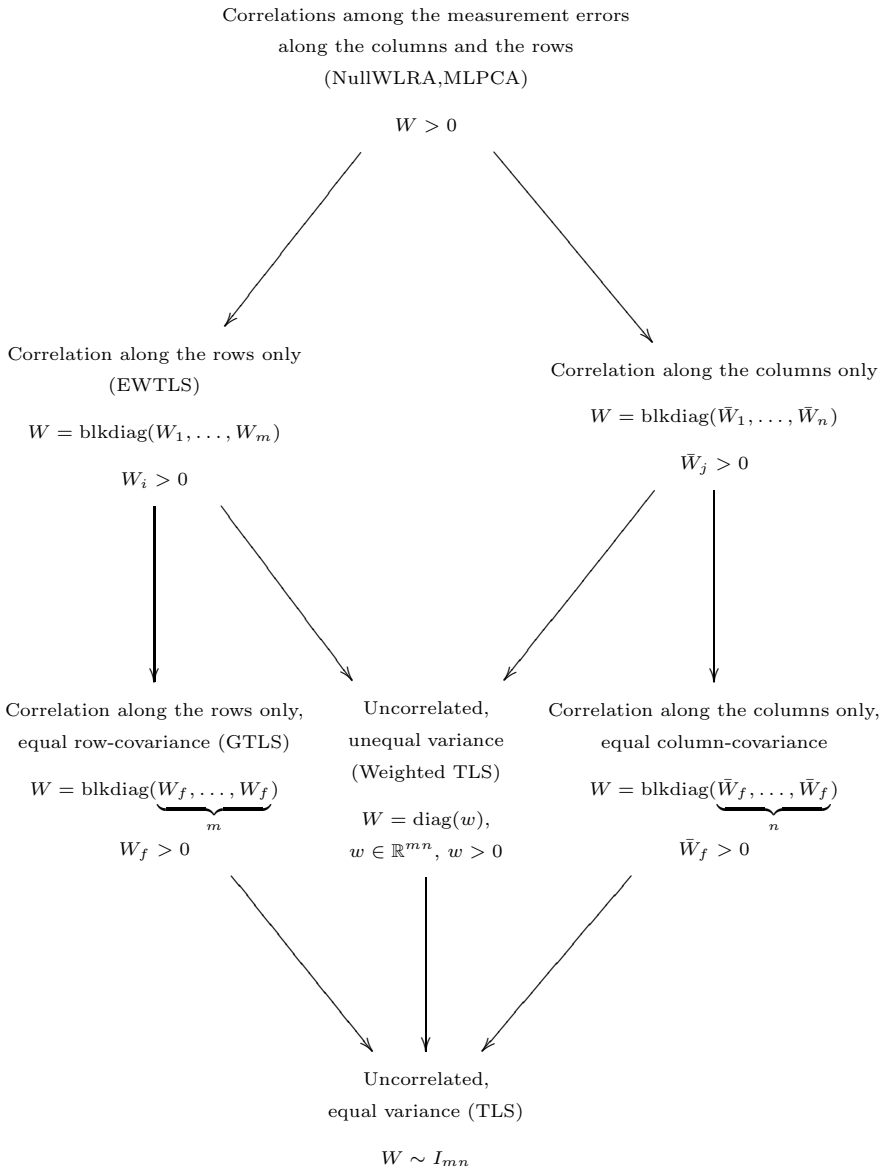


Figure 2.1: Hierarchy of covariance structures of the measurement noise. The problems that use the assumption are given between brackets.

Chapter 3

WLRA problem in chemometrics

In the previous chapter, we discussed the NullWLRA method, the MLPCA method, and the EW-TLS and the GTLS methods from the TLS family. We have shown that these four approaches tackle an equivalent weighted low rank approximation problem, but that they use different algorithms to come up with the best weighted low rank matrix approximation. The MLPCA method has been devised in chemometrics as a generalization of the well-known Principal Component Analysis (PCA) method in order to derive consistent estimators in the presence of errors with known error distribution. For similar reasons, the TLS method has been generalized in the field of computational mathematics and engineering to maintain consistency of the parameter estimates in linear models with measurement errors of known distribution. In this chapter, we will discuss the use of TLS-like algorithms in chemometrics. We will compare the computation times of MLPCA and EW-TLS on simulated as well as real-life chemical data and discuss their convergence behaviour.

3.1 Existing approaches

In general, the primary goal of chemometrics is to develop and utilize models for chemical measurements. Principal Component Analysis (PCA), described in subsection 3.1.1, has perhaps been the most powerful tool of the chemometrician in this regard. Initially employed by statisticians to describe the variance and covariance of random variables, PCA is more commonly used in chemometrics to describe deterministic relationships among variables, especially in

cases where a high degree of collinearity exists. In this context, the advantage of PCA is that it allows multivariate data to be represented by a smaller number of variables, called principal components. In applications such as mixture analysis, the object is to develop an r-dimensional linear model (i.e. an r-dimensional hyperplane) to describe the data within experimental error. In this case, r is sometimes called the chemical rank of the data set to distinguish it from the mathematical rank, which is nearly always maximized owing to the presence of experimental error. The chemical rank is typically related to the number of underlying chemical components present.

PCA has been very successfully applied to modelling in chemistry. The objective of the chemist is generally twofold: (i) to determine the correct form of the model; (ii) to obtain the best estimate of the parameters associated with the model. Unfortunately, PCA is often not an optimal procedure for the estimation of model parameters and can lead to poor models in certain cases. Of course, there are a variety of optimization criteria to be considered when evaluating parameter estimation methods. One widely used approach is to employ a maximum likelihood criterion. Simply put, for a given r-dimensional model the maximum likelihood solution for the model parameters is the one that is most likely to give rise to the observed measurements. Maximum likelihood estimates are generally recognized as having desirable statistical properties and their use has become commonplace. For example, PCA can be considered to be a maximum likelihood method if all measurement error standard deviations have the same normal distribution (i.e. i.i.d.). When minor variations from the assumptions for maximum likelihood estimation are observed, PCA can still be useful, but when the violations become large, it becomes ineffective. This has been remedied in part by incorporating various scaling techniques to reduce the data to i.i.d., but this will not work in the general case. A maximum likelihood method which is more general in its approach to modelling is needed. In this perspective, the Maximum Likelihood PCA (MLPCA) has been developed. MLPCA is an errors-in-variables modelling method in that it accounts for measurement errors in the estimation of model parameters. The method is described in subsection 3.1.2.

3.1.1 PCA/PCR

Before getting to a description of PCA, mathematical concepts that will be used in PCA are described in Appendix C. The appendix covers standard deviation, variance and covariance. This background knowledge can be skipped if the concepts are already familiar.

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

Historically, a distinction has been made between PCA of column-variables and that of row-variables. These are referred to as R-mode or Q-mode PCA, respectively. The modern approach is to consider both analyses as dual. PCA generates the EVD for a cross product of the data D. This cross product can be calculated in covariance or in correlation form. The focus can be on the column (Q-mode) space or the row (R-mode) space. The Q-mode eigenvectors (of $DD^\top$) are named *scores* while the R-mode eigenvectors (of $D^\top D$) are named *loadings*. The first eigenvector explains most of the variance in the measurement matrix, the second eigenvector explains a bit less, and so forth.

The heart of the machinery behind PCA is the Singular Value Decomposition (SVD). The SVD has been described in the introductory chapter, in Theorem 1. PCA uses the SVD to decompose the *expression* matrix D into a set of scores and loadings that are truncated to the first r factors. An outline of the PCA algorithm is given in Algorithm 7. The PCA algorithm computes maximum likelihood estimates under the assumption of homogeneous and independent (i.e. i.i.d.) measurement errors. So, by writing $D = \widehat{D} + \Delta D$, this means that $Cov(\Delta D) \equiv \sigma^2 I$, where $Cov(C) \in \mathbb{R}^{mn \times mn}$ denotes the covariance matrix of a matrix $C \in \mathbb{R}^{m \times n}$.

Algorithm 7 PCA algorithm

1: Input: data matrix $D \in \mathbb{R}^{m \times n}$ and a rank specification r
2: Compute the SVD of the given data matrix D:

$$D = U \Sigma V^\top,$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, Σ is a matrix with the only non-zero elements $\sigma_1, \sigma_2, \ldots, \sigma_k$ on its diagonal, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$ and $\text{rank}(D) = k$.
3: Truncate the matrices U, Σ and V: Σ_r is obtained from Σ by setting all but the first r singular values to zero and U_r and V_r are the matrices formed by the first r columns of U and V, respectively.
4: Derive the approximated data set $\widehat{D}$:

$$\widehat{D} = T P^\top,$$

where $T = U_r \Sigma_r$ is the matrix of scores and $P = V_r$ is the matrix of loadings.
5: Output: $\widehat{D}$.

Instead of decomposing the expression matrix $D = [X\, Y]$, Principal Component Regression (PCR) is based on the decomposition of the matrix X and the PCR model to solve is $XB \approx Y$. In PCR first a PCA is performed on X, then the variables in the Y-matrix are regressed on these principal components of X. So, the variables in the Y-matrix do not play a role at the stage of

computing the principal components of X. The PCR algorithm is described in Algorithm 8. The number p of principal components can be determined by some model validation procedure. For example, the application of the *bootstrap* can be an approach to model validation. We refer the interested reader to [104] for details about the bootstrap methodology in general.

Algorithm 8 PCR algorithm

1: Input: data matrix $D = [X\, Y]$, with $X \in \mathbb{R}^{m\times(n-d)}$ and $Y \in \mathbb{R}^{m\times d}$, and a number p of principal components.
2: Compute the PCA of the given data set X:

$$\widehat{X} = TP^\top,$$

where $T = U_p\Sigma_p$, $P = V_p$, $X = U\Sigma V^\top$ is the SVD of X, Σ_p is obtained from Σ by setting all but the first p singular values to zero and U_p and V_p are the matrices formed by the first p columns of U and V, respectively.
3: Compute the multivariate least squares regression of Y on the major p principal components using the unit-norm singular vectors U_p or the principal components T:

$$\widehat{Y} = U_pU_p^\top Y = T(T^\top T)^{-1}T^\top Y$$

4: Compute the PCR model coefficient matrix B. This can be obtained in a variety of equivalent ways:

$$\widehat{B} = (\widehat{X}^\top\widehat{X})^{-1}\widehat{X}^\top\widehat{Y} = V_p\Sigma_p^{-1}U_p^\top Y = V_p(T^\top T)^{-1}T^\top Y.$$

5: Output: $\widehat{B}$, $\widehat{D} = [\widehat{X}\,\widehat{Y}]$.

By using PCA in Algorithm 8, the dimension reduction of the data set X is done in such a way as to maintain the maximum amount of information. The neglected minor components are supposed to contain noise that is in no way relevant for the relation with Y.

3.1.2 MLPCA

A number of assumptions have been made in the development of the MLPCA method and these should be made clear first. It is assumed that there is a true underlying r-dimensional model for the data. Second, deviations of the measurements from this model are the result only of random measurement errors (no model errors or outliers). Third, the method assumes that the true values of the measurement standard deviations (or error covariance matrices) are known, while in practice, only estimates of these values are normally available. But, MLPCA can still be used effectively with the variance estimates as has been shown in [101].

The MLPCA problem has been discussed in the previous chapter. Here, we will only repeat its problem formulation:

$$\min_{T,P} vec^\top(D - \widehat{D})W^{-1}vec(D - \widehat{D}) \quad \text{s.t. } \widehat{D} = TP^\top, \tag{3.1}$$

with $T \in \mathbb{R}^{m\times r}$ and $P \in \mathbb{R}^{n\times r}$. The MLPCA algorithm, see Algorithm 9, represents a completely general treatment for the case of correlated measurement errors. This algorithm utilizes the elegant simplicity of an Alternating Least Squares (ALS) solution and converges globally to an optimal solution. The convergence rate, however, is linear and the method can be rather slow in practice. Briefly, this ALS solution involves alternating between the row and column spaces of the given data matrix $D \in \mathbb{R}^{m\times n}$. The projections for a trial solution in one space are used to estimate the solution in the alternative space, and the process is repeated until convergence is achieved. In the MLPCA algorithm, the weighting matrix $W \in \mathbb{R}^{mn\times mn}$ is the full error covariance matrix for D and Ω is the full error covariance matrix for $D^\top$ and contains the same information as W in a different arrangement. The relationship between W and Ω is given by the commutation matrix K, which is an $mn \times mn$ permutation matrix with mn non-zero elements and has the properties

$$\Omega = KWK^\top \text{ and } vec(D^\top) = Kvec(D).$$

In practice, the commutation matrix can be computed as follows: begin with an $mn \times 1$ vector a such that $a_i = i$, for all i. Reshape a so that it forms a $m \times n$ matrix A and set $vec(A^\top) = b$. Now, the corresponding elements of a and b are the row and column indices, respectively, of the elements of the commutation matrix K that should be set to one. The remaining elements of K should be set to zero, making it a sparse matrix with mn non-zero elements.

Although the MLPCA algorithm can accommodate correlated measurement errors, two drawbacks have limited its practical utility in these cases: (1) an inability to handle rank deficient error covariance matrices; (2) demanding memory and computational requirements. In the completely general case the covariance matrix will have m^2n^2 elements and easily exceeds the storage capacity of most machines for large matrices unless special measures are used. Fortunately, for many real chemical problems, error covariance is limited to either the row or column directions. In these cases, either the matrix W or Ω will be block diagonal and can be stored as a sparse matrix. Such simplifications are described in the next two subsections.

Simplification 1: row correlations only

In many chemical applications, it may be reasonable to assume that measurement errors are correlated only along the rows or only along the columns. Traditionally, in calibration problems, individual spectra form the rows of the $m \times n$ data matrix D. In the case of natural calibration or a well-designed

Algorithm 9 MLPCA algorithm

1: Input: data matrix $D \in \mathbb{R}^{m\times n}$, a corresponding matrix $W \in \mathbb{R}^{mn\times mn}$ of measurement error covariances for $vec(D)$, a commutation matrix K for D, a rank specification r and a convergence tolerance ε.
2: Compute a rank r truncated SVD approximation of D to obtain an initial approximation to the MLPCA solution: $D^{(0)} = U_r S_r V_r^\top$, with the SVD of the matrix D equal to $USV^\top$, U_r the truncation of the matrix U to $m \times r$, S_r the truncation of S to $r \times r$ and V_r the truncation of V to $n \times r$, $U_r^\top U_r = I$, $V_r^\top V_r = I$.
3: $k = 0$.
4: **repeat**
5: Transpose $D^{(k)}$ and calculate the maximum likelihood estimates in the alternate space using V_r:

$$\begin{aligned} \Omega^{-1} &= KW^{-1}K^\top \\ \mathcal{V} &= I_m \otimes V_r \\ vec(D^{(k+1)}) &= \mathcal{V}(\mathcal{V}^\top \Omega^{-1}\mathcal{V})^{-1}\mathcal{V}^\top \Omega^{-1} vec(D^{(k)}) \end{aligned} \tag{3.2}$$

Calculate the objective function

$$S_1^2 = vec(D^{(k)} - D^{(k+1)})^\top \Omega^{-1} vec(D^{(k)} - D^{(k+1)})$$

6: Reconstruct $D^{(k+1)}$ from $vec(D^{(k+1)})$ and compute the truncated SVD of $D^{(k+1)}$: $D^{(k+2)} = U_r S_r V_r^\top$.
7: $k = k + 2$.
8: Transpose $D^{(k)}$ and calculate the maximum likelihood estimates in the original space using U_r:

$$\begin{aligned} W^{-1} &= K^\top \Omega^{-1} K \\ \mathcal{U} &= I_n \otimes U_r \\ vec(D^{(k+1)}) &= \mathcal{U}(\mathcal{U}^\top W^{-1}\mathcal{U})^{-1}\mathcal{U}^\top W^{-1} vec(D^{(k)}) \end{aligned} \tag{3.3}$$

Calculate the objective function

$$S_2^2 = vec(D^{(k)} - D^{(k+1)})^\top W^{-1} vec(D^{(k)} - D^{(k+1)})$$

9: Reconstruct $D^{(k+1)}$ (original dimensions) and compute the truncated SVD of $D^{(k+1)}$ in the original space: $D^{(k+2)} = U_r S_r V_r^\top$.
10: $k = k + 2$.
11: **until** $(S_1^2 - S_2^2)/S_2^2 < \varepsilon$.
12: Output: $\widehat{D} = D^{(k)}$.

experiment, there should be no correlation in the measurement errors for different samples, so, in this subpart, we will treat the situation where correlations exist only along the rows. The case of column covariance will not be handled separately, because the given data matrix can be transposed and only minor modifications are needed.

In the case of error covariance in the rows only, all of the covariance information is contained in the $n \times n$ row covariance matrices Σ_i, for $i = 1, \ldots, m$, defined as follows:

$$\Sigma_i = E[(d_i - d_i^0)^\top \cdot (d_i - d_i^0)],$$

where d_i and d_i^0 are row vectors of D and D_0, respectively, and E stands for the expectation value. The method assumes that the true values, D_0, of the matrix D are known, while in practice, only estimates of these values are normally available. The full covariance matrix Ω will now be block diagonal, consisting of m blocks of dimensions $n \times n$. This reduces the number of non-zero elements from m^2n^2 to mn^2. While this is still a large number in practice, it makes a critical difference in many cases to store Ω as a sparse matrix. Furthermore, the block diagonal form allows Ω to be inverted by inversion of the individual covariance blocks:

$$\Omega^{-1} = \begin{bmatrix} \Sigma_1^{-1} & & & \\ & \Sigma_2^{-1} & & \\ & & \ddots & \\ & & & \Sigma_m^{-1} \end{bmatrix}.$$

This improves the numerical stability of the MLPCA algorithm. The inverse of the companion covariance matrix W can be obtained by using the commutation matrix K, discussed above:

$$W^{-1} = K^\top \Omega^{-1} K.$$

Although the simplifications described make the implementation of the MLPCA algorithm, Algorithm 9, more feasible, there is still one more practical consideration which arises from the initial estimation of the covariance matrix. In many real applications, the estimated error covariance matrix will be rank deficient and cannot be inverted. To solve this problem, a small value can be added to the diagonal elements of Σ_i prior to its inversion. To do so, the following equation will be used in the MLPCA algorithm:

$$\widehat{\Sigma}_i' = \widehat{\Sigma}_i + I_n \parallel \widehat{\Sigma}_i \parallel \cdot \epsilon \cdot 100n,$$

where ϵ represents the machine precision.

Simplification 2: identical row correlations only

The procedure described above can even be improved for certain cases where the error covariance matrix Σ is the same for each row. Based on the assumption of equal row covariance matrices, a number of practical advantages in storage and speed can be gained. Since Σ does not change, it is now possible to project the entire data matrix D at once, rather than one row vector at a time. Instead of equation (3.2), the following equation can be used:

$$\widehat{D} = D\Sigma^{-1}V_r(V_r^\top\Sigma^{-1}V_r)^{-1}V_r^\top. \tag{3.4}$$

The objective function is then calculated as follows:

$$S^2 = tr[(D-\widehat{D})^\top\Sigma^{-1}(D-\widehat{D})], \tag{3.5}$$

where $tr[A]$ stands for the trace of a matrix A. As usual, an analog must be found in the column space in order to use the ALS solution. This can be done by denoting that

$$\Omega^{-1} = I_m \otimes \Sigma^{-1}.$$

Using the properties of the commutation matrix K, the following holds:

$$\begin{aligned} W^{-1} &= K^\top\Omega^{-1}K \\ &= K^\top(I_m\otimes\Sigma^{-1})K \\ &= \Sigma^{-1}\otimes I_m. \end{aligned}$$

Substituting into equation (3.3) and using properties of the Kronecker product gives:

$$\begin{aligned} vec(\widehat{D}) &= (I_n\otimes U_r)[(I_n\otimes U_r)^\top(\Sigma^{-1}\otimes I_m)(I_n\otimes U_r)]^{-1} \\ &\quad (I_n\otimes U_r)^\top(\Sigma^{-1}\otimes I_m)vec(D) \\ &= (I_n\otimes U_r)[(I_n\otimes U_r^\top)(\Sigma^{-1}\otimes U_r)]^{-1}(\Sigma^{-1}\otimes U_r^\top)vec(D) \\ &= (I_n\otimes U_r)[\Sigma^{-1}\otimes I_r]^{-1}(\Sigma^{-1}\otimes U_r^\top)vec(D) \\ &= (I_n\otimes U_r)(\Sigma\otimes I_r)(\Sigma^{-1}\otimes U_r^\top)vec(D) \\ &= (\Sigma\otimes U_r^\top)(\Sigma^{-1}\otimes U_r^\top)vec(D) \\ &= (I_n\otimes U_rU_r^\top)vec(D) \\ &= vec(U_rU_r^\top DI_n) = vec(U_rU_r^\top D) \end{aligned}$$

or

$$\widehat{D} = U_rU_r^\top D. \tag{3.6}$$

Remark that, when the row covariance matrices are equal, the maximum likelihood projection in the column space (3.6) is equal to the orthogonal projection employed for PCA. By employing equations (3.4), (3.5) and (3.6), the procedure for including correlated errors is greatly simplified and storage requirements are reduced from m^2n^2 to n^2.

3.2 TLS in chemometrics: state-of-the-art

Although TLS, which is described in the previous chapter, has already been applied in chemometrics since 1984 [105], its concept and properties are not well known in chemometrics and in fact a lot of confusion exists about its proper use and relation to well-known chemometric techniques such as PCA, Partial Least Squares (PLS) and other EIV approaches such as Latent Root Regression (LRR).

Several authors, such as A. Burnham et al. [9], S. de Jong [18, 19] and A. Phatak [73, 74, 96], have described the close relationships between a variety of multivariate regression techniques, including TLS, mainly from a theoretical point of view. Although TLS should be primarily used for parameter estimation, it can be used for prediction provided the solution is appropriately computed [101, 105]. In [105], the prediction accuracy and stability of the TLS estimator is compared to that of ordinary LS, PCR, ridge regression and shrunken estimators in a chemical example in multivariate regression in the presence of multicollinearities among the independent variables. In many chemical applications of multivariate regression like the present example of relationships between chemical structure and biological activity, the predictive properties of the model are of prime importance and the regression estimates therefore often need to be stabilized. The example in [105] shows that minimum norm TLS gives a solution to the multivariate regression problem which is stabilized in comparison with the LS solution and which has (at least in the example investigated) minimal prediction error. Stabilization is performed by reducing the matrix $[X\ Y]$, see (2.4), to a matrix of much smaller rank.

More generally, the usefulness of an EIV model for providing insights into the multivariate calibration problem [65] was demonstrated by E. Thomas [87, 96]. Predicting new concentrations of a multicomponent sample from the associated new spectra can be viewed as parameter estimation in an EIV framework with maximum likelihood properties. Moreover, using the responses (spectra) from the available members of the prediction set together improves the accuracy of the new concentrations compared to estimation based on only one response of the current member of the prediction set.

Another EIV approach to multivariate calibration and multivariate regression, two kernel problems in analytical chemistry, was developed by P. Wentzell et al. [101], who generalized the PCA method to Maximum Likelihood PCA (MLPCA)[102, 103], and is described in the previous subsection. Using MLPCA, two approaches to multivariate calibration have been described that allow information on measurement uncertainties to be included in the calibration process in a statistically meaningful way. These methods, referred to as maximum likelihood PCR (MLPCR) and maximum likelihood LRR (MLLRR) are based on principles of maximum likelihood parameter estimation and generalize PCR,

which has been widely used in chemistry, and LRR [41, 100], which has been almost ignored in this field. Whereas MLPCR is based on the decomposition of the calibration matrix X by MLPCA, MLLRR decomposes the augmented data matrix $[X\ Y]$ using MLPCA. In multivariate calibration, this is the original calibration matrix of response variables X augmented by the corresponding concentration vectors Y. (ML)LRR is equivalent to (EW)TLS in the sense that it computes the best lower rank approximation $\widehat{D} = [\widehat{X}\ \widehat{Y}]$ of the augmented data matrix $D = [X\ Y]$ in a (weighted) least squares sense. If $\widehat{X} \in R(\widehat{Y})$, TLS computes the minimum norm solution which equals the minimum norm LS solution and hence TLS is equivalent to LRR, as reported in [101]. However, if only nonpredictive multicollinearities (i.e. linear dependencies among the columns of X only) have been removed via the rank reduction, then $\widehat{X} \notin R(\widehat{Y})$ and the TLS problem is called nongeneric. In this case, the LRR solution differs from the nongeneric TLS solution, as shown in [92]. After removing the nonpredictive multicollinearities, the LRR estimate is found by minimizing a sum of squared residuals, while the nongeneric TLS solution is found by minimizing the corrections applied to the data such that the corrected data satisfy the constraint $\widehat{X} \in R(\widehat{Y})$. By using estimates of the measurement error variance, MLPCR and MLLRR are able to extract the optimum amount of information from each measurement and, thereby, outperform conventional multivariate calibration methods such as PCR and PLS when there is nonuniform error structure. Using simulated and experimental data of three-component mixtures, it is shown that the maximum likelihood methods outperform PCR and PLS in case of nonuniform errors. MLLRR generally performed better than MLPCR, but in most cases the improvement was marginal.

Finally, it should be noted that structured TLS (STLS) clearly differs from dynamic PCA [56]. Although both methods focus on modelling dynamic systems in an EIV context, dynamic PCA is based on computing the truncated SVD of an input-output data matrix and as such does not preserve the matrix structure during rank reduction in contrast with STLS and is hence not optimal.

3.3 The use of TLS and extensions in multivariate (ML)PCR

From the previous section it should be clear that TLS and its extensions can be used in any multivariate (ML) regression/calibration problem $XB \approx Y$ by computing the best lower rank approximation $\widehat{D} = [\widehat{X}\,\widehat{Y}]$ of the augmented data matrix $D = [X\,Y]$ according to a weighted norm. As such, (EW)TLS is equivalent to (ML)LRR in problem formulation. However, (EW)TLS can also be used in any multivariate (ML)PCR problem $XB \approx Y$ where an MLPCA of X is sought prior to the computation of B. Indeed, assume that the true

pseudorank of X is r and the measurement errors on the elements of X are i.i.d. normal, then the optimal rank r matrix approximation $\widehat{X} \in \mathbb{R}^{m \times n}$ in a maximum likelihood sense is obtained by solving the TLS problem $X_1 B \approx X_2$ where X is split up in two parts $X = [X_1 \; X_2]$ such that X_1 has r columns and rank r. The TLS approach then computes minimal (with respect to the Frobenius norm) corrections $\widehat{\Delta X} = [X_1 - \widehat{X}_1 \; X_2 - \widehat{X}_2]$ such that $\widehat{X} = [\widehat{X}_1 \; \widehat{X}_2]$ is of rank r. In other words, TLS projects the data X into a lower dimensional (of dimension r) subspace $R(\widehat{X})$ such that exactly $n - r$ independent linear relations are imposed between $\widehat{X}_1$ and $\widehat{X}_2$. Equivalently, for correlated normally distributed errors where the error covariance matrix W_i of the errors in each row of the data matrix X is known, an optimal lower rank approximation in a maximum likelihood sense is obtained by minimizing a weighted sum of squared corrections. If all W_i are equal to each other, EW-TLS reduces to GTLS. As such, TLS and EW-TLS algorithms can be applied to compute the PCA and MLPCA of a matrix $D = X$, as used in PCR and MLPCR, or $D = [X \, Y]$, as used in LRR and MLLRR.

It is the goal of this chapter to compare the performance of the algorithms used for solving TLS problems with the algorithms used in chemometrics for computing (ML)PCA.

3.4 Performance comparison of MLPCA versus EW-TLS

In this section the different approaches of the MLPCA and the EW-TLS method are compared with each other in order to discuss their individual potential and limitations. For the discussion, several simulated and experimental data sets from chemical measurements are used. The results presented are obtained by implementing the different algorithms in Matlab (version 6.1) on a PC i686 with 800 MHz and 256 MB memory. The data sets which are used differ in the structure of the covariance matrix W of the measurement noise. In the first subsection we use data sets with uncorrelated measurement errors. The second subsection contains data sets with row-wise correlated measurement errors.

3.4.1 Uncorrelated measurement errors

In this subsection we use data sets without any correlation among their measurement errors. For these data sets the weighting matrix W in problem (2.1) is diagonal.

	relative error	time (sec)
MLPCA	0.15834292416862	37.6150
GTLS	0.15834292417331	4.1760

Table 3.1: MLPCA and GTLS for near-infrared spectroscopic data described in Example 3.1.

Equal row error variances

In order to give results for the case of uncorrelated measurement noise with a diagonal covariance matrix W_f which is equal for all rows, we use an experimental data set which was used previously by Wentzell and Andrews [102]. The reader is referred to the original work for a complete description of the experiment. Here, we only describe the experiment briefly.

Example 3.1 31 three-component mixtures containing toluene, chlorobenzene and heptane were derived from an augmented, three-level, three-factor, full factorial design. The spectra were obtained over the range 400-2500 nm on an NIRSystems model 6500 grating spectrometer at intervals of 2 nm and were the average result of 32 scans. Only standard deviations calculated from the replicate data for the first sample were available. The standard deviations for the first sample were used for all samples. So, the error covariance matrix was the same for every row of the measurement error matrix. Because there was no information about the correlations among the data, we used a diagonal covariance matrix for each row. △

Because the error covariance matrix was the same for each row, the GTLS approach could be used. We applied both approaches, the MLPCA and the GTLS, to the noisy data matrix D of size 31×1050 in order to estimate the error-free measurement data matrix $\widehat{D}$ of rank 3. The MLPCA algorithm that we used for this data set is described in Algorithm 9. The GTLS algorithm that we applied is Algorithm 6 from the previous chapter. For both algorithms, the MLPCA and the GTLS, the relative error $\frac{\|D-\widehat{D}\|_F}{\|D\|_F}$ and the computation time is presented in Table 3.1. The table demonstrates that GTLS is a practical alternative to MLPCA for this specific class of structure in the measurement error matrix. The measurement errors are uncorrelated and the covariance matrices of the errors in the rows are assumed to be equal for all rows. This class of measurement noise structure (uncorrelated errors with equal row error variance) is quite common for real life data.

approach	relative error
mlpca	0.00262998780684
mlpca(gtls)	0.00267695997595
ewtls	0.29211400246120
ewtls(opt)	0.00263335308541

Table 3.2: MLPCA and EW-TLS for data with uncorrelated measurement error described in Example 3.2.

Unequal row error variances

To discuss the performance of the MLPCA and the EW-TLS approach for data with uncorrelated measurement errors, we use simulated data sets.

Example 3.2 The data sets all have 10 rows, but the number of columns n increases from 10 till 200 in steps of 10. So, this simulation set contains data sets of size $10 \times n$ and pseudorank 2. For each data set the error-free data matrix D_0 is generated by multiplying a 10×2 random matrix of full rank (with Matlab's `rand`) by a $2 \times n$ matrix that also is created with `rand`. The matrix of the measurement standard deviations corresponding to this $10 \times n$ matrix is determined by generating a $10 \times n$ matrix of uniform distributed random numbers between 0 and 0.01. This ensures that there is no pattern in the standard deviation matrix. Now, the $10 \times n$ matrix of measurement errors ΔD is generated by taking a $10 \times n$ random matrix with normally distributed elements (with Matlab's `randn`) and multiplying this, element-wise, by the standard deviation matrix. Finally, the noisy data matrix D is the sum of the error-free matrix D_0 and the noise matrix ΔD. △

We applied the MLPCA method to the given data matrix D and the EW-TLS method to the transposed data matrix $D^\top$ (see Note 6) to approximate the 'best' rank 2 approximation matrix $\widehat{D}$ and the results are presented in Table 3.2 and Figure 3.1. For the *'mlpca'* approach (see Algorithm 9) the truncated SVD solution is used as initial value. For the *'mlpca(gtls)'* approach the GTLS approximation (see Algorithm 6) is used as initial value instead of the truncated SVD solution. Table 3.2 and Figure 3.1 also contain the results of applying the EW-TLS approach with two different algorithms. For *'ewtls'* Algorithm 2 is used and for *'ewtls(opt)'* Algorithm 4 is applied to the transposed data. For both algorithms we have used the GTLS approximation as initial starting point. Table 3.2 contains the relative error $\frac{\|D_0 - \widehat{D}\|_F}{\|D_0\|_F}$ after applying the four algorithms to a data set of size 10×200. The results in the table are the average of the relative error over 100 repetitions for different noise realizations. From the table it is clear that the *'ewtls'* Algorithm 2 diverges. It is very sensitive to local minima. Nevertheless, the *'ewtls(opt)'* Algorithm 4

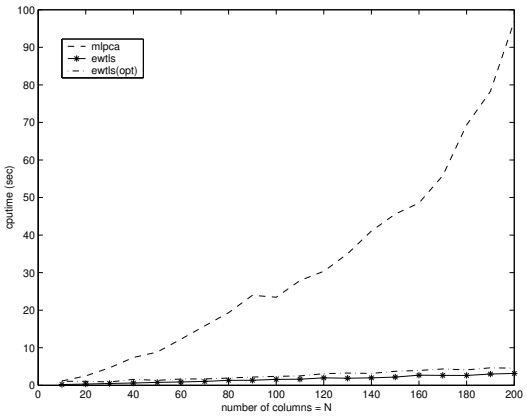


Figure 3.1: The CPUtime of the MLPCA and the EW-TLS approaches for data with uncorrelated measurement error described in Example 3.2.

converges to the same solution as the MLPCA algorithms do. Figure 3.1 shows the computation time of the algorithms. For each n, the experiment is repeated 100 times for different noise realizations and the average execution times are reported. The EW-TLS method outperforms the MLPCA method for this specific case of measurement error. Especially for data sets of size $10 \times n$ with $10 \ll n$, the EW-TLS is much more efficient.

3.4.2 Row-wise correlated measurement errors

In calibration problems [103], the rows of the data matrix are formed by individual spectra. Hence, it is reasonable to discuss the case where correlations between the measurement errors exist only along the rows. So, the errors are uncorrelated along the columns of the data matrix. In these cases the full covariance matrix W of the matrix of measurement errors will be block diagonal and can be stored as a sparse matrix. To calculate the inverse of W the diagonal blocks can be inverted individually. For the discussion of this specific class of measurement structure a simulated data set from chemical measurements is used which was previously described in a paper by Wentzell and Lohnes [103].

Example 3.3 The simulated data set contained spectra from 10 samples of three-component mixtures. The concentration of each component in each of the 10 mixtures had a value between 0 and 1 from a uniform random number distribution. The spectral profiles of the three components were Gaussian with a standard deviation of 20 nm and maximum molar absorptivities at 480 nm, 500 nm and 520 nm, respectively. Pure spectral vectors were generated between 400 nm and 600 nm at 20 nm intervals. The noise-free data matrix D_0 was calculated by multiplying the 10×3 matrix of concentrations by the 3×11

approach	cost ($\sim 10^5$)	relative error
mlpca	0.00058253884923	0.00472272972683
mlpca(gtls)	0.00058233611546	0.00468049099448
ewtls Algorithm 2	9.21164965017561	0.44610730154588
ewtls(opt) Algorithm 4	0.00058356157547	0.00478011794589

Table 3.3: MLPCA and EW-TLS for data with row error covariance only as described in Example 3.3.

matrix of pure component spectra. To add a noise matrix ΔD of correlated measurement errors to the noise-free matrix D_0, first a 10×11 matrix D' of uncorrelated measurement errors was generated. The matrix of the measurement standard deviations corresponding to this 10×11 matrix is determined by generating a 10×11 matrix of uniform distributed random numbers between 0 and 0.01. This ensures that there is no pattern in the standard deviation matrix. Now, the 10×11 matrix of uncorrelated measurement errors D' is generated by taking a 10×11 random matrix with normally distributed elements (with Matlab's `randn`) and multiplying this, element-wise, by the standard deviation matrix. To introduce correlations among the errors within the rows, the rows of matrix D' were filtered using a 1×5 moving average digital filter (see [102, eq. 34-36] for the definition) to construct the correlated error matrix ΔD. This error matrix ΔD was added to the noise-free part D_0 in order to complete the noisy data matrix $D = D_0 + \Delta D$ of size 10×11. $\triangle$

To this simulated data set we have applied the MLPCA approach and the EW-TLS algorithms. Although the standard EW-TLS algorithms Algorithm 2 and Algorithm 4 are not optimized for handling row-wise correlated measurement errors in data sets where $m \ll n$, as is the case here (see Note 6), we nevertheless applied it to this data set. After all, from the previous section it is clear that the EW-TLS approach can be very useful for problems in chemometrics. We compare the cost which is defined by $vec^\top(D - \widehat{D})W^{-1}vec(D - \widehat{D})$ and the relative error $\frac{\|D_0 - \widehat{D}\|_F}{\|D_0\|_F}$. The results are presented in Table 3.3. The table contains the results of applying the MLPCA approach with two different initial starting points. For the *'mlpca'* approach the truncated SVD solution is used. For the *'mlpca(gtls)'* approach the GTLS approximation (Algorithm 6) is used as initial value instead of the truncated SVD solution. We expect that the MLPCA algorithm with the GTLS approximation as initial value needs fewer iterations to converge. Figure 3.2 shows this is indeed the case. Table 3.3 also contains the results of applying the EW-TLS approach with two different algorithms. These are the same as in the previous subsection. For *'ewtls'* Algorithm 2 is used and for *'ewtls(opt)'* Algorithm 4 is applied to the data. For both algorithms we have used the GTLS approximation as initial starting point. From the table it is obvious that we obtain the same

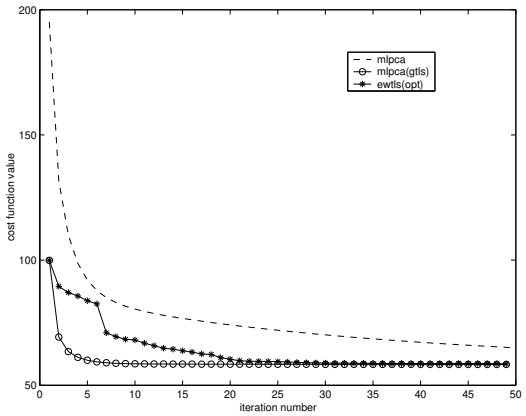


Figure 3.2: The cost function value per iteration number is displayed for the MLPCA and the EW-TLS approaches applied to data with row error covariance only as described in Example 3.3.

solution as the MLPCA algorithms with the *'ewtls(opt)'* algorithm. Figure 3.2 contains the results of the computation time of the MLPCA and the EW-TLS approaches. The *'mlpca(gtls)'* approach performs the best for this specific case of measurement error and matrix size. As said before, the classical EW-TLS algorithm, Algorithm 4, is not optimal for this kind of data set. Figure 3.2 shows that it is better to use the GTLS approximation as initial value for the MLPCA algorithm.

For this kind of data, where $m \leq n$ and the measurements are row-wise correlated, it is better to compare the MLPCA algorithm with the adapted EW-TLS algorithm, Algorithm 5. For the discussion of the performance of the adapted EW-TLS algorithm compared with the classical EW-TLS algorithm and the MLPCA algorithm, three simulated data sets are used: two Monte-Carlo simulations are used and a simulated data set from chemical measurements is used which is previously described in Example 3.3.

Example 3.4 The simulated data set contained matrices $D \in \mathbb{R}^{10\times n}$, for $n = 6, 7, \ldots, 15$. The noise-free data matrix D_0 of rank 2 was calculated by multiplying an arbitrary 10×2 matrix by an arbitrary $2 \times n$ matrix. For every n, 100 different noise realizations were added to D_0 in order to construct a full rank noisy matrix $D = D_0 + \Delta D$. ΔD was a noise matrix with correlations only within the rows. $\triangle$

Both, the classical EW-TLS algorithm (Algorithm 4) and the adapted EW-TLS algorithm (Algorithm 5) were applied to D in order to find the best low rank $r = 2$ approximation matrix $\widehat{D}$ of D. For every n, the mean value of

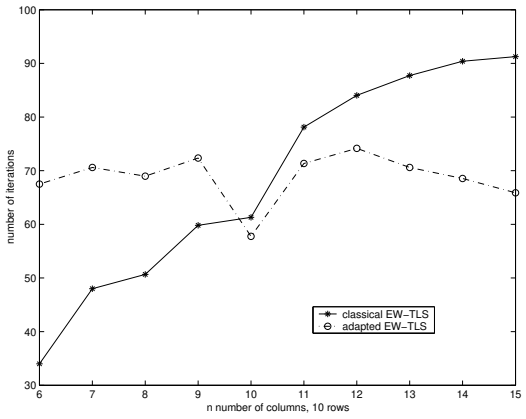


Figure 3.3: Number of iterations needed to compute the classical EW-TLS solution and the adapted EW-TLS solution of the data set described in Example 3.4.

approach	cost	iterations
MLPCA	43.40462099863502	37
classical EW-TLS	43.75752145651096	84
adapted EW-TLS	43.40462102116915	25

Table 3.4: MLPCA and classical/adapted EW-TLS applied to the chemical data set described in Example 3.3.

the number of iterations over the 100 runs is of interest and is visualized in Figure 3.3. From the figure it is clear that the number of iterations needed to compute the adapted EW-TLS solution is less than the number of iterations needed to compute the classical EW-TLS for this case of data set. Indeed, the average number of iterations depends on the number of parameters in the optimization problem to solve. Applying the classical EW-TLS approach to the transposed data matrix $D^\top$ ends up in an optimization problem (2.14) of $r(n-r)$ parameters while the final optimization problem (2.22) for the adapted EW-TLS approach only contains $r(m-r)$ parameters.

To the simulated data set explained in Example 3.3 we have applied the MLPCA algorithm and the EW-TLS algorithms. All the algorithms start from the same initial approximation, the truncated SVD solution. We compare the cost which is defined by $vec^\top(D-\widehat{D})W^{-1}vec(D-\widehat{D})$ and the number of iterations needed to converge. The results are presented in Table 3.4. From the table it is clear that the adapted EW-TLS algorithm needs the fewest number of iterations for the case when the data matrix has size $m \times n$, $m \leq n$, and only row-wise correlated measurement errors.

In order to draw general conclusions about the right choice of algorithms to solve specific problems in chemometrics and other application fields, we created a simulation example, Example 3.5, which contains data matrices of size $m \times n$ with $m < n$, $m > n$ or $m = n$. All the data matrices have measurement errors which are only correlated among the rows.

Example 3.5 The simulated data set contained $m \times n$ matrices D, for $m = 6, 7, \ldots, 13$ and $n = 20 - m$. The noise-free data matrix D_0 of rank r was calculated by multiplying an arbitrary $m \times r$ matrix by an arbitrary $r \times n$ matrix. The desired rank r was varied from 1 to 4. For each combination of m and r, 50 independently generated data matrices D were generated as follows. First, an $m \times n$ matrix D' of uncorrelated measurement errors was generated. The matrix of the measurement standard deviations corresponding to this $m \times n$ matrix is determined by generating an $m \times n$ matrix of uniformly distributed random numbers between 0 and 0.01. This ensures that there is no pattern in the standard deviation matrix. Now, the $m \times n$ matrix of uncorrelated measurement errors D' is generated by taking an $m \times n$ random matrix with normally distributed elements (with Matlab's `randn`) and multiplying this matrix, element-wise, by the standard deviation matrix. To introduce correlations among the errors within the rows, the rows of matrix D' were filtered using a 1×5 moving average digital filter (see [102, eq. 34-36] for the definition) to construct the correlated error matrix ΔD. This error matrix ΔD was added to the noise-free part D_0 in order to complete the noisy data matrix $D = D_0 + \Delta D$ of size $m \times n$. △

We have applied the MLPCA algorithm and the two EW-TLS algorithms to each of the data matrices D of the data set described in Example 3.5. The three algorithms were run from equivalent initial approximations obtained via the SVDs and the stopping criteria were set to the same tolerance. In all the runs, the same solution was found. As expected, the average number of iterations depends on the number of optimization parameters: the fewer the optimization variables, the fewer the average number of iterations for convergence. Numerical results are shown in Table 3.5. The table shows the number of iterations for each algorithm and for the different m, n and r.

From the table it is clear that the adapted EW-TLS algorithm needs the fewest number of iterations for the case when the data matrix has size $m \times n$ with $m < n$ and the measurement errors are only row-wise correlated. When $m > n$, the classical EW-TLS algorithm converges to the right solution within the fewest number of iterations. For square matrices with $m = n$, the MLPCA algorithm seems to converge in the fewest number of iterations.

Nevertheless, in order to draw general conclusions about the right method of choice for a specific problem in chemometrics we also need, in addition to the number of iterations, to take into account the number of floating point oper-

$m \times n$		6×14	7×13	8×12	9×11	10×10	11×9
$r = 1$	MLPCA	(16)	(16)	(16)	17	(**11**)	(17)
	classical EW-TLS	15	16	16	(13)	14	**10**
	adapted EW-TLS	**7**	**8**	**11**	**11**	23	13
$r = 2$	MLPCA	(27)	(28)	(30)	(32)	(**13**)	35
	classical EW-TLS	56	58	46	38	48	(**30**)
	adapted EW-TLS	**13**	**19**	**19**	**21**	28	34
$r = 3$	MLPCA	37	(41)	(46)	(50)	(**18**)	51
	classical EW-TLS	88	84	72	67	66	(53)
	adapted EW-TLS	(**15**)	**19**	**30**	**36**	41	**50**
$r = 4$	MLPCA	46	50	(53)	62	(**24**)	(64)
	classical EW-TLS	88	89	83	(78)	77	**63**
	adapted EW-TLS	(**16**)	(**28**)	**42**	**46**	57	65

Table 3.5: Number of iterations needed by the MLPCA and the EW-TLS algorithms when a
data set described in Example 3.5.

ations (flops) per iteration for each of the algorithms discussed. The MLPCA algorithm is dominated by an SVD, whose computational cost is of the order of $4m^2n+8mn^2+9n^3$ for an $m \times n$ matrix. We obtained the following theoretical number of flops for minimizing the cost function in the adapted EW-TLS algorithm, in step 3 of Algorithm 5: $2mn(m-r)+(m-r)^2+mn^2(m-r)^2+2/3n^3(m-r)^3+n^2(m-r)^2+n(m-r)$. The theoretical number of flops for minimizing the cost function f, expressed in (2.15), in the classical EW-TLS algorithm is of the order $m(2n(n-r)+n^2(n-r)+n(n-r)^2+2/3(n-r)^3+(n-r)^2+(n-r))$. Given the iterative nature of the algorithms, the total number of flops is a multiple of the flops necessary to execute one iteration.

Based on the number of iterations given in Table 3.5 and the theoretical number of flops per iteration, we computed the total number of flops for each algorithm and for the different sizes of m, n and r. By putting the number of iterations between brackets in the table, we emphasize which algorithm has the smallest computational load for the specific choices of m, n and r. We clearly see the computational advantage of the adapted EW-TLS algorithm for the cases of $m << n$ and $r > 2$ and the computational advantage of the classical EW-TLS algorithm for the case when $m >> n$. The classical EW-TLS algorithm seems to behave better. The reason for this is that we could not avoid the Kronecker product in the adapted EW-TLS algorithm. The EW-TLS algorithms are only developed for cases of row-wise correlated measurement errors. So, for more general cases of measurement correlations the MLPCA algorithm should still be the method of choice.

Based on this experiment, we can conclude that the EW-TLS-like algorithms can indeed be nice alternatives to the MLPCA algorithm for the specific cases of row-wise correlated measurement errors and data matrices which are far from squared. Moreover, in subsection 3.4.1 we showed that for uncorrelated measurement errors with equal row variances the GTLS algorithm performs best and that the classical EW-TLS algorithm is also useful for cases where the measurement errors have unequal row variances. These conclusions are visualized in Figure 3.4. In the figure, we suggest which algorithm is the most appropriate one to use for different structures of the covariance matrix W of the measurement noise.

3.5 Conclusion

In this chapter we have described the PCA method and the MLPCA method that are frequently used in chemometrics in order to derive consistent estimators in the presence of errors with known error distribution. Moreover, we have presented the status of the Total Least Squares method and its usefulness in chemometrics. We have compared the computation times of MLPCA and

EW-TLS on simulated as well as real-life chemical data and discussed their convergence behaviour. From these discussions we can conclude that it is better to use the GTLS approximation as initial value for the MLPCA algorithm instead of the truncated SVD. The simulations show that the EW-TLS-like algorithms are nice alternatives to the MLPCA algorithm for the specific cases of row-wise correlated measurement errors and data matrices which are far from squared. For more general cases of correlations between the measurements the MLPCA algorithm should still be the method of choice.

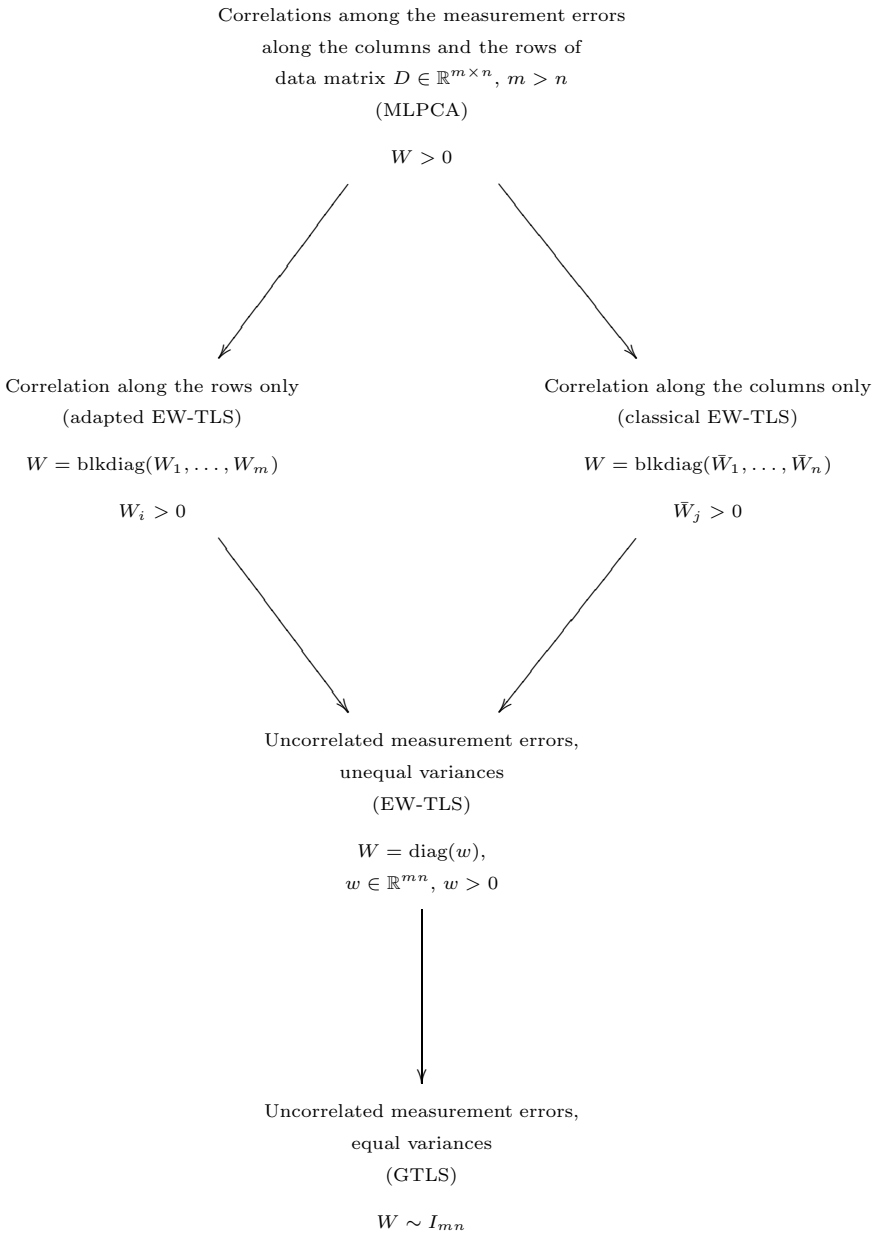


Figure 3.4: Hierarchy of covariance structures of the measurement noise. The algorithms to use are given between brackets.

Chapter 4

Structured WLRA problem

This chapter extends the Weighted Low Rank Approximation (WLRA) approach towards linearly structured matrices. The outline of this chapter is as follows. In the first two sections the Structured WLRA (SWLRA) problem is formulated. The rest of this chapter is divided into two parts. The first part tackles the SWLRA problem in the case of Hankel matrices, while the second part deals with block-row Hankel matrices. In both parts, an equivalent unconstrained optimization problem is derived by means of the method of Lagrange multipliers. An algorithm for solving this unconstrained optimization problem is given and, finally, both parts conclude with some numerical experiments illustrating the statistical optimality of the SWLRA algorithms. The correctness of the SWLRA algorithm for Hankel matrices is verified on a benchmark problem. The statistical accuracy and numerical efficiency of the proposed algorithm are compared with that of STLNB, which is a previously proposed algorithm for solving Hankel WLRA problems. In the case of block-row Hankel matrices simulation experiments confirm the improved statistical accuracy of the SWLRA algorithm for block-row Hankel matrices compared to that of the known HTLSstack algorithm.

4.1 Introduction

Multivariate linear problems play an important role in many applications, including Multiple-Input Multiple-Output (MIMO) system identification and deconvolution problems with multiple outputs. These linear models can be described as $AX \approx B$ or equivalently as $[A\,B][X^\top\ -I]^\top \approx 0$ with $A \in \mathbb{R}^{m\times(n-d)}, B \in \mathbb{R}^{m\times d}$ the observed variables and $X \in \mathbb{R}^{(n-d)\times d}$ the parameter matrix to be estimated. Since often all variables (i.e., those in A and B) are perturbed by noise, the Total Least Squares (TLS) approach is used instead of

the Least Squares (LS) approach. As already mentioned in chapter 2, the TLS approach solves the following problem:

$$\min_{\Delta A, \Delta B, X} \|[\Delta A \ \Delta B]\|_F^2 \text{ such that } (A + \Delta A)X = B + \Delta B, \tag{4.1}$$

where $\|.\|_F$ stands for the Frobenius norm, $D \equiv [A \ B]$ is the observed data matrix and $\Delta D \equiv [\Delta A \ \Delta B]$ is the correction applied to D. The standard way for solving the TLS problem is by means of the Singular Value Decomposition (SVD) [36, 92], as described in Theorem 2.

However, in many signal processing applications, such as Magnetic Resonance Spectroscopy, Speech Compression and Filter Design, the observed data matrix D has a specific *linear structure* and since the SVD does not preserve structure, $[A + \Delta A \ B + \Delta B]$ will typically be unstructured. In [2] it was shown that this loss of structure implies -under noise conditions that occur naturally in many problems- a loss of statistical accuracy of the estimated parameter matrix. These specific noise conditions can best be explained by representing the structured matrix D by its minimal vector representation s (e.g., a Hankel matrix $D \in \mathbb{R}^{m \times n}$ is converted into a minimal representation $s \in \mathbb{R}^{m+n-1}$). The noise part of s has to obey a Gaussian i.i.d. distribution in order to generate the noise condition mentioned earlier. To obtain a maximum likelihood parameter matrix, in these frequently occurring structured cases, one has to solve a so-called Structured Weighted Low Rank Approximation(SWLRA) problem. An exact problem formulation will be given in the next section.

4.2 Problem formulation

To retrieve an SWLRA matrix $\widehat{D}$, the following problem needs to be solved:

$$\min_{\substack{\widehat{D} \in \mathbb{R}^{m \times n} \\ rank(\widehat{D}) \leq r < k \\ \widehat{D} \in \Omega}} \| D - \widehat{D} \|_W^2, \tag{4.2}$$

with $D \in \mathbb{R}^{m \times n}$, rank$(D) \equiv k$, $m \geq n$ (if $m < n$, replace D by $D^\top$), Ω the set of all matrices having the same structure as D and $\|C\|_W^2 \equiv vec(C)^\top W vec(C)$ where $vec(C)$ stands for the vectorized form of C, i.e., a vector constructed by stacking the consecutive columns of C in one vector and W is a positive-definite, symmetric weighting matrix. So, for a given structured data matrix D of a certain rank k, we are looking for the nearest (in a weighted norm $\|.\|_W^2$-sense) lower rank matrix $\widehat{D}$ with the same structure as D. We define $\Delta r \equiv n - r$. The rank condition in (4.2) has to be less than or equal to r, but not necessarily exactly equal to r. It is possible that a given target matrix D does not have a nearest rank r structured matrix approximation, but does have a nearest rank $r - 1$ or lower structured matrix approximation.

For specific choices of W, Δr and Ω, (4.2) corresponds to well-known problems:

- If W is the identity matrix, denoted as $W = I$, and $\Omega = \mathbb{R}^{m \times n}$, problem (4.2) is reduced to the well-studied TLS problem (4.1) with $\widehat{D} = [A + \Delta A \; B + \Delta B]$ and $d = \Delta r$, i.e., we enforce Δr independent linear relations among the columns of $\widehat{D}$ in order to reduce $\widehat{D}$ to rank $\leq r$.

- When Ω is a set of matrices sharing a particular linear structure (for example, Hankel matrices) and W is a diagonal matrix, then (4.2) corresponds to the so-called STLS problem. STLS problems with $\Delta r = 1$ have been studied extensively [1, 2, 23, 77]. For $\Delta r > 1$ only a few algorithms are described [13, 95], most of which yield statistically suboptimal results such as the algorithm to find lower rank structured matrices by alternating iterations between lower rank matrices and structured matrices [13].

- The case in which $\Omega = \mathbb{R}^{m \times n}$ corresponds to the previously introduced unstructured WLRA problem (2.1).

In this chapter we consider (4.2) with Ω the set of a particular type of linearly structured matrices (for example, the set of Hankel matrices). From a statistical point of view it only makes sense to treat identical elements in an identical way, e.g., in a Hankel matrix all the elements on one antidiagonal should be treated the same way. Therefore (4.2) is reformulated as:

$$\min_{\substack{vec_2(\widehat{D}) \\ rank(\widehat{D}) \leq r < k}} \|D - \widehat{D}\|_V^2, \tag{4.3}$$

with $\|C\|_V^2 \equiv vec_2(C)^\top V vec_2(C)$, where $vec_2(C)$ is a minimal vector representation of the linearly structured matrix C and V is a positive-definite, symmetric weighting matrix. Note that due to the one-to-one relation between C and $vec_2(C)$ the condition $\widehat{D} \in \Omega$ no longer appears in (4.3). Problem (4.3) is the so-called SWLRA problem.

The major contribution of this chapter is the extension of the concept and the algorithm presented in [61] to linearly structured matrices (i.e., the extension of unstructured WLRA to SWLRA problems). Problem (4.3) can also be interpreted as an extension of the STLS problems described in [1, 2, 23, 77] by allowing Δr to be larger than 1.

4.3 SWLRA problem for scalar Hankel matrices

From this point on we will focus on a specific Ω, since as will become clear further on, it is not possible to derive a general algorithm that can deal with any type of linearly structured matrices. In this section, the Hankel structure will be investigated since this is one of the most frequently occurring structures in signal processing applications. As a result Toeplitz matrices are also dealt with since they can be converted into Hankel matrices with a simple permutation of the rows, whereas the solution of the corresponding SWLRA problem does not change. When Ω represents the set of Hankel matrices, $vec_2(C)$, in problem (4.3), is a vector containing the different elements of the different antidiagonals of matrix C.

4.3.1 An equivalent unconstrained optimization problem

The first step in the derivation of an algorithm is to reformulate (4.3) into an equivalent double-minimization problem:

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^\top N=I}} \left(\min_{\substack{vec_2(\widehat{D})\\ \widehat{D}N=0}} \|D-\widehat{D}\|_V^2 \right). \tag{4.4}$$

The second step consists of finding a closed form expression $f(N)$ for the solution of the inner minimization $\min_{\substack{vec_2(\widehat{D})\\ \widehat{D}N=0}} \|D-\widehat{D}\|_V^2$. The latter is obtained as follows. Applying the technique of Lagrange multipliers to the inner minimization of (4.4) yields the Lagrangian

$$\psi(L,\widehat{D}) = vec_2(D-\widehat{D})^\top V vec_2(D-\widehat{D}) \ - tr(L^\top(\widehat{D}N)), \tag{4.5}$$

where $tr(C)$ stands for the trace of matrix C and L is the matrix of Lagrange multipliers. Using the equalities

$$\begin{aligned} vec(A)^\top vec(B) &= tr(A^\top B),\\ vec(ABC) &= (C^\top \otimes A)vec(B), \end{aligned}$$

(4.5) becomes

$$\psi(L,\widehat{D}) = vec_2(D-\widehat{D})^\top V vec_2(D-\widehat{D}) \ - vec(L)^\top(N^\top\otimes I)vec(\widehat{D}). \tag{4.6}$$

Note that when $\widehat{D}$ is a linearly structured matrix it is straightforward to write down a relation between $vec(\widehat{D})$ and its minimal vector representation $vec_2(\widehat{D})$:

$$vec(\widehat{D}) = H vec_2(\widehat{D}), \tag{4.7}$$

where $H \in \mathbb{R}^{mn \times q}$ and q is the number of different elements in $\widehat{D}$. E.g., in the case of a Hankel matrix, $q = m + n - 1$ and $vec_2(\widehat{D})$ can be constructed from the first column and last row of $\widehat{D}$. Substituting (4.7) in (4.6), the following expression is obtained for the Lagrangian:

$$\psi(L, \widehat{D}) = vec_2(D - \widehat{D})^\top V vec_2(D - \widehat{D}) \; - vec(L)^\top (N^\top \otimes I_m) H vec_2(\widehat{D}). \quad (4.8)$$

Setting the derivatives of ψ with respect to $vec_2(\widehat{D})$ and L equal to 0 yields the following set of equations:

$$\begin{bmatrix} 2V & -H^\top (N \otimes I_m) \\ (N^\top \otimes I_m) H & 0 \end{bmatrix} \begin{bmatrix} vec_2(\widehat{D}) \\ vec(L) \end{bmatrix} = \begin{bmatrix} 2V vec_2(D) \\ 0 \end{bmatrix}. \quad (4.9)$$

Using the fact that

$$\begin{bmatrix} A & -B \\ B^\top & 0 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} - A^{-1}B(B^\top A^{-1} B)^{-1} B^\top A^{-1} & * \\ -(B^\top A^{-1} B)^{-1} B^\top A^{-1} & * \end{bmatrix}$$

and setting $H_2 \equiv (N \otimes I_m)^\top H$, it follows from (4.9) that

$$\begin{aligned} vec_2(\widehat{D}) &= \left(I_q - V^{-1} H_2^\top (H_2 V^{-1} H_2^\top)^{-1} H_2\right) vec_2(D) \Rightarrow \\ vec_2(D - \widehat{D}) &= V^{-1} H_2^\top (H_2 V^{-1} H_2^\top)^{-1} H_2 vec_2(D) \end{aligned} \quad (4.10)$$

As a result, the double-minimization problem (4.4) can be written as the following optimization problem:

$$\min_{\substack{N \in \mathbb{R}^{n \times (n-r)} \\ N^\top N = I}} vec_2(D)^\top H_2^\top (H_2 V^{-1} H_2^\top)^{-1} H_2 vec_2(D), \quad (4.11)$$

with $H_2 \equiv (N \otimes I_m)^\top H$.

4.3.2 Algorithm

The straightforward approach for solving (4.4) would be to apply a nonlinear least squares (NLLS) solver to (4.11). For $\Delta r = 1$ this works fine but when $\Delta r > 1$ this approach breaks down by yielding the trivial solution $\widehat{D} = 0$. This can easily be understood by considering (4.4) with V equal to the identity matrix. As can be seen from (4.10), the inner minimization of (4.4) corresponds to an orthogonal projection of $vec_2(D)$ on the orthogonal complement of the column space of $H_2^\top \in \mathbb{R}^{(m+n-1) \times m\Delta r}$. Therefore it is clear that for $\Delta r = 1$ the orthogonal complement of the column space of $H_2^\top$ will never be empty (the dimension of the column space of $H_2^\top = \text{rank}(H_2^\top) = \text{m}$) but for $\Delta r > 1$ the dimension of the column space of $H_2^\top$ is equal to $m + n - 1$ since $m \geq n$. Hence, the dimension of the orthogonal complement of the column space of $H_2^\top$ will typically be zero. As a result the projection $vec_2(\widehat{D})$ on the latter subspace

will yield $\widehat{D} = 0$. The problem lies in an overparameterization of the matrix $H_2^\top = H^\top(N \otimes I_m)$.
To find a solution to the latter problem we first state the inner minimization in (4.4) in words: "Given a matrix N whose column space is a nullspace of a non-trivial Hankel matrix, find the Hankel matrix $\widehat{D}$ closest to D such that the nullspace of $\widehat{D}$ is spanned by the column space of N". A parameterization of the nullspace of a Hankel matrix can be found by means of the so-called Vandermonde decomposition[1] of a Hankel matrix. Given a rank deficient Hankel matrix $\widehat{D} \in \mathbb{R}^{m\times n}$ of rank r, it is straightforward to see that it can be parameterized as follows:

$$\widehat{D} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_r \\ \vdots & \vdots & & \vdots \\ z_1^{m-1} & z_2^{m-1} & \dots & z_r^{m-1} \end{bmatrix} \begin{bmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_r \end{bmatrix} \begin{bmatrix} 1 & z_1 & \dots & z_1^{n-1} \\ 1 & z_2 & \dots & z_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_r & \dots & z_r^{n-1} \end{bmatrix}, \tag{4.12}$$

with $z_i, i = 1, \dots, r$ the so-called complex signal poles and $c_i, i = 1, \dots, r$ the so-called complex amplitudes.
By defining a vector $p = [p_1\, p_2 \dots p_{r+1}]^\top$ such that the following polynomial in λ:

$$p_1 + p_2\lambda + \dots + p_{r+1}\lambda^r$$

has roots $z_i, i = 1, \dots, r$, it is easy to see that the nullspace of $\widehat{D}$ can be parameterized by $p_i, i = 1, \dots, r+1$ in the following matrix whose rows span the nullspace of $\widehat{D}$:

$$\begin{bmatrix} p_1 & p_2 & \dots & p_{r+1} & 0 & 0 & \dots & 0 \\ 0 & p_1 & p_2 & \dots & p_{r+1} & 0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \dots & 0 & p_1 & p_2 & \dots & p_{r+1} \end{bmatrix} \in \mathbb{R}^{(n-r)\times n}$$

We could now use the previous matrix to parameterize N in (4.11) but a different approach is taken here. First note that from (4.12) it follows that the vector $vec_2(\widehat{D})$ (with $\widehat{D}$ being a rank r Hankel matrix) can be written as the following linear combination:

$$vec_2(\widehat{D}) = c_1 b_1 + c_2 b_2 + \dots + c_r b_r,$$

with vectors $b_i \equiv [1\ z_i\ \dots\ z_i^{m+n-2}]^\top$ and c_i scalar coefficients for $i = 1, \dots, r$. As a result, it is clear that the rows of the following matrix

$$H_3 \equiv \begin{bmatrix} p_1 & p_2 & \dots & p_{r+1} & 0 & 0 & \dots & 0 \\ 0 & p_1 & p_2 & \dots & p_{r+1} & 0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \dots & 0 & p_1 & p_2 & \dots & p_{r+1} \end{bmatrix} \in \mathbb{R}^{(m+n-1-r)\times(m+n-1)}$$

[1] This is only true for the generic cases in which $z_i \neq z_j$ for $i \neq j$.

span the space perpendicularly to $b_i, i = 1, \ldots, r$. Therefore -bearing in mind the goal of the inner minimization of (4.4)- the rank deficient Hankel matrix $\widehat{D}$ closest to D can be found by means of the following orthogonal projection:

$$vec_2(\widehat{D}) = \left(I - V^{-1}H_3^\top (H_3 V^{-1} H_3^\top)^{-1} H_3\right) vec_2(D), \tag{4.13}$$

and (4.11) thus becomes

$$\min_{p \in \mathbb{R}^{(r+1)\times 1}} \; vec_2(D)^\top H_3^\top (H_3 V^{-1} H_3^\top)^{-1} H_3 vec_2(D), \tag{4.14}$$

with $p \equiv [p_1\, p_2 \ldots p_{r+1}]^\top$. Remark that the new expressions are similar as before, although we now work with a *vector* p instead of a *matrix* N.

The algorithm for solving the SWLRA for Hankel matrices can therefore be summarized in Algorithm 10.

Algorithm 10 SWLRA algorithm

1: Input: Hankel matrix D with first column $= [d_1\ d_2\ \ldots d_m]^\top$ and last row $= [d_m\ \ldots d_{m+n-1}]$, $D \in \mathbb{R}^{m\times n}$, rank r and weighting matrix V.
2: Construct matrix D' by rearranging the elements of matrix D such that $D' \in \mathbb{R}^{(m+n-r-1)\times(r+1)}$ is a Hankel matrix with first column $= [d_1\ \ldots d_{m+n-r-1}]^\top$ and last row $= [d_{m+n-r-1}\ \ldots d_{m+n-1}]$.
3: Compute the SVD of D': $D' = U\Sigma V^\top$, with $V \in \mathbb{R}^{(r+1)\times(r+1)}$.
4: Take starting value p_0 equal to the $(r+1)$-right singular vector of D': $p_0 = v_{r+1}$, with v_{r+1} the last column of V.
5: Minimize the cost function $f(p)$ in (4.14) with

$$f(p) = vec_2(D)^\top H_3^\top (H_3 V^{-1} H_3^\top)^{-1} H_3 vec_2(D).$$

6: Compute $vec_2(\widehat{D})$ using (4.13):

$$vec_2(\widehat{D}) = \left(I - V^{-1}H_3^\top (H_3 V^{-1} H_3^\top)^{-1} H_3\right) vec_2(D).$$

7: Reconstruct matrix $\widehat{D}$ from $vec_2(\widehat{D})$.
8: Output: Hankel matrix $\widehat{D}$ of rank $\leq r$ such that $\widehat{D}$ is as close as possible to D in $\|\cdot\|_V$-sense.

In step 5 of Algorithm 10 a standard NLLS (Matlab's `lsqnonlin`) is used. In order to use a NLLS routine, the cost function has to be cast in the form $F^\top F$ and in order to do so the Cholesky decomposition of $H_3^\top (H_3 V^{-1} H_3^\top)^{-1} H_3$ has to be computed. This can be done by a QR factorisation of $V^{-1/2} H_3^\top$. The computationally most intensive step is step 5.

4.3.3 Numerical experiments

In this subsection we first consider a small SWLRA problem of which the solution is known in order to illustrate the correctness of the proposed algorithm Algorithm 10. Secondly, we compare the efficiency and statistical accuracy of the SWLRA algorithm, Algorithm 10, and the STLNB algorithm proposed in [95]. The STLNB algorithm is presented in Appendix D.

Benchmarks

To illustrate the numerical correctness of the SWLRA algorithm, we use two small SWLRA problems of which the exact solution can be calculated analytically. In [23], modelling problems have been discussed, which approximate a given data sequence $z_k \in \mathbb{C}^{m+n-1}$ by the impulse response $\hat{z}_k \in \mathbb{C}^{m+n-1}$ of a finite-dimensional linear time-invariant (LTI) system of a given order, yielding the following SWLRA problem:

$$\min_{\hat{z}_k} \sum_{k=1}^{m+n-1} \left((z_k - \hat{z}_k)w_k\right)^2 \text{ s.t. } \widehat{D} \begin{bmatrix} X \\ -I_d \end{bmatrix} = 0, \tag{4.15}$$

where $\widehat{D} \in \mathbb{C}^{m \times n}$, $m \geq n$, is a Toeplitz matrix constructed from $\hat{z}_k$ and w_k are appropriate weights, accounting for the repetitions of the elements $\hat{z}_k$ in the matrix $\widehat{D}$. The constraint imposing that $\widehat{D}$ has at most rank $n-d$ ensures that $\hat{z}_k$ is the impulse response of a finite-dimensional linear system of order $n-d$ at most.

As proven in [23], there is one special case for which the global minimum of (4.15) can be found explicitly. This is for the case when an approximation of a given data sequence z_k is sought by a first order LTI impulse response, which can be parameterized as $\hat{z}_k = \alpha\beta^{k-1}$. We then obtain as a minimization problem:

$$\min_{\alpha,\beta} \sum_{k=1}^{m+n-1} \left((z_k - \alpha\beta^{k-1})w_k\right)^2 = \min_{\alpha,\beta} g(\alpha,\beta). \tag{4.16}$$

Setting the derivatives with respect to α and β to zero, we obtain

$$\begin{aligned} \frac{\partial g}{\partial \alpha} = 0 \quad &\Rightarrow \quad \sum_{k=1}^{m+n-1} (z_k - \alpha\beta^{k-1})w_k^2(\beta^{k-1}) = 0, \\ \frac{\partial g}{\partial \beta} = 0 \quad &\Rightarrow \quad \sum_{k=2}^{m+n-1} (z_k - \alpha\beta^{k-1})w_k^2(-(k-1)\alpha\beta^{k-2}) = 0. \end{aligned}$$

Eliminating α from the first equation and substituting it in the second, we get

the following polynomial in β:

$$\left(\sum_{k=2}^{m+n-1}(k-1)w_k^2\beta^{k-2}\right)\left(\sum_{k=1}^{m+n-1}w_k^2\beta^{2k-2}\right)$$
$$-\left(\sum_{k=1}^{m+n-1}w_k^2\beta^{k-1}\right)\left(\sum_{k=2}^{m+n-1}(k-1)w_k^2\beta^{2k-3}\right)=0. \qquad (4.17)$$

This is a polynomial of degree $3(m+n-1)-5$ in β. One has to select the root of this polynomial which gives the minimum in (4.16).

Example 4.1 Consider the following LTI-system of order 2:

$$w_{k+1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} w_k,$$
$$z_k = [1\,0]w_k,$$
$$w_0 = \begin{bmatrix} 6 \\ 1 \end{bmatrix}.$$

△

We arranged the first six data samples z_k in the 5×2 Toeplitz matrix

$$D = \begin{bmatrix} 5 & 6 \\ 4 & 5 \\ 3 & 4 \\ 2 & 3 \\ 1 & 2 \end{bmatrix}$$

and determined the best rank 1 structure-preserving approximation $\widehat{D}$ to D by solving the corresponding SWLRA problem (4.15) with weights $(w_1^2, w_2^2, \ldots, w_6^2) = (1,2,2,2,2,1)$. We obtained the optimal solution with $\hat{z}_k/\hat{z}_{k-1} = 0.76292301$, for $k = 2,\ldots,6$ as determined in [23]. The result is only accurate up to 8 digits, since accuracy is lost due to the squaring effect of the data matrix D in the cost function of (4.14).

Example 4.2 Consider the following LTI-system of order 4:

$$w_{k+1} = diag(0.4, 0.3, 0.2, 0.1)w_k,$$
$$z_k = [1\,1\,1\,1]w_k,$$
$$w_0 = [1\,1\,1\,1]^\top.$$

△

The first eight samples z_k of the impulse response are arranged in a Toeplitz matrix D:

$$D = \begin{bmatrix} 0.1 & 0.3 & 1 & 4 \\ 0.0354 & 0.1 & 0.3 & 1 \\ 0.013 & 0.0354 & 0.1 & 0.3 \\ 0.00489 & 0.013 & 0.0354 & 0.1 \\ 0.00187 & 0.00489 & 0.013 & 0.0354 \end{bmatrix}.$$

In order to compute the best rank 1 structure-preserving approximation $\widehat{D}$ to D, we applied the SWLRA algorithm to problem (4.15) using the weights $(w_1^2, w_2^2, \ldots, w_8^2) = (1, 2, 3, 4, 4, 3, 2, 1)$. We obtained the same ratio $\hat{z}_k/\hat{z}_{k-1} = 0.26025661$ for $k = 2, \ldots, 8$, as for the true minimum of (4.15), obtained by finding the roots of polynomial (4.17). Here again, the accuracy is limited in the same way as before by the squaring effect. We lose accuracy because of the squaring effect of our cost function $F^\top F$ of (4.14) that has to be minimized by the SWLRA algorithm.

Statistical accuracy and numerical efficiency

To compare the statistical accuracy and the computational efficiency of the SWLRA algorithm and the STLNB algorithm on larger examples, we perform Monte-Carlo simulations. In Table 4.1, the results for a 30×20 Hankel matrix are presented, whereas Table 4.2 contains the results for a 50×30 Hankel matrix. The Monte-Carlo simulations are performed as follows. A Hankel matrix of the required rank, i.e., $r = n - \Delta r$ is constructed. The latter matrix is perturbed by a Hankel matrix that is constructed using a vector containing i.i.d. Gaussian noise of standard deviation 10^{-3}. Finally, the SWLRA and the STLNB algorithm, Algorithm 10 and Algorithm 15 respectively, are applied to this perturbed Hankel matrix. Each Monte-Carlo simulation consists of 100 runs, i.e., it considers 100 noisy realizations. The results presented are averaged over these runs and obtained by coding both algorithms in Matlab (version 6.1) and running on a PC i686 with 800 MHz and 256 MB memory.

From both tables we can conclude the following. The statistical accuracy of the SWLRA algorithm is worse than that of the STLNB algorithm for small Δr but for large Δr the SWLRA algorithm performs best. From the computational point of view, the SWLRA algorithm outperforms the STLNB algorithm for large Δr. The behaviour of the computational time for both algorithms can be explained as follows. The number of flops for minimizing the cost function (step 5) in the SWLRA algorithm, Algorithm 10, is dominated by a QR decomposition (NLLS solver) whose computational cost is of the order of $2n^2(m-n/3)$ for a $m \times n$ matrix, so in our case of the order of $2(r+1)^2(m+n-1-r-(r+1)/3)$. As a result, the number of flops decreases for decreasing r, or equivalently for increasing Δr. On the contrary, the cost of the STLNB algorithm is dominated by a QR factorization of a $(m + n - 1 + m\Delta r) \times (m + n - 1 + n\Delta r)$ matrix

Δr	SWLRA		STLNB	
	$\| D - \widehat{D} \|_F$	cputime (sec)	$\| D - \widehat{D} \|_F$	cputime (sec)
2	$9.527\ 10^{-5}$	1.12660	$9.235\ 10^{-5}$	0.07310
4	$8.726\ 10^{-5}$	1.07155	$8.453\ 10^{-5}$	0.11515
6	$7.813\ 10^{-5}$	1.01847	$7.535\ 10^{-5}$	0.18930
8	$6.881\ 10^{-5}$	0.99223	$1.172\ 10^{-3}$	1.02175
10	$6.460\ 10^{-5}$	0.96322	$9.460\ 10^{-4}$	0.96742
12	$5.819\ 10^{-5}$	0.95240	$9.284\ 10^{-4}$	1.71100
14	$5.292\ 10^{-5}$	0.92596	$7.987\ 10^{-4}$	1.59431
16	$4.812\ 10^{-5}$	0.88455	$7.005\ 10^{-4}$	1.50456
18	$4.333\ 10^{-5}$	0.82621	$6.232\ 10^{-4}$	1.38712

Table 4.1: Numerical results for the 30×20 matrix

Δr	SWLRA		STLNB	
	$\| D - \widehat{D} \|_F$	cputime (sec)	$\| D - \widehat{D} \|_F$	cputime (sec)
2	$8.803\ 10^{-5}$	4.61080	$8.470\ 10^{-5}$	0.20520
4	$7.689\ 10^{-5}$	4.06400	$7.367\ 10^{-5}$	0.87415
6	$7.148\ 10^{-5}$	4.04730	$6.820\ 10^{-5}$	2.86147
8	$7.118\ 10^{-5}$	4.15380	$2.096\ 10^{-3}$	7.69475
10	$6.714\ 10^{-5}$	4.02128	$1.686\ 10^{-3}$	7.29424
12	$6.475\ 10^{-5}$	3.96598	$1.521\ 10^{-3}$	8.59522
14	$3.027\ 10^{-4}$	3.99784	$1.585\ 10^{-3}$	15.22531
16	$3.318\ 10^{-4}$	3.86665	$1.722\ 10^{-3}$	22.32086
18	$2.995\ 10^{-4}$	3.80046	$1.743\ 10^{-3}$	28.51761
20	$2.723\ 10^{-4}$	3.62653	$1.571\ 10^{-3}$	27.52066
22	$2.504\ 10^{-4}$	3.55803	$1.892\ 10^{-3}$	27.93990
24	$2.308\ 10^{-4}$	3.40264	$1.756\ 10^{-3}$	27.04059
26	$2.212\ 10^{-4}$	3.27834	$1.641\ 10^{-3}$	25.48300
28	$2.080\ 10^{-4}$	3.12491	$1.528\ 10^{-3}$	23.93134

Table 4.2: Numerical results for the 50×30 matrix

requiring $2(m+n-1+n\Delta r)^2(m+n-1+m\Delta r-(m+n-1+n\Delta r)/3)$ flops, which results in an increasing number of flops for increasing Δr. However, in both algorithms the computational cost can significantly decrease by exploiting the matrix structure in the computation of the QR decomposition, e.g., by using displacement rank theory [53, 49].

4.4 SWLRA problem for block-row Hankel matrices

In this section the block-Hankel structured matrices with blocks consisting of row vectors of length s will be investigated. This is the basic matrix structure found in Multiple-Input Single-Output (MISO) system identification problems. In these problems the system impulse response needs to be estimated from given noisy measured input and output data acquired by, for example, a Magnetic Resonance scanner [43]. Due to the applications in this field, we will work with block-row Hankel matrices of which the elements are Markov parameters. These parameters define the relation between the input and the output of a system. E.g., for a single antenna y with s sources, the relation between the inputs $U(i) = [u_1(i)u_2(i)\dots u_s(i)]^\top$, where $u_k(i)$ is the kth input for $k = 1,\dots,s$, and the output y at time $i = 0, 1, \dots, m-1$ can be written as follows:

$$y(i) = h_0U(i) + h_1U(i-1) + \dots + h_{t-1}U(i-t+1),$$

where $h_l \in \mathbb{R}^{1\times s}$, for $l = 0,\dots,t-1$, are the given Markov parameters. In matrix form this relation becomes

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ \\ y(m-1) \end{bmatrix} = \begin{bmatrix} h_{t-1} & \dots & h_1 & h_0 & 0 & 0 & \dots & 0 \\ 0 & h_{t-1} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ & & \ddots & & & \ddots & & \\ & & & & & & & \\ 0 & \dots & 0 & 0 & h_{t-1} & \dots & h_1 & h_0 \end{bmatrix} \begin{bmatrix} U(-t+1) \\ U(-t+2) \\ \vdots \\ U(-1) \\ U(0) \\ \vdots \\ U(m-1) \end{bmatrix}.$$

The previous equation can be written as $Y = \mathcal{H}\mathcal{U}$, with $\mathcal{H}$ a block-row structured matrix composed of the previously mentioned Markov parameters. By focusing in this section on block-row Hankel matrices, also block-row Toeplitz matrices are dealt with since they can be converted into block-row Hankel matrices by a simple permutation of the rows. The solution of the corresponding SWLRA problem can be found by applying the same permutation to the solution of the Hankel case.

The main contribution of this section is the extension of the ideas presented in [61] to linearly structured matrices such as block-row Hankel or Toeplitz

matrices. A linearly structured matrix is a matrix that can be written as a linear combination of fixed basis matrices. Instead of using the algorithms in [61], we will use standard optimization routines to solve the extended problem formulation for reasons explained below. In the previous section, the SWLRA for a particular structure, namely the scalar Hankel structure was presented. When Ω represents the set of block-row Hankel matrices $D \in \mathbb{R}^{m \times st}$ of the form

$$D = \begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_t \\ d_2 & d_3 & \dots & & \\ d_3 & \dots & & & \\ \vdots & & & & \vdots \\ d_m & \dots & & & d_{m+t-1} \end{bmatrix} \tag{4.18}$$

with d_i a row vector of length s, for $i = 1, \dots, m+t-1$, $vec_2(D)$, in problem (4.3), is a vector equal to $vec_2(D) = [d_1 \dots d_{m+t-1}]^\top$.

The outline of this section is as follows. In subsection 4.4.1 a problem equivalent to the SWLRA problem involving block-row Hankel matrices is derived. Subsection 4.4.2 introduces an algorithm for solving the equivalent problem. In the last subsection some numerical results are presented to compare the SWLRA with the algorithm HTLSstack, described in [69, 90]. By fitting the Markov parameters (elements of a Hankel/Toeplitz matrix) to an exponential data model of order r, HTLSstack finds a suboptimal Hankel/Toeplitz matrix approximation of rank r by filling in the fitted elements instead of the original ones. Whereas HTLS [94] only applies to scalar Hankel/Toeplitz matrices, HTLSstack [69, 90] extends the HTLS algorithm to block-row Hankel/Toeplitz matrices. The difference between HTLSstack and TLS-ESPRIT [78] is that HTLSstack finds a suboptimal solution to (4.2) by making use of the Singular Value Decomposition (SVD) of the data matrix D, whereas TLS-ESPRIT finds the same solution by making use of the eigendecomposition of the corresponding sample covariance matrix [69].

4.4.1 An equivalent unconstrained optimization problem

As in the previous section we need to solve the following double-minimization problem, equivalent to problem (4.3):

$$\min_{\substack{N \in \mathbb{R}^{st \times (st-r)} \\ N^\top N = I}} \left(\min_{\substack{vec_2(\widehat{D}) \\ \widehat{D}N = 0}} \|D - \widehat{D}\|_V^2 \right). \tag{4.19}$$

The first step in the derivation consists of finding a closed form expression, $f(N)$, for the solution of the inner minimization of (4.19).

Theorem 1: Let $f(N)$ be defined as $f(N) = \min_{\substack{vec_2(\widehat{D}) \\ \widehat{D}N = 0}} \|D - \widehat{D}\|_V^2$, with $D \in \mathbb{R}^{m \times st}$ a given block-row Hankel matrix of the form (4.18), and $V \in \mathbb{R}^{q \times q}$ a

positive-definite, symmetric weighting matrix with q equal to the number of elements in $vec_2(\widehat{D})$. Then a closed-form expression for $f(N)$ is given by

$$f(N) = vec_2(D)^\top H_2^\top (H_2 V^{-1} H_2^\top)^{-1} H_2 vec_2(D), \tag{4.20}$$

where $H_2 \equiv (N \otimes I_m)^\top H$, and $H \in \mathbb{R}^{mst \times q}$ such that $vec(\widehat{D}) = H vec_2(\widehat{D})$. Note that H depends only on the structure of $\widehat{D}$. The minimizing $\widehat{D}$ is given by

$$\begin{aligned} vec_2(\widehat{D}) = (I_q - V^{-1} H^\top (N \otimes I_m)[(N \otimes I_n)^\top H V^{-1} H^\top (N \otimes I_m)]^{-1} \\ (N \otimes I_n)^\top H) vec_2(D). \end{aligned} \tag{4.21}$$

Proof : The reader is referred to subsection 4.3.1 for a complete derivation of equations (4.20) and (4.21). The proof is left out here for reasons of duplication. It is similar to the derivation of equations (4.10) and (4.11).

□

As a result of the theorem, the double-minimization problem (4.19) can be written as the following optimization problem:

$$\min_{\substack{N \in \mathbb{R}^{st \times (st-r)} \\ N^\top N = I}} vec_2(D)^\top H_2^\top (H_2 V^{-1} H_2^\top)^{-1} H_2 vec_2(D). \tag{4.22}$$

By replacing D with $\widetilde{D} \in \mathbb{R}^{s(m+t-r-1) \times (r+1)}$ (and similarly replacing $\widehat{D}$ by $\widehat{\widetilde{D}}$)

$$\widetilde{D} = \begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_t & \dots & d_{m+t-r-1} \\ d_2 & d_3 & \dots & & & & \\ d_3 & \dots & & & & & \\ \vdots & & & & & & \vdots \\ d_{r+1} & \dots & & & & \dots & d_{m+t-1} \end{bmatrix}^\top \tag{4.23}$$

with d_i a row vector of length s for $i = 1, \dots, m+t-1$ and by setting $\Delta r = 1$, a problem formulation equivalent to (4.19) can be derived.
Theorem 2: Problem formulation

$$\min_{\substack{p \in \mathbb{R}^{r+1} \\ p^\top p = 1}} \left(\min_{\substack{vec_2(\widehat{\widetilde{D}}) \\ \widehat{\widetilde{D}} p = 0}} \|\widetilde{D} - \widehat{\widetilde{D}}\|_V^2 \right) \tag{4.24}$$

is equivalent to the minimization problem (4.19) under the assumption that r is a multiple of s, where N is replaced by p, to indicate that it is a column vector instead of a matrix.

Proof : First note that the cost function in problem (4.24) is equal to the cost function in problem (4.19) since the $vec_2(\cdot)$ operator yields a minimal vector

representation which is the same in the case of D (resp. $\widehat{D}$) and $\widetilde{D}$ (resp. $\widehat{\widetilde{D}}$):

$$\begin{aligned}\|\widetilde{D}-\widehat{\widetilde{D}}\|_V^2 &= vec_2(\widetilde{D}-\widehat{\widetilde{D}})^\top V vec_2(\widetilde{D}-\widehat{\widetilde{D}})\\ &= vec_2(D-\widehat{D})^\top V vec_2(D-\widehat{D})\\ &= \|D-\widehat{D}\|_V^2.\end{aligned}$$

To prove the equivalence, we still have to show that the rank deficiency conditions are equivalent in the sense that they both yield the same parametric representation for the rank deficient matrix. A matrix $\widehat{D}\in\mathbb{R}^{m\times st}$ having the structure of D and rank r can be written as follows:

$$\widehat{D}=\begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_t \\ d_2 & d_3 & \dots & & \\ d_3 & \dots & & & \\ \vdots & & & & \vdots \\ d_m & \dots & & & d_{m+t-1}\end{bmatrix}$$

with $d_j=[d_{j1}\dots d_{js}]\in\mathbb{R}^{1\times s}$ for $j=1,\dots,m+t-1$. In system theoretic terms, $vec_2(\widehat{D})$ of a rank deficient matrix $\widehat{D}$ having the structure of D and rank r equal to a multiple of s is an impulse response of a system of order r [63, Lemma 1]. Hence, the blocks of $\widehat{D}$ can be parameterized by means of the state-space representative matrices as follows:

$$d_j=CA^{j-1}B,\quad j=1,\dots,m+t-1 \tag{4.25}$$

with $C\in\mathbb{R}^{1\times r}, B\in\mathbb{R}^{r\times s}, A\in\mathbb{R}^{r\times r}$. So, matrix $\widehat{D}$ can be written as follows:

$$\begin{aligned}\widehat{D} &= \begin{bmatrix} CB & CAB & \dots & CA^{t-1}B \\ CAB & CA^2B & \dots & CA^tB \\ \vdots & & & \vdots \\ CA^{m-1}B & \dots & & CA^{m+t-2}B\end{bmatrix}\\ &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{m-1}\end{bmatrix}\begin{bmatrix} B & AB & A^2B & \dots & A^{t-1}B\end{bmatrix}.\end{aligned}$$

By filling in the calculated values of the parametric matrices C, A and B for d_j, $j=1,\dots,m+t-1$ obtained from $\widehat{D}$ into the parameterization of the blocks of $\widehat{\widetilde{D}}$, we also find a solution of (4.24):

$$\widehat{\widetilde{D}}=\begin{bmatrix} d_1^\top & d_2^\top & d_3^\top & \dots & d_{r+1}^\top \\ d_2^\top & d_3^\top & \dots & & \\ d_3^\top & \dots & & & \\ \vdots & & & & \vdots \\ d_{m+t-r-1}^\top & \dots & & & d_{m+t-1}^\top\end{bmatrix}$$

$$= \begin{bmatrix} B^\top \\ B^\top A^\top \\ B^\top (A^\top)^2 \\ \vdots \\ B^\top (A^\top)^{m+t-r-2} \end{bmatrix} \begin{bmatrix} C^\top & A^\top C^\top & (A^\top)^2 C^\top & \dots & (A^\top)^r C^\top \end{bmatrix},$$

where the second equality follows from (4.25). Because we form $\widehat{\widetilde{D}}$ using the matrices (A, B, C) calculated from d_j, $j = 1, \dots, m+t-1$ obtained from $\widehat{D}$, $\widehat{\widetilde{D}}$ has the same structure as $\widetilde{D}$ and the rank of this matrix $\widehat{\widetilde{D}} \in \mathbb{R}^{s(m+t-r-1)\times(r+1)}$ is equal to r. So, a solution of (4.24) can be found. The proof is similar for the other way around. □

For this equivalent problem (4.24) formulations (4.22) and (4.21) become:

$$\min_{\substack{p\in\mathbb{R}^{r+1} \\ p^\top p=1}} vec_2(\widetilde{D})^\top \widetilde{H}_2^\top (\widetilde{H}_2 V^{-1} \widetilde{H}_2^\top)^{-1} \widetilde{H}_2 vec_2(\widetilde{D}), \tag{4.26}$$

with $\widetilde{H}_2 \equiv (p \otimes I)^\top \widetilde{H} \in \mathbb{R}^{s(m+t-r-1)\times s(m+t-1)}$ and I the identity matrix of size $s(m+t-r-1)$.
A formulation of the vectorized form of $\widehat{\widetilde{D}}$ is equal to:

$$vec_2(\widehat{\widetilde{D}}) \quad = \quad \left(I_q - V^{-1}\widetilde{H}_2^\top (\widetilde{H}_2 V^{-1} \widetilde{H}_2^\top)^{-1} \widetilde{H}_2\right) vec_2(\widetilde{D}). \tag{4.27}$$

Note that the proof of Theorem 2 was based on the particular parameterization of a rank deficient block-row Hankel matrix. If the matrix has a more general linear structure, first a new parameterization of that type of linear structure has to be derived before any new (and equivalent) problem formulation can be proposed.

4.4.2 Algorithm

The cost function in the optimization problem (4.22) is non-convex. A simple counter example is the function:

$$f(n_1, n_2) = \frac{2(n_1+n_2)^2(n_1^2+n_2^2-n_1n_2)}{(n_1^4+n_1^2n_2^2+n_2^4)},$$

which is a special case of the cost function in (4.22) for

$$D = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; N = [n_1\, n_2]^\top \text{ and } V = I_3.$$

Due to the non-convexity of the problem, we consider a standard method for *local* optimization: the Levenberg-Marquardt algorithm, which is a nonlinear

least squares optimization algorithm. The minimization algorithms in [61] are not used. In order to use those algorithms, the gradient and the Hessian of the cost function in (4.22) are needed and they are more complicated to compute than in [61], especially the Hessian. Moreover, our main goal in this thesis is the extension of the concept in [61] to block Hankel matrices (and not the extension of the algorithms in [61]). In order to use the standard Levenberg-Marquardt algorithm, the cost function f must be recast in the form $F^\top F$. Let $\mathcal{L}^\top\mathcal{L}$ be the Cholesky factorization of $H_2^\top(H_2V^{-1}H_2^\top)^{-1}H_2$. Then f is equal to $F^\top F$ with $F := \mathcal{L}\,vec_2(D)$ (Note that the evaluation of $F(N)$ is cheaper than the one of $f(N)$). The performance of the algorithm is discussed in subsection 4.4.3. We will explain how we build in the normalization constraint $N^\top N = I$ and how we deal with the local minima of our cost function.

In this section the algorithm is derived. Using a standard nonlinear least squares (NLLS) solver for (4.22), good results are obtained for $\Delta r = 1$. When $\Delta r > 1$, this approach always yields the trivial solution, $\widehat{D} = 0$. The problem lies in an overparametrization of $H_2^\top$ as explained in subsection 4.3.2. To avoid this problem, we will consider the equivalent optimization problem (4.26) with $\Delta r = 1$.

The algorithm for solving the SWLRA for block-row Hankel matrices is summarized in Algorithm 11.

In step 5 a standard NLLS (Matlab's `lsqnonlin`) is used. In order to use a NLLS routine, the cost function must be recast in the form $f^\top f$ and in order to do so the Cholesky decomposition of $\widetilde{H}_2^\top(\widetilde{H}_2V^{-1}\widetilde{H}_2^\top)^{-1}\widetilde{H}_2$ has to be computed. This can be done by means of a QR factorization of $V^{-1/2}\widetilde{H}_2^\top$. The computationally most expensive step in the algorithm is step 5.

4.4.3 Numerical results

In this subsection we compare the statistical accuracy of algorithm block-row SWLRA and the algorithm HTLSstack proposed in [69]. The HTLSstack algorithm is presented in Appendix E. We expect that the block-row SWLRA algorithm will perform better than the algorithm HTLSstack. Indeed, SWLRA seeks for the closest lower-rank Hankel/Toeplitz matrix by solving (4.26), while HTLSstack only computes a suboptimal Hankel/Toeplitz matrix estimate by exploiting the shift-invariant structure of the block-row Hankel matrix. For the comparison, we perform Monte-Carlo simulations.

In Table 4.3, the results for a 25×12 block-row Hankel matrix D are presented with blocks of length 2. Table 4.4 contains the results for a 50×24 block-row Hankel matrix with blocks of length 4. The Monte-Carlo simulations are performed as follows. A block-row Hankel matrix of the required rank r is constructed. The latter matrix is perturbed by a block-row Hankel matrix

r	σ	SWLRA	HTLSstack
2	10^{-3}	0.00095	0.00096
	10^{-2}	0.00941	0.00950
	0.1	0.11534	0.11622
	1.0	0.63870	0.64713
4	10^{-3}	0.00105	0.00107
	10^{-2}	0.00932	0.00936
	0.1	0.10341	0.10452
	1.0	0.58715	0.59550
6	10^{-3}	0.00091	0.00093
	10^{-2}	0.00929	0.00965
	0.1	0.07790	0.08497
	1.0	0.55468	0.59148
8	10^{-3}	0.00088	0.00090
	10^{-2}	0.00652	0.00712
	0.1	0.07417	0.07476
	1.0	0.54678	0.67650
10	10^{-3}	0.00069	0.00075
	10^{-2}	0.00781	0.00873
	0.1	0.07522	0.08262
	1.0	0.52649	0.71059

Table 4.3: Approximation of a 25×12 block-Hankel matrix D with 1×2 blocks by a matrix $\widehat{D}$ of rank r. Comparison of the relative norm $\frac{\|D-\widehat{D}\|_F}{\|D\|_F}$, computed via the algorithms block-row SWLRA and HTLSstack.

r	σ	SWLRA	HTLSstack
4	10^{-3}	0.00092	0.00093
	10^{-2}	0.00917	0.00929
	0.1	0.09080	0.09201
	1.0	0.63603	0.64525
8	10^{-3}	0.00087	0.00091
	10^{-2}	0.00872	0.00909
	0.1	0.08645	0.08935
	1.0	0.60860	0.69709
12	10^{-3}	0.00080	0.00085
	10^{-2}	0.00801	0.00852
	0.1	0.07879	0.08337
	1.0	0.57914	0.68251
16	10^{-3}	0.00082	0.00091
	10^{-2}	0.00820	0.00902
	0.1	0.08079	0.09395
	1.0	0.54452	0.77840
20	10^{-3}	0.00077	0.00120
	10^{-2}	0.00690	0.00810
	0.1	0.06773	0.07665
	1.0	0.48384	0.69717

Table 4.4: Approximation of a 50×24 block-Hankel matrix D with 1×4 blocks by a matrix $\widehat{D}$ of rank r. Comparison of the relative norm $\frac{\|D-\widehat{D}\|_F}{\|D\|_F}$, computed via the algorithms block-row SWLRA and HTLSstack.

Algorithm 11 block-row SWLRA algorithm

1: Input: block-row Hankel matrix $D \in \mathbb{R}^{m \times st}$ of rank k, in which the blocks are row vectors $a_1, a_2, \ldots, a_{m+t-1}$ of length s, a desired rank r, and a weighting matrix V.
2: Construct matrix $\widetilde{D}$ by rearranging the elements of matrix D such that $\widetilde{D} \in \mathbb{R}^{s(m+t-r-1)\times(r+1)}$ is a block Hankel matrix with blocks given by the column vectors $a_1^\top, \ldots, a_{m+t-1}^\top$ (see (4.23)).
3: Compute SVD of $\widetilde{D}$: $\widetilde{D} = U\Sigma V^\top$.
4: Initialize p_0 with the right singular vector of $\widetilde{D}$ corresponding to its $(r+1)$th singular value.
5: Minimize the cost function $f(p)$ in (4.26):

$$f(p) = vec_2(\widetilde{D})^\top \widetilde{H}_2^\top (\widetilde{H}_2 V^{-1} \widetilde{H}_2^\top)^{-1} \widetilde{H}_2 vec_2(\widetilde{D}).$$

6: Compute $vec_2(\widehat{\widetilde{D}})$ using (4.27):

$$vec_2(\widehat{\widetilde{D}}) = \left(I_q - V^{-1}\widetilde{H}_2^\top (\widetilde{H}_2 V^{-1} \widetilde{H}_2^\top)^{-1} \widetilde{H}_2\right) vec_2(\widetilde{D}).$$

7: Reconstruct $\widehat{\widetilde{D}}$ from $vec_2(\widehat{\widetilde{D}})$.
8: Rearrange the elements of $\widehat{\widetilde{D}}$ into a matrix $\widehat{D}$ such that $\widehat{D} \in \mathbb{R}^{m\times st}$ and $\widehat{D}$ has the same structure as D (i.e. similar rearrangement as going from $\widetilde{D}$ to D in (4.23)).
9: Output: block-row Hankel matrix $\widehat{D}$ of rank at most r with $r < k$, such that $\widehat{D}$ is as close as possible to D in $\|\cdot\|_V$-sense.

that is constructed using a vector containing i.i.d. Gaussian noise of standard deviation σ. Finally, the block-row SWLRA algorithm, Algorithm 11, and the HTLSstack algorithm, Algorithm 16, are applied to this perturbed block-row Hankel matrix. To apply the block-row SWLRA algorithm, we need to choose the initial condition p_0 very carefully, because the convergence of the algorithm depends on it due to the appearance of local minima. A good choice for p_0 is the one taken in step 4 of Algorithm 11. In this way p_0 belongs to the nullspace of matrix $\widetilde{D}$. The normalization constraint $p^\top p = 1$ is not implemented in the `lsqnonlin` routine of Matlab, but instead we used the constraint $p(r+1) = -1$. Due to the scaling invariance of the cost function in (4.26) this is allowed. We exclude non-generic cases where $p(r+1) = 0$ at the solution. Instead of minimizing the cost function $f(p)$ in (4.26), we will reduce the dimension of the optimization problem by one by defining a cost function $f(q)$ with $p = [q^\top \; -1]^\top \in \mathbb{R}^{r+1}$. As initial condition for the minimization of $f(q)$ we take $q_0 = -p_0(1:r)/p_0(r+1)$. Hence, the last coordinate of the variable p is set to -1. Each Monte-Carlo simulation consists of 100 runs, i.e. considers 100 noisy realizations, for every considered σ and r. For both

algorithms the relative error $\frac{\|D-\widehat{D}\|_F}{\|D\|_F}$ is presented. The error is fully explained by the stochastic variation of the reduced order model due to the noise added. Moreover, we checked the Lagrangian partial derivative equations (4.9) for the approximation solution $\widehat{D}$. Because these equations are fulfilled, we can confidently say that our optimization problem $\| D-\widehat{D} \|_V^2$ is solved to machine precision. The results presented are averaged over the runs and obtained by implementing both algorithms in Matlab (version 6.1) on a PC i686 with 800 MHz and 256 MB memory. From both tables we can conclude the following. As expected, the block-row SWLRA algorithm performs better than the algorithm HTLSstack for any choice of r.

4.5 Conclusion

In this chapter we have developed an extension of the Weighted Low Rank Approximation problem introduced by Manton et al. [61] to linearly structured matrices. For particular types of structure, namely the Hankel and the block-row Hankel structure, algorithms were developed. The numerical accuracy of the SWLRA algorithm for Hankel matrices was tested on a benchmark problem. Furthermore, the statistical accuracy and the computational efficiency are compared for the STLNB [95] and the SWLRA algorithm. For both properties, the SWLRA algorithm performs better than the STLNB algorithm for large Δr. Simulation experiments confirm the improved statistical accuracy of the SWLRA algorithm for block-row Hankel matrices compared to that of the known algorithm HTLSstack.

Chapter 5

The shape from moments problem

In this chapter, we discuss the problem of recovering the vertices of a planar polygon from its measured complex moments. Because the given, measured moments can be noisy, the recovered vertices are only estimates of the true ones. The literature offers many algorithms for solving such an estimation problem. We will restrict our discussion to the Total Least Squares (TLS) data fitting models Hankel TLS (HTLS) and Structured TLS (STLS), the matrix pencil method Generalized Pencil of Function (GPOF) and a tensor-based algorithm. We show the close link between the HTLS and the GPOF methods and use the HTLS method to compute starting values for the STLS method. We compare the statistical accuracy of these four methods on simulated data. The data matrix formed by the given sequence of complex moments will be Hankel structured. We will show that the vertices estimated via the STLS algorithm are more accurate than the ones estimated via the pencil method because the STLS method preserves the Hankel structure of the matrix. Through experiments it will become clear that the parameter accuracy may improve by arranging the data sequence in a higher-order tensor and estimating the model parameters via a multilinear generalization of the SVD.

5.1 Introduction

As it turns out, the formulation of the problem of recovering the vertices of a planar polygon from its measured complex moments is very similar to several other diverse applications found in the literature, such as identifying an

auto-regressive system using its output, decomposing a signal built as a linear mixture of complex exponentials and estimating the direction of arrival in array processing. All these applications lead to the very same formulation and, therefore, to the same estimation problem when noise is involved. The literature offers many algorithms for solving such an estimation problem. We will discuss the Hankel Total Least Squares (HTLS) method [94], the Structured TLS (STLS) method [52, 55, 54], the Generalized Pencil of Function (GPOF) method [45] and a tensor-based algorithm [72] for the reconstruction of binary polygons from their given, estimated, noisy complex moments. In [67, 39, 42], the importance and relevance is discussed of the problem to recover the vertices of a planar polygon from its measured complex moments. First of all, Milanfar et al. [67] introduced the problem of reconstructing a planar polygon from a set of its complex moments. Moreover, by exploiting the relationship of this shape-from-moments problem with similar problems in signal and array processing, a number of algorithms based on Prony's method were obtained, which could be applied to the reconstruction problem. Then, better numerical procedures, based upon matrix pencils, were proposed for the shape-from-moments reconstruction problem, described in [39]. Also, an analysis of the sensitivity of the shape-from-moments problem is presented in this reference. Later on, instead of concentrating mainly on the numerical aspects of the noiseless case, in [26] the treatment of the reconstruction problem is extended to a given noisy but longer set of moments.

The outline of this chapter is as follows. In section 5.2, the shape-from-moments problem is formulated. In section 5.3, a short description is given of the GPOF method, the HTLS method, the STLS method and the tensor-based method. These methods can be used to solve the shape-from-moments problem. In this chapter, we will show the link between the HTLS and the GPOF method. We will compare the HTLS, the STLS, the GPOF and the tensor-based method on simulated data and discuss their statistical accuracy. This will be the topic of section 5.5. It will become clear that it is useful to compute the HTLS solution in order to get starting values for the STLS method. The STLS method is an optimal method and we expect to get a better accuracy than with the GPOF method. Conclusions are drawn in section 5.6.

5.2 Problem formulation

The reconstruction problem of a closed N-sided planar polygon from a set of its complex moments can be described as follows. An arbitrary closed N-sided planar polygon P is assumed. Its vertices are denoted by z_n with $n = 1, \ldots, N$, where these values are scalar and complex. Based on Davis' Theorem [15, 16], there exists a set of N coefficients a_n, $n = 1, \ldots, N$, depending only on the

vertices, such that for any analytic function $f(z)$ in the closure of P, we have

$$\int\int_P f''(z)dxdy = \sum_{n=1}^{N} a_n f(z_n). \tag{5.1}$$

Davis' Theorem shows that the coefficients a_n with $n = 1, \ldots, N$ are related to the vertices via the equation

$$a_n = \frac{i}{2}\left(\frac{\bar{z}_{n-1} - \bar{z}_n}{z_{n-1} - z_n} - \frac{\bar{z}_n - \bar{z}_{n+1}}{z_n - z_{n+1}}\right),$$

where $i = \sqrt{-1}$ and $\bar{z}$ is the complex conjugate of z. Note that since the polygon is closed, we define $z_{N+n} = z_n$ for all n.

A special case of interest is obtained for the analytic function $f(z) = z^k$. Using (5.1), we get

$$\begin{aligned}\int\int_P f''(z)dxdy &= k(k-1)\int\int_P z^{k-2}dxdy \\ &= \sum_{n=1}^{N} a_n f(z_n) = \sum_{n=1}^{N} a_n z_n^k.\end{aligned}$$

The expression $\int\int_P z^{k-2}dxdy$ stands for the $(k-2)$th moment computed over the indicator function defined as 1 inside the polygon and zero elsewhere. We denote $k(k-1)\int\int_P z^{k-2}dxdy$ as the complex moment τ_k. Clearly, by definition, we have that $\tau_0 = \tau_1 = 0$.

Now that we have defined complex moments, we are ready to formulate the reconstruction problem of a closed N-sided planar polygon from a set of its complex moments as follows. Assume that $M + 1$ complex moments τ_k with $k = 0, \ldots, M$ are measured. We want to recover the polygon vertices z_n with $n = 1, \ldots, N$ using the following relationship between moments and vertices:

$$\tau_k = \sum_{n=1}^{N} a_n z_n^k, \tag{5.2}$$

with a_n for $n = 1, \ldots, N$ coefficients that only depend on the vertices. We assume that the number of vertices N is known. In the presence of noise, equation (5.2) does not have an exact solution. In the rest of this chapter we assume that the measured complex moments are perturbed by white Gaussian noise. Hence, equation (5.2) leads to an estimation problem.

5.3 Methods to solve the shape-from-moments problem

We briefly describe the GPOF method, the HTLS method, the STLS method and a tensor-based method in this section. The literature offers many other algorithms to solve the shape-from-moments problem. We refer the interested reader to section 5.4 for more information about related work presented in [10, 60, 7, 34]. In this section, we will discuss the GPOF method, the HTLS method, the STLS method and a tensor-based method. The GPOF and the HTLS method are both non-iterative subspace-based parameter estimation methods. In this section, we will show that their approaches differ in the way of reducing the dimensionality of the shape-from-moments problem and in the way of solving the low-dimensional core problem. On the other hand, the STLS method is an iterative method that preserves the Hankel structure of the matrix formed by the given data sequence τ_k with $k = 0, \ldots, M$. In section 5.5, it will become clear that the vertices estimated via the STLS method are more accurate than the ones estimated via the GPOF method and the HTLS method. And finally, we will also apply the tensor approach to the shape-from-moments problem. The problem of harmonic analysis was for the first time cast in a tensor framework in [47, 83]. In section 5.5, it is shown that the parameter accuracy may improve by arranging the data sequence in a higher-order tensor and estimating the model parameters via a multilinear generalization of the SVD.

5.3.1 GPOF

In this subsection a short description of the GPOF method is given. This method was presented originally by Hua and Sarkar [45] and used later on in [58, 57, 76]. From the data sequence τ_k with $k = 0, \ldots, M$ a Hankel matrix H with $M - L + 1$ rows, $(M + 1)/2 \geq L \geq N$, is constructed as follows:

$$H = \begin{bmatrix} \tau_0 & \tau_1 & \tau_2 & \cdots & \tau_L \\ \tau_1 & \tau_2 & \cdots & & \tau_{L+1} \\ \tau_2 & \cdots & & & \tau_{L+2} \\ \vdots & & & & \\ \tau_{M-L} & \tau_{M-L+1} & \cdots & & \tau_M \end{bmatrix}. \tag{5.3}$$

The GPOF method makes use of two Hankel matrices T_0 and T_1 to compute the N vertices z_n, $n = 1, \ldots, N$. The matrix T_0 contains the first L columns of matrix H, expressed in (5.3), and matrix T_1 is equal to the submatrix of H, which contains the last L columns. Starting from (5.2), there exists a relationship between the matrices T_0 and T_1 which leads to the GPOF method. Define $t_{\{k_1,k_2\}}$ as the column vector of length k_2 containing the complex moments starting with τ_{k_1}. Define also $\mathbf{V}_{\{k_1,k_2\}}$ as the Vandermonde matrix of

size $k_2 \times N$ built from the vertices z_n, $n = 1, \ldots, N$ with powers starting with k_1. Finally, define the vector $\mathbf{a}$ as a column vector of length N containing the parameters a_n, $n = 1, \ldots, N$. Then, equation (5.2) can be rewritten as

$$t_{\{0,M+1\}} = \mathbf{V}_{\{0,M+1\}}\mathbf{a}. \tag{5.4}$$

We now show that the matrices T_0 and T_1 satisfy a pencil relationship. By using equation (5.4), matrix T_0 can be expressed as follows:

$$T_0 = \mathbf{V}_{\{0,M-L+1\}}\mathrm{Diag}(\mathbf{a})\mathbf{V}_{\{0,L\}}^{\top}.$$

In the above equation, we define the operator Diag as the construction of a diagonal matrix from a given vector. In an analog way, matrix T_1 can be expressed as follows:

$$T_1 = \mathbf{V}_{\{0,M-L+1\}}\mathrm{Diag}(\mathbf{a})\mathrm{Diag}(\mathbf{z})\mathbf{V}_{\{0,L\}}^{\top},$$

where the matrix $\mathrm{Diag}(\mathbf{z})$ is an $N \times N$ diagonal matrix with z_n, $n = 1, \ldots, N$ on its main diagonal. The Vandermonde matrix $\mathbf{V}_{\{0,L\}}^{\top}$ has N rows and $L \geq N$ columns and, due to the nondegenerate polygon, it is of full rank N. Thus, we have

$$\begin{aligned}
& T_1\mathbf{V}_{\{0,L\}}(\mathbf{V}_{\{0,L\}}^{\top}\mathbf{V}_{\{0,L\}})^{-1} \\
= \; & \mathbf{V}_{\{0,M-L+1\}}\mathrm{Diag}(\mathbf{a})\mathrm{Diag}(\mathbf{z}) \\
= \; & \mathbf{V}_{\{0,M-L+1\}}\mathrm{Diag}(\mathbf{a})\mathbf{V}_{\{0,L\}}^{\top}\mathbf{V}_{\{0,L\}}(\mathbf{V}_{\{0,L\}}^{\top}\mathbf{V}_{\{0,L\}})^{-1}\mathrm{Diag}(\mathbf{z}) \\
= \; & T_0\mathbf{V}_{\{0,L\}}(\mathbf{V}_{\{0,L\}}^{\top}\mathbf{V}_{\{0,L\}})^{-1}\mathrm{Diag}(\mathbf{z}).
\end{aligned} \tag{5.5}$$

Here, we have used the fact that $(\mathbf{V}_{\{0,L\}}^{\top}\mathbf{V}_{\{0,L\}})$ is a square $N \times N$ full rank matrix, thus, it is invertible. From equation (5.5), it is obvious that the matrices T_0 and T_1 indeed satisfy a pencil relationship. Moreover, this relationship implies that for the pair of matrices T_0 and T_1 the vertices z_n, $n = 1, \ldots, N$ are their generalized eigenvalues.

So, it is clear now that, for the GPOF method, we need to form the two Hankel matrices T_0 and T_1 from the given sequence of moments τ_k, $k = 0, \ldots, M$ and then solve the following generalized eigenvalue problem for their generalized eigenvalues:

$$(T_1 - \lambda T_0)v = 0. \tag{5.6}$$

In order to solve this rectangular generalized eigenvalue problem, the GPOF method makes use of the squaring idea of multiplying both sides of relationship (5.6) with T_0^{+}, where T^{+} denotes the Moore-Penrose pseudo-inverse of T. From the obtained square $L \times L$ pencil, only N eigenvalues will correspond to the vertices we are looking for. The remaining $L - N$ refer to the common null space of the matrices $T_0^{+}T_1$ and $T_0^{+}T_0$. So, the eigenvectors v of interest for the $L \times L$ pencil should be spanned by Y_1, where Y_1 consists of the first N columns

of Y, defined by the Singular Value Decomposition (SVD) WDY^H of matrix T_0. Hence, the following $L \times N$ pencil is obtained:

$$(T_0^+ T_1 - \lambda T_0^+ T_0) Y_1 \alpha = 0.$$

Furthermore, the GPOF method shrinks the obtained $L \times N$ pencil into the following square $N \times N$ pencil in order to compute the N vertices:

$$(D_1^{-1} W_1^H T_1 Y_1 - \lambda I)\alpha, \tag{5.7}$$

where W_1 is the truncation of W given by the first N columns of W and D_1 is equal to the $N \times N$ upper left part of D. The estimated vertices z_n, $n = 1, \ldots, N$ are equal to the eigenvalues of (5.7). For more details about the GPOF method, we refer to [45, 26].

From equation (5.7), it is clear that the GPOF method has reduced the dimensionality of the shape-from-moments problem in Least Squares (LS)-sense [38], via computing the SVD of only one of the two matrices T_0 and T_1. Furthermore, by multiplying both sides of equation (5.6) with T_0^+ in order to square the generalized eigenvalue problem, the GPOF method also solves the shape-from-moments problem in LS-sense. In order to square the pencil (5.6), the adjoint of T_0, denoted by T_0^H, can also be used. In this case, the final squared $N \times N$ pencil will be a generalized eigenvalue problem, instead of an eigenvalue problem as in (5.7).

In [89] exactly the same algorithm is called the LS-LS algorithm.

5.3.2 HTLS

In this subsection a short description of the HTLS method is given. The HTLS method makes direct use of the matrix H, expressed in (5.3), to search for good estimates of the vertices z_n, $n = 1, \ldots, N$. Note that matrix H is different from the augmented matrix $[T_0 \quad T_1]$; duplicates of columns have been removed.

Using equation (5.2), Hankel matrix H can be written in terms of Vandermonde matrices:

$$\begin{aligned} H &= \begin{bmatrix} 1 & 1 & \ldots & 1 \\ z_1 & z_2 & \ldots & z_N \\ \vdots & & & \\ z_1^{M-L} & z_2^{M-L} & \ldots & z_N^{M-L} \end{bmatrix} \begin{bmatrix} a_1 & 0 & \ldots & 0 \\ 0 & a_2 & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \ldots & a_N \end{bmatrix} \begin{bmatrix} 1 & z_1 & \ldots & z_1^L \\ 1 & z_2 & \ldots & z_2^L \\ \vdots & & & \\ 1 & z_N & \ldots & z_N^L \end{bmatrix} \\ &\equiv SAC^T. \end{aligned} \tag{5.8}$$

From this Vandermonde decomposition, the vertices z_n, $n = 1, \ldots, N$ can immediately be derived. Instead of computing the decomposition of a Hankel matrix to Vandermonde form directly, the vertices are determined in another

way. For the HTLS method, the vertices are determined as follows. From the equation in (5.8), it is clear that S satisfies the following shift-invariance property:

$$\overline{S} = \underline{S}Z, \tag{5.9}$$

where $\overline{S}$ and $\underline{S}$ are submatrices of S, obtained by deleting the top and the bottom row of S, respectively, and where Z is a diagonal matrix with the vertices z_n, $n = 1, \ldots, N$ on its diagonal. Using this shift-invariance property and a basis transformation via the SVD of the matrix H, the vertices can be determined. The HTLS algorithm is summarized in Algorithm 12.

Algorithm 12 HTLS algorithm

1: Input: a given data sequence τ_k, $k = 0, \ldots, M$.
2: Arrange the data points τ_k, $k = 0, \ldots, M$ in a Hankel matrix H as in equation (5.3).
3: Compute the SVD of H and truncate H to rank N: $H = U\Sigma V^H$, $H_{trun} = U_N \Sigma_N V_N^H$, where U_N and V_N consist of the first N columns of U and V, respectively, and Σ_N is the $N \times N$ upper left submatrix of Σ.
4: Solve the following overdetermined set of equations in TLS-sense (see Theorem 2 for the TLS-solution):

$$\overline{U_N} \approx \underline{U_N} E, \tag{5.10}$$

where $\overline{U_N}$ and $\underline{U_N}$ are submatrices of U_N, obtained by deleting the first and the last row of U_N, respectively.
5: Compute the eigenvalues of the TLS-solution E. These eigenvalues are the estimation of the N vertices z_n, $n = 1, \ldots, N$.
6: Fill in the estimated vertices $\hat{z}_n$, $n = 1, \ldots, N$ into the model function of equation (5.2) and solve the obtained system of N equations in LS-sense to determine the estimates $\hat{a}_n$, $n = 1, \ldots, N$ of the coefficients a_n, $n = 1, \ldots, N$.
7: Output: The estimation of the N vertices z_n, $n = 1, \ldots, N$ and the N coefficients a_n, $n = 1, \ldots, N$.

In this chapter, we are only interested in the estimation of the N vertices z_n, $n = 1, \ldots, N$ of a given N-sided planar polygon. Therefore, only a subpart of the HTLS algorithm, Algorithm 12, is needed. If also the estimated data sequence $\hat{\tau}_k$ with $k = 0, \ldots, M$ is of interest, the entire HTLS algorithm can be used to compute not only the estimated vertices $\hat{z}_n$, $n = 1, \ldots, N$, but also the estimated coefficients $\hat{a}_n$, $n = 1, \ldots, N$.

Instead of computing the SVD of the matrix T_0, which is the case in the GPOF method, the HTLS method computes the SVD of the matrix H. Hence, in the HTLS method, the information of both matrices T_0 and T_1 has been taken into account. So, it is clear that the HTLS method has reduced the dimensionality

of the shape-from-moments problem in TLS-sense. Moreover, also problem (5.10) in step 4 of the HTLS algorithm is solved in TLS-sense.

In [89], a hybrid form of the HTLS algorithm is presented. It is called the TLS-LS algorithm because problem (5.10) is solved in LS-sense.

The simulations in section 5.5 will show that the estimated vertices, computed via the HTLS method, will not be more accurate than the ones computed by the GPOF method. So, by using the HTLS method, there is no gain in accuracy, at least for the shapes under investigation. The computationally most intensive part of the algorithm is the computation of the SVD of the $(M - L + 1) \times (L + 1)$ matrix H, which requires $\mathcal{O}((M - L + 1)(L + 1)^2 + (L + 1)^3)$ floating-point operations (where $\mathcal{O}(\cdot)$ denotes the order of magnitude). For the GPOF algorithm, the computational cost is mainly due to the computation of the SVD of the $(M-L+1) \times L$ matrix T_0, which requires $\mathcal{O}((M-L+1)L^2+L^3)$ floating-point operations. So, the computational cost of the GPOF algorithm and the HTLS algorithm are comparable. Moreover, for the examples that we checked in section 5.5, there is no accuracy improvement for the HTLS algorithm either. Nevertheless, by considering other shapes, the HTLS algorithm can perform better. In [94], it is shown that e.g. in the case of vertices with a small difference between their angles or situated near the origin, there will be a gain in accuracy by using the HTLS method. However, for the investigated polygons, there is no improvement for the HTLS algorithm, despite the use of TLS instead of LS. The reason for this is the structure of matrix H, which has not been taken into account. In the next section, we will present an algorithm that will take the structure into account.

5.3.3 STLS

Also useful for the shape-from-moments problem, are techniques for solving the STLS problem. These methods preserve the structure of the Hankel matrix H, formed by the given data sequence τ_k with $k = 0, \ldots, M$. There are also other methods that exploit the Hankel structure, e.g., Cadzow's algorithm [10] and the Iterative Quadratic Maximum Likelihood algorithm [8], but they are suboptimal, as proven in [52, 54].

The STLS problem is defined as follows [52, 55, 54]. Given an overdetermined set of m linear equations $Ax \approx b$ in $n \times 1$ unknowns x. Find

$$\min_{\Delta A, \Delta b, x} \| [\Delta A \ \Delta b] \|_F^2 \quad \text{subject to} \begin{cases} (A + \Delta A)x = b + \Delta b, \\ [A + \Delta A \ b + \Delta b] \text{ and } [A \, b] \text{ have} \\ \text{the same structure.} \end{cases} \tag{5.11}$$

No closed-form solution is known for problem (5.11), but several iterative approaches have been proposed. Possible approaches are the Constrained TLS approach [2], the Structured Total Least Norm (STLN) approach [95] and the

Riemannian SVD approach [54]. In [52], it is shown that all three approaches are equivalent in the sense that they yield the same parameter vector x. For the shape-from-moments problem, we will focus on the STLN approach. This is the most straightforward approach. In order to obtain the same structure in $[A + \Delta A \; b + \Delta b]$ as in $[A \, b]$, the STLN approach arranges the different errors on A in a vector $\alpha \in \mathbb{C}^q$ with $q \leq m + n - 1$ and the different errors on b (which are not already accounted for in α) in a vector $\beta \in \mathbb{C}^d$ with $d \leq m$. The structure-dependent matrices relating the elements of α and β to their location in ΔA and Δb are: $X \in \mathbb{C}^{m \times q}$, $P_1 \in \mathbb{C}^{m \times q}$ and $P_2 \in \mathbb{C}^{m \times d}$. They are defined by the following equalities:

$$\begin{aligned} \Delta A \, x &= X \alpha \\ \Delta b &= P_2 \, \beta + P_1 \alpha. \end{aligned}$$

Matrix X is constructed by following the rule: *If α_k is the (i, j)th element of ΔA, then x_j is the (i, k)th element of X, with $i = 1, \ldots, m$, $j = 1, \ldots, n$ and $k = 1, \ldots, q$.* For example, assume that a 4×3 Toeplitz matrix A is subject to error, then matrix ΔA is also Toeplitz with first column $[\alpha_3 \, \alpha_4 \, \alpha_5 \, \alpha_6]^\top$ and first row $[\alpha_3 \, \alpha_2 \, \alpha_1]$. If we define α as $\alpha = [\alpha_1 \, \alpha_2 \cdots \alpha_6]^\top$, then matrix X is equal to:

$$X = \begin{bmatrix} x_3 & x_2 & x_1 & 0 & 0 & 0 \\ 0 & x_3 & x_2 & x_1 & 0 & 0 \\ 0 & 0 & x_3 & x_2 & x_1 & 0 \\ 0 & 0 & 0 & x_3 & x_2 & x_1 \end{bmatrix}.$$

Using the nomenclature mentioned, the STLN problem is formulated as follows:

$$\min_{\alpha, \beta, x} \alpha^\top D_\alpha^2 \, \alpha + \beta^\top D_\beta^2 \, \beta \quad \text{such that } \hat{r} = 0, \tag{5.12}$$

where $\hat{r}$ is the modified residue $b + \Delta b - (A + \Delta A)x$ and the matrices D_α and D_β are weighting matrices that account for the repetitions of elements in α and β.

There are several algorithms that can be used to solve the constrained minimization problem (5.12) [38, 95]. The solution of problem (5.12) is computed using algorithm STLN, which is described in Algorithm 13. In each iteration, a Quadratic Programming (QP) problem [29] is solved, namely the quadratic objective function remains the same, but the constraints are linearized around the current iterate.

Approaches for solving the STLS problem (5.11) can also be used to solve the reconstruction problem of an N-sided polygon from a set of its complex moments. This will become clear as follows. From (5.2), we see that the given moments τ_k, $k = 0, \ldots, M$ satisfy an Nth order difference equation. So, we can write the following:

$$p_N \tau_k + p_{N-1} \tau_{k+1} + \cdots + p_1 \tau_{k+N-1} + p_0 \tau_{N+k} = 0, \tag{5.13}$$

Algorithm 13 STLN algorithm

1: Input: An STLN problem (5.12) with specified matrices $[A\,b]$, D_α, D_β and a tolerance ε.
2: Set $\alpha = 0$ and $\beta = 0$.
3: Compute the initial solution for x by solving the LS problem:

$$\min_x \parallel b - Ax \parallel_2 .$$

x is initialized as $(A^\top A)^{-1}A^\top b$.
4: Define $v_{tot} = [\alpha^\top\ \beta^\top\ x^\top]^\top$.
5: Set $\hat{r} = b - Ax$.
6: **repeat**
7: Compute the gradient g and the Hessian H_{ss} of the cost function in (5.12) as follows:

$$g = \begin{bmatrix} D_\alpha^2\alpha \\ D_\beta^2\beta \\ 0 \end{bmatrix}, \quad H_{ss} = \begin{bmatrix} D_\alpha^2 & 0 & 0 \\ 0 & D_\beta^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

8: Compute the Jacobian J of $\hat{r}$.
9: Solve the following QP problem:

$$\min_{\Delta v} f(v_{tot}) + g^\top\,\Delta v + \frac{1}{2}\Delta v^\top H_{ss}\Delta v \quad \text{such that } \hat{r} + J\Delta v = 0,$$

with $f(v_{tot}) = \frac{1}{2}(\alpha^\top D_\alpha^2\alpha + \beta^\top D_\beta^2\beta)$.
10: Set $v_{tot} = v_{tot} + \Delta v$.
11: Extract α, β and x from v_{tot}.
12: Construct ΔA from α, Δb from β and compute $\hat{r} = (b+\Delta b) - (A+\Delta A)x$.
13: **until** ($\parallel \Delta x \parallel, \parallel \Delta\alpha \parallel, \parallel \Delta\beta \parallel \leq \varepsilon$).
14: Output: The correction matrix $[\Delta A\ \Delta b]$ and the STLN-solution x such that $[A + \Delta A\ b + \Delta b]$ has the same structure as the given matrix $[A\,b]$.

for $k = 0, \ldots, M - N$. Without loss of generality, we can assume that $p_0 = 1$, by dividing throughout by p_0. Thus, we get a system of $M - N + 1$ equations of the form:

$$\begin{bmatrix} p_n & \ldots & p_1 & 1 & & 0 \\ & \ddots & & \ddots & \ddots & \\ 0 & & p_N & \ldots & p_1 & 1 \end{bmatrix} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_M \end{bmatrix} = 0.$$

Simple reordering of this set of equations leads to a regular linear set of equa-

tions:

$$\begin{bmatrix} \tau_0 & \tau_1 & \cdots & \tau_{N-1} \\ \tau_1 & \tau_2 & \cdots & \tau_N \\ \tau_2 & \tau_3 & \cdots & \tau_{N+1} \\ \vdots & & \ddots & \\ \tau_{M-N} & \tau_{M-N+1} & \cdots & \tau_{M-1} \end{bmatrix} \begin{bmatrix} p_N \\ p_{N-1} \\ \vdots \\ p_1 \end{bmatrix} = - \begin{bmatrix} \tau_N \\ \tau_{N+1} \\ \tau_{N+2} \\ \vdots \\ \tau_M \end{bmatrix}. \tag{5.14}$$

Using the noisy moments, the above equation is no longer exact. The difference equation coefficients p_n, $n = 1, \ldots, N$ can now be estimated using STLS to solve the system of equations (5.14). Armed with these difference equation coefficients, the vertices z_n, $n = 1, \ldots, N$ can be found by computing the roots of the polynomial

$$P(z) = \prod_{N=1}^{N} (z - z_n) = z^N + \sum_{n=1}^{N} p_n z^{N-n}.$$

The above polynomial equation can be obtained by applying a Z-Transformation to the difference equation (5.13), but we will not go into further detail here. So, it should be clear now that STLS can be used to solve the reconstruction problem of an N-sided polygon from a set of its complex moments by computing the STLS solution of $Ax \approx b$ with A equal to the first N columns of matrix H, expressed in (5.3), b equal to the last column of H and $L = N$. The STLS solution $\hat{x}$ is a vector that contains the coefficients of a polynomial $P(z) = z^N + \sum_{n=1}^{N} \hat{x}_n \, z^{N-n}$ of which the roots are the N vertices.

The STLS method has a higher computational cost than the two non-iterative methods GPOF and HTLS. Moreover, since a nonlinear optimization problem needs to be solved, the use of good starting values is of great importance for an STLS approach to converge within a reasonable amount of time. Nevertheless, from section 5.5, it will become clear that a higher accuracy can be obtained for the estimated vertices by using the STLS method. The higher accuracy can be explained by the fact that STLS preserves the matrix structure thereby maintaining consistency of its solution and the maximum likelihood property in case of Gaussian measurement errors [52].

5.3.4 The tensor approach

In this subsection, a short description of a tensor-based algorithm is presented. The problem of harmonic analysis was, for the first time, cast in a tensor framework in [47, 83]. Before presenting the tensor-based algorithm which we will use to solve the shape-from-moments problem, we will describe some tensor-algebraic material in order to clarify the derivations towards the algorithm first.

Background material

The tensor theory is a natural generalization of the matrix theory. In analogy with the matrix case, a third-order tensor $\mathcal{A}$ is called a rank-1 tensor when it can be written as the outer product of three vectors $U^{(1)}, U^{(2)}, U^{(3)}$:

$$a_{i_1 i_2 i_3} = u^{(1)}_{i_1} u^{(2)}_{i_2} u^{(3)}_{i_3},$$

for all index values, which we will denote as

$$\mathcal{A} = U^{(1)} \circ U^{(2)} \circ U^{(3)}.$$

A tensor that can be written as the sum of R, but not less than R, rank-1 terms is called a rank-R tensor. An n-mode vector of a tensor $\mathcal{A}$ is a vector obtained from $\mathcal{A}$ by varying only the nth index. As such, the n-mode vectors are the generalization of column and row vectors in the matrix case. The subspace spanned by all the n-mode vectors, for a given value of n, is called the n-mode vector space. The dimension of this subspace is called the n-mode rank. A difference between matrices and tensors is that, for tensors, the different n-mode ranks are not necessarily the same. A third-order tensor whose n-mode ranks are equal to R_n $(1 \leq n \leq 3)$ is denoted as a rank-(R_1, R_2, R_3) tensor. A second difference between matrices and tensors is that the rank of a tensor is not necessarily equal to its n-mode ranks. From the definition we always have that $R_n \leq R$.

A third-order natural extension of the matrix SVD is pointed out in the next theorem.

Theorem 5 (Third-order SVD) *Every complex $(I_1 \times I_2 \times I_3)$ tensor $\mathcal{A}$ can be written as the product*

$$\mathcal{A} = \mathcal{S} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}, \tag{5.15}$$

so,

$$a_{i_1 i_2 i_3} = \sum_{j_1 j_2 j_3} s_{j_1 j_2 j_3} v^{(1)}_{j_1 i_1} v^{(2)}_{j_2 i_2} v^{(3)}_{j_3 i_3}$$

for all values of the indices, in which

- $V^{(n)} = \left[V^{(n)}_1 V^{(n)}_2 \dots V^{(n)}_{I_n} \right]$ *is a unitary $(I_n \times I_n)$ matrix,*
- *$\mathcal{S}$ is a complex $(I_1 \times I_2 \times I_3)$ tensor of which the subtensors $\mathcal{S}_{i_n=\alpha}$, obtained by fixing the nth index to α, have the properties of:*
  - all-orthogonality*: two subtensors $\mathcal{S}_{i_n=\alpha}$ and $\mathcal{S}_{i_n=\beta}$ are orthogonal for all possible values of n, α and β subject to $\alpha \neq \beta$:*

$$\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0 \text{ when } \alpha \neq \beta.$$

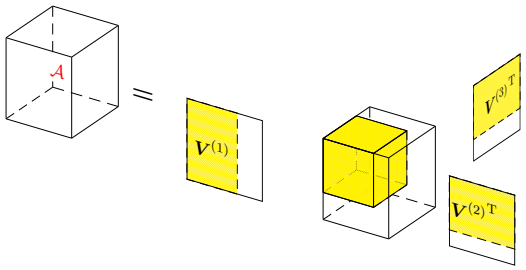


Figure 5.1: Visualization of the Higher Order Singular Value Decomposition for a third-order tensor. If $\mathcal{A}$ is n-mode rank deficient, only the shaded part of the core-tensor may contain entries different from zero.

– ordering*:*

$$\| \mathcal{S}_{i_n=1} \| \geq \| \mathcal{S}_{i_n=2} \| \geq \ldots \geq \| \mathcal{S}_{i_n=I_n} \| \geq 0 \tag{5.16}$$

for all possible values of n*.*

The Frobenius norm $\| \mathcal{S}_{i_n=i} \|$*, symbolized by* $\sigma_i^{(n)}$*, is the* i*th* n*-mode singular value of* $\mathcal{A}$ *and the vector* $V_i^{(n)}$ *is the* n*-mode singular vector. The decomposition for third order tensors is visualized in Figure 5.1.*

It can be proved that it is generally impossible to diagonalize a higher-order tensor by means of unitary transformations: the shaded part of the core tensor in Figure 5.1 is generally a full (R_1, R_2, R_3) tensor. Instead, it satisfies the condition of all-orthogonality. All-orthogonality is a sufficiently strong condition for many key properties of the matrix SVD to carry over the Higher Order SVD (HOSVD) (namely, properties expressing a link between the distribution of the column/row (n-mode) vectors on one hand, and the (HO)SVD components on the other hand). In particular, the n-mode vectors corresponding to non-zero n-mode singular values form an orthonormal basis for the n-mode vector space of $\mathcal{A}$. The n-mode vectors corresponding to the vanishing n-mode singular values form an orthonormal basis for the orthogonal complement of the n-mode vector space. The number of (significant) n-mode singular values equals the (numerical) n-mode rank of $\mathcal{A}$. Matrix $V^{(n)}$ $(1 \leq n \leq 3)$ can easily be computed as the matrix of left singular vectors of a matrix in which all n-mode vectors of $\mathcal{A}$ are stacked one after the other. The n-mode singular values are the singular values of this matrix. The core tensor $\mathcal{S}$ follows from bringing $V^{(1)}$, $V^{(2)}$, $V^{(3)}$ to the left side of equation (5.15). The inner product of two tensors $\mathcal{A}$ and $\mathcal{B}$ of the same dimensionality is given by:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1 i_2 i_3} a_{i_1 i_2 i_3} b^*_{i_1 i_2 i_3},$$

in which the summation is over all the indices. The Frobenius norm of a tensor then is $\| \mathcal{A} \| = (\langle \mathcal{A}, \mathcal{A} \rangle)^{\frac{1}{2}}$. The best rank R approximation problem in matrix algebra, can be generalized in the following way: *Given a complex third-order tensor* $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$*, find a rank* (R_1, R_2, R_3) *tensor* $\widehat{\mathcal{A}}$ *that minimizes the least-squares cost function*

$$f(\widehat{\mathcal{A}}) = \| \mathcal{A} - \widehat{\mathcal{A}} \|^2 .$$

Due to the n-rank constraints, $\mathcal{A}$ can be decomposed as:

$$\widehat{\mathcal{A}} = \mathcal{B} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}, \tag{5.17}$$

in which $V^{(1)} \in \mathbb{C}^{I_1 \times R_1}$, $V^{(2)} \in \mathbb{C}^{I_2 \times R_2}$, $V^{(3)} \in \mathbb{C}^{I_3 \times R_3}$ each have orthonormal columns and $\mathcal{B} \in \mathbb{C}^{R_1 \times R_2 \times R_3}$ is an all-orthogonal tensor. As is well-known in the matrix case, the best rank R approximation can simply be obtained by truncation of the matrix SVD. An important difference between matrices and higher-order tensors, however, is that the best rank (R_1, R_2, R_3) approximation cannot in general be obtained by mere truncation of the HOSVD. This is due to the fact that the core tensor is not diagonal. Nevertheless, due to the ordering constraint (5.16), the truncated HOSVD is usually a pretty good approximation, albeit not the optimal one. The best rank (R_1, R_2, R_3) approximation can be calculated by means of tensor generalizations of algorithms for the calculation of a dominant subspace. In [21], a Higher-Order Orthogonal Iteration (HOOI) method is presented. The HOOI algorithm is analog to the Alternating Least Squares (ALS) type (see Algorithm 9) and its convergence rate is linear.

Readers who are interested in the tensor-algebraic material are referred to [20] for more details.

Tensor-based algorithm

First, the data sequence τ_k, $k = 0, \ldots, M$ is stacked in a third-order tensor as follows. A first $(I_1 \times I_2)$ Hankel matrix is built using the P $(P < M + 1)$ first samples of the signal $\tau_0, \ldots, \tau_{P-1}$. A second one is built with the segment $\tau_1, \ldots, \tau_P$ obtained by shifting the previous segment over one sample, and so on until the end of the signal. Then, these Hankel matrices are stored one behind each other to form a Hankel tensor as visualized in Figure 5.2. An element $h_{i_1 i_2 i_3}$ of this $(I_1 \times I_2 \times I_3)$ tensor $\mathcal{H}$ is given by

$$h_{i_1 i_2 i_3} = \tau_{(i_1 - 1) + (i_2 - 1) + (i_3 - 1)} = \tau_{i_1 + i_2 + i_3 - 3},$$

with $1 < i_1 < I_1$, $1 < i_2 < I_2$, $1 < i_3 < I_3$, $I_1 > N$, $I_2 > N$ and $I_1 + I_2 + I_3 = M + 3$. As long as the previous constraints are fulfilled, the user may choose the dimensions of the tensor. The one-dimensional complex signal τ_k is modelled

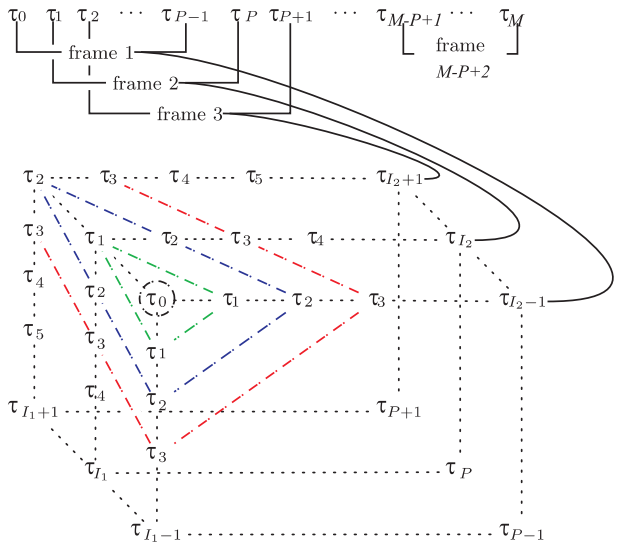


Figure 5.2: Segmentation of the signal and construction of a tensor with Hankel matrices. The dotted lines delimit the tensor while dashed lines show its Hankel structure in the three directions. A dashed line generates a diagonal tensor slice on which all entries are equal.

by equation (5.2). Using this equation, like in subsection 5.3.2, we have that

$$h_{i_1 i_2 i_3} = \sum_{n=1}^{N} a_n (z_n^{i_1 - 1} \cdot z_n^{i_2 - 1} \cdot z_n^{i_3 - 1}).$$

In other words, the tensor $\mathcal{H}$ is a weighted sum of third-order rank-1 tensors, consisting of the outer product of three Vandermonde vectors:

$$\mathcal{H} = \sum_{n=1}^{N} a_n \cdot \begin{bmatrix} 1 \\ z_n^1 \\ z_n^2 \\ \vdots \\ z_n^{I_1'} \end{bmatrix} \circ \begin{bmatrix} 1 \\ z_n^1 \\ z_n^2 \\ \vdots \\ z_n^{I_2'} \end{bmatrix} \circ \begin{bmatrix} 1 \\ z_n^1 \\ z_n^2 \\ \vdots \\ z_n^{I_3'} \end{bmatrix},$$

where $I_j' = I_j - 1$. This can also be written as

$$\mathcal{H} = \mathcal{A} \times_1 S^{(1)} \times_2 S^{(2)} \times_3 S^{(3)}, \tag{5.18}$$

in which $\mathcal{A}$ is the $(N \times N \times N)$ pseudo-diagonal core-tensor containing the N complex amplitudes a_n and in which $S^{(1)} \in \mathbb{C}^{N \times I_1}$, $S^{(2)} \in \mathbb{C}^{N \times I_2}$ and $S^{(3)} \in \mathbb{C}^{N \times I_3}$ are Vandermonde matrices. This decomposition is called a Higher-Order Vandermonde Decomposition (HOVDMD) in analogy with the VDMD (5.8) in

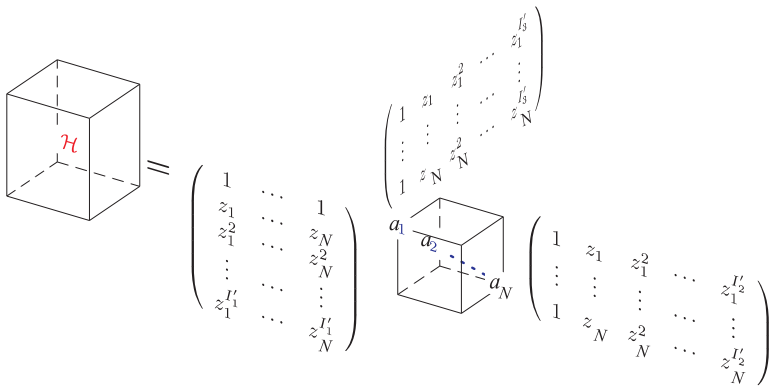


Figure 5.3: Visualization of the HOVDMD for a third-order Hankel tensor.

the matrix case and is visualized in Figure 5.3. By way of comparison, note that in the matrix case we can write:

$$\begin{aligned} H &= \sum_{n=1}^{N} a_n \cdot \begin{bmatrix} 1 \\ z_n^1 \\ z_n^2 \\ \vdots \\ z_n^{M-L} \end{bmatrix} \circ \begin{bmatrix} 1 \\ z_n^1 \\ z_n^2 \\ \vdots \\ z_n^{L} \end{bmatrix} \\ &= S \times_1 A \times_2 C, \end{aligned}$$

where S, A and C are defined in (5.8).

Because of the structure induced by (5.18), in the absence of noise, the HOSVD of tensor $\mathcal{H}$ takes the following form:

$$\mathcal{H} = \widehat{\mathcal{D}} \times_1 \widehat{U}^{(1)} \times_2 \widehat{U}^{(2)} \times_3 \widehat{U}^{(3)},$$

in which $\widehat{\mathcal{D}}$ is an all-orthogonal, ordered, complex $(N \times N \times N)$ tensor and $\widehat{U}^{(n)} = [U_1^{(n)} \dots U_N^{(n)}]$ is a complex $(I_n \times N)$ matrix whose orthonormal columns span the column space of $S^{(n)}$ for all $1 \leq n \leq 3$. In the presence of noise, $\mathcal{H}$ will be a full n-rank tensor. Just like in the matrix case, it makes sense to proceed with the HOSVD components of the best rank-(R_1, R_2, R_3) approximation of $\mathcal{H}$ (cf. equation (5.17)). We claim that $\widehat{U}^{(n)}$ equals $S^{(n)}$, for $n = 1, 2, 3$, up to a multiplication by a square non-singular matrix Q (just like in the matrix case):

$$\widehat{U}^{(n)} = S^{(n)} Q, \tag{5.19}$$

where $Q \in \mathbb{C}^{N \times N}$. On the other hand, the shift-invariant property still holds for the tensor case:

$$\underline{S}^{(n)} Z = \overline{S}^{(n)}, \tag{5.20}$$

where Z is a diagonal matrix with the vertices $z_n, n = 1, \ldots, N$ on its diagonal. Combining (5.19) and (5.20) gives the following matrix equation:

$$\overline{\widehat{U}}^{(n)} = \underline{\widehat{U}}^{(n)} Q^{-1} Z Q = \underline{\widehat{U}}^{(n)} \widetilde{Z}. \tag{5.21}$$

This equation is similar to (5.10) and can be processed in the same way.

To conclude this subsection, the tensor-based algorithm for the estimation of the N vertices $z_n, n = 1, \ldots, N$ of an oversampled signal $\tau_k, k = 0, \ldots, M$ is summarized in Algorithm 14.

Algorithm 14 tensor-based algorithm

1: Input: a given data sequence $\tau_k, k = 0, \ldots, M$.
2: Map the given signal $\tau_k, k = 0, \ldots, M$ on a third-order tensor $\mathcal{H}$ as depicted in Figure 5.2.
3: Find the best rank-(N, N, N') approximation of $\mathcal{H}$, with $N' \leq N$ (for instance, by applying the HOOI algorithm [21]) and obtain in this way the matrices $\widehat{U}^{(1)}, \widehat{U}^{(2)}$ and $\widehat{U}^{(3)}$.
4: Compute the TLS solution $\widehat{\widetilde{Z}}$, see Theorem 2, of the overdetermined set of linear equations (5.21):

$$\widehat{\widetilde{Z}} = -W_{12} W_{22}^{-1},$$

where

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

is obtained from the SVD of the augmented matrix $[\underline{\widehat{U}}^{(n)} \; \overline{\widehat{U}}^{(n)}] = Y D W^H$.
5: Compute the eigenvalues of $\widehat{\widetilde{Z}}$. These eigenvalues are the estimated vertices.
6: Output: The estimation of the N vertices $z_n, n = 1, \ldots, N$.

As already mentioned before, a difference between matrices and tensors is that, for tensors, the different n-mode ranks are not necessarily the same. Therefore, in step 3 of Algorithm 14, the rank 3-mode can be lower than the rank 1-mode and the rank 2-mode. Taking a value for N' that is strictly lower than N is particularly useful when there is a large gap between the 3-mode singular values. In these situations, it is numerically preferable to extract the dominant N'-dimensional mode-3 vector space instead of the N-dimensional one. The tensor framework offers more versatility in terms of the choice of dimensions of the approximation than the matrix framework does. In section 5.5, an example is given for which the tensor technique gives better results than matrix-based algorithms.

5.4 Related work

The literature offers many other algorithms to solve the shape-from-moments problem than the ones described in section 5.3. In this section, we will point out previous work that has been done on the shape-from-moments problem. In short, we will describe other methods from the literature and explain their relationship with the methods presented in the previous section. For more details, we refer the reader to the papers cited below.

The GPOF method was presented originally by Hua and Sarkar [45] and used later on in [58, 57, 76]. First of all, Milanfar et al. [67] suggested a number of algorithms based on Prony's method to apply to the problem of reconstructing a planar polygon from a set of its complex moments. Later on, Golub et al. [39, 26] proposed better numerical procedures such as the GPOF method, based upon matrix pencils, for the shape-from-moments reconstruction problem.

An alternative technique for our HTLS method is presented in [60]. In their paper, F. Luk and D. Vandevoorde have presented an algorithm to approximate the vertices $z_n, n = 1, \ldots, N$ by computing the poles of the Hankel-Vandermonde decomposition in equation (5.8) and selecting the dominant ones. In our HTLS method the Vandermonde decomposition is not computed directly.

Prony's method [67, 26], solved via TLS rather than plainly LS, solves the same problem as the HTLS method, but in a different way. The TLS-Prony method, which is Prony's method solved via TLS, was for the first time described by Pisarenko. Later on, this method was called LPTLS in Van Huffel's paper [94]. The superior performance of LPTLS compared to LPSVD (= Prony's method) has been proven in [88, 93]. HTLS and LPTLS solve the same problem, but in a different way. HTLS is indeed quite similar to LPTLS up to truncation of the data matrix H to rank N in step 3 of Algorithm 12. The data matrix used in the SVD is the same as well as the criteria used for truncation of the singular values. However, after truncation the methods take separate ways. LPTLS carries out polynomial rooting in order to find the estimated vertices, while HTLS avoids rooting by formulating the problem in a state space setting thereby increasing numerical robustness. A detailed comparison of the underlying theories of LPSVD and HSVD is given in [106]. It was found that the HSVD estimate is more precise than the LPSVD estimate, but that the difference is not huge.

The STLS method exploits the Hankel structure of the data matrix. Besides the STLS method, there exist also other methods that exploit the Hankel structure, e.g., Cadzow's algorithm [10] and the Iterative Quadratic Maximum Likelihood algorithm [8] but they are suboptimal, as shown in [23, 54]. Cadzow's algorithm obtains a rank deficient Hankel matrix approximation to a given Hankel matrix as follows. First, the best rank deficient least squares approximation to a given matrix is obtained from the SVD of this matrix. If the given matrix is

structured, however, the rank deficient approximation obtained by truncating the SVD of the given matrix, will typically have no structure at all. One could then try to find the best least squares fit to the obtained rank deficient matrix, that has the required structure. For Hankel matrices, it turns out that this fit is found by replacing the elements on the anti-diagonals by their average. This new matrix will be no longer rank deficient. One could again use a truncation of the SVD in order to find a rank deficient approximation etc. In [10], Cadzow proves that the procedure mentioned above indeed converges to a rank deficient Hankel matrix. Nevertheless, the STLS method will give better results because it is an optimal method instead of a suboptimal method.

The STLS method is closely related to the recent work of Boutry et al. [7]. In their paper the following minimal perturbation approach is presented for solving non-square generalized eigenvalue problems:

$$\begin{aligned} &\text{Given} && \text{matrices } A, B \text{ of size } m \times n, \\ &\text{Find} && \min_{\substack{A_0, B_0, \\ \lambda_k, v_k}} \| A_0 - A \|_F^2 + \| B_0 - B \|_F^2 \\ & && \text{s.t. } (A_0 - \lambda_k B_0) v_k = 0 \text{ and } \| v_k \|_2^2 = 1, \end{aligned} \tag{5.22}$$

for $k = 1, \ldots, n$. The given matrices A, B are assumed to be noisy and originating from the pair A_0, B_0. Furthermore, it is assumed that there exist n distinct eigenpairs of the form $(A_0 - \lambda_k B_0)v_k = 0$, with $k = 1, \ldots, n$ and the original pair A_0, B_0. It is shown in Boutry's work that for the case of $n = 1$ problem (5.22) is equivalent to the classical unstructured TLS problem. By rewriting equation (5.22) this relation becomes obvious:

$$\min_{a_0, b_0} \| [a_0, b_0] - [a, b] \|_F^2 \text{ s.t. } \operatorname{rank}([a_0, b_0]) = 1.$$

Nevertheless, in [7], it is shown that the relationship between the approach discussed by Boutry and the classical TLS no longer holds for larger n.

The problem of reconstructing a planar polygon [39, 67] from its measured moments is extended by Goldenshluger et al. [34]. In paper [34], the problem of reconstructing a planar convex set from noisy observations of its moments is considered. A planar polygon is just one example of a planar convex set, but also, e.g., a circle or an ellipse are possible examples. An estimation method based on point by point recovering of the support function of the set is developed in [34]. Although various algorithms have been developed in the literature to solve the reconstruction problem of a planar polygon from its noisy moments, their statistical properties have not been studied thoroughly. Moreover, usually geometric and/or complex moments are used in the shape from moments problem for reasons of simplicity and convenience [39, 67]. In Goldenshluger's paper, the situation is explored where the noisy moments with respect to the Legendre polynomials, called the Legendre moments, can be observed. In this reference, an optimal and computationally efficient algorithm for estimating

	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
GPOF	2.6e-6	1.2e-1	2.4e-6	1.1e-1	2.4e-6	1.3e-1	2.5e-6	1.1e-1
HTLS	2.7e-6	1.3e-1	2.4e-6	1.2e-1	2.5e-6	1.5e-1	2.6e-6	1.2e-1
STLS	2.4e-6	5.2e-2	2.3e-6	5.3e-2	2.1e-6	5.9e-2	2.1e-6	4.8e-2
Tensor	2.4e-6	1.5e-1	2.6e-6	1.6e-1	2.6e-6	1.4e-1	2.7e-6	2.1e-1

	z_9	z_{10}	overall error
GPOF	2.6e-6	1.8e-1	0.0932
HTLS	2.8e-6	1.9e-1	0.1029
STLS	2.3e-6	1.4e-1	0.0558
Tensor	2.4e-6	1.7e-1	0.1098

Table 5.1: RMSE for each method for the star shaped polygon described in Example 5.1.

convex compact planar regions from noisy measured Legendre moments is presented.

Shape reconstruction from the knowledge of moments can be realized in case the object is a polygon [39], or when it defines a quadrature domain in the complex plane [42]. Recently, in paper [14] it is shown how shape reconstruction from the knowledge of moments can also be realized in the case of general compact objects, not only in two but also in higher dimensions. In this reference, the presented technique is able to deal with very general shapes: nonconvex shapes, shapes with nonconnected boundary and higher-dimensional objects.

5.5 Performance comparison between the methods

In this section, we compare the statistical accuracy of the GPOF method, the HTLS method, the STLS method and the tensor-based method on simulated data.

For this comparison, first, we use two experiments that are presented in [26]. Here, we will not explain the setup of the experiments into detail. The reader is referred for more details to the paper of Elad et al. [26]. In short, the experiments are constructed as follows: first a polygon is created and its complex moments are computed. Secondly, complex Gaussian white noise is added to the moments. Finally, the different algorithms - GPOF, HTLS, STLS with the HTLS solution as a starting value and the tensor-based algorithm - are applied to this set of noisy complex moments in order to obtain the estimated vertices. The presented results are obtained by implementing the different algorithms in Matlab (version 6.1) on a PC i686 with 800 MHz and 256 MB memory.

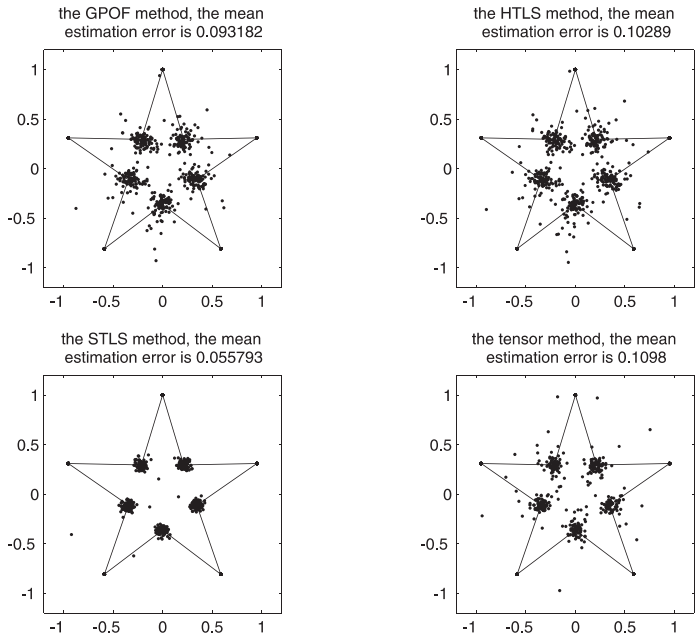


Figure 5.4: Estimated vertices for the star shaped polygon described in Example 5.1.

	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
GPOF	6.2e-4	7.1e-5	5.7e-2	1.6e-1	1.7e-1	5.9e-2	7.1e-5	6.1e-4
HTLS	6.8e-4	7.5e-5	6.5e-2	1.7e-1	1.9e-1	6.9e-2	7.2e-5	6.6e-4
STLS	4.3e-4	5.9e-5	2.5e-2	4.9e-2	5.2e-2	2.5e-2	6.2e-5	4.4e-4
Tensor	5.6e-4	6.5e-5	4.9e-2	1.2e-1	2.6e-1	5.6e-2	7.4e-5	6.1e-4

	overall error
GPOF	0.0869
HTLS	0.0966
STLS	0.0282
Tensor	0.1060

Table 5.2: RMSE for each method for the "C" shaped polygon described in Example 5.2.

Example 5.1 In this experiment we use a star-shaped polygon with 10 vertices. We assume that 101 noisy complex moments $\tau_0, \ldots, \tau_{100}$ are given with a noise variance of $2 \cdot 10^{-4}$. The results after applying the different methods to this noisy data set are presented in Table 5.1 and in Figure 5.4. Table 5.1 contains, for each method, the average Root Mean Squared Error (RMSE) for each vertex z_n with $n = 1, \ldots, 10$ over 100 Monte Carlo simulations, as well as the overall error. The notation e-a is used instead of 10^{-a}, where a is an integer. In Figure 5.4 the location of the 100 estimated sets of vertices for each method is presented. The ten vertices are located counter-clockwise, starting from vertex z_1 with (0,1) as its coordinates, z_2 with (-0.21,0.29) as its coordinates, z_3 with (-0.95,0.31) as its coordinates, etc. △

Example 5.2 In this experiment we use a "C" shaped polygon with 8 vertices. We assume that 81 noisy complex moments are given with a noise variance of $6 \cdot 10^{-3}$. The results after applying the different methods to this noisy data set are presented in Table 5.2 and in Figure 5.5. Table 5.2 contains, for each method, the average RMSE for each vertex over 100 Monte Carlo simulations as well as the overall error. In Figure 5.5 the location of the 100 estimated sets of vertices for each method is presented. The eight vertices are located clockwise, starting from vertex z_1 with (-0.28,0.90) as its coordinates, z_2 with (0.43,0.90) as its coordinates, z_3 with (0.43,0.54) as its coordinates, etc. △

From the tables and the figures it is clear that the estimated vertices computed via the STLS method are more accurate than the ones computed via the other methods. The tensor method and the matrix-based methods GPOF and HTLS perform quite similarly. Note from Table 5.1 that some vertices are much more accurate than others, e.g., $z_1, z_3, \ldots, z_9$. For an explanation of this behaviour, we refer to [39].

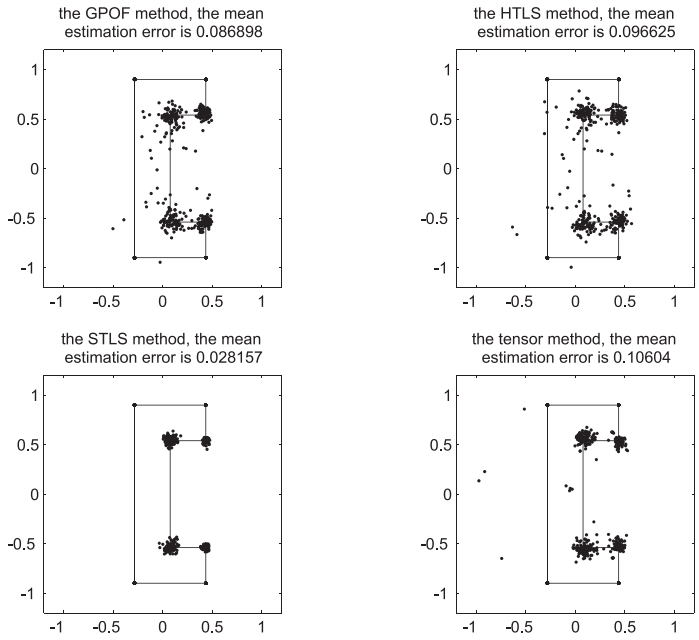


Figure 5.5: Estimated vertices for the "C" shaped polygon described in Example 5.2.

When the noise variance is reduced, accuracy is improved for all the algorithms and the difference in accuracy between the three algorithms becomes weaker. On the other hand, all the algorithms fail when the noise level exceeds a common threshold. The phenomenon of failure is related to the problem formulation (5.2) itself. When the noise increases, the number of moments τ_k lying above the noise level decreases. So, there are not enough 'reliable' moments left in order to get meaningful estimates of the vertices. In these cases, unreliable results are not due to a wrong choice of particular algorithms, but to the ill-conditioning of the problem itself.

Secondly, we use an experiment which is similar to Experiment 1. Instead of creating one star-shaped polygon with 10 fixed vertices, in the following experiment, we vary the five inner vertices $z_2, z_4, \ldots, z_{10}$ during the simulations while keeping the outer vertices fixed. Hence, we can better understand the dependency between the behaviour of the applied methods and the distance between the inner and outer vertices of a star-shaped polygon. Like before, first a certain star-shaped polygon is created and its complex moments are computed. Secondly, complex Gaussian white noise is added to the moments. Finally, the different algorithms - GPOF, HTLS, STLS and the tensor-based algorithm - are applied to this set of noisy complex moments in order to obtain the estimated vertices. The presented results are obtained by implementing the different algorithms in Matlab (version 6.1) on a PC i686 with 800 MHz and 256 MB memory.

Example 5.3 In this experiment we use star-shaped polygons with 10 vertices. For each polygon, we use the same outer vertices $z_1, z_3, \ldots, z_9$ as in Example 5.1. For each polygon, however, the inner vertices $z_2, z_4, \ldots, z_{10}$ are different. We let them vary in distance r_2 to the origin from 1 to 0.82 in small steps of 0.001. We assume that, for every polygon, 101 noisy complex moments $\tau_0, \ldots, \tau_{100}$ are given with a noise variance of 10^{-3}. In order to compute the estimated vertices via the tensor approach, we stacked the sequence of noisy moments in a third-order $(49 \times 49 \times 4)$ tensor $\mathcal{H}$ as depicted in Figure 5.2 and applied the tensor-based algorithm, Algorithm 14, to tensor $\mathcal{H}$. In step 3 of the algorithm, we performed a best rank-$(10, 10, 4)$ approximation of $\mathcal{H}$ and used the 1-mode dominant subspace $\widehat{U}^{(1)}$ to estimate the vertices. The results after applying the tensor-based method and the matrix-based methods, GPOF, HTLS and STLS, to the noisy moments are presented in Figure 5.6. For each method, the figure contains the RMSE of the inner vertex z_2 over 100 Monte Carlo simulations, with respect to the ratio r_1/r_2. For each polygon, the ratio r_1/r_2 is the ratio between the fixed modulus $r_1 = 1$ of the outer vertices, which is the same for all the polygons, and the modulus r_2 of the inner vertices, which is different for each polygon and varies from 1 to 0.82 in small steps of 0.001. By varying the distance of the inner vertices to the origin from one polygon to another, the polygons vary from a star-shaped polygon which almost is a circle to a star-shaped polygon which almost is a pentagon. $\triangle$

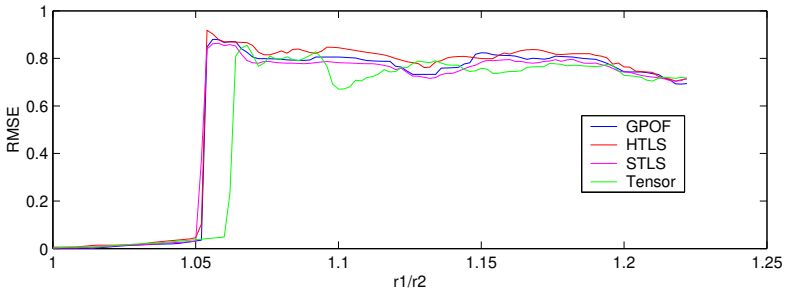


Figure 5.6: RMSE of the second vertex z_2 for each method, for the set of star-shaped polygons described in Example 5.3. The fixed distance between the outer vertices and the origin is denoted by $r_1 = 1$ and the varying distance between the inner vertices and the origins is denoted by r_2.

From Figure 5.6 it is clear that the tensor approach and the matrix approaches perform nearly the same although the tensor approach is more accurate in a region around $r_1/r_2 = 1.06$. For the other inner vertices $z_4, z_6, \ldots, z_{10}$, similar results were obtained as in Figure 5.6. For the outer vertices $z_1, z_3, \ldots, z_9$, we obtained for each method a RMSE of the order of the noise level of 10^{-3}. Hence, for these outer vertices the difference in accuracy between the methods is negligible. The results show that concentrating on the dominant part of $\mathcal{H}$ increases the robustness.

5.6 Conclusion

In this chapter, we discussed the problem to recover the vertices of a planar polygon from its measured complex moments. Because the given moments can be noisy, the recovered vertices are estimations of the true ones. The literature offers many algorithms for solving such an estimation problem. We restricted our discussion to the HTLS, the STLS, the GPOF and a tensor-based method. We showed the close link between the HTLS and the GPOF methods. Through simulated data we compared the accuracy of the four methods mentioned. We found that the HTLS method and the GPOF method perform similarly. Moreover, we showed that the vertices estimated via the STLS algorithm are more accurate than the ones estimated via the HTLS and the GPOF algorithms. By arranging the data sequence in a higher-order tensor and estimating the vertices via a multilinear generalization of the SVD, the parameter accuracy improved over the matrix approaches for a small set of star-shaped polygons.

Chapter 6

Conclusions and further research

In the first section of this chapter the conclusions are summarized. A second section describes open problems and outlines suggestions for further research.

6.1 Conclusions

In this thesis we have treated the algorithms and applications for Weighted Low Rank Approximation problems. We now present the main conclusions and original contributions.

- The tight equivalences between the Weighted Low Rank Approximation (WLRA) problem and the Total Least Squares (TLS) problem were explored. Despite the seemingly different problem formulations of WLRA and TLS, it was shown that both methods can be reduced to the same mathematical kernel problem, i.e. finding the closest (in a certain sense) weighted low rank matrix approximation where the weight is derived from the distribution of the errors in the data. Different solution approaches, as used in WLRA and TLS, were discussed. In particular, we discussed the Null space parameterized WLRA (NullWLRA), the Maximum Likelihood Principal Component Analysis (MLPCA), the Element-wise Weighted-TLS (EW-TLS) and the Generalized TLS (GTLS) methods. It was shown that these four approaches tackle an equivalent weighted low rank approximation problem, but different algorithms are used to determine the best approximation matrix.

- The WLRA problem in the application field of chemometrics was studied. In chemometrics, existing approaches to find the best weighted low rank matrix approximation are the well-known Principal Component Analysis (PCA) method and the MLPCA method. The use of TLS-like algorithms in chemometrics was discussed. An adapted version of the EW-TLS algorithm was developed to solve the WLRA problem for data matrices in chemometrics with more columns than rows and with only row-wise correlated measurement errors. We found that our newly developed algorithm performed better than the existing MLPCA method for this specific type of matrices. Moreover, we found that the existing EW-TLS algorithm outperformed the MLPCA algorithm for matrices of size $m \times n$ with $m > n$ and with only correlations among the rows of the measurement errors.
- The WLRA approach was extended towards linearly structured matrices. The Hankel structure was investigated since this is one of the most frequently occurring structures in signal processing applications. We studied the cases of scalar Hankel matrices and block-row Hankel matrices. For both cases, new algorithms were presented in order to solve these structured WLRA problems. The correctness of the proposed algorithms was verified on benchmark problems. Through simulation experiments the improved statistical accuracy of the proposed algorithms compared to known algorithms from the literature was confirmed.
- The structured WLRA problem was studied in the application field of recovering the vertices of a planar polygon from its measured complex moments. In the literature, the use of the structured TLS approach to solve this shape-from-moments problem has not been discussed. Therefore, in chapter 5, the STLS algorithm has been proposed as an alternative algorithm to solve the reconstruction problem. By means of experiments it was demonstrated how accurate the STLS algorithm is in comparison with algorithms that do not take into account the structure of the data matrix built from the given complex moments.

6.2 Further research

- By applying TLS-like algorithms to the field of chemometrics, it became clear that these TLS-like algorithms are good alternatives to the MLPCA algorithm for the specific cases of row-wise correlated measurement errors and data matrices which are far from squared. The computational efficiency of the MLPCA algorithm can be improved by combining its good convergence with the efficiency of the TLS-like algorithms.
- In chapter 4 we developed new algorithms in order to solve the structured WLRA problem for scalar Hankel matrices and block-row Hankel matri-

ces. By means of simulation experiments the improved statistical accuracy of the proposed algorithms compared to known algorithms from the literature was confirmed. However, the computational cost of these new algorithms can significantly decrease by exploiting the matrix structure in the computation of the QR decomposition, e.g., by using displacement rank theory [49, 53].

- In the shape-from-moments problem, it became clear that the parameter accuracy improved by arranging the data sequence in a higher-order tensor and estimating the model parameters via a multilinear generalization of the SVD. However, the region where the tensor approach outperformed the matrix approaches, was quite small. To further explore the potential of the tensor framework, a thorough analysis is necessary to see in which typical cases of harmonic analysis the tensor approach performs best. From our experiments, it seems that the tensor approach performs best in cases where the signal poles become collinear. It should be checked if this behaviour can be generalized for all cases where the signal poles become linearly dependent.

- Further research is necessary to decrease the computational cost of the rank reduction algorithm used in the tensor-based algorithm. The use of more sophisticated procedures that take the matrix structure into account, can reduce the computation time.

Appendix

A Proof of Theorem 3

Theorem *Let $f(N)$ be defined as $f(N) = \min_{\substack{\widehat{D}\in\mathbb{R}^{m\times n} \\ \widehat{D}N=0}} \| D - \widehat{D} \|_W^2$, with $D \in \mathbb{R}^{m\times n}$ a given data matrix and $W \in \mathbb{R}^{mn\times mn}$ a positive-definite, symmetric weighting matrix. Then a closed-form expression for $f(N)$ is given by*

$$f(N) = vec^\top(D)(N \otimes I_m)[(N \otimes I_m)^\top W(N \otimes I_m)]^{-1}(N \otimes I_m)^\top vec(D),$$

where $\otimes$ denotes the Kronecker product. $f(N)$ depends only on the column space of N: for any invertible matrix S, $f(NS) = f(N)$. The minimizing $\widehat{D}$ is given by

$$vec(\widehat{D}) = (I - W(N \otimes I_m)[(N \otimes I_m)^\top W(N \otimes I_m)]^{-1} (N \otimes I_m)^\top)vec(D).$$

Proof : Applying the technique of Lagrange multipliers to the minimization $f(N)$ yields the Lagrangian

$$\psi(L, \widehat{D}) = vec^\top(D - \widehat{D})W^{-1}vec(D - \widehat{D}) \ - tr(L^\top(\widehat{D}N)), \tag{A.1}$$

where $tr(A)$ stands for the trace of matrix A and L is the matrix of Lagrange multipliers. Using the equalities

$$vec^\top(A)vec(B) = tr(A^\top B),$$
$$vec(ABC) = (C^\top \otimes A)vec(B),$$

(A.1) becomes

$$\psi(L, \widehat{D}) = vec^\top(D - \widehat{D})W^{-1}vec(D - \widehat{D}) \ - vec^\top(L)(N^\top \otimes I_m)vec(\widehat{D}). \tag{A.2}$$

Setting the derivatives of ψ with respect to $vec(\widehat{D})$ and $vec(L)$ equal to 0 yields the following set of equations:

$$\begin{bmatrix} 2W^{-1} & -(N \otimes I_m) \\ (N^\top \otimes I_m) & 0 \end{bmatrix} \begin{bmatrix} vec(\widehat{D}) \\ vec(L) \end{bmatrix} = \begin{bmatrix} 2W^{-1}vec(D) \\ 0 \end{bmatrix}. \tag{A.3}$$

Using the fact that

$$\begin{bmatrix} A & -B \\ B^\top & 0 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} - A^{-1}B(B^\top A^{-1}B)^{-1}B^\top A^{-1} & * \\ -(B^\top A^{-1}B)^{-1}B^\top A^{-1} & * \end{bmatrix},$$

it follows from (A.3) that

$$vec(\widehat{D}) = (I - W(N \otimes I_m)[(N \otimes I_m)^\top W(N \otimes I_m)]^{-1} (N \otimes I_m)^\top)vec(D).$$

Substituting this minimizing solution into the cost function of $f(N)$ immediately gives

$$f(N) = vec^\top(D)(N \otimes I_m)[(N \otimes I_m)^\top W(N \otimes I_m)]^{-1}(N \otimes I_m)^\top vec(D).$$

Finally, the reason that $f(N) = f(NS)$ for any invertible matrix S is the fact that $\widehat{D}N = 0$ if and only if $\widehat{D}NS = 0$. So, the constraint set $\{\widehat{D} : \widehat{D}N = 0\}$ equals the set $\{\widehat{D} : \widehat{D}NS = 0\}$. □

B Proof of Theorem 4

Theorem *Define* $f(N)$ *as*

$$f(N) = vec^\top(D)(N \otimes I_m)[(N \otimes I_m)^\top W(N \otimes I_m)]^{-1}(N \otimes I_m)^\top vec(D), \quad \text{(B.1)}$$

and define $g(K)$ *as*

$$g(K) \quad = \quad f(\phi(K)) = f(N + N_\perp K), \quad \text{(B.2)}$$

with fixed $N \in \mathbb{R}^{n\times(n-r)}$ *and* $N_bot \in \mathbb{R}^{n\times r}$. *Then, the gradient of* $g(K)$ *about* $K = 0$ *is*

$$grad\ g = 2N_\perp^\top (D - B)^\top A, \quad \text{(B.3)}$$

where $A \in \mathbb{R}^{m\times(n-r)}$ *and* $B \in \mathbb{R}^{m\times n}$ *are the unique matrices that satisfy*

$$\begin{aligned} vec(A) &= [(N \otimes I_m)^\top W(N \otimes I_m)]^{-1} vec(DN) \\ vec(B) &= W vec(AN^\top). \end{aligned} \quad \text{(B.4)}$$

Proof: The gradient grad g of $g(K)$ is defined to be the unique matrix grad $g \in \mathbb{R}^{r\times(n-r)}$ for which $dg(dK) = tr((\text{grad } g)^\top dK)$. Thus, it is necessary to first compute $dg(dK)$. The first differential of $g(K)$ about $K = 0$ is given by

$$dg(dK) = df(N_\perp dK).$$

The differential df is now determined. For convenience, define the symmetric bilinear function $h(N_1, N_2)$:

$$h(N_1, N_2) \quad = \quad \frac{1}{2}[(N_1 \otimes I_m)^\top W(N_2 \otimes I_m) + (N_2 \otimes I_m)^\top W(N_1 \otimes I_m)].$$

Define $h(N) = h(N, N)$ and $h^{-1}(N) = [h(N)]^{-1}$. Then, expression (B.1) becomes

$$f(N) \quad = \quad vec^\top(DN)h^{-1}(N)vec(DN).$$

Since $dh(dN) = 2h(N, dN)$, the following holds:

$$\begin{aligned} df(dN) &= 2vec^\top(DN)h^{-1}(N)vec(DdN) - 2vec^\top(DN)h^{-1}(N)h(N, dN) \\ &\quad h^{-1}(N)vec(DN) \\ &= 2vec^\top(A)vec(DdN) - 2vec^\top(A)(N \otimes I_m)^\top W(dN \otimes I_m)vec(A), \end{aligned}$$

where A is defined in (B.4). Defining B as in (B.4), shows that df can be compactly written as

$$\begin{aligned} df(dN) &= 2tr(A^\top DdN) - 2vec^\top(B)vec(AdN^\top) \\ &= 2tr(A^\top(D - B)dN). \end{aligned}$$

Therefore, $dg(dK) = df(N_\perp dK) = 2tr(A^\top(D - B)N_\perp dK)$, verifying the expression (B.3) for the gradient of g. □

C Mathematical background to PCA

This section will attempt to give some elementary background mathematical skills that will be required to understand the process of Principal Component Analysis (PCA) [48]. This section covers the standard deviation, the variance, the covariance and the covariance matrix.

Standard deviation

To understand standard deviation, we need a data set. Statisticians are usually concerned with taking a *sample* of a *population*. To use election polls as an example, the population is equal to all the people in the country, whereas a sample is a subset of the population that the statisticians measure. The great thing about statistics is that by only measuring (e.g. by phone surveying) a sample of a population, you can work out what is most likely to be the measurement if you used the entire population.

Consider the following example set:

$$D = [1\,2\,4\,6\,12\,15\,25\,45\,68\,67\,65\,98].$$

The symbol n will be used to refer to the number of elements in set D and subscripts on the symbol D will be used to indicate a specific element in the set.

There are a number of objects that can be calculated from the data set. For example, the mean $\bar{D}$ of the sample can be calculated:

$$\bar{D} = \frac{\sum_{i=1}^{n} D_i}{n},$$

where D_i stands for the ith element of D. Unfortunately, the mean does not give a lot of information about the data set except for a sort of middle point. For example, the following data sets have exactly the same mean, but are quite different:

$$[0\,8\,12\,20] \text{ and } [8\,9\,11\,12]$$

The difference between these two data sets is the *spread*. In order to measure how spread out a data set is, the standard deviation can be calculated. The standard deviation can be defined to be equal to the average distance from the mean of the data set to a point and can be calculated as follows:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(D_i - \bar{D})^2}{n-1}},$$

where s is the usual symbol for standard deviation of a sample.

Variance

Variance is another measure of the spread of data in a data set. In fact, it is almost identical to the standard deviation. The formula for the variance is:

$$s^2 = \frac{\sum_{i=1}^{n}(D_i - \bar{D})^2}{n-1}$$

So, the variance is simply equal to the standard deviation squared.

Covariance

The last two measures are purely 1-dimensional. A data set like this could be: heights of all people in the room. However, many data sets have more than one dimension and the aim of the statistical analysis of these data sets is to see if there is any relationship between the dimensions. For example, as a data set both the height of all members of a basketball club can be given and the mark they received for winning goals. Then, statistics can be performed to see if the height of a player has any effect on his mark. Standard deviation and variance are both measures that only operate on one dimension. However, it is useful to have a similar measure to find out how much the dimensions vary with respect to each other. Covariance is such a measure. The covariance is always measured between 2 dimensions. If the covariance between one dimension and itself is calculated, actually, the variance is obtained. The covariance can be calculated as follows

$$Cov(D^{(1)}, D^{(2)}) = \frac{\sum_{i=1}^{n}(D_i^{(1)} - \bar{D}^{(1)})(D_i^{(2)} - \bar{D}^{(2)})}{n-1}$$

Take the following example data set: the heights of the basketball players are known together with their scores. So, a 2-dimensional data set is given, the first dimension H contains the heights and the second dimension S the scores. The calculation of the covariance $Cov(H, S)$ gives the following information: if the value is positive, that indicates that both dimensions increase together. So, if the height increases, the final score will also increase. If the value is negative, then one dimension increases, while the other decreases. In the last case, if the covariance is zero, it indicates that the two dimensions are independent of each other.

Covariance matrix

Covariance is always measured between two dimensions, as explained in the previous subsection. If a data set has more than two dimensions, there is more than one covariance measurement that can be calculated. For example, from a 3 dimensional data set (dimensions x, y, z), covariances $Cov(x, y), Cov(x, z)$ and $Cov(y, z)$ can be calculated. In fact, for an n-dimensional data set, $\frac{n!}{2*(n-2)!}$ different covariance values can be calculated. A useful way to get all the possible covariance values between the different dimensions is to calculate them all and

put them in a matrix. So, the definition for the covariance matrix for a set of data with n dimensions is:

$$C = (c_{ij})_{i,j} \text{ with } c_{ij} = Cov(dim_i, dim_j) \text{ for } i, j = 1, \ldots, n$$

where C is an $n \times n$ matrix and dim_x is the xth dimension. So, the covariance matrix for a 3 dimensional data set, using the usual dimensions x, y and z, is the following:

$$C = \begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix},$$

where on the diagonal the covariance value $Cov(a, a)$ is equal to the variance of a and with $Cov(a, b) = Cov(b, a)$. So, the covariance matrix is symmetric.

D STLNB algorithm

The STLNB method [95] is a generalization of the Structured Total Least Norm (STLN) method for solving overdetermined sets of m linear equations with multiple right-hand sides:

$$AX \approx B,$$

with $A \in \mathbb{R}^{m\times n}, m > n$, $B \in \mathbb{R}^{m\times d}$ and structured Toeplitz matrix $[A\,B]$. The STLNB method preserves the structure of $[A\,B]$ and allows the possibility of errors in the elements of a given data matrix $[A\,B]$. In a Toeplitz matrix $A \in \mathbb{R}^{m\times n}$, $m+n-1$ different elements are subject to error and the vector $\alpha \in \mathbb{R}^{m+n-1}$ is used to represent the corresponding elements of the Toeplitz correction matrix ΔA of A:

$$\Delta A = \begin{bmatrix} \alpha_n & \alpha_{n-1} & \dots & \alpha_1 \\ \alpha_{n+1} & \alpha_n & \dots & \alpha_2 \\ \alpha_{n+2} & \alpha n+1 & \dots & \alpha_3 \\ \vdots & & \ddots & \\ \alpha_{m+n-1} & \alpha_{m+n-2} & \dots & \alpha_m \end{bmatrix},$$

with $\alpha = [\alpha_1\alpha_2 \dots \alpha_{m+n-1}]^\top$. The correction matrix ΔA applied to A keeps the same Toeplitz structure as A. A vector β representing possible corrections in selected different elements of B is defined, which is similar to α representing errors in A. When different errors can occur in $m+d-1$ elements of B, then the corrections applied to these elements are represented by vector $\beta = [\beta_1\,\beta_2 \cdots \beta_{m+d-1}]^\top = [\beta_1\,\beta_2 \cdots \beta_d\,\alpha_1 \cdots \alpha_{m-1}]^\top$. Define $\hat{\beta} = [\beta_1 \cdots \beta_d]^\top$. Let $D_\alpha \in \mathbb{R}^{(m+n-1)\times(m+n-1)}$ be a diagonal weighting matrix that accounts for the repetition of elements of α in the correction matrix of A. Define $D_{\hat{\beta}} \in \mathbb{R}^{(m+d-1)\times(m+d-1)}$ as the diagonal weighting matrix that accounts for the repetition of elements of β in the correction matrix of B. A matrix F is introduced, so that the corrections applied to B are the same as $F\beta$: $F\beta = [F_1\beta \cdots F_d\beta]$, where each $F_i \in \mathbb{R}^{m\times(m+d-1)}$ for $i = 1, 2, \dots, d$ is defined in such a way that the corrections applied to each ith column of B are given by $F_i\beta$:

$$[F_1\beta \dots F_d\beta] = \begin{bmatrix} \beta_d & \beta_{d-1} & \dots & \beta_1 \\ \beta_{d+1} & \beta_d & \dots & \beta_2 \\ \beta_{d+1} & \beta_{d+1} & \dots & \beta_3 \\ \vdots & & \ddots & \\ \beta_{m+d-1} & \beta_{m+d-2} & \dots & \beta_m \end{bmatrix}.$$

The correction matrix $F\beta$ applied to B keeps the same structure as B. For each $F_i\beta$, $i = 1, \dots, d$ the following holds:

$$F_i\beta = J_i\hat{\beta} + \hat{F}_i\alpha,$$

where matrices J_i and $\hat{F}_i$ are defined as follows:

$$\begin{cases} \begin{array}{lll} & \begin{matrix} d-i & i \end{matrix} & \\ J_i = & \begin{bmatrix} 0 & I_i \\ 0 & 0 \end{bmatrix} & \begin{matrix} i \\ m-i \end{matrix} \end{array} \quad \text{and } \hat{F}_i = \begin{array}{lll} & \begin{matrix} m-i & n+i-1 \end{matrix} & \\ & \begin{bmatrix} 0 & 0 \\ I_{m-i} & 0 \end{bmatrix} & \begin{matrix} i \\ m-i \end{matrix} \end{array}, \text{ if } i < m \\ \\ \begin{array}{lll} & \begin{matrix} d-i & m & i-m \end{matrix} & \\ J_i = & \begin{bmatrix} 0 & I_m & 0 \end{bmatrix} & m \end{array} \quad \text{and } \hat{F}_i = 0, \quad \text{if } i \geq m (d \geq m) \end{cases}$$

Now, the STLNB problem can be formulated as that of finding $\alpha \in \mathbb{R}^{m+n-1}$, $\beta \in \mathbb{R}^{m+d-1}$ and $x = vec(X)$ such that

$$\min_{\hat{r}=0,\alpha,\beta,x} \left\| \begin{bmatrix} \hat{r}(\alpha,\beta,x) \\ D_\alpha \alpha \\ D_\beta \beta \end{bmatrix} \right\|_p, \tag{D.1}$$

where $\|\cdot\|_p$ is the vector p-norm for $p = 1, 2$ or ∞, and $\hat{r} = vec(B) + F\beta - diag(A + \Delta A, \ldots, A + \Delta A)x$ where $diag(C, \ldots, C)$ is a diagonal matrix with C on its diagonal for an arbitrary matrix C. In order to compute the STLN solution $x = vec(X)$, we need to construct an $md \times (m + n - 1)$ matrix $X_{\Delta A}$ such that

$$diag(\Delta A, \ldots, \Delta A)x = X_{\Delta A}\alpha. \tag{D.2}$$

By writing $X_{\Delta A} = [X_{\Delta A_1}\, X_{\Delta A_2}\, \ldots\, X_{\Delta A_d}]^\top$, equation (D.2) gives $\Delta A\, X_i = X_{\Delta A_i}\alpha$ for $1 \leq i \leq d$ and X_i is the ith column of X. Because $X_{\Delta A_i}$ is completely determined by the structure of ΔA and ΔA itself is independent of the column index i of X, all $X_{\Delta A_i}$'s have the same structure. Only their matrix elements are different. Namely, $X_{\Delta A_1}$ is filled with the n elements of the first column of X, X_{Δ_2} with the n elements of the second column of X etc. For example, assume that a 4×3 Toeplitz matrix A is subject to error, then the matrix ΔA is also Toeplitz with first column $[\alpha_3\, \alpha_4\, \alpha_5\, \alpha_6]^\top$ and first row $[\alpha_3\, \alpha_2\, \alpha_1]$. If we define α as $\alpha = [\alpha_1\, \alpha_2 \cdots \alpha_6]^\top$, then matrix $X_{\Delta A}$ is equal to:

$$X_{\Delta A} = \begin{bmatrix} x_3 & x_2 & x_1 & 0 & 0 & 0 \\ 0 & x_3 & x_2 & x_1 & 0 & 0 \\ 0 & 0 & x_3 & x_2 & x_1 & 0 \\ 0 & 0 & 0 & x_3 & x_2 & x_1 \end{bmatrix}.$$

The solution of problem (D.1) is computed using algorithm STLNB which is described in Algorithm 15.

The STLNB algorithm, Algorithm 15, can also be used to compute the best SWLRA matrix $\widehat{D}$ of rank r for a given Toeplitz structured matrix $D \in \mathbb{R}^{m\times n}$ with $m > n$. This approximation matrix $\widehat{D}$ can be computed as follows:

- Set $D = [A\, B]$, with $A \in \mathbb{R}^{m\times r}$ of rank r and $B \in \mathbb{R}^{m\times(n-r)}$.

Algorithm 15 STLNB algorithm

1: Input: An STLNB problem (D.1) with a specified Toeplitz matrix $[A\,B]$, $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times d}$ and a tolerance ε.
2: Choose a large number ω.
3: Set $\Delta A = 0$, $\alpha = 0$ and $\beta = 0$.
4: Compute the initial solution for $x = vec(X)$ by solving d least square problems:

$$\min_{X_i} \parallel B_i - AX_i \parallel_2,$$

where X_i and B_i are the ith column of X and B, respectively, for $i = 1, \ldots, d$. Each X_i is initialized as $(A^\top A)^{-1}A^\top B_i$.
5: Construct matrix $X_{\Delta A} \in \mathbb{R}^{md\times(m+n-1)}$ from $x = vec(X)$ as defined in equation (D.2).
6: Set $\hat{r} = [\hat{r}_1 \cdots \hat{r}_d]$ with $\hat{r}_i = B_i - AX_i$.
7: **repeat**
8: Solve the following minimization problem:

$$\min_{\substack{\Delta\alpha\\ \Delta\hat{\beta}\\ \Delta x}} \left\| \begin{bmatrix} \omega(X_{\Delta A} - \hat{F}) & -\omega J & \omega\, diag(A+\Delta A, \ldots, A+\Delta A) \\ D_\alpha & 0 & 0 \\ 0 & D_{\hat{\beta}} & 0 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta\hat{\beta} \\ \Delta x \end{bmatrix} + \begin{bmatrix} -\omega\hat{r} \\ D_\alpha\alpha \\ D_{\hat{\beta}}\hat{\beta} \end{bmatrix} \right\|_p,$$

with $\begin{bmatrix} D_{\hat{\beta}} & 0 \\ 0 & D_\alpha \end{bmatrix} = diag(1, 2, \ldots, p-1, p, p, \ldots, p, p-1, \ldots, 2, 1)$, $D_\alpha \in \mathbb{R}^{(m+n-1)\times(m+n-1)}$, $D_{\hat{\beta}} \in \mathbb{R}^{d\times d}$, $p = \min(m, n+d)$, $J = [J_1^\top \cdots J_d^\top]^\top$ and $\hat{F} = [\hat{F}_1^\top \cdots \hat{F}_d^\top]^\top$.
9: Set $x = x + \Delta x$, $\alpha = \alpha + \Delta\alpha$, $\beta(d+1 : d+m-1) = \alpha(1 : m-1)$ and $\beta(1 : d) = \beta(1 : d) + \Delta\hat{\beta}$.
10: Construct ΔA from α, $X_{\Delta A}$ from x and compute

$$\hat{r} = vec(B + F\beta) - diag(A + \Delta A, \ldots, A + \Delta A)x.$$

11: **until** ($\parallel \Delta x \parallel, \parallel \Delta\alpha \parallel, \parallel \Delta\beta \parallel \leq \varepsilon$).
12: Output: α, β, x.

- Compute the STLN solution α, β of the corresponding STLNB problem (D.1).
- Construct ΔA from α and $F\beta$ from β.
- Compute $\widehat{D} = [A + \Delta A\ B + F\beta]$.

The results presented also hold for Hankel structures, since Hankel matrices simply transform to Toeplitz matrices by permutations.

E HTLSstack algorithm

The HTLSstack method [69] is a subspace-based parameter estimation technique that operates directly on the data matrix in order to estimate the frequencies, dampings, amplitudes and phases of complex damped exponentials in the presence of noise. In various applications of digital signal processing, such as MR Spectroscopy and speech processing, complex damped exponentials are used as a model function. The signal $d(t)$ is represented as a sum of K complex damped exponentials, perturbed by independently and identically distributed (i.i.d.) complex Gaussian noise $\varepsilon(t)$ with $t = 0, 1, \ldots, N-1$:

$$d(t) = \sum_{k=1}^{K} c_k z_k^t + \varepsilon(t) = \sum_{k=1}^{K} (a_k e^{j\phi_k})(e^{(j2\pi f_k - \delta_k)/f_{sample}})^t + \varepsilon(t), \qquad \text{(E.1)}$$

with $c_k \equiv a_k e^{j\phi_k}$ the complex amplitudes and $z_k \equiv e^{(j2\pi f_k - \delta_k)/f_{sample}}$ the signal poles for $k = 1, \ldots, K$ and where f_{sample} represents the sampling frequency. The problem is to estimate the amplitudes a_k, the frequencies f_k, the phases ϕ_k and the damping factors δ_k for a given set of N data points $d(t)$, $t = 0, 1, \ldots, N-1$.

In order to estimate the signal poles, the HTLSstack approach makes use of several decimated sequences of the given signal sequence $d(t)$, $t = 0, 1, \ldots, N-1$. After the signal poles have been estimated, the phases and amplitudes are calculated as the least squares solution of equation (E.1), with z_k replaced by the estimates $\hat{z}_k$. From the original data sequence $d(0), d(1), \ldots, d(N-1)$, s different decimated sequences are formed as follows:

$$d_i = [d(i), d(i+s), \ldots, d(i+(N/s-1)s)] \equiv [d_i(0), d_i(1), \ldots d_i(N/s-1)],$$

with $i = 0, 1, \ldots, s-1$ and s the decimation factor. In order to avoid aliasing, s should be chosen such that $|f_k| < f_{sample}/(2s)$ for $k = 1, 2, \ldots, K$. Using equation (E.1), this can also be written as follows:

$$d_i(t) = \sum_{k=1}^{K} c_k z_k^{(st+i)} + \varepsilon(st+i) = \sum_{k=1}^{K} (c_{ki})(z'_k)^t + \varepsilon(st+i), \qquad \text{(E.2)}$$

with $t = 0, 1, \ldots, N/s-1$, $z'_k \equiv z_k^s$ for $k = 1, 2, \ldots, K$ and $c_{ki} \equiv c_k z_k^i$ for $k = 1, 2, \ldots, K$ and $i = 0, 1, \ldots, s-1$. Hence, a Hankel data matrix D_i constructed from a noiseless decimated sequence $d_i(t)$ can be written in terms of Vandermonde matrices:

$$D_i = \begin{bmatrix} d_i(0) & d_i(1) & d_i(2) & \ldots & d_i(N/s-m) \\ d_i(1) & d_i(2) & d_i(3) & \ldots & d_i(N/s-m+1) \\ d_i(2) & d_i(3) & \ldots & & \\ \vdots & & & & \vdots \\ d_i(m-1) & d_i(m) & \ldots & & d_i(N/s-1) \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} 1 & 1 & \dots & 1 \\ z'_1 & z'_2 & \dots & z'_K \\ \vdots & & & \\ (z'_1)^{m-1} & (z'_2)^{m-1} & \dots & (z'_K)^{m-1} \end{bmatrix} \cdot \begin{bmatrix} c_{1i} & 0 & \dots & 0 \\ 0 & c_{2i} & \dots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & c_{Ki} \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} 1 & z'_1 & \dots & (z'_1)^{N/s-m} \\ 1 & z'_2 & \dots & (z'_2)^{N/s-m} \\ \vdots & & & \\ 1 & z'_K & \dots & (z'_K)^{N/s-m} \end{bmatrix} \\
&\equiv SC_iT^\top. \qquad \text{(E.3)}
\end{aligned}
$$

Matrix D_i only contains a fraction of the available data points, i.e. N/s of the N data points. From a statistical point of view it is clear that the use of all data points will yield better results. The HTLSstack method makes use of all decimated sequences and is described below.

It is assumed that the decimated time series d_i, for $i = 0, 1, \dots, s-1$, is noiseless. Then, from (E.3) it is clear that the matrices D_i, $i = 0, 1, \dots, s-1$, have the same column space and thus share the same shift-invariant property:

$$\overline{S} = \underline{S}Z, \qquad \text{(E.4)}$$

where $\overline{S}$ and $\underline{S}$ are submatrices of S, obtained by deleting the top and the bottom row of S, respectively, and where Z is a diagonal $K \times K$ matrix with the s-th powers of the K signal poles z_k, $k = 1, 2, \dots, K$ on its diagonal. As a consequence, matrix D_{stack} constructed from s Hankel matrices D_i, $i = 0, 1, \dots, s-1$, can be decomposed as follows:

$$D_{stack} \equiv [D_0\, D_1\, \dots\, D_{s-1}] = S[C_0T^\top\, C_1T^\top\, \dots\, C_{s-1}T^\top]. \qquad \text{(E.5)}$$

Therefore, still in the noiseless case, not only the column space of D_i, $i = 0, 1, \dots, s-1$, but also that of D_{stack} share the same shift-invariant property (E.4). Using this shift-invariant property and a basis transformation via the Singular Value Decomposition (SVD) of matrix D_{stack}, the HTLSstack algorithm computes the estimated signal poles for a noisy data sequence. The HTLSstack algorithm is summarized in Algorithm 16.

The HTLSstack algorithm, Algorithm 16, can also be used to compute the best SWLRA matrix $\widehat{D}$ of rank r for a given block-row Hankel matrix $D \in \mathbb{R}^{m \times s(N/s-m+1)}$ of the form

$$
D = \begin{bmatrix} d(0) & d(1) & d(2) & \dots & d(N/s-m) \\ d(1) & d(2) & \dots & & \\ d(2) & \dots & & & \\ \vdots & & & & \vdots \\ d(m-1) & d(m) & \dots & & d(N/s-1) \end{bmatrix}
$$

Algorithm 16 HTLSstack algorithm

1: Input: a given set of N data points $d(t)$ for $t = 0, 1, \ldots, N-1$.
2: Arrange the data points $d(t)$, $t = 0, 1, \ldots, N-1$ in a matrix D_{stack} of dimensions $m \times s(N/s - m + 1)$ with $m \geq K$ and $N/s - m > K$ as in equation (E.5).
3: Compute the SVD of D_{stack} and truncate D_{stack} to rank K:

$$D_{stack} = U \Sigma V^H,$$

$$D_{stack}^{trun} = U_K \Sigma_K V_K^H,$$

where U_K and V_K consist of the first K columns of U and V, respectively, and Σ_K is the $K \times K$ upper left submatrix of Σ.
4: Impose the shift-invariant property and solve the following overdetermined set of equations in Total Least Squares (TLS) sense (see Theorem 2 for the TLS-solution):

$$\overline{U_K} \approx \underline{U_K} E,$$

where $\overline{U_K}$ and $\underline{U_K}$ are submatrices of U_K, obtained by deleting the first and the last row of U_K, respectively.
The eigenvalues of E are the estimates $\hat{z}'_k$, $k = 1, 2, \ldots, K$ of the s-th powers of the signal poles z_k, from which the frequencies $\hat{f}_k$ and dampings $\hat{\delta}_k$ are easily obtained.
5: Fill in the estimated $\hat{z}'_k$ into the model function of (E.2) and solve the obtained system of N equations in Least Squares sense to determine the estimated $\hat{c}_{ki}$, for $k = 1, 2, \ldots, K$ and $i = 0, 1, \ldots, s-1$, of c_{ki}, from which the amplitudes $\hat{a}_k$ are easily obtained.
6: Output: estimated amplitudes $\hat{a}_k$, phases $\hat{\phi}_k$, frequencies $\hat{f}_k$ and damping factors $\hat{\delta}_k$.

with $d(t) = [d_0(t)\, d_1(t)\, \ldots\, d_{s-1}(t)]$ a row vector of length s, for $t = 0, \ldots, N/s - 1$. The structure-preserving approximation matrix $\widehat{D}$ of rank r can be computed as follows:

- Construct matrix $D_{stack} = [D_0\, D_1\, \ldots\, D_{s-1}]$ with

$$D_i = \begin{bmatrix} d_i(0) & d_i(1) & d_i(2) & \ldots & d_i(N/s-m) \\ d_i(1) & d_i(2) & d_i(3) & \ldots & d_i(N/s-m+1) \\ d_i(2) & d_i(3) & \ldots & & \\ \vdots & & & & \vdots \\ d_i(m-1) & d_i(m) & \ldots & & d_i(N/s-1) \end{bmatrix},$$

  for $i = 0, 1, \ldots, s-1$.

- Apply the HTLSstack algorithm to matrix D_{stack} with $K = r$ in order to find the estimated $\hat{z}'_k$ and $\hat{c}_{ki}$ for $k = 1, 2, \ldots, K$ and $i = 0, 1, \ldots, s-1$.

- Construct
$$\hat{d}_i(t) = \sum_{k=1}^{K} (\hat{c}_{ki})(\hat{z}'_k)^t,$$
for $i = 0, 1, \ldots, s-1$ and $t = 0, 1, \ldots, N/s - 1$.

- Construct
$$\widehat{D}_i = \begin{bmatrix} \hat{d}_i(0) & \hat{d}_i(1) & \hat{d}_i(2) & \ldots & \hat{d}_i(N/s-m) \\ \hat{d}_i(1) & \hat{d}_i(2) & \hat{d}_i(3) & \ldots & \hat{d}_i(N/s-m+1) \\ \hat{d}_i(2) & \hat{d}_i(3) & \ldots & & \\ \vdots & & & & \\ \hat{d}_i(m-1) & \hat{d}_i(m) & \ldots & & \hat{d}_i(N/s-1) \end{bmatrix},$$
for $i = 0, 1, \ldots, s-1$.

- Reconstruct
$$\widehat{D} = \begin{bmatrix} \hat{d}(0) & \hat{d}(1) & \hat{d}(2) & \ldots & \hat{d}(N/s-m) \\ \hat{d}(1) & \hat{d}(2) & \ldots & & \\ \hat{d}(3) & \ldots & & & \\ \vdots & & & & \vdots \\ \hat{d}(m-1) & \hat{d}(m) & \ldots & & \hat{d}(N/s-1) \end{bmatrix},$$
with $\hat{d}(t) = [\hat{d}_0(t)\, \hat{d}_1(t) \ldots \hat{d}_{s-1}(t)]$ a row vector of length s, for $t = 0, \ldots, N/s - 1$.

Bibliography

[1] T. J. Abatzoglou and J. M. Mendel, Constrained total least squares. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, pp. 1485 – 1488, 1987.

[2] T. J. Abatzoglou, J. M. Mendel and G. A. Harada, The constrained total least squares technique and its applications to harmonic super-resolution. *IEEE Transactions on Signal Processing*, vol. 39, pp. 1070 – 1087, 1991.

[3] D. Achlioptas, A. Fiat, A. R. Karlin, F. McSherry, Web search via Hub Synthesis. *IEEE Symposium on Foundations of Computer Science*, pp. 500 – 509, 2001.

[4] D. Achlioptas, F. Mc Sherry, Fast computation of low rank matrix approximation, in Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC 01), Hersonissos, Greece, 2001, pp. 611 - 618.

[5] R. J. Adcock, A problem in least squares. *The Analyst*, vol. 4, pp. 183 – 184, 1877.

[6] K. S. Arun, A unitarily constrained total least-squares problem in signal-processing. *SIAM Journal on Matrix Analysis and Applications*, vol. 13, pp. 729 – 745, 1992.

[7] G. Boutry, M. Elad, G. Golub, P. Milanfar, The Generalized Eigenvalue Problem for Non-Square Pencils Using a Minimal Perturbation Approach. *SIAM Journal on Matrix Analysis and Applications*, vol. 27(2), pp. 582 – 601, Nov. 2005.

[8] Y. Bresler and A. Macovski, Exact Maximum Likelihood parameter estimation of superimposed exponential signals in noise. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34(5), pp. 1081 – 1089, Oct. 1986.

[9] A. Burnham, R. Viveros and J. MacGregor, Frameworks for latent variable multivariate regression. *Journal of Chemometrics*, vol. 10, pp. 31 – 46, 1996.

[10] J. A. Cadzow, Signal enhancement: a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36(1), pp. 49 – 62, Jan. 1988.

[11] C.-L. Cheng and J.W. Van Ness, Statistical regression with measurement error. Arnold, London, 1999.

[12] D. Cheng, R. Kannan, S. Vempala, G. Wang, A divide-and-merge methodology for clustering. *Proceedings of the ACM Symposium on Principles of Database Systems*, 2005.

[13] M. T. Chu, R. E. Funderlic, R. J. Plemmons, Structured Lower Rank Approximation. *Linear Algebra and Its Applications*, vol. 366, pp. 157 – 172, 2003.

[14] A. Cuyt, G. Golub, P. Milanfar, B. Verdonk, Multidimensional integral inversion with applications in shape reconstruction. *SIAM Journal on Scientific Computing*, vol. 27(3), pp. 1058 – 1070, 2005.

[15] P. Davis, Triangle formulas in the complex plane. *Mathematics of Computation*, vol. 18, pp. 569 – 577, 1964.

[16] P. Davis, Plane regions determined by complex moments. *Journal of Approximation Theory*, vol. 19, pp. 148 – 153, 1977.

[17] R. D. Degroat and E. M. Dowling, The data least squares problem and channel equalization. *IEEE Transactions on Signal Processing*, vol. 41, pp. 407 – 411, 1993.

[18] S. de Jong, PLS fits closer than PCR. *Journal of Chemometrics*, vol. 7, pp. 551 – 557, 1993.

[19] S. de Jong, Regression coefficients in multilinear PLS. *Journal of Chemometrics*, vol. 12(1), pp. 77 – 81, 1998.

[20] L. De Lathauwer, Signal Processing based on multilinear algebra, Ph.D.thesis, KULeuven, 1997.

[21] L. De Lathauwer, B. De Moor, J. Vandewalle, On the best rank-1 and rank $(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, vol. 21(4), pp. 1324 – 1342, 2000.

[22] B. De Moor, The singular value decomposition and long and short spaces of noisy matrices. *IEEE Transactions on Signal Processing*, vol. 41(9), pp. 2826 – 2838, 1993.

[23] B. De Moor, Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing*, vol. 42, pp. 3004-3113, 1994.

[24] E. M. Dowling and R. D. Degroat, The equivalence of the total least-squares and minimum norm methods. *IEEE Transactions on Signal Processing*, vol. 39, pp. 1891 – 1892, 1991.

[25] G. Eckart and G. Young, The approximation of one matrix by another of lower rank. *Psychometrica*, vol. 1, pp. 211 – 218, 1936.

[26] M. Elad, P. Milanfar and G. H. Golub, Shape from Moments–An Estimation Theory Perspective. *IEEE Transactions on Signal Processing*, vol. 52, pp. 1814 – 1829, 2004.

[27] Y. Ephraim, H. L. Van Trees, A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, vol. 3(4), pp. 251 – 266, 1995.

[28] 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*, vol. 18, pp. 1223 – 1241, 1997.

[29] R. Fletcher, Practical Methods of Optimization. John Wiley & Sons, New York, 1987.

[30] A. Frieze, R. Kannan, S. Vempala, Fast Monte-Carlo algorithms for finding low rank approximations, in Proceedings of the 39th Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1998), pp. 370 – 378.

[31] W. A. Fuller, Error measurement models. John Wiley, New York, 1987.

[32] D. G. Gadian, Nuclear Magnetic Resonance and its applications to living systems. Clarendon Press, Oxford, 1982.

[33] L. J. Gleser, Estimation in a multivariate "errors in variables" regression model : Large sample results. *Annals of Statistics*, vol. 9, pp. 24 – 44, 1981.

[34] A. Goldenshluger, V. Spokoiny, On the shape from moments problem and recovering edges from noisy radon data. *Probability Theory and Related Fields*, vol. 128(1), pp. 123 – 140, Jan. 2004.

[35] G. H. Golub, Some modified matrix eigenvalue problems. *Siam Review*, vol. 15, pp. 318 – 344, 1973.

[36] G. H. Golub, C. F. Van Loan, An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis* vol. 17, pp. 883 – 893, 1980.

[37] G. H. Golub, A. Hoffman, G. W. Stewart, A generalization of the Eckart-Young-Mirsky matrix approximation theorem. *Linear Algebra and its Applications*, vol. 88/89, pp. 322 – 327, 1987.

[38] G. H. Golub and C. F. Van Loan, Matrix computations. 3rd ed., The Johns Hopkins Univ. Press, Baltimore, 1996.

[39] G. H. Golub, P. Milanfar, and J. Varah, A Stable Numerical Method for Inverting Shape from Moments. *SIAM Journal on Scientific Computing*, vol. 21(4), pp. 1222 – 1243, Dec. 1999.

[40] P. Guillaume and R. Pintelon, A Gauss–Newton-like optimization algorithm for "weighted"nonlinear least-squares problems. *IEEE Transactions on Signal Processing*, vol. 44(9), pp. 2222 – 2228, 1996.

[41] R. F. Gunst, J. T. Webster and R. L. Mason, A comparison of least squares and latent root regression estimators. *Technometrics*, vol. 18, pp. 75 – 83, 1976.

[42] B. Gustafsson, C. He, P. Milanfar, M. Putinar, Reconstructing planar domains from their moments. *Inverse Problems*, vol. 16(4), pp. 1053 – 1070, Aug. 2000.

[43] L. Hermoye, L. Annet, P. Lemmerling, F. Peeters, F. Jamar, P. Gianello, S. Van Huffel, B. E. Van Beers, Calculation of the renal perfusion and glomerular filtration rate from the renal impulse response obtained with MRI. *Magnetic Resonance in Medicine*, vol. 51, pp. 1017 – 1025, 2004.

[44] K. Hermus, P. Wambacq, Assessment of signal subspace based speech enhancement for noise robust speech recognition. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Montreal, Canada, vol. I, pp. 945 – 948, 2004.

[45] Y. Hua, T. K. Sarkar, Matrix pencil method and its performance, Proceedings of IEEE ICASSP, pp. 2476 – 2479, 1988.

[46] J. Jensen, R. Heusdens, S. H. Jensen, A perceptual subspace approach for modeling of speech and audio signals with damped sinusoids. *IEEE Transactions on Speech and Audio Processing*, vol. 12, pp. 121 – 132, 2004.

[47] T. Jiang, N. D. Sidiropoulos, J. M. F. ten Berge, Almost Sure Identifiability of Multidimensional Harmonic Retrieval. *IEEE Transactions on Signal Processing*, vol. 9, pp. 1849 – 1859, September 2001.

[48] I. T. Joliffe, Principal Component Analysis. Springer, New York, 1986.

[49] T. Kailath and A. H. Sayed, editors, Fast reliable algorithms for matrices with structure. SIAM, Philadelphia, 1999.

[50] A. Kukush, I. Markovsky and S. Van Huffel, Consistency of the structured total least squares estimator in a multivariate model. *Journal of Statistical Planning and Inference*, vol. 133(2), pp. 315 – 358, June 2005.

[51] A. Kukush and S. Van Huffel, Consistency of elementwise-weighted total least squares estimator in a multivariate errors-in-variables model AX=B. *Metrika*, vol. 59(1), pp. 75 – 97, 2004.

[52] P. Lemmerling, Structured total least squares: analysis, algorithms and applications, Ph.D. thesis, Dept. of Electrical Engineering, Katholieke Universiteit Leuven, Belgium, 1999.

[53] P. Lemmerling, N. Mastronardi and S. Van Huffel, Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms*, vol. 23, pp. 371-392, 2000.

[54] P. Lemmerling, L. Vanhamme, S. Van Huffel, B. De Moor, IQML-like algorithms for solving structured total least squares problems: a unified view. *Signal Processing*, vol. 81, pp. 1935 – 1945, 2001.

[55] P. Lemmerling, S. Van Huffel, Structured total least squares: analysis, algorithms and applications, in Total Least Squares and Errors-in-Variables Modelling, edited by S. Van Huffel, P. Lemmerling, Kluwer Academic Publishers, pp.79 – 91, 2002.

[56] W. Li and J. Qin, Consistent dynamic PCA based on errors-in-variables subspace identification. *Journal of Process Control*, vol. 11, pp. 661 – 678, 2001.

[57] Y. Li, K. J. R. Liu, J. Razavilar, A parameter estimation scheme for damped sinusoidal signals based on low-rank Hankel approximation. *IEEE Transactions on Signal Processing*, vol. 45, pp. 481 – 486, Feb. 1997.

[58] K. J. R. Liu, D. P. O'Leary, G. W. Stewart, Y. J. Wu, URV ESPRIT for tracking time-varying signals. *IEEE Transactions on Signal Processing*, vol. 42, pp. 3441 – 3448, Dec. 1994.

[59] W.-S. Lu, S.-C. Pei, P.-H. Wang, Weighted low rank approximation of general complex matrices and its application in the design of 2-D digital filters. *IEEE Transactions on Circuits Systems I*, vol. 44, pp. 650 – 655, July 1997.

[60] F. Luk and D. Vandevoorde, Decomposing a signal into a sum of exponentials, Iterative Methods in Scientific Computing, R. Chan, T. Chan, and G. Golub, eds., Springer-Verlag, Singapore, 1997, pp. 329 – 357.

[61] J. H. Manton, R. Mahony and Y. Hua, The geometry of weighted low rank approximations. *IEEE Transactions on Signal Processing*, vol. 51(2), pp. 500 – 514, 2003.

[62] I. Markovsky, M. L. Rastello, A. Premoli, A. Kukush and S. Van Huffel, The element-wise weighted total least squares problem. *Journal of Computational Statistics and Data Analysis*, vol. 50(1), pp. 181 – 209, Oct. 2005.

[63] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor and R. Pintelon, Application of structured total least squares for system identification and model reduction. *IEEE Transactions on Automatic Control*, Special Issue

on System Identification: Linear vs Nonlinear, vol. 50(10), pp. 1490 – 1500, Oct. 2005.

[64] I. Markovsky, J. C. Willems, B. De Moor, S. Van Huffel, Exact and Approximate Modeling of Linear Systems: A Behavioral Approach. SIAM, Philadelphia, US, 2006.

[65] H. Martens and T. Naes, Multivariate Calibration. John Wiley & Sons, New York, 1989.

[66] N. Mastronardi, P. Lemmerling and S. Van Huffel, Fast regularized structured total least squares algorithm for solving the basic deconvolution problem. *Numerical Linear Algebra with Applications*, vol. 12, pp. 201 – 209, 2005.

[67] P. Milanfar, G. Verghese, W. Karl, A. Willsky, Reconstructing polygons from moments with connections to array processing. *IEEE Transactions on Signal Processing*, vol. 43, pp. 432 – 443, 1995.

[68] L. Mirsky, Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics Oxford*, vol. 11, pp. 50 – 59, 1960.

[69] G. Morren, P. Lemmerling and S. Van Huffel, Decimative subspace-based parameter estimation techniques. *Signal Processing*, vol. 83(5), pp. 1025 – 1033, May 2003.

[70] C. Paige, Z. Strakoš, Core problems in linear algebraic systems. *SIAM Journal on Matrix Analysis and Applications*, vol. 27(3), pp. 861 – 875, 2006.

[71] J. M. Papy, L. De Lathauwer, S. Van Huffel, Exponential data fitting using multilinear algebra: The decimative case. *IEEE Transactions on Signal Processing*, to appear.

[72] J. M. Papy, L. De Lathauwer, S. Van Huffel, Exponential data fitting using multilinear algebra: The single-channel and multichannel case. *Numerical Linear Algebra with Applications*, vol. 12(8), pp. 809 – 826, Oct. 2005.

[73] A. Phatak, Evalauation of some multivariate methods and their applications in chemical engineering. PhD thesis, University of Waterloo, 1993.

[74] A. Phatak and S. de Jong, The geometry of partial least squares. *Journal of Chemometrics*, vol. 11(4), pp. 311 – 338, 1997.

[75] A. Premoli and M. L. Rastello, The parametric quadratic form method for solving problems with element-wise weighting. In S. Van Huffel and P. Lemmerling, editors, Total least squares and errors-in-variables modeling: Analysis, algorithms and applications, pp. 67 – 76, Kluwer, 2002.

[76] J. Razavilar, Y. Li, K. J. R. Liu, Structured low-rank matrix pencil for spectral estimation and system identification. *Signal Processing*, vol. 65, pp. 363 – 372, March 1998.

[77] J. B. Rosen, H. Park, and J. Glick, Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Analysis and Applications*, vol. 17(1), pp. 110 – 126, 1996.

[78] R. Roy and T. Kailath, Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37(7), pp. 984 – 995, July 1989.

[79] M. Schuermans, P. Lemmerling and S. Van Huffel, Structured Weighted Low Rank Approximation. *Numerical Linear Algebra with Applications*, vol. 11, pp. 609-618, 2004.

[80] M. Schuermans, P. Lemmerling, S. Van Huffel, Block-row Hankel Weighted Low Rank Approximation. *Numerical Linear Algebra with Applications*, in press.

[81] M. Schuermans, I. Markovsky, P. D. Wentzell, S. Van Huffel, On the equivalence between Total Least Squares and Maximum Likelihood PCA. *Analytica Chimica Acta*, vol. 544(1-2), pp. 254 – 267, 2005.

[82] M. Schuermans, P. Lemmerling, L. De Lathauwer, S. Van Huffel, The use of total least squares data fitting in the shape from moments problem. *Signal Processing*, vol. 86(5), pp. 1109 – 1115, 2006.

[83] N. D. Sidiropoulos, Generalizing Caratheodory's Uniqueness of Harmonic Parameterization to N Dimensions. *IEEE Transactions on Information Theory*, vol. 4, pp. 1687 – 1690, May 2001.

[84] D. Sima, S. Van Huffel and G. H. Golub, Regularized Total Least Squares based on quadratic eigenvalue problem solvers. *BIT Numerical Mathematics*, vol. 44(4), pp. 793 – 812, Dec. 2004.

[85] H. D. Simon, H. Zha, Low rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM Journal on Scientific Computing*, 21(6), pp. 2257 – 2274.

[86] V. Strakhov, M. Brodsky, On the uniqueness of the inverse logarithmic potential problem. *SIAM Journal of Applied Mathematics*, vol. 46, pp. 324 – 344, 1986.

[87] E. V. Thomas, Errors-in-variables estimation in multivariate calibration. *Technometrics*, vol. 33, pp. 405 – 413, 1991.

[88] C. F. Tirendi and J. F. Martin, Quantitative Analysis of NMR Spectra by Linear Prediction and Total Least Squares. *Journal of Magnetic Resonance*, vol. 85, pp. 162 – 169, Oct. 1989.

[89] A. van der Veen, E. F. Deprettere, SVD-based low-rank approximations of rational models, in SVD and Signal Processing II, edited by R. Vaccaro, Elsevier Science Publishers B. V., pp. 431 – 454, 1991.

[90] L. Vanhamme and S. Van Huffel, Multichannel quantification of biomedical Magnetic Resonance Spectroscopic signals. Invited paper for the session on 'Sensor array signal processing', Proceedings SPIE conference on Advanced Signal Processing: Algorithms, Architectures, and Implementations VIII, ed. F.T. Luk, San Diego, pp. 237 – 248, 22-24 July, 1998.

[91] 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 of Matrix Analysis and Applications*, vol. 10(3), pp. 294 – 315, 1989.

[92] S. Van Huffel and J. Vandewalle, The total least squares problem: computational aspects and analysis. Frontiers in Applied Mathematics series, vol. 9, SIAM, Philadelphia, 1991.

[93] S. Van Huffel, L. Aerts, J. Bervoets, J. Vandewalle, C. Decanniere, P. Van Hecke, Improved quantitative time-domain analysis of NMR data by total least squares, in Signal Processing VI: Theories and Applications, (Vandewalle J., Boite R., Moonen M., and Oosterlinck A., eds.), Elsevier, pp. 1721 – 1724, 1992.

[94] S. Van Huffel, H. Chen, C. Decanniere, P. Van Hecke, Algorithm for time-domain nmr data fitting based on total least squares. *Journal of Magnetic Resonance A*, vol. 110, pp. 228 – 237, 1994.

[95] S. Van Huffel, H. Park, J. B. Rosen, Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Transactions on signal processing*, vol. 44(10), pp. 2464 – 2474, Oct. 1996.

[96] S. Van Huffel, editor, Recent advances in total least squares techniques and errors-in-variables modeling. SIAM Proceedings series, SIAM, Philadelphia, 1997.

[97] S. Van Huffel and P. Lemmerling, editors, Total least squares and errors-in-variables modeling: Analysis, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 2002.

[98] G. A. Watson, On a class of algorithms for total approximation. *Journal of Approximation Theory*, vol. 45, pp. 219 – 231, 1985.

[99] G. A. Watson, On a general class of matrix nearness problems. *Constructive Approximation*, vol. 7, pp. 299 – 314, 1991.

[100] J. T. Webster, R. F. Gunst and R. L. Mason, Latent root regression analysis. *Technometrics*, vol. 16, pp. 513 – 522, 1974.

[101] P. D. Wentzell, D. T. Andrews and B. R. Kowalski, Maximum likelihood multivariate calibration. *Analytical Chemistry*, vol. 69, pp. 2299 – 2311, 1997.

[102] P. D. Wentzell, D. T. Andrews, D. C. Hamilton, K. Faber and B. R. Kowalski, Maximum likelihood principal component analysis. *Journal of Chemometrics*, vol. 11, pp. 339 – 366, 1997.

[103] P. D. Wentzell and M. T. Lohnes, Maximum likelihood principal component analysis with correlated measurement errors: theoretical and practical considerations. *Chemometrics and Intelligent Laboratory Systems*, vol. 45, pp. 65 – 85, 1999.

[104] R. Wehrens, W. van der Linden, Bootstrapping principal regression models. *Journal of Chemometrics*, vol. 11, pp. 157 – 172, 1997.

[105] S. Wold, A. Ruhe, H. Wold and W. J. Dunn, The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, vol. 5(3), pp. 735 – 743, 1984.

[106] H. Yan and J. C. Gore, The relation of HSVD to LPSVD for fitting time-domain signals. *Journal of Magnetic Resonance*, vol. 80, pp. 324 – 327, Nov. 1988.

[107] Z. Zhang, H. Zha, H. D. Simon, Low rank approximations with sparse factors I: basic algorithms and error analysis. *SIAM Journal on Matrix Analysis and Applications*, vol. 23(3), pp. 706 – 727, 1999.

[108] M. D. Zoltowski, Generalized minimum norm and constrained total least squares with applications to array processing. In Proceedings SPIE, Advanced Algorithms and Architectures for Signal Processing, III, San Diego, CA, vol. 975, pp. 78 – 85, 1988.

List of publications

Articles in International Journals

- M. Schuermans, P. Lemmerling, S. Van Huffel. Structured weighted low rank approximation, Numerical Linear Algebra with Applications, vol. 11, no. 5-6, Jun. 2004, pp. 609 – 618.

- M. Schuermans, P. Lemmerling, S. Van Huffel. Block-row Hankel Weighted Low Rank Approximation, Accepted for publication in Numerical Linear Algebra with Applications.

- M. Schuermans, I. Markovsky, P. D. Wentzell, S. Van Huffel. On the equivalence between Total Least Squares and Maximum Likelihood PCA, Analytica Chimica Acta, vol. 544, no. 1-2, Jul. 2005, pp. 254 – 267.

- M. Schuermans, P. Lemmerling, L. De Lathauwer, S. Van Huffel. The use of total least squares data fitting in the shape from moments problem, Signal Processing, vol. 86, no. 5, May 2006, pp. 1109 – 1115.

- N. Mastronardi, M. Schuermans, M. Van Barel, R. Vandebril, S. Van Huffel. A Lanczos-like reduction of symmetric structured matrices into semiseparable ones, Calcolo, vol. 42, no. 3-4, Aug. 2005, pp. 227 – 241.

Articles in Conference Proceedings

- N. Mastronardi, M. Schuermans, M. Van Barel, R. Vandebril, S. Van Huffel. The Lanczos reduction to semiseparable matrices, in Proceedings of the 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Paris, France, July 11 – 15, 2005, pp. 1 – 6.

Abstracts for International Conferences

- M. Schuermans, P. Lemmerling, S. Van Huffel. Hankel Weighted Low Rank Approximation, Conference on Preconditioning methods for optimal control and constrained optimization (PMOCCO), Nijmegen, Nederland, Oct. 23 – 25, 2002.
- M. Schuermans, P. Lemmerling, S. Van Huffel. Hankel Weighted Low Rank Approximation, Third workshop of the ERCIM Working Group on Matrix Computations and Statistics, Neuchatel, Zwitserland, Nov. 9 – 10, 2002.
- M. Schuermans, P. Lemmerling, S. Van Huffel. Block-Hankel Weighted Low Rank Approximation, Numerical Linear Algebra and its Applications, International school and workshop, Monopoli (Bari), Italy, Sept. 22 – 24, 2003.
- M. Schuermans, I. Markovsky, P. D. Wentzell, S. Van Huffel. On the equivalence between Total Least Squares and Maximum Likelihood PCA, 9th Chemometrics in Analytical Chemistry (CAC) Conference, Lisbon, Portugal, Sept. 20 – 23, 2004.
- M. Schuermans, I. Markovsky, P. D. Wentzell, S. Van Huffel. Equivalent methods for the low rank matrix approximation problem, Sixth workshop of the ERCIM Working Group on Matrix Computations and Statistics, Copenhagen, Denmark, April 1 – 3, 2005.
- N. Mastronardi, M. Schuermans, S. Van Huffel, M. Van Barel, and R. Vandebril. The Lanczos reduction to semiseparable matrices and applications in computing dominant subspaces, The Sixteenth International Workshop on Operator Theory and Applications, IWOTA 2005, University of Connecticut, Storrs, USA, July 24 – 27, 2005.
- N. Mastronardi, M. Schuermans, M. Van Barel, R. Vandebril, S. Van Huffel. The Lanczos reduction to semiseparable matrices, 22nd GAMM-Seminar Leipzig on Large Scale Eigenvalue Computations, Leipzig, Germany, Jan. 19 – 21, 2006.

Internal Reports (submitted papers)

- M. Schuermans, I. Markovsky, S. Van Huffel. An adapted version of the Element-wise Weighted Total Least Squares method for applications in chemometrics, Submitted to Chemometrics and Intelligent Laboratory Systems.

Curriculum Vitae

Mieke Schuermans was born in Maaseik (Belgium) on April 3, 1978. In 1998, she obtained the Bachelors degree in Mathematics at the 'Limburgs Universitair Centrum' in Diepenbeek, Belgium. She qualified as a Master of Science in Mathematics, option pure Mathematics, at the 'Katholieke Universiteit Leuven' in 2000. Her master thesis was in the field of group theory devoted to '*Asymptotic Prime Divisors*' and was obtained under the supervision of Prof. dr. Raymond Debremaeker. In 2002 she received a Teaching Degree in Sciences at the Katholieke Universiteit Leuven.
In September 2000, she worked part-time as a teacher at the 'Voorbereidend Instituut' and part-time as a course tutor in Algebra for the students of the first year of engineering at the 'Katholieke Universiteit Leuven'. In September 2001, she started working as a researcher at the Electrical Engineering Department ESAT in the research unit of Signals, Identification, System Theory and Automation (SCD/SISTA) at the Katholieke Universiteit Leuven. During her research she kept on tutoring the students in Algebra. Her supervisor was Prof. dr. ir. Sabine Van Huffel.