Optimization Methods and Large-Scale Algorithms in Small Area Estimation

Doctoral Thesis

Submitted to the Department IV at the University of Trier in partial fulfillment of the Requirements for the Degree of

Doctor Rerum Naturalium
(Dr. rer. nat.)

by

M.Sc. Julian Alexander Wagner

Trier, 2018

The present thesis was developed within the research training group on Algorithmic Optimization (ALOP) at the University of Trier.

Supervisors: Prof. Dr. Ralf Münnich (University of Trier)
Prof. Dr. Ekkehard Sachs (University of Trier)
Prof. Dr. Göran Kauermann (LMU Munich)
Prof. Dr. Volker Schulz (University of Trier)

Submission date: September 19, 2018

Disputation date: November 30, 2018

Acknowledgment

First of all, I would like to express my deepest gratitude to Prof. Dr. Ralf Münnich for giving me the opportunity to conduct research under his professional guidance. His constant support and the inspiring discussions have made an essential contribution to this thesis. In the same vein, I would like to extend my sincere thanks to Prof. Dr. Ekkehard Sachs for sharing his knowledge and passion on the topic of numerical optimization. He strongly inspired my academic work already from the beginning of my studies. I really enjoyed the cooperation with both of them on scientific but also on personal basis.

I also want to thank all my colleagues from the department of mathematics as well as the department of survey statistics for creating a pleasant, cooperative, and amicable working atmosphere. Some of them became good friends over the years and without their support and proofreading this thesis would not have reached its present form.

I am grateful for being part of the research training group on Algorithmic Optimization (ALOP) at the University of Trier, funded by the German Research Foundation. The provided financial resources and educational programs allowed me to broaden my knowledge in a number of ways and to gain insights into a variety of interesting research topics.

Last, but definitely not least, my heartfelt thanks goes to my family, especially to my parents, for their loving and unconditional support at all times and in every circumstances. My professional and personal career would not have been possible without them.

Contents

German Summary

Im Rahmen von Stichprobenerhebungen ist es immer häufiger von Interesse nicht nur Statistiken für eine Grundgesamtheit, sondern auch für bestimmte Unterpopulationen auszuweisen. Dabei können mitunter sehr kleine Teilstichprobenumfänge auftreten, so dass klassische Schätzverfahren keine Schätzungen mit ausreichender Genauigkeit für diese Subgruppen mehr erlauben. In diesem speziellen Fall können moderne Schätzverfahren, die so genannten Small Area-Verfahren, Abhilfe schaffen. Diese sind so konzipiert, dass sie die erhobenen Stichprobendaten mit weiteren verfügbaren Hilfsinformationen über ein statistisches Modell verbinden, um so auch trotz der teilweise sehr geringen Teilstichprobenumfänge noch akkurate Schätzungen zu ermöglichen. Im Gegensatz zu den klassischen Verfahren weisen die Small Area-Verfahren zwar Verzerrungen auf, kompensieren diese aber in der Regel durch eine wesentlich geringere Variabilität. Die Verzerrungen resultieren dabei zumeist aus unzutreffenden, aber vereinfachenden Modellannahmen, wie beispielsweise einem (verallgemeinert) linearen Zusammenhang zwischen der Untersuchungsvariablen und den zusätzlich verwendeten Hilfsinformationen. Eine möglichst flexible und realitätsnahe Modellierung lässt daher auf eine Verbesserung der Small Area-Schätzungen hoffen. Aus diesem Grund beschäftigt sich die vorliegende Arbeit unter anderem mit den folgenden Fragestellungen:

1. Wie lassen sich nichtlineare und hochflexible Modellierungsmethoden in die Small Area-Verfahren einbinden und wie wirkt sich diese Modellierung auf die Small Area-Schätzungen aus?

2. Wie können wieter Informationen über den (globalen) Zusammenhang zwischen den Variablen, so genannte Shape-Constraints, in der Modellierung berücksichtigt werden, ohne dabei die Flexibilität des Modells einzuschränken?

Um eine möglichst flexible Modellierung zu erreichen, wird in der vorliegenden Arbeit die penalisierte Spline-Methode vorgestellt. Deren Verwendung für die Small Area-Statistik erfolgt hier über einen neuartigen Ansatz, welcher die Modellparameter über die Lösung eines Optimierungsproblems bestimmt. Auf Basis dieser Reformulierung lassen sich schließlich beliebige Shape-Constraints in Form von linearen Ungleichungsnebenbedingungen in das Optimierungsproblem einarbeiten. Insgesamt resultiert ein innovativer Small Area-Schätzer, welcher sowohl hochgradig komplexe Zusammenhänge in den Daten als auch Informationen über deren globalen Verlauf berücksichtigt. Dies wurde in der Literatur bislang noch vernachlässigt. Um die Verwendbarkeit dieses Small Area-Schätzers in der Praxis zu ermöglichen, wird anschließend eine Resampling-Methode zur Schätzung des zugehörigen mittleren quadratischen Fehlers herausgearbeitet. Anhand einer Simulationsstudie und einer ausgewählten praktischen Anwendung wird das Potential dieser neuartigen Herangehensweise verdeutlicht, aber auch etwaige Schwierigkeiten kritisch diskutiert.

Neben einer geeigneten Modellierung spielen auch Qualität und Verfügbarkeit der verwendeten Hilfsinformationen eine entscheidende Rolle. In diesem Zusammenhang liefern Big Data-Anwendungen völlig neue und andersartige Möglichkeiten Small Area-Schätzungen zu verbessern, stellen aber gleichzeitig Herausforderungen für die derzeit existierenden Methoden dar. Um die eigentlich in großer Menge verfügbaren Hilfsinformationen für die Small Area-Schätzungen zu nutzen, müssen die entwickelten Verfahren multivariabel, also für die Verwendung mehrerer Hilfsvariablen, erweitert werden. Im Gegensatz zu den klassischen Verfahren ist dies für die flexiblen splinebasierten Modelle nicht trivial und liefert eine weitere Problemstellung, welche in dieser Arbeit diskutiert wird:

3. Wie lassen sich multiple Hilfsinformationen in die splinebasierten Small Area-Modelle unter Erhaltung der Struktur einbeziehen?

Zu diesem Zweck wird in dieser Arbeit ein Tensorprodukt-Ansatz vorgestellt, welcher eine strukturerhaltende multivariable Erweiterung der penalisierten Spline-Methode ermöglicht. Da diese Erweiterung also die allgemeine Struktur der Splines und daher die des betrachteten Optimierungsproblems beibehält, lassen sich die Shape-Constraints in Analogie zum eindimensionalen Fall berücksichtigen.

Eine gravierende Einschränkung bei der Verwendung von Tensorprodukten liegt darin, dass mit der Hinzunahme jeder weiteren Hilfsvariablen die Dimension des zugrunde liegenden Optimierungsproblems, und damit insbesondere der Speicher- und Rechenaufwand der klassischen Lösungsalgorithmen, exponentiell anwächst. Dies führt dazu, dass die in dieser Arbeit auftretenden Optimierungsprobleme bereits für eine moderate Anzahl von Hilfsvariablen nicht mehr auf handelsüblichen Computersystemen gespeichert werden können, was die Verwendung von vorimplementierten Lösungsverfahren verhindert. Die vorliegende Arbeit widmet sich daher abschließend der folgenden Problematik:

4. Wie lassen sich die betrachteten Optimierungsprobleme speicher- und recheneffizient lösen und wie sieht eine geeignete Implementierung der zugehörigen Algorithmen aus?

Unter Ausnutzung der zugrunde liegenden Tensorprodukt-Struktur werden zunächst Matrixoperationen für bestimmte Klassen von Matrizen hergeleitet, welche ohne die konkrete Speicherung dieser Matrizen durchgeführt werden können. Dies ermöglicht eine speichereffiziente Implementierung geeigneter Lösungsalgorithmen für die betrachteten Optimierungsprobleme und erlaubt so letztendlich die Verwendung multipler Hilfsvariablen in den splinebasierten Small Area-Modellen. Entscheidend für die Laufzeit der Lösungsalgorithmen ist dabei die (wiederholte) Anwendung eines matrixfreien Verfahrens der konjugierten Gradienten. Um neben der Speichereffizienz auch die Recheneffizienz der implementierten Algorithmen zu verbessern wird zusätzlich ein matrixfreies Mehrgitterverfahren als Präkonditionierer für die konjugierte Gradienten-Methode entwickelt und implementiert, dessen Potential in numerischen Tests verdeutlicht wird.

List of Figures

List of Tables

List of Algorithms

List of Symbols

For reasons of clarity and comprehensibility and to avoid ambiguous definitions, the list of symbols is restricted to symbols in global use. Symbols in local use are left out and are defined in the respective context as clearly as possible.

Survey and Small Area Framework

$\widehat{(\cdot)}$	Estimator or estimate
$\mathrm{BIAS}(\cdot)$	Bias
b, B	Running index for bootstrap samples $b = 1, \ldots, B$
C	Linkage matrix
d, D	Running index for area index $d = 1, \ldots, D$
ε	Vector of random errors
ε_i, $\varepsilon_{i,d}$	Random error for population and subpopulation unit
$\mathbb{E}(\cdot)$	Expectation
i	Population unit
$\mathrm{MSE}(\cdot)$	Mean squared error
`MSE_boot`	Bootstrap MSE-estimator
$\mu_{(\cdot)}$, $\mu_{(\cdot),d}$	Mean value of a variable for population and subpopulation
n, n_d	Size of sample and subsample
N, N_d	Size of population and subpopulation
$\mathcal{N}(\cdot, \cdot)$	Normal distribution
$p(\cdot)$	Sampling design
P	Number of covariates
π_i, $\pi_{i,j}$	First- and second-order inclusion probability
r, R	Running index for MC-replications $r = 1, \ldots, R$
$\mathcal{S}$, $\mathcal{S}_d$	Sample and subsample
$\mathbb{S}$	Set of all possible samples
$\sigma_{(\cdot)}$	Variance parameter
θ, θ_d	Arbitrary parameter of interest for population and subpopulation
$\tau_{(\cdot)}$, $\tau_{(\cdot),d}$	Total value of a variable for population and subpopulation
u	Vector of area intercepts
u_d	Area intercept
$\mathcal{U}$, $\mathcal{U}_d$	Population and subpopulation
$\mathrm{VAR}(\cdot)$	Variance
W	Area indication matrix
x_i, $x_{i,d}$	Covariate vector for population and subpopulation unit
y	Vector of the variable of interest

y_i, $y_{i,d}$	Variable of interest for population and subpopulation unit

Optimization and Multigrid Framework

A_g	Coefficient matrix on grid level g
C_{smooth}	Iteration matrix of a smoothing iteration
$C_{\text{MG},g}$	Iteration matrix of the MG method on grid level g
`CG_coarse`	CG function as coarse grid solver
e, e_j	Error vector or unit vector
g, G	Running index for grid level $g = 1, \ldots, G$
h, h_p	Mesh size
h_c	Objective function for penalty method
I_h^{2h}, I_{g+1}^{g}	Restriction matrix
I_{2h}^{h}, I_{g}^{g+1}	Prolongation matrix
`JAC`	Jacobi smoothing function
$\mathcal{LS}(\cdot)$	Least squares functional
$M(\cdot)$	Matrix valued function
ν, ν_1, ν_2	Number of smoothing iterations
$\mathcal{O}(\cdot)$	Landau-Bachmann notation
ω	Damping factor of a smoothing iteration
`v_cycle`	V-cycle function
`v_cycle_mf`	Matrix-free v-cycle function
$\mathcal{WLS}(\cdot)$	Weighted least squares functional

Spline Framework

α, α_j, α_k	Spline coefficients
D^p	Discrete penalty matrix in the p-th direction
$\Delta_{r_p}^{p}$	Difference penalty matrix in the p-th direction
$\eta_{j,q}$	Truncated power series function
Γ_r, $\Gamma_{r_p}^{p}$	Matrix for shape constraints
$I_{\leq}$, $I_{\geq}$	Index sets for shape constraints
j, J	Running index $j = 1, \ldots, J$ or multiindex $1 \leq j \leq J$
k, K	Running index $k = 1, \ldots, K := \dim(\mathcal{S}_q(\mathcal{K}))$
κ_j	Knot
$\mathcal{K}$	Knot set
λ, λ_s, λ_u	Regularization parameters
Λ	Penalty matrix
m, m_p	Number of knots
$\nu(\cdot)$	Lexicographical ordering map
Ω, Ω_p	Rectangle or interval
p ,P	Running index for spatial dimension $p = 1, \ldots, P$
$\mathcal{P}(\cdot)$	Regularization term
$\phi_{j,q}$, $\phi_{k,q}$	Arbitrary spline basis function

$\varphi_{j,q}$, $\varphi_{k,q}$	B-spline basis function
Φ, Φ_p	Spline basis matrix
Ψ_r, $\Psi^p_{r_p}$	Curvature penalty matrices
q, q_p	Spline degree
r_p	Derivative indicator
s	Spline function $s \in \mathcal{S}_q(\mathcal{K})$
$\mathcal{S}_q(\mathcal{K})$	Space of splines in one and multiple dimensions
t, T	Running index $t = 1, \ldots, T$ for discretization of Ω
$W_{r,\leq}(\alpha)$, $W_{r,\geq}(\alpha)$	Indicator matrix for penalty terms

General Notations

$\otimes$, $\bigotimes$	Kronecker product
$\odot$, $\bigodot$	Khatri-Rao product
$\times$, $\bigtimes$	Cartesian product
$(\cdot)^T$	Transposition
$(\cdot)^{-1}$	Inverse
$\overset{!}{=}$	Linear system
$\Vert \cdot \Vert$	Spectral norm
$\Vert \cdot \Vert_2$	Euclidean norm
$\Vert \cdot \Vert_A$	Induced norm
$\Vert \cdot \Vert_{L^2(\Omega)}$	Norm on $L^2(\Omega)$
$\langle \cdot, \cdot \rangle_{L^2(\Omega)}$	Scalar product in $L^2(\Omega)$
$[\cdot, \cdot]$, $[\cdot]$	Specific entries of a matrix or a vector
$\succ$, $\succeq$	Positive definite and positive semi definite
∇, ∇^2	Gradient and Hessian
$\mathbb{1}$	Indicator function
$\mathcal{C}^q(\Omega)$	Space of q-times continuously differentiable functions on Ω
$\mathrm{cond}_2(\cdot)$	Condition number of a matrix
$\mathrm{diag}(\cdot)$	Diagonal matrix or diagonal of a matrix
$\dim(\cdot)$	Dimension of a matrix or function space
$\det(\cdot)$	Determinant
$I_{(\cdot)}$	Identity matrix
$\mathrm{kern}(\cdot)$	Kernel of a mapping
$\log(\cdot)$	Logarithm
$L^2(\Omega)$	Space of square integrable functions on Ω
$\lambda_{\min}$, $\lambda_{\max}$	Smallest and largest eigenvalue
$\mathbb{N}$	Natural numbers
$\mathbb{N}_0$	Natural numbers including zero
$\mathbb{N}_0^P$	Set of multiindices
∂	Differential operator
$\mathcal{P}^q(\Omega)$	Space of polynomials of degree q on Ω
$\mathbb{R}$	Real numbers
$\mathrm{sgn}(\cdot)$	Algebraic sign

$\mathrm{supp}(\cdot)$	Support of a function
$S_{(\cdot)}$	Hat matrix
$\mathrm{tr}(\cdot)$	Trace of a matrix

List of Abbreviations

ALS	Airborne laser scanning
ANOVA	Analysis of variances
AVRRMSE	Average relative root mean squared error
BHF	Battese-Harter-Fuller
BIASMSE	Bias of an MSE-estimator
BLUE	Best linear unbiased estimator
BLUP	Best linear unbiased predictor
CG	Conjugate gradient
CI	Confidence interval
CICR	Confidence interval coverage rate
CIL	Confidence interval length
CPU	Central processing unit
CSC	Compressed sparse column
CV	Cross validation
EBLUE	Empirical best linear unbiased estimator
EBLUP	Empirical best linear unbiased predictor
GB	Gigabyte
GCV	Generalized cross validation
GHz	Gigahertz
GNFI	German National Forest Inventory
GPU	Graphics processing unit
GREG	Generalized regression
HT	Horvitz-Thompson
iid	Independent and identically distributed
ind	Independent distributed
JAC	Jacobi
LMM	Linear mixed model
MARB	Mean absolute relative bias
MB	Megabyte
MC	Monte Carlo
MCIL	Mean confidence interval length
MG	Multigrid
MGCG	Multigrid preconditioned conjugate gradient
MGCG_JAC	Multigrid preconditioned conjugate gradient with Jacobi smoother
MGCG_SSOR	Multigrid preconditioned conjugate gradient with SSOR smoother
ML	Maximum likelihood
MSE	Mean squared error

PCG	Preconditioned conjugate gradient
P-spline	Penalized spline
QP	Quadratic program
RAM	Random access memory
RBIAS	Relative bias
RBIASMSE	Relative bias of an MSE-estimator
RDISP	Relative Dispersion
REML	Restricted maximum likelihood
ResStRS	Restricted stratified random sampling
RLP	Rhineland-Palatinate
RRMSE	Relative root mean squared error
SAE	Small area estimation
SFI	State Forest Inventory
SLMM	P-spline linear mixed model
SOPT	P-spline optimization
SOPT_CON	Shape constraints P-spline optimization
SOR	Successive over-relaxation
SRS	Simple random sampling
SSOR	Symmetric successive over-relaxation
StRS	Stratified random sampling
TG	Two-grid

Chapter 1

Introduction

Sample surveys are a widely used and cost effective tool to gain information about a population under consideration. Nowadays, there is an increasing demand not only for information on the population level but also on the level of subpopulations, called areas or domains, defined geographically or by content. For some of these subpopulations of interest, however, very small subsample sizes might occur such that the application of traditional estimation methods is not straightforward. In order to provide reliable information also for those so called small areas, small area estimation (SAE) methods have to be applied. The present thesis mainly focuses on the development and the numerical implementation of small area estimation methods that are applicable to very complex types of data sets. This includes for one thing highly nonlinear and shape restricted relationships within the data and secondly the utilization of multiple auxiliary variables within the estimation process.

This chapter starts with a motivational example on small area estimation, namely the estimation of timber reserves in several forest districts of Germany, that highlights the need for the development of the aforementioned estimation methods. Afterwards, a short outline of this thesis is given and the main research contributions are presented.

1.1 Motivation: Timber Reserve Estimation in Rhineland-Palatinate

By storing immense amounts of biomass and acting as carbon dioxide sinks, forests fulfill important ecological, economical, and socio-economical functions. At the same time, however, forest ecosystems are threatened by regional impacts of global warming as well as changing socio-economic conditions (cf. Foley et al., 2005). Therefore, several national and international commitments - such as the Kyoto Protocol (cf. UNFCCC, 1998) or the Montréal Process (cf. McRoberts et al., 2004) - are dedicated to forest resources, sustainable forest management, and biodiversity in order to counter these threats. This leads to an increasing demand for very extensive and detailed information on forest resources.

In this context, the German National Forest Inventory (GNFI) is designed to provide information about forest conditions and resources at the national level on a sample basis and is supplemented by several State Forest Inventories (SFI) at the federal state level. The inventory data are collected on permanent sampling points at regular time intervals and are mainly designed for per the hectare estimation of available timber volume in cubic meters (cf. Polley et al., 2006). The selection of the trees at the sampling points is based on a probability proportional to size sampling using the tree basal area per hectare as related size measure (cf. Kangas and Maltamo, 2006). In Rhineland-Palatinate (RLP), a federal state of Germany, the SFI is organized as square sample plots, based on a regular grid, covering the forestland of RLP and was lastly conducted in 2007. Since most evaluations and plannings regarding forest resources in RLP are carried out on the forest district level, the focus has changed towards estimating forest resources, and in particular timber reserves, on a more regional level instead of the state or the national level. However, the data acquisition within the forest inventories is complex and expensive such that the number of district-specific samples is mainly too small for traditional estimation methods to provide reliable estimates on the forest district level. To overcome this issue and to provide accurate estimates, suitable auxiliary information can be incorporated into the estimation process by means of adequate statistical models. The use of auxiliary variables derived from satellite and airborne laser scanning (ALS) data for model-based and model-assisted estimation methods has gained great interest in the recent past (cf. Breidt et al., 2005 and Opsomer et al., 2007). ALS data, collected by the State Office for Surveying and Geographic Information (Landesamt für Vermessung und Geobasisinformation), are available for nearly the entire territory of RLP. They provide the mean canopy height of the forest stands in meter and lend themselves as auxiliary information to be used in the estimation process of timber reserves in RLP, due to the expectedly high correlation between the two variables (cf. Münnich et al., 2016 and Wagner et al., 2017). Small area estimation (SAE) methods (cf. Rao and Molina, 2015) provide the opportunity to combine these remote sensing auxiliary information and the sampled timber reserve information via a statistical model and thereby obtain reliable estimates at the forest district level even if the particular subsample sizes are very small and possibly nonexistent.

The use of remote sensing data for small area estimation dates back to Battese et al. (1988) and was adapted for environmental statistics by Flores and Martinez (2000), Gallego (2004), and Breidenbach and Astrup (2012), amongst others. Their utilization is based on the assumption of a (generalized) linear relationship between the remote sensing data and the variable of interest. The dependency of the timber reserves on the mean canopy height, however, is expected to be neither linear nor of monotone curvature. This can be seen in Figure 1.1, where the available timber volume per hectare in cubic meters is plotted against the mean canopy height in meter for the observations from the SFI 2007 in RLP. The displayed linear and quadratic regression functions are clearly insufficient to reflect the curvature within the data. Especially at the left margin negative timber volumes are predicted for small canopy heights, which provides inadmissible results since the available timber volume has to be nonnegative.

In order to incorporate more complex nonlinearities into the small area estimation process, Opsomer et al. (2008) and Ugarte et al. (2009) provide a very powerful and innovative approach. If the functional relationship between the variable of interest and the covariates cannot be specified a priori, the penalized spline (P-spline) method (cf. Eilers and Marx, 1996) is a popular modeling technique. By reformulating the P-spline regression model as a linear mixed model (LMM), which constitutes the foundation for common model-based small area estimation methods, the authors create a useful connection to utilize the P-spline method within the context of small area estimation. The resulting P-spline fit to the sample data of the SFI 2007 in RLP is also presented in Figure 1.1. Although the P-spline function now adequately reflects the curvature within the forest inventory data, negative and therefore inadmissible timber volumes are still predicted for small canopy heights.

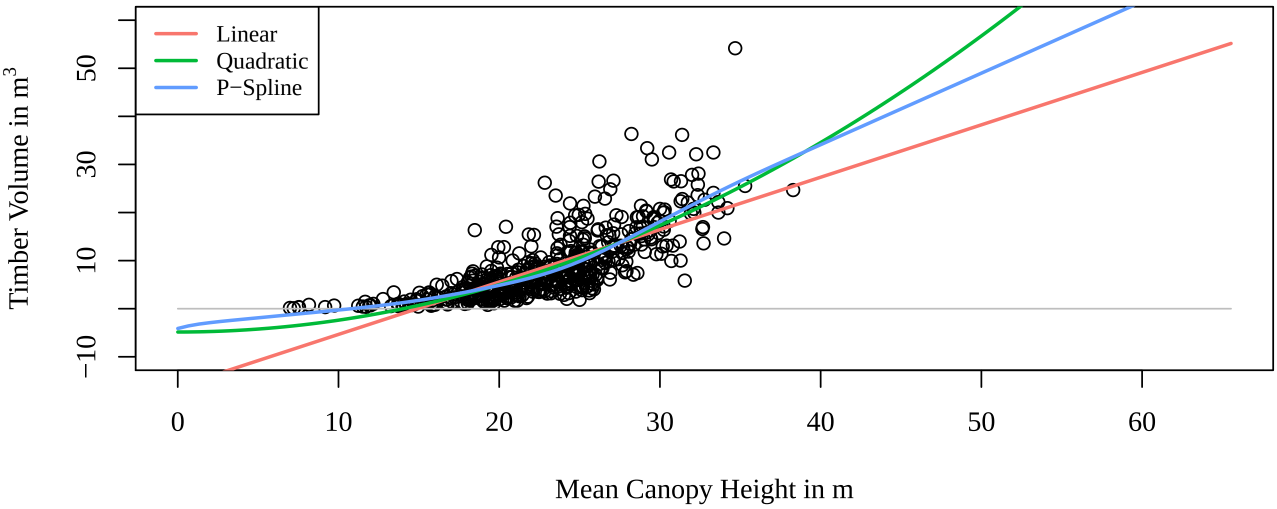

Figure 1.1: Linear, quadratic, and P-spline fit to the sample data of the SFI 2007 in RLP.

Therefore, besides the flexible modeling of complex relationships within the data, it is of practical interest to restrict the underlying model function by appropriate shape constraints. That is, the restriction of its global form to ensure a desirable and more realistic behavior of the underlying regression function in practical applications. Especially in small area estimation, where the subsample sizes are usually too small to sufficiently reflect the general trend in the underlying data, the incorporation of shape constraints may yield more appropriate, or in the first place feasible estimates (cf. Wagner et al., 2017). For example for the forest inventory data of the SFI 2007 in RLP, a monotonically increasing and nonnegative relationship between timber volume and mean canopy height seems realistic. Especially the restriction to nonnegativity is crucial since negative timber reserves cannot exist for observed timber stocks. These corresponding shape constraints, however, cannot immediately be considered within the traditional mixed model formulation such as the spline-based small area method proposed by Opsomer et al. (2008) and Ugarte et al. (2009). A first approach to incorporate at least the monotonicity property into small area estimation is given by Rueda and Lombardía (2012). At the present time, however, there exists no satisfying approach to small area estimation allowing for both a highly flexible model and arbitrary kinds of shape constraints.

In the context of small area estimation, the utilization of Big Data introduces completely new opportunities but also challenges on the applied methods (cf. Marchetti et al., 2015). Especially for remote sensing data, the Sentinel-2 mission[1] developed by the European Space Agency has marked the dawn of the Big Data era at the latest (cf. Münnich et al., 2016). The Sentinel-2 satellites, launched in 2015 and 2017, provide multi-spectral data with 13 bands and spatial resolution of 10, 20, and 60 meters with a timely repetition rate of five days, leading to an annual data stream of approximately two terabytes just for the federal state RLP. Such an amount of data offers a huge potential to significantly improve estimates, not only in the case of SAE, such that it is desirable for the considered methods to be capable of efficiently processing multiple input variables in order to utilize extensive amounts of auxiliary information. In this context, the curse of dimensionality (cf. Bellman, 1957) plays a crucial role, describing the exponential growth of required computing power and especially storage capacity

[1] https://sentinel.esa.int/web/sentinel/missions/sentinel-2 accessed on July 17, 2018.

within the number of auxiliary variables. Despite of nowadays available computing power, this issue prevents a straightforward extension of the existing spline-based small area methods to multiple dimensions. In order to summon the full potential of the multi-dimensional covariate information, as for example the Sentinel-2 mission data, the development and implementation of computational as well as memory highly efficient solution algorithms is inevitable to allow modern small area estimation methods the access into the era of Big Data.

1.2 Thesis Outline

The present thesis is divided into six chapters and organized as follows. Following this introduction, we briefly and succinctly present the mathematical background required for the general understanding of this thesis in Chapter 2. We review some basic theory on survey statistics and selected estimation methods and present linear mixed models as they provide the fundamental concept for the most common small area estimation methods. We consider special types of matrices, namely Kronecker and Khatri-Rao products, that mainly appear in the course of modeling with spline functions in multiple dimensions. The small area estimation methods developed within the scope of this monograph require the solution of large-scale linear systems and convex optimization problems on a regular basis, wherefore we provide the related theory of and adequate solution algorithms for these specific problem classes.

Chapter 3 deals with the penalized spline method as a very flexible approach to regression analysis. We review the general theory on spline functions and provide their extension to multiple input variables via a tensor product approach. We introduce the P-spline method from a theoretical as well as from a practical point of view and address some related computational aspects. By exploiting the unique basis representation of the (tensor product) splines, we determine the vector of spline coefficients $\alpha \in \mathbb{R}^K$ of the P-spline function as the solution of the optimization problem

$$\min_{\alpha \in \mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda \alpha^T \Lambda \alpha. \tag{1.1}$$

The integer K thereby denotes the dimension of the underlying spline space, $\Phi \in \mathbb{R}^{n \times K}$ is a matrix representing the basis functions evaluated at the n covariates, and $y \in \mathbb{R}^n$ gives the vector of the realizations of the variable of interest. The matrix $\Lambda \in \mathbb{R}^{K \times K}$ represents an adequate penalty matrix that measures the smoothness of the resulting P-spline whose influence is regulated by the parameter $\lambda > 0$. In order to take shape constraints into account, we illustrate their incorporation as linear inequality constraints into the optimization problem (1.1) as

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^K} \quad & \|\Phi\alpha - y\|_2^2 + \lambda \alpha^T \Lambda \alpha \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0, \ \ r \in I_\leq \\ & \Gamma_r \alpha \geq 0, \ \ r \in I_\geq. \end{aligned} \tag{1.2}$$

The sets $I_\leq$ and $I_\geq$ thereby denote adequate index sets and the matrices $\Gamma_r \in \mathbb{R}^{T \times K}$ represent the respective shape constraints with T denoting the number of utilized discretization points. Finally, we state existence and uniqueness results for the both considered optimization problems (1.1) and (1.2).

Chapter 4 is devoted to the utilization of the previously introduced shape constrained P-spline method in the context of small area estimation. We review fundamental notations and methods for small area estimation with a special emphasis on an existing spline-based small area model. This approach is based on the reformulation of the P-spline model as a linear mixed model which, however, prevents the incorporation of the required shape constraints. We therefore develop an alternative problem formulation which is, in analogy to the regression framework, based on the solution of the optimization problem

$$\min_{\alpha\in\mathbb{R}^K, u\in\mathbb{R}^D} \|\Phi\alpha + Wu - y\|_2^2 + \lambda_s \alpha^T \Lambda \alpha + \lambda_u \|u\|_2^2. \tag{1.3}$$

Besides the vector of spline coefficients $\alpha \in \mathbb{R}^K$, the vector of area-specific intercept $u \in \mathbb{R}^D$ is additionally considered within the optimization process, where D denotes the number of considered areas. The matrix $W \in \mathbb{R}^{n\times D}$ thereby expresses the link between a sample unit and the corresponding area. This optimization problem formulation then enables, again in analogy to the P-spline regression approach, the incorporation of shape constraints as linear inequality constraints into the optimization problem (1.3) as

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^K, u\in\mathbb{R}^D} \quad & \|\Phi\alpha + Wu - y\|_2^2 + \lambda_s \alpha^T \Lambda \alpha + \lambda_u \|u\|_2^2 \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0, \;\; r \in I_{\leq} \\ & \Gamma_r \alpha \geq 0, \;\; r \in I_{\geq}. \end{aligned} \tag{1.4}$$

For the both considered optimization problems (1.3) and (1.4), we state existence and uniqueness results. The practical advantage of the resulting small area estimators is highlighted by the application to the real-world example of timber volume estimation in Rhineland-Palatinate (cf. Section 1.1). In order to estimate the mean squared error of the small area estimators developed within this chapter, we propose a Monte-Carlo bootstrap method. To further investigate the performance of the small area estimators as well as its related precision estimator, we also conduct a simulation study.

The utilization of multiple covariates within the penalized spline method leads to an exponential growth of the dimension of the underlying spline space, i.e.

$$K = \mathcal{O}(2^P), \tag{1.5}$$

where P denotes the number of considered covariates and $\mathcal{O}(\cdot)$ is the Bachmann-Landau notation. This is problematical for two reasons. For one thing, the number of unknown spline coefficients that has to be determined grows rapidly within the number of covariates and secondly, the memory required to store the occurring matrices instantly exceeds the available internal memory of currently available computer systems even for moderate P. In order to make the previously developed methods also applicable to multiple covariates, the focus of Chapter 5 is on the development and implementation of computational and especially memory efficient large-scale solution algorithms for the various considered optimization problems. By exploiting the special structure of the occurring matrices, we develop adequate matrix-free solution algorithms, i.e. methods that do not explicitly require to assemble and store the matrices. These implementations finally allow to apply the proposed spline-based small area estimation methods for an arbitrary number of covariates. Furthermore, in order to drastically improve the convergence speed of the proposed matrix-free solution algorithms, we present a numerically highly efficient preconditioning method based on the multigrid idea. The potential of the developed solution algorithms is finally illustrated by several numerical test examples.

The closing Chapter 6 summarizes and concludes the thesis and points out its main findings. Additionally, we present a short outlook on future research topics and further possible applications.

1.3 Research Contributions

Within the scope of small area estimation, the utilization of nonlinear and highly flexible modeling techniques that enable a close to reality representation of the relationship between given observations is very promising. In this context, Opsomer et al. (2008) and Ugarte et al. (2009) reformulate a P-spline regression model as a linear mixed model which can directly be adapted to a small area model. While obtaining a more realistic model, the incorporation of further shape constraints is expected to significantly improve the estimates, especially in SAE, where the small sample data frequently do not reflect the general trend within the data at hand. Although the incorporation of shape constraints into the P-spline regression model is comparatively straightforward, these same constraints prevent a mixed model representation. Thus, a direct incorporation of shape constraints into spline-based small area estimation methods is not possible with the currently available approaches. In order to fill this gap in literature, the present thesis focuses on a different framework that enables both the utilization of the P-spline method in small area estimation and the incorporation of arbitrary shape constraints into the modeling process. This new approach is based on determining the model parameters as a solution of an optimization problem instead as from a LMM. The optimization framework additionally provides the advantage that no specific distributional assumptions are required in comparison to the LMM framework. This is especially reflected by the possibility of a retrospective adjustment of the smoothness of the underlying P-spline, resulting in a much more flexible approach. The optimization problem formulation further allows for the incorporation of arbitrary shape constraints as additional linear inequality constraints into the optimization problem. Thus, by allowing for both the utilization of nonlinear and highly flexible modeling techniques and the incorporation of shape constraints via an innovative optimization approach, the present thesis provides a significant contribution to the field of small area estimation.

Besides the possibility of providing a flexible and close to reality small area model via shape constrained P-splines, it is desired to extend these method to multiple covariates. By utilizing a tensor product approach, this multivariable extension is straightforward. This tensor product approach, however, causes an exponential growth of the number of model parameters with each additional covariate such that the memory complexity of the P-spline method becomes tremendously large even for a moderate number of covariates. Here, a further advantage of the optimization problem formulation is revealed, since it facilitates the application of memory and computationally efficient solution algorithms for the large-scale optimization problems. For this purpose, a matrix-free multigrid preconditioned conjugate gradient method is developed that finally enables the incorporation of arbitrary shape constraints as well as an increasing number of covariates into a very flexible spline-based small area model. Due to the development of computational efficient large-scale solution algorithms for special types of optimization problems and their memory efficient implementation, the present thesis also provides a significant contribution to the field of algorithmic optimization.

In summary, the subject of the monograph contributes to two disciplines, namely small area statistics and numerical optimization. These fields are not considered separately and the present thesis occupies a strong interdisciplinary character in the sense that the arising synergy effects influence the further development of statistical methods as well as the numerical implementations. In this way, the natural interdependency between mathematics and statistics is deepened such that this thesis additionally provides a contribution to the connection of the two disciplines.

Chapter 2

Mathematical Background

In this chapter, we provide preliminary terminologies and concepts which are fundamental for the general understanding of the present thesis and for the development of efficient solution algorithms. We aim for brief and concise information and refer to the related subject literature for further details. The topics presented in this chapter are as follows. We begin with fundamental concepts of survey statistics in Section 2.1 and introduce linear mixed models in Section 2.2, since both are fundamental for the theory of small area estimation. In Section 2.3, we consider special types of matrix products that frequently arise during this thesis while modeling with P-splines in multiple dimensions. To determine the corresponding model parameters the solution of a linear system or a convex optimization problem is required, wherefore Section 2.4 and Section 2.5, respectively, provide the related theory and adequate algorithms.

2.1 Fundamentals of Survey Statistics

The present thesis deals with the topic of small area estimation which belongs to the field of survey statistics. In the following, we therefore introduce fundamental definitions, notations, and methods of survey statistics according to Särndal et al. (1992, Chapter 2).

Notations and Definitions

We consider a fixed and finite population $\mathcal{U}$ of size $N \in \mathbb{N}$ and denote the response value of the variable of interest for each population unit $i \in \mathcal{U}$ as $y_i \in \mathbb{R}$. In many applications it is desired to determine a population parameter or a statistic

$$\theta := f(y_i : i \in \mathcal{U}), \tag{2.1}$$

which is given as a function of the variable of interest in the population. Common examples of such a statistic are the population total of the variable of interest,

$$\theta = \tau_Y := \sum_{i \in \mathcal{U}} y_i, \tag{2.2}$$

or the population mean of the variable of interest,

$$\theta = \mu_Y := \frac{1}{N} \sum_{i \in \mathcal{U}} y_i = \tau_Y / N. \tag{2.3}$$

If the y_i are known for all units $i \in \mathcal{U}$, the parameter of interest can directly be calculated. In practice, however, the variable of interest is not observed for the entire population, but only for a sample $\mathcal{S} \subset \mathcal{U}$ of size n, where $n \ll N$. Thus, the parameter θ can not be directly determined, but has to be estimated from the sampled observations. Let $\mathbb{S} := \{\mathcal{S} : \mathcal{S} \subseteq \mathcal{U}\}$ denote the set of all possible samples of $\mathcal{U}$ and let

$$p \colon \mathbb{S} \to [0,1] \tag{2.4}$$

be a function that gives the probability of selecting a specific sample. The function p is frequently called sampling design and is in the design-based framework the only stochastic element on which inference can be based (cf. Lehtonen and Veijanen, 2009, p. 219). Common sampling designs are for example simple random sampling (SRS), stratified random sampling (StRS), and cluster sampling. However, since sampling designs are not further considered within this thesis, we refer to Lohr (1999) and Cochran (2007) for further reading on the topic of sampling designs. The probability that a particular unit $i \in \mathcal{U}$ is included in a sample is known as (first-order) inclusion probability and obtained from the sampling design as

$$\pi_i := \sum_{\{\mathcal{S} \in \mathbb{S} : i \in \mathcal{S}\}} p(\mathcal{S}). \tag{2.5}$$

Analogously, the (second-order) inclusion probability that the particular units i and j are both included in the sample is defined as

$$\pi_{i,j} := \sum_{\{\mathcal{S} \in \mathbb{S} : i,j \in \mathcal{S}\}} p(\mathcal{S}). \tag{2.6}$$

These inclusion probabilities are of crucial importance for the construction of design-based estimators, which are introduced later on.

As already mentioned, the main task is to determine an estimate of the unknown population parameter θ out of the realized sample $\mathcal{S}$. An estimator of the parameter θ is typically denoted as $\widehat{\theta}$ and corresponds to a real valued random variable on the sample space $\mathbb{S}$. A particular realization $\widehat{\theta}(\mathcal{S})$ is called an estimate of the estimand θ and it is common practice to denote the estimate itself as $\widehat{\theta}$ and to use the terms estimator and estimate interchangeably. In order to describe important aspects of an estimator, adequate performance measures are needed. According to Särndal et al. (1992, Section 2.7), the expectation of an estimator $\widehat{\theta}$ is given as

$$\mathbb{E}(\widehat{\theta}) := \sum_{\mathcal{S} \in \mathbb{S}} p(\mathcal{S}) \widehat{\theta}(\mathcal{S}). \tag{2.7}$$

It is a weighted average of all possible values $\widehat{\theta}(\mathcal{S})$ weighted with the probabilities $p(\mathcal{S})$. The bias of an estimator $\widehat{\theta}$ is given as

$$\mathrm{BIAS}(\widehat{\theta}) := \mathbb{E}(\widehat{\theta}) - \theta \tag{2.8}$$

and is a measure of the average deviation of an estimated value from the parameter of interest. An estimator is referred to as unbiased if $\mathrm{BIAS}(\widehat{\theta}) = 0$, i.e. $\mathbb{E}\left(\widehat{\theta}\right) - \theta$, otherwise it is called biased. Unbiasedness, at least asymptotically, is a desirable property of an estimator but does not give any information on the dispersion of the various estimates. In order to measure this dispersion, we define the variance of an estimator $\widehat{\theta}$ as

$$\mathrm{VAR}(\widehat{\theta}) := \mathbb{E}\left(\left[\widehat{\theta} - \mathbb{E}(\widehat{\theta})\right]^2\right) = \sum_{\mathcal{S}\in\mathbb{S}} p(\mathcal{S}) \left[\widehat{\theta}(\mathcal{S}) - \mathbb{E}(\widehat{\theta})\right]^2. \tag{2.9}$$

That is, the variance of an estimator is defined as the expectation of the squared deviation of the estimator from its expected value. In order to assess the precision of an estimator, we define the mean squared error (MSE) as a combined measure of deviation and dispersion as

$$\mathrm{MSE}\left(\widehat{\theta}\right) := \mathbb{E}\left(\left[\widehat{\theta} - \theta\right]^2\right) = \mathrm{VAR}(\widehat{\theta}) + \mathrm{BIAS}(\widehat{\theta})^2. \tag{2.10}$$

If the estimator $\widehat{\theta}$ is unbiased, it follows $\mathrm{MSE}(\widehat{\theta}) = \mathrm{VAR}(\widehat{\theta})$. A small MSE of an estimator is a desirable property, since it indicates that the estimator produces estimates that are concentrated close to the true parameter θ. However, it might occur that for a particular sample the estimate is still far removed from the true parameter. Note that the introduced performance measures of an estimator are computed with respect to the specific sampling design (2.4). This is due to the fact that in this framework the only source of randomness is the sampling design, wherefore it is frequently referred to as design-based or randomization approach (cf. Lehtonen and Veijanen, 2009, p. 219).

Estimators can generally be distinguished into direct and indirect estimators. The former estimators make use of information solely from the population of interest, whereas the latter incorporate further information from outside in order to improve the estimates. The concept of incorporating information from outside the population is often referred to as borrowing strength and is of particular interest in the context of small area estimation (cf. Chapter 4). A further distinction of estimators is between design-based, model-assisted, and model-based estimators. Examples for a design-based and a model-assisted estimator are briefly present in the following, whereas model-based estimators are comprehensively discussed in Chapter 4. For reasons of clarity, in the following we restrict ourselves to the estimation of the population total (2.2) of the variable of interest. Estimates for the population mean (2.3) are obtained by dividing the respective estimates of the total by the true population size N or by the estimated population size

$$\widehat{N} := \sum_{i\in\mathcal{S}} \pi_i^{-1}. \tag{2.11}$$

The estimated population size is frequently used to compensate for sampling designs with variable sample sizes, even if the true population size is known (cf. Münnich et al., 2013, p. 154).

The Horvitz-Thompson Estimator

As already mentioned, design-based estimation of a finite population parameter refers to an estimation approach where the randomness is solely introduced by the sampling design. A further characteristic is, that no auxiliary information is used within the estimation process. One of the most popular design-based estimators of the population total is the Horvitz-Thompson

(HT) estimator, proposed by Horvitz and Thompson (1952). It is defined as

$$\widehat{\tau}_Y^{\text{HT}} := \sum_{i \in \mathcal{S}} \frac{y_i}{\pi_i}, \tag{2.12}$$

where the reciprocal first-order inclusion probabilities (2.5) are used as design weights. Since the HT-estimator exclusively uses information from the population of interest, it is declared as a direct estimator. According to Särndal et al. (1992, Section 2.8) the HT-estimator is design-unbiased with variance

$$\text{VAR}(\widehat{\tau}_Y^{\text{HT}}) = \sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{U}} \left(\frac{\pi_{i,j}}{\pi_i \pi_j} - 1 \right) y_i y_j, \tag{2.13}$$

which can be unbiasedly estimated via

$$\widehat{\text{VAR}}(\widehat{\tau}_Y^{\text{HT}}) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \left(1 - \frac{\pi_i \pi_j}{\pi_{i,j}} \right) \frac{y_i y_j}{\pi_i \pi_j}. \tag{2.14}$$

Thereby, the second-order inclusion probabilities (2.6) are required. Note that the presented formulas for the HT-estimator significantly simplify for the most common sampling designs.

The Generalized Regression Estimator

The previously introduced HT-estimator does not make explicit use of potentially available auxiliary information. Methods that incorporate such covariates into the estimation process via a model in order to reduce the variance compared to design-based estimators are called model-assisted. Note that the basic design-based properties do not depend on the validity of the model, but only the efficiency of the estimator depends on its goodness of fit. The most popular model-assisted estimator is the generalized regression (GREG) estimator, proposed by Cassel et al. (1976). Let therefore $x_i \in \mathbb{R}^P$ denote individual auxiliary vectors that are assumed to be known for each unit $i \in \mathcal{U}$, for example from a register. The GREG-estimator uses these auxiliary information from a linear regression model in order to correct the HT-estimator and is defined as

$$\widehat{\tau}_Y^{\text{GREG}} := \widehat{\tau}_Y^{\text{HT}} + \left(\tau_X - \widehat{\tau}_X^{\text{HT}} \right)^T \widehat{\beta}, \tag{2.15}$$

where

$$\widehat{\beta} := \left(\sum_{i \in \mathcal{S}} \pi_i^{-1} x_i x_i^T \right)^{-1} \sum_{i \in \mathcal{S}} \pi_i^{-1} y_i x_i \tag{2.16}$$

denotes the vector of the estimated regression coefficients from the underlying design weighted linear regression model and τ_X and $\widehat{\tau}_X^{\text{HT}}$ are defined as in (2.2) and (2.12), respectively. According to Särndal et al. (1992, Section 6.4), the GREG-estimator is asymptotically unbiased and its variance can be estimated via the residual variance estimator

$$\widehat{\text{VAR}}(\widehat{\tau}_Y^{\text{GREG}}) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \left(1 - \frac{\pi_i \pi_j}{\pi_{i,j}} \right) \frac{(y_i - x_i^T \widehat{\beta})(y_j - x_j^T \widehat{\beta})}{\pi_i \pi_j}. \tag{2.17}$$

Note that, depending on which level the regression coefficients $\widehat{\beta}$ are estimated, the GREG-estimator can be declared as direct or as indirect estimator. For more detailed information on the GREG-estimator, we refer to Särndal et al. (1992, Chapter 6).

2.2 Linear Mixed Models

In the context of model-based small area estimation, linear mixed models are a frequently used tool to exploit similar structures within a population while, at the same time, allow for differences between the particular areas. Due to its fundamentality in model-based small area estimation, we briefly introduced the general linear mixed model (LMM) in the following. It is given by

$$y = X\beta + Z\gamma + \varepsilon, \tag{2.18}$$

where:

- $y \in \mathbb{R}^n$ is a known vector of observations,
- $\beta \in \mathbb{R}^K$ is an unknown vector of fixed effects,
- $\gamma \in \mathbb{R}^D$ is an unknown vector of random effects with $\gamma \overset{\text{ind}}{\sim} \mathcal{N}(0, G)$,
- $\varepsilon \in \mathbb{R}^n$ is an unknown vector of random errors with $\varepsilon \overset{\text{ind}}{\sim} \mathcal{N}(0, R)$,
- γ and ε are independent,
- $X \in \mathbb{R}^{n\times K}$ and $Z \in \mathbb{R}^{n\times D}$ are known design matrices,
- $G \in \mathbb{R}^{D\times D}$ and $R \in \mathbb{R}^{n\times n}$ are positive definite covariance matrices.

Note that the assumption of normality of γ and ε is not claimed in general linear mixed models but is usually assumed in the context of small area estimation. If the covariance matrices G and R are known, Henderson (1950) proved that the best linear unbiased estimator (BLUE) $\widehat{\beta}^{\text{BLUE}}$ of β and the best linear unbiased predictor (BLUP) $\widehat{\gamma}^{\text{BLUP}}$ of γ exist and are given as the unique solution of the mixed model equation

$$\begin{bmatrix} X^T R^{-1} X & X^T R^{-1} Z \\ Z^T R^{-1} X & Z^T R^{-1} Z + G^{-1} \end{bmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \overset{!}{=} \begin{pmatrix} X^T y \\ Z^T y \end{pmatrix}. \tag{2.19}$$

According to Henderson (1975), the solution is given by

$$\widehat{\beta}^{\text{BLUE}} = \left(X^T V^{-1} X\right)^{-1} X^T V^{-1} y \tag{2.20}$$

and

$$\widehat{\gamma}^{\text{BLUP}} = G Z^T V^{-1} \left(y - X\widehat{\beta}^{\text{BLUE}}\right), \tag{2.21}$$

where $V := R + ZGZ^T$ denotes the variance-covariance matrix of y under the linear mixed model (2.18).

In many applications of linear mixed models, such as small area estimation, it is not only of interest to predict the random effects themselves but also to predict mixed effects of the form

$$\theta := l^T\beta + m^T\gamma \tag{2.22}$$

with known vectors $l \in \mathbb{R}^K$ and $m \in \mathbb{R}^D$. According to Jiang and Lahiri (2006), the BLUP of the mixed effect (2.22) is given as

$$\widehat{\theta}^{\text{BLUP}} = l^T\widehat{\beta}^{\text{BLUE}} + m^T\widehat{\gamma}^{\text{BLUP}} = l^T\widehat{\beta}^{\text{BLUE}} + m^T GZ^T V^{-1}\left(y - X\widehat{\beta}^{\text{BLUE}}\right), \tag{2.23}$$

using the BLUE (2.20) of β and the BLUP (2.21) of γ.

In practice, the covariance matrices G and R are unknown and have to be estimated as well. Plugging the estimates $\widehat{G}$ and $\widehat{R}$ into the equations (2.20) and (2.21), we obtain the empirical best linear unbiased estimator (EBLUE) $\widehat{\beta}^{\text{EBLUE}}$ of β and the empirical best linear unbiased predictor (EBLUP) $\widehat{\gamma}^{\text{EBLUP}}$ of γ (cf. Kackar and Harville, 1981) and thus the EBLUP of the mixed effect (2.22) as

$$\widehat{\theta}^{\text{EBLUP}} = l^T\widehat{\beta}^{\text{EBLUE}} + m^T\widehat{\gamma}^{\text{EBLUP}}. \tag{2.24}$$

In order to estimate the covariance matrices it is common to assume that G and R depend on some variance parameters $\delta \in \mathbb{R}^q$ such that $V = V(\delta)$. These variance parameters are then estimated by an adequate method, such as the maximum likelihood (ML) method. The log-likelihood under the linear mixed model (2.18) is given by

$$\ell(\delta) := c - \frac{1}{2}\left[\log(\det(V)) + (y - X\beta)^T V^{-1}(y - X\beta)\right], \tag{2.25}$$

with some constant c independent of δ. Under the assumption of normality, the partial derivative of ℓ with respect to the j-th component of δ, $j = 1, \ldots, q$, is given as (cf. Rao, 2003, Section 6.2.4)

$$\frac{\partial \ell}{\partial \delta_j}(\delta) = -\frac{1}{2}\left[\operatorname{tr}\left(V^{-1}\frac{\partial V}{\partial \delta_j}\right) + (y - X\beta)^T V^{-1}\frac{\partial V}{\partial \delta_j}V^{-1}(y - X\beta))\right]. \tag{2.26}$$

Setting all partial derivatives equal to zero and solving the resulting linear system leads to the ML-estimator of the variance components δ. Alternatives to the ML-estimator are for example the restricted maximum likelihood (REML) method or the analysis of variances (ANOVA) method. However, since variance parameter estimation is not considered within the present thesis, we refer to Harville (1977), Kackar and Harville (1984), and Searle et al. (2009) for detailed information on the topic.

2.3 Kronecker and Khatri-Rao Products

During the course of this monograph, we frequently have to face large-scale optimization problems where the occurring matrices are constructed as special matrix products, namely Kronecker products and Khatri-Rao products, of matrices of considerably lower dimensions. These structures originate from the utilization of tensor product splines. The aforementioned matrix products and their basic properties are considered in the following.

Kronecker Products

Definition 2.3.1 (Kronecker product)
For given matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{r \times s}$ the matrix

$$A \otimes B := \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mr \times ns} \tag{2.27}$$

is called Kronecker product of A and B.

The Kronecker product is extended to arbitrary many factors $A_p \in \mathbb{R}^{m_p \times n_p}$, $p = 1, \dots, P$, by

$$\bigotimes_{p=1}^{P} A_p := A_1 \otimes \dots \otimes A_P := ((\dots((A_1 \otimes A_2) \otimes A_3) \dots) \otimes A_P) \in \mathbb{R}^{m \times n}, \tag{2.28}$$

where $m := m_1 \cdot \ldots \cdot m_P$ and $n := n_1 \cdot \ldots \cdot n_P$. The following lemma reveals some important properties of the Kronecker product.

Lemma 2.3.2 (cf. Graham, 1981)
For arbitrary matrices A, B, C, D, identity matrices I and $\tilde{I}$, and a constant $\alpha \in \mathbb{R}$ the following statements about the Kronecker product hold, whenever they are well-defined:

1. *Bilinearity:* $A \otimes (B + C) = (A \otimes B) + (C \otimes D)$,
   $(A + B) \otimes C = (A \otimes C) + (B \otimes C)$,
   $\alpha(A \otimes B) = (\alpha A) \otimes B = A \otimes (\alpha B)$.
2. *Associativity:* $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
3. *Mixed product property:* $(A \otimes B)(C \otimes D) = AC \otimes BD$.
4. *Normal factor decomposition:* $A \otimes B = (A \otimes I)(\tilde{I} \otimes B)$.
5. *Distributivity of transposition:* $(A \otimes B)^T = A^T \otimes B^T$.
6. *Invertible product property: $A \otimes B$ is nonsingular if and only if both, A and B, are nonsingular. Then it holds* $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.
7. *Positive (semi-) definiteness : $A \otimes B$ is positive (semi-) definite if and only if both, A and B, are positive (semi-) definite.*
8. *Diagonal property:* $\operatorname{diag}(A \otimes B) = \operatorname{diag}(A) \otimes \operatorname{diag}(B)$.

Khatri-Rao Products

The Khatri-Rao product is closely related to the Kronecker product and is defined as columnwise Kronecker product of matrices with the same number of columns.

Definition 2.3.3 (Khatri-Rao product)
For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{r\times n}$ the matrix

$$A \odot B := \Big[A[\cdot,1] \otimes B[\cdot,1], \quad \dots, \quad A[\cdot,n] \otimes B[\cdot,n]\Big] \in \mathbb{R}^{mr\times n} \tag{2.29}$$

is called Khatri-Rao product of A and B, where $A[\cdot,i]$ and $B[\cdot,i]$ denote the i-th column of A and B, respectively.

As for the Kronecker product, the Khatri-Rao product is extended to arbitrary many factors $A_p \in \mathbb{R}^{m_p\times n}$, $p = 1,\dots,P$, by

$$\bigodot_{p=1}^{P} A_p := A_1 \odot \ldots \odot A_P := ((\ldots((A_1 \odot A_2) \odot A_3)\ldots) \odot A_P) \in \mathbb{R}^{m\times n}, \tag{2.30}$$

where $m := m_1 \cdot \ldots \cdot m_P$. By definition, some of the properties of the Kronecker product carry directly over to the Khatri-Rao product. These properties, also in relation to the Kronecker product, are given by the following lemma.

Lemma 2.3.4 (cf. Liu and Trenkler, 2008)
For arbitrary matrices A, B, C, D and a constant $\alpha \in \mathbb{R}$ the following statements hold, whenever they are well-defined:

1. *Bilinearity:* $A \odot (B + C) = (A \odot B) + (C \otimes D)$,
   $(A + B) \odot C = (A \odot C) + (B \odot C)$,
   $\alpha(A \odot B) = (\alpha A) \odot B = A \odot (\alpha B)$.
2. *Associativity:* $(A \odot B) \odot C = A \odot (B \odot C)$.
3. $(A \otimes B)(C \odot D) = AC \odot BD$.
4. $(A \otimes B)^{-1} = \left(A^T A * B^T B\right)^{-1} (A \odot B)^T$, *where $*$ denotes element-wise multiplication.*

2.4 Iterative Methods for Linear Systems

Within the course of this thesis it is frequently required to solve a linear system of equations

$$Ax \overset{!}{=} b \tag{2.31}$$

with a positive definite coefficient matrix $A \in \mathbb{R}^{n\times n}$, a right-hand side vector $b \in \mathbb{R}^n$, and a vector of unknowns $x \in \mathbb{R}^n$. These systems originate from the determination of the parameters of a P-spline. For example, as shown later on, the optimization problems (1.1) and (1.3) can be reformulated as a linear system of the above form. Especially, if multiple covariates are considered during the modeling process, the dimension n of the linear system becomes very large such that direct solvers become impracticable due to unacceptable computational and memory complexity. Thus, iterative methods have to be applied that approximately solve the linear system (2.31). Some iterative methods utilized in the present thesis are briefly introduced in this section. For further reading and detailed proofs, we refer to the monographs of Hackbusch

(1994) and Saad (2003).

2.4.1 Relaxation Methods

For a nonsingular matrix $M \in \mathbb{R}^{n\times n}$, the identity matrix $I_n \in \mathbb{R}^{n\times n}$, and an arbitrary initial guess $x_0 \in \mathbb{R}^n$ the iteration

$$x_{j+1} := x_j - M^{-1}(Ax_j - b) = (I_n - M^{-1}A)x_j + M^{-1}b \tag{2.32}$$

is called a linear stationary iterative method or a relaxation method. The matrix

$$C := I_n - M^{-1}A \tag{2.33}$$

is referred to as the iteration matrix of the relaxation method and different choices of M define several iterations. Frequently used relaxation methods are presented in the following. They are usually based on the splitting of the coefficient matrix

$$A = D + U + L, \tag{2.34}$$

where D denotes the diagonal part, U the strictly upper, and L the strictly lower triangular part of A. Note that, since A is symmetric and positive definite, all eigenvalues of A are real and positive and D is nonsingular.

Richardson Method

The (damped) Richardson iteration is defined as

$$x_{j+1} := x_j - \omega(Ax_j - b) \tag{2.35}$$

and is obtained for the choice of $M := \omega^{-1}I_n$ with some damping factor $\omega \neq 0$. The Richardson method converges for all $0 < \omega < 2/\lambda_{\max}(A)$, where $\lambda_{\max}(A)$ denotes the maximal eigenvalue of the symmetric and positive definite matrix A (cf. Hackbusch, 1994, p. 82).

Jacobi Method

Using $M := \omega^{-1}D$ with some damping factor $\omega \neq 0$ yields to the (damped) Jacobi iteration

$$x_{j+1} := x_j - \omega D^{-1}(Ax_j - b). \tag{2.36}$$

Since the Jacobi method is of great importance for the later development of memory and computationally efficient solution algorithms for the particular class of linear systems arising within this thesis (cf. Chapter 5), we present the general Jacobi procedure in Algorithm 2.1. The Jacobi method converges for all $0 < \omega < 2/\lambda_{\max}(D^{-1}A)$, where $\lambda_{\max}(D^{-1}A)$ denotes the maximal eigenvalue of the symmetric and positive definite matrix $D^{-1}A$ (cf. Hackbusch, 1994, p. 89).

Algorithm 2.1: Jacobi method

```
$j \leftarrow 0$
$r_j \leftarrow Ax_j - b$
while $\|r_j\|_2^2 > tol$ do
    $j \leftarrow j + 1$
    $x_j \leftarrow x_{j-1} - \omega D^{-1} r_{j-1}$
    $r_j \leftarrow Ax_j - b$
end
return $x_j$
```

SOR Method

The method of successive over-relaxation (SOR) is defined as

$$x_{j+1} := x_j - \omega(D + \omega L)^{-1}(Ax_j - b) \tag{2.37}$$

and is obtained for the choice of $M := \omega^{-1}D + L$ with some damping factor $\omega \neq 0$. The SOR method converges for all $0 < \omega < 2$ (cf. Hackbusch, 1994, p. 134) and for the special case of $\omega = 1$ the iteration is also called the Gauss-Seidel method.

SSOR Method

If the coefficient matrix A of the linear system (2.31) is symmetric, the same holds true for the iteration matrix of the Richardson and the Jacobi method. For the iteration matrix of the SOR method, however, this property is not fulfilled. A symmetric variant of the SOR method, the method of symmetric successive over-relaxation (SSOR), is obtained by $M := (w(2-\omega))^{-1}(D + \omega L)D^{-1}(D + \omega U)$ with some damping factor $\omega \notin \{0, 2\}$. The SSOR method then reads

$$x_{j+1} := x_j - \omega(2-\omega)(D + \omega U)^{-1}D(D + \omega L)^{-1}(Ax_j - b). \tag{2.38}$$

and converges for all $0 < \omega < 2$ (cf. Hackbusch, 1994, p. 117). As for the SOR method, for the special case of $\omega = 1$ the iteration is called symmetric Gauss-Seidel method.

2.4.2 Multigrid Method

The introduced relaxation methods are straightforward to implement and computationally inexpensive per iteration but suffer from a very poor convergence rate. Therefore, they are rarely applied explicitly as solution methods for linear systems but rather appear within much more complex algorithms. For example, within the multigrid (MG) method they are used for reasons of the so called (error) smoothing property. Multigrid techniques exploit a hierarchy of discretizations with different mesh sizes of a given problem to obtain optimal convergence from a relaxation method. The availability of a hierarchy of meshes and the corresponding linear problems yields a lot of advantages compared to methods that have only access to the coefficient matrix A and the right-hand side b of the linear system (2.31). However, the understanding of the MG method requires a detailed look at the original problem and in particular at the error frequencies associated with different meshes. In general, the hierarchy originates from the discretization of partial differential equations with refined mesh sizes. In the present thesis, however, the hierarchy originates from modeling with splines on refined knot sets. In this subsection, the general idea of the MG method is introduced based on a model problem following Hackbusch (1985) and Saad (2003, Chapter 13).

Model Problem

We start with the introduction of a model problem, which is frequently used for understanding the motivation and the theory behind the MG method. For that purpose, we consider the one-dimensional Poisson equation with homogeneous Dirichlet boundary conditions on the unit interval, that is

$$\begin{aligned} -u''(x) &= f(x) \qquad && \text{on } [0,1], \\ u(x) &= 0 && \text{in } \{0,1\}. \end{aligned} \tag{2.39}$$

For a fixed grid level $g \in \mathbb{N}$ the finite difference discretization with mesh size $h := (n+1)^{-1}$, where $n := 2^g - 1$, leads to the linear system

$$h^{-2}Ax = b, \tag{2.40}$$

where $A \in \mathbb{R}^{n \times n}$ is a tridiagonal matrix with values 2 on the main and -1 on the off diagonal,

$$x := (u(x_1), \dots, u(x_n))^T \in \mathbb{R}^n \tag{2.41}$$

is the unknown vector of function values at the discretization points $x_i := ih$, $i = 1, \dots, n$, and

$$b := (f(x_1), \dots, f(x_n))^T \in \mathbb{R}^n \tag{2.42}$$

is the known right-hand side. For further details on the construction of the model problem we refer to Hackbusch (1985, Chapter 2.1).

Smoothing Property

We now analyze the performance of the Jacobi method (2.36) applied to the linear system (2.40) representative for all relaxation methods introduced in Subsection 2.4.1. The iteration matrix of the Jacobi method is then given as

$$C_{\text{JAC}} = I_n - \omega h^2 A. \tag{2.43}$$

Note that in this special case the Jacobi method coincides with the Richardson iteration (2.35). According to Hackbusch (1985, p. 20), the eigenvectors of C_{JAC} are given as

$$v^\mu := \sqrt{2h}(\sin(\mu\pi h), \sin(2\mu\pi h), \dots, \sin(n\mu\pi h))^T \in \mathbb{R}^n \tag{2.44}$$

with the corresponding eigenvalues

$$\lambda_\mu(\omega) := 1 - 4\omega \sin^2(\mu\pi h/2),\ \ \mu = 1, \ldots, n. \tag{2.45}$$

With the choice of $\omega = 1/2$ the spectral radius $\rho(C_{\mathrm{JAC}})$, i.e. the maximum absolute value of the eigenvalues of C_{JAC}, is given as

$$\rho(C_{\mathrm{JAC}}) = 1 - 2\sin(\pi h/2) = 1 - \mathcal{O}(h^2), \tag{2.46}$$

where $\mathcal{O}(\cdot)$ denotes the common Bachmann-Landau notation. This especially shows that the convergence speed of the Jacobi method depends strongly on h and g, respectively, and becomes very slow for refining grid levels g. Note that in this case the choice of $\omega = 1/2$ is optimal in the sense that it minimizes the spectral radius of the iteration matrix. For other choices of ω the convergence rates are therefore even worse (cf. Hackbusch, 1985, p. 20).

However, the basic observation on which the MG method is founded is that convergence rate is not similar for all components of the error vector. Let therefore x^* denote the unique solution of the linear system (2.40) and let x_j denote the j-th Jacobi iterate. The error vector $e_j := x_j - x^*$ after j Jacobi iterations is then given as

$$e_j = C_{\mathrm{JAC}}(\omega)^j e_0 = \sum_{\mu=1}^{n} \lambda_\mu(\omega)^j \alpha_\mu v^\mu, \tag{2.47}$$

where the α_μ, $\mu = 1, \ldots, n$, are the coefficients of the eigenbasis representation of e_0, i.e.

$$e_0 = \sum_{\mu=1}^{n} \alpha_\mu v^\mu. \tag{2.48}$$

Again, we refer to Hackbusch (1985, p. 20) for further details. This shows that each component of the error vector is reduced by the factor $\lambda_\mu(\omega)^j$ after j Jacobi iterations, which is slow for components with small eigenvalues and fast for components with large eigenvalues. We therefore define the high frequency components or oscillatory part of the error vector as those components, where the related eigenvalue is greater or equal than one half, i.e.

$$\{\mu : \lambda_\mu(\omega) \geq 1/2\}. \tag{2.49}$$

The remaining components are referred to as low frequency components or smooth part of the error vector. For further details we refer to Saad (2003, p. 429). Loosely speaking, relaxation methods efficiently damp error components of high frequency, leaving behind the smooth part of the error vector. The slow reduction of the low frequency error components is responsible for the poor convergence rate of the relaxation methods. Since after a few iterations only the smooth part of the error vector remains, relaxation methods are frequently referred to as smoothing methods. These so called smoothing property of relaxation methods becomes vivid in Figure 2.1, where the error vector after several numbers of Jacobi iterations is presented. There, the Jacobi method with $\omega = 1/3$ is applied to the model problem (2.40) with $n = 31$, i.e. $g = 5$, $b = 0$, and initial guess

$$x_0 := (\sin(\pi h) + \sin(18\pi h), \sin(\pi 2h) + \sin(18\pi 2h), \ldots, \sin(\pi nh) + \sin(18\pi nh)) \in \mathbb{R}^n. \tag{2.50}$$

Obviously, the error vector becomes smooth after only a few iterations, but the smooth part only reduces very slowly and is still apparent after a comparatively large number of iterations.

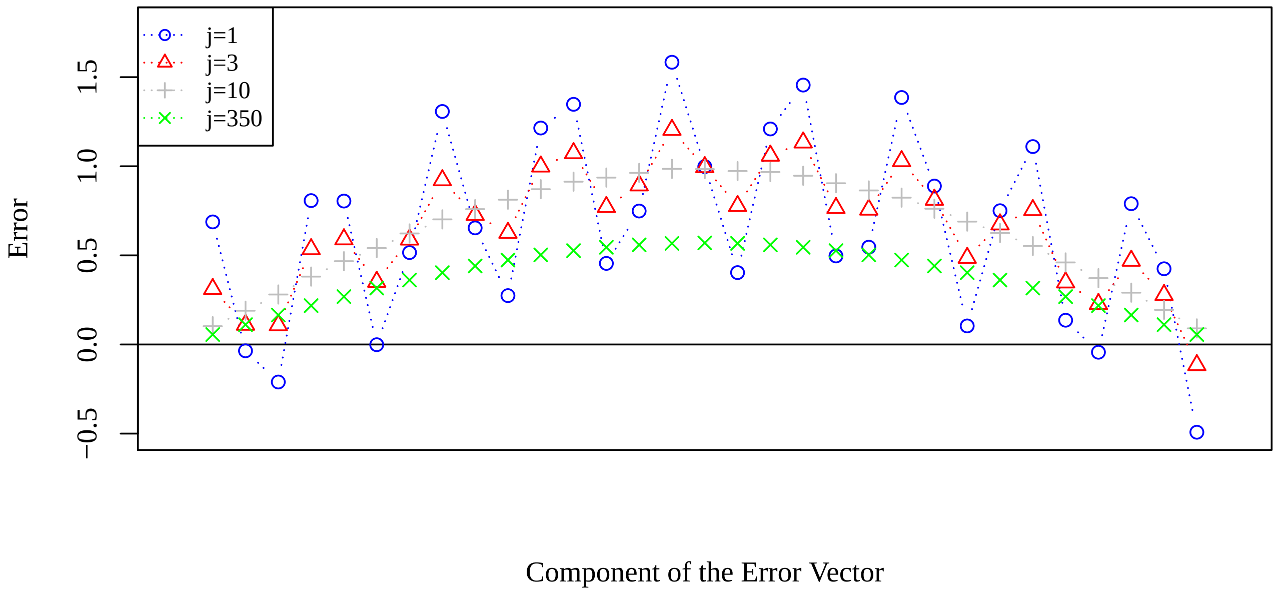

Figure 2.1: Error vector after several numbers of Jacobi iterations.

A more formal definition of the smoothing property is due to Hackbusch (1985, Definition 6.1.3).

Definition 2.4.1 (Smoothing property)
A relaxation method with iteration matrix C is said to possess the smoothing property if

$$\|AC^{\nu}\| \leq \eta(\nu)\|A\| \tag{2.51}$$

for some function η with $\eta(\nu) \to 0$ as $\nu \to \infty$. Here, $\|\cdot\|$ denotes the spectral norm, i.e.

$$\|A\| := \max_{\|x\|_2=1} \|Ax\|_2. \tag{2.52}$$

Note that all of the relaxation methods introduced in Subsection 2.4.1 satisfy the smoothing property (cf. Hackbusch, 1985, Chapter 6.2).

TG Method

As previously illustrated, relaxation methods are efficient methods for reducing the oscillatory part of the error vector, while the overall convergence is lacking with respect to the smooth components. The basic idea of the multigrid method is therefore to combine a relaxation method with a further iteration having complementary properties, i.e. provides a fast reduction of the low frequency part of the error vector. This iteration is constructed by means of the so called coarse grid correction. For that purpose, we consider a hierarchy of, in a geometrical sense, coarsening representations of the initial linear system, i.e. for a finest grid level $G \in \mathbb{N}$ we consider linear systems of the form

$$A_g x_g \overset{!}{=} b_g, \;\; g = 1, \ldots, G, \tag{2.53}$$

where the subscript g indicates for the various grid levels. The linear system (2.40) with varying grid levels g serves as an example. We aim for solving the linear system (2.53) on the finest grid level G. After a few iterations of a smoothing method with arbitrary initial guess we obtain a (very poor) approximated solution x_G whose (unknown) error

$$e_G := x_G^* - x_G \tag{2.54}$$

is large but smooth, i.e. the high frequencies of the error are removed. The linear system (2.53) on the finest grid G is equivalent to the residual equation

$$A_G e_G \overset{!}{=} r_G, \tag{2.55}$$

where $r_G := A_G x_G - b_G$ denotes the current residual. If e_G is known, the exact solution of (2.53) on the finest grid is given by $x_G^* = x_G - e_G$. The residual equation (2.55) is of the same form as the initial problem and therefore as difficult to solve. However, e_G can be much better approximated compared to x_G^* since e_G is known to bee a smooth vector. This is of key importance for the MG method since only smooth vectors can be represented well by means of a coarser grid (cf. Hackbusch, 1985, p. 21).

To approximate the error vector e_G, we approximate the residual equation (2.55) by the coarse grid residual equation

$$A_{G-1} e_{G-1} \overset{!}{=} r_{G-1}, \tag{2.56}$$

where

$$r_{G-1} := I_G^{G-1} r_G \tag{2.57}$$

is defined by means of an adequate linear and surjective map I_G^{G-1}, called restriction operator. Typical restriction operators in the context of finite element discretizations are the injection operator or the full weighting operator (cf. Saad, 2003, p. 438). Let $e_{G-1} := A_{G-1}^{-1} r_{G-1}$ denote the solution of the coarse grid residual equation (2.56), which is expected to be an approximation to the error vector e_G but on the coarser grid $G-1$. To interpolate this vector to the finer grid G, we define

$$\tilde{e}_G := I_{G-1}^G \tag{2.58}$$

by means of an adequate linear and injective map I_{G-1}^G, called prolongation operator. Here, a typical prolongation operator is a piecewise linear interpolation (cf. Saad, 2003, p. 436). In practice, it is frequently useful to chose the grid transfer operators such that

- $I_G^{G-1} = c\left(I_{G-1}^G\right)^T$ for some constant $c \neq 0$ and
- $A_{G-1} \approx I_G^{G-1} A_G I_{G-1}^G$, which is referred to as Garlerkin property.

Since $\tilde{e}_G$ is assumed to be an adequate approximation of the true error e_G, we update the actual iterate by

$$x_G^{\text{new}} := x_G - \tilde{e}_G = x_G - I_{G-1}^G A_{G-1}^{-1} I_G^{G-1}(A_G x_G - b_G). \tag{2.59}$$

The procedure (2.59) is therefore a linear iterative method, called coarse grid correction, with iteration matrix

$$C_{\text{CGC}} := I_{n_G} - I_{G-1}^G A_{G-1}^{-1} I_G^{G-1}, \tag{2.60}$$

where $n_G := \dim(A_G)$. Note that, according to Hackbusch (1985, Note 2.3.1), the coarse grid correction (2.59) itself does not converge, i.e. $\rho(C_{\text{CGC}}) > 1$.

It is the combination of smoothing iteration and coarse grid correction that leads to a very fast convergence, whereas both iterations by themselves converge only slowly or not at all (cf. Hackbusch, 1985, p. 23). Applying the coarse grid correction with ν_1 pre- and ν_2 post-smoothing iterations of an arbitrary smoothing iteration leads to the two-grid (TG) method to solve the linear system (2.53) on the finest grid G. The naming is due to the fact that the two grid levels G and $G-1$ are involved. The corresponding iteration matrix of the TG method is given by

$$C_{\text{TG}} := C_{\text{smooth}}^{\nu_2} C_{\text{CGC}} C_{\text{smooth}}^{\nu_1}, \tag{2.61}$$

where C_{smooth} denotes the iteration matrix of the utilized smoothing iteration. The efficiency of the TG method becomes obvious in Figure 2.2, where the error after one single iteration with three pre- and one post Jacobi smoothing iterations is compared to the error of the pure Jacobi iteration from Figure 2.1. More detailed information on the TG method, especially on its convergence analysis, are given by Hackbusch (1985, Chapter 6) but are not of further interest in the present thesis.

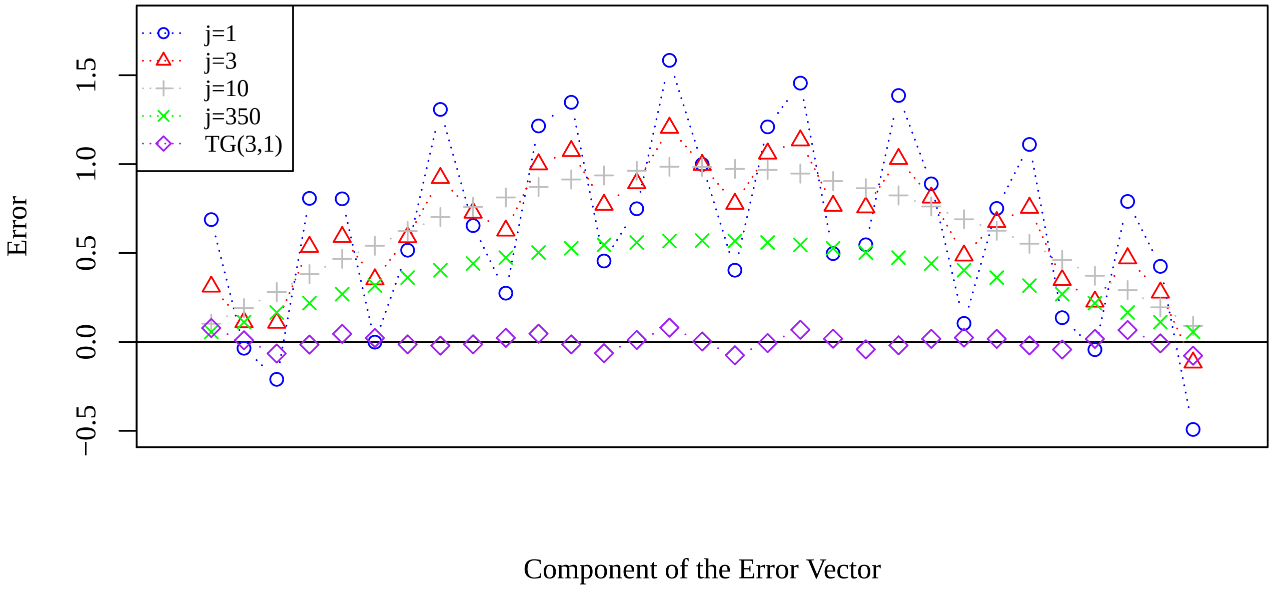

Figure 2.2: Error vector after several numbers of Jacobi iterations compared to one TG iteration with $\nu = (3, 1)$ Jacobi smoothing steps.

MG Method

The utilization of the TG method requires the solution of the coarse grid residual equation (2.56) within each iteration. Depending on the initial linear system, the dimension of the coarse grid residual equation might still be very large such that it is expensive to solve. However, the coarse grid residual equation is of the same form as the residual equation for the initial system such that the TG method can be applied, involving the grid levels $G-1$ and $G-2$. Recursive application of the TG method until the coarsest grid level $g = 1$ is reached is referred to as v-cycle and is presented in Algorithm 2.2. The name originates form the particular v-shape of the corresponding workflow.

Algorithm 2.2: `v_cycle`: Recursive application of the TG method.

```
v_cycle($A_g, b, x, g$)
    if $g = 1$ then
        $x \leftarrow A_g^{-1} b$                    // solve coarse grid residual equation (2.56)
    end
    else
        $x \leftarrow$ smooth$^{\nu_1}(A_g, b, x)$          // $\nu_1$-fold pre-smoothing
        $r \leftarrow A_g x - b$
        $r \leftarrow I_g^{g-1} r$
        $e \leftarrow$ v_cycle($A_{g-1}, r, 0, g-1$)        // recursive function call
        $x \leftarrow x - I_{g-1}^{g} e$
        $x \leftarrow$ smooth$^{\nu_2}(A_g, b, x)$          // $\nu_2$-fold post-smoothing
    end
    return $x$
end
```

Algorithm 2.2 has to be understand as follows. We define the function `v_cycle` as a function of a coefficient matrix, a right-hand side vector, the actual iterate, and the actual grid level. Starting on the finest grid $g = G$, we begin with an initial guess x for the solution of the linear system $A_G x_G \overset{!}{=} b$. First, ν_1 smoothing iterations with an adequate smoothing method are applied to this linear system, for example the Jacobi method from Algorithm 2.1, and the related residual is computed. This residual is restricted to the next coarser grid level and provides the right-hand side vector for the coarse grid residual equation on the grid $g = G-1$. To solve this coarse grid residual equation with coefficient matrix A_{G-1}, we recursively call the function `v_cycle`. However, the function is now called on a coarser grid level for the residual equation, such that the right-hand side is the restricted residual vector on the grid level $g = G-1$ and an adequate initial guess is the zero vector. The procedure is repeated until the coarsest grid level $g = 1$ is reached, where the residual equation is solved exactly. This solution is an approximation of the error vector on the coarsest grid $g = 1$ which is then successively prolongated and pre-smoothed until the finest grid level $g = G$ is reached.

The iterative application of the `v_cycle` function finally leads to the multigrid (MG) method as solver for the linear system (2.53) on the finest grid G, which is implemented in Algorithm 2.3. As for the TG method, the naming is due to the fact that the mutiple grid levels $g = 1, \dots, G$ are utilized within the solution process. As the TG method, the MG method has an interpretation as a linear iteration and the related iteration matrix is given by (cf. Saad, 2003, p. 446)

$$\begin{aligned} C_{\mathrm{MG},G} &= C_{\mathrm{smooth}}^{\nu_2}\left(I_{n_G} - I_{G-1}^{G}\left(I_{n_{G-1}} - C_{\mathrm{MG},G-1}\right)A_{G-1}^{-1}I_{G}^{G-1}A_G\right)C_{\mathrm{smooth}}^{\nu_1}, \\ C_{\mathrm{MG},1} &= 0. \end{aligned} \tag{2.62}$$

Note that the iteration matrix is only for theoretical investigations and is never assembled in practice. For example, statements on the convergence of the MG method are based on this iteration matrix but strongly depend on the underlying linear system, the grid transfer operators, and the smoothing iteration (cf. Saad, 2003, Chapter 13). A more detailed convergence analysis is beyond the scope of this thesis but it should be mentioned that the MG method is the most optimal approach in terms of required iterations in order to solve a linear system of the form (2.53). However, the implementation of such a method can be cumbersome such that it strongly depends on the application whether a MG method has to be used or not. If the linear system has to be solved multiple times, e.g. in simulation study, the best possible performance might be required, whereas for a single solution a more simple method as presented in the following can be sufficient (cf. Saad, 2003, p. 464).

Algorithm 2.3: Multigrid method

```
$j \leftarrow 0$
$r_j \leftarrow A_G x_j - b$
while $\|r_j\|_2^2 > tol$ do
    $j \leftarrow j + 1$
    $x_j \leftarrow$ v_cycle($A_G, b, x_{j-1}, G$)          // apply Algorithm 2.2
    $r_j \leftarrow A_G x_j - b$
end
return $x_j$
```

2.4.3 Conjugate Gradient Method

We now turn back to the solution of the linear system (2.31) with a positive definite coefficient matrix $A \in \mathbb{R}^{n \times n}$.

CG Method

The conjugate gradient (CG) method, introduced by Hestenes and Stiefel (1952), is the most popular algorithms to solve a linear system with a symmetric and positive definite coefficient matrix. The general procedure of the CG method is presented in Algorithm 2.4. Basically, the CG method is a realization of an orthogonal projection technique onto the Krylov subspace

$$\mathcal{K}_j(A, r_0) := x_0 + \mathrm{span}\{r_0, Ar_0, \ldots, A^j r_0\}, \tag{2.63}$$

where $r_0 := Ax_0 - b$ is the initial residual to the arbitrary initial guess x_0 and j is the actual iteration number (cf. Saad, 2003, p. 196). Note that typical initial guesses are $x_0 = 0$ or $x_0 = b$. As a Krylov subspace method the CG iteration computes the exact solution $x^* := A^{-1}b$ in at most n iterations, exact arithmetic provided. However, since n might be very large in practice, The CG method is generally used as an iterative method. The computational complexity mainly depends on the condition number of the coefficient matrix, that is

$$\operatorname{cond}_2(A) := \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \geq 1, \tag{2.64}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the largest and the smallest eigenvalue of the symmteric and positive definite matrix A, respectively. According to Saad (2003, p. 215), it holds

$$\|x_j - x^*\|_A \leq 2 \left(\frac{\sqrt{\operatorname{cond}_2(A)} - 1}{\sqrt{\operatorname{cond}_2(A)} + 1} \right)^j \|x_0 - x^*\|_A \tag{2.65}$$

for the j-th iteration of the CG method, where $\|\cdot\|_A$ denotes the norm induced by A defined by $\|x\|_A^2 := x^T A x$ for all $x \in \mathbb{R}^n$. A condition number close to one is therefore desirable and with an increasing condition number the computational complexity of the CG method is expected to deteriorate.

Algorithm 2.4: Conjugate gradient method

```
$j \leftarrow 0$
$p_j \leftarrow r_j \leftarrow b - Ax_j$
while $\|r_j\|_2^2 > tol$ do
    $v_j \leftarrow Ap_j$
    $w_j \leftarrow \|r_j\|_2^2 / p_j^T v_j$
    $x_{j+1} \leftarrow x_j + w_j p_j$
    $r_{j+1} \leftarrow r_j - w_j v_j$
    $p_{j+1} \leftarrow r_{j+1} + (\|r_{j+1}\|_2^2 / \|r_j\|_2^2) p_j$
    $j \leftarrow j + 1$
end
return $x_j$
```

PCG Method

If the coefficient matrix A of the linear system (2.31) is of poor condition, the convergence of the CG method is expected to be very slow. In order to improve the computational complexity of the CG method, preconditioning methods are frequently used in practice. The main idea is to reformulate the linear system by the use of a preconditioner $P \in \mathbb{R}^{n \times n}$ as

$$\begin{aligned} Ax &\overset{!}{=} b \\ \Leftrightarrow \quad PAx &\overset{!}{=} Pb \end{aligned} \tag{2.66}$$

in such a way that $\operatorname{cond}_2(PA) \ll \operatorname{cond}_2(A)$. Then, the CG method is applied to the preconditioned linear system (2.66). The preconditioner P has to be symmetric and positive definite in order to maintain the requirements of the CG iteration. The procedure of the preconditioned conjugate gradient (PCG) method is presented in Algorithm 2.5. Widely used preconditioning methods are incomplete factorizations, like the incomplete upper-lower (ILU) or the incomplete Cholesky factorization. A review is given by Saad (2003, Chapter 9).

Algorithm 2.5: Preconditioned conjugate gradient method

```
$j \leftarrow 0$
$r_j \leftarrow b - Ax_j$
$p_j \leftarrow z_j \leftarrow Pr_j$
while $\|r_j\|_2^2 > tol$ do
    $v_j \leftarrow Ap_j$
    $w_j \leftarrow r_j^T z_j / p_j^T v_j$
    $x_{j+1} \leftarrow x_j + w_j p_j$
    $r_{j+1} \leftarrow r_j - w_j v_j$
    $z_{j+1} \leftarrow Pr_{j+1}$
    $p_{j+1} \leftarrow z_{j+1} + (r_{j+1}^T z_{j+1} / r_j^T z_j) p_j$
    $j \leftarrow j + 1$
end
return $x_j$
```

MGCG Method

If the linear system to solve is of the hierarchical form (2.53) the MG method from Algorithm 2.3 can be applied to solve the linear system on the finest grid G. As already mentioned, the MG method possesses an interpretation as a relaxation method with iteration matrix $C_{\mathrm{MG},G}$, where the iteration matrix represents one v-cycle, i.e. the application Algorithm 2.2. Since $C_{\mathrm{MG},G}$ is symmetric and positive definite, provided the iteration matrix of the applied smoothing iteration is symmetric, the v-cycle is applicable as preconditioning method for the CG iteration, that is $P = C_{\mathrm{MG},G}$. In practice, this procedure is frequently much more efficient then the direct application of the MG method as a solver. The application of one v-cycle as preconditioner per CG iteration leads to the multigrid preconditioned conjugate gradient (MGCG) method, presented in Algorithm 2.6.

Algorithm 2.6: Multigrid preconditioned conjugate gradient method

```
$j \leftarrow 0$
$r_j \leftarrow b - Ax_j$
$p_j \leftarrow z_j \leftarrow$ v_cycle($A_G, r_j, 0, G$)                // apply Algorithm 2.2
while $\|r_j\|_2^2 > tol$ do
    $v_j \leftarrow Ap_j$
    $w_j \leftarrow r_j^T z_j / p_j^T v_j$
    $x_{j+1} \leftarrow x_j + w_j p_j$
    $r_{j+1} \leftarrow r_j - w_j v_j$
    $z_{j+1} \leftarrow$ v_cycle($A_G, r_{j+1}, 0, G$)           // apply Algorithm 2.2
    $p_{j+1} \leftarrow z_{j+1} + (r_{j+1}^T z_{j+1} / r_j^T z_j) p_j$
    $j \leftarrow j + 1$
end
return $x_j$
```

2.5 Fundamentals of Convex Optimization

Besides of linear systems, we frequently have to solve convex optimization problems within the course of this thesis. These convex problems basically originate from the incorporation of shape constraints into the penalized spline models. For example, as shown later on, the optimization problems (1.2) and (1.4) are convex optimization problems, or more precise convex quadratic programs. In the following, we briefly present basic theory and solution algorithms for convex optimization problems. For further information on the topic and detailed proofs, we refer to Nocedal and Wright (2006).

Convex Optimization Problem

An optimization problem of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0,\ i = 1, \dots, m, \end{aligned} \tag{2.67}$$

with convex functions $f, g_1, \dots, g_m \colon \mathbb{R}^n \to \mathbb{R}$ is called a convex optimization problem. We refer to the set

$$\mathcal{F} := \{x \in \mathbb{R}^n : g_i(x) \leq 0,\ i = 1, \dots, m\} \tag{2.68}$$

as the feasible set and call the convex optimization problem feasible if $\mathcal{F} \neq \emptyset$. If $\mathcal{F} = \mathbb{R}^n$, the optimization problem (2.67) is referred to as unconstrained, otherwise as constrained. The value

$$f^* := \inf_{x\in\mathcal{F}} f(x) \tag{2.69}$$

denotes the optimal value and we call a vector $x^* \in \mathbb{R}^n$ an optimal solution of the convex optimization problem if $x^* \in \mathcal{F}$ and $f(x^*) = f^*$. With X^*, we denote the set of all optimal solutions. Convex optimization problems possess important properties compared to common constrained optimization problems, which are stated by the following remark.

Remark 2.5.1 (cf. Nocedal and Wright, 2006, pp. 15-17)
For the convex optimization problem (2.67) it holds:

1. *Each local minimum is already a global minimum.*
2. *If the objective function f is strictly convex and if $X^* \neq \emptyset$, then the optimal solution is unique.*
3. *If f is continuously differentiable and if $\mathcal{F} = \mathbb{R}^n$, then $X^* = \{x \in \mathbb{R}^n : \nabla f(x) = 0\}$, where ∇f denotes the gradient of f.*

Penalty Method

A variety of methods for constrained optimization exists, depending on the specific form of the objective function and the respective constraints. An important class of methods finds the solution to the original constrained problem by replacing it by a sequence of unconstrained subproblems, which can be solved with unconstrained optimization algorithms (cf. Nocedal and Wright, 2006, p. 491). In the quadratic penalty approach the constrained optimization problem (2.67) is replaced by

$$\min_{x \in \mathbb{R}^n} h_c(x), \tag{2.70}$$

where the quadratic penalty function

$$h_c(x) := f(x) + \frac{c}{2} \sum_{i=1}^{m} \max\{0, g_i(x)\}^2 \tag{2.71}$$

consist of the original objective function and an additional penalty term which is positive if a constraint is violated and zero otherwise. The penalty parameter $c \geq 0$ controls for the influence of the penalty term. By driving $c \to \infty$ constraint violations are penalized with increasing severity, forcing the minimizer of the penalty function (2.70) closer to the feasible set of the initial constrained problem (2.67). The advantage is, that the penalty problem is now an unconstrained problem which can be solved using techniques from unconstrained optimization. The general procedure of the quadratic penalty method is presented in Algorithm 2.7.

Algorithm 2.7: Penalty method

```
$x^{\text{new}} \leftarrow \min_{x \in \mathbb{R}^n} f(x)$                // solve unconstrained problem
while stopping criterion not reached do
    $x^{\text{old}} \leftarrow x^{\text{new}}$
    $x^{\text{new}} \leftarrow \operatorname{argmin}_{x \in \mathbb{R}^n} h_c(x)$    // solve penalty problem with $x^{\text{old}}$ as initial guess
    $c \leftarrow \eta c$                                     // increase penalty parameter
end
return $x^{\text{new}}$
```

It is common practice to start with a solution of the unconstrained optimization problem as first initial guess for the constrained problem. Then, the penalty parameter is successively increased and the current iterate is used as initial guess for the penalty problem with increased penalty parameter. This procedure is of practical relevance since the Hessian of the penalty function can become ill conditioned for large penalty parameters such that the applied solution methods for the penalty problem might perform poorly. This difficulties can be overcome by the above mentioned choice of the starting points Nocedal and Wright (cf. 2006, p. 495). The convergence of the quadratic penalty method is finally stated by the following theorem and for further information on penalty methods we refer to (Nocedal and Wright, 2006, Chapter 17).

Theorem 2.5.2 (cf. Nocedal and Wright, 2006, Theorem 17.1)
Each limit point of the sequence produced by Algorithm 2.7 provides a solution of the initial optimization problem (2.67).

Newton Method

Applying the quadratic penalty method to the convex optimization problem (2.67) leads to the repetitive solution of an unconstrained convex problem (2.70). If the objective function of the unconstrained penalty problem is twice continuously differentiable, the Newton method is applicable. For a more convenient notation, we consider the general optimization problem

$$\min_{x \in \mathbb{R}^n} h(x) \tag{2.72}$$

with a twice continuously differentiable and strictly convex objective function. A line search strategy to solve this problem is an iterative method of the form

$$x_{j+1} := x_j + \alpha_j d_j, \tag{2.73}$$

with an initial guess $x_0 \in \mathbb{R}^n$, a search direction $d_j \in \mathbb{R}^n$, and a step length $\alpha_j \in \mathbb{R}$. The search direction

$$d_j := -\nabla^2 h(x_j)^{-1} \nabla h(x_j) \tag{2.74}$$

defines the Newton method which is presented in Algorithm 2.8. The step size can be determined within a backtracking line search by means of several conditions (cf. Nocedal and Wright, 2006, Section 3.1) but this issue is not further considered within this thesis.

Algorithm 2.8: Newton method

```
$j \leftarrow 0$
while $\|\nabla h(x_j)\|_2^2 > tol$ do
    $d_j \leftarrow -\nabla^2 h(x_j)^{-1} \nabla h(x_j)$        // solve linear system
    compute $\alpha_j$                                          // backtracking line search
    $x_{j+1} \leftarrow x_j + \alpha_j d_j$
    $j \leftarrow j + 1$
end
return $x_j$
```

Due to the line search strategy, the Newton method is globally convergent and if the actual iterate is sufficiently close to the true solution, the rate of convergence is quadratic. For further information and practical modifications of the Newton method we refer to Nocedal and Wright (2006, Chapter 6).

Convex Quadratic Program

A special type of a convex optimization problem is a convex quadratic program (QP), which is an optimization problem of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^T A x - a^T x \\ \text{s.t.} \quad & Cx \leq c \end{aligned} \tag{2.75}$$

with a symmetric positive semidefinite matrix $A \in \mathbb{R}^{n\times n}$, an arbitrary matrix $C \in \mathbb{R}^{m\times n}$, and arbitrary vectors $a \in \mathbb{R}^n$ and $c \in \mathbb{R}^m$. With

$$\begin{aligned} &f\colon \mathbb{R}^n \to \mathbb{R},\ x \mapsto \frac{1}{2}x^T A x - a^T x,\\ &g_i\colon \mathbb{R}^n \to \mathbb{R},\ x \mapsto C[i,\cdot]x - c_i,\ i = 1,\dots,m. \end{aligned} \tag{2.76}$$

the problem (2.75) is indeed a convex optimization problem, since

$$\nabla f(x) = Ax - a \quad \text{and} \quad \nabla^2 f(x) = A \succeq 0, \tag{2.77}$$

such that f is convex. Note that f is strictly convex if and only if $A \succ 0$. Remark 2.5.1 for convex optimization problems can therefore directly be adapted to a convex QP leading to the following remark.

Remark 2.5.3
The following statements hold for the convex QP (2.75):

1. *If $A \succ 0$ and if $X^* \neq \emptyset$, then the optimal solution is unique.*
2. *If $\mathcal{F} = \mathbb{R}^n$, then $X^* = \{x \in \mathbb{R}^n : Ax = a\} \neq \emptyset$.*

In the following theorem, we state an equivalent formulation for the existence of a solution of a convex QP.

Theorem 2.5.4 (cf. Bertsekas et al., 2003, p. 101)
For a convex QP it holds

$$X^* \neq \emptyset \ \Leftrightarrow\ -\infty < f^* < \infty, \tag{2.78}$$

where X^* denotes the set of all optimal solutions and f^* the related optimal value as defined in (2.69).

Common approaches to solve quadratic programs are for example active-set and interior-point methods. However, to the size of the QPs considered in this thesis, these methods are no longer applicable and are therefore not further discussed. Instead, we refer to Nocedal and Wright (2006, Chapter 16).

Chapter 3

Penalized Splines in Regression Analysis

Regression analysis is a branch of statistics that mainly aims for examining and modeling the relationship between different variables of a data set. For that purpose, a regression function is fitted to the given observations, where it is basically distinguished between parametric and nonparametric methods. In a parametric model, the form of the regression function is predetermined and fully described by a fixed number of parameters that have to be estimated from the data. By contrast, a nonparametric model determines the regression function from an undetermined and possibly infinite set of parameters that depends on the data. Parametric models are much easier to determine, understand, and interpret and provide preferable statistical properties compared to the nonparametric models. For more complex observations, however, the parametric approach is not flexible enough to capture the fine-scaled structures within the data at hand (cf. Eubank, 1988, pp. 3-5).

The method of penalized splines (P-splines) provides a bridge between the classical parametric and nonparametric approaches since it combines the fixed number of parameters of parametric regression splines with the flexibility of nonparametric smoothing splines. Penalized splines trace back to Silverman (1985) and O'Sullivan (1986) and have become popular for statistical applications due to Eilers and Marx (1996). A comprehensive insight into regression modeling is given by Fahrmeir et al. (2013) and for further reading on the penalized spline method itself, we refer, amongst others, to the monographs of Eubank (1988), Wahba (1990), Green and Silverman (1993), and Ruppert et al. (2003). The main idea of the P-spline method is summarized as follows:

1. Express the regression function as a spline function with a generous number of knots such that the spline is flexible enough to represent even highly complex data structures.
2. Introduce an additional roughness penalty term that prevents overfitting and minimize the regularized least squares criterion instead of the ordinary one.

Although the P-spline method does not impose any restrictions on the particular form of the regression function, it frequently occurs in practice that there exists restrictions on its global shape, such as monotonicity or convexity. The incorporation of selected shape constraints into the P-spline method has recently been achieved under certain limitations (cf. Bollaerts et al., 2006; Meyer, 2008, 2012). A general framework for shape constrained P-splines, however, has not been considered up to the present time. In this chapter, we therefore provide the required theory in order to incorporate shape constraints into the P-spline method.

Since spline functions provide the foundation for the penalized spline method, Section 3.1 is devoted to splines in one and multiple input variables. In Section 3.2, we introduce the P-spline method in the regression context and consider some related computational aspects. In order to obtain more realistic regression functions, we address the concept of shape constrained P-splines in Section 3.3.

3.1 Spline Functions

Spline functions are basically defined as piecewise polynomials with a global degree of smoothness and have become a popular tool in mathematical and statistical modeling. The fundamental theory is briefly introduced in the following an for further reading and detailed proofs, we refer to the textbooks of de Boor (1978), Schumaker (1981), and Dierckx (1993).

3.1.1 Splines in One Variable

Basic Definitions

Let $\Omega := [a, b] \subset \mathbb{R}$ be a finite and closed interval. For $q \in \mathbb{N}_0$, let $\mathcal{C}^q(\Omega)$ denote the space of q-times continuously differentiable functions on Ω and let $\mathcal{P}_q(\Omega)$ denote the space of polynomials of degree q on Ω. For $m \in \mathbb{N}_0$, let

$$\mathcal{K} := \{a = \kappa_0 < \ldots < \kappa_{m+1} = b\} \tag{3.1}$$

be a partition of Ω into the $m + 1$ subintervals $[\kappa_{j-1}, \kappa_j]$, $j = 1, \ldots, m + 1$.

Definition 3.1.1 (Spline)
A function $s \in \mathcal{C}^{q-1}(\Omega)$ is called a spline of degree $q \in \mathbb{N}_0$ with knots $\mathcal{K}$ if

$$s|_{[\kappa_{j-1},\kappa_j]} \in \mathcal{P}_q([\kappa_{j-1}, \kappa_j]) \quad \textit{for all} \;\; j = 1, \ldots, m + 1. \tag{3.2}$$

The space of splines of degree q with knots $\mathcal{K}$ is denoted as $\mathcal{S}_q(\mathcal{K})$.

If the knots are equally spaced, that is there exists a mesh size $h > 0$ such that

$$\kappa_{j+1} - \kappa_j = h \;\; \text{for all} \;\; j = 1, \ldots, m, \tag{3.3}$$

a spline is referred to as a uniform spline. The utilization of uniform splines frequently leads to significant simplifications in computational implementations, which can be seen later on. According to Schumaker (1981, Theorem 4.4), the spline space $\mathcal{S}_q(\mathcal{K})$ is a finite dimensional linear space of dimension

$$J := \dim(\mathcal{S}_q(\mathcal{K})) = m + q + 1 \tag{3.4}$$

and with $\{\phi_{1,q}, \ldots, \phi_{J,q}\}$ we denote an arbitrary basis of $\mathcal{S}_q(\mathcal{K})$. Thus, every spline function $s \in \mathcal{S}_q(\mathcal{K})$ possesses a unique expression as a linear combination of the basis functions, that is

$$s = \sum_{j=1}^{J} \alpha_j \phi_{j,q}, \tag{3.5}$$

and we call the numbers $\alpha_j \in \mathbb{R}$ the spline coefficients. The most basic example for such a basis is the so called truncated power series basis (cf. Schumaker, 1981, pp. 110-112), defined by the functions

$$\eta_{j,q}(x) := \begin{cases} x^{j-1}, & \text{if } j \leq q+1 \\ \max\{0, (x - \tau_{j-(q+1)})^q\}, & \text{else} \end{cases}, \ j = 1, \dots, J. \tag{3.6}$$

This basis, however, is not well suited for numerical applications (cf. Schumaker, 1981, p. 112). For one thing, the truncated power series basis functions (3.6) grow rapidly and unbounded which results in numerical precision problems when Ω is a large interval. Secondly, many basis functions are nonzero when evaluated at some value at the right boundary of Ω which results in dense design matrices. This weakness can be addressed by using the so called B-spline basis, introduced in the following.

B-Splines

In order to define the B-spline basis, let $2q$ additional knots outside of the interval Ω be given, i.e.

$$\kappa_{-q} \leq \dots \leq \kappa_{-1} \quad \text{and} \quad \kappa_{m+2} \leq \dots \leq \kappa_{m+q+1}. \tag{3.7}$$

We refer to these knots as outer knots, whereas the already existing knots from (3.1) are referred to as inner knots. For equally spaced knots the definition of the outer knots is according to the inner knots, that is

$$\kappa_j := a + jh, \ j = -q, \dots, m+q+1, \tag{3.8}$$

where $h = (b-a)/(m+1)$ is the related mesh size. For unequally distributed knots, however, different knot placing strategies exist (cf. Fahrmeir et al., 2013, p. 429), for example

$$\kappa_{-q} = \dots = \kappa_{-1} = a \quad \text{and} \quad \kappa_{m+2} = \dots = \kappa_{m+q+1} = b. \tag{3.9}$$

Based on these outer knots, B-splines can be recursively defined according to Cox (1972) and de Boor (1972).

Definition 3.1.2 (B-spline)
For $j = 1, \dots, J$, we refer to point-wise defined function

$$\varphi_{j,q}(x) := \begin{cases} \left(\frac{x - \kappa_{j-(q+1)}}{\kappa_{j-1} - \kappa_{j-(q+1)}}\right) \varphi_{j,q-1}(x) + \left(\frac{\kappa_{j-q} - x}{\kappa_{j-q} - \kappa_j}\right) \varphi_{j+1,q-1}(x), & \textit{if } q > 0 \\ \mathbb{1}_{[\kappa_{j-1}, \kappa_j[}(x), & \textit{if } q = 0 \end{cases} \tag{3.10}$$

as the j-th B-spline, where $\mathbb{1}$ denotes the indicator function.

According to Schumaker (1981, pp. 116-117), every B-spline is indeed a spline function, i.e. $\varphi_{j,q} \in \mathcal{S}_q(\mathcal{K})$, and the set of B-splines

$$\{\varphi_{j,q} \;:\; j = 1, \ldots, J\} \tag{3.11}$$

forms a basis of the spline space $\mathcal{S}_q(\mathcal{K})$. As an example, the uniform B-spline bases of various degrees on the unit interval $\Omega = [0, 1]$ with $m = 6$ equally spaced knots are presented in Figure 3.1.

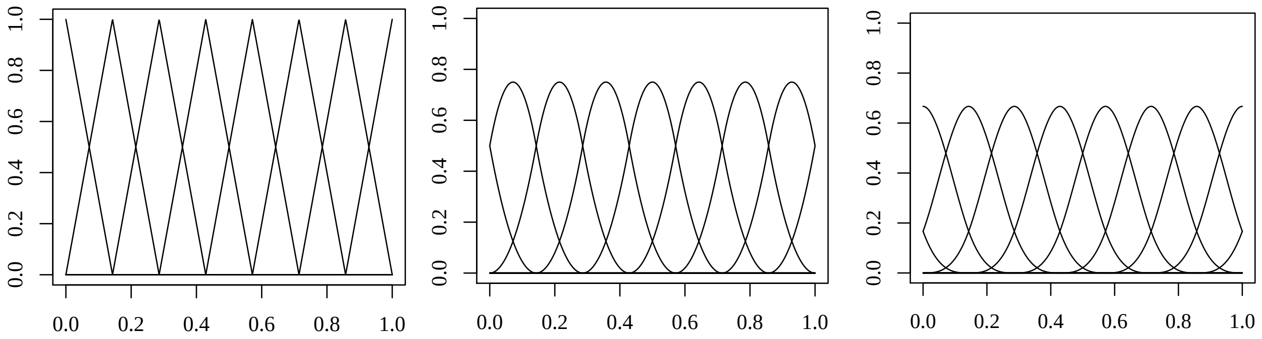

Figure 3.1: Uniform B-spline basis of degree $q = 1, 2, 3$ (from left to right) on $\Omega = [0, 1]$ with $m = 6$ equally spaced knots.

Important characteristics of B-splines which make them well suited for numerical applications are (cf. Dierckx, 1993, pp. 8-11):

1. Normalization: $\varphi_{j,q}(x) \in [0, 1]$ for all $x \in \Omega$.
2. Local support: $\operatorname{supp}(\varphi_{j,q}) = \left[\kappa_{j-(q+1)}, \kappa_j\right]$.
3. Partition of unity: $\sum\limits_{j=1}^{J} \varphi_{j,q}(x) = 1$ for all $x \in \Omega$.
4. Derivatives: $\partial\varphi_{j,q} = q\left(\dfrac{\varphi_{j,q-1}}{\kappa_{j-1} - \kappa_{j-(q+1)}} + \dfrac{\varphi_{j+1,q-1}}{\kappa_{j-q} - \kappa_j}\right)$.
5. Integrals: $\int\limits_{\kappa_{j-(q+1)}}^{\kappa_j} \varphi_{j,q} = \dfrac{\kappa_j - \kappa_{j-(q+1)}}{q+1}$.

Note that the evaluation of B-splines and the related computation of derivatives and integrals simplify for uniform B-splines.

Uniform B-Spline Refinement

According to Höllig (2003, p. 32), numerical approximations with (uniform) spline functions usually involve a sequence of refined mesh sizes in order to evaluate the accuracy of the approximations. Mesh refinements are also necessary in a neighborhood of singularities and to resolve small details of solutions. For such purposes the following subdivision formula for uniform B-splines turns out to be very useful in the present thesis. More precise, in the course of this thesis it is required for the development and implementation of a multigrid method for linear systems originating from a discretization with uniform B-spline functions. For a more convenient notation let $\mathcal{K}^{2h}$ be an equally spaced knot set with mesh size $2h$ and let $\mathcal{K}^h$ be an equally spaced knot set with mesh size h, obtained by dividing each subinterval of $\mathcal{K}^{2h}$ into half. Further, let

$$\{\varphi_{j,q}^{2h} \;:\; j = 1, \dots, J^{2h} := m + q + 1\} \quad \text{and} \quad \{\varphi_{j,q}^{h} \;:\; j = 1, \dots, J^{h} := 2m + q + 2\} \tag{3.12}$$

denote the B-spline bases of $\mathcal{S}_q(\mathcal{K}^{2h})$ and $\mathcal{S}_q(\mathcal{K}^h)$, respectively.

Theorem 3.1.3 (cf. Höllig, 2003, p. 32)
For all B-splines $\varphi_{j,q}^{2h} \in \mathcal{S}_q(\mathcal{K}^{2h})$ it holds

$$\varphi_{j,q}^{2h} = \sum_{i=0}^{q+1} c_i \varphi_{2j-(q+1)+i,q}^{h}, \tag{3.13}$$

where

$$c_i := \frac{1}{2^q} \binom{q+1}{i} \tag{3.14}$$

denotes a weighted binomial coefficient.

For a uniform spline $s \in \mathcal{S}_q(\mathcal{K}^{2h}) \subset \mathcal{S}_q(\mathcal{K}^h)$ it holds

$$s = \sum_{j=1}^{J^{2h}} \alpha_j^{2h} \varphi_{j,q}^{2h} \quad \text{and} \quad s = \sum_{j=1}^{J^{h}} \alpha_j^{h} \varphi_{j,q}^{h} \tag{3.15}$$

and Theorem 3.1.3 now provides the link between the B-spline basis functions and the related B-spline coefficients for the different mesh sizes. This is stated by the following lemma. For a more convenient notation, let

$$\varphi^h := \left(\varphi_{1,q}^h, \dots, \varphi_{J^h,q}^h\right)^T \quad \text{and} \quad \varphi^{2h} := \left(\varphi_{1,q}^{2h}, \dots, \varphi_{J^{2h},q}^{2h}\right)^T \tag{3.16}$$

denote the vectors of B-spline basis functions and let

$$\alpha^h := \left(\alpha_1^h, \dots, \alpha_{J^h}^h\right)^T \quad \text{and} \quad \alpha^{2h} := \left(\alpha_1^{2h}, \dots, \alpha_{J^{2h}}^{2h}\right)^T \tag{3.17}$$

denote the vectors of the related B-spline coefficients.

Lemma 3.1.4
Let the matrix $I_{2h}^h \in \mathbb{R}^{J^h \times J^{2h}}$ be element-wise defined as

$$I_{2h}^h[i,j] := \frac{1}{2^q} \binom{q+1}{i - 2j + q + 1} = c_{i-2j+q+1}, \tag{3.18}$$

where $c_{i-2j+q+1}$ is defined as in (3.14). Then it holds $\alpha^h = I_{2h}^h \alpha^{2h}$ and $\varphi^{2h} = \left(I_{2h}^h\right)^T \varphi^h$.

Proof. Due to Theorem 3.1.3, it holds

$$\begin{aligned} s &= \left(\varphi^{2h}_{1,q}, \dots, \varphi^{2h}_{J^{2h},q}\right) \alpha^{2h} \\ &= \left(\sum_{i=0}^{q+1} c_i \varphi^h_{2-(q+1)+i,q}, \sum_{i=0}^{q+1} c_i \varphi^h_{4-(q+1)+i,q}, \dots, \sum_{i=0}^{q+1} c_i \varphi^h_{2J^{(2h)}-(q+1)+i,q}\right) \alpha^{2h} \\ &= \left(\varphi^h_{1-q,q}, \dots, \varphi^h_{2J^{2h},q}\right) C\alpha^{2h}, \end{aligned}$$

where $C \in \mathbb{R}^{(2J^{2h}+q)\times J^{2h}}$ is element-wise given as $C[i,j] := c_{(i-1)-2(j-1)}$. Since $\varphi^h_{i,q} = 0$ on Ω for $i \notin \{1, \dots, J^h = 2J^{2h} - q\}$, it follows

$$s = \left(\varphi^h_{1,q}, \dots, \varphi^h_{J^h,q}\right) \bar{C}\alpha^{2h} = \left(\varphi^h\right)^T \bar{C}\alpha^{2h},$$

where $\bar{C} \in \mathbb{R}^{J^h \times J^{2h}}$ is obtained by deleting the first q and the last q rows of C. That is, $\bar{C}$ is element-wise defined as

$$\bar{C}[i,j] = c_{(i-1)-2(j-1)+q} = c_{i-2j+q+1}$$

such that $\tilde{C} = I^h_{2h}$. Finally, it holds

$$s = \left(\varphi^h\right)^T \alpha^h = \left(\varphi^h\right)^T I^h_{2h}\alpha^{2h} = \left(\varphi^{2h}\right)^T \alpha^{2h},$$

which completes the proof. □

3.1.2 Splines in Multiple Variables

To extend the concept of spline functions to multiple input variables $x \in \mathbb{R}^P$, $P > 1$, we utilize a tensor product approach. First, for more convenience while working with tensor products, we introduce the multiindex notation.

Definition 3.1.5 (Multiindex)
A P-tupel $j := (j_1, \dots, j_P)^T \in \mathbb{N}_0^P$ of nonnegative integers is called a multiindex. For multi-indices $j, k \in \mathbb{N}_0^P$ and a vector $x \in \mathbb{R}^P$, the following definitions are stated:

1. *Partial order: $j \leq k \;:\Leftrightarrow\; j_p \leq k_p$ for all $p = 1, \dots, P$.*
2. *Absolute value: $|j| := \sum\limits_{p=1}^{P} j_p$.*
3. *Factorial: $j! := \prod\limits_{p=1}^{P} (j_p!)$.*
4. *Power: $x^j := \prod\limits_{p=1}^{P} x_p^{j_p}$.*
5. *Partial derivatives: $\partial^j := \dfrac{\partial^{|j|}}{\partial_1^{j_1} \dots \partial_P^{j_P}}$.*

Tensor Product Splines

To define splines on a finite rectangle

$$\Omega := \bigtimes_{p=1}^{P} \Omega_p \subset \mathbb{R}^P, \text{ where } \Omega_p := [a_p, b_p] \subset \mathbb{R}, \tag{3.19}$$

we consider for each dimension $p = 1, \ldots, P$ the one-dimensional spline spaces $\mathcal{S}_{q_p}(\mathcal{K}_p)$ of degree $q_p \in \mathbb{N}_0$ with knots $\mathcal{K}_p$ (cf. Definition 3.1.1) and a related basis

$$\{\phi^p_{j_p,q_p} \ : \ j_p = 1, \ldots, J_p := m_p + q_p + 1\}. \tag{3.20}$$

Utilizing the multiindex notation, we define the tensor product basis functions

$$\phi_{j,q} \colon \Omega \subset \mathbb{R}^P \to \mathbb{R}, \ x = (x^1, \ldots, x^P) \mapsto \prod_{p=1}^{P} \phi^p_{j_p,q_p}(x^p) \tag{3.21}$$

for $1 \leq j \leq J$. Note that within this definition the subscripts $j \in \mathbb{N}_0^P$ and $q \in \mathbb{N}_0^P$ denote multiindices and not common indices as in definitions within Subsection 3.1.1. However, for $P = 1$ the both frameworks coincide such that the utilization of the same symbols is justified and should not cause any misunderstandings. We now define the space of multivariable splines as the space spanned by the tensor product basis functions.

Definition 3.1.6 (Tensor product splines)
We call the set

$$\mathcal{S}_q(\mathcal{K}) := \operatorname{span}\{\phi_{j,q} \ : \ 1 \leq j \leq J\} \tag{3.22}$$

the space of tensor product splines of degree $q \in \mathbb{N}_0^P$ with knots $\mathcal{K} := \mathcal{K}_1 \times \ldots \times \mathcal{K}_P \subset \mathbb{R}^P$.

Again, note that the same symbols are used as in the case of ordinary spline functions as introduced in Subsection 3.1.1. The distinction whether a tensor product spline or an ordinary spline is considered becomes always clear from the context. By definition, the space of tensor product splines $\mathcal{S}_q(\mathcal{K})$ is a finite dimensional linear space of dimension

$$K := \dim(\mathcal{S}_q(\mathcal{K})) = \prod_{p=1}^{P} J_p = \prod_{p=1}^{P} \dim(\mathcal{S}_{q_p}(\mathcal{K}_p)). \tag{3.23}$$

Tensor Product B-Splines

If especially the B-spline basis $\{\varphi^p_{j_p,q_p} \ : \ j_p = 1, \ldots, J_p := m_p + q_p + 1\}$ is used for all $p = 1, \ldots, P$ to construct the tensor product spline space, we refer to the basis function

$$\varphi_{j,q} \colon \Omega \subset \mathbb{R}^P \to \mathbb{R}, \ x = (x^1, \ldots, x^P) \mapsto \prod_{p=1}^{P} \varphi^p_{j_p,q_p}(x^p) \tag{3.24}$$

as the j-th tensor product B-spline. In Figure 3.2, we graph an exemplary tensor product B-spline of degree $q = (3, 3)$ for $P = 2$ dimensions.

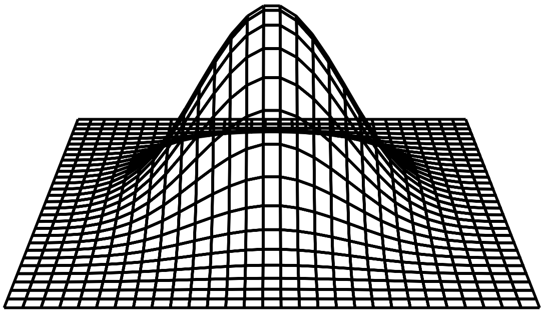

Figure 3.2: Shape of a bicubic tensor product B-spline.

By construction, the following properties of one-dimensional B-splines directly transfer to tensor product B-splines:

1. Normalization: $\varphi_{j,q}(x) \in [0, 1]$ for all $x \in \Omega \subset \mathbb{R}^P$.
2. Local support: $\operatorname{supp}(\varphi_{j,q}) = \bigtimes_{p=1}^{P} [\kappa^p_{j_p-(q_p+1)}, \kappa^p_{j_p}]$.
3. Partition of unity: $\sum_{1 \le j \le J} \varphi_{j,q}(x) = 1$ for all $x \in \Omega$.
4. Integrals: $\int_{\operatorname{supp}(\varphi_{j,q})} \varphi_{j,q} = \prod_{p=1}^{P} \frac{\kappa_{j_p} - \kappa_{j_p-(q_p+1)}}{q_p + 1}$.

Remarks and Notations

In order to implement numerical computations for tensor product spline related problems, we need to sort arbitrary sets of multiindices. Therefore, we consider the lexicographical order which is defined via the bijective map

$$\begin{aligned} \nu\colon \{j \in \mathbb{N}_0^P \mid M \le j \le N\} &\to \left\{1, \dots, \prod_{p=1}^{P} (N_p - M_p + 1)\right\}, \\ j &\mapsto 1 + \sum_{p=1}^{P} \left((j_p - M_p) \left(\prod_{k=1}^{p-1} (N_k - M_k + 1) \right) \right). \end{aligned} \tag{3.25}$$

In the following, whenever sets of arbitrary multiindices are sorted, the sorting is achieved according to this lexicographical order. The sorting of multiindices allows to uniquely identify multiindices from the set $\{j \mid 1 \le j \le J\}$ with common indices from the set $\{1, \dots, K\}$, where K denotes the dimension of the tensor product spline space as in (3.23). Thus, every tensor product spline $s \in \mathcal{S}_q(\mathcal{K})$ possesses a unique expression in terms of its tensor product spline basis as

$$s = \sum_{1\leq j\leq J} \alpha_j \phi_{j,q} = \sum_{\nu(j)=1}^{K} \alpha_{\nu(j)} \phi_{\nu(j),q} = \sum_{k=1}^{K} \alpha_k \phi_{k,q}. \tag{3.26}$$

This notation now provides the link of tensor products and Kronecker products (cf. Definition 2.3.1) as stated in the following lemma. For a more convenient notation, we first define for a given $x \in \mathbb{R}^P$ the vectors

$$\phi(x) := (\phi_{1,q}(x), \dots \phi_{K,q}(x))^T \quad \text{and} \quad \phi^p(x^p) := \left(\phi^p_{1,q_p}(x^p), \dots, \phi^p_{J_p,q_p}(x^p)\right)^T. \tag{3.27}$$

Lemma 3.1.7
For all $x \in \Omega$ it holds

$$\phi(x) = \bigotimes_{p=1}^{P} \phi^p(x^p), \tag{3.28}$$

where $\bigotimes$ denotes the Kronecker product (2.28).

Proof. For arbitrary $k \in \{1, \dots, K\}$ let $j := \nu^{-1}(k)$, where ν denotes the lexicographical order (3.25). Then it holds

$$\phi(x)[k] = \phi_{k,q}(x) = \phi_{j,q}(x) = \prod_{p=1}^{P} \phi^p_{j_p,q_p}(x^p) = \left(\bigotimes_{p=1}^{P} \phi^p(x^p)\right)[k],$$

where $[k]$ denotes the k-th entry of the respective vector. Since k is chosen arbitrarily, we conclude the proof. □

Uniform Tensor Product B-Spline Refinement

If all of the knot sets $\mathcal{K}_p$ are equally spaced with mesh size h_p, the subdivision formula of Theorem 3.1.3 has a natural expansion to uniform tensor product B-splines. As in the one-dimensional case, let

$$\{\varphi^{2h}_{j,q} \,:\, 1 \leq j \leq J^{2h}\} \quad \text{and} \quad \{\varphi^{h}_{j,q} \,:\, 1 \leq j \leq J^{h}\} \tag{3.29}$$

denote the uniform tensor product B-spline basis of the spline spaces $\mathcal{S}_q(\mathcal{K}^{2h})$ and $\mathcal{S}_q(\mathcal{K}^{h})$, respectively, where $h := (h_1, \dots, h_P)^T$. For a uniform spline

$$s \in \mathcal{S}_q(\mathcal{K}^{(2h)}) \subset \mathcal{S}_q(\mathcal{K}^{(h)}) \tag{3.30}$$

it holds

$$s = \left(\varphi^{2h}\right)^T \alpha^{2h} \quad \text{and} \quad s = \left(\varphi^{h}\right)^T \alpha^{h} \tag{3.31}$$

and Lemma 3.1.4 exhibits the following natural extension to uniform tensor product B-splines.

Lemma 3.1.8
For $p = 1, \dots, P$ let the matrices $I_{2h_p}^{h_p} \in \mathbb{R}^{J^{h_p} \times J^{2h_p}}$ be defined as in (3.18). It holds

$$\alpha^h = I_{2h}^h \alpha^{2h} \quad \textit{and} \quad \varphi^{2h} = \left(I_{2h}^h\right)^T \varphi^h, \tag{3.32}$$

where

$$I_{2h}^h := \bigotimes_{p=1}^P I_{2h_p}^{h_p} \in \mathbb{R}^{K^h \times K^{2h}}. \tag{3.33}$$

Proof. Due to Lemma 3.1.7, it holds

$$\left(\varphi^{2h}(x)\right)^T \alpha^{2h} = \left(\bigotimes_{p=1}^P \varphi^{2h_p,p}(x^p)\right)^T \alpha^{2h}$$

for all $x \in \Omega$. Lemma 3.1.4 in combination with the properties of the Kronecker product (cf. Lemma 2.3.2) implies

$$\begin{aligned}\bigotimes_{p=1}^P \varphi^{2h_p,p}(x^p) &= \bigotimes_{p=1}^P \left(\left(I_{2h_p}^{h_p}\right)^T \varphi^{2h_p,p}(x^p)\right) = \left(\bigotimes_{p=1}^P \left(I_{2h_p}^{h_p}\right)^T\right)\left(\bigotimes_{p=1}^P \varphi^{h_p,p}(x^p)\right) \\ &= \left(\bigotimes_{p=1}^P \left(I_{2h_p}^{h_p}\right)^T\right) \varphi^h(x).\end{aligned}$$

Finally, due to the distributivity of the transposition of the Kronecker product (cf. Lemma 2.3.2), it holds

$$(\varphi^{2h})^T \alpha^{2h} = (\varphi^h)^T I_{2h}^h \alpha^{2h} = (\varphi^h)^T \alpha^h$$

which concludes the proof. □

3.2 Penalized Splines

In this section, we present the basic concept of the P-spline method. For further reading and detailed information, we refer to Eubank (1988), Wahba (1990), Green and Silverman (1993), Ruppert et al. (2003), and Fahrmeir et al. (2013). Let a data set

$$\{(x_i, y_i) \in \mathbb{R}^P \times \mathbb{R} : i = 1, \dots, n\} \tag{3.34}$$

be given, where the $y_i \in \mathbb{R}$ are observations of a continuous response variable and the vectors $x_i := (x_i^1, \dots, x_i^P)^T \in \mathbb{R}^P$ represent the corresponding values of the continuous covariates. We assume that the response variable is connected to the covariates via the regression model

$$y_i = s(x_i) + \varepsilon_i, \ \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2), \ i = 1, \dots, n, \tag{3.35}$$

with a sufficiently smooth but further unspecified regression function s. We model the regression function by an adequate spline function, i.e. we assume $s \in \mathcal{S}_q(\mathcal{K})$ with an appropriate and fixed spline space.

3.2.1 Regularization Approach

The most common approach to determine the regression spline s is the least squares method, where s is determined as minimizer of the least squares functional

$$\mathcal{LS}(s) := \sum_{i=1}^{n} \left(s(x_i) - y_i\right)^2 . \tag{3.36}$$

In order to take differing accuracies of the y_i into account the weighted least squares functional

$$\mathcal{WLS}(s) := \sum_{i=1}^{n} \omega_i \left(s(x_i) - y_i\right)^2 , \tag{3.37}$$

with adequate weights ω_i can also be used. This is a special case of the well established generalized least squares method with a diagonal covariance matrix. For example, if heteroscedasticity in the measurement errors is assumed, i.e. $\varepsilon_i \overset{\text{ind}}{\sim} \mathcal{N}(0, \sigma_i^2)$, a typical weighting is $\omega_i := \sigma_i^{-2}$. From a numerical point of view, the consideration of these weights is straightforward since it does not affect the algorithms presented in Chapter 5. However, if there is no reasonable assumption on the accuracy of the observations, it is recommended to consider the unweighted least squares functional (cf. Dierckx, 1993, p. 45). Therefore, we restrict ourselves to the unweighted case (3.36) for reasons of clarity and comprehensibility.

By construction, the least squares term (3.36) measures the proximity of the regression spline to the given observations and its pure minimization leads to a spline function with the best possible fit to the observations (in the least squares sense). If possible, this least squares spline interpolates the data at hand. Since the data contain measurement errors, an as accurate as possible representation is undesirable, but it is rather of interest to capture the general trend in the data instead of the local variation. To achieve this goal, there are basically two approaches (cf. Fahrmeir et al., 2013, p. 431):

1. The adaptive choice of the knots $\mathcal{K}$ based on model choice strategies.
2. The regularization of the least squares problem through the consideration of a roughness penalty.

Within the adaptive approach, multivariate adaptive regression splines provide a widely used method (cf. Friedman, 1991 and Hastie et al., 2009, Chapter 9.4). For the intended incorporation of shape constraints (cf. Chapter 3.3), however, the adaptive approach is improper such that we proceed with the regularization approach, also referred to as the roughness penalty approach. An additional regularization term

$$\mathcal{P} \colon \mathcal{S}_q(\mathcal{K}) \to [0, \infty[\tag{3.38}$$

is introduced, that penalizes a wiggling behavior of s. That means, loosely speaking, the value $\mathcal{P}(s)$ decreases as s becomes smoother. This yields the following general definition of a P-spline in the context of regression modeling.

Definition 3.2.1 (Regression P-spline)
For a given spline space $\mathcal{S}_q(\mathcal{K})$, we call a solution of

$$\min_{s \in \mathcal{S}_q(\mathcal{K})} \mathcal{LS}(s) + \lambda \mathcal{P}(s) \tag{3.39}$$

a regression P-spline in $\mathcal{S}_q(\mathcal{K})$.

As already mentioned, the least squares term (3.36) within the definition of a regression P-spline measures the goodness of fit to the given observations, whereas the regularization term (3.38) penalizes its roughness. The regularization parameter $\lambda > 0$ continuously balances the two competitive terms. As $\lambda \to 0$, the effect of the penalty disappears such that a regression P-spline converges to the ordinary least squares spline which tries to interpolate the given data. As $\lambda \to \infty$, the penalty term dominates the objective function such that a regression P-spline converges to an element in the kernel of $\mathcal{P}$, i.e. an element in the set

$$\text{kern}(\mathcal{P}) := \{s \in \mathcal{S}_q(\mathcal{K}) \ : \ \mathcal{P}(s) = 0\}. \tag{3.40}$$

This fact becomes apparent in Figure 3.3, where a regression P-spline for various regularization parameters is presented. By definition, the choice of the regularization parameter globally influences the resulting regression P-spline. In some applications it might be of interest to control the P-spline only locally, for example if the measurements are assumed to be very exact in some regions while extremely noisy in others. In this case, the regression P-spline should (approximately) interpolate in the exact regions, i.e. less penalization, while being much stronger penalized in the other regions. This could be obtained by defining the regularization parameter as a weighting function

$$\lambda \colon \Omega \to]0, \infty[, \ x \mapsto \lambda(x). \tag{3.41}$$

This approach, however, is not further considered within this thesis since it prevents the use of common criteria for regularization paramter selection (cf. Subsection 3.2.4).

3.2.2 Bases and Penalties

A wide spectrum of P-spline methods exists, depending on the spline basis and the related regularization functional. The most common variants are introduced in this subsection.

Truncated Power Series Basis with Coefficient Penalty

We first consider the case of one single covariate, that is $x_i \in \mathbb{R}$. Ruppert et al. (2003, Chapter 5.5) suggest the use of the quadratic truncated power series basis (3.6) based on

$$m := \min\{\lfloor n/4 \rfloor, 35\} \tag{3.42}$$

knots located by the rule

$$\kappa_j := \left(\frac{j+1}{m+2}\right)\text{-th sample quantile of the } x_i, \;\; j = 1, \dots, m. \tag{3.43}$$

Note that this choice of the knots depends on the data x_i. It holds

$$s(x) = \sum_{j=1}^{J} \alpha_j \eta_{j,q}(x) = \sum_{j=1}^{q+1} \alpha_j x^{j-1} + \sum_{j=q+2}^{J} \alpha_j \max\{0, (x - \kappa_{j-(q+1)})^q\} \tag{3.44}$$

and the penalty term is defined as penalty on the variability of the truncated polynomials, that is

$$\mathcal{P}(s) := \sum_{j=q+2}^{J} \alpha_j^2 = \alpha^T D \alpha, \quad \text{where } D := \operatorname{diag}(\underbrace{0, \dots, 0}_{q+1}, \underbrace{1, \dots, 1}_{J-(q+1)}) \in \mathbb{R}^{J \times J}. \tag{3.45}$$

Remark that in this case the kernel of $\mathcal{P}$ is the space of polynomials of degree q on Ω, i.e.

$$\operatorname{kern}(\mathcal{P}) = \mathcal{P}_q(\Omega). \tag{3.46}$$

According to Fahrmeir et al. (2013, p. 530), this approach can directly be extended to multiple covariates $x_i \in \mathbb{R}^P$ by using the tensor product truncated power series basis and

$$\mathcal{P}(s) := \alpha^T \left(\sum_{p=1}^{P} I_{J_1} \otimes \ldots \otimes I_{J_{p-1}} \otimes D^p \otimes I_{J_{p+1}} \otimes \ldots \otimes I_{J_P} \right) \alpha \tag{3.47}$$

as penalty, where

$$D^p := \operatorname{diag}(\underbrace{0, \dots, 0}_{q_p+1}, \underbrace{1, \dots, 1}_{J_p-(q_p+1)}) \in \mathbb{R}^{J_p \times J_p} \tag{3.48}$$

denotes the one-dimensional penalty matrix for the p-th covariate, I is the identity matrix of respective dimension, and $\otimes$ denotes the Kronecker product. As already mentioned, the truncated power series basis is numerically unstable and therefore rarely used in practice. For theoretical reasons, however, the truncated power series basis with coefficient penalty can be useful (cf. Ruppert et al., 2003, p.70).

B-Spline Basis with Differences Penalty

For one single covariate $x_i \in \mathbb{R}$, Eilers and Marx (1996, 2010) suggest the cubic B-spline basis (3.10) with an arbitrarily large number of equally spaced knots and to base the penalty on higher-order differences of the coefficients of adjacent B-splines. That is

$$\mathcal{P}(s) := \sum_{j=r+1}^{J} \Delta_r(\alpha_j) = \alpha^T (\Delta_r)^T \Delta_r \alpha = \|\Delta_r \alpha\|_2^2, \tag{3.49}$$

where $\Delta_r(\cdot)$ denotes the r-th order backwards difference operator and

$$\Delta_r \in \mathbb{R}^{(J-r)\times J} \tag{3.50}$$

denotes the related difference matrix. As for the truncated power series basis, the B-spline basis representation can be extended to the tensor product B-spline basis representation (3.26) for multiple covariates. The related difference penalty reads

$$\mathcal{P}(s) := \alpha^T \left(\sum_{p=1}^{P} I_{J_1} \otimes \ldots \otimes I_{J_{p-1}} \otimes \left(\Delta^p_{r_p}\right)^T \Delta^p_{r_p} \otimes I_{J_{p+1}} \otimes \ldots \otimes I_{J_P} \right) \alpha, \tag{3.51}$$

where $\Delta^p_{r_p} \in \mathbb{R}^{(J_p - r_p)\times J_p}$ denotes the related difference matrix for the p-th covariate (cf. Fahrmeir et al., 2013, p. 508 and p. 530).

B-Spline Basis with Curvature Penalty

A common measure of smoothness for univariable functions in $\mathcal{C}^2(\Omega)$ with $\Omega \subset \mathbb{R}$ is the integrated squared second derivate (cf. O'Sullivan, 1986), that is

$$\int_\Omega \left(s''(x)\right)^2 \mathrm{d}x. \tag{3.52}$$

This measure is extended to multivariable functions $s \in \mathcal{C}^2(\Omega)$ with $\Omega \subset \mathbb{R}^P$ as integrated square of all partial derivatives of total order two (cf. Eubank, 1988, p. 287 and Green and Silverman, 1993, p. 159), that is

$$\mathcal{P}(s) := \int\limits_\Omega \sum_{p_1=1}^{P} \sum_{p_2=1}^{P} \left(\frac{\partial^2}{\partial x_{p_1} \partial x_{p_2}} s(x) \right)^2 \mathrm{d}x. \tag{3.53}$$

For this curvature penalty, we observe

$$\mathcal{P}(s) = \sum_{|r|=2} \frac{2}{r!} \int\limits_\Omega \left(\partial^r s(x)\right)^2 \mathrm{d}x = \sum_{|r|=2} \frac{2}{r!} \|\partial^r s\|^2_{L^2(\Omega)}, \tag{3.54}$$

where $r \in \mathbb{N}_0^P$ denotes a multiindex and $\|\cdot\|_{L^2(\Omega)}$ denotes the norm on the Lebesgue space $L^2(\Omega)$. More precise, $L^2(\Omega)$ is the space of functions for which the square of the absolute value is Lebesgue integrable, where functions which agree almost everywhere are identified, and $\|\cdot\|_{L^2(\Omega)}$ denotes the related norm. Using the tensor product B-spline representation (3.26), it holds

$$\|\partial^r s\|^2_{L^2(\Omega)} = \langle \partial^r s, \partial^r s\rangle_{L^2(\Omega)} = \sum_{k=1}^{K}\sum_{\ell=1}^{K} \alpha_k \alpha_\ell \left\langle \partial^r \varphi_{k,q}, \partial^r \varphi_{\ell,q} \right\rangle_{L^2(\Omega)} = \alpha^T \Psi_r \alpha \tag{3.55}$$

for each term of the penalty, where $\Psi_r \in \mathbb{R}^{K\times K}$ is element-wise defined as

$$\Psi_r[k,\ell] = \langle \partial^r \varphi_{k,q}, \partial^r \varphi_{\ell,q}\rangle_{L^2(\Omega)} = \int\limits_\Omega \partial^r \varphi_{k,q}(x) \partial^r \varphi_{\ell,q}(x) \mathrm{d}x. \tag{3.56}$$

For numerical reasons, which become clear in Chapter 5, we recommend $m_p = 2^G + 1$ equally spaced knots for some $G \in \mathbb{N}$ and all $p = 1, \dots, P$. From a theoretical point of view, the utilization of the truncated power series basis is also possible with the curvature penalty, but the computation of the occurring integrals tremendously simplifies for the uniform B-spline basis.

Existence and Uniqueness Results

Common to all of the introduced P-splines is that they can be reformulated in terms of an optimization problem for the spline coefficients $\alpha \in \mathbb{R}^K$, i.e. as

$$\min_{\alpha \in \mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda\alpha^T\Lambda\alpha, \tag{3.57}$$

with an adequate basis matrix $\Phi \in \mathbb{R}^{n\times K}$, element-wise defined by $\Phi[i,k] := \phi_{k,q}(x_i)$, and a related symmetric and positive semidefinite penalty matrix $\Lambda \in \mathbb{R}^{K\times K}$. The following theorem facilitates to state existence and uniqueness results on regression P-splines.

Theorem 3.2.2
The optimization problem (3.57) is feasible and a solution is given as a solution of the linear system

$$\left(\Phi^T\Phi + \lambda\Lambda\right)\alpha \overset{!}{=} \Phi^T y \tag{3.58}$$

and vice versa. Further, the solution is unique if and only if $\Phi^T\Phi + \lambda\Lambda \succ 0$.

Proof. Due to

$$\|\Phi\alpha - y\|_2^2 + \lambda\alpha^T\Lambda\alpha = \alpha^T\Phi^T\Phi\alpha - 2(\Phi^T y)^T\alpha + y^T y + \lambda\alpha^T\Lambda\alpha,$$

the optimization problem (3.57) is equivalent to the optimization problem

$$\min_{\alpha \in \mathbb{R}^K} \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda\Lambda\right)\alpha - (\Phi^T y)^T\alpha. \tag{3.59}$$

Since $\Phi^T\Phi \succeq 0$, $\Lambda \succeq 0$, and $\lambda > 0$ it also holds $\Phi^T\Phi + \lambda\Lambda \succeq 0$ such that (3.59) is an unconstrained convex QP. Remark 2.5.3 yields that a solution always exists and is given by a solution of the linear system (3.58) and vice versa. Further, also because of Remark 2.5.3, the solution is unique if and only if $\Phi^T\Phi + \lambda\Lambda \succ 0$. □

In order to state uniqueness results for regression P-splines, we consider the occurring matrices in further detail. We restrict ourselves to the curvature penalty, since it plays a fundamental role within the development of efficient solution algorithms (cf. Chapter 5). For that purpose, the following lemma states some results on the curvature penalty (3.53) itself.

Lemma 3.2.3 (cf. Green and Silverman, 1993, p. 159)
For the curvature penalty (3.53) it holds $0 \le \mathcal{P}(s) < \infty$ *for all* $s \in \mathcal{S}_q(\mathcal{K})$. *Further,* $\mathcal{P}(s) = 0$

if and only if s is an affine hyperplane, that is

$$\operatorname{kern}(\mathcal{P}) = \left\{ s\colon \Omega \subset \mathbb{R}^P \to \mathbb{R} \;:\; \exists \beta \in \mathbb{R}^{P+1} \text{ such that } s(x) = \beta_0 + \sum_{p=1}^{P} \beta_p x^p \;\forall x \in \Omega \right\}. \tag{3.60}$$

Due to this lemma, we are able to state conditions on the covariates to guarantee the uniqueness of the regression P-spline with curvature penalty.

Theorem 3.2.4
A regression P-spline with curvature penalty (3.53) is unique if and only if the covariate vectors $x^p \in \mathbb{R}^n$, $p = 1, \dots, P$, are linearly independent and nonconstant.

Proof. Due to Theorem 3.2.2, a regression P-spline is unique if and only if $\Phi^T\Phi + \lambda\Lambda \succ 0$. Let $0 \neq \alpha \in \mathbb{R}^K$ be given with $\alpha^T\Lambda\alpha = 0$ and let $0 \neq s \in \mathcal{S}_q(\mathcal{K})$ denote the spline with spline coefficients α. Then it holds $\mathcal{P}(s) = 0$ such that s is an affine hyperplane (cf. Lemma 3.2.3). Therefore, it holds

$$\begin{aligned} & s(x_i) = s(x_i^1, \dots, x_i^P) = \beta_0 + \sum_{p=1}^{P} \beta_p x_i^p \quad \text{for all } i = 1, \dots, n \\ \Leftrightarrow\ & (s(x_1), \dots, s(x_n))^T = X\beta, \end{aligned}$$

with adequate coefficients $\beta_0, \dots, \beta_P$ not all equal to zero and

$$X := [1, x^1, \dots, x^P] \in \mathbb{R}^{n \times (P+1)}.$$

Further, it holds

$$\begin{aligned} & (s(x_1), \dots, s(x_n))^T = \sum_{k=1}^{K} \alpha_k \varphi_{j,q}(x_i), \quad \text{for all } i = 1, \dots, n \\ \Leftrightarrow\ & (s(x_1), \dots, s(x_n))^T = \Phi\alpha. \end{aligned}$$

It follows (cf. Björck, 1996, p. 6)

$$\alpha^T\Phi^T\Phi\alpha = 0 \Leftrightarrow \beta^T X^T X\beta = 0 \Leftrightarrow \operatorname{rank}(X) \neq P + 1$$

such that $\alpha^T\Phi^T\Phi\alpha = 0$ if and only if $x^1, \dots, x^P$ are linearly dependent and nonconstant. □

Remark that the assumption of $x^1, \dots, x^P \in \mathbb{R}^n$ being linearly independent and nonconstant is not restrictive in practice. It requires $n > P$ which is trivially fulfilled in applications in the survey statistics and small area framework. Further, if the vectors $x^1, \dots, x^P \in \mathbb{R}^n$ are linearly dependent, we determine a maximal linearly independent subset to perform regression analysis without loosing any information. Finally, a constant vector does not give any information and is simply excluded from the regression process. For the further P-spline variants introduced in Subsection 3.2.2, uniqueness results can be stated in a similar manner under very mild assumption on the covariates. However, we do not further discuss this issue in the following, but assume that the regression P-spline is unique.

Remark 3.2.5
In the following, we assume that the matrix $\Phi^T\Phi + \lambda\Lambda$ is positive definite. Especially, the regression P-spline always exists and is unique (cf. Theorem 3.2.2).

3.2.3 Linear Mixed Model Formulation

As shown in the previous subsection, the splines coefficients $\alpha \in \mathbb{R}^K$ of a regression P-spline are determined via the solution ot the optimization problem (3.57) which is equivalent to the solution of the linear system (3.58). In an alternative approach (cf. Currie and Durban, 2002, Ruppert et al., 2003, Chapter 4.9, and Kauermann, 2005), the P-spline coefficients are determined via a linear mixed model, which have been introduced in Section 2.2. This linear mixed model representation of the P-spline method is outlined in the following. For that purpose, let

$$\Lambda = U\Sigma U^T \tag{3.61}$$

be the eigenvalue decomposition of the symmetric and positive semidefinite penalty matrix Λ. That is, Σ is a diagonal matrix containing the K real and nonnegative eigenvalues of Λ in descending order and U is an orthogonal matrix containing the related eigenvectors as columns. Let

$$r := \operatorname{rank}(\Lambda) \tag{3.62}$$

denote the rank of Λ such that the last $K - r$ eigenvalues of Λ are zero and let $\Sigma_+ \in \mathbb{R}^{r\times r}$ be the diagonal matrix containing the strictly positive eigenvalues of Λ. Further, let

$$U = [U_+, U_0] \tag{3.63}$$

be decomposed into the matrix $U_+ \in \mathbb{R}^{K\times r}$ containing the eigenvectors related to the strictly positive eigenvalues and $U_0 \in \mathbb{R}^{K\times(K-r)}$ containing the eigenvectors related to the zero eigenvalues of Λ. This implies

$$\Lambda = U\Sigma U^T = U_+\Sigma_+U_+^T. \tag{3.64}$$

Let $\Sigma_+^{1/2}$ denote the square root of Σ_+, i.e. a diagonal matrix containing the square roots of the nonnegative eigenvalues of Λ, and define

$$\tilde{X} := U_0 \in \mathbb{R}^{K\times(K-r)} \quad \text{and} \quad \tilde{Z} := U_+\Sigma_+^{1/2} \in \mathbb{R}^{K\times r}. \tag{3.65}$$

Then it holds:

1. $\operatorname{rank}([\tilde{X}, \tilde{Z}]) = \operatorname{rank}([U_0, U_+\Sigma_+^{1/2}]) = K$.
2. $\tilde{Z}^T\Lambda\tilde{Z} = (\Sigma_+^{1/2})^T U_+^T U_+ \Sigma_+ U_+^T U_+^T \Sigma_+^{1/2} = (\Sigma_+^{1/2})^T I_r \Sigma_+ I_r \Sigma_+^{1/2} = I_r$.
3. $\tilde{X}^T\Lambda\tilde{X} = U_0^T U_+ \Sigma_+ U_+^T U_0 = 0\Sigma_+ 0 = 0$, since the columns of U are orthogonal.

For arbitrary $\alpha \in \mathbb{R}^K$ let $\beta \in \mathbb{R}^{K-r}$ and $\gamma \in \mathbb{R}^r$ be defined by

$$\alpha = \tilde{X}\beta + \tilde{Z}\gamma \;\Leftrightarrow\; (\beta^T, \gamma^T)^T := [\tilde{X}, \tilde{Z}]^{-1}\alpha. \tag{3.66}$$

Due to this preliminaries, it follows for the optimization problem (3.57)

$$\begin{aligned}
&\min_{\alpha\in\mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda\alpha^T\Lambda\alpha \\
\Leftrightarrow\ &\min_{\alpha\in\mathbb{R}^K} \alpha^T\Phi^T\Phi\alpha + \lambda\alpha^T\Lambda\alpha - 2(\Phi^T y)^T\alpha \\
\Leftrightarrow\ &\min_{\beta\in\mathbb{R}^{K-r},\gamma\in\mathbb{R}^r} \beta^T\tilde{X}^T\Phi^T\Phi\tilde{X}\beta + \gamma^T\tilde{Z}^T\Phi^T\Phi\tilde{Z}\gamma + \gamma^T\tilde{Z}^T\Phi^T\Phi\tilde{X}\beta + \beta^T\tilde{X}^T\Phi^T\Phi\tilde{Z}\gamma \\
&\qquad + \lambda\beta^T\tilde{X}^T\Lambda\tilde{X}\beta + \lambda\gamma^T\tilde{Z}^T\Lambda\tilde{Z}\gamma - 2\beta^T\tilde{X}^T\Phi^T y - 2\gamma^T\tilde{Z}^T\Phi^T y \\
\Leftrightarrow\ &\min_{\beta\in\mathbb{R}^{K-r},\gamma\in\mathbb{R}^r} \beta^T\tilde{X}^T\Phi^T\Phi\tilde{X}\beta + \gamma^T\tilde{Z}^T\Phi^T\Phi\tilde{Z}\gamma + \gamma^T\tilde{Z}^T\Phi^T\Phi\tilde{X}\beta + \beta^T\tilde{X}^T\Phi^T\Phi\tilde{Z}\gamma \\
&\qquad + \lambda\gamma^T\gamma - 2\beta^T\tilde{X}^T\Phi^T y - 2\gamma^T\tilde{Z}^T\Phi^T y.
\end{aligned} \tag{3.67}$$

By defining

$$X := \Phi\tilde{X} \in \mathbb{R}^{n\times(K-r)} \quad\text{and}\quad Z := \Phi\tilde{Z} \in \mathbb{R}^{n\times r}, \tag{3.68}$$

we deduce

$$\begin{aligned}
&\min_{\alpha\in\mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda\alpha^T\Lambda\alpha \\
\Leftrightarrow\ &\min_{\beta\in\mathbb{R}^{K-r},\gamma\in\mathbb{R}^r} \beta^T X^T X\beta + \gamma^T Z^T Z\gamma + \gamma^T Z^T X\beta + \beta^T X^T Z\gamma + \lambda\gamma^T\gamma - 2\beta^T X^T y - 2\gamma^T Z^T y \\
\Leftrightarrow\ &\min_{\beta\in\mathbb{R}^{K-r},\gamma\in\mathbb{R}^r} \frac{1}{2}(\beta^T,\gamma^T)\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + \lambda I_r\end{bmatrix}\begin{pmatrix}\beta\\ \gamma\end{pmatrix} - \left(y^T X, y^T Z\right)\begin{pmatrix}\beta\\ \gamma\end{pmatrix}.
\end{aligned} \tag{3.69}$$

This is an unconstrained convex QP, since it is equivalent to the unconstrained convex QP (3.59). Because of Remark 2.5.3 it finally follows

$$\begin{aligned}
&\min_{\alpha\in\mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda\alpha^T\Lambda\alpha \\
\Leftrightarrow\ &\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + \lambda I_r\end{bmatrix}\begin{pmatrix}\beta\\ \gamma\end{pmatrix} \overset{!}{=} \begin{pmatrix} X^T y\\ Z^T y\end{pmatrix}.
\end{aligned} \tag{3.70}$$

For fixed $\lambda := \sigma_\varepsilon^2/\sigma_\gamma^2$, this is equivalent to the mixed model equation (2.19) to the linear mixed model

$$y = X\beta + Z\gamma + \varepsilon,\ \gamma \sim \mathcal{N}(0, \sigma_\gamma^2 I_r),\ \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_n). \tag{3.71}$$

This allows to embed the penalized spline method into the theory of mixed models and especially to apply the related LMM methodology and software.

3.2.4 Regularization Parameter Selection

A beneficial property of the regression P-spline is that its form is continuously controlled from the least squares hyperplane (as $\lambda \to \infty$) to the least squares spline (as $\lambda \to 0$) by the one single regularization parameter $\lambda > 0$. This fact becomes apparent from Figure 3.3, where the regression P-spline for various regularization parameters is presented. Therefore, the exact number and position of the knots are of minor importance, provided the regularization parameter is appropriately chosen (cf. Fahrmeir et al., 2013, p. 433). In the following, we present some approaches to data-driven regularization parameter selection.

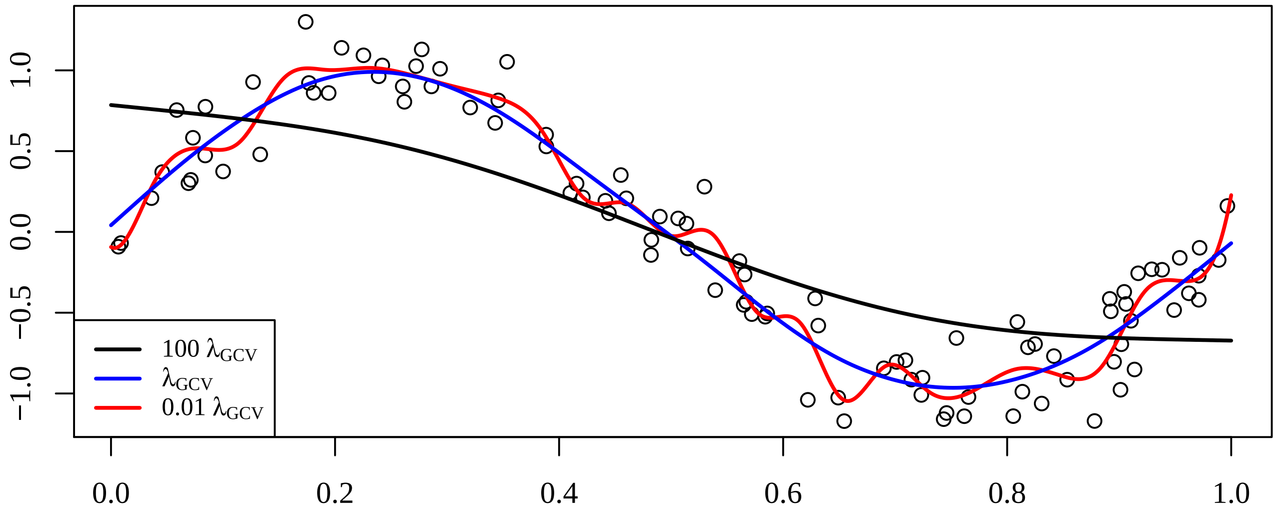

Figure 3.3: Regression P-spline for various regularization parameters.

CV and GCV Method

In the context of regularization parameter selection, cross-validation (CV) methods are widely used (cf. Wahba, 1990, Chapter 4). Let therefore

$$\widehat{\alpha}_\lambda := \left(\Phi^T\Phi + \lambda\Lambda\right)^{-1} \Phi^T y \tag{3.72}$$

denote the unique solution of the linear system (3.58). Consequently, the vector of the evaluations of the regression P-spline at the covariates, i.e. the model predicts, is given as

$$\widehat{y}_\lambda := \Phi\widehat{\alpha}_\lambda = \Phi \left(\Phi^T\Phi + \lambda\Lambda\right)^{-1} \Phi^T y = S_\lambda y, \tag{3.73}$$

where the matrix

$$S_\lambda := \Phi \left(\Phi^T\Phi + \lambda\Lambda\right)^{-1} \Phi^T \in \mathbb{R}^{n\times n} \tag{3.74}$$

is referred to as the hat matrix or the prediction matrix. The initial cross-validation approach determines the regularization parameter λ_{CV} as a solution of

$$\min_{\lambda>0} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_\lambda[i]}{1 - S_\lambda[i,i]} \right)^2, \tag{3.75}$$

where $[\cdot]$ and $[\cdot,\cdot]$ denote the entries of the respective vector and matrix, respectively. In practice, the calculation of the hat matrix and its diagonal elements can be numerically complex.

For this reason, the diagonal elements $S_\lambda[i,i]$ are frequently approximated by their average

$$S_\lambda[i,i] \approx \frac{1}{n}\sum_{j=1}^{n} S_\lambda[j,j] = \frac{1}{n}\mathrm{tr}(S_\lambda), \;\; i = 1,\dots,n, \tag{3.76}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix. This leads to the generalized cross-validation (GCV) method, where the regularization parameterλ_{GCV} is determined as a solution of

$$\min_{\lambda>0} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \widehat{y}_\lambda[i]}{1-\mathrm{tr}(S_\lambda)n^{-1}}\right)^2 . \tag{3.77}$$

In order to solve the optimization problems (3.75) and (3.77) for determining the regularization parameter, it is common to perform a grid search. That is, a reasonable subset of parameters is specified and the parameter with the smallest objective function value is selected.

L-Curve Method

The L-curve criterion, initially proposed by Hansen (1992), describes the trade off between the goodness of fit and the smoothness of the regression function. For arbitrary λ let again

$$\widehat{\alpha}_\lambda := \left(\Phi^T\Phi + \lambda\Lambda\right)^{-1}\Phi^T y \tag{3.78}$$

denote the unique solution of the regularized least squares problem (3.57). Defining the functions

$$\omega_1,\omega_2\colon\,]0,\infty[\to \mathbb{R}, \quad \omega_1(\lambda) := \log\left(\|\Phi\widehat{\alpha}_\lambda - y\|_2^2\right), \quad \omega_2(\lambda) := \log\left(\widehat{\alpha}_\lambda^T\Lambda\widehat{\alpha}_\lambda\right), \tag{3.79}$$

the set

$$L := \{(\omega_1(\lambda),\omega_2(\lambda)) \;:\; \lambda > 0\} \tag{3.80}$$

is called the L-curve of the regularized least squares problem (3.57). The name originates from its particular L-shape. The L-curve method determines the regularization parameter λ_{L} as a solution of

$$\max_{\lambda>0} \frac{{\omega_1}'(\lambda){\omega_2}''(\lambda) - {\omega_1}''(\lambda){\omega_2}'(\lambda)}{\left({\omega_1}'(\lambda)^2 + {\omega_2}'(\lambda)^2\right)^{3/2}}. \tag{3.81}$$

In this approach, the regularization parameter is also determined by a grid search.

ML and REML Method

An attractive consequence of the LMM representation of the penalized spline method presented in (3.71) is that in this case the regularization parameter is already given as the variance ratio

$$\lambda := \sigma_\varepsilon^2/\sigma_\gamma^2. \tag{3.82}$$

As mentioned in Section 2.2, these variance parameter are in general unknown in practice but can be estimated from the given data. Using the maximum likelihood (ML) or the restricted maximum likelihood (REML) estimator for the variance components, we obtain the regularization parameters

$$\lambda_{\mathrm{ML}} := \frac{\widehat{\sigma}^2_{\varepsilon,\mathrm{ML}}}{\widehat{\sigma}^2_{\gamma,\mathrm{ML}}} \quad \text{and} \quad \lambda_{\mathrm{REML}} := \frac{\widehat{\sigma}^2_{\varepsilon,\mathrm{REML}}}{\widehat{\sigma}^2_{\gamma,\mathrm{REML}}}, \tag{3.83}$$

respectively. For further details we refer to Ruppert et al. (2003, Chapter 5.2) and Kauermann (2005).

3.3 Penalized Splines with Shape Constraints

In several applications, the regression function is completely unknown but there exists some information about its global shape, such as monotonicity or convexity. Although the penalized spline method is a very flexible approach to regression analysis, it does not take the required shape constraints into account. In order to provide more realistic regression functions, it is important to develop adequate approaches to shape constrained P-splines. For the B-spline basis with differences penalty (cf. Subsection 3.2.2) special types of shape constraints are considered by Bollaerts et al. (2006) and Meyer (2012). A general framework for shape constrained P-splines, however, is not introduced up to the present time. Therefore, we provide a general framework for the incorporation of shape constraints into the penalized spline method in the following.

Shape Constrained Regression P-Splines

Constraints on the shape of a sufficiently often differentiable function can always be translated into constraints on the sign of its (partial) derivatives. For example, in order to enforce a monotonically increasing behavior of a spline function $s \in \mathcal{S}_q(\mathcal{K})$ in the first covariate, we have to claim

$$\frac{\partial}{\partial x_1} s(x^1, \ldots, x^P) \geq 0. \tag{3.84}$$

A concave behavior in the third covariate is exemplary ensured by

$$\frac{\partial^2}{\partial x_3^2} s(x^1, \ldots, x^P) \leq 0. \tag{3.85}$$

More general, let for $p = 1, \ldots, P$ the sets $I^p_{\geq} \subset \mathbb{N}_0$ and $I^p_{\leq} \subset \mathbb{N}_0$ denote index sets indicating where

$$\frac{\partial^{r_p}}{\partial x_p} s \geq 0, \; r_p \in I^p_{\geq}, \quad \text{and} \quad \frac{\partial^{r_p}}{\partial x_p} s \leq 0, \; r_p \in I^p_{\leq}, \tag{3.86}$$

is required. For example, if we demand for a spline function to be nonnegative and monotone decreasing in the first covariate and concave in the second covariate, we define $I^1_{\geq} = \{0\}$,

$I^1_\leq = \{1\}$, and $I^2_\leq = \{2\}$. For a more convenient notation, let e_p denote the p-th unit vector and define the sets of multiindices

$$\begin{aligned} I_\geq &:= \{r \in \mathbb{N}_0^P \ : \ r = r_p e_p, \ r_p \in I^p_\geq, \ p = 1, \dots, P\}, \\ I_\leq &:= \{r \in \mathbb{N}_0^P \ : \ r = r_p e_p, \ r_p \in I^p_\leq, \ p = 1, \dots, P\}. \end{aligned} \tag{3.87}$$

This allows for a neat incorporation of shape constraints into the most general form of the penalized spline method (3.39) as stated in the following definition.

Definition 3.3.1 (Shape constrained regression P-spline)
For a given spline space $\mathcal{S}_q(\mathcal{K})$, we call a solution of the optimization problem

$$\begin{aligned} \min_{s \in \mathcal{S}_q(\mathcal{K})} \quad & \mathcal{LS}(s) + \lambda \mathcal{P}(s) \\ \text{s.t.} \quad & \partial^r s \geq 0, \ r \in I_\geq \\ & \partial^r s \leq 0, \ r \in I_\leq \end{aligned} \tag{3.88}$$

a shape constrained regression P-spline in $\mathcal{S}_q(\mathcal{K})$.

For the numerical implementation of the shape constraints it suffices to control the partial derivatives at an appropriate discretization of Ω. Therefore, let

$$\bigtimes_{p=1}^{P} \{\tau^p_{j_p} \ : \ j_p = 1, \dots, M_p\} = \{\tau_j := (\tau^1_{j_1}, \dots, \tau^P_{j_P})^T \ : \ 1 \leq j \leq M\} \tag{3.89}$$

denote an arbitrary discretization of Ω. Defining $T := M_1 \cdot \ldots \cdot M_P \in \mathbb{N}$ and using the lexicographical order (3.25), that is

$$\begin{aligned} \nu \colon \{j \in \mathbb{N}^P \ : \ 1 \leq j \leq M\} &\to \{1, \dots, T\}, \\ j &\mapsto 1 + \sum_{p=1}^{P} \left((j_p - 1) \left(\prod_{k=1}^{p-1} M_p \right) \right), \end{aligned} \tag{3.90}$$

we uniquely identify the discretization points as $\tau_j = \tau_{\nu(j)} = \tau_t$, where $t := \nu(j)$. The basis representation of a tensor product spline (3.26) then yields

$$\begin{pmatrix} \partial^r s(\tau_1) \\ \vdots \\ \partial^r s(\tau_T) \end{pmatrix} = \Gamma_r \alpha, \quad \text{where } \Gamma_r := \begin{bmatrix} \partial^r \phi_{1,q}(\tau_1) & \dots & \partial^r \phi_{K,q}(\tau_1) \\ \vdots & \ddots & \vdots \\ \partial^r \phi_{1,q}(\tau_T) & \dots & \partial^r \phi_{K,q}(\tau_T) \end{bmatrix} \in \mathbb{R}^{T \times K}. \tag{3.91}$$

We reformulate the shape constrained regression P-spline problem (3.88) in terms of the related spline coefficients as

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^K} \quad & \|\Phi\alpha - y\|_2^2 + \lambda \alpha^T \Lambda \alpha \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0, \ r \in I_\leq \\ & \Gamma_r \alpha \geq 0, \ r \in I_\geq. \end{aligned} \tag{3.92}$$

It should be noted that the reformulation (3.92) of the shape constrained regression P-spline problem (3.88) is not an equivalent reformulation, but only a sufficient. That is, there might be a solution of (3.92) that is not a solution of (3.88). To ensure equivalence of the the both optimization problems the choice of the discretization of Ω is crucial. An equivalent formulation, however, is only possible for some special types of shape constraints (cf. Meyer, 2012) at first appearance, since there is no control of the behavior of the P-spline function between the discretization points. At this point, the penalty term plays a decisive role. Due to the penalty, the shape constrained regression P-spline is sufficiently smooth such that, provide the discretization of Ω is sufficiently fine, no wiggling behavior between the discretization points is possible. Therefore, without loss of generality, we can assume equivalence of the optimization problems (3.92) and (3.88) for a sufficiently fine (equally spaced) discretization of Ω.

Existence and Uniqueness Results

As for the (unconstrained) regression P-spline, we equivalently reformulate the optimization problem (3.88) in terms of the spline coefficients of the shape constrained regression P-spline (3.92) and therefore obtain the optimization problem

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^K} \quad & \frac{1}{2}\alpha^T \left(\Phi^T\Phi + \lambda\Lambda\right)\alpha - (\Phi^T y)^T\alpha \\ \text{s.t.} \quad & \Gamma_r\alpha \leq 0, \ \ r \in I_{\leq} \\ & \Gamma_r\alpha \geq 0, \ \ r \in I_{\geq}. \end{aligned} \tag{3.93}$$

Due to Remark 3.2.5, this is a strictly convex QP which allows to state an existence and uniqueness result for shape constrained regression P-splines in the following theorem.

Theorem 3.3.2
A shape constrained regression P-spline, defined as a solution of (3.88), exists and is unique.

Proof. The shape constrained regression P-spline problem (3.88) is equivalent to the strictly convex QP (3.93) such that the solution is unique, provided it exists (cf. Remark 2.5.3). To show the existence of the solution it suffices to show that the optimal value f^* of (3.93) is finite (cf. Theorem 2.5.4). Since 0 is a feasible point of the QP, it holds $f^* \leq 0 < \infty$. Let $\tilde{f}$ denote the optimal value of the unconstrained convex QP (3.59) such that $\tilde{f} \leq f^*$. Finally, Remark 2.5.3 implies $-\infty < \tilde{f}$ which concludes the proof. □

Chapter 4

Penalized Splines in Small Area Estimation

Sample surveys are a widely used and cost effective tool to provide estimates of unknown population parameters such as means, totals, or proportions. Nowadays, there is an increasing demand for information not merely on the level of the target population, but also on the level of subpopulations, called areas or domains, defined geographically or by content. Due to cost restrictions or a priori unplanned domains, it frequently occurs that some of theses subpopulations are not adequately reflected within the overall sample in the sense that the related subsamples are very small or even nonexistent.

In this context, these areas are therefore often referred to as small areas. The classical design-based estimators, however, require relatively large sample sizes to provide reliable estimates, i.e. they generally yield estimates of inadequate statistical precision for those small areas. To obtain accurate small area information, model-based estimation techniques have to be applied. These methods increase the area-specific effective sample size by borrowing strength from other similar areas through adequate statistical models, providing a link to the related areas by the use of auxiliary data. Thus, the determination of suitable linking models is crucial for satisfactory small area estimation (SAE). The resulting models are referred to as small area models and often rely on a simple, mainly (generalized) linear relationship within the data. Since model misspecification can result in biased estimators, it is beneficial to consider more flexible, not necessarily linear, models which adjust the data at hand in a more realistic manner. As a consequence, the penalized spline method has recently been considered in the context of small area estimation (cf. Opsomer et al., 2008; Ugarte et al., 2009).

In addition to that, even more realistic small area models could be obtained through the incorporation of reasonable shape constraints. Especially in small area estimation, where the small sample data frequently do not reflect the general trend within the underlying population, the incorporation of shape constraints is expected to yield more accurate estimates. At the present time, however, shape constrained small area models have not been considered in an entirely satisfactory manner (cf. Wagner et al., 2017). In this chapter, we therefore introduce a flexible small area model based on the penalized spline method that allows for the incorporation of arbitrary kinds of shape constraints as well as an autonomous control of the smoothness of the underlying P-spline function.

This chapter is organized as follows. Section 4.1 presents the fundamental concepts and notations for small area estimation and especially introduces the penalized spline small area model in the linear mixed model framework. In Section 4.2, we first develop an alternative approach to determine the parameters of the penalized spline small area model via the solution of an optimization problem instead as from a linear mixed model. Based on this optimization approach, we incorporate arbitrary kinds of shape constraints for the P-spline function that underlies the small area model in Section 4.3. Section 4.4 presents the connection between the penalized spline method in small area estimation and in the regression context and in Chapter 4.5, we develop an appropriate mean squared error (MSE) estimator for the previously considered shape constrained P-spline small area estimator. In Section 4.6, we conduct a simulation study in order to analyze the performance of the proposed point estimator and the related MSE-estimator. Finally, in Section 4.7, we apply the shape constrained spline-based small area estimator for the estimation of spruce timber reserves in RLP (cf. Section 1.1).

4.1 Fundamentals of Small Area Estimation

As mentioned in Section 2.1, estimation methods in survey statistics can basically be distinguished into design-based and model-based approaches. In the design-based framework, randomness of an estimator is completely determined by the sampling design and the associated (design-based) inference is based on the set of all possible samples. This is in contrast to the model-based framework, where the finite population itself is treated as a random realization from a superpopulation model (cf. Bolfarine and Zacks, 1992). The model-based approach, especially in the context of small area estimation, is further addressed in this section following Rao (2003), Datta (2009), Münnich et al. (2013), and Rao and Molina (2015).

4.1.1 Basic Terminology

Within the survey framework, as introduced in Section 2.1, the interest is frequently not only in estimating a target parameter for the entire population, but also in the simultaneous estimation of this parameter for several subgroups of the population, called areas or domains. When subgroup membership is determined by geographical regions such as states, counties, districts, or municipalities, it is referred to as areas. Subgroups defined by content, such as specific age or sex groups, are associated with the term domains. For reasons of simplification, only the term area is used in the following.

To obtain an adequate notation, let the population $\mathcal{U}$ be divided into $D \in \mathbb{N}$ exhaustive and mutually exclusive areas $\mathcal{U}_d$, $d = 1, \ldots, D$, each of size $N_d := |\mathcal{U}_d|$ such that $N = N_1 + \ldots + N_D$. In order to indicate for the area membership of a unit, we define

$$y_{i,d} := \begin{cases} y_i, & \text{if } i \in \mathcal{U}_d \\ 0, & \text{else} \end{cases} \tag{4.1}$$

for the response value of the variable of interest and analogously denote

$$x_{i,d} := \begin{cases} x_i, & \text{if } i \in \mathcal{U}_d \\ 0, & \text{else} \end{cases} \tag{4.2}$$

for the unit-specific auxiliary information. According to the population, the sample $\mathcal{S}$ is divided into the area-specific subsamples $\mathcal{S}_d := \mathcal{S} \cap \mathcal{U}_d$, $d = 1, \ldots, D$, each of size $n_d := |\mathcal{S}_d|$ such that

$n = n_1 + \ldots + n_D$. We denote the related subpopulation parameters as θ_d, $d = 1, \ldots, D$, for example the area-specific total of the variable of interest

$$\theta_d = \tau_{Y,d} := \sum_{i \in \mathcal{U}_d} y_{i,d} \tag{4.3}$$

or the area-specific mean of the variable of interest

$$\theta_d = \mu_{Y,d} := \frac{1}{N_d} \sum_{i \in \mathcal{U}_d} y_{i,d} = \tau_{Y,d}/N_d. \tag{4.4}$$

Under an independent treatment of the areas, i.e. we consider each area as a separate population, the traditional direct methods such as the HT-estimator (2.12) or the GREG-estimator (2.15) can be applied to provide estimates also on the area-level. This is in general possible if the areas are planned, that is if the area membership of the units is known a priori and incorporated into the sampling design. In practice, however, unplanned areas frequently occur, where the area membership is not incorporated into the sampling design such that the area-specific sample sizes are random. In that case, very small subsample sizes for at least some of these areas can occur and even the case of unsampled areas, i.e. $n_d = 0$, is possible. Note that, due to cost restrictions on the overall sample size, very small subsample sizes can also occur for planned areas. Areas with small subsample sizes are difficult to handle by the traditional direct methods. If the area-specific sample size is small, the unit-specific inclusion probabilities (2.5) and (2.6) become small as well, which results in large variance estimates for the direct estimators as indicated by (2.14) and (2.17). For unsampled areas, these direct methods can even not be applied. This leads to the widely used definition of the term small area according to Rao and Molina (2015, p. 2).

Definition 4.1.1 (Small area)
An area is regarded as small, if the area-specific sample size is not large enough to support direct estimates of adequate precision.

Even if Definition 4.1.1 is widely used, it has to be controversially discussed how adequate precision is defined, since the precision requirement strongly depends on the application. As a point in case, within the scope of the German Census 2011 a relative root mean squared error (RRMSE) of less or equal than 0.5% is required (cf. Münnich et al., 2012).

An alternative approach is to define the term small area via the sample fraction n_d/N_d. Purcell and Kish (1979) for example refer to an area as small if the sample fraction is less than 10%. In the following, we only assume that (some of) the area-specific subsamples are very small or even nonexistent and leave the definition of when direct estimates are inadequate to the user.

4.1.2 Small Area Estimators

In order to obtain reliable small area estimates, indirect methods have to applied that borrow strength from related areas or time periods to increase the effective sample size. Traditional indirect estimation methods are based on implicit models that provide a link to related small areas through supplementary data. Such estimators include synthetic estimators and composite estimators (cf. Rao and Molina, 2015, Chapter 3). These estimators are generally design biased and the design bias does not vanish with increasing sample size. Their design variances, however, are usually small relative to the design variances of the direct methods. This can result in a reduction of the MSE, which is the main reason for the utilization of indirect estimators. In contrast to implicit linking models, explicit linking models take into account for area-specific random effects that in particular allow to consider variation between the areas that is not explained by the covariates. These models are referred to as small area models and the related estimators are called model-based small area estimators. The use of model-based small area estimators offers several advantages (cf. Rao and Molina, 2015, p. 5):

- Optimal estimators can be derived under the assumed model.
- Area-specific precision measures can be associated with each estimator.
- Models can be validated from the sample data.
- A variety of models can be utilized depending on the nature of the response variable and the complexity of the data structure.

Model-based small area estimators, however, can be considerably biased if the assumed model is incorrect. Therefore, the determination of suitable small area models is crucial in order to obtain reliable small area estimates. These models are basically classified into area-level models, where only aggregated auxiliary information on area-level is available as covariates, and unit-level models, where complete individual information is utilized as auxiliary data. The most common (generalized) linear area- and unit-level model as well as a small area model based on the penalized spline method are introduced in the following.

Basic Area-Level Model

Let $\widehat{\theta}_d^{\text{DIR}}$ denote a direct estimator of the area-specific target parameter θ_d and let $z_d \in \mathbb{R}^P$ denote known area-specific auxiliary data, for example the true area-specific mean of the covariate vectors

$$\mu_{X,d} := \frac{1}{N_d} \sum_{i \in \mathcal{U}_d} x_{i,d}. \tag{4.5}$$

The most popular small area model on the area-level is proposed by Fay and Herriot (1979) and consists of two stages:

1. Sampling model: $\widehat{\theta}_d^{\text{DIR}} = \theta_d + \varepsilon_d$, $\varepsilon_d \overset{\text{ind}}{\sim} \mathcal{N}(0, \sigma_d^2)$, $d = 1, \dots, D$.
2. Linking model: $\theta_d = z_d^T \beta + u_d$, $u_d \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_u^2)$, $d = 1, \dots, D$.

The area-specific random effects u_d are further assumed to be independent of the sampling errors ε_d. In matrix vector notation the area-level model reads

$$\widehat{\theta}^{\text{DIR}} = Z\beta + u + \varepsilon,\ u \sim \mathcal{N}(0, \sigma_u^2 I_D),\ \varepsilon \sim \mathcal{N}(0, \operatorname{diag}(\sigma_1^2, \dots, \sigma_D^2)), \tag{4.6}$$

where $\widehat{\theta}^{\text{DIR}} := (\widehat{\theta}_1^{\text{DIR}}, \dots, \widehat{\theta}_D^{\text{DIR}})^T$ and $Z \in \mathbb{R}^{D \times P}$ is row-wise defined by z_d^T. This is a linear mixed model as introduced in Chapter 2.2 such that the BLUE of β and the BLUP of u can be obtained, leading to the BLUP of the target parameters θ_d for all areas $d = 1, \dots, D$.

Definition 4.1.2 (FH-estimator)
The Fay-Herriot (FH) estimator of the area-specific target parameters θ_d, $d = 1, \ldots, D$, is defined as

$$\widehat{\theta}_d^{FH} := z_d^T \widehat{\beta}^{BLUE} + \widehat{u}_d^{BLUP}. \tag{4.7}$$

According to Rao and Molina (2015, p. 124) it holds $V = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_D^2) + \sigma_u^2 I_D$ and we obtain

$$\widehat{u}_d^{\text{BLUP}} = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_d^2} \left(\widehat{\theta}_d^{\text{DIR}} - z_d^T \widehat{\beta}^{\text{BLUE}} \right). \tag{4.8}$$

This yields the more common representation of the FH-estimator as

$$\begin{aligned} \widehat{\theta}_d^{\text{FH}} &= z_d^T \widehat{\beta}^{\text{BLUE}} + \frac{\sigma_u^2}{\sigma_u^2 + \sigma_d^2} \left(\widehat{\theta}_d^{\text{DIR}} - z_d^T \widehat{\beta}^{\text{BLUE}} \right) \\ &= \gamma_d \widehat{\theta}_d^{\text{DIR}} + (1 - \gamma_d) z_d^T \widehat{\beta}^{\text{BLUE}}, \end{aligned} \tag{4.9}$$

where

$$\gamma_d := \frac{\sigma_u^2}{\sigma_u^2 + \sigma_d^2} \in [0, 1]. \tag{4.10}$$

The FH-estimator is therefore a composite estimator with shrinkage coefficient γ_d. For areas with a small sampling variance σ_d^2 relative to the model variance σ_u^2, more weight is assigned to the direct estimator, whereas in areas with a comparatively small model variance more emphasis is given to the synthetic part.

Basic Unit-Level Model

Assume now that complete unit-specific covariate information is observed, that is x_i is known for all $i \in \mathcal{U}$. The most popular small area model on the unit-level is proposed by Battese et al. (1988) and given by the nested error linear regression model

$$y_{i,d} = x_{i,d}^T \beta + u_d + \varepsilon_{i,d}, \; i \in \mathcal{S}. \tag{4.11}$$

The area-specific random effects $u_d \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_u^2)$ are assumed to be independent of the unit-specific random errors $\varepsilon_{i,d} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$. In matrix vector notation the basic unit-level model reads

$$y = X\beta + Wu + \varepsilon, \; u \sim \mathcal{N}(0, \sigma_u^2 I_D), \; \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_n), \tag{4.12}$$

where $W \in \mathbb{R}^{n \times D}$ is element-wise defined as

$$W[i, d] := \begin{cases} 1, & \text{if } i \in \mathcal{S}_d \\ 0, & \text{else} \end{cases} \tag{4.13}$$

and expresses the link between a sampled unit and its corresponding area. This is a special case of a linear mixed model and is referred to as random intercept model. Utilizing the BLUE of β and the BLUP of u, we obtain the model predictions

$$\widehat{y}_{i,d} := x_{i,d}^T \widehat{\beta}^{\text{BLUE}} + \widehat{u}_d^{\text{BLUP}}, \ i \in \mathcal{U}, \tag{4.14}$$

for the entire population, which defines a model-based small area estimator on the unit-level.

Definition 4.1.3 (BHF-estimator)
The Battese-Harter-Fuller (BHF) estimator of the area-specific target parameters θ_d, $d = 1, \ldots, D$, is defined as

$$\widehat{\theta}_d^{BHF} := f(\widehat{y}_{i,d} : i \in \mathcal{U}_d). \tag{4.15}$$

If the interest is for example in the estimation of the area-specific means, the BHF-estimator simplifies to

$$\widehat{\mu}_{Y,d}^{\text{BHF}} = \mu_{X,d}^T \widehat{\beta}^{\text{BLUE}} + \widehat{u}_d^{\text{BLUP}}, \ d = 1, \ldots, D. \tag{4.16}$$

This in particular shows that $\widehat{\mu}_{Y,d}^{\text{BHF}}$ is the BLUP of $\mu_{Y,d}$. According to Rao and Molina (2015, Chapter 7.1.1) it holds

$$\widehat{u}_d^{\text{BLUP}} = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_\varepsilon^2 n_d^{-1}} \left(\widehat{\mu}_{Y,d} - \widehat{\mu}_{X,d} \widehat{\beta}^{\text{BLUE}} \right), \tag{4.17}$$

where

$$\widehat{\mu}_{Y,d} := \frac{1}{n_d} \sum_{i \in \mathcal{S}_d} y_{i,d} \quad \text{and} \quad \widehat{\mu}_{X,d} := \frac{1}{n_d} \sum_{i \in \mathcal{S}_d} x_{i,d} \tag{4.18}$$

denote the area-specific sample means of the dependent variable and the auxiliary information, respectively. Defining

$$\gamma_d := \frac{\sigma_u^2}{\sigma_u^2 + \sigma_\varepsilon^2 n_d^{-1}} \in [0, 1] \tag{4.19}$$

yields the more common composite form of the BHF-estimator for the area-specific mean as

$$\begin{aligned} \widehat{\mu}_d^{\text{BHF}} &= \mu_{X,d}^T \widehat{\beta}^{\text{BLUE}} + \gamma_d \left(\widehat{\mu}_{Y,d} - \widehat{\mu}_{X,d}^T \widehat{\beta}^{\text{BLUE}} \right) \\ &= \gamma_d \left(\widehat{\mu}_{Y,d} + (\mu_{X,d} - \widehat{\mu}_{X,d})^T \, \widehat{\beta}^{\text{BLUE}} \right) + (1 - \gamma_d) \widehat{\mu}_{X,d}^T \widehat{\beta}^{\text{BLUE}}. \end{aligned} \tag{4.20}$$

The synthetic component is referred to as synthetic regression estimator, whereas the utilized direct part of the BHF-estimator is the multilevel GREG-estimator, i.e. the GREG-estimator (2.15) where the regression coefficients are determined on the population level (cf. Münnich et al., 2013).

P-Spline Linear Mixed Model

The basic unit level model (4.11) relies on the assumption of a linear relationship between the variable of interest and the covariates. This assumption, however, is rarely satisfied in practice. As a case in point, the Rhineland-Palatinate forest inventory sample data, displayed in Figure 1.1, show that a (generalized) linear relationship fits these data only poorly. Therefore, it is advantageous to allow for more flexible dependencies in the small area model. Recently, Opsomer et al. (2008) and Ugarte et al. (2009) proposed to consider the model

$$y_{i,d} = s(x_{i,d}) + u_d + \varepsilon_{i,d}, \;\; i \in \mathcal{S}, \tag{4.21}$$

where, as in the linear model, the area-specific random effects $u_d \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_u^2)$ are assumed to be independent of the unit-specific random errors $\varepsilon_{i,d} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$. Instead of assuming a linear regression function, the authors only demand s to be sufficiently smooth without further specification. The authors apply the penalized spline method (cf. Section 3.2) to represent the regression function s, that is

$$s := \sum_{k=1}^{K} \widehat{\alpha}_k \phi_{k,q}, \tag{4.22}$$

where $\hat{\alpha}$ is given as the unique solution of the optimization problem (3.57), i.e.

$$\min_{\alpha \in \mathbb{R}^K} \|\Phi\alpha - y\|_2^2 + \lambda \alpha^T \Lambda \alpha. \tag{4.23}$$

In this context, Opsomer et al. (2008) suggest the truncated power series basis based on 35 to 50 quantile placed knots with the related coefficient penalty. Ugarte et al. (2009) extend this idea to the numerically more stable B-spline basis based on $\min\{\lfloor n/4 \rfloor, 40\}$ equally spaced knots with second order difference penalty. Both of these P-spline variants have been presented in Subsection 3.2.2. As shown in Subsection 3.2.3, fixing $\lambda := \sigma_\varepsilon^2/\sigma_\gamma^2$ allows to determine the solution to the above optimization problem from the LMM (3.71), that is

$$y = X\beta + Z\gamma + \varepsilon, \;\; \begin{pmatrix} \gamma \\ \varepsilon \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \sigma_\gamma^2 I_r & 0 \\ 0 & \sigma_\varepsilon^2 I_n \end{bmatrix}\right). \tag{4.24}$$

The P-spline small area model (4.21) therefore reads

$$y = X\beta + [Z, W] \begin{pmatrix} \gamma \\ u \end{pmatrix} + \varepsilon, \;\; \begin{pmatrix} \gamma \\ u \\ \varepsilon \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \sigma_\gamma^2 I_r & 0 & 0 \\ 0 & \sigma_u^2 I_D & 0 \\ 0 & 0 & \sigma_\varepsilon^2 I_n \end{bmatrix}\right), \tag{4.25}$$

where W is defined as in (4.13). The BLUP of the area-specific mean values is given as

$$\widehat{\mu}_{Y,d}^{\text{BLUP}} := \bar{X}_d \widehat{\beta}^{\text{BLUE}} + \bar{Z}_d \widehat{\gamma}^{\text{BLUP}} + \widehat{u}_d^{\text{BLUP}}, \tag{4.26}$$

where

$$\bar{X}_d := \frac{1}{N_d} \sum_{i \in \mathcal{U}_d} X[i, \cdot] \;\; \text{and} \;\; \bar{Z}_d := \frac{1}{N_d} \sum_{i \in \mathcal{U}_d} Z[i, \cdot]. \tag{4.27}$$

The estimation of further area-specific target parameters, however, is achieved in analogy to the BHF-estimator. Let therefore

$$\widehat{\alpha} := \tilde{X}\widehat{\beta}^{\mathrm{BLUE}} + \tilde{Z}\widehat{\gamma}^{\mathrm{BLUP}} \tag{4.28}$$

denote the spline coefficients of the underlying P-spline function (cf. Subsection 3.2.3) and let

$$\widehat{y}_{i,d} := \widehat{s}(x_{i,d}) + \widehat{u}_d^{\mathrm{BLUP}}, \; i \in \mathcal{U}, \tag{4.29}$$

denote the related model predictions for the entire population. This allows for the definition of a spline-based small area estimator on the unit-level.

Definition 4.1.4 (SLMM-estimator)
We define the P-spline linear mixed model (SLMM) estimator of the area-specific target parameters θ_d, $d = 1, \dots, D$, as

$$\widehat{\theta}_d^{\mathrm{SLMM}} := f(\widehat{y}_{i,d} : i \in \mathcal{U}_d). \tag{4.30}$$

Discussion of the SLMM-Estimator

The utilization of the penalized spline method in small area estimation, as proposed by Opsomer et al. (2008) and Ugarte et al. (2009), offers numerous considerable advantages compared to the common small area models based on (generalized) linear regression functions. Due to the underlying spline approximation, they allow for the consideration of very complex and highly nonlinear relationships in the observed data, especially when no assumption on the specific form of the regression function seems legit. Further, the representation as a linear mixed model allows to embed the SLMM-estimator into the common setting of small area estimation. Thereby, the well established mixed model theory and related software can be utilized. Moreover, the mixed model implementation directly yields the regularization parameter $\lambda := \sigma_\varepsilon^2/\sigma_\gamma^2$ (cf. Subsection 3.2.4) as ratio of the respective variances.

Even though the SLMM-estimator comes with several improvements compared to small area estimation using a (generalized) linear regression function, particular challenges and potential enhancements remain. The sample data of the subpopulations considered in the context of SAE are typically small and therefore often only insufficiently display the general trend within the target population. Therefore, the incorporation of further shape constraints on the P-spline function could lead to a more realistic representation of the data as soon as there exists reasonable knowledge about the considered variable, e.g. that there are only nonnegative values as for the estimation of timber volume in RLP (cf. Section 1.1). On the one hand, the incorporation of shape constraints into the penalized spline method is theoretically and numerically straightforward (cf. Section 3.3). On the other hand, this extension prevents from the linear mixed model representation of the related P-spline problem. Hence, in order to incorporate shape constraints into a P-spline based small area model, an alternative formulation and solution strategy is necessary. The need of developing an alternative problem formulation bears the potential to circumvent two further drawbacks of the SLMM-estimator. Firstly, for the parameter estimation in the linear mixed model relatively strict assumptions are made with respect to the random parts of the model, i.e. $u \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_u^2 I_D)$ and $\varepsilon \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2 I_n)$. These assumptions are necessary to derive the estimates from a linear mixed model, but could be circumvented in a different solution strategy. Secondly, the regularization parameter $\lambda := \sigma_\varepsilon^2 / \sigma_\gamma^2$ is fixed in the context of the SLMM-estimator due to the assumptions made on the random parts of the model and cannot be controlled by the user. This can both lead to an inappropriate fit of the underlying P-spline function due to an inadequate choice of the regularization parameter λ. In order to address all of these issues, we develop an alternative approach to determine the parameters of a P-spline small area model in the following.

4.2 Optimization Framework for Penalized Spline based Small Area Models

The previous section shows that small area estimation based on P-splines comes with several improvements, but that several challenges and potential enhancements remain. Again, these are in particular:

1. The relaxation of the strict model assumptions.
2. An autonomous control of the regularization parameter.
3. The incorporation of shape constraints on the P-spline function.

To achieve these goals, we develop an alternative approach to determine the parameters of a P-spline based small area model, namely the spline coefficients $\alpha \in \mathbb{R}^K$ and the vector of the area-specific intercepts $u \in \mathbb{R}^D$. This alternative approach is based on the formulation of an optimization problem for these parameters instead of a linear mixed model and is discussed in the following.

4.2.1 Formulation of the Optimization Problem

We consider the small area model

$$y_{i,d} = s(x_{i,d}) + u_d + \varepsilon_{i,d}, \; i \in \mathcal{S}, \tag{4.31}$$

with zero mean unit-specific random errors $\varepsilon_{i,d}$, zero mean area-specific random effects u_d, and sufficiently smooth but further unspecified regression function $s\colon \mathbb{R}^P \to \mathbb{R}$. Note that, compared to the model (4.21), no further distributional assumptions are made. We model the regression function as a regression P-spline, that is as solution of

$$\min_{s \in \mathcal{S}_q(\mathcal{K})} \mathcal{LS}(s) + \lambda \mathcal{P}(s) \;\Leftrightarrow\; \min_{s \in \mathcal{S}_q(\mathcal{K})} \sum_{i \in \mathcal{S}} \left(s(x_{i,d}) - y_{i,d}\right)^2 + \lambda \mathcal{P}(s), \tag{4.32}$$

and incorporate the area-specific intercepts u_d, $d = 1, \ldots, D$, as a further variable. Therefore, we apply a ridge regression approach on the intercept vector u to simultaneously determine the area-specific effects and the P-spline function. That is, we consider the optimization problem

$$\min_{s \in \mathcal{S}_q(\mathcal{K}), u \in \mathbb{R}^D} \sum_{i \in \mathcal{S}} \left(s(x_{i,d}) + u_d - y_{i,d}\right)^2 + \lambda_s \mathcal{P}(s) + \lambda_u \|u\|_2^2. \tag{4.33}$$

The regularization parameter $\lambda_s > 0$ still controls for the smoothness of the spline function. The regularization of u is necessary to provide the existence of a unique solution (cf. Theorem 4.2.2). The further regularization parameter $\lambda_u > 0$ allows to regulate the influence of the both regularization terms independently. The choice of the multiple regularization parameters is addressed in Subsection 4.2.3. In order to distinguish between the regression P-spline defined by (3.39) and the P-spline defined by the above optimization problem, we state the following definition.

Definition 4.2.1 (Small area P-spline)
For a given spline space $\mathcal{S}_q(\mathcal{K})$ let $\widehat{s} \in \mathcal{S}_q(\mathcal{K})$ and $\widehat{u} \in \mathbb{R}^D$ denote a solution of the optimization problem (4.33). We refer to $\widehat{s}$ as a small area P-spline in $\mathcal{S}_q(\mathcal{K})$.

Due to the basis representation of a tensor product spline (3.26), we obtain an equivalent formulation of the optimization problem (4.33) in terms of the spline coefficients and the area-specific intercepts as

$$\min_{\alpha \in \mathbb{R}^K, u \in \mathbb{R}^D} \|\Phi\alpha + Wu - y\|_2^2 + \lambda_s \alpha^T \Lambda \alpha + \lambda_u \|u\|_2^2, \tag{4.34}$$

where W is defined as in (4.13). It holds

$$\begin{aligned} &\|\Phi\alpha + Wu - y\|_2^2 + \lambda_s \alpha^T \Lambda \alpha + \lambda_u \|u\|_2^2 \\ &= \alpha^T \Phi^T \Phi \alpha + u^T W^T W u + y^T y + \alpha^T \Phi^T W u + u^T W^T \Phi \alpha - 2\alpha^T \Phi^T y \\ &\quad - 2u^T W^T y + \lambda_s \alpha^T \Lambda \alpha + \lambda_u u^T u \\ &= (\alpha^T, u^T) \begin{bmatrix} \Phi^T \Phi + \lambda_s \Lambda & \Phi^T W \\ W^T \Phi & W^T W + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} - 2 \begin{bmatrix} y^T \Phi & y^T W \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} \end{aligned} \tag{4.35}$$

such that the optimization problem (4.34) is equivalent to

$$\min_{\alpha \in \mathbb{R}^K, u \in \mathbb{R}^D} \frac{1}{2} (\alpha^T, u^T) \begin{bmatrix} \Phi^T \Phi + \lambda_s \Lambda & \Phi^T W \\ W^T \Phi & W^T W + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} - \begin{bmatrix} y^T \Phi & y^T W \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix}. \tag{4.36}$$

Based on this reformulation, we state an existence and uniqueness result for the optimization problem (4.33) in the following theorem.

Theorem 4.2.2
The optimization problem (4.33) possesses a unique solution. It is given by the unique solution of the linear system

$$\begin{bmatrix} \Phi^T \Phi + \lambda_s \Lambda & \Phi^T W \\ W^T \Phi & W' W + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} \overset{!}{=} \begin{bmatrix} \Phi^T \\ W^T \end{bmatrix} y. \tag{4.37}$$

Proof. To prove the statement, we show that the symmetric matrix

$$A := \begin{bmatrix} \Phi^T \Phi + \lambda_s \Lambda & \Phi^T W \\ W^T \Phi & W^T W + \lambda_u I_D \end{bmatrix} \in \mathbb{R}^{(K+D)\times(K+D)},$$

is positive definite such that the equivalent optimization problem (4.36) is an unconstrained strictly convex QP. Remark 2.5.3 then ensures the existence of a unique solution given by the unique solution of the linear system (4.37). The Schur complement condition for positive definiteness states that $A \succ 0$ holds if and only if:

1. $\Phi^T \Phi + \lambda_s \Lambda \succ 0$.
2. $W^T W + \lambda_u I_D \succ 0$.
3. $M := \Phi^T \Phi + \lambda_s \Lambda - \Phi^T W \left(W^T W + \lambda_u I_D \right)^{-1} W^T \Phi \succ 0$.

The positive definiteness of $\Phi^T \Phi + \lambda_s \Lambda$ is stated in Remark 3.2.5. Since

$$W^T W = \operatorname{diag}(n_1, \dots, n_D)$$

and $\lambda_u > 0$ it also follows $W^T W + \lambda_u I_D \succ 0$. In order to show the positive definiteness of M, we define

$$C := W \left(W^T W + \lambda_u I_D \right)^{-1} W^T \in \mathbb{R}^{n \times n}. \tag{4.38}$$

For arbitrary $0 \neq \alpha \in \mathbb{R}^K$ and $v := \Phi \alpha \in \mathbb{R}^n$ it holds

$$\alpha^T M \alpha = \|v\|_2^2 + \lambda_s \alpha^T \Lambda \alpha - v^T C v.$$

Since $\alpha^T \Lambda \alpha \geq 0$, it suffices to show $v^T C v < \|v\|_2^2$ to obtain the positive definiteness of M. First, note that it holds

$$\left(W^T W + \lambda_u I_D \right)^{-1} = \operatorname{diag}\left((n_1 + \lambda_u)^{-1}, \dots, (n_D + \lambda_u)^{-1} \right)$$

such that

$$C[i,j] = \begin{cases} (n_d + \lambda_u)^{-1} & , \text{ if } i,j \in \mathcal{S}_d \\ 0 & , \text{ else} \end{cases} , \quad i,j = 1, \dots, n,$$

follows from the definition of W. This yields

$$v^T C v = \sum_{i=1}^{n} \sum_{j=1}^{n} C[i,j] v[i] v[j] = \sum_{d=1}^{D} \left(\sum_{i \in \mathcal{S}_d} \sum_{j \in \mathcal{S}_d} (n_d + \lambda_u)^{-1} v[i] v[j] \right)$$

and the multinomial theorem implies

$$v^T C v = \sum_{d=1}^{D} (n_d + \lambda_u)^{-1} \left(\sum_{i \in \mathcal{S}_d} v[i] \right)^2 .$$

Due to the Cauchy-Schwarz inequality in $\mathbb{R}^{n_d}$ for $d = 1, \dots, D$ it follows

$$v^T C v \leq \sum_{d=1}^{D} (n_d + \lambda_u)^{-1} \left(n_d \sum_{i \in \mathcal{S}_d} v[i]^2 \right) .$$

Finally, because of $\lambda_u > 0$ it holds

$$v^T C v < \sum_{d=1}^{D} \sum_{i \in \mathcal{S}_d} v[i]^2 = \sum_{i \in \mathcal{S}} v[i]^2 = \|v\|_2^2,$$

which concludes the proof. □

Remark that in the previous proof we obtain the positive definiteness of the coefficient matrix of the linear system (4.37) from the fact that

$$\frac{n_d}{n_d + \lambda_u} < 1, \;\; d = 1, \ldots, D, \tag{4.39}$$

which is due to $\lambda_u > 0$. Therefore, as already mentioned, the regularization of u in the optimization problem (4.33) is necessary in order to obtain a well-posed problem.

4.2.2 Derivation of the Small Area Estimator

The SLMM-estimator (4.30) is based on the reformulation of the P-spline problem as a linear mixed model and the resulting determination of the parameters. In the previous, an alternative approach to determine the model parameters is presented based on the formulation as the optimization problem (4.34). As soon as the parameters $\widehat{\alpha}$ and $\widehat{u}$ are determined as the unique solution of the linear system (4.37), small area estimates can be obtained in analogy to the SLMM-estimator. Let therefore

$$\widehat{y}_{i,d} := \widehat{s}(x_{i,d}) + \widehat{u}_d, \;\; i \in \mathcal{U}, \tag{4.40}$$

denote the model predictions for the entire population, where

$$\widehat{s} := \sum_{k=1}^{K} \widehat{\alpha}_k \phi_{k,q} \tag{4.41}$$

denotes the small area P-spline related to $\widehat{\alpha}$. This allows for a further definition of a spline-based small area estimator on the unit-level.

Definition 4.2.3 (SOPT-estimator)
We define the P-spline optimization problem (SOPT) estimator of the area-specific target parameters θ_d, $d = 1, \ldots, D$, as

$$\widehat{\theta}_d^{\mathrm{SOPT}} := f(\widehat{y}_{i,d} : i \in \mathcal{U}_d). \tag{4.42}$$

The SOPT-estimator (4.42) and the SLMM-estimator (4.30) both utilize P-splines, but differ within the derivation of the model parameters. The SOPT determines the model parameters via an optimization problem, whereas the SLMM employs a linear mixed model. Despite the fact that the underlying spline models are similar, the resulting estimators possess several different features. The SLMM-estimator requires the strong statistical assumptions of $u \sim \mathcal{N}(0, \sigma_u^2 I_D)$ and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_n)$ in order to determine the model parameters from a LMM. Further, the

SLMM is restricted to the regularization parameter $\lambda = \sigma_\varepsilon^2/\sigma_\gamma^2$. Thus, an inadequate choice of λ can result. Due to the optimization approach of the SOPT-estimator, these issues are circumvented. In particular, the SOPT-estimator allows for an autonomous control of the regularization parameter λ_s by the user, which can result in a more realistic P-spline function and therefore in more precise small area estimates. With an identical choice of the regularization parameters $\lambda_s = \lambda$, however, the both estimators provide quite similar small area estimates. Based on the mixed model representation of the SLMM-estimator, statistical features such as the BLUP property can be obtained. For the SOPT-estimator such properties cannot be proven. Nevertheless, the optimization problem formulation provides further important numerical features. It enables the straightforward incorporation of shape constraints into the small area model, which is discussed in more detail in Section 4.3. Further, this framework allows for the utilization of numerically highly advanced methods in order to determine the model parameters, which is of special interest if the number of utilized covariates P increases. The implementation of these algorithms is addressed in Chapter 5.

4.2.3 Regularization Parameter Selection

Compared to the regresion P-spline, the small area P-spline now requires the determination of the both regularization parameters $\lambda_s > 0$ and $\lambda_u > 0$. In the following, we propose some approaches based on the methods introduced for regularization parameter selection for the regression P-splines (cf. Subsection 3.2.4). Recall first that the small area P-spline is defined by the unique solution of the linear system (4.37), that is

$$\begin{pmatrix} \widehat{\alpha} \\ \widehat{u} \end{pmatrix} := \left(\begin{bmatrix} \Phi^T\Phi + \lambda_s\Lambda & \Phi^T W \\ W^T\Phi & W^T W + \lambda_u I_D \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi^T \\ W^T \end{bmatrix} y. \tag{4.43}$$

Simultaneous Parameter Selection

For the simultaneous determination of λ_s and λ_u, we define $\lambda := (\lambda_s, \lambda_u)^T$ and let, analogously to (3.74),

$$S_\lambda := [\Phi, W] \left(\begin{bmatrix} \Phi^T\Phi + \lambda_s\Lambda & \Phi^T W \\ W^T\Phi & W^T W + \lambda_u I_D \end{bmatrix} \right)^{-1} \begin{bmatrix} \Phi^T \\ W^T \end{bmatrix} \tag{4.44}$$

denote the hat matrix related to (4.37) with the resulting model predicts $\widehat{y}_\lambda := S_\lambda y$. The extension of the CV and the GCV method to the case of multiple regularization parameters is straightforward, i.e.

$$\lambda_{\text{CV}} := \underset{\lambda_s, \lambda_u > 0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_\lambda[i]}{1 - S_\lambda[i,i]} \right)^2, \; \lambda_{\text{GCV}} := \underset{\lambda_s, \lambda_u > 0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_\lambda[i]}{1 - \operatorname{tr}(S_\lambda)} \right)^2. \tag{4.45}$$

Although the extension of these methods is straightforward, the simultaneous determination of multiple regularization parameters leads to significantly increasing computational demands since a two-dimensional grid search has to be performed. Therefore, it might be useful to follow a different approach.

Sequential Parameter Selection

The regularization parameter λ_s controls for the smoothness of the underlying P-spline function, whereas the regularization parameter λ_u is mainly introduced to ensure for a unique solution. In order to obtain a suitable spline approximation, we propose to determine the parameter λ_s at first by means of the methods introduced in Subsection 3.2.4, for example $\lambda_s = \lambda_{s,\mathrm{CV}}$. Based on this fixed parameter, we then determine the remaining parameter λ_u. There, the most simplest choice is an identical parameter selection, i.e. $\lambda_u = \lambda_s$. For some applications, however, this approach can result in an inappropriate choice of the small area intercept u. Since λ_s is now fixed, the hat matrix solely depends on λ_u and the CV and the GCV method simplify to

$$\lambda_{u,\mathrm{CV}} := \underset{\lambda_u > 0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_{\lambda_u}[i]}{1 - S_{\lambda_u}[i,i]} \right)^2 , \ \lambda_{u,\mathrm{GCV}} := \underset{\lambda_u > 0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \widehat{y}_{\lambda_u}[i]}{1 - \mathrm{trace}(S_{\lambda_u})} \right)^2 . \qquad (4.46)$$

The determination of the both regularization parameters within the sequential selection approach demands for the solution of two one-dimensional optimization problems, each by a grid search. This requires by far less effort than the two-dimensional grid search utilized for the simultaneous parameter selection.

4.3 Penalized Spline based Small Area Models with Shape Constraints

In the previous section, an optimization framework in order to determine the parameters of a P-spline based small area model is introduced. An important feature of this approach is the fact that the incorporation of shape constraints on the small area P-spline is as straightforward as for the regression P-spline presented in Section 3.3.

4.3.1 Incorporation of the Shape Constraints

The determination of the model parameters for a P-spline based small area model as a solution of an optimization problem (4.33) allows for a straightforward incorporation of shape constraints on the small area P-spline function. This is due to the fact that these constraints solely affect the spline function s, but not the area-specific intercepts u_d, $d = 1, \dots, D$. We therefore determine s and u simultaneously as a solution of the optimization problem

$$\begin{aligned} \min_{s \in \mathcal{S}_q(\mathcal{K}), u \in \mathbb{R}^D} \quad & \sum_{i \in \mathcal{S}} (s(x_{i,d}) + u_d - y_{i,d})^2 + \lambda_s \mathcal{P}(s) + \lambda_u \|u\|_2^2 \\ \text{s.t.} \quad & \partial^r s \geq 0, \ r \in I_{\geq} \\ & \partial^r s \leq 0, \ r \in I_{\leq}, \end{aligned} \qquad (4.47)$$

where the index sets $I_{\geq}$ and $I_{\leq}$ are defined as in (3.87).

Definition 4.3.1 (Shape constrained small area P-spline)
For a given spline space $\mathcal{S}_q(\mathcal{K})$ let $\widehat{s} \in \mathcal{S}_q(\mathcal{K})$ and $\widehat{u} \in \mathbb{R}^D$ denote a solution of the optimization problem (4.47). We refer to $\widehat{s}$ as a shape constrained small area P-spline in $\mathcal{S}_q(\mathcal{K})$.

As for the unconstrained case, we obtain an equivalent formulation of the optimization problem (4.47) as

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^K, u\in\mathbb{R}^D} \quad & \|\Phi\alpha + Wu - y\|_2^2 + \lambda_s \alpha^T \Lambda \alpha + \lambda_u \|u\|_2^2 \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0,\ \ r \in I_{\leq} \\ & \Gamma_r \alpha \geq 0,\ \ r \in I_{\geq}, \end{aligned} \tag{4.48}$$

where the matrices $\Gamma_r \in \mathbb{R}^{T\times K}$ are defined as in (3.91). As for the unconstrained optimization problem (4.34), we obtain an equivalent formulation of the optimization problem (4.48) as

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^K, u\in\mathbb{R}^D} \quad & \frac{1}{2}(\alpha^T, u^T) \begin{bmatrix} \Phi^T\Phi + \lambda_s\Lambda & \Phi^T W \\ W^T\Phi & W^T W + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} - \begin{bmatrix} y^T\Phi & y^T W \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0,\ \ r \in I_{\leq} \\ & \Gamma_r \alpha \geq 0,\ \ r \in I_{\geq}. \end{aligned} \tag{4.49}$$

Due to Theorem 4.2.2, this is a strictly convex QP and we state an existence and uniqueness result for the optimization problem (4.47) in the following theorem based on the reformulation (4.49).

Theorem 4.3.2
The optimization problem (4.47) possesses a unique solution.

Proof. To prove the statement, we consider the equivalent reformulation as strictly convex QP (4.49). Remark 2.5.3 already yields that the solution is unique, provided it exists. In order to show the existence of a solution it suffices to show that the optimal value f^* of the strictly convex QP (4.49) is finite (cf. Theorem 2.5.4). Since 0 is a feasible point of the strictly convex QP, it holds $f^* \leq 0 < \infty$. Let $\tilde{f}$ denote the optimal value of the unconstrained strictly convex QP (4.36) such that $\tilde{f} \leq f^*$ holds. Due to Theorem 4.2.2, it holds $-\infty < \tilde{f}$, which concludes the proof. □

4.3.2 Derivation of the Small Area Estimator

In analogy to the SOPT-estimator (4.42), we define a small area estimator based on the shape constrained small area P-spline. Let therefore $\widehat{\alpha}$ and $\widehat{u}$ denote the unique solution of the strictly convex QP (4.49) and let

$$\widehat{y}_{i,d} := \widehat{s}(x_{i,d}) + \widehat{u}_d,\ \ i \in \mathcal{U}, \tag{4.50}$$

denote the related model predictions for the entire population, where

$$\widehat{s} := \sum_{k=1}^{K} \widehat{\alpha}_k \phi_{k,q} \tag{4.51}$$

denotes the shape constrained small area P-spline related to $\widehat{\alpha}$. As for the SOPT-estimator, we use these predicts to obtain a small area estimator. This leads to the definition of a spline-based small area estimator on the unit-level that further allows the consideration of shape constraints.

Definition 4.3.3 (SOPT_CON-estimator)
We define the shape constrained P-spline optimization problem (SOPT_CON) estimator of the area-specific target parameters θ_d, $d = 1, \ldots, D$, as

$$\widehat{\theta}_d^{\text{SOPT\_CON}} := f(\widehat{y}_{i,d} : i \in \mathcal{U}_d). \tag{4.52}$$

Compared to the SOPT-estimator (4.42), the SOPT_CON-estimator (4.52) additionally allows for the incorporation of arbitrary shape constraints on the underlying small area P-spline. This enables a more realistic modeling of the relationship within the sample data and can result in more precise small area estimates. Especially in small area estimation, where the sample sizes are frequently too small to reflect the general trend within the population, the shape constraints provide a promising feature. If the data at hand are rich enough to represent the underlying shapes, the small area P-spline of the SOPT-estimator and the shape constrained small area P-spline of the SOPT_CON-estimator coincide. Therefore, the both estimators provide the same small area estimates in this case. From a numerical point of view, however, they lead to different problems, namely a linear system (4.37) and a strictly convex QP (4.49). The latter one is much more expensive to solve (cf. Chapter 5) such that the incorporation of shape constraints should be conducted with deliberation.

4.4 Linkage to Penalized Splines in Regression Analysis

For the special case of $u = 0$, the small area model

$$y_{i,d} = s(x_{i,d}) + u_d + \varepsilon_{i,d},\ \ i \in \mathcal{S}, \tag{4.53}$$

with zero mean unit-specific random errors $\varepsilon_{i,d}$ corresponds with the regression model

$$y_i = s(x_i) + \varepsilon_i,\ \ i = 1, \ldots, n \tag{4.54}$$

with zero mean unit-specific random errors ε_i. Therefore, in this case the (shape constrained) regression P-spline and the (shape constrained) small area P-spline coincide. Thus, the (shape constrained) regression P-spline is a special case of the (shape constrained) small area P-spline and therefore the related optimization problems (3.58) and (3.93) are special cases of the optimization problems (4.37) and (4.48). This relationship is revealed in the following and is of special interest for the implementation of numerical efficient solution algorithms presented in Chapter 5.

Unconstrained P-Splines

The linear system (4.37) underlying the determination of the coefficients of the (unconstrained) small area P-spline reads

$$\begin{bmatrix} \Phi^T\Phi + \lambda_s\Lambda & \Phi^T W \\ W^T\Phi & W^T W + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ u \end{pmatrix} \overset{!}{=} \begin{bmatrix} \Phi^T \\ W^T \end{bmatrix} y. \tag{4.55}$$

The following lemma states a characterization of the unique solution of this linear system.

Lemma 4.4.1
The unique solution of the linear system (4.37) is given by

$$\widehat{u} := \operatorname{diag}\left((n_1+\lambda_u)^{-1}, \dots, (n_D+\lambda_u)^{-1}\right) W^T \left(y - \Phi\widehat{\alpha}\right), \tag{4.56}$$

where $\widehat{\alpha}$ denotes the unique solution of the linear system

$$\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\alpha \overset{!}{=} \Phi^T\left(y - Cy\right) \tag{4.57}$$

with $C \in \mathbb{R}^{n\times n}$ defined as in (4.38).

Proof. The linear system (4.37) is equivalent to the both coupled linear systems

$$\begin{aligned} \left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha + \Phi^T W u &\overset{!}{=} \Phi^T y, &(4.58)\\ W^T\Phi\alpha + \left(W^T W + \lambda_u I_D\right) u &\overset{!}{=} W^T y. &(4.59)\end{aligned}$$

For arbitrary u let

$$\widehat{\alpha}(u) := \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T(y - Wu) \tag{4.60}$$

denote the unique solution of (4.58), which exists due to Remark 3.2.5. Since

$$W^T W + \lambda_u I_D = \operatorname{diag}\left((n_1+\lambda_u)^{-1}, \dots, (n_D+\lambda_u)^{-1}\right) \succ 0,$$

we obtain

$$\begin{aligned} \widehat{u} &= \left(W^T W + \lambda_u I_D\right)^{-1} W^T\left(y - \Phi\widehat{\alpha}(u)\right)\\ &= \operatorname{diag}\left((n_1+\lambda_u)^{-1}, \dots, (n_D+\lambda_u)^{-1}\right) W^T\left(y - \Phi\widehat{\alpha}(u)\right)\end{aligned}$$

as unique solution of (4.59), depending on $\widehat{\alpha}(u)$. Plugging $\widehat{u}$ into (4.60) and defining $\widehat{\alpha} := \widehat{\alpha}(\widehat{u})$ yields

$$\begin{aligned} \widehat{\alpha} &= \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T y - \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T W\left(W^T W + \lambda_u I_D\right)^{-1} W^T\left(y - \Phi\widehat{\alpha}\right)\\ &= \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T y - \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T C\left(y - \Phi\widehat{\alpha}\right) &(4.61)\\ &= \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T(y - Cy) + \left(\Phi^T\Phi + \lambda_s\Lambda\right)^{-1}\Phi^T C\Phi\widehat{\alpha}.\end{aligned}$$

Multiplying (4.61) by $\left(\Phi^T\Phi + \lambda_s\Lambda\right)$ and resorting the terms finally yields

$$\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\widehat{\alpha} = \Phi^T\left(y - Cy\right),$$

i.e. $\widehat{\alpha}$ is a solution of the linear system (4.57). As shown in the proof of Theorem 4.2.2 it holds $\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi \succ 0$, such that the solution $\widehat{\alpha}$ is unique. □

Lemma 4.4.1 especially yields that the spline coefficients of the small area P-spline are given by the unique solution of the linear system (4.57), i.e.

$$\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\alpha \overset{!}{=} \Phi^T\left(y - Cy\right). \tag{4.62}$$

For the special case of $C = 0$ the linear system (4.57) simplifies to

$$\left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha \overset{!}{=} \Phi^T y \tag{4.63}$$

which coincides with the linear system (3.58) determining the spline coefficients of the (unconstrained) regression P-spline.

Shape Constrained P-Splines

The determination of the shape constrained small area P-spline requires the solution of the strictly convex QP (4.49). Based on the reformulation for the linear system (4.37) to determine the unconstrained small area P-spline in Lemma 4.4.1, we state a characterization of the unique solution of the optimization problem (4.49) in the following lemma.

Lemma 4.4.2

The unique solution of the the strictly convex QP (4.49) is given by

$$\widehat{u} = \operatorname{diag}\left((n_1+\lambda_u)^{-1}, \ldots, (n_D+\lambda_u)^{-1}\right) W^T\left(y - \Phi\widehat{\alpha}\right), \tag{4.64}$$

where $\widehat{\alpha}$ denotes the unique solution of the strictly convex QP

$$\begin{aligned}
\min_{\alpha\in\mathbb{R}^K} \quad & \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\alpha - \left(\Phi^T(y - Cy)\right)^T\alpha \\
\text{s.t.} \quad & \Gamma_r\alpha \le 0,\ r \in I_{\le} \\
& \Gamma_r\alpha \ge 0,\ r \in I_{\ge}
\end{aligned} \tag{4.65}$$

with $C \in \mathbb{R}^{n\times n}$ defined as in (4.38).

Proof. Since the shape constraints solely affect the spline coefficients α, Lemma 4.4.1 yields

$$\widehat{u}(\alpha) = \operatorname{diag}\left((n_1+\lambda_u)^{-1}, \ldots, (n_D+\lambda_u)^{-1}\right) W^T\left(y - \Phi\alpha\right)$$

as optimal solution of (4.49) for arbitrary α. Plugging $\widehat{u}(\alpha)$ into the objective function of the strictly convex QP (4.49) yields

$$\begin{aligned}
&\frac{1}{2}(\alpha^T, \widehat{u}(\alpha)^T)\begin{bmatrix}\Phi^T\Phi + \lambda_s\Lambda & \Phi^T W \\ W^T\Phi & W^T W + \lambda_u I_D\end{bmatrix}\begin{pmatrix}\alpha \\ \widehat{u}(\alpha)\end{pmatrix} - \begin{bmatrix}y^T\Phi & y^T W\end{bmatrix}\begin{pmatrix}\alpha \\ \widehat{u}(\alpha)\end{pmatrix} \\
&= \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha + \alpha^T\Phi^T W\widehat{u}(\alpha) - y^T\Phi\alpha - y^T W\widehat{u}(\alpha) + \frac{1}{2}\widehat{u}(\alpha)^T\left(W^T W + \lambda_u I_D\right)\widehat{u}(\alpha) \\
&= \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha + \alpha^T\Phi^T C(y - \Phi\alpha) - y^T\Phi\alpha - y^T C(y - \Phi\alpha) \\
&\quad + \frac{1}{2}(y - \Phi\alpha)^T\left[C^T C + \lambda_u W\left(W^T W + \lambda_u I_D\right)^{-1}\left(W^T W + \lambda_u I_D\right)^{-1} W^T\right](y - \Phi\alpha).
\end{aligned}$$

Due to

$$\lambda_u \left(W^TW + \lambda_u I_D\right)^{-1} \left(W^TW + \lambda_u I_D\right)^{-1} = \operatorname{diag}\left(\frac{\lambda_u}{(n_1+\lambda_u)^2}, \ldots, \frac{\lambda_u}{(n_D+\lambda_u)^2}\right)$$

and

$$\begin{aligned} C^TC &= W\left(W^TW + \lambda_u I_D\right)^{-1} W^TW \left(W^TW + \lambda_u I_D\right)^{-1} W^T \\ &= W \operatorname{diag}\left(\frac{n_1}{(n_1+\lambda_u)^2}, \ldots, \frac{n_D}{(n_D+\lambda_u)^2}\right) W^T \end{aligned}$$

it holds

$$\begin{aligned} &C^TC + \lambda W \left(W^TW + \lambda_u I_D\right)^{-1} \left(W^TW + \lambda_u I_D\right)^{-1} W^T \\ =\ & W \left[\operatorname{diag}\left((n_1+\lambda_u)^{-1}, \ldots, (n_D+\lambda_u)^{-1}\right)\right] W^T \\ =\ & W\left(W^TW + \lambda_u I_D\right)^{-1} W^T \\ =\ & C. \end{aligned}$$

This yields

$$\begin{aligned} &\frac{1}{2}(\alpha^T, \widehat{u}(\alpha)^T) \begin{bmatrix} \Phi^T\Phi + \lambda_s\Lambda & \Phi^TW \\ W^T\Phi & W^TW + \lambda_u I_D \end{bmatrix} \begin{pmatrix} \alpha \\ \widehat{u}(\alpha) \end{pmatrix} - \begin{bmatrix} y^T\Phi & y^TW \end{bmatrix} \begin{pmatrix} \alpha \\ \widehat{u}(\alpha) \end{pmatrix} \\ =\ & \frac{1}{2}\alpha^T \left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha + \alpha^T\Phi^TC(y-\Phi\alpha) - y^T\Phi\alpha - y^TC(y-\Phi\alpha) + \frac{1}{2}(y-\Phi\alpha)^TC(y-\Phi\alpha) \\ =\ & \frac{1}{2}\alpha^T \left(\Phi^T\Phi + \lambda_s\Lambda\right)\alpha + \alpha^T\Phi^TCy - \alpha^T\Phi^TC\Phi\alpha - y^T\Phi\alpha - y^TCy + y^TC\Phi\alpha \\ &+ \frac{1}{2}y^TCy + \frac{1}{2}\alpha^T\Phi^TC\Phi\alpha - \alpha^T\Phi^TCy \\ =\ & \frac{1}{2}\alpha^T \left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^TC\Phi\right)\alpha + \alpha^T\Phi^TCy - \alpha^T\Phi^Ty - \frac{1}{2}y^TCy \\ =\ & \frac{1}{2}\alpha^T \left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^TC\Phi\right)\alpha + \alpha^T\Phi^T(Cy - y) - \frac{1}{2}y^TCy \end{aligned}$$

for the objective function of the strictly convex QP (4.49) such that (4.49) is equivalent to the optimization problem (4.65). As shown in the proof of Theorem 4.2.2 it holds $\Phi^T\Phi + \lambda_s\Lambda - \Phi^TC\Phi \succ 0$ such that (4.65) is a strictly convex QP itself. Finally, due to Theorem 4.3.2, the strictly convex QP (4.65) possesses a unique solution. □

Lemma 4.4.2 especially yields that the spline coefficients of the shape constrained small area P-spline are given by the unique solution of the strictly convex QP (4.65), i.e.

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^K} \quad & \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^TC\Phi\right)\alpha - \left(\Phi^T(y - Cy)\right)^T\alpha \\ \text{s.t.} \quad & \Gamma_r\alpha \leq 0,\ r \in I_\leq \\ & \Gamma_r\alpha \geq 0,\ r \in I_\geq. \end{aligned} \tag{4.66}$$

For the special case of $C = 0$ the strictly convex QP (4.65) simplifies to

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^K} \quad & \frac{1}{2} \alpha^T \left(\Phi^T \Phi + \lambda_s \Lambda \right) \alpha - \left(\Phi^T y \right)^T \alpha \\ \text{s.t.} \quad & \Gamma_r \alpha \leq 0, \;\; r \in I_{\leq} \\ & \Gamma_r \alpha \geq 0, \;\; r \in I_{\geq}, \end{aligned} \tag{4.67}$$

which coincides with the strictly convex QP (3.93) determining the spline coefficients of the shape constrained regression P-spline.

4.5 Mean Squared Error Estimation

As introduced in Section 2.1, the MSE of an estimator $\widehat{\theta}$ is defined as

$$\text{MSE}\left(\widehat{\theta}\right) := \mathbb{E}\left(\left[\widehat{\theta} - \theta\right]^2\right) = \text{VAR}(\widehat{\theta}) + \text{BIAS}(\widehat{\theta})^2 \tag{4.68}$$

and yields an adequate measure of precision. Since the exact value of the MSE depends on the unknown parameter θ, it can in general not be computed and has to be estimated form the sample $\mathcal{S}$ as well. For example, for the HT-estimator and the GREG-estimator the related precision estimators are given in (2.14) and (2.17), respectively. In practice, an estimator is only useful if a related MSE-estimator is stated, wherefore the development of an MSE estimation technique for the SOPT-estimator and the SOPT_CON-estimator is crucial.

Common precision estimators are based on Taylor linearization methods (cf. Särndal et al., 1992, Chapter 5.5), but for those complex estimators as the SOPT and the SOPT_CON no closed form MSE-estimator can be obtained. In this case, resampling methods such as the bootstrap or the jackknife (cf. Wu, 1986, for an overview) may aid in finding an appropriate precision estimate for the point estimators. The basic idea of the bootstrap method, initially proposed by Efron (1979) and Efron and Tibshirani (1993), is to resample a large number of subsamples with replacement out of the original sample and to determine the MSE of the point estimator based on the bootstrap samples conditioned on the original sample. According to Särndal et al. (1992, Chapter 11.6), a classical bootstrap procedure works as follows:

1. Using the sample $\mathcal{S}$, construct an artificial population $\mathcal{U}^{\text{boot}}$ that is assumed to mimic the unknown population $\mathcal{U}$. Frequently, $\mathcal{U}^{\text{boot}} := \mathcal{S}$ is used.
2. Draw a series of $B \in \mathbb{N}$ independent subsamples $\mathcal{S}^b \subset \mathcal{U}^{\text{boot}}$, $b = 1, \ldots, B$, by a design identical to the one by which $\mathcal{S}$ was drawn from $\mathcal{U}$.
3. For each bootstrap repetition, compute the bootstrap estimate $\widehat{\theta}_b$ from the bootstrap sample $\mathcal{S}^b$ in the same way as $\widehat{\theta}$ was calculated from the original sample $\mathcal{S}$.
4. Compute the bootstrap MSE-estimator

$$\widehat{MSE}(\widehat{\theta}) := \frac{1}{B} \sum_{b=1}^{B} \left(\widehat{\theta} - \widehat{\theta}^b \right)^2, \tag{4.69}$$

which coincides with the empirical MSE of the bootstrap estimates $\widehat{\theta}_b$, $b = 1, \ldots, B$.

Crucial for the performance and the applicability of a bootstrap MSE-estimator is the procedure under which the bootstrap samples $\mathcal{S}^b$ are generated. Depending on the construction of the point estimator $\widehat{\theta}$ and the structure of the underlying data, a variety of bootstrap sample strategies exists. In order to estimate the MSE of the spline-based small area estimators SLMM, SOPT, and SOPT_CON, we recall that the estimates are obtained from the model predictions

$$\widehat{y}_{i,d} := \widehat{s}(x_{i,d}) + \widehat{u}_d, \;\; i \in \mathcal{U}, \tag{4.70}$$

where $\widehat{s}$ denotes the underlying P-spline. The three methods mainly differ in their strategy to determine the function $\widehat{s}$. Let

$$\widehat{\varepsilon}_{i,d} := \widehat{y}_{i,d} - y_{i,d}, \;\; i \in \mathcal{S}, \tag{4.71}$$

denote the residuals related to model predicts. Since the spline function $\widehat{s}$ is considered as fix, the random parts of the estimator that has to be reflected by the bootstrap sample are the area effects $\widehat{u}_d$, $d = 1, \dots, D$ and the unit-specific errors $\widehat{\varepsilon}_{i,d}$, $i \in \mathcal{S}$. According to Kauermann et al. (2009), we implement a wild bootstrap to draw the bootstrap errors $\varepsilon^b := (\varepsilon^b_1, \dots, \varepsilon^b_n)^T$, that is

$$\varepsilon^b_{i,d} := w^b_{i,d} \cdot \widehat{\varepsilon}_{i,d}, \quad w^b_{i,d} := \begin{cases} (1-\sqrt{5})/2 & \text{with probability} (\sqrt{5}+1)/(2\sqrt{5}) \\ (1+\sqrt{5})/2 & \text{with probability} (\sqrt{5}-1)/(2\sqrt{5}) \end{cases}. \tag{4.72}$$

The wild bootstrap is in particular suited when the model exhibits heteroskedasticity. Since no parametric assumptions on the distribution of the area effects are made, we utilize a non-parametric bootstrap to draw the bootstrap sample $u^b := (u^b_1, \dots, u^b_D)^T$, that is u^b is obtained by sampling D times with replacement from $\{\widehat{u}_1, \dots, \widehat{u}_D\}$. The final bootstrap procedure to estimate $\mathrm{MSE}(\widehat{\theta}_d)$, $d = 1, \dots, D$, is presented in Algorithm 4.1.

Algorithm 4.1: `MSE_boot`: Bootstrap MSE-estimator for spline-based small area estimators.

```
for $b = 1, \dots, B$ do
    1. Draw a parametric bootstrap sample $u^b$.
    2. Draw a wild bootstrap sample $\varepsilon^b$.
    3. Simulate bootstrap data
           $y^b_{i,d} := \widehat{s}(x_{i,d}) + u^b_d + \varepsilon^b_{i,d}.$
    4. Define the bootstrap sample
           $\mathcal{S}^b := \{y^b_{i,d} : i \in \mathcal{S}\}$
       and compute the bootstrap estimates $\widehat{\theta}^b_d$, $d = 1, \dots, D$.
end
return $\widehat{\mathrm{MSE}}(\widehat{\theta}_d)$, computed according to (4.69).
```

The MSE_boot has been proposed for the spline-based small area estimators SLMM,S OPT, and SOPT_CON by Wagner et al. (2017). As an alternative bootstrap procedure, the prescaled random effects block bootstrap for multilevel data (cf. Chambers and Chandra, 2013) is implemented. Both of the MSE-estimators, however, provide very similar results such that we restrict ourselves to the MSE_boot MSE-estimator in the following.

4.6 Simulation Study

To analyze the performance of the developed point estimators SOPT and SOPT_CON and of the related MSE_boot MSE-estimator, we conduct a quasi-design-based Monte Carlo (MC) simulation study in the following. In this framework, a finite population is drawn once as a realization of a superpopulation model and kept fixed throughout the simulation. In each of the $r = 1, \dots, R \in \mathbb{N}$ simulation replications a sample is drawn out of the finite population according to an a priori specified sampling design. Each of the samples provides an estimate $\widehat{\theta}_r$ of the parameter of interest. Since the true parameter is known in the simulation setup, adequate performance measures of the estimator can be computed, e.g.

$$\mathrm{BIAS}(\widehat{\theta}) = \mathrm{E}(\widehat{\theta}) - \theta \approx R^{-1} \sum_{r=1}^{R} \widehat{\theta}_r - \theta. \tag{4.73}$$

For a sufficiently large number R of random experiments, the MC-approximation of the expectation becomes arbitrarily close to the true value. In practice, a number of $R = 10{,}000$ is frequently used.

4.6.1 Performance Measures

To evaluate the results of a large MC-simulation study, the information needs to be reduced to a manageable amount of indicators and figures. As mentioned in Section 2.1, two main features of an estimator are of interest, namely its deviation and its dispersion. In this subsection, we therefore present performance measures for point estimators and MSE-estimators that are frequently used in simulation studies.

Performance Measures for Point Estimators

Let θ_d denote the area-specific parameter of interest and let $\widehat{\theta}_d$ denote a related point estimator. In each simulation run, i.e. $r = 1, \dots, R$, we obtain an estimate $\widehat{\theta}_{d,r}$ of the estimand θ_d. To assess whether a bias in an area is present, the (Monte-Carlo) relative bias (RBIAS) is considered. It is defined as

$$\mathrm{RBIAS}_d := \frac{R^{-1} \sum_{r=1}^{R} \widehat{\theta}_{d,r} - \theta_d}{\theta_d} \in \mathbb{R}. \tag{4.74}$$

If the true value θ_d is close to zero, the (Monte-Carlo) absolute bias is considered instead. Usually, there exists a trade-off between the variability of an estimator and its bias such that an biased estimator can still be preferable to an unbiased one. A measure that takes both of these aspects into account is the (Monte-Carlo) relative root mean squared error (RRMSE). It is defined as

$$\mathrm{RRMSE}_d := \frac{\sqrt{R^{-1}\sum_{r=1}^{R}\left(\widehat{\theta}_{d,r}-\theta_d\right)^2}}{\theta_d} \geq 0. \tag{4.75}$$

As the RRMSE approximates the relative standard error of the estimator it is a widely used measure in practice. In analogy to the RBIAS, the (Monte-Carlo) MSE, i.e.

$$\mathrm{MSE}_d := R^{-1}\sum_{r=1}^{R}\left(\widehat{\theta}_{d,r}-\theta_d\right)^2, \tag{4.76}$$

is considered if θ_d is close to zero. A more robust measure of the variation of an estimator is the relative dispersion (RDISP). It is defined as

$$\mathrm{RDISP}_d := \frac{\mathcal{Q}(\widehat{\theta}_d, 0.95) - \mathcal{Q}(\widehat{\theta}_d, 0.05)}{\theta} \geq 0, \tag{4.77}$$

where $\mathcal{Q}(\widehat{\theta}_d, q)$ denotes the q-quantile of the (Monte-Carlo) distribution of $\widehat{\theta}_d$. It is a robust measure in the sense that the outlying 10% of the simulation results are rejected. The joint analysis of bias and variation frequently suffices to provide appropriate insight into the performance of a point estimator. The presented performance measures for a point estimator provide area-specific information, but in practice it might be more convenient to assess the overall performance of an estimator. For the RBIAS the mean absolute relative bias (MARB) is considered. It is defined as

$$\mathrm{MARB} := D^{-1}\sum_{d=1}^{D}|\mathrm{RBIAS}_d| \geq 0. \tag{4.78}$$

The results on the RRMSE are summarized through the average relative root mean squared error (AVRRMSE). It is defined as

$$\mathrm{AVRRMSE} := D^{-1}\sum_{d=1}^{D}\mathrm{RRMSE}_d \geq 0. \tag{4.79}$$

Performance Measures for MSE-Estimators of Point Estimators

Besides the performance of a point estimator $\widehat{\theta}_d$, the interest is also in the performance of the related MSE-estimator. This can be analyzed either by their bias or by confidence interval rates. In analogy to the RBIAS of a point estimator, the (Monte-Carlo) relative bias of the MSE-estimator (RBIASMSE) is defined as

$$\mathrm{RBIASMSE}_d := \frac{R^{-1}\sum_{r=1}^{R}\widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r}) - \mathrm{MSE}_d}{\mathrm{MSE}_d} \in \mathbb{R}. \tag{4.80}$$

Since the true mean squared error is unknown, the Monte-Carlo MSE is used in this definition. Note that the Monte-Carlo MSE of $\widehat{\theta}_d$ is a second moment and does therefore not converge as fast as a first moment. To overcome this issue it is common practice to compute the MSE-estimates $\widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r})$ for a given number of replications (often $R = 1{,}000$) and to compare this against the MSE_d calculated from significantly more replications (often $R = 10{,}000$). A negative RBIASMSE indicates for underestimation of the true MSE, whereas a positive value points to overestimation. An MSE-estimator with generally positive RBIASMSE is called conservative. If MSE_d is close to zero, which is the desirable state, the absolute bias of the MSE-estimator (BIASMSE) is considered instead of the RBIASMSE. It is defined as

$$\mathrm{BIASMSE}_d := R^{-1} \sum_{r=1}^{R} \widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r}) - \mathrm{MSE}_d \in \mathbb{R}. \tag{4.81}$$

A further approach to assess the performance of an MSE-estimator is based on the (estimated) confidence interval (CI) of significance level α, i.e.

$$\mathrm{CI}_\alpha(\hat{\theta}_{d,r}) := \left[\widehat{\theta}_{d,r} - z_\alpha \sqrt{\widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r})}, \widehat{\theta}_{d,r} + z_\alpha \sqrt{\widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r})}\right] \subset \mathbb{R}. \tag{4.82}$$

The value z_α denotes the related $(1-\alpha/2)$-quantile of the standard normal distribution with a typical value of $\alpha = 0.05$. The confidence interval coverage rate (CICR) is defined as

$$\mathrm{CICR}_d := R^{-1} \sum_{r=1}^{R} \mathbb{1}_{\mathrm{CI}_\alpha(\widehat{\theta}_{d,r})}(\theta_d) \in [0,1] \tag{4.83}$$

and measures the proportion of MC-replications for which the confidence interval covers the true value θ_d. In a simulation study, it is expected that $(1-\alpha)\cdot 100\%$ of these confidence intervals cover the true value, i.e. $\mathrm{CICR}_d \approx 1-\alpha$. For a biased estimator, however, the confidence intervals are shifted from the true value such that the expected CICR can be undershot (cf. Särndal et al., 1992, Chapter 5.2). Also a systematic underestimation of the true MSE of the point estimator by the applied MSE-estimator causes a break down of CICR, since the confidence intervals become too short. On the other hand, an excessively high MSE-estimate results in a CICR of 100%. Therefore, it is of interest to compare the CICR against the mean confidence interval length (MCIL) defined as

$$\mathrm{MCIL}_d := R^{-1} \sum_{r=1}^{R} \mathrm{CIL}(\widehat{\theta}_{d,r}), \tag{4.84}$$

where

$$\mathrm{CIL}(\widehat{\theta}_{d,r}) := 2 z_\alpha \sqrt{\widehat{\mathrm{MSE}}(\widehat{\theta}_{d,r})} \geq 0 \tag{4.85}$$

denotes the length of the confidence interval $\mathrm{CI}_\alpha(\widehat{\theta}_{d,r})$ and, as before, z_α denotes the respective quantile of the standard normal distribution.

4.6.2 Simulation Setup

In the following, we present the general setup of the conducted simulation study. This basically includes the process of generating the data and the settings for the examined point and MSE-estimators. Further, we introduce two different sampling designs resulting in two scenarios of the simulation study.

Population and Sampling

We consider a finite population of $N = 30{,}000$ units allocated to $D = 30$ areas, each of size $N_d = 1,000$. The covariates are uniformly located in the interval $[0, 1]$ and the finite population is generated according to the model

$$\begin{aligned} y_{i,d} &= f(x_{i,d}) + u_d + w_{i,d}\varepsilon_{i,d}, \ i = 1, \dots, N, \\ u_d &\overset{\text{iid}}{\sim} \mathcal{N}(0, 0.05^2), \\ \varepsilon_{i,d} &\overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1^2). \end{aligned} \tag{4.86}$$

The function f denotes a so called sigmoid function defined as

$$f\colon [0,1] \to [1,2], \ x \mapsto 1 + \frac{1}{1 + \exp\left(-8\left(x - 0.3\right)\right)} \tag{4.87}$$

and the $w_{i,d} := x_{i,d} + 0.5$ are unit-specific weighting factors of the disturbance. In each of the $R = 10{,}000$ MC-replications a sample of total size $n = 300$ is drawn. To introduce some variation, the area-specific subsample sizes are as follows:

- $n_d = 3$ for the areas $d = 1, \dots, 10$,
- $n_d = 9$ for the areas $d = 11, \dots, 20$,
- $n_d = 18$ for the areas $d = 21, \dots, 30$.

For further variation, we systematically allocate the units to the areas proportional to the size of the covariate. That is, units with comparatively small covariate value x_i are more likely allocated to an area with a lower index d, whereas units with higher covariate value are preferably located to areas with increased index. This systematic allocation leads to variations in the mean values of the covariate $\mu_{X,d}$ and of the variable of interest $\mu_{Y,d}$ for the specific areas. Table 4.1 summarizes the setup for the MC-simulation study, where decimals are rounded to two digits.

For the sampling design which is utilized to draw a sample in each MC-replication, we consider two different scenarios. The first design is stratified random sampling (StRS), where each area is considered as stratum. That is, from each area the required number of samples n_d is drawn completely at random. In order to investigate the impact of the introduced shape constraints of the SOPT_CON-estimator, we further apply StRS, but restrict the sampled units to covariates with $x_{i,d} \geq 0.35$. We refer to this sampling design as restricted stratified random sampling (ResStRS).

	$n_d = 3, \quad n_d/N_d = 0.3\%$									
	1	2	3	4	5	6	7	8	9	10
$\mu_{X,d}$	0.09	0.18	0.25	0.31	0.35	0.38	0.42	0.44	0.45	0.48
$\mu_{Y,d}$	1.16	1.07	1.15	1.13	1.22	1.42	1.48	1.45	1.47	1.40

	$n_d = 9, \quad n_d/N_d = 0.9\%$									
	11	12	13	14	15	16	17	18	19	20
$\mu_{X,d}$	0.50	0.52	0.52	0.52	0.56	0.55	0.57	0.57	0.58	0.58
$\mu_{Y,d}$	1.36	1.54	1.53	1.45	1.57	1.55	1.54	1.55	1.63	1.56

	$n_d = 18, \quad n_d/N_d = 1.8\%$									
	21	22	23	24	25	26	27	28	29	30
$\mu_{X,d}$	0.59	0.61	0.59	0.62	0.62	0.62	0.61	0.63	0.64	0.64
$\mu_{Y,d}$	1.65	1.63	1.62	1.65	1.67	1.64	1.57	1.59	1.71	1.72

Table 4.1: Setup for the MC-simulation study

The utilized superpopulation model and the sampling designs are motivated from the practical application of timber volume estimation introduced in Section 1.1. The underlying data are presented in Figure 1.1 and indicate for an S-shape as well as a heteroscedastic error. This is reproduced within the simulation study due to the sigmoid function and the increasing weighting factors $w_{i,d} = x_{i,d} + 0.5$. Further, observations with small values of the auxiliary information are not located in the sample which is modeled by the sampling design with cut off, i.e. ResStRS. Finally, the variation of the mean canopy height between the forest districts is represented by the systematic area allocation.

Point and MSE-Estimators

To estimate the area-specific mean values

$$\theta_d := \mu_{Y,d} = \frac{1}{N_d} \sum_{i \in \mathcal{U}_d} y_{i,d}, \;\; d = 1, \dots, 30, \tag{4.88}$$

we apply the SLMM-estimator (4.30), the SOPT-estimator (4.42), and the SOPT_CON-estimator (4.52). To allow for an appropriate comparison of the estimators, we utilize the same spline model for all of the three spline-based estimators. We employ the cubic B-spline basis with $m = 35$ equally spaced knots and related difference penalty of order two (cf. Subsection 3.2.2). For the SOPT- and the SOPT_CON-estimator, we determine the regularization parameters λ_s and λ_u according to the sequential parameter selection approach presented in Subsection 4.2.3. For the SOPT_CON-estimator, we additionally demand for a monotonically increasing function with function values at least one, i.e. we impose $s' \geq 0$ and $s \geq 1$ on Ω as shape constraints. To estimate the MSE of the point estimators, we apply the `MSE_boot` MSE-estimator defined in Algorithm 4.1 with $B = 99$ bootstrap replication for the first 1,000 out of the $R = 10{,}000$ MC-runs.

4.6.3 Simulation Results

In this subsection, the results of the MC-simulation study are discussed. We compare the performance of the applied point estimators and further analyze the suitability of the related bootstrap MSE-estimator.

Performance of the Point Estimators

The main reason for considering spline models and in particular shape constraints in small area estimation is to achieve a more realistic model. This is expected to reduce the bias of an estimator. Figure 4.1 presents the RBIAS of the applied methods as a boxplot over all areas and as a line plot versus all areas under the different sampling designs.

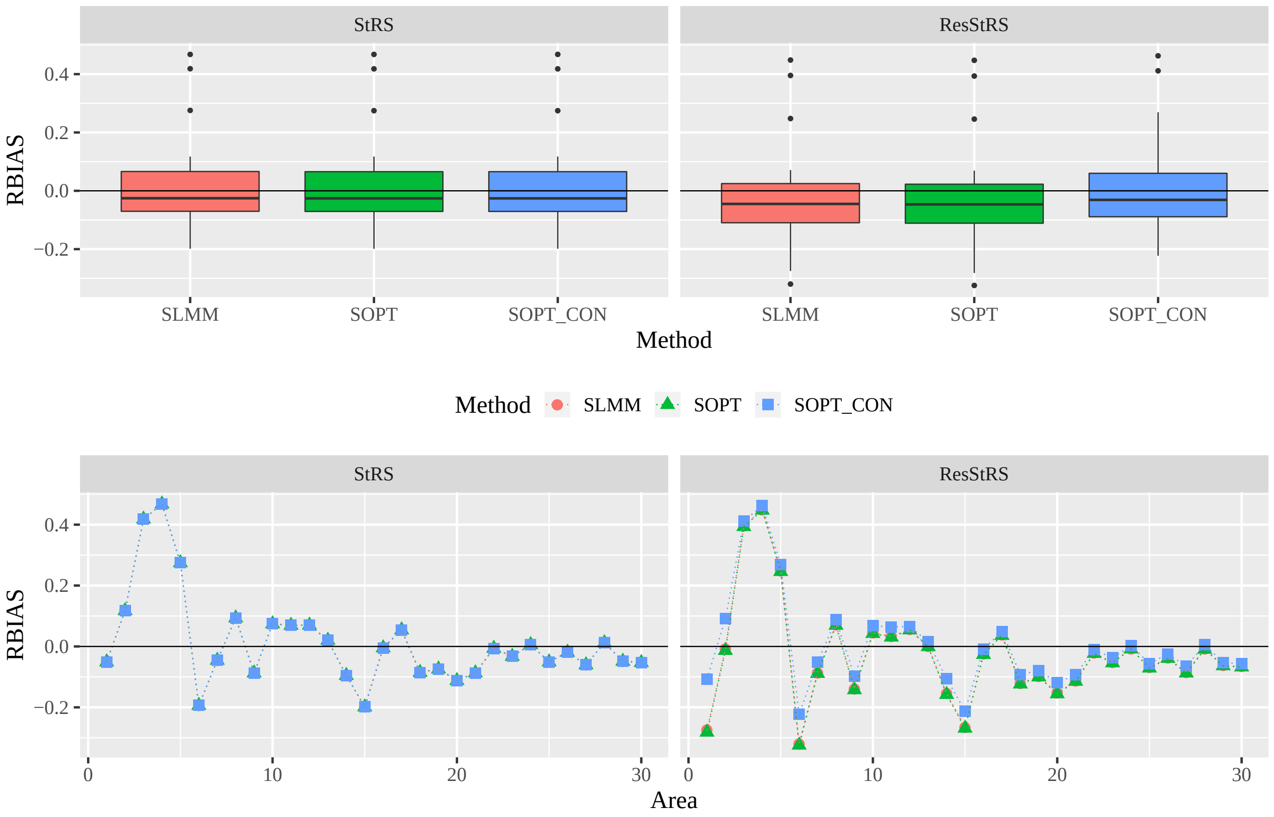

Figure 4.1: RBIAS of the point estimators under the sampling designs.

For the first scenario, all of the three estimators perform identically in terms of the RBIAS. This is due to the fact that the same underlying spline model is applied to recover the general trend within the data. Further, the sample data are rich enough to adequately represent the entire finite population such that the additional shape constraints utilized by the SOPT_CON-estimator do not have a visible effect. Therefore, the three estimators only differ by the selected regularization parameter. The different approaches to determine the regularization parameter, however, provide similar results for this first scenario. The spline model underlying the three estimators recovers the data sufficiently accurate such that a RBIAS of less than 10% is achieved for most of the areas. Frequently, the RBIAS even falls below 5%. The comparatively largest biases occur for the areas with small subsample sizes, i.e. $d = 1, \ldots, 10$. Despite the fact that the spline model accurately recovers the general trend, the small subsample sizes affect the estimation of the area intercept u_d. The small subsample sizes do not support an appropriate estimation of these effects, which causes larger biases for those areas. Further, an unexpected relatively large bias is observed for the area with index $d = 15$. For the case of the ResStRS-design, the SLMM-estimator and the SOPT-estimator still perform similar in terms of the RBIAS. Due to the cut off introduced by the sampling design, the sample data become less informative such that the RBIAS of these two methods increases compared to the first scenario. This becomes in particular visible for the areas with mainly small covariate values, i.e. with a smaller index d, since the cut off of the sampling design is located at the left margin of the data. There, the two unconstrained spline methods are not able to recover the underlying trend. This can be seen in Figure 4.2, where the P-spline fits underlying the respective estimators to selected MC-sample data is graphed. In this scenario, the shape constraints utilized by the SOPT_CON-estimator have a visible effect and ensure that the underlying P-spline still adequately represents the general trend in the data. Therefore, the ResStRS-design only slightly affects the performance of the SOPT_CON-estimator compared to the StRS-design. Due to this fact, the SOPT_CON-estimator is much more robust towards the cut off of the sampling design and outperforms the two unconstrained methods.

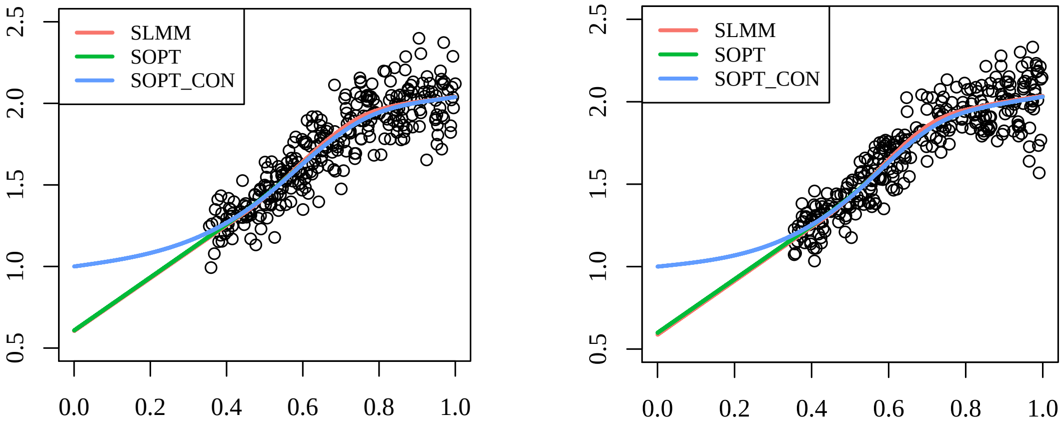

Figure 4.2: P-spline fits to selected MC-samples under the ResStRS-design.

Figure 4.3 presents the RRMSE of the applied methods as a boxplot over all areas and as a line plot versus all areas under the different sampling designs. As model-based estimators, the three methods exhibit small variances such that the RRMSE is dominated by the related biases. Therefore, the results on the RRMSE confirm the former findings. For the first scenario, the estimators perform identical in terms of the RRMSE and the relatively large RRMSEs occur for the areas that already occupy larger biases. These are the areas with small subsample size $d = 2, 3, 4$ and the area with index $d = 15$. For the ResStRS-design, the SOPT_CON-estimator still outperforms the unconstrained methods and the areas with relatively large RRMSEs coincide with those possessing a larger bias. An unexpected result in this scenario is the observation of a slight advantage for the SLMM- compared to the SOPT-estimator in terms of the RRMSE. Since the RBIAS of the both methods approximately coincide, this indicates for a higher variance of the SOPT-estimator in this case. Figure 4.4 presents the boxplots of all $R = 10{,}000$ MC-runs corrected by the true value, i.e. $\widehat{\theta}_{d,r} - \theta_d$, for this scenario for all estimators and all areas. This confirms the assumption of a higher variability of SOPT-estimator since the related boxes show significantly more outliers and a wider spread.

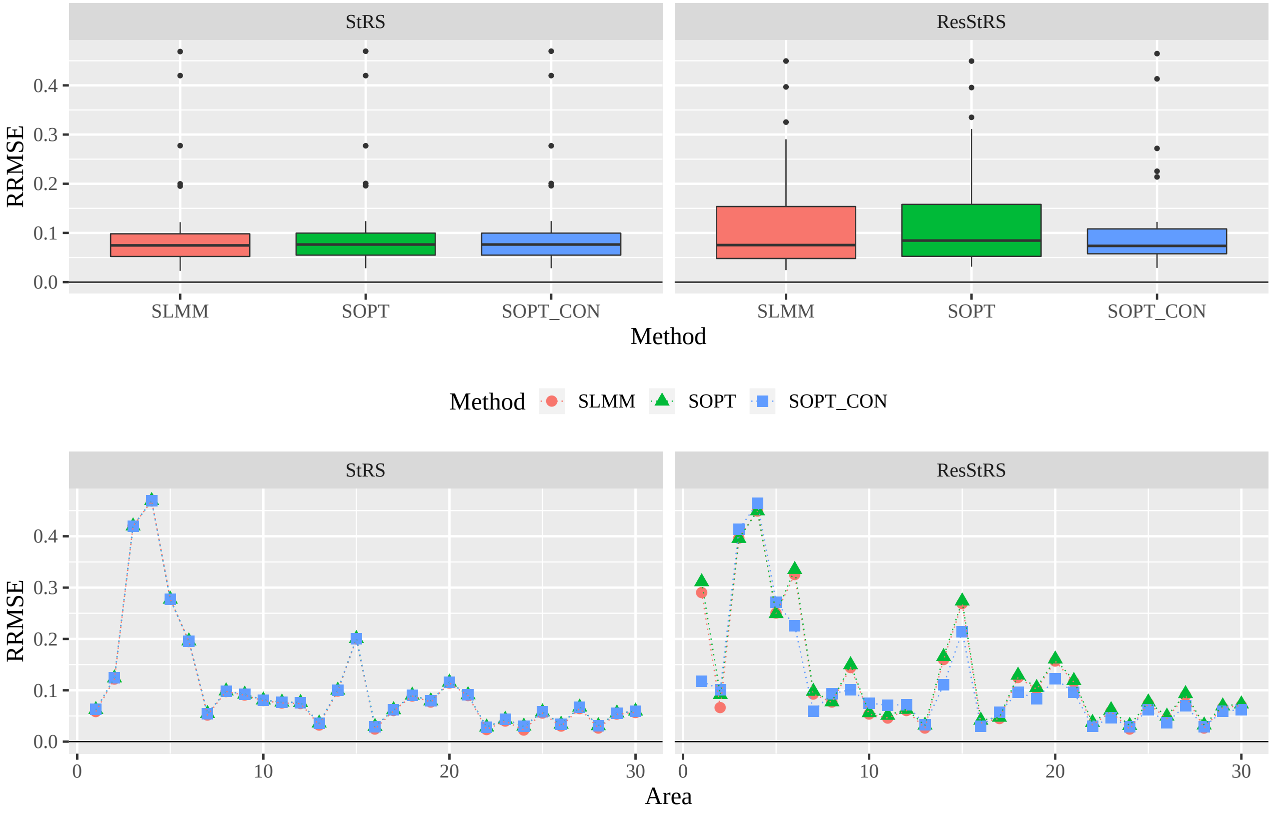

Figure 4.3: RRMSE of the point estimators under the sampling designs.

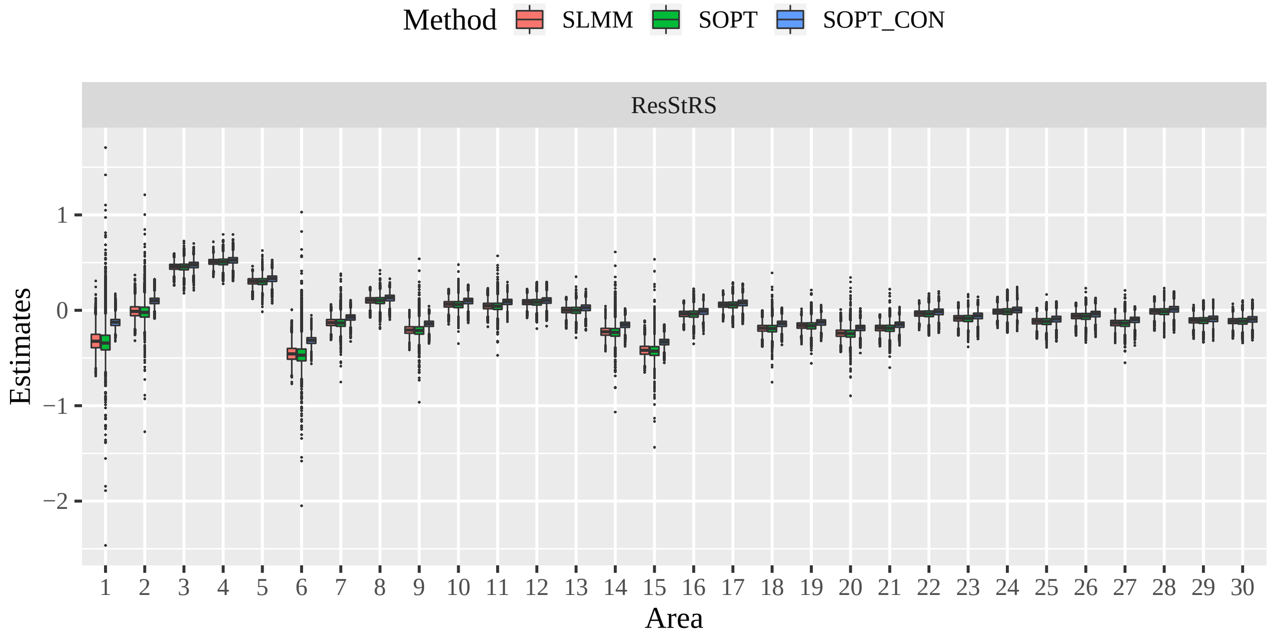

Figure 4.4: Point estimates for the $R = 10{,}000$ MC-replications under the ResStRS-design.

The increased variability of the SOPT- compared to the SLMM-estimator is due to the fact that the process of the simulation study does not allow for a retrospective adjustment of the regularization parameter λ_s. Therefore, an uncontrolled behavior of the P-spline underlying the SOPT- and the SOPT_CON-estimator can occur. This is displayed in the left-hand graph of Figure 4.5. In the sample the P-spline underlying the SOPT- and the SOPT_CON-estimator heavily overfits the data. At the left margin, the P-spline underlying the SOPT-estimator is pulled upwards due to comparatively high values of the leftmost sample data. This effect also concerns the P-spline underlying the SOPT_CON-estimator, but the monotonicity constraint steers in the opposite direction. This volatile behavior indicates for an inadequate choice of the regularization parameter λ_s. Indeed, for this selected MC-sample the utilized regularization parameters are $\lambda_{\text{SLMM}} = 66.1$ and $\lambda_s = 0.01$. The value of 0.01 marks the smallest value considered within the grid search to determine the regularization parameter for the SOPT and the SOPT_CON. This is a further indicator of an inadequate parameter selection. In a concrete application, this unrealistic behavior is controlled by a retrospective adjustment of the regularization parameter. Resetting $\lambda_s = \lambda_{\text{SLMM}}$ yields the right-hand graph of Figure 4.5, which is in line with the expected P-spline fits. The automatic selection of the regularization parameter within a grid search without manual adjustment is therefore inadequate in practice, since it deprives the SOPT- and the SOPT_CON-estimator of one of their strengths, namely the control of the smoothness of the underlying P-spline. For the simulation study, however, the automatic selection is necessary and explains the increased RRMSE of the SOPT- compared to the SLMM-estimator.

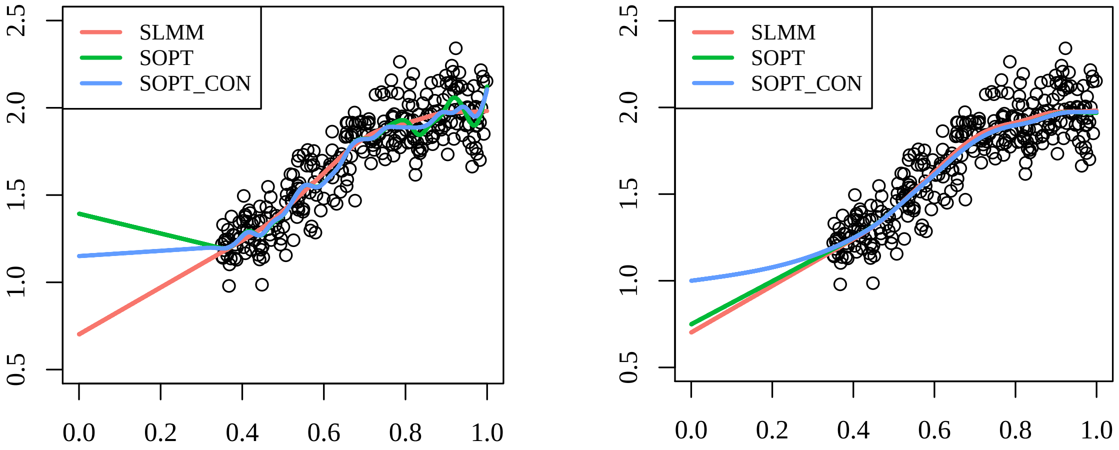

Figure 4.5: Uncontrolled (left) and controlled (right) behavior of the P-splines of the SOPT- and the SOPT_CON-estimator for a selected MC-sample under the ResStRS-design.

Performance of the MSE-Estimator

Figure 4.6 depicts the BIASMSE of the bootstrap MSE-estimator for all of the three methods under both scenarios. For the first scenario, the bootstrap MSE-estimator performs identical in terms of the BIASMSE for all of the three point estimators. This is due to the fact that the point estimators themselves are identical in this scenario. The bootstrap MSE-estimator is almost unbiased, but slightly overestimates in each area and is therefore a conservative MSE-estimator. For the areas with small subsample sizes the BIASMSE is slightly larger, which stabilizes with increasing sample sizes. For the second scenario, the bootstrap MSE-estimator for all methods is much more biased. Especially for the areas that are more affected by the cut off of the ResStRS-design the BIASMSE increases. A bit unexpected is the observation that the BIASMSE also increases for the areas that are not affected by the sampling design. The utilization of the shape constraints by the SOPT_CON-estimator leads to a decreased bias of the bootstrap MSE-estimator. For all three point estimators the bootstrap MSE-estimator produces mainly conservative results. In practice, this is much more acceptable than underestimation. In the areas with index $d = 3, 4, 5$, however, the bootstrap MSE-estimator underestimates the true value for all methods. Note that these are the areas that are also concerned with a relatively large RBIAS.

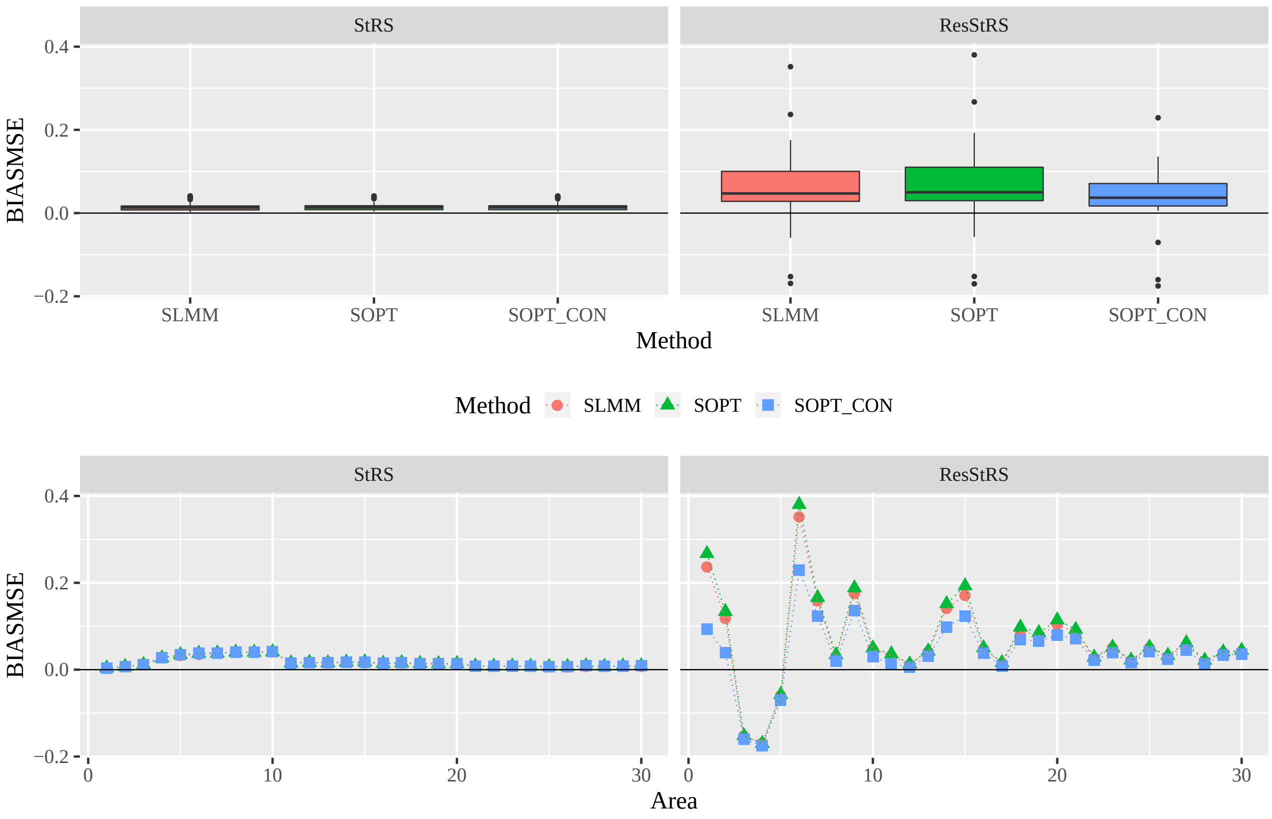

Figure 4.6: BIASMSE of the MSE-estimator for the point estimators under the sampling designs.

Figure 4.7 depicts the CICR of the bootstrap MSE-estimator for all point estimators versus the area index, the MCIL, and the RBIAS of the point estimators. For the first scenario, the CICR is close to the nominal coverage rate of 95% for all of the three methods and is still at 75% in the worst case. With a mean value of approximately 0.5, the MCILs for all methods are moderate such that the MSE-estimator performs satisfactory for all methods under the StRS-design. For those areas with a comparatively large RBIAS of the point estimator also the MCIL is large such that the CICR is there also satisfactory but not meaningful. For the ResStRS-design the picture is slightly different. The nominal coverage rate is much more often reached which is due to an also increasing MCIL. This increased MCIL compared to the first scenario is caused by the increased RRMSE of the related point estimators. For the areas with index $d = 3, 4, 5$

the coverage rate breaks down to approximately 50%. This is due to the comparatively large biases in these areas since in this case the confidence intervals are shifted from the true value.

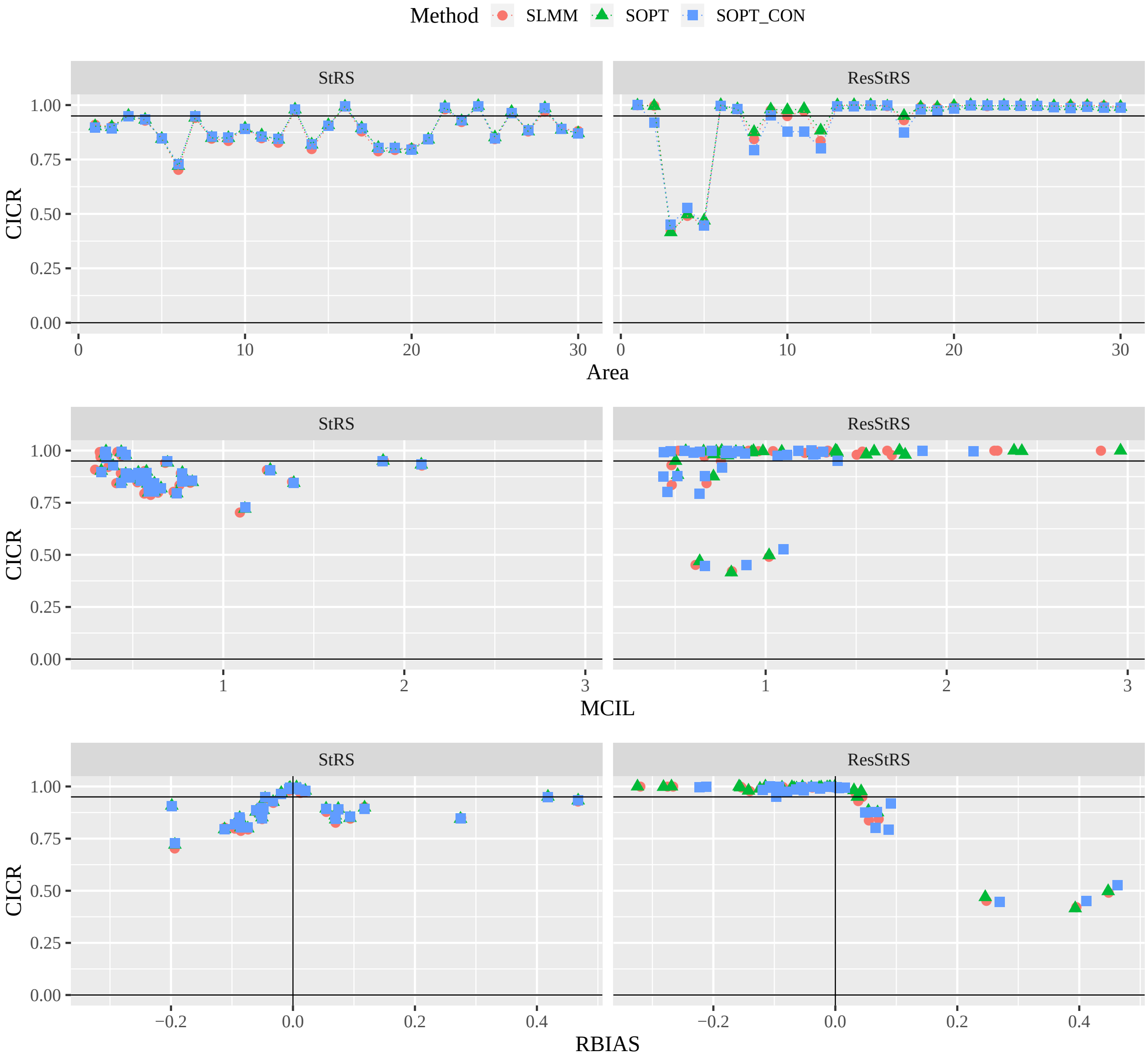

Figure 4.7: CICR of the `MSE_boot` MSE-estimator for the point estimators under the sampling designs.

4.6.4 Summary and Discussion

In the simulation study, the performance of the newly developed point estimators SOPT (4.42) and SOPT_CON (4.52), also in comparison to the SLMM-estimator (4.30), under two different sampling designs is examined. The superpopulation model utilized for the simulation study and the applied sampling designs are motivated by a real-world application. If the sample data adequately represent the underlying finite population, the point estimators turn out to perform identical and provide satisfying estimates in terms of bias and of variance. In this scenario, also the `MSE_boot` MSE-estimator (defined in Algorithm 4.1) for the point-estimators yields adequate results. We conclude that in the case of the StRS-design all of the three estimators are appropriate and that the `MSE_boot` MSE-estimator is applicable to each of the point estimators. The much more interesting case is the scenario resulting from the ResStRS-design. The cut off of the sample data generates variation that enables to expose the differences of the applied methods. A bit surprising is the fact, that the SLMM-estimator performs superior to the SOPT-estimator in this case. This is due to the increased variability of the SOPT-estimator caused by an automatization of the regularization parameter selection. Within a more sophisticated parameter selection algorithm, we expect the both methods to perform also quite similar in this ResStRS-design scenario. As expected, under the ResStRS-design the SOPT_CON-estimator outperforms the remaining estimators due to a more accurate P-spline fit resulting from the incorporated shape constraints. In this scenario, the `MSE_boot` MSE-estimator performs inferior compared to the first scenario. However, especially for the SOPT_CON-estimator the results are still satisfying.

An interesting observation that is not further examined within this simulation study is the fact that the utilization of a heteroscedastic error term does not noticeably effect the performance of the estimators. The expectation that a heteroscedastic error term causes a rather careful parameter selection, i.e. a relatively large λ_s and λ_{SLMM}, is not confirmed by the simulation study. This might be due to the fact that the unit specific terms $w_{i,d}\varepsilon_{i,d}$ are still moderate in size and are compensated by the spline model. Here, the further examination of the impact of error terms with a wider spread as well as of outliers is worth to be considered. In this context, the incorporation of the weighted least squares functional (3.37) into the SOPT- and the SOPT_CON-estimator is a promising feature. Finally, a conspicuous behavior of all of the estimators is observed in the area $d = 15$. This is, however, neither caused by the superpopulation model nor by the sampling design and demands for a further investigation.

4.7 Application: Timber Reserve Estimation in Rhineland-Palatinate

We finally turn back to the real-world application of estimating timber reserves in the several forest districts of Rhineland-Palatinate (RLP). This application is already introduced in the motivational Section 1.1 and also addressed by Münnich et al. (2016) and Wagner et al. (2017). To provide area-specific timber reserve estimates, we apply the SOPT- (4.42) and the SOPT_CON-estimator (4.52) as well as the SLMM- (4.30) and the BHF-estimator (4.15) for comparative reasons. At first, we introduce the data utilized for this application in more detail.

Data Basis

In order to conduct the state forest inventory in RLP, the forest land of the federal state is, in simplified terms, overlaid with $N = 3{,}065{,}696$ grid cells. For $n = 521$ uniformly cells, the mean (spruce) timber reserve in cubic meter is observed within the scope of the SFI 2007.

These values are considered as response variable y_i within the study which aims in estimating the mean timber reserves per hectare for the $D = 46$ forest districts of RLP. That is, the area-specific parameter of interest is given as

$$\theta_d := \kappa_d \cdot \mu_{Y,d} \ , d = 1, \ldots, D, \tag{4.89}$$

where $\kappa_d \in \mathbb{R}$ denotes a given conversion factor (cf. Mandelkow, 2012, Chapter 5). For nearly the whole territory of RLP, airborne laser scanning data are available. They are collected by the state surveying office over the course of twelve years from 2002 to 2013 and thus in particular in 2007. From these data a normalized surface model is created from which the mean canopy height in meter for each cell is derived (cf. St-Onge et al., 2003). These values are considered as auxiliary covariate x_i and the relationship between response variable and covariate for the sampled units is presented in Figure 1.1. A high correlation between canopy height and timber volume is observable, but a (generalized) linear relationship cannot be assumed. Further, a nonnegative and monotonically increasing relationship is to expect. The ALS data are (partially) available for $D = 44$ out of the 46 forest districts of RLP. For the two further forest districts neither the sample data nor the ALS data are available such that these districts are excluded from the study. We refer to Mandelkow (2012) and the references therein for more details on the data collection process. Figure 4.8 presents the area-specific subsample sizes n_d and the related sample fractions on a percentage basis, i.e. $(n_d/N_d) \cdot 100\%$. Due to the very small and partially nonexisting subsamples, the estimation of timber reserves in the several forest districts of RLP requires the application of small area estimation methods. Further, since design information are rarely available and difficult to obtain, the model-based approach has to be pursued.

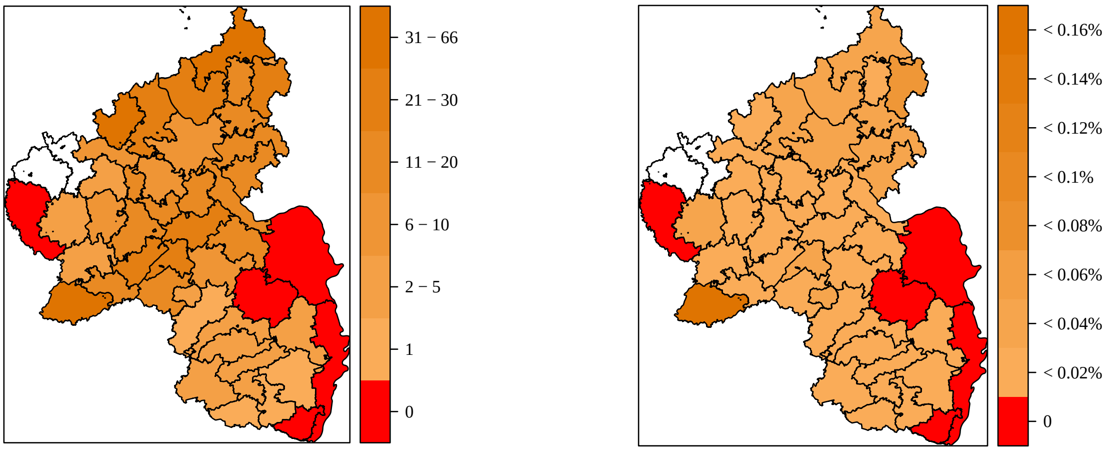

Figure 4.8: Area-specific sample sizes (left) and sample fractions on a percentage basis (right) for the $D = 44$ considered forest districts in RLP.

Small Area Estimates

In order to estimate the mean timber reserves per hectare in the several forest districts of RLP, that is

$$\theta_d := \kappa_d \cdot \mu_{Y,d}, \;\; d = 1, \ldots, D = 44, \tag{4.90}$$

we apply the estimators BHF, SLMM, SOPT, and SOPT_CON. For the spline parameters of all of the spline-based estimators, we chose cubic splines with $m = 35$ equally spaced knots. According to Ugarte et al. (2009), we base the SLMM-estimator on the B-spline basis with difference penalty of order two. For the SOPT- and the SOPT_CON-estimator, we utilize the B-spline basis with curvature penalty and determine the regularization parameters by the identic parameter selection approach (cf. Subsection 4.2.3) and the cross-validation method (cf. Subsection 3.2.4). For the SOPT_CON-estimator, we additionally impose a nonnegativity constraint on the small area P-spline, which is reasonable since timber volume cannot be negative. Further constraints such as a monotonically increasing behavior of the small area P-spline are also plausible, but the nonnegativity constraint turns out to suffice to provide reliable small area estimates. The resulting small area estimates of the applied estimators are illustrated in Figure 4.9. The compared estimators yield a quite similar overall picture, i.e. the related small area estimates most often do not tremendously differ. However, it is apparent that the BHF-estimator results in four negative estimates for the timber reserves and that the SLMM-estimator produces one negative estimate as well. Since timber volume has to be nonnegative, these results are infeasible. Note that the case of infeasible estimates only occurs for unsampled areas. A simplistic approach to overcome this issue is to set the negative estimates to zero such that all of the estimates become feasible. Estimating nonexisting timber volume, however, would still be inaccurate since positive canopy heights are observed in the related areas, which automatically results in strictly positive timber volumes. Therefore, zero estimates of timber reserves are not accepted in practice. The SOPT- and the SOPT_CON-estimator, on the contrary, yield feasible small area estimates without any exception, i.e. even for the unsampled areas.

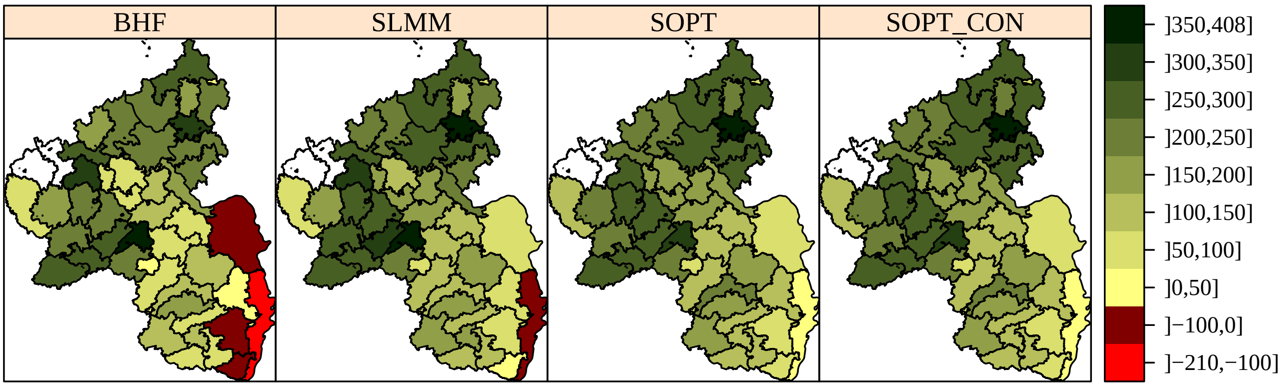

Figure 4.9: Estimates of the mean timber reserves per hectare for the $D = 44$ forest districts in RLP obtained by the estimators BHF, SLMM, SOPT, and SOPT_CON.

Underlying Regression Functions

To further analyze the obtained estimation results, the underlying regression functions are displayed in Figure 4.10. The linear approximation of the BHF-estimator is clearly insufficient since the curvature of the sample data is not reflected. Further, the regression line underlying the BHF-estimator yields highly negative predictions for small canopy heights which is unrealistic and therefore unacceptable. This explains the several negative small area estimates obtained by the BHF-estimator. By contrast, all of the three spline functions adequately reflect the curvature of the sampled data. In the sample all of the splines approximately coincide, but differ out of the sample. There, the P-spline function underlying the SLMM-estimator deviates from the P-splines underlying the SOPT- and the SOPT_CON-estimator. This is due to the different regularization strategies and regularization parameter selections. Especially at the left margin, the spline function underlying the SLMM-estimator yields negative and therefore unrealistic function values. This insufficient behavior explains the one infeasible small area estimate of the SLMM-estimator. The (unconstrained) small area P-spline produces negative values as well, but these are as less negative as the values of the spline underlying the SLMM-estimator. Therefore, the SOPT-estimator provides exclusively feasible estimates, even though the underlying small area P-spline is insufficient at the left margin. The small area P-splines, both unconstrained and with shape constraints, only differ on the left margin. There, the non-negativity constraint utilized by the SOPT_CON-estimator has a visible effect such that the unrealistic negative function values are avoided. This slight modification guarantees feasible small area estimates and realistic function values.

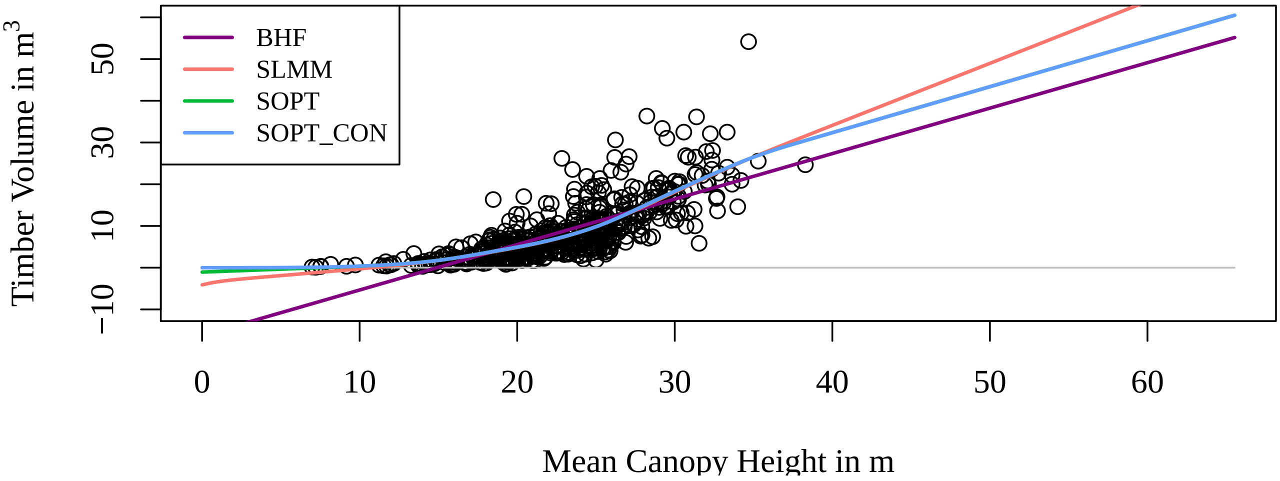

Figure 4.10: Regression functions of the applied estimators BHF, SLMM, SOPT, and SOPT_CON.

MSE-Estimates

Besides the plausibility of the estimates, we analyze the performance of the spline-based small area estimators in terms of their relative root mean squared error (RRMSE). Figure 4.11 presents the estimated RRMSEs on a percentage basis of the applied estimators, obtained by 499 repetitions of the `MSE_boot` MSE-estimator presented in Algorithm 4.1. The SOPT- and the SOPT_CON-estimator yield much more stable estimates in terms of the RRMSE compared to the SLMM-estimator. This is due to the numerical more stable optimization approaches and the more flexible choice of the regularization parameter. The area-specific RRMSEs of the SLMM-estimator frequently exceed 10%, most of them by far. On the contrary, for the SOPT-estimator the area-specific RRMSEs are all less than 10%, expect one, most of them less than 5%. The RRMSEs of the SOPT- and the SOPT_CON-estimator are quite similar, but, due to the shape constraints, even for the last vacant area the RRMSE gets reduced to less than 10%. Thus, despite the fact that the SOPT-estimator already results in exclusively feasible small area estimates, the incorporation of the shape constraints pays off in terms of RRMSE.

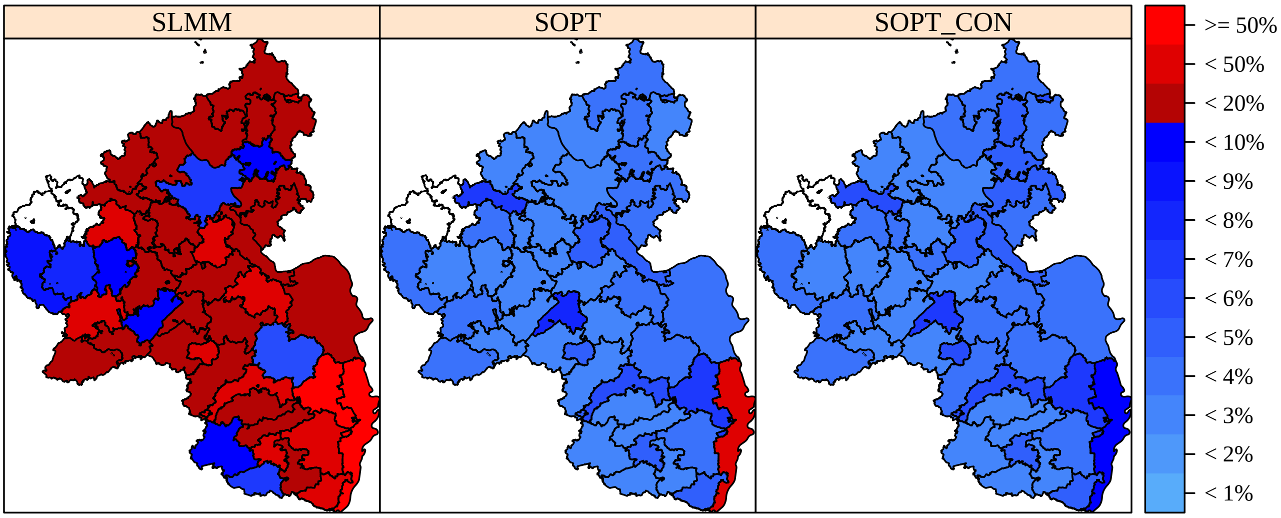

Figure 4.11: RRMSE of the estimators SLMM, SOPT, and SOPT_CON, estimated by the `MSE_boot` MSE-estimator with $B = 499$ repetitions.

Chapter 5

Large-Scale Algorithms for Penalized Spline Methods

As shown in the previous chapters, the determination of the small area P-spline (cf. Definition 4.2.1) requires the solution of the linear system (4.37), whereas the shape constrained small area P-spline (cf. Definition 4.47) is based on the solution of the strictly convex QP (4.65). That is

$$\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\alpha \overset{!}{=} \Phi^T\left(y - Cy\right) \tag{5.1}$$

for the unconstrained case and

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^K} \quad & \frac{1}{2}\alpha^T\left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right)\alpha - \left(\Phi^T(y - Cy)\right)^T\alpha \\ \text{s.t.} \quad & \Gamma_r\alpha \le 0,\ r \in I_{\le} \\ & \Gamma_r\alpha \ge 0,\ r \in I_{\ge} \end{aligned} \tag{5.2}$$

for the shape constrained case. For the special case of $C = 0$, we obtain the regression P-spline (3.39) and the shape constrained regression P-spline (3.88), respectively. At first appearance, these problems can be solved by the classical solution algorithms for the respective problem classes. The problem dimension K, however, corresponds to the dimension of the underlying spline space $\mathcal{S}_q(\mathcal{K})$ and is given as

$$K = \dim(\mathcal{S}_q(\mathcal{K})) = \prod_{p=1}^{P}\dim(\mathcal{S}_{q_p}(\mathcal{K}_p)) = \prod_{p=1}^{P}(m_p + q_p + 1) = \mathcal{O}(2^P) \tag{5.3}$$

and therefore depends exponentially on the number utilized covariates P. This exponential dependency of K on P is often referred to as the curse of dimensionality (cf. Bellman, 1957). The exemplarily choice of $m_p = 36$ and $q_p = 3$ for all $p = 1, \ldots, P$, which is consistent with the suggestions made in Section 3.2, leads to a problem dimension of $K = 40^P$. The related spline space dimensions for this concrete example are presented in Table 5.1.

	$P = 2$	$P = 3$	$P = 4$	$P = 5$
$\dim(\mathcal{S}_q(\mathcal{K}))$	1,600	64,000	2,560,000	102,400,000

Table 5.1: Exponential dependency of the spline space dimension $K = \dim(\mathcal{S}_q(\mathcal{K}))$ on the covariate dimension P.

It becomes evident that even a moderate number of covariates leads to computationally very challenging large-scale problems (5.1) and (5.2). A serious issue thereby is the required amount of memory caused by the high-dimensional matrices occurring within the considered problems. For example, for $P = 3$ already the storage of the penalty matrix Λ of dimension $K \times K$ requires approximately two gigabyte (GB) of random access memory (RAM) using the compresses sparse column (CSC) format from the `Matrix` package (cf. Bates and Maechler, 2018) of the statistical computing software `R` (cf. R Core Team, 2018). Since Λ is not the only matrix that has to be stored and since the matrices further need to be manipulated, i.e. the problems (5.1) and (5.2), respectively, have to be solved, this clearly exceeds the internal memory of common computer systems which averages 8-16 GB of RAM. For further increasing P, even the memory of currently available supercomputers is no longer sufficient. This rapid growth of the memory requirements prevents the utilization of classical pre-implemented solution algorithms and demands for much more sophisticated strategies in order to make the P-spline methods developed within this thesis applicable to multiple covariates. Therefore, this chapter is devoted to the development of computational efficient solution algorithms for the particular large-scale problems (5.1) and (5.2) and especially to the memory efficient implementation of these algorithms. In this context, computational efficiency is related to the number of required iterations to solve a given problem and is therefore a property of the applied algorithm, whereas the memory efficiency mainly depends on the implementation. The runtime of an algorithm, i.e. the time required to terminate, is of further interest and is affected by both the applied algorithm and the related implementation.

As pointed out before, the occurring matrices do not fit into the internal memory of nowadays available computer systems. Therefore, the developed solution algorithms have to be matrix-free, i.e. they do not assemble and store the matrices explicitly, but only accesses them by performing matrix operations such as evaluating matrix-vector products. In Section 5.1, we therefore implement the desired matrix operations in a memory efficient manner by exploiting the special inherent structure of these matrices. To solve the large-scale linear system (5.1), we present a matrix-free conjugate gradient method in Section 5.2 based on the implemented matrix operations. Section 5.3 is devoted to the large-scale strictly convex QP (5.2). We apply a quadratic penalty approach to reformulate the constrained optimization problem (5.2) as an unconstrained convex optimization problem which is solved by the Newton method. There, the large-scale linear system to determine the Newton direction is memory efficiently solved by a modification of the matrix-free CG method for the linear system (5.1). In order to improve the computational complexity of the utilized matrix-free CG methods, we implement a matrix-free preconditioner in Section 5.4 based on the multigrid idea. The performance of this MGCG method is analyzed in numerical test examples in Section 5.5, also in comparison to the unpreconditioned CG method.

5.1 Memory Efficient Matrix Operations

Due to the curse of dimensionality, the matrices Φ, Λ, and Γ_r cannot be stored for increasing covariate dimension P. Therefore, the respective solution algorithms have to be designed in such a way that they only access these matrices by performing adequate matrix operations. By exploiting their inherent structure, we develop several memory efficient operations for these special types of matrices in this section.

5.1.1 Matrix Structures

Due to Definition 3.1.6, a tensor product spline basis function $\phi_{j,q}$ is given as the component-wise product of spline basis functions in one variable, i.e.

$$\phi_{k,q} = \phi_{j,q} \colon \Omega \subset \mathbb{R}^P \to \mathbb{R},\ x = (x^1, \dots, x^P)^T \mapsto \prod_{p=1}^{P} \phi^p_{j_p,q_p}(x^p). \tag{5.4}$$

Based on this tensor product nature, we derive convenient representations of the high-dimensional P-spline related matrices in terms of their one-dimensional counterparts in the following.

Tensor Product Spline Basis Matrix

Since the covariates $x_1, \dots, x_n \in \mathbb{R}^P$ are scattered, the tensor product spline basis matrix Φ does not exhibit a special structure as well. Due to the aforementioned tensor product nature, however, we obtain a representation of Φ^T as Khatri-Rao product (cf. Definition 2.3.3) of the related transposed spline basis matrices in one variable. This is stated by the following lemma.

Lemma 5.1.1
For the matrices

$$\Phi_p \in \mathbb{R}^{n \times J_p},\ \ \Phi_p[i, j_p] := \phi^p_{j_p,q_p}(x^p_i),\ \ p = 1, \dots, P, \tag{5.5}$$

it holds

$$\Phi^T = \bigodot_{p=1}^{P} \Phi_p^T \in \mathbb{R}^{K \times n}. \tag{5.6}$$

Proof. Let $i \in \{1, \dots, n\}$ be arbitrary. Due to Lemma 3.1.7 and Definition 2.3.3, it holds

$$\Phi^T[\cdot, i] = \phi(x_i) = \bigotimes_{p=1}^{P} \phi^p(x^p_i) = \bigotimes_{p=1}^{P} \Phi_p^T[\cdot, i] = \left(\bigodot_{p=1}^{P} \Phi_p^T \right)[\cdot, i]$$

for the i-th column of Φ^T. Since i is chosen arbitrarily, we already conclude the proof. □

Penalty Matrix

The explicit form of the penalty matrix Λ depends on the utilized regularization term (cf. Subsection 3.2.2). For the truncated power series basis the penalty matrix reads

$$\Lambda = \sum_{p=1}^{P} I_{J_1} \otimes \ldots \otimes I_{J_{p-1}} \otimes D^p \otimes I_{J_{p+1}} \otimes \ldots \otimes I_{J_P} \in \mathbb{R}^{K \times K}, \tag{5.7}$$

whereas the differences penalty matrix for the B-spline basis is given as

$$\Lambda = \sum_{p=1}^{P} I_{J_1} \otimes \ldots \otimes I_{J_{p-1}} \otimes \left(\Delta_{r_p}^{p}\right)^T \Delta_{r_p}^{p} \otimes I_{J_{p+1}} \otimes \ldots \otimes I_{J_P} \in \mathbb{R}^{K \times K}. \tag{5.8}$$

Thus, both penalty matrices are given as the sum of Kronecker matrices (cf. Definition 2.3.1). The dimension of the Kronecker factors of each Kronecker matrix is independent of P such that their storage does not cause any problems. For the curvature penalty (3.53) with B-spline basis, that is

$$\Lambda = \sum_{|r|=2} \frac{2}{r!} \Psi_r \in \mathbb{R}^{K \times K}, \tag{5.9}$$

we aim for a similar representation. Based on the tensor nature of the B-spline basis functions, the following lemma states that each of the Gramian matrices Ψ_r is given as the Kronecker product of the related Gramian matrices in one variable.

Lemma 5.1.2
For the matrices

$$\Psi_{r_p}^{p} \in \mathbb{R}^{J_p \times J_p}, \;\; \Psi_{r_p}^{p}[j_p, \ell_p] = \left\langle \partial^{r_p} \varphi_{j_p,q_p}^{p}, \partial^{r_p} \varphi_{\ell_p,q_p}^{p} \right\rangle_{L^2(\Omega_p)}, \;\; p = 1, \ldots, P, \tag{5.10}$$

it holds

$$\Psi_r = \bigotimes_{p=1}^{P} \Psi_{r_p}^{p} \in \mathbb{R}^{K \times K}. \tag{5.11}$$

Proof. To prove the lemma, we show

$$\Psi_r[k,\ell] = \left(\Psi_{r_1}^{1} \otimes \ldots \otimes \Psi_{r_P}^{P}\right)[k,\ell]$$

for arbitrary $k, \ell \in \{1, \ldots, K\}$. Let ν denote the bijective lexicographical sorting map (3.25) and let $i = (i_1, \ldots, i_P) := \nu^{-1}(k)$ and $j = (j_1, \ldots, j_P) := \nu^{-1}(\ell)$ denote the related inverse images. It holds

$$\begin{aligned}
\Psi_r[k,\ell] = \langle \partial^r \varphi_{i,q}, \partial^r \varphi_{j,q} \rangle_{L^2(\Omega)} &= \int_{\Omega} \partial^r \varphi_{i,q}(x) \partial^r \varphi_{j,q}(x) \mathrm{d}x \\
&= \int_{\Omega_1} \cdots \int_{\Omega_P} \prod_{p=1}^{P} \partial^{r_p} \varphi_{i_p,q_p}^{p}(x^p) \partial^{r_p} \varphi_{j_p,q_p}^{p}(x^p) \mathrm{d}x^P \ldots \mathrm{d}x^1 \\
&= \prod_{p=1}^{P} \left(\int_{\Omega_p} \partial^{r_p} \varphi_{i_p,q_p}^{p}(x^p) \partial^{r_p} \varphi_{j_p,q_p}^{p}(x^p) \mathrm{d}x^p \right) \\
&= \prod_{p=1}^{P} \left\langle \partial^{r_p} \varphi_{i_p,q_p}^{p}, \partial^{r_p} \varphi_{j_p,q_p}^{p} \right\rangle_{L_2(\Omega_p)}
\end{aligned}$$

$$= \left(\bigotimes_{p=1}^{P} \Psi_{r_p}^p \right) [\nu(i), \nu(j)] = \left(\bigotimes_{p=1}^{P} \Psi_{r_p}^p \right) [k, \ell].$$

Since k and ℓ are chosen arbitrarily, we conclude the proof. □

Matrix of the Shape Constraints

The matrices representing the shape constraints (3.91) are defined as the partial derivatives of the tensor product spline basis functions evaluated at an adequate discretization of Ω. Utilizing the tensor nature of the spline basis functions, we obtain a representation of each of the Γ_r as Kronecker product of the one-dimensional counterparts. This is stated by the following lemma.

Lemma 5.1.3
For the matrices

$$\Gamma_{r_p}^p := \begin{bmatrix} \partial^{r_p} \phi_{1,q_p}^p(\tau_1^p) & \dots & \partial^{r_p} \phi_{J_p,q_p}^p(\tau_1^p) \\ \vdots & \ddots & \vdots \\ \partial^{r_p} \phi_{1,q_p}^p(\tau_{M_p}^p) & \dots & \partial^{r_p} \phi_{J_p,q_p}^p(\tau_{M_p}^p) \end{bmatrix} \in \mathbb{R}^{M_p \times J_p}, \; p = 1, \dots, P, \tag{5.12}$$

it holds

$$\Gamma_r = \bigotimes_{p=1}^{P} \Gamma_{r_p}^p \in \mathbb{R}^{T \times K}. \tag{5.13}$$

Proof. For arbitrary $t \in \{1, \dots, T\}$ and arbitrary $k \in \{1, \dots, K\}$, we show

$$\Gamma_r[t, k] = \left(\bigotimes_{p=1}^{P} \Gamma_{r_p}^p \right) [t, k].$$

Let ν denote the bijective lexicographical sorting map (3.25) and let $i = (i_1, \dots, i_P) := \nu^{-1}(t)$ and $j = (j_1, \dots, j_P) := \nu^{-1}(k)$ denote the related inverse images. It holds

$$\begin{aligned} \Gamma_r[t, k] &= \partial^r \phi_{j,q}((\tau_{i_1}^1, \dots, \tau_{i_P}^P)^T) = \prod_{p=1}^{P} \partial^{r_p} \phi_{j_p,q_p}^p(\tau_{i_p}^p) \\ &= \left(\bigotimes_{p=1}^{P} \Gamma_{r_p}^p \right) [\nu(i), \nu(j)] = \left(\bigotimes_{p=1}^{P} \Gamma_{r_p}^p \right) [t, k]. \end{aligned}$$

Since t and k are chosen arbitrarily, we conclude the proof. □

Remark

As previously shown, the memory demanding matrices Φ, Λ, and Γ_r that occur in the large-scale problems (5.1) and (5.2) are given as Khatri-Rao or as Kronecker product of matrices of much smaller dimensions. Especially, the dimensions of the Khatri-Rao and the Kronecker factors do not depend on the number of covariates P, but only the number of factors depends on P. Therefore, the required memory to store these factors is linear in P, which is obviously a great improvement compared to the exponential dependency for storing the full matrices as stated in (5.3). Note that all of the occurring factors are of comparatively small dimensions such that their storage requires a negligible amount of RAM. As a case in point, the storage of the Kronecker factors of Λ for the $P = 3$ case requires the storage of three matrices of dimension 40 times 40, i.e. 4,800 doubles, which is 38400 bytes (= 0.0000384 GB). On the contrary, as already mentioned, the storage of the full matrix Λ in CSC format requires approximately two GB of RAM.

5.1.2 Matrix Operations

In order to develop memory efficient solution algorithms for the large-scale problems (5.1) and (5.2), we aim at exploiting the special structures of the occurring matrices. Therefore, in the following we implement several matrix operations with arbitrary Kronecker and Khatri-Rao matrices that only require the storage of the respective low-dimensional factors.

Khatri-Rao Matrix

For given matrices $A_p \in \mathbb{R}^{m_p \times n}$, $p = 1, \ldots, P$, we consider the Khatri-Rao matrix

$$A := \bigodot_{p=1}^{P} A_p \in \mathbb{R}^{m \times n}, \quad \text{where} \quad m := \prod_{p=1}^{P} m_p. \tag{5.14}$$

Due to the definition of the Khatri-Rao product (cf. Definition 2.3.3), it holds

$$A = \bigodot_{p=1}^{P} A_p = \left[\bigotimes_{p=1}^{P} A_p[\cdot, 1], \ldots, \bigotimes_{p=1}^{P} A_p[\cdot, n] \right] \tag{5.15}$$

such that for all $x \in \mathbb{R}^n$ it follows

$$Ax = \sum_{i=1}^{n} x[i] v_i, \quad \text{where} \quad v_i := \bigotimes_{p=1}^{P} A_p[\cdot, i] \in \mathbb{R}^m \tag{5.16}$$

denotes the i-th column of A. As a result, Algorithm 5.1 allows to compute the matrix-vector product Ax by only accessing the Khatri-Rao factors $A_1, \ldots, A_P$ of A.

Algorithm 5.1: Matrix-vector product with a Khatri-Rao matrix

```
Input: $A_1, \ldots, A_P, x$
Output: $w := (A_1 \odot \ldots \odot A_P)x$
$w \leftarrow 0$
for $i = 1, \ldots, n$ do
    $v \leftarrow A_1[\cdot, i] \otimes \ldots \otimes A_P[\cdot, i]$
    $w \leftarrow w + x[i]v$
end
return $w$
```

Analogously, for all $z \in \mathbb{R}^m$ it holds

$$A^T z = \left(v_1^T z, \ldots, v_n^T z\right)^T, \tag{5.17}$$

which yields Algorithm 5.2 to calculate the matrix-vector product $A^T z$ by only accessing the Khatri-Rao factors $A_1, \ldots, A_P$ of A.

Algorithm 5.2: Matrix-vector product with a transposed Khatri-Rao matrix

```
**Input:** $A_1, \ldots, A_P, z$
**Output:** $w := (A_1 \odot \cdots \odot A_P)^T z$
$w \leftarrow 0$
for $i = 1, \ldots, n$ do
    $v \leftarrow A_1[\cdot, i] \otimes \ldots \otimes A_P[\cdot, i]$
    $w[i] \leftarrow v^T z$
end
return $w$
```

For the aspired development of computationally efficient solution algorithms and the related memory efficient implementation not only matrix-vector products, but also the extraction of diagonal elements of several matrices is required. Recall that these operations still have to be performed matrix-free, i.e. the diagonal of the respective matrix has to be computed without explicitly assembling the matrix itself. In order to compute the diagonal of the matrix product $AA^T \in \mathbb{R}^{m \times m}$, where A still denotes the Khatri-Rao matrix (5.14), let e_j be the j-th unit vector for $j = 1, \ldots, m$. For the j-th diagonal element of AA^T it holds

$$\begin{aligned} (AA)^T [j, j] &= e_j^T AA^T e_j = \|A^T e_j\|_2^2 = \left\| \left(v_1^T e_j, \ldots, v_n^T e_j\right)^T \right\|_2^2 \\ &= \left\| (v_1[j], \ldots, v_n[j])^T \right\|_2^2 = \sum_{i=1}^{n} v_i[j]^2, \end{aligned} \tag{5.18}$$

where v_i is the i-th column of A as in (5.16). Algorithm 5.3 allows for the extraction of the diagonal of AA^T by only accessing the Khatri-Rao factors of A.

Algorithm 5.3: Diagonal extraction for Khatri-Rao products I

```
**Input:** $A_1, \ldots, A_P$
**Output:** $d := \mathrm{diag}(AA^T)$, where $A := A_1 \odot \ldots \odot A_P$
$d \leftarrow 0$
for $i = 1, \ldots, n$ do
    $v \leftarrow A_1[\cdot, i] \otimes \ldots \otimes A_P[\cdot, i]$
    for $j = 1, \ldots, m$ do
        $d[j] \leftarrow d[j] + v[j]^2$
    end
end
return $d$
```

For a further given matrix $B \in \mathbb{R}^{n \times n}$ it analogously holds

$$\begin{aligned} e_j ABA^T e_j &= (v_1[j], \ldots, v_n[j])\, B\, (v_1[j], \ldots, v_n[j])^T \\ &= \sum_{k=1}^{n} v_k[j] \left(\sum_{i=1}^{n} B[k,i] v_i[j] \right) = \sum_{k=1}^{n} \left(\sum_{i=1}^{n} v_k[j] B[k,i] v_i[j] \right) \end{aligned} \tag{5.19}$$

for the j-th diagonal element of the matrix product $ABA^T \in \mathbb{R}^{m \times m}$. The Algorithm 5.4 computes the diagonal of ABA^T by only accessing the Khatri-Rao factors $A_1, \ldots, A_P$ and the matrix B.

Algorithm 5.4: Diagonal extraction for Khatri-Rao products II

```
Input: $A_1, \ldots, A_P, B$
Output: $d := \operatorname{diag}(ABA^T)$, where $A := A_1 \odot \ldots \odot A_P$
for $i = 1, \ldots, n$ do
    $v \leftarrow A_1[\cdot, i] \otimes \ldots \otimes A_P[\cdot, i]$
    for $k = 1, \ldots, n$ do
        $\tilde{v} \leftarrow A_1[\cdot, k] \otimes \ldots \otimes A_P[\cdot, k]$
        for $j = 1, \ldots, m$ do
            $d[j] \leftarrow d[j] + B[k,i]\tilde{v}[j]v[j]$
        end
    end
end
return $d$
```

Kronecker Matrix

For given matrices $A_p \in \mathbb{R}^{m_p \times n_p}$, we now consider the Kronecker matrix

$$A := \bigotimes_{p=1}^{P} A_p \in \mathbb{R}^{m \times n}, \quad \text{where} \quad m := \prod_{p=1}^{P} m_p \quad \text{and} \quad n := \prod_{p=1}^{P} n_p. \tag{5.20}$$

In the case of quadratic Kronecker factors, that is $m_p = n_p$ for all $p = 1, \ldots, P$, Benoit et al. (2001) provide an implementation to compute the matrix-vector product $z^T A$ for arbitrary $z \in \mathbb{R}^m$ by only accessing the Kronecker factors $A_1, \ldots, A_P$. For the solution algorithms developed and implemented within this thesis, however, Kronecker matrices with nonquadratic factors occur. Therefore, we extend their approach to compute the matrix-vector product Ax for all $x \in \mathbb{R}^n$ with arbitrary Kronecker factors in the following. Due to the normal factor decomposition of the Kronecker product (cf. Lemma 2.3.2), it holds

$$A = \prod_{p=1}^{P} \left(I_{n_1} \otimes \ldots \otimes I_{n_{p-1}} \otimes A_p \otimes I_{m_{p+1}} \otimes \ldots \otimes I_{m_P} \right) = \prod_{p=1}^{P} \left(I_{l_p} \otimes A_p \otimes I_{r_p} \right), \tag{5.21}$$

where

$$l_p := \prod_{j=1}^{p-1} n_j, \quad r_p := \prod_{j=p+1}^{P} m_j \tag{5.22}$$

and I denotes the identity matrix of respective dimension. In order to compute the matrix-vector product Ax, it suffices to successively multiply with each normal factor, that is, starting with $v_P := x \in \mathbb{R}^n$, we compute

$$v_{p-1} := \left(I_{l_p} \otimes A_p \otimes I_{r_p}\right) v_p \in \mathbb{R}^{l_p m_p r_p},\ p = P, \ldots, 1, \tag{5.23}$$

and obtain $v_0 = Ax \in \mathbb{R}^m$. To multiply with the p-th normal factor, note that

$$\left(I_{l_p} \otimes A_p \otimes I_{r_p}\right) v_p = \left(I_{l_p} \otimes R_p\right) v_p, \tag{5.24}$$

where

$$R_p := A_p \otimes I_{r_p} \in \mathbb{R}^{m_p r_p \times n_p r_p}. \tag{5.25}$$

The matrix $I_{l_p} \otimes R_p$ is a block-diagonal matrix consisting of l_p blocks containing the matrix R_p in each block. We decompose the vector v_p into the l_p chunks $\bar{v}_{p,1}, \ldots, \bar{v}_{p,l_p}$, each of length $n_p r_p$, that is

$$v_p = \begin{pmatrix} \bar{v}_{p,1} \\ \vdots \\ \bar{v}_{p,l_p} \end{pmatrix}, \quad \bar{v}_{p,s} := \bar{v}_s[(s-1)n_p r_p + 1 : s n_p r_p] := \begin{pmatrix} v_p[(s-1)n_p r_p + 1] \\ \vdots \\ v_p[s n_p r_p] \end{pmatrix} \in \mathbb{R}^{n_p r_p} \tag{5.26}$$

for $s = 1, \ldots, l_p$. This yields

$$\left(I_{l_p} \otimes A_p \otimes I_{r_p}\right) v_p = \begin{pmatrix} R_p \bar{v}_{p,1} \\ \vdots \\ R_p \bar{v}_{p,l_p} \end{pmatrix} = \begin{pmatrix} w_{p,1} \\ \vdots \\ w_{p,l_p} \end{pmatrix}, \tag{5.27}$$

where

$$w_{p,s} := R_p \bar{v}_{p,s} \in \mathbb{R}^{m_p r_p}. \tag{5.28}$$

In order to compute $w_{p,s}$ for $s = 1, \ldots, l_p$, note that

$$R_p = A_p \otimes I_{r_p} = \begin{bmatrix} A_p[1,1] I_{r_p} & \ldots & A_p[1,n_p] I_{r_p} \\ \vdots & \ddots & \vdots \\ A_p[m_p,1] I_{r_p} & \ldots & A_p[m_p,n_p] I_{r_p} \end{bmatrix} \tag{5.29}$$

such that each row of R_p consist of the elements of a row of A_p at distance r_p apart and zeros for the rest. We define the vectors

$$z_{p,s,t} := \bar{v}_{p,s}[t, t + r_p, \ldots, t + (n_p - 1) r_p] := \begin{pmatrix} \bar{v}_{p,s}[t] \\ \bar{v}_{p,s}[t + r_p] \\ \vdots \\ \bar{v}_{p,s}[t + (n_p - 1) r_p] \end{pmatrix} \in \mathbb{R}^{n_p} \tag{5.30}$$

for $t = 1, \ldots, r_p$, such that each of the matrix-vector products

$$\bar{z}_{p,s,t} := A_p z_{p,s,t} \in \mathbb{R}^{m_p} \tag{5.31}$$

provides m_p elements of the vector $w_{p,s}$, but at incorrect positions. Since the true positions are at distance r_p apart, we deduce

$$w_{p,s}[t, t + r_p, \ldots, t + (m_p - 1)r_p] := \begin{pmatrix} w_{p,s}[t] \\ w_{p,s}[t + r_p] \\ \vdots \\ w_{p,s}[t + (m_p - 1)r_p] \end{pmatrix} = \bar{z}_{p,s,t}. \tag{5.32}$$

Finally, looping backwards over all normal factors of A leads to Algorithm 5.5 to compute the matrix-vector product Ax by only accessing the Kronecker factors of A.

Algorithm 5.5: Matrix-vector product with a Kronecker matrix

```
Input: $A_1, \ldots, A_P, x$
Output: $v_0 := (A_1 \otimes \ldots \otimes A_P)x$
$v_P \leftarrow x$
for $p = P, \ldots, 1$ do                                   // loop over normal factors
    for $s = 1, \ldots, l_p$ do                                  // loop over chunks
        $\bar{v}_{p,s} \leftarrow v_p[(s-1)n_p r_p + 1 : s n_p r_p]$
        $w_{p,s} \leftarrow 0$
        for $t = 1, \ldots, r_p$ do                               // chunk details
            $z_{p,s,t} \leftarrow \bar{v}_{p,s}[t, t + r_p, \ldots, t + (n_p - 1)r_p]$
            $\bar{z}_{p,s,t} \leftarrow A_p z_{p,s,t}$
            $w_{p,s}[t, t + r_p, \ldots, t + (m_p - 1)r_p] \leftarrow \bar{z}_{p,s,t}$
        end
        $v_{p-1}[(s-1)m_p r_p + 1 : s m_p r_p] \leftarrow w_{p,s}$
    end
end
return $v_0$
```

As already mentioned, for the aspired development of computationally efficient solution algorithms and the related memory efficient implementation not only matrix-vector products, but also the extraction of diagonal elements of several matrices is required. Due to the properties of the Kronecker product (cf. Lemma 2.3.2), namely the distributivity of transposition, the mixed product property, and the diagonal property, it holds

$$\operatorname{diag}(A^T A) = \bigotimes_{p=1}^{P} \operatorname{diag}\left(A_p^T A_p\right). \tag{5.33}$$

Algorithm 5.6 extracts the diagonal of the matrix $A^T A \in \mathbb{R}^{n \times n}$ by only accessing the Kronecker factors $A_1, \ldots, A_P$.

```
Algorithm 5.6: Diagonal extraction for Kronecker products I
Input: $A_1, \dots, A_P$
Output: $d := \operatorname{diag}(A^T A)$, where $A := A_1 \otimes \dots \otimes A_P$
for $p = 1, \dots, P$ do
    $d_p \leftarrow \operatorname{diag}\left(A_p^T A_p\right)$
end
return $d \leftarrow d_1 \otimes \dots \otimes d_P$
```

Let $W := \operatorname{diag}(w_1, \dots, w_m) \in \mathbb{R}^{m \times m}$ denote an arbitrary diagonal matrix. The j-th diagonal element of $A^T W A$ is given as

$$e_j^T A^T W A e_j = (A[\cdot, j])^T W A[\cdot, j] = \sum_{k=1}^{m} w_k A[k, j]^2, \tag{5.34}$$

where e_j denotes the j-th unit vector and

$$A[\cdot, j] = \left(\bigotimes_{p=1}^{P} A_p \right) [\cdot, j] = \bigotimes_{p=1}^{P} (A_p[\cdot, i_p]) \tag{5.35}$$

with $i = (i_1, \dots, i_P) := \nu^{-1}(j)$ denoting the inverse image of the lexicographical sorting map (3.25). Algorithm 5.7 computes the diagonal of $A^T W A$ while assembling neither A nor W.

```
Algorithm 5.7: Diagonal extraction for Kronecker products II
Input: $A_1, \dots, A_P$, $w_1, \dots, w_m$
Output: $d := \operatorname{diag}(A^T W A)$, where $A := A_1 \otimes \dots \otimes A_P$ and $W := \operatorname{diag}(w_1, \dots, w_m)$
for $j = 1, \dots, n$ do
    $i \leftarrow \nu^{-1}(j)$
    $v \leftarrow A_1[\cdot, i_1] \otimes \dots \otimes A_P[\cdot, i_P]$
    $d[j] \leftarrow \sum_{k=1}^{m} w_k v[k]^2$
end
return $d$
```

5.2 Matrix-Free Conjugate Gradient Method

Section 5.1 provides several matrix operations with special types of matrices, namely Kronecker and Khatri-Rao matrices, that do not require their explicit storage. Based on these memory efficient implementations, we develop a matrix-free solution algorithm for the large-scale linear system (5.1) in this section. Due to Lemma 4.4.1, the coefficient matrix

$$\Phi^T \Phi + \lambda_s \Lambda - \Phi^T C \Phi \in \mathbb{R}^{K \times K} \tag{5.36}$$

of this linear system is symmetric and positive definite such that an appropriate method to solve the large-scale linear system (5.1) is the CG method (cf. Algorithm 2.4). The crucial part of this approach is the computation of a matrix-vector product with the coefficient matrix (5.36) in each CG iteration such that a straightforward application is computationally infeasible. The CG algorithm, however, does not require the explicit storage of the coefficient matrix, but only needs to access it by forming matrix-vector products, i.e. it can be implemented as a matrix-free method. For that purpose, we require the implementation of matrix-vector products with the matrices Φ^T, Φ, and Λ without explicitly storing them. Due to Lemma 5.1.1, the matrix Φ^T is given as a Khatri-Rao product such that the required matrix-vector product is achieved by Algorithm 5.1. Further, we obtain the right-hand side $\Phi^T(y - Cy)$ of the linear system (5.1) by Algorithm 5.1 as well. Algorithm 5.2 provides the required matrix-vector product with the transposed Khatri-Rao matrix Φ. In order to multiply with the penalty matrix Λ, we remark that the penalty matrix is given as a (weighted) sum of Kronecker matrices. A matrix-vector product with Λ is therefore computed by the repetitive application of Algorithm 5.5. In conclusion, we present a memory efficient implementation of the matrix-free CG method to solve the large-scale linear system (5.1) in Algorithm 5.8.

Algorithm 5.8: Conjugate gradient method for the linear system (5.1)

```
$\alpha \leftarrow 0$
$p \leftarrow r \leftarrow \Phi^T(y - Cy)$                          // apply Algorithm 5.1
while $\|r\|_2^2 > tol$ do
    $v \leftarrow \left(\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right) p$        // apply Algorithm 5.2, 5.1, and 5.5
    $w \leftarrow \|r\|_2^2 / p^T v$
    $\alpha \leftarrow \alpha + wp$
    $\tilde{r} \leftarrow r$
    $r \leftarrow r - wv$
    $p \leftarrow r + (\|r\|_2^2 / \|\tilde{r}\|_2^2) p$
end
return $\alpha$
```

Note that for a fixed P the Algorithm 5.8 solely requires the storage of all occurring Kronecker and Khatri-Rao factors. These storage demands, however, are a priori fixed and comparatively small, especially when a sparse storage format is used. Thus, the memory requirement depends linearly on P and is therefore $\mathcal{O}(P)$. This is a tremendous improvement compared to $\mathcal{O}(2^P)$ required by the naive implementation of the CG method in the full matrix approach. The application of the matrix-free CG method allows to determine the (unconstrained) regression P-spline as well as the (unconstrained) small area P-spline for increasing covariate dimension P with a negligible amount of RAM. The convergence rate of the CG method, however, heavily depends on the condition number of the coefficient matrix and due to its construction, we have to expect that the matrix $\Phi^T\Phi + \lambda\Lambda - \Phi^T C\Phi$ is of rather poor condition. We therefore have to face tremendously increasing computational complexity with increasing covariate dimensions P. This demands for adequate preconditioning methods in order to make the matrix-free CG method more practicable (cf. Subsection 2.4.3). These preconditioning methods, however, also have to be matrix-free which prevents the application of widely used incomplete factorization methods such as the incomplete Cholesky factorization (cf. Nocedal and Wright, 2006, pp. 119-120). The memory efficient implementation of an adequate preconditioner for the matrix-free CG method is further addressed in Section 5.4.

5.3 Matrix-Free Newton Method for the Quadratic Penalty Approach

The matrix-free CG method (cf. Algorithm 5.8) provides a suitable approach to determine the coefficients of an unconstrained (small-area) P-spline with an arbitrary number of covariates. This section is now devoted to the memory efficient implementation of solution algorithms for the large-scale strictly convex QP (5.2) in order to allow also for the determination of a shape constrained (small-area) P-spline in arbitrary covariate dimensions. For the memory efficient implementation of a matrix-free solution algorithm for the large-scale optimization problem (5.2), we apply a quadratic penalty method to transform the strictly convex QP into an unconstrained convex optimization problem (cf. Section 2.5). We then apply a Newton method to solve the resulting convex problem, where in each Newton step the related linear system is solved by a matrix-free CG method similarly to Algorithm 5.8.

5.3.1 Quadratic Penalty Problem and Newton Method

According to (2.71), we define the objective function $h_c\colon \mathbb{R}^K \to \mathbb{R}$ of the quadratic penalty method for the strictly convex QP (5.2) as

$$\begin{aligned} h_c(\alpha) :=\ & \frac{1}{2}\alpha^T \left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right]\alpha - \left[\Phi^T\left[y - Cy\right]\right]^T \alpha \\ & + \frac{c}{2}\left[\sum_{r\in I_\leq}\sum_{t=1}^{T}\max\{0, \Gamma_r[t,\cdot]\alpha\}^2 + \sum_{r\in I_\geq}\sum_{t=1}^{T}\max\{0, -\Gamma_r[t,\cdot]\alpha\}^2\right] \\ =\ & \frac{1}{2}\alpha^T \left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right]\alpha - \left[\Phi^T\left[y - Cy\right]\right]^T \alpha \\ & + \frac{c}{2}\alpha^T\left[\sum_{r\in I_\leq}\Gamma_r^T W_{r,\leq}(\alpha)\Gamma_r + \sum_{r\in I_\geq}\Gamma_r^T W_{r,\geq}(\alpha)\Gamma_r\right]\alpha. \end{aligned} \tag{5.37}$$

The matrices $W_{r,\leq}(\alpha)$ and $W_{r,\geq}(\alpha)$ thereby denote diagonal matrices of dimension $T \times T$ with diagonal elements

$$W_{r,\leq}(\alpha)[t,t] := \begin{cases} 1\ , & \text{if } \Gamma_r[t,\cdot]\alpha > 0 \\ 0\ , & \text{else} \end{cases}\quad ,\ W_{r,\geq}(\alpha)[t,t] := \begin{cases} 1\ , & \text{if } \Gamma_r[t,\cdot]\alpha < 0 \\ 0\ , & \text{else} \end{cases}\quad . \tag{5.38}$$

To simplify the notation, we define the matrix valued function

$$M\colon \mathbb{R}^K \to \mathbb{R}^{K\times K},\ \alpha \mapsto \sum_{r\in I_\leq}\Gamma_r^T W_{r,\leq}(\alpha)\Gamma_r + \sum_{r\in I_\geq}\Gamma_r^T W_{r,\geq}(\alpha)\Gamma_r \tag{5.39}$$

such that the quadratic penalty objective function (5.37) reads

$$h_c(\alpha) = \frac{1}{2}\alpha^T\left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi + cM(\alpha)\right]\alpha - \left[\Phi^T\left[y - Cy\right]\right]^T\alpha. \tag{5.40}$$

To apply the quadratic penalty method (cf. Algorithm 2.7), we have to repeatedly solve the unconstrained large-scale optimization problem

$$\min_{\alpha \in \mathbb{R}^K} h_c(\alpha) \tag{5.41}$$

for systematically increasing $c > 0$. For that purpose, we first reveal some characteristics of the matrix valued function (5.39) in the following lemma.

Lemma 5.3.1
For the matrix valued function M and for all $\alpha \in \mathbb{R}^K$ it holds:

1. $M(\alpha) \succeq 0$,
2. $\partial_\alpha M(\alpha) = 0$.

Proof. Let $\alpha, \tilde{\alpha} \in \mathbb{R}^K$, $* \in \{\leq, \geq\}$, and $r \in I_{\leq} \cup I_{\geq}$ all be arbitrary. For $v := \Gamma_r \tilde{\alpha}$ it holds

$$\tilde{\alpha}^T \Gamma_r^T W_{r,*}(\alpha) \Gamma_r \tilde{\alpha} = v^T W_{r,*}(\alpha) v = \sum_{t=1}^{T} W_{r,*}(\alpha)[t,t] v_t^2 \geq 0$$

and we conclude $M(\alpha) \succeq 0$ for all $\alpha \in \mathbb{R}^K$. To prove the second statement, it suffices to show that $\partial_\alpha W_{r,*}(\alpha) = 0$. By definition of the Fréchet derivative it holds

$$\partial_\alpha W_{r,*}(\alpha) := \lim_{\ell \to 0} \frac{1}{\ell} \left[W_{r,*}(\alpha + \ell\alpha) - W_{r,*}(\alpha) \right]$$

and for sufficiently small $|\ell|$ it holds

$$\operatorname{sgn}(\Gamma_r[t,\cdot]\alpha + \ell \Gamma_r[t,\cdot]\alpha) = \operatorname{sgn}(\Gamma_r[t,\cdot]\alpha), \; t = 1, \ldots, T,$$

where $\operatorname{sgn}(\cdot)$ denotes the algebraic sign of a real number. This yields $W_{r,*}(\alpha + \ell\alpha) = W_{r,*}(\alpha)$ for $|\ell|$ sufficiently small, which implies

$$\partial_\alpha W_{r,*}(\alpha) := \lim_{\ell \to 0} \frac{1}{\ell} \left[W_{r,*}(\alpha + \ell\alpha) - W_{r,*}(\alpha) \right] = \lim_{\ell \to 0} \frac{0}{\ell} = 0$$

and we conclude the proof. □

Due to Lemma 5.3.1, the objective function h_c of the quadratic penalty method possesses the following properties:

1. $\nabla h_c(\alpha) = \left[\Phi^T \Phi + \lambda_s \Lambda - \Phi^T C \Phi + cM(\alpha) \right] \alpha + c\alpha^T \left[\partial_\alpha M(\alpha) \right] \alpha - \Phi^T \left[y - Cy \right]$

   $= \left[\Phi^T \Phi + \lambda_s \Lambda - \Phi^T C \Phi + cM(\alpha) \right] \alpha - \Phi^T \left[y - Cy \right],$

2. $\nabla^2 h_c(\alpha) = \Phi^T \Phi + \lambda_s \Lambda + c \left[\partial_\alpha M(\alpha) \right] \alpha - \Phi^T C \Phi + cM(\alpha)$

   $= \Phi^T \Phi + \lambda_s \Lambda - \Phi^T C \Phi + cM(\alpha),$

3. $\nabla^2 h_c(\alpha) \succ 0 \;\forall\; \alpha \in \mathbb{R}^K$.

Since $\nabla^2 h_c(\alpha) \succ 0$ for all $\alpha \in \mathbb{R}^K$, i.e. h_c is strictly convex, the optimization problem (5.41) is an unconstrained strictly convex optimization problem. The quadratic penalty method (cf. Algorithm 2.7) requires the repeated solution of the large-scale unconstrained strictly convex optimization problem (5.41) for systematically increasing parameters c. Since h_c is a twice continuously differentiable and strictly convex function, we aim for a matrix-free implementation of the Newton method (cf. Algorithm 2.8). This requires the determination of the Newton direction in each iteration step, i.e. the solution of the linear system

$$\nabla^2 h_c(\alpha) d \overset{!}{=} -\nabla h_c(\alpha) \tag{5.42}$$

with fixed and given $\alpha \in \mathbb{R}^K$. This is a large-scale linear system with symmetric and positive definite coefficient matrix

$$\nabla^2 h_c(\alpha) = \Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi + cM(\alpha) \in \mathbb{R}^{K\times K} \tag{5.43}$$

and right-hand side

$$-\nabla h_c(\alpha) = -\left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi + cM(\alpha)\right]\alpha + \Phi^T\left[y - Cy\right] \in \mathbb{R}^K. \tag{5.44}$$

In order to solve the linear system (5.42), we aim for a memory efficient implementation of a matrix-free CG method in analogy to Algorithm 5.8.

5.3.2 Conjugate Gradient Method for the Newton Direction

Comparing the linear system (5.42) to determine the Newton direction in the quadratic penalty method to the linear system (5.1) for the unconstrained small area P-spline it can be seen that the coefficient matrices (5.36) and (5.43) solely differ by the term $cM(\alpha)$. Thus, to adapt the matrix-free CG method for the linear system (5.1) to the linear system (5.42), we additionally require to implement memory efficient matrix-vector products with the matrix $M(\alpha)$. By definition it holds

$$M(\alpha) = \sum_{r\in I_\leq} \Gamma_r^T W_{r,\leq}(\alpha)\Gamma_r + \sum_{r\in I_\geq} \Gamma_r^T W_{r,\geq}(\alpha)\Gamma_r \tag{5.45}$$

and Lemma 5.1.3 yields

$$\Gamma_r = \bigotimes_{p=1}^{P} \Gamma_{r_p}^p. \tag{5.46}$$

Remark that all of the Kronecker factors of Γ_r fit into the internal memory and further that the storage of the diagonal matrices $W_{r,\geq}(\alpha)$ and $W_{r,\leq}(\alpha)$ only requires to store the respective diagonal elements. Therefore, we can compute a matrix-vector product with $M(\alpha)$ by the repetitive application of Algorithm 5.5. This allows to compute the right-hand side of the linear system (5.42) as well as the memory efficient implementation of a matrix-free CG method to determine the Newton direction as solution of (5.42). The procedure is presented in Algorithm 5.9.

Algorithm 5.9: Conjugate gradient method for the linear system (5.42)

```
$d \leftarrow 0$
$p \leftarrow r \leftarrow -\left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi + cM(\alpha)\right]\alpha + \Phi^T\left[y - Cy\right]$    // apply Alogrithm 5.2, 5.1, and 5.5
while $\|r\|_2^2 > tol$ do
    $v \leftarrow \left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi + cM(\alpha)\right] p$    // apply Algorithm 5.2, 5.1, and 5.5
    $w \leftarrow \|r\|_2^2 / p^T v$
    $d \leftarrow d + wp$
    $\tilde{r} \leftarrow r$
    $r \leftarrow r - wv$
    $p \leftarrow r + (\|r\|_2^2 / \|\tilde{r}\|_2^2)p$
end
return $d$
```

This memory efficient implementation of a matrix-free CG method to determine the Newton direction finally enables to solve the large-scale strictly convex QP (5.2). Especially, it allows for the determination of the shape constrained regression P-spline as well as the shape constrained small area P-spline for increasing covariate dimension P. The final procedure is presented in Algorithm 5.10.

Algorithm 5.10: Newton method for the quadratic penalty formulation of the strictly convex QP (5.2)

```
$\alpha \leftarrow \left[\Phi^T\Phi + \lambda_s\Lambda - \Phi^T C\Phi\right]^{-1}\Phi^T\left[y - Cy\right]$    // apply Algorithm 5.8
while stopping criterion not reached do
    $c \leftarrow \eta c$
    while $\|\nabla h_c(\alpha)\|_2^2 > tol$ do
        $d \leftarrow -\nabla^2 h_c(\alpha)^{-1}\nabla h_c(\alpha)$    // apply Algorithm 5.9
        compute $v$    // backtracking line search
        $\alpha \leftarrow \alpha + vd$
    end
end
return $\alpha$
```

As for Algorithm 5.8, the memory requirements of Algorithm 5.9 are $\mathcal{O}(P)$, but the condition of the underlying coefficient matrix is again expected to be poor. The following section is therefore devoted to the memory efficient implementation of a preconditioning method for the both matrix-free CG algorithms.

5.4 Matrix-Free Multigrid Preconditioner

Both of the developed methods to solve the large-scale problems (5.1) and (5.2) require the application of a matrix-free CG method presented in Algorithm 5.8 and Algorithm 5.9. The convergence rate of the CG method mainly depends on the condition number of the coefficient matrix of the considered linear system (cf. Subsection 2.4.3) and we have to expect that the coefficient matrices are of rather poor condition. To improve the computational complexity and therefore the runtime of the matrix-free CG methods, we aim for the memory efficient implementation of adequate preconditioning methods to apply the preconditioned CG method (cf. Algorithm 2.5). Since the considered coefficient matrices cannot be stored, the preconditioning methods also have to be matrix-free. We therefore implement a matrix-free v-cycle (cf. Algorithm 2.2) for both considered linear systems, leading to a matrix-free version of the multigrid preconditioned conjugate gradient method as presented in Algorithm 2.6. In order to apply a v-cycle to the linear systems, we require the following fundamental components (cf. Subsection 2.4.2):

1. Hierarchization: Introduction of a hierarchy of, in a geometrical sense, coarsening representations of the initial linear system, based on different discretization grid levels.
2. Smoothing iteration: Application of a relaxation method to the linear system at each grid level.
3. Grid transfer: Prolongation and restriction operations in order to transfer information between the grids.
4. Coarse grid solver: Solution method for the linear system at the coarsest grid level.

Note that none of these operations can be performed by storing the upcoming matrices, but need to be implemented as matrix-free methods. For the implementation of a matrix-free MGCG method, we have to restrict ourselves to the uniform B-spline basis with curvature penalty as introduced in Subsection 3.2.2. That is:

1. $\Lambda = \sum_{|r|=2} \frac{2}{r!} \Psi_r \in \mathbb{R}^{K\times K}$, where $\Psi_r[k,\ell] := \langle \partial^r \varphi_{k,q}, \partial^r \varphi_{\ell,q} \rangle_{L^2(\Omega)}$,
2. $\Phi \in \mathbb{R}^{n\times K}$, where $\Phi[i,k] := \varphi_{k,q}(x_i)$,
3. $\Gamma_r \in \mathbb{R}^{T\times K}$, where $\Gamma_r[t,k] := \partial^r \varphi_{k,q}(\tau_t)$.

To implement a v-cycle for the linear systems, we introduce a maximum grid level $G \in \mathbb{N}$ and base the spline space $\mathcal{S}_q(\mathcal{K})$ on

$$m^p := m^p_G := 2^G - 1,\ p = 1, \dots, P, \tag{5.47}$$

equally spaced knots. We aim at solving the large-scale linear system

$$\left[(\Phi_G)^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G \right] \alpha \overset{!}{=} (\Phi_G)^T [y - Cy] \tag{5.48}$$

for α and the large-scale linear system

$$\begin{aligned} &\left[(\Phi_G)^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G + c M_G(\alpha) \right] d \\ \overset{!}{=} -&\left[(\Phi_G)^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G + c M_G(\alpha) \right] \alpha + (\Phi_G)^T [y - Cy]. \end{aligned} \tag{5.49}$$

for d (with given α). As mentioned in Subsection 5.3.2, the both systems coincide if $\alpha = 0$ in the latter system. Especially, the linear system (5.49) can be seen as a generalization of the linear system (5.48) such that we focus on a memory efficient implementation of a matrix-free MGCG method for the more general problem (5.49). For a more convenient notation, we define

$$A_G := \begin{cases} (\Phi_G)^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G + c M_G(\alpha), & \text{for system (5.49)} \\ \Phi_G^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G, & \text{for system (5.48)} \end{cases} \tag{5.50}$$

and

$$b := \begin{cases} -\left[(\Phi_G)^T \Phi_G + \lambda_s \Lambda_G - (\Phi_G)^T C \Phi_G + c M_G(\alpha)\right] \alpha + (\Phi_G)^T [y - Cy], & \text{for (5.49)} \\ (\Phi_G)^T [y - Cy], & \text{for (5.48)} \end{cases} \tag{5.51}$$

and consider the linear system

$$A_G x \overset{!}{=} b. \tag{5.52}$$

Hierarchization

In order to achieve an adequate hierarchization of the linear system (5.52), we define the nested sequence of knot sets

$$\mathcal{K}_g \subset \mathcal{K}_{g+1}, \ g = 1, \ldots, G-1, \tag{5.53}$$

where $\mathcal{K}_g$ consists of $m_g^p := 2^g - 1$ equidistant knots for all $p = 1, \ldots, P$. This hierarchy of knot sets yields a hierarchy of the related tensor product spline spaces

$$\mathcal{S}_q(\mathcal{K}_g) \subset \mathcal{S}_q(\mathcal{K}_{g+1}), \ g = 1, \ldots, G-1, \tag{5.54}$$

and subsequently a hierarchy of the matrices Φ_g, Λ_g, and $M_g(\alpha)$. This finally leads to a hierarchy of coefficient matrices A_g, $g = 1, \ldots, G$. The subscript grid number g indicates that the respective matrix is obtained by using the B-spline basis of $\mathcal{S}_q(\mathcal{K}_g)$. Note that $y \in \mathbb{R}^n$, $\lambda_s \in \mathbb{R}$, and $C \in \mathbb{R}^{n \times n}$ are independent of the grid level g. With

$$K_g := \dim(\mathcal{S}_q(\mathcal{K}_g)) = \prod_{p=1}^{P} (2^g + q_p), \ g = 1, \ldots, G, \tag{5.55}$$

we denote the dimension of the quadratic coefficient matrices A_g at the grid level g.

Smoothing Iteration

Smoothing iterations in the multigrid context, as introduced in Subsection 2.4.1, are based on a splitting of the coefficient matrix A_g for each grid level $g = 1, \ldots, G$. Since these coefficient matrices are not explicitly available, the required splitting matrices are neither. The application of the Jacobi method (cf. Algorithm 2.1), however, solely requires multiplications with the coefficient matrix and its diagonal. Since the respective diagonals are vectors of length K_g, $g = 2, \ldots, G$, their memory demand is comparatively small. To efficiently extract the desired diagonals, we apply Algorithm 5.3 to compute the diagonal of $(\Phi_g)^T \Phi_g$, Algorithm 5.4 for the diagonal of $(\Phi_g)^T C \Phi_g$, Algorithm 5.6 to each term of Λ_g to obtain its diagonal, and finally Algorithm 5.7 to each term of $M_g(\alpha)$ for the last diagonal. A memory efficient implementation of the matrix-free Jacobi method as smoothing iteration for the aspired v-cycle is presented in Algorithm 5.11.

Algorithm 5.11: JAC: Jacobi smoothing for the v-cycle

```
JAC($x, b, g, \nu$)
    $D \leftarrow 1/\mathrm{diag}(A_g)$            // apply Algorithm 5.3, 5.4, 5.6, and 5.7
    for $1, \dots, \nu$ do
        $r \leftarrow b - A_g x$                  // apply Algorithm 5.2, 5.1, and 5.5
        $x \leftarrow x + \omega D^T r$
    end
end
return $x$
```

The application of the SSOR method as smoothing iteration additionally requires the triangular part of the coefficient matrix which does not fit into the storage. It is possible to compute the desired elements entry-wise within each smoothing iteration, but this is computationally very expensive since these elements have to be computed repeatedly in each iteration and for each grid level. Therefore, we commit to the memory efficient implementation of the Jacobi method as smoothing iteration. Nevertheless, we point out the differences of both smoothing iterations in Section 5.5.

Grid Transfer Operations

To transfer vectors between the different grid levels, we require prolongation and restriction matrices

$$I_g^{g+1} \in \mathbb{R}^{K_{g+1} \times K_g} \quad \text{and} \quad I_{g+1}^{g} \in \mathbb{R}^{K_g \times K_{g+1}}, \ g = 1, \dots, G-1. \tag{5.56}$$

In (3.33), we provide adequate matrices to exactly transfer B-spline coefficients from a coarser to a finer grid. Note that for this reason we restrict ourselves to the uniform B-spline basis for the implementation of the MGCG method. According to Lemma 3.1.8, we define the prolongation matrices

$$I_g^{g+1} := I_{h_g}^{h_{g+1}} \in \mathbb{R}^{K_{g+1} \times K_g}, \ g = 1, \dots, G-1, \tag{5.57}$$

where

$$h_g := \left((m_g^1 - 1)^{-1}, \dots, (m_g^P - 1)^{-1} \right)^T, \ g = 1, \dots, G, \tag{5.58}$$

denotes the vector of mesh sizes for the different grids. Based on the suggestions in Subsection 2.4.2, we define the restriction matrices as

$$I_{g+1}^{g} := \left(I_g^{g+1} \right)^T \in \mathbb{R}^{K_g \times K_{g+1}}, \ g = 1, \dots, G-1, \tag{5.59}$$

which yields the desired Garlerkin property as stated by the following lemma.

Lemma 5.4.1
The Garlerkin property holds for the determined choice of restriction and prolongation matrices, that is

$$A_g = I_{g+1}^{g} A_{g+1} I_{g}^{g+1} \tag{5.60}$$

for $g = 1, \dots, G-1$.

Proof. Let $g \in \{1, \dots, G-1\}$ be arbitrary. Since $I_{g+1}^{g} = \left(I_{g}^{g+1}\right)^T$ and due to the definition of Λ_g and $M_g(\alpha)$ it suffices to show:

1. $\Phi_g = \Phi_{g+1} I_g^{g+1}$,
2. $\Psi_{r,g} = I_{g+1}^{g} \Psi_{r,g+1} I_g^{g+1}$,
3. $\Gamma_{r,g} = \Gamma_{r,g+1} I_g^{g+1}$.

Let $\alpha^g \in \mathbb{R}^{K_g}$ be arbitrary and let

$$s := \sum_{k=1}^{K_g} \alpha_k^g \varphi_{k,q}^g \in \mathcal{S}_q(\mathcal{K}_g)$$

denote the spline with B-spline coefficients α^g. Due to Lemma 3.1.8, it holds

$$s = \sum_{k=1}^{K_{g+1}} \alpha_k^{g+1} \varphi_{k,q}^{g+1} \in \mathcal{S}_q(\mathcal{K}^{g+1})$$

for $\alpha^{g+1} := I_g^{g+1} \alpha^g \in \mathbb{R}^{K_{g+1}}$. This yields

$$\Phi_g \alpha^g = \Phi_{g+1} \alpha^{g+1} = \Phi_{g+1} I_g^{g+1} \alpha^g$$

such that $\Phi_g = \Phi_{g+1} I_g^{g+1}$ holds, since α^g is arbitrarily chosen. Further, it holds

$$\begin{aligned}
\left(\alpha^g\right)^T \Psi_{r,g} \alpha^g = \|\partial^r s\|_{L^2(\Omega)}^2 &= \left\langle \sum_{k=1}^{K_{g+1}} \alpha_k^{g+1} \partial^r \varphi_{k,q}^{g+1}, \sum_{k=1}^{K_{g+1}} \alpha_k^{g+1} \partial^r \varphi_{k,q}^{g+1} \right\rangle_{L^2(\Omega)} \\
&= \sum_{k=1}^{K_{g+1}} \sum_{\ell=1}^{K_{g+1}} \alpha_k^{g+1} \alpha_\ell^{g+1} \left\langle \partial^r \varphi_{k,q}^{g+1}, \partial^r \varphi_{\ell,q}^{g+1} \right\rangle_{L^2(\Omega)} \\
&= \left(\alpha^{g+1}\right)^T \Psi_{r,g+1} \alpha^{g+1} \\
&= \left(\alpha^g\right)^T I_{g+1}^{g} \Psi_{r,g+1} I_g^{g+1} \alpha^g.
\end{aligned}$$

Since α^g is arbitrarily chosen, we conclude $\Psi_{r,g} = I_{g+1}^{g} \Psi_{r,g+1} I_g^{g+1}$. Finally, it holds

$$\Gamma_{r,g} \alpha^g = \begin{pmatrix} \partial^r s(\tau_1) \\ \vdots \\ \partial^r s(\tau_T) \end{pmatrix} \text{ and } \Gamma_{r,g+1} I_g^{g+1} \alpha^g = \Gamma_{r,g+1} \alpha^{g+1} = \begin{pmatrix} \partial^r s(\tau_1) \\ \vdots \\ \partial^r s(\tau_T) \end{pmatrix},$$

which yields $\Gamma_{r,g} = \Gamma_{r,g+1} I_g^{g+1}$, since α^g is arbitrarily chosen. □

Note that the Garlerkin property is not fulfilled for the other penalties introduced in Subsection 3.2.2. For this reason, we restrict ourselves to the curvature penalty for the implementation of the MGCG method. The grid transfer matrices depend on K_g and therefore do not fit into the working memory. But, due to Lemma 3.1.8, it holds

$$I_g^{g+1} = I_{h_g}^{h_{g+1}} = \bigotimes_{p=1}^{P} I_{2h_p}^{h_p} \tag{5.61}$$

such that matrix-vector products with the prolongation matrices I_g^{g+1} are achieved by Algorithm 5.5. Due to the distributivity of transposition (cf. Lemma 2.3.2), it further holds

$$I_{g+1}^{g} = \left(I_g^{g+1}\right)^T = \left(\bigotimes_{p=1}^{P} I_{2h_p}^{h_p}\right)^T = \bigotimes_{p=1}^{P} \left(I_{2h_p}^{h_p}\right)^T \tag{5.62}$$

such that matrix-vector products with the restriction matrices I_{g+1}^{g} are performed by Algorithm 5.5 as well.

Coarse Grid Solver

The implementation of a v-cycle requires the solution of the residual equation on the coarsest grid, that is

$$A_1 e \overset{!}{=} z \tag{5.63}$$

for a given right-hand side vector $z \in \mathbb{R}^{K_1}$. This linear system is of the same form as the initial linear system (5.42) such that we can apply the matrix-free CG method as implemented in Algorithm 5.8 and 5.9, respectively, to obtain the coarse grid solution. This procedure is presented in Algorithm 5.12. Note that the CG method as coarse grid solver causes only a fraction of the computational costs of the CG method as solver for the initial linear system, which is due to $K_1 \ll K_G$.

Algorithm 5.12: `CG_coarse`: Conjugate gradient method as coarse grid solver for the v-cycle

```
CG_coarse($A_1, z$)
    $e \leftarrow 0$
    $p \leftarrow r \leftarrow z$
    while $\|r\|_2^2 > tol$ do
        $v \leftarrow A_1 p$                          // apply Algorithm 5.2, 5.1, and 5.5
        $w \leftarrow \|r\|_2^2 / p^T v$
        $e \leftarrow e + wp$
        $\tilde{r} \leftarrow r$
        $r \leftarrow r - wv$
        $p \leftarrow r + (\|r\|_2^2 / \|\tilde{r}\|_2^2) p$
    end
end
return $e$
```

V-Cycle and MGCG Method

Finally, we have all constitutive parts at hand for a memory efficient implementation of a matrix-free v-cycle for the large-scale linear systems (5.52). This procedure is presented in Algorithm 5.13. The `v_cycle_mf` function can now be applied as MG method (cf. Algorithm 2.3) to solve the large-scale linear system (5.52) or as preconditioning method for the CG iteration, resulting MGCG method (cf. Algorithm 2.6). The memory efficient implementation of the matrix-free CG method (cf. Algorithm 5.8 and 5.9) and the memory efficient implementation of the matrix-free `v_cycle_mf` function (cf. Algorithm 5.13) finally leads to a memory efficient implementation of a matrix-free MGCG method to solve the large-scale linear system (5.52), given in Algorithm 5.14.

Algorithm 5.13: `v_cycle_mf`: V-cycle for the linear system (5.52)

```
v_cycle_mf$(x, b, g, \nu)$
    if $g = 1$ then
        $x \leftarrow$ CG_coarse$(A_1, b)$                      // apply Algorithm 5.12
    end
    else
        $x \leftarrow$ JAC$(x, b, g, \nu_1)$                    // apply Algorithm 5.11
        $r \leftarrow A_g x - b$                                 // apply Algorithm 5.2, 5.1, and 5.5
        $r \leftarrow I_g^{g-1} r$                               // apply Algorithm 5.5
        $e \leftarrow$ v_cycle_mf$(0, r, g-1, \nu)$             // apply Algorithm 5.13
        $x \leftarrow I_{g-1}^{g} e$                             // apply Algorithm 5.5
        $x \leftarrow$ JAC$(x, b, g, \nu_2)$                    // apply Algorithm 5.11
    end
end
return $x$
```

Algorithm 5.14: Multigrid preconditioned conjugate gradient method for the linear system (5.52)

```
$x \leftarrow 0$
$r \leftarrow b$                                                 // apply Algorithm 5.2, 5.1, and 5.5
$p \leftarrow z \leftarrow$ v_cycle_mf$(0, r, G, \nu)$          // apply Algorithm 5.13
while $\|r\|_2^2 > tol$ do
    $v \leftarrow A_G p$                                         // apply Algorithm 5.2, 5.1, and 5.5
    $w \leftarrow \|r\|_2^2 / p^T v$
    $x \leftarrow x + wp$
    $\tilde{r} \leftarrow r$
    $r \leftarrow r - wv$
    $\tilde{z} \leftarrow z$
    $z \leftarrow$ v_cycle_mf$(0, r, G, \nu)$                   // apply Algorithm 5.13
    $p \leftarrow z + (r^T z / \tilde{r}^T \tilde{z}) p$
end
return $x$
```

The main reason for introducing a preconditioner is to improve the computational complexity and therefore the runtime of the applied CG algorithm. The performance of the proposed MGCG methods for the large-scale P-spline related problems also in comparison to the unpreconditioned CG method is further addressed in the following section.

5.5 Complexity of the Multigrid Preconditioned Conjugate Gradient Method

In this section, we analyze the performance of the matrix-free multigrid preconditioned conjugate gradient method as implemented in Algorithm 5.14 in terms of computational complexity and runtime. For reasons of simplification, we restrict ourselves to the determination of an (unconstrained) regression P-spline, that is we focus on the solution of the large-scale linear system

$$\left(\Phi^T\Phi + \lambda\Lambda\right)\alpha \overset{!}{=} \Phi^T y \tag{5.64}$$

by Algorithm 5.14. We also compare the algorithmic complexity of the MGCG method to the algorithmic complexity of the unpreconditioned CG method as implemented in Algorithm 5.8. The following computations are performed on a computer system equipped with Intel Core i7-6700 3.4 GHz Quad-Core Processor and 32 GB of RAM. The underlying codes are programmed within the statistical computing software R (version 3.4.4) and the algorithmic building blocks, which are critical to computational performance, are accelerated by using the RCPP extension library (cf. Eddelbuettel and François, 2011 and Eddelbuettel, 2013) and programmed in the C++ programming language.

5.5.1 Computational Setup

In order to analyze the performance of the MGCG method, we consider the finite and discrete data sets

$$\{(x_i, y_i) \in \mathbb{R}^P \times \mathbb{R} \mid i = 1, \dots, 100{,}000\},\ P = 2, 3, 4. \tag{5.65}$$

The covariates x_i are uniformly distributed on the rectangle $\Omega := [0,1]^P$ and the variable of interest is generated from the model

$$y_i := f_P(x_i) + \varepsilon_i,\ \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1^2), \tag{5.66}$$

where

$$f_P \colon \Omega \to [0,1],\ x \mapsto \frac{1}{1 + \exp\left(-16\left(\|x\|_2^2 P^{-1} - 0.5\right)\right)} \tag{5.67}$$

denotes a normalized sigmoid function in P dimensions. For one and two dimensions the related sigmoid functions f_P are graphed in Figure 5.1. We consider these functions, since they are irrational functions that show a similar behavior for varying covariate dimensions P. However, since the performance of the solution algorithms is in focus and not the quality of the resulting P-spline, the exact form of the generating function is of minor importance. As proposed in Section 5.4, we apply the penalized spline method with uniform cubic B-spline basis and curvature penalty to obtain a regression P-spline. For maximum grid levels $G \in \{4, 5, 6, 7\}$, we utilize $m_p = 2^G - 1$ knots for each covariate dimension $p = 1, \ldots, P$. This results in the linear system

$$\left[(\Phi_G)^T \Phi_G + \lambda \Lambda_G \right] \alpha \overset{!}{=} (\Phi_G)^T y. \tag{5.68}$$

To solve this large-scale linear system, we apply the matrix-free CG method (cf. Algorithm 5.8) as well as the matrix-free MGCG method (cf. Algorithm 5.14). For the MGCG method, we apply $\nu = (6, 3)$ Jacobi smoothing iterations with a damping factor of $\omega_{\text{JAC}} := 1/3$. We further apply the MGCG algorithm with the SSOR method as smoothing iteration (cf. Chapter 2.4.1) with $\nu = (6, 3)$ and $\omega_{\text{SSOR}} := 1$ for comparative reasons. Note that for this approach the coefficient matrix is completely assembled and stored in the CSC format. In order to distinguish between both of the MGCG methods, we refer to the MGCG_JAC method for the matrix-free implementation with Jacobi smoother and to the MGCG_SSOR method for the sparse implementation with SSOR smoother.

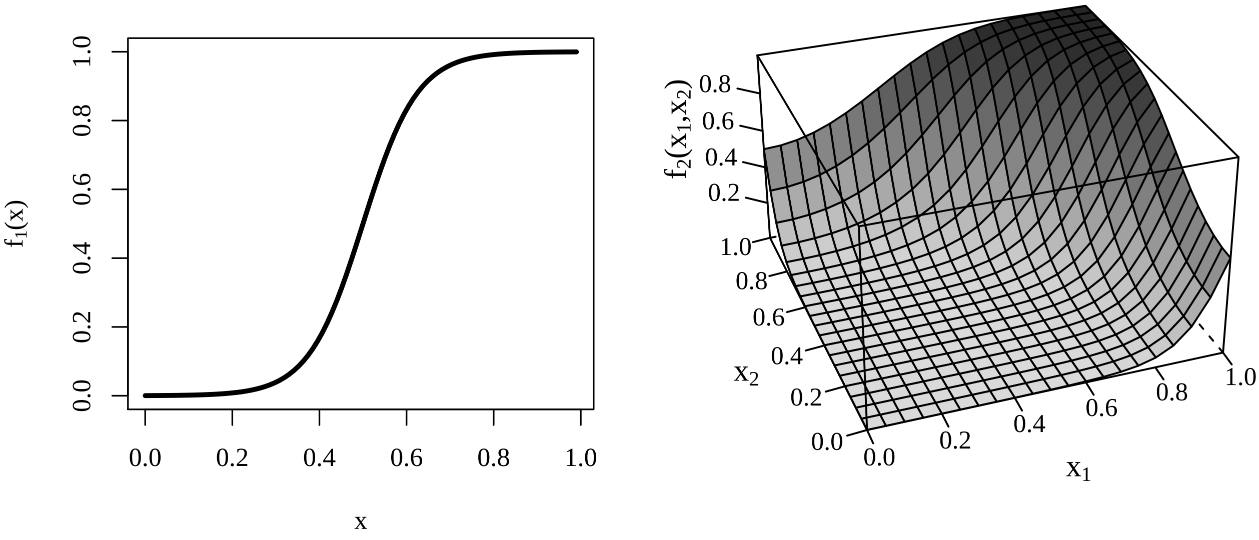

Figure 5.1: Sigmoid functions for $P = 1$ and $P = 2$ dimensions.

5.5.2 Algorithmic Results

This subsection presents the results of the CG, the MGCG_JAC, and the MGCG_SSOR method applied to solve the linear system (5.68) for various covariate dimensions P and various maximum grid levels G. At this, we analyze the convergence behavior, special features of the MGCG method, and the approximation quality of the resulting P-spline.

Convergence and Algorithmic Complexity

The main interest is in the convergence behavior of Algorithm 5.14 in terms of the number of required iterations and runtime to reach an adequate stopping criterion as well as in its required memory. As stopping criterion, we chose a relative error reduction by a factor of five, that is

$$\frac{\| \left[(\Phi_G)^T\Phi_G + \lambda\Lambda_G\right] \alpha - (\Phi_G)^T y\|_2^2}{\|(\Phi_G)^T y\|_2^2} \leq 10^{-5}. \tag{5.69}$$

This stopping criterion is sufficient for the most applications and results in an adequate P-spline fit (cf. Figure 5.4). In a first test case, we fix the maximum grid level to $G = 5$, as suggested in Section 5.4. The number of required iterations and related runtimes for various covariate dimension P is presented in Table 5.2. The number of iterations deteriorates for increasing P for all of the three methods, which is due to the fact that the condition number of the coefficient matrix increases with increasing P. The increasing number of required iterations and the increasing size of the coefficient matrix explains the deteriorating runtime of the methods for increasing P. The CG method and the MGCG_JAC method perform quite similar in small dimension $P = 1$ and $P = 2$ in terms of runtime. The CG method requires significantly more iterations than the both MGCG method, but, since a CG iteration is much cheaper, this is not reflected in the overall runtime. In this small dimensions the MGCG_SSOR method clearly outperforms the other methods in number of iterations and in computational time. However, it has to noted that the comparison of computational time is not legitimated in this case since only the computational time of the solution iteration is considered, but not the offline costs. Further, the MGCG_SSOR method does not rely on the matrix-free approach. Since complete matrices have to be assembled for the MGCG_SSOR method, its offline costs are much higher compared to the other methods. In $P = 3$ dimensions, however, the MGCG with SSOR smoothing for the test problem requires approximately 30 GB of RAM, which is at the limit of the utilized computer system. By contrast, the matrix-free methods require approximately 16 MB (CG) and 78 MB (MGCG_JAC) of RAM. This is reflected in the overall runtime since the MGCG_SSOR method now requires much more computational time compared to the other methods, despite the fact that it remains still superior in terms of required iterations. For $P = 4$, the memory requirements of the MGCG_SSOR method exceed the internal memory such that it does not produce any results. For this dimension the matrix-free CG and MGCG_JAC method are still applicable, since they are not restricted by memory constraints. The advantage of the MGCG_JAC method compared to the CG method here becomes more obvious. The required number of CG iterations growth much more rapidly then the number of MGCG_JAC iterations, which is also reflected in the overall running time.

	CG		MGCG_JAC		MGCG_SSOR	
P=1	22	(1.41)	2	(1.13)	1	(<0.10)
P=2	73	(4.34)	4	(3.72)	2	(<0.10)
P=3	367	(107.41)	14	(82.02)	3	(454.80)
P=4	747	(2956.98)	19	(2001.58)	–	

Table 5.2: Required number of iterations and algorithmic runtime of the methods CG, MGCG_JAC, and MGCG_SSOR for $G = 5$ grids and varying dimensions P.

Condition Number and Eigenvalues

Crucial for the computational complexity of the CG method in the unpreconditioned and in the preconditioned case is the condition number of the respective coefficient matrix. That is

$$(\Phi_G)^T\Phi_G + \lambda\Lambda_G \tag{5.70}$$

for the unpreconditioned CG method and

$$C_{\mathrm{MG},G}\left[(\Phi_G)^T\Phi_G + \lambda\Lambda_G\right] \tag{5.71}$$

for the MGCG method, where the iteration matrix of the multigrid method $C_{\mathrm{MG},G}$ is given as in (2.62). Note that for the MGCG_JAC the Jacobi smoother is utilized and for the MGCG_SSOR the SSOR smoother. Therefore, different preconditioning matrices $C_{\mathrm{MG},G,\mathrm{JAC}}$ and $C_{\mathrm{MG},G,\mathrm{SSOR}}$ occur. Table 5.3 presents the condition number of the respective coefficient matrices of the applied methods for $G = 5$ grids and $P = 2$ dimensions. The condition numbers confirm the results of the analysis of the computational complexity. The condition for the (unpreconditioned) CG method is comparatively huge. Amongst the multigrid preconditioned methods the SSOR smoother performs superior to the Jacobi smoother, which nevertheless provides a significant improvement compared to the unpreconditioned case.

CG	MGCG_JAC	MGCG_SSOR
1933.27	1.82	1.03

Table 5.3: Condition number of the coefficient matrices of the methods CG, MGCG_JAC, and MGCG_SSOR for $G = 5$ grids and $P = 2$ dimensions.

To further explain the different performance of the MGCG methods, Figure 5.2 displays the eigenvalues of the related coefficient matrices on a logarithmic scale. For the unpreconditioned coefficient matrix of the CG method very large eigenvalues occur, which is responsible for its comparatively large condition number. Since the eigenvalues are slightly clustered, the computational complexity is still acceptable. Both MGCG methods can handle the large eigenvalues very well such that for both coefficient matrices the maximum eigenvalue is approximately one. The MGCG with Jacobi smoother is incapable to handle the small eigenvalues such that the smallest eigenvalue of the unpreconditioned case remains nearly unchanged. By contrast, the MGCG_SSOR method is also able to handle the small eigenvalues which explains its superior performance in terms of required iterations.

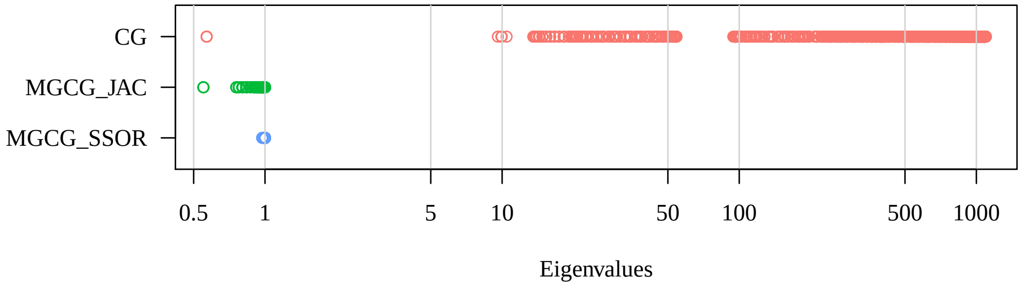

Figure 5.2: Eigenvalues of the coefficient matrices of the methods CG, MGCG_JAC, and MGCG_SSOR for $G = 5$ grids and $P = 2$ dimensions.

Grid Independence

To analyze the performance of the proposed algorithms, we fixed a maximum grid level $G = 5$ and varied the covariate dimension P in a first test case. For the following test case, we fix the spatial dimension to be $P = 2$ and apply the solution methods for various maximum grid levels $G \in \{4, 5, 6, 7\}$. The results in terms of the required number of iterations are presented in Figure 5.3 on a logarithmic scale. We observe that the number of (unpreconditioned) CG iterations significantly increases under grid refinements. For both, the MGCG_JAC and the MGCG_SSOR method, the number of iterations is almost constant, i.e. 1-2 for MGCG_SSOR and 4 for MGCG_JAC. This result illustrates that the multigrid preconditioner provides a scalable solver for the linear system (5.68). Clearly, the computational times are increasing since we run only a single core code. A standard approach is to distribute the matrix between an increasing number of processors in a cluster computer in order to obtain almost constant runtimes in the sense of weak scalability. Note that this is not to achievable for the plain CG solver due to the increasing computational complexity.

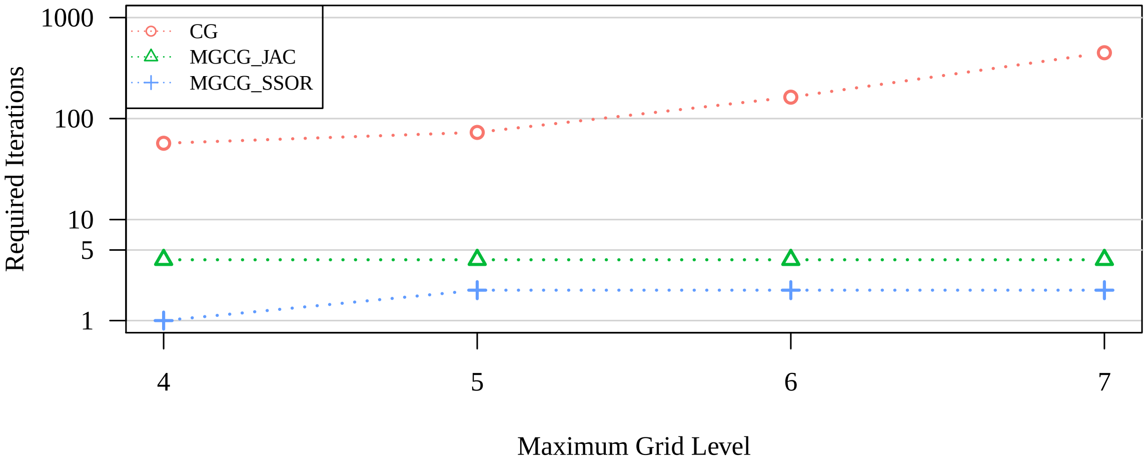

Figure 5.3: Grid independent convergence of the MGCG_JAC and the MGCG_SSOR method in comparison to the CG method for $P = 2$ dimensions.

Regression P-Spline

Finally, Figure 5.4 shows the resulting regression P-spline in $P = 2$ dimensions with a maximum grid level of $G = 5$ (left) and the corresponding residuals to the 100,000 noisy data points (right). This shows that the resulting regression P-spline recovers the underlying sigmoid test function with adequate precision, since it can not be distinguished by eye from the data generating function in Figure 5.1. Also the presented residuals do not show any irregularities and behave like drawn from the $\mathcal{N}(0, 0.1^2)$ distribution. This is, however, a feature of the penalized spline method itself and is not due to the utilized algorithm. Nevertheless, it shows that the introduced methods are rational and sensibly implemented and lead to adequate results not only in terms of computational complexity but also from a application point of view.

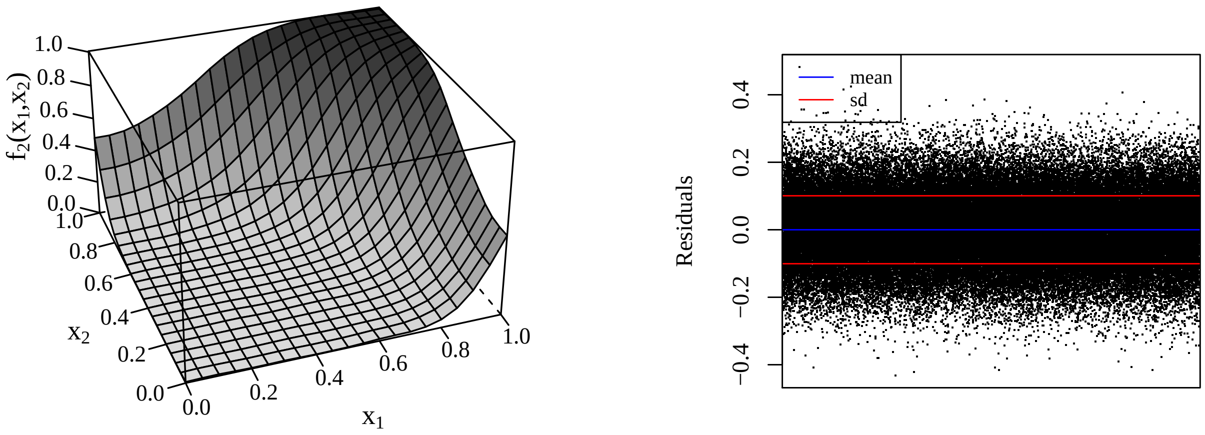

Figure 5.4: Regression P-spline fit and related residuals for $P = 2$ dimensions.

Chapter 6

Conclusion and Outlook

In this closing chapter, we conclude the present thesis and point out its main findings. Further, we present a short outlook on future research topics and possible applications.

Conclusion

Motivated from a practical application in Section 1.1, the present thesis addresses the following research topics:

1. How can nonlinear and highly flexible modeling techniques be utilized for model-based small area estimation methods? How does this modeling affect the quality of the resulting small area estimates?
2. How can (global) shape constraints on the relationship between the variables be incorporated into the modeling process without restricting the flexibility?

For that purpose, Section 3.2 introduces the penalized spline method as a highly flexible modeling technique. The utilization of P-splines in small area estimation dates back to Opsomer et al. (2008) and Ugarte et al. (2009) and is based on linear mixed model techniques. In this thesis a different framework is developed, where the model parameters are determined via the solution of an unconstrained optimization problem (cf. Section 4.2) instead as from a linear mixed model. Due to this optimization approach, the incorporation of shape constraints into the modeling process is achieved in terms of additional linear inequality constraints on the optimization problem (cf. Section 3.3 and Section 4.3). This new framework results in the innovative small area estimators (4.42) and (4.52) that allow for both the utilization of the penalized spline method as a highly flexible modeling technique and the incorporation of arbitrary shape constraints on the underlying P-spline function. This has not been considered in literature so far such that the present thesis provides a significant contribution to the field of small area estimation. Small area estimation not only addresses the issue of providing reliable estimates of area-specific characteristics of interest for subpopulations with very few sample data, but also focuses on the assessment of the precision of theses small area estimates. For that purpose, we introduce a bootstrap MSE-estimator in Section 4.5 for the implemented point estimators that enables the application of the proposed methods in practice. Within the scope of a simulation study in Section 4.6 and by means of the real-world application presented in Section 4.7, the huge potential of this innovative methods is exposed. The present thesis therefore not only provides theoretical results in the field of small area estimation, but also yields methods of practical relevance.

Besides the possibility of utilizing a highly flexible and close to reality modeling technique, it is desired to extend the developed methods to multiple covariates. Therefore, a further research topic discussed by the present thesis is:

3. How can multiple covariates be incorporated into the various P-spline models while preserving the underlying structure?

For that purpose, we employ a tensor product approach to extend the penalized spline method to multiple input variables as introduced in Section 3.2. This extension affects neither the general structure of the spline models nor the structure of the resulting optimization problem. Therefore, the extension to shape constraints and the utilization in the context of small area estimation is achieved in analogy to the case of a single covariate.

A serious limitation of the tensor product approach is caused by an exponential growth of the size of the underlying optimization problems with each additional covariate. This leads to an unjustifiable complexity of the applied solution algorithms in terms of runtime and in terms of memory requirements. This rapid growth causes a tremendous memory demand such that the internal memory of common computer systems does not support a naive extension of the spline-based methods to more than two or three covariates. Conclusively, the present thesis is also devoted to the following research question:

4. How can the occurring (constrained and unconstrained) large-scale optimization problems be solved in a computationally and memory efficient manner? How are the related large-scale algorithms to implement?

By exploiting the underlying tensor nature of the spline functions, we develop and implement various operations for particular classes of matrices that come along without assembling and storing these matrices (cf. Section 5.1). This allows for a memory efficient implementation of adequate solution algorithms for the considered optimization problems in Section 5.2 and Section 5.3. A crucial part within the proposed large-scale algorithms is the (repetitive) application of a matrix-free CG method. In order to improve the computational complexity of these algorithms, we finally implement a matrix-free MGCG method in Section 5.4. The algorithmic complexity of the proposed MGCG method is analyzed in numerical test cases (cf. Section 5.5) that show the advantageous characteristics of this implementation. By the development of computationally efficient solution algorithms for special types of optimization problems and their memory efficient implementation, the present thesis also provides a significant contribution to the field of algorithmic optimization.

Especially by the interplay of the application driven need for alternative and extended small area models and the resulting demand for highly developed large-scale solution algorithms, this thesis provides an illustration on the development of cross-disciplinary optimization methods and algorithms for challenging problems motivated by real world applications. In this way, the natural interdependency between mathematics and statistics is deepened such that the thesis additionally provides a contribution to the connection of both disciplines.

Outlook

Regarding the issue of algorithmic runtime, the multigrid preconditioned conjugate gradient method provides an adequate solution approach. A further speed up can be obtained by utilizing a parallel implementation of the algorithm. A trivial parallelization of the matrix operations, e.g. the parallel computation of all required matrix-vector products, is not productive since the per core operations are very fast but a lot of direct memory access is required. In fact, it turns out that the trivial parallelization even decelerates the algorithmic runtime. In order to apply parallel computing methods, a more sophisticated implementation of the proposed algorithms is required. In this context, the programming with graphics processing units (GPU) instead of central processing units (CPU) can provide a significant improvement of the algorithmic runtime of the proposed methods.

From a practical point of view, a possible application is already mentioned in Section 1.1. The sentinel-2 satellites provide multi-spectral data with 13 bands that can be used as auxiliary information. Certainly, the incorporation of 13 covariates into the penalized spline method is not reasonable, neither from a numerical nor from a practical point of view. The question arises how many covariates are useful and can be incorporated into the presented methods. In this context, efficient variable selection procedures that are adequate for the applied spline methods need to be developed. Further, the proposed estimators are obviously not restricted to forest applications. In official statistics for example a frequent demand is coercivity of the small area estimates, which means that the estimates need to add up to a given value on higher aggregation levels. In order to achieve coercive small area estimates from the developed methods, the coercivity restriction has to be transformed into constraints on the considered optimization problem. A further topic that arises from the simulation study is the fact that the sampling design influences the quality of the estimators. The impact of the sampling design on small area estimates is an ongoing research topic (cf. Münnich and Burgard, 2012 and Burgard et al., 2014) and needs to be further examined for the particular estimators proposed in this thesis.

Bibliography

Bates, D. and Maechler, M. (2018). *Matrix: Sparse and Dense Matrix Classes and Methods.*

Battese, G. E., Harter, R., and Fuller, W. A. (1988). An error-components model for prediction of county crop areas using survey and satellite data. *Journal of the American Statistical Association*, 83(401):28–36.

Bellman, R. (1957). *Dynamic Programming.* Princeton University Press.

Benoit, A., Plateau, B., and Stewart, W. J. (2001). Memory efficient iterative methods for stochastic automata networks. Technical Report 4259, INRIA.

Bertsekas, D. P., Nedić, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization.* Athena Scientific.

Björck, A. (1996). *Numerical Methods for Least Squares Problems.* SIAM.

Bolfarine, H. and Zacks, S. (1992). *Prediction Theory for Finite Populations.* Springer Series in Statistics.

Bollaerts, K., Eilers, P. H., and van Mechelen, I. (2006). Simple and multiple P-splines regression with shape constraints. *British Journal of Mathematical and Statistical Psychology*, 59:451–469.

Breidenbach, J. and Astrup, R. (2012). Small area estimation of forest attributes in the Norwegian National Forest Inventory. *European Journal of Forest Research*, 131:1255–1267.

Breidt, F., Claeskens, G., and Opsomer, J. (2005). Model-assisted estimation for complex surveys using penalised splines. *Biometrika*, 92:831–846.

Burgard, J. P., Münnich, R., and Zimmermann, T. (2014). The impact of sampling designs on small area estimates for business data. *Journal of Official Statistics*, 30(4):749–771.

Cassel, C. M., Särndal, C. E., and Wretman, J. H. (1976). Some results on generalized difference estimation and generalized regression estimation for finite populations. *Biometrika*, 63(3):615–620.

Chambers, R. and Chandra, H. (2013). A random effect block bootstrap for clustered data. *Journal of Computational and Graphical Statistics*, 22:452–470.

Cochran, W. G. (2007). *Sampling Techniques.* Wiley, New York.

Cox, M. G. (1972). The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics*, 10:134–149.

Currie, I. D. and Durban, M. (2002). Flexible smoothing with P-splines: A unified approach. *Statistical Modelling*, 2:333–349.

Datta, G. S. (2009). Model-based approach to small area estimation. In *Handbook of Statistics (pp. 251-288)*. Elsevier.

de Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. Oxford University Press Inc.

Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. Springer, New York.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

Eilers, P. H. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11:89–121.

Eilers, P. H. and Marx, B. D. (2010). Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(6):637–653.

Eubank, R. L. (1988). *Spline Smoothing and Nonparametric Regression*. Marcel Dekker Inc.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. D. (2013). *Regression: Models, Methods and Applications*. Springer-Verlag, Berlin Heidelberg.

Fay, R. E. and Herriot, R. A. (1979). Estimates of income for small places: An application of James-Stein procedures to census data. *Journal of the American Statistical Association*, 74(366):269–277.

Flores, L. and Martinez, L. (2000). Land cover estimation in small areas using ground survey and remote sensing. *Remote Sensing of Environment*, 74:240–248.

Foley, J. A., DeFries, R., Asner, G. P., Barford, C., Bonan, G., Carpenter, S. R., Chapin, F. S., Coe, M. T., Daily, G. C., and Gibbs, H. K. (2005). Global consequences of land use. *Science*, 309:570–574.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67.

Gallego, F. (2004). Remote sensing and land cover area estimation. *International Journal of Remote Sensing*, 25:3019–3047.

Graham, A. (1981). *Kronecker Products and Matrix Calculus: With Applications*. Horwood.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach*. Chapman and Hall.

Hackbusch, W. (1985). *Multi-Grid Methods and Applications*. Springer, Heidelberg.

Hackbusch, W. (1994). *Iterative Solution of Large Sparse Systems of Equations*. Springer.

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34(4):561–580.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72:322–340.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning*. Springer.

Henderson, C. R. (1950). Estimation of genetic parameters. *Annals of Mathematical Statistics*, 21:309–310.

Henderson, C. R. (1975). Best linear unbiased estimation and prediction under a selection model. *Biometrics*, 31(2):423–447.

Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):406 – 436.

Höllig, K. (2003). *Finite Element Methods with B-Splines*. SIAM.

Horvitz, D. and Thompson, D. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47:663–685.

Jiang, J. and Lahiri, P. (2006). Mixed model prediction and small area estimation. *TEST*, 15(1):1–96.

Kackar, R. N. and Harville, D. A. (1981). Unbiasedness of two-stage estimation and prediction procedures for mixed linear models. *Communications in Statistics - Theory and Methods*, 10(13):1249–1261.

Kackar, R. N. and Harville, D. A. (1984). Approximations for standard errors of estimators of fixed and random effect in mixed linear models. *Journal of the American Statistical Association*, 79(388):853–862.

Kangas, A. and Maltamo, M. (2006). *Forest inventory: methodology and applications*, volume 10. Springer Science & Business Media.

Kauermann, G. (2005). A note on smoothing parameter selection for penalized spline smoothing. *Journal of Statistical Planning and Inference*, 127:53–69.

Kauermann, G., Claeskens, G., and Opsomer, J. D. (2009). Bootstrapping for penalized spline regression. *Journal of Computational and Graphical Statistics*, 18:126–146.

Lehtonen, R. and Veijanen, A. (2009). Design-based methods of estimation for domains and small areas. In *Handbook of statistics (pp. 219-249)*. Elsevier.

Liu, S. and Trenkler, G. (2008). Hadamard, khatri-rao, kronecker and other matrix products. *International Journal of Information and Systems Sciences*, 4(1):160–177.

Lohr, S. (1999). *Sampling: Design and Analysis*. Cengage Learning.

Mandelkow, A. (2012). Holzvorratsschätzung mit Small Area Estimation. Master's thesis, Universität Trier.

Marchetti, S., Giusti, C., Pratesi, M., Salvati, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., Pappalardo, L., and Gabrielli, L. (2015). Small area model-based estimators using big data sources. *Journal of Official Statistics*, 31(2):263–281.

McRoberts, R., McWilliams, W., Reams, G., Schmidt, T., Jenkins, J., O'Neilla, K., Miles, P., and Brand, G. (2004). Assessing sustainability using data from the forest inventory and analysis program of the united states forest service. *Journal of Sustainable Forestry*, 18:23–46.

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *Annals of Applied Statistics*, 3(2):1013–1033.

Meyer, M. C. (2012). Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206.

Münnich, R. and Burgard, J. (2012). On the influence of sampling design on small area estimates. *Journal of the Indian Society of Agricultural Statistics*, 66(1):145–156.

Münnich, R., Burgard, J. P., and Vogt, M. (2013). Small Area-Statistik: Methoden und Anwendungen. *AStA Wirtschafts- und Sozialstatistisches Archiv*, 6:149–191.

Münnich, R., Gabler, S., Ganninger, M., Burgard, J. P., and Kolb, J. P. (2012). Stichprobenoptimierung und Schätzung im Zensus 2011. Statistik und Wissenschaft. *Statistisches Bundesamt, Wiesbaden.*

Münnich, R., Wagner, J., Hill, J., Stoffels, J., Buddenbaum, H., and Udelhoven, T. (2016). Schätzung von Holzvorräten unter Verwendung von Fernerkundungsdaten. *AStA Wirtschafts- und Sozialstatistisches Archiv*, 10:95–112.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization.* Springer, New York.

Opsomer, J. D., Breidt, F. J., Moisen, G. G., and Kauermann, G. (2007). Model-assisted estimation of forest resources with generalized additive models (with discussion). *Journal of the American Statistical Association*, 102(478):400 – 416.

Opsomer, J. D., Claeskens, G., Ranalli, M. G., Kauermann, G., and Breidt, F. J. (2008). Non-parametric small area estimation using penalized spline regression. *Journal of the Royal Statistical Society: Series B*, 70:265–286.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4):502–527.

Polley, H., Hennig, P., Schmitz, F., Dunger, K., and Schwitzgebel, F. (2006). The second national forest inventory: Results ; covering the national forest inventory surveys of 2001-2002 and 1986-1988. *Federal Ministry of Food, Agriculture and Consumer Protection.*

Purcell, N. P. and Kish, L. (1979). Estimation for small domains. *Biometrics*, 35(2):365–384.

R Core Team (2018). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Rao, J. N. K. (2003). *Small Area Estimation.* Wiley, New York.

Rao, J. N. K. and Molina, I. (2015). *Small Area Estimation.* Wiley, New York.

Rueda, C. and Lombardía, M. (2012). Small area semiparametric additive monotone models. *Statistical Modelling*, 12:527–549.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression.* Cambridge University Press.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems.* SIAM.

Särndal, C. E., Swensson, B., and Wretman, J. H. (1992). *Model Assisted Survey Sampling.* Springer, New York.

Schumaker, L. L. (1981). *Spline Functions: Basic Theory.* Wiley.

Searle, S. R., Casella, G., and McCulloch, C. E. (2009). *Variance Components.* John Wiley and Sons.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B*, 47(1):1–52.

St-Onge, B., Treitz, P., and Wulder, M. A. (2003). Tree and canopy height estimation with scanning LIDAR. In *Remote Sensing of Forest Environments (pp. 489-509).* Springer.

Ugarte, M. D., Goicoa, T., Militino, A. F., and Durban, M. (2009). Spline smoothing in small area trend estimation and forecasting. *Computational Statistics and Data Analysis*, 53:3616–3629.

UNFCCC (1998). The kyoto protocol to the convention on climate change. *United Nations Framework Convention on Climate Change.*

Wagner, J., Münnich, R., Hill, J., Stoffels, J., and Udelhoven, T. (2017). Non-parametric small area models using shape-constrained penalized B-splines. *Journal of the Royal Statistical Society: Series A*, 180(4):1089–1109.

Wahba, G. (1990). *Spline Models for Observational Data.* SIAM.

Wu, C.-F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295.