universitätbonn igg

Institut für
Geodäsie und Geoinformation

Schriftenreihe 7

Hamza Alkhatib

On Monte Carlo Methods with Applications to the Current Satellite Gravity Missions

universitätbonn igg

Institut für
Geodäsie und Geoinformation

Schriftenreihe **7**

Hamza Alkhatib

On Monte Carlo Methods with Applications to the Current Satellite Gravity Missions

Diese Arbeit wurde am 29. März 2007 als Dissertation zur Erlangung
des Grades Doktor-Ingenieur (Dr.-Ing.) der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn vorgelegt.

Referent:	Prof. Dr. Wolf-Dieter Schuh
Korreferent:	PD Dr.-Ing. Jürgen Kusche
Tag mündlichen Prüfung:	1. Juni 2007

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
http://hss.ulb.uni-bonn.de/diss_online
elektronisch und durchgängig mit farbigen Abbildungen publiziert.

Schriftenreihe des Instituts für Geodäsie und Geoinformation
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Herausgeber:
Prof. Dr.-Ing. Wolfgang Förstner
Prof. Dr.-Ing. Karl Heinz Ilk
Prof. Dr.-Ing. Theo Kötter
Prof. Dr.-Ing. Heiner Kuhlmann
Prof. Dr. Lutz Plümer
Prof. Dr. Wolf-Dieter Schuh

Die Aufnahme dieser Arbeit in die Schriftenreihe wurde von den
Herausgebern der Reihe geprüft und gemeinsam beschlossen.

Bonn 2008 ISSN 1864-1113

Vorwort

Moderne Messsensoren erlauben eine automatisierte Beobachtung von physikalischen Prozessen mit immer höherer Auflösung. Dies betrifft sowohl die zeitliche als auch die räumliche Komponente. Bei der Anpassung der Messdaten an mathematische Modelle sind somit Hunderttausende Parameter keine Seltenheit. Die Berechnung der Parameter anhand der Messwerte in diskreten Punkten bildet ein inverses Problem (Datenanpassungsproblem) und erfordert die Lösung zumeist großer Gleichungssysteme. Modellbildungen und die Wahl von Basisfunktionen sind dabei sehr anwendungsspezifisch. Je nach Anwendungstyp treten sowohl dünn als auch voll besetzte Systeme auf. Aber auch die Einbindung von vorprozessierten Daten und a priori Information in Form von Normalgleichungsteilen wird vielfach gefordert. Für die Parameterschätzung aus diesen heterogenen Datensätzen werden zumeist maßgeschneiderte Algorithmen eingesetzt. Speziell für sehr große Systeme kommen oftmals iterative Löser auf massiv parallelen Systemen zum Einsatz.

Für geodätische Anwendungen weisen diese Löser zwei gravierende Nachteile auf. Erstens ist die Berechnung der Varianz/Kovarianzinformation nur sehr aufwendig zu bewerkstelligen und zweitens ist es schwer, die Redundanzanteile einzelner Beobachtungsgruppen zu ermitteln. Dies liegt darin begründet, dass man die dafür notwendigen Normalgleichungen und Inversen auf Grund der Größe nicht explizit berechnet, sondern im iterativen Löser eine implizite Darstellung verwendet.

Herr Dr.-Ing. Hamza Alkhatib beschäftigt sich in dieser Arbeit mit zwei Verfahren, welche diese Mängel von iterativen Lösern beseitigen. Somit eröffnen sich neue Möglichkeiten der Anwendung. Basierend auf Monte-Carlo Integration wird einerseits ein alternatives Verfahren zur Schätzung der Varianz/Kovarianzinformation aufgezeigt und andererseits wird eine stochastische Schätzung der Varianzkomponenten für die optimale Gewichtung von heterogenen Datensätzen erarbeitet. Einzige Voraussetzung für diese Lösungsansätze bildet ein effizienter Löser des Gesamtproblems, welcher mehrere Lösungen in einem Schritt ermitteln kann. Die hier dargestellten Verfahren können also auf beliebige Datenkombinationen adaptiert werden und ermöglichen so auch bei sehr großen Systemen die effiziente Schätzung der Varianz/Kovarianzinformation und der Varianzkomponenten.

Bonn, im Juli 2008

On Monte Carlo methods with applications to the current satellite gravity missions

Abstract

Large-scaled least squares problems require tailored numerical techniques to overcome the computational burden. For these types of problems iterative strategies are suitable because of their flexibility and effectiveness. The first shortcoming of iterative strategies in least squares estimation is the fact that the inverse of the normal equation matrix as the carrier of the covariance information is not available or very expensive to compute. Another shortcoming within iterative strategies arises when different types of observation groups with different stochastic properties are to be combined. In this case the choice of optimum weight factors, and eventually regularization parameters, by means of variance component estimation is essential for obtaining reliable estimates of the unknown parameters. Unfortunately, the conventional method of variance component estimation requires the repeated inversion of large products of matrices. This thesis presents algorithms based on Monte Carlo methods, which can be integrated very efficiently into iterative solvers, and which are demonstrated to close the aforementioned gaps. Tailored strategies for different types of solution techniques with respect to normal equations, observation equations and combined models are treated. In addition, the thesis presents new criteria to define confidence regions of the estimated variance information of the parameters, as well as for all additional derived quantities. The developed algorithms for computing variance/covariance matrices and for obtaining variance components are tailored to be integrated into the Preconditioned Conjugate Gradients Multiple Adjustment (PCGMA) algorithm of Schuh 1996 and Boxhammer 2006. These algorithms are applied in a case study to simulated GOCE data, where Satellite Gravity Gradiometry (SGG) data in form of observation equations and Satellite-to-Satellite tracking (SST) data in form of normal equations are combined for recovering the Earth's gravity field.

Zusammenfassung

In großdimensionierten Ausgleichungsproblemen lassen sich die numerischen Kosten oft nur mit speziell angepassten Techniken in erträglichen Grenzen halten. Iterative Strategien sind hier aufgrund ihrer Flexibilität und Effizienz besonders geeignet. Allerdings wird dabei der Gewinn an Rechenzeit mit einem Verlust an Information erkauft, da iterative Löser zumeist nicht effizient sind, um die Varianz-Kovarianz-Matrix der Unbekannten zu liefern. Ein weiterer Nachteil zeigt sich, wenn Beobachtungsgruppen mit unterschiedlichen stochastischen Eigenschaften miteinander kombiniert werden sollen. In diesem Fall ist es notwendig, die optimalen Gewichtungsfaktoren - eventuell auch Regularisierungsparameter - über die Schätzung von Varianzkomponenten zu bestimmen. Die herkömmliche Methode der Varianzkomponentenschätzung erfordert dabei die wiederholte Inversion großer Matrizen. In dieser Arbeit wird gezeigt, daß sich diese Probleme mit auf Monte-Carlo-Methoden basierenden Algorithmen lösen lassen, die sehr effizient in iterative Löser integriert werden können. Speziell angepasste Algorithmen für Löser, die verschiedene Beobachtungsgruppen sowohl auf Basis von Normalgleichungen als auch auf Basis von Beobachtungsgleichungen oder auch Kombinationen von Beobachtungs- und Normalgleichungen verarbeiten können, werden vorgestellt. Darüber hinaus werden in dieser Arbeit neue Kriterien angegeben, mit denen Konfidenzintervalle für die Varianzinformation der geschätzten Parameter und auch aller abgeleiteten Größen angegeben werden können. Die entwickelten Algorithmen zur Berechnung der Varianz-Kovarianz-Matrizen und zur Schätzung der Varianzkomponenten werden in den PCGMA-Algorithmus von Schuh 1996 und Boxhammer 2006 integriert. Sie werden auf ein Testszenario mit simulierten GOCE-Daten angewendet, in dem zur Bestimmung des Erdschwerefeldes "Satellite Gravity Gradiometry"-Daten auf Basis von Beobachtungsgleichungen mit "Satellite-to-Satellite-Tracking"-Daten auf Basis von Normalgleichungen miteinander kombiniert werden.

Contents

List of Abbreviations

AR	Acceptance Review	see Sect. 6.2.1
ARMA	Auto Regressive Moving Average	see Sect. 1.1
BLUE	Best Linear Unbiased Estimator	see Sect. 3.1.1
BIQUE	Best Invariant Quadratic Unbiased Estimation	see Sect. 4.2
CDF	Cumulative Density Function	see Sect. 2.2.1
CHAMP	Challenging Minisatellite Payload	see Sect. 1
CG	Conjugate Gradient	see Sect. 1.1
CS	Core Solvers	see Sect. 1.1
EGM	Earth Gravitational Model	see Sect. 6.1.1
EGG-C	European GOCE Gravity Consortium	see Sect. 1.1
ESA	European Space Agency	see Sect. 6.2.1
DFAC	Drag Free and Attitude Control	see Sect. 6.2.1
DNA	Distributed Non-approximative Adjustment	see Sect. 1.1
GOCE	Gravity Field and Steady-State Ocean Circulation Explorer	see Sect. 1
GMM	Gauss-Markov-Model	see Sect. 3.1.1
GPS	Global Positioning System	see Sect. 5.1
GRACE	Gravity Recovery And Climate Experiment	see Sect. 1
GRF	Gradiometer Reference Frame	see Sect. 6.2.1
GOCE-GRAND II	GOCE-GRavitationsfeld-ANalyse Deutschland II	see Sect. 5.1
HPF	High Level Processing Facility	see Sect. 1.1
IGG	Institute for Geodesy and Geoinformation	see Sect. 1.1
IML	Iterated Maximum Likelihood	see Sect. 4.2
IAPG	Institute of Astronomical and Physical Geodesy	see Sect. 1.1
INAS	Institute of Navigation and Satellite Geodesy	see Sect. 1.1
JUMP	Jülich Multi Processor	see Sect. 6.2.2
MC	Monte Carlo	see Sect. 2.1.1
MCMC	Markov Chain Monte Carlo	see Sect. 3.2
MCVCE	Monte Carlo Variance Component Estimation	see Sect. 4.3.1
MCVCM	Monte Carlo Variance/Covariance Matrix	see Sect. 3
MINQUE	Minimum Norm Quadratic Unbiased Estimation	see Sect. 4.2
LSQR	Least Squares QR Decomposition	see Sect. 1.1
PCGMA	Preconditioned Conjugate Gradients Multiple Adjustment	see Sect. 1.1
PDF	Power Density Function	see Sect. 2.1.2
POD	Precise Orbit Determination	see Sect. 6.1.1
QL-GFA	Quick-Look Gravity Field Analysis	see Sect. 1.1
SGG	Satellite Gravity Gradiometry	see Sect. 1.1
SST	Satellite-to-Satellite Tracking	see Sect. 1.1
TUM	Technical University Munich	see Sect. 1.1
TUG	Technical University Graz	see Sect. 1.1
VCE	Variance Component Estimation	see Sect. 1.1
VCM	Variance/Covariance Matrix	see Sect. 1.1

List of mathematical Symbols

Scalar	
λ	Regularization parameter
ω	Weighting parameter
$\hat{\sigma_i^2}$	Variance component of the i^{th} group
$\hat{\sigma_\mu^2}$	Variance component of the regularization term
r_i	Partial redundancy of the i^{th} group
r_μ	Partial redundancy of the regularization term
n	Number of observations
m	Number of parameters
$\mathfrak{X}$	Random variable

Matrix	
$\boldsymbol{A}$	Design matrix $\in \mathbb{R}^{n\times m}$
$\boldsymbol{P}$	Weight matrix $\in \mathbb{R}^{n\times n}$
$\boldsymbol{\Sigma}$	Covariance matrix of observations $\in \mathbb{R}^{n\times n}$
$\boldsymbol{\Sigma}\{\widehat{\mathfrak{X}}\}$	Covariance matrix of parameters $\in \mathbb{R}^{m\times m}$
$\boldsymbol{P}_\mu$	Regularization matrix $\in \mathbb{R}^{m\times m}$
$\boldsymbol{N}$	Normal equation matrix $\in \mathbb{R}^{m\times m}$
$\boldsymbol{N}_\oplus$	Preconditioning matrix $\in \mathbb{R}^{m\times m}$
$\boldsymbol{R}$	Upper triangular matrix $\in \mathbb{R}^{m\times m}$
$\boldsymbol{I}$	Identity matrix

Vector	
$\hat{\boldsymbol{\sigma}}$	Vector of variance componets $\in \mathbb{R}^o$
$\boldsymbol{\mu}$	Vector of apriori parameters $\in \mathbb{R}^m$
$\boldsymbol{x}$	Vector of parameters $\in \mathbb{R}^m$
$\boldsymbol{l}$	Vector of observations $\in \mathbb{R}^n$
$\boldsymbol{\mathfrak{X}}$	Vector of random variables
$\boldsymbol{\mathfrak{L}}$	Random vector of observations
$\boldsymbol{\xi}$	Vector of true values of parameter
$\boldsymbol{\lambda}$	Vector of true values of observations
$\mathbf{x}$	Realization of a random vector $\in \mathbb{R}^n$
$\boldsymbol{b}$	Right hand side vector of the normal equation system
$\boldsymbol{\alpha}$	Auxiliary parameter vector of groups given by observation equations
$\boldsymbol{\beta}$	Auxiliary parameter vector of groups given by normal equation matrix

Operations	
tr $\boldsymbol{B}$	Trace of matrix $\boldsymbol{B}$
det $\boldsymbol{B}$	Determinant of matrix $\boldsymbol{B}$
$E\{.\}$	The expected value operator $\mathbb{R}^n \rightarrow \mathbb{R}^n$
$\Sigma\{.\}$	Covariance matrix $\mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$
$p_{\mathfrak{x}}(\mathbf{x})$	Power density function of random vector
$\|\boldsymbol{a}\|$	Norm of $\boldsymbol{a}$
$\boldsymbol{B}^T$	Transpose of matrix $\boldsymbol{B}$
$\boldsymbol{B}^{-1}$	Inverse of a positive definite matrix $\boldsymbol{B}$
$\boldsymbol{B}(i,i)$	The i^{th} entry of matrix $\boldsymbol{B}$
$\boldsymbol{B}(i,:)$	The i^{th} row of matrix $\boldsymbol{B}$
$\boldsymbol{B}(:,i)$	The i^{th} column of matrix $\boldsymbol{B}$

1 Introduction

The Earth's gravity field has been subject to intensive research for a few years. This research aims at improving our understanding in many fields of application, particularly in oceanography, geodynamics, and geodesy. In the field of oceanography, the sea level can be determined with an accuracy of a few centimeters. This surface is compared with the geoid to determine, for instance, the amount of sea currents. This is one reason why the accuracy of the geoid itself is required to be a few centimeters. In geodynamics it is also fundamental to have the Earth's gravity field available as it constitutes a side condition for seismic models of the Earth's crust. In geodesy, an accurate determination of the Earth's gravity field is needed to improve the geoid, as the equipotential surface of the Earth's gravity field, as a reference surface to various height systems. A detailed study regarding application and use of the Earth's gravity field is found in ESA (1999).
Only satellite missions, designed in particular for the purpose of gravity field determination, namely, the CHAMP (Challenging Minisatellite Payload, cf. Reigber et al. 2004) mission, the current GRACE (Gravity Recovery and Climate Experiment, cf. Tapley et al. 2005) mission and the future GOCE (Gravity and Steady State Ocean Circulation Explorer, cf. ESA 2002) mission have the potential to reach an accuracy of a few centimeters with respect to geoid heights derived from the Earth's gravity field.

1.1 Numerical challenge of GOCE

Modeling the gravity field requires that thousands of parameters are estimated from a huge amount of data in each of the above mentioned satellite missions. Finding computational techniques that are capable of handling very large sets of data and parameters is a great challenge.
In this thesis, we will focus on the GOCE mission, which is based on the sensor concepts of Satellite to Satellite Tracking (SST) and Satellite Gravity Gradiometry (SGG). The SGG and SST observations obtained by these sensors are used to estimate, by means of a least squares adjustment, the parameters of the Earth's gravity field in terms of spherical harmonic coefficients up to degree and order 240, resulting in about 60,000 unknown parameters.

In the last few years, a numerical determination of the gravity field from such huge systems has been based on a number of strategies, which may be divided into two main categories: the *space-wise approach* and the *time-wise approach*. The space-wise approach requires data to be transformed onto a regular grid by means of a rather complex algorithm. On the other hand, with the *time-wise* approach, the data are treated as time series along the satellite's orbit (see e.g. Rummel et al. 1993 for a general discussion of both approaches). The *space-wise* approach will not be further discussed in this thesis ; the reader is referred to Migliaccio et al. (2004) instead.

Within the context of the *time-wise* approach, a variety of processing strategies has been developed and investigated. In the framework of the ESA project "GOCE High-level Processing" (HPF, Rummel et al. 2004), an operational software for the scientific processing of GOCE data has been set up by the European GOCE Gravity Consortium (EGG-C, Rummel et al. 2004). One main task of this system is performed by the institute of navigation and satellite geodesy (INAS) at the Technical University of Graz (TUG) along with the Institute of Geodesy and Geoinformation (IGG) at the University of Bonn and the Institute of Astronomical and Physical Geodesy (IAPG) at the Technical University of Munich (TUM). Beside the so-called *quick-look gravity field analysis* (QL-GFA), which is based on the semi-analytic approach due to Sneeuw (2000) and Pail and Plank (2002), a diagnostic tool for quickly checking the quality of the input data, *core solvers* (CS) have evolved as tools for computing highly accurate solutions to the normal equations.

With respect to CS, one distinguishes between direct and iterative procedures for solving the normal equation system. In Plank (2004) and Pail and Plank (2002), a program system based on the *time-wise* approach was presented that computes high-resolution gravity models via a parallel and distributed assemblance of the fully populated normal equation matrix. This method is known as *distributed non-approximative adjustment* (DNA). Within the category of iterative solvers, Schuh (1996) and Boxhammer (2006) implemented a modified version of the *conjugate gradient* (CG) algorithm that avoids an explicit assemblance of the normal equation matrix. This strategy is based on a very flexible PCGMA algorithm (**P**reconditioned **C**onjugate **G**radients **M**ultiple **A**djustment) which is especially adapted to work with different types of observations.
Colored noise causes correlations between the SGG observations, which results in covariance matrix which has a banded structure with a broad band. In order to decorrelate these observations, Schuh (1996) and Pail and Plank (2002) proposed autoregressive moving average (ARMA) filter strategy as a whitening process.

In order to find an optimal solution to the unknown parameters, the reliable weighting factor between SGG and SST must be estimated. Unfortunately, the normal equation system is ill-conditioned due to the data gaps such as polar gap and the downward continuation problem. In order to overcome the ill-condition, a positive definite regularization matrix (scaled by an unknown regularization parameter) must be added to the normal equation matrix (which is known as Tikhonov regularization); see Kusche and Klees (2002) and Ditmar et al. (2003). To resolve both the choices of weighting factors and regularization parameters in large-scale least squares problems, the method of variance component estimation (VCE) has been demonstrated by Koch and Kusche (2002) to be suitable procedure.

In the course of the last years, the application of VCE for estimating weight factors and regularization parameters has became more and more relevant in the context of global gravity field determination. For instance, Mayer-Gürr et al. (2005) discussed a scenario where a one-year CHAMP orbit is divided into short arcs to establish the observation equations. For the combination of the normal equations of each short arc, optimal variance factors were estimated. A potential change in measurement accuracy from one arc to the next could then be taken into account, and arcs that were supposed to have outliers were thereby assigned lower weight factors. In van Loon and Kusche (2005), the method of VCE was used to estimate parameters of the stochastic model by means of an iterative procedure, because the stochastic model for the values of the potential from the energy balance is heterogeneous due to the varying quality of the GPS orbits. Bauer and Kusche (2006), in computing the optimum regularization parameters using the VCE, presents a tailored strategy for geopotential recovery from satellite data of GOCE and GRACE, which based on iterative least squares using QR decomposition (LSQR). The method of VCE has not only played a vital role in global determination of the Earth's gravity field, but also in regional gravity field determination. In Eicker et al. (2005), for example, the Earth's surface was divided into patches, and for each of these, an individual local gravity field was computed. The optimal regularization parameter for each of the patches was estimated by VCE. Yet another field of application for VCE is airborne gravimetry, where regularization parameters must also be estimated (see Mueller and Mayer-Guerr (2004)).

A further challenge for the GOCE mission is the computation of the full covariance information which characterizes the quality of the estimated parameters. In linear least squares, this means that the covariance matrix is given by the inverse matrix of the normal equations, multiplied by an estimated variance factor. In the light of the huge number of unknowns determined by the SGG observations, the computation of the corresponding normal equations and its inverse becomes in fact unfeasible, because of the size of the systems.

Unfortunately, previous articles only provide incomplete estimates of the covariance information or not are applicable for large-scaled variance propagation. For example, Tscherning et al. (1999) have given an approximative solution by taking into consideration only the diagonal part of the covariance matrix for variance propagation. Abwerzger (1999) used the partial inverse ('kite' structure) of the normal equations to compute the variances of second level products, employing however a very time consuming, iterative procedure. This approach is also limited because it is applicable only in the case that particular variances are of interest. A first implementation of a full variance propagation is given in Gundlich et al. (2003), see also Sect. 3.2.2, in which a Monte Carlo approach is used to overcome the huge computational burden. In particular, a Gibbs sampler (see Sec. 3.2.1) is adopted to compute the inverse of the normal equation matrix and to estimate the variance/covariance matrix (VCM) in a tailored and efficient way. Unfortunately, this approach requires the presence of a full normal equation matrix which is not available in iterative solvers

It follows that both the optimal estimation of gravity field parameters (taking into account the determination of weight factors as well as regularization parameters) and the computation of the full large-scaled covariance matrix of the parameters are essential to fill the gaps within an iterative solution procedure, specific to the PCGMA algorithm.

1.2 The goals of the work

The main purpose of this thesis is to present tailored direct and iterative solvers based on Monte Carlo techniques for computing covariance information as well as optimal variance components for huge data sets. The novelty of the proposed algorithms is that normal equations, observation equations, or combined equation systems may all be processed. Furthermore, they are applicable to direct solver as well as to iterative techniques e.g. con-

jugate gradients. It is demonstrated that these Monte Carlo strategies complement the PCGMA algorithm to estimate, in addition to the gravity field parameters, accurate variance/covariance information as well as the optimal weighting and regularization parameters. It should be mentioned that the development of a quality assessment of the computed gravity field, as well as the optimum weighting and regularization parameters are official products of the GOCE-HPF.

This thesis is organized as follows: In chapter 2 the basics and the algorithms for random sample generation are presented. In addition, a review of the basic theory with respect to Monte Carlo integration and error propagation of linear transformations is given.

In chapter 3, Monte Carlo methods are suggested for estimating the moments of a random sample. In particular, Monte Carlo algorithms for estimating the second moments, which is equivalent to computing the inverse of the normal matrix, are discussed. Chapter 3 presents a method to obtain the accuracy of the estimated covariance matrix after the process of generating samples. In addition, the variance reduction technique by blocking of the generated samples is applied.

Chapter 4 represents the linear model with unknown variance components for heterogeneous data and apriori information on the parameters. In addition, a Monte Carlo technique based on stochastic trace estimation for the normal equation is presented. The method is restructured and developed for the case of mixed observation equations and normal matrices.

In chapter 5, tailored versions of the algorithms described in chapters 3 and 4 are integrated into the iterative solver PCGMA. The required modifications of PCGMA are discussed in detail for the GOCE data combination.

In chapter 6, the numerical results for two test scenarios are presented for the purpose of validating the concepts and to demonstrate the performance of the proposed algorithms.

The seventh chapter is dedicated to a discussion of the proposed algorithms and concludes with an outlook to future investigations.

2 Theory of Monte Carlo methods

2.1 Introduction to Monte Carlo methods

2.1.1 A brief history of Monte Carlo methods

Monte Carlo (MC) methods belong to the most popular numerical statistical methods that make use of sequences of random numbers to perform simulations. MC simulations have been discovered and applied in the previous century.
In 1777 for instance, Comte de Buffon performed the following experiment: A needle of length L is thrown onto a board with equidistant parallel lines separated by distance d. He demonstrated that the probability of the event that the needle bisects a line is:

$$p = \frac{2L}{\pi d}. \tag{2.1}$$

Years later, Laplace proved that this method may be used for obtaining a rough estimate of the number π. Modern MC methods arose after the second world war at the National Laboratory in Los Alamos. They were given their name in 1942 by the physicist Fermi, who, while working on a nuclear reactor, was the first to generate random samples. The name is derived from the town of Monte Carlo which became famous through its casino. The reason for this is that the first tables with random numbers contained the roulette results in the casino of Monte Carlo. The problems related to games, were reason enough for science to tackle questions regarding the randomness of events. In 1946 Stanislaw Ulam proposed to simulate the course of neutrons by generating random numbers. Then, John von Neumann developed the approach in detail. There were also cases where random numbers were applied to solve numerical problems (Kalos and Whitlock 1986).
However, within the last few years MC methods developed rapidly and were used as mature standard procedure to solve complex numerical problems. Nowadays, MC methods are applied in various fields of research. They are typically used for simulation of complex physical experiments, but also for optimization of finance models (see e.g. Glasserman 2003). Furthermore, they are suitable for solving analytical problems such as high-dimensional integrals or particular types of differential equations with complex boundary conditions.
In this thesis MC methods are used for computation of large-scale covariance matrices (cf. chapter 3) and for estimation of the redundancy contributions of disjunctive observation groups (cf. chapter 4), respectively.

2.1.2 Components of a MC algorithm

In this subsection, the main building blocks of the MC method will be summarized. These components constitute the basis for most of the MC applications, and to understand them is essential for any readers who wish to either reproduce the Monte Carlo simulations given in this thesis or to conduct their own simulations. The primary components of a MC simulation may be described as follows:

1. Probability density function (PDF): The physical or mathematical system is described by one or a collection of PDFs.

2. Random number generator: Functions or routines that produce random numbers from a uniform distribution.

3. Sampling role: Various MC algorithms to produce random samples.

4. Parallelization and efficiency: Optimization of the MC algorithm for parallel computing can be done by serial efficiency and parallel speedup.

5. Scoring of the generated random samples: the generated samples are added up to estimate the unknown quantity.

6. Error estimations: Estimation of the statistical error (variance) as a function of all the generated random samples.

7. Variance reduction techniques: Techniques to minimize the error of the estimated solution. They play a decisive role to reduce the required number of samples in order to reduce the processing time of the simulation.

The first 3 parts of Fig. 2.1 represent the core components to generate random samples in a MC simulation. They are essential in every MC algorithm. The other components are primarily responsible for the analysis of the random samples and for increasing the efficiency in a MC simulation.

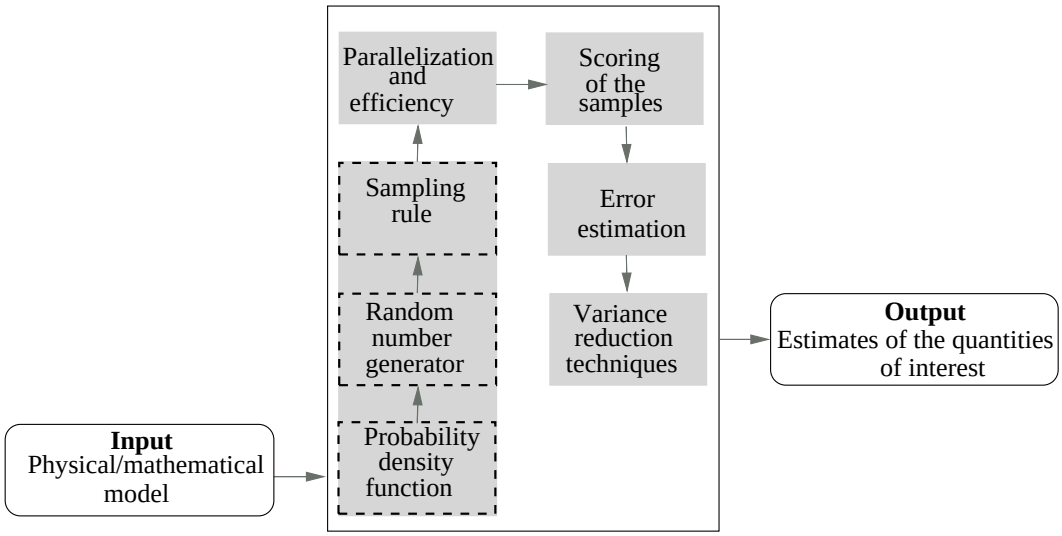


Fig. 2.1: The main components of a MC Algorithm

2.2 Some probability theory and statistical basics

The purpose of the current subsection is to give an overview of the concepts from traditional probability theory required for a general understanding of Monte Carlo methods. With regard to the generation of random samples, discrete and continuous random variables, sample PDFs, and quantities derived from such variables will be considered. The distributions required for chapters 3 and 4 will be defined. In order to estimate variance/covariance matrices (chapter 3) and variance components (chapter 4), samples must be produced, which are based on a generation of random numbers. For this purpose, procedures to generate random vectors will be presented. For a detailed introduction into probability theory, the reader is referred to Pitman (1993) and Feller (1968). Definitions and statistical concepts are found, e.g., in Koch (1999).

2.2.1 Random variables, PDF and CDF

A Monte Carlo simulation is given by a number of numerical computations of stochastic nature, because they are a sequence of probabilistic events. In probability theory, discrete and continuous random variables are distinguished. Definitions of discrete and continuous random variables are found, for instance, in Koch (1999) or Dudewicz and Mishra (1988). Only continuous random variables occur in this thesis. Therefore the discrete random variables will be abstract away from the following description.
The random variables within the continuous random vectors $\mathfrak{X}$ may take all the values $\mathfrak{X}_i$ with $-\infty < \mathfrak{X}_i < +\infty$, $i \in \{1, \cdots, n\}$. The cumalative distribution function (CDF) $F(\mathbf{x})$ of the random vector $\mathfrak{X}$ is given by the probability of the event $\mathfrak{X}_1 < x_1, \cdots, \mathfrak{X}_n < x_n$, i.e. that the corresponding random variables take values less than $x_1, \cdots, x_n$ (Koch 2000, p. 24), in symbols:

$$F(\mathbf{x}) = F(x_1, \cdots, x_n) = P\{\mathfrak{X}_1 < x_1, \cdots, \mathfrak{X}_n < x_n\}. \tag{2.2}$$

The continuous distribution function is defined as:

$$F(x_1, \cdots, x_n) = \int_{-\infty}^{x_n} \cdots \int_{-\infty}^{x_1} p_{\mathfrak{x}}(t_1, \cdots, t_n) dt_1 \cdots dt_n, \tag{2.3}$$

where $t_1, \cdots, t_n$ are the integration variables. The Eq. 2.3 can be rewritten in a vector notation as:

$$F(\mathbf{x}) = \int_{-\infty}^{x_n} \cdots \int_{-\infty}^{x_1} p_{\mathfrak{x}}(\mathbf{t}) d(\mathbf{t}). \tag{2.4}$$

The continuous probability density function is obtained from:

$$p_{\mathfrak{x}}(\mathbf{x}) = \frac{\partial^n F(\mathbf{x})}{\partial x_1 \partial x_2 \cdots \partial x_n}. \tag{2.5}$$

In addition, the following conditions hold for the integration domain χ of the random vectors $\boldsymbol{\mathfrak{X}}$:

$$p_{\mathfrak{x}}(\mathbf{x}) \geq 0 \qquad \text{and} \qquad \int_{\chi} p_{\mathfrak{x}}(\mathbf{x}) d\mathbf{x} = 1. \tag{2.6}$$

The distributions required in chapters 3 and 4 shall be described in Sect. 2.2.4; see , e.g., Johnson and Kotz (1970b), Johnson and Kotz (1970c), Koch (1999), and Koch (2000).

2.2.2 Conditional distribution

If a random vector $\boldsymbol{\mathfrak{X}}$ is partitioned into two partial vectors according to

$$\boldsymbol{\mathfrak{X}} = (\boldsymbol{\mathfrak{X}}_1, \boldsymbol{\mathfrak{X}}_2)^T, \boldsymbol{\mathfrak{X}}_1 = [\mathfrak{X}_1, \cdots, \mathfrak{X}_i]^T \qquad \text{and} \qquad \boldsymbol{\mathfrak{X}}_2 = [\mathfrak{X}_{i+1}, \cdots, \mathfrak{X}_n]^T, \tag{2.7}$$

then the conditional probability density $p_{\mathfrak{x}}(\mathbf{x}_1|\mathbf{x}_2)$ is given by the probability density of the vector $\boldsymbol{\mathfrak{X}}_1$ of random variables evaluated at $\mathbf{x}_1$ under the condition that, for the vector $\boldsymbol{\mathfrak{X}}_2$ of random variables, $\boldsymbol{\mathfrak{X}}_2 = \mathbf{x}_2$ holds. The relation between the jointly PDF $p_{\mathfrak{x}}(\mathbf{x}_1, \mathbf{x}_2)$, the marginal PDF $p_{\mathfrak{x}}(\mathbf{x}_1)$ and $p_{\mathfrak{x}}(\mathbf{x}_2)$, respectively and the conditional PDF are given as (cf. Koch 1999, p. 91):

$$p_{\mathfrak{x}}(\mathbf{x}_1, \mathbf{x}_1) = p_{\mathfrak{x}}(\mathbf{x}_1) p_{\mathfrak{x}}(\mathbf{x}_2|\mathbf{x}_1) = p_{\mathfrak{x}}(\mathbf{x}_2) p_{\mathfrak{x}}(\mathbf{x}_1|\mathbf{x}_2). \tag{2.8}$$

2.2.3 Expected values, variance and covariance

Definitions of the expected values and the covariance matrix of continuous random variable are found in Koch (2000, p. 38-43). For the expectation value, one has

$$E_{p_{\mathfrak{x}}}\{\mathfrak{X}_i\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} x_i p_x(x_1, \cdots, x_n) dx_1 \cdots dx_n. \tag{2.9}$$

Accordingly, one obtain for the expectation value $E\{f(\mathfrak{X}_i)\}$ of the function $f(\mathfrak{X}_i)$

$$E_{p_{\mathfrak{x}}}\{f(\mathfrak{X}_i)\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_i) p_x(x_1, \cdots, x_n) dx_1 \cdots dx_n. \tag{2.10}$$

The variance $V\{\mathfrak{X}_i\}$ of the random variable $\mathfrak{X}_i$ (the second central moment) is defined by:

$$V_{p_{\mathfrak{x}}}\{\mathfrak{X}_i\} = E\left\{(\mathfrak{X}_i - E\{\mathfrak{X}_i\})^2\right\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_i - E\{\mathfrak{X}_i\})^2 p_x(x_1, \cdots, x_n) dx_1 \cdots dx_n. \tag{2.11}$$

The covariance $\Sigma(\mathfrak{X}_i, \mathfrak{X}_k)$ of two random variables $\mathfrak{X}_i$ and $\mathfrak{X}_k$ is defined by:

$$\begin{aligned} \Sigma_{p_{\mathfrak{x}}}\{\mathfrak{X}_i, \mathfrak{X}_k\} &= E\{(\mathfrak{X}_i - E\{\mathfrak{X}_i\})(\mathfrak{X}_k - E\{\mathfrak{X}_k\})\} \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} ((x_i - E\{\mathfrak{X}_i\})(x_k - E\{\mathfrak{X}_k\})) p_x(x_1, \cdots, x_n) dx_1 \cdots dx_n. \end{aligned} \tag{2.12}$$

If the single random variables $\mathfrak{X}_1, \cdots, \mathfrak{X}_n$ are combined to a random vector $\boldsymbol{\mathfrak{X}}$, then Eq. (2.9) may be written as:

$$E_{p_{\mathfrak{x}}}\{\boldsymbol{\mathfrak{X}}\} = \int_{\chi} \mathbf{x} p_{\mathfrak{x}}(\mathbf{x}) d\mathbf{x}, \tag{2.13}$$

and for the function of a random vector:

$$E_{p_{\mathfrak{x}}}\{\boldsymbol{f}(\boldsymbol{\mathfrak{X}})\} = \int_{\chi} \boldsymbol{f}(\mathbf{x}) p_{\mathfrak{x}}(\mathbf{x}) d\mathbf{x}. \tag{2.14}$$

Analogous to Eq. 2.12, the covariance matrix of a random vector $\mathfrak{X}$ can be written as:

$$\Sigma_{p_{\mathfrak{x}}}\{\mathfrak{X}\} = \int_{\chi} (\mathbf{x} - E\{\mathfrak{X}\})\,(\mathbf{x} - E\{\mathfrak{X}\})^T\, p_{\mathfrak{x}}\,(\mathbf{x})\, d\mathbf{x}, \tag{2.15}$$

where the main diagonal comprises the variances of the random vector and the off-diagonals the covariances.

For the sake of simplicity, the notation $E\{\mathfrak{X}\}$ will be used instead of $E_{p_{\mathfrak{x}}}\{\mathfrak{X}\}$ and $E\{\boldsymbol{f}(\mathfrak{X})\}$ instead of $E_{p_{\mathfrak{x}}}\{\boldsymbol{f}(\mathfrak{X})\}$, keeping in mind that the expectation is always based on an associated PDF. In addition, the covariance information $\Sigma_{p_{\mathfrak{x}}}\{\boldsymbol{f}(\mathfrak{X})\}$ is simplified in the short notation $\Sigma\{\boldsymbol{f}(\mathfrak{X})\}$ and the PDF $p_{\mathfrak{x}}(\mathbf{x})$ in the short notation $p(\mathbf{x})$.

Error Propagation

In order to obtain the covariance matrix $\boldsymbol{\Sigma}\{\mathfrak{Y}\}$ of a transformed random vector

$$\mathfrak{Y} = \mathbf{A}\mathfrak{X} + \mathbf{b}, \tag{2.16}$$

where $\mathbf{A}$ is a constant $m \times n$ matrix and $\mathbf{b}$ a constant $m \times 1$ vector, the law of error propagation is applied (see, e.g., Koch 1999, p. 99):

$$\begin{aligned}
\boldsymbol{\Sigma}\{\mathfrak{Y}\} &= E\left\{(\mathfrak{Y} - E\{\mathfrak{Y}\})(\mathfrak{Y} - E\{\mathfrak{Y}\})\}^T\right\} \\
&= E\left\{(\mathbf{A}\mathfrak{X} + \mathbf{b} - E\{\mathbf{A}\mathfrak{X} - \mathbf{b}\})(\mathbf{A}\mathfrak{X} + \mathbf{b} - E\{\mathbf{A}\mathfrak{X} - \mathbf{b}\})^T\right\} \\
&= \mathbf{A}E\{(\mathfrak{X} - E\{\mathfrak{X}\})(\mathfrak{X} - E\{\mathfrak{X}\})^T\}\mathbf{A}^T \\
&= \mathbf{A}\boldsymbol{\Sigma}\{\mathfrak{X}\}\mathbf{A}^T.
\end{aligned} \tag{2.17}$$

2.2.4 Special continuous distribution functions

Uniform distribution

A random variable $\mathfrak{X}$ is said to be uniformly distributed within the domain (a, b), in symbols $\mathfrak{X} \sim U(a, b)$, if its PDF is given by:

$$p(x) = \begin{cases} \frac{1}{a-b} & \text{for all } x \in [a, b] \\ 0 & \text{else.} \end{cases} \tag{2.18}$$

Normal distribution

The normal distribution is the most widely used distribution in statistics. A random variable $\mathfrak{X}$ is called normally distributed with parameters μ and σ^2, in symbols $\mathfrak{X} \sim \mathcal{N}(\mu, \sigma^2)$, if its PDF is given by:

$$p\,(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-0.5\left(\frac{x-\mu}{\sigma}\right)^2\right\} \qquad \text{for } (-\infty < x < +\infty). \tag{2.19}$$

The standardized normal distribution $\mathcal{N}(0, 1)$ is obtained by means of the variable transformation:

$$\mathfrak{Z} = \frac{\mathfrak{X} - \mu}{\sigma}. \tag{2.20}$$

From Eq. (2.19) and Eq. (2.20) one gets (cf. Koch 1999, p. 107):

$$p\,(z) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{z^2}{2}\right\}. \tag{2.21}$$

The random variables $\mathfrak{X}_1, \cdots, \mathfrak{X}_n$, which build up the $n \times 1$ random vector $\mathfrak{X}$ has a multivariate normal distribution, written as $\mathfrak{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if its PDF is given by (see, e.g., Koch 1999, p. 117):

$$p\,(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}(\det \boldsymbol{\Sigma})^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}. \tag{2.22}$$

χ^2-distribution

Let random vector $\mathbf{x}$ be normally distributed with expectation $\mathbf{0}$ and covariance matrix $\boldsymbol{I}$ (the unity matrix), i.e. $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Then, the sum of squared vector components $v = \mathbf{x}^T\mathbf{x}$ has a χ^2-distribution with m degrees of freedom, in short $v \sim \chi^2(m)$, and its PDF is (Koch 1999, p. 124):

$$f\,(v) = \frac{1}{2^{m/2}\Gamma(m/2)} v^{(m/2)-1} \exp(-v/2) \qquad \text{for } 0 < v < +\infty, \tag{2.23}$$

where $\Gamma(m/2)$ is the gamma function of the integer $m/2$.

2.3 Sampling from probability distribution functions

Any Monte Carlo simulation requires random numbers. Random numbers are generated on a computer by means of deterministic procedures. Therefore, the numbers generated in this way are actually not random in a strict sense, but are rather called pseudo-random. More specifically one performs for each proposed pseudo number generator a series of different tests (GENTLE 2003, p. 61-86). If the outcome of one test differs significantly from what one would expect from a truly random sequence, the pseudo number generator is classified as bad. A good pseudo random number generator should satisfy many of criteria, which are summarized in GENTLE (2003).
In particular, uniformly distributed random numbers are generated, which may then in turn be transformed into pseudo-random numbers of random variables having other distributions (as explained in Sect. 2.3.2), for instance, into numbers of a normally distributed random variable (see Sect. 2.3.3).

2.3.1 Generation of uniformly distributed random numbers

The standard random number generators constitute the starting point for most Monte Carlo algorithms. By means of these random generators, realizations of random variables may be produced that are uniformly distributed on the unit interval $[0, 1]$. A widely used standard procedure to generate pseudo-random numbers makes use of the following linear congruence method (KOCH 2000, p. 183).

The random numbers are produced through an equation of the form:

$$U_{i+1} = (aU_i + b) \mod M \quad \text{with} \quad M \in \mathbb{N} \quad \text{and} \quad a, b, U_0 \in \{0, \cdots M-1\}\,. \tag{2.24}$$

Considerations for suitable choices of the constants as well as descriptions of other types of pseudo-random number generators, such as non-linear congruence generators, shift register generators, lagged Fibonacci generators, and combinations of such generators, can be found, e.g., in GLASSERMAN (2003) or RUBINSTEIN (1981).

2.3.2 General sampling methods

Starting from pseudo-random numbers $u_1, u_2, \cdots, u_n$ generated by one of the standard methods as outlined in Sect. 2.3.1, some random numbers $x_1, x_2, \cdots$ may be generated which may be viewed as realizations of random variables $\mathfrak{X}_1, \mathfrak{X}_2, \cdots, \mathfrak{X}_n$ with another distribution, for instance as realizations $x_1, x_2, \cdots, x_n$ of Poisson-, binomial-, or normally distributed random variables $\mathfrak{X}_1, \mathfrak{X}_2, \cdots, \mathfrak{X}_n$. This process makes in particular use of the so-called inversion method or acceptance-rejection method, whose basic ideas shall now be explained in the current section. A far more comprehensive discussion of such algorithms are found, e.g., in FISHMAN (2003), ROBERT and CASELLA (1999), or KOCH (2000).

Inversion method

The following property of a inverse function F^{-1} of an arbitrary monotone increasing distribution function $F(x)$ serves as the basis for generating pseudo-random numbers $x_1, x_2, \cdots, x_n$. If $\mathfrak{U}_1, \mathfrak{U}_2, \cdots, \mathfrak{U}_n$ are sequences of independent and uniformly distributed random variables on $[0, 1]$, then the random variables $\mathfrak{X}_1, \mathfrak{X}_2, \cdots, \mathfrak{X}_n$ with $\mathfrak{X}_i = F^{-1}(\mathfrak{U}_i)$ for every $i \in \{1, 2, \cdots, n\}$ are independent and have the distribution function F. The main disadvantage of this method is that the distribution function $F(x)$ and its inverse function must be analytically specified, which generally is not always possible. The following Alg. 2.1 illustrates this procedure:

Algorithmus 2.1 (Inversion method)

Purpose:	To generate a sample $\mathfrak{X}$ from $p(x)$
Output:	Realization of $\mathfrak{X}$

1. Generate the random value u for the random variable $\mathfrak{U} \sim \mathcal{U}(0, 1)$.
2. Set u equal to the distribution function, that is: $F(x) = u$.
3. Invert the distribution function and isolate x, that is: $x = F^{-1}(u)$.

Acceptance-rejection method

The acceptance-rejection method is one of the most widely applicable method for generating random samples (GENTLE 2003, p. 113). This method generates samples from a target distribution by first generating candidates from a more convenient distribution and then rejecting a random subset of the generated candidates.
Suppose, that one wishes to generate samples from $p(x)$. Let $g(x)$ be a pdf which we know how to generate samples and with the property that (see, e.g., KOCH 2000, p. 186)

$$C \geq p(x)/g(x), \qquad \text{for all} \qquad x \in \mathbb{R}, \tag{2.25}$$

where C is a constant with $C \geq 1$. In the acceptance-rejection method, we generate sample $\mathfrak{X}$ from g and accept the sample with probability $p(x)/Cg(x)$; this can be implemented by sampling $\mathfrak{U}$ uniformly over $[0, 1]$ and accepting $\mathfrak{X}$ if $p(x) \leq Cg(x)$. If $\mathfrak{X}$ is rejected, a new candidate is sampled from $g(x)$ and the acceptance applied again. The process repeats until the acceptance test is passed; the accepted value is returned as a sample from $p(x)$. The following algorithm illustrates a generic implementation:

Algorithmus 2.2 (Acceptance-rejection method)

Purpose:	To generate a sample $\mathfrak{X}$ from $p(x)$
Output:	Realisations of $\mathfrak{X}$

1. Generate the random value x for the random variable $\mathfrak{X}$ with probability density function g.
2. Generate the random value u for the random variable $\mathfrak{U} \sim \mathcal{U}(0, 1)$.
3. Compute the fraction $q = \frac{p(x)}{Cg(x)}$.
4. If $u \leq q$, then accept the value x, otherwise go back to the first step.

2.3.3 Generation of standard-normally distributed random numbers

Normally distributed random numbers are produced by transforming uniformly distributed random numbers. For this purpose, various algorithms exist that are based on the methods described in Sect. 2.3.2. The Box-Muller (BOX and MULLER 1958) transformation method is probably the most popular algorithm for generating normally distributed random values. The idea behind this method is to transform realizations of uniformly distributed random variables $\mathfrak{U}_1, \mathfrak{U}_2 \sim \mathcal{U}(0, 1)$ into values of random variables $\mathfrak{X}_1, \mathfrak{X}_2 \sim \mathcal{N}(0, 1)$. The computational steps are summarized in Alg. 2.3 according to (GENTLE 2003, p. 172):

Algorithmus 2.3 (Box-Muller Algorithm)

Purpose:	To generate a sample $\boldsymbol{\mathfrak{X}}$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$
Output:	Realisations of $\boldsymbol{\mathfrak{X}}$

1. Generate realizations of random variables $\mathfrak{U}_1, \mathfrak{U}_2 \sim \mathcal{U}(0, 1)$.
2. Compute the transformations:

$$x_1 = \sqrt{-2 \ln u_1} \cos(2\pi u_2) \qquad \text{and} \qquad x_2 = \sqrt{-2 \ln u_1} \sin(2\pi u_2)$$

3. The resulting realizations of the random variables are $\mathcal{N}(0, 1)$-distributed.

The computational cost of Alg. 2.3 may be reduced by avoiding trigonometric functions, which is accomplished, for instance, through a transformation into polar coordinates (see, e.g., GENTLE 2003, p. 173). Modern algorithms such as the *Ziggurat* algorithm (GENTLE 2003, p. 174) even avoid the evaluations of the roots and are based on purely multiplicative operations.

The standard version of MATLAB (THE MATH WORKS 2006A) and the Statistics Toolbox (THE MATH WORKS 2006B) comprise a random number generator which produces scalar $\mathcal{N}(0, 1)$-distributed numbers. As it will be seen later on, the generation of scalars is in fact sufficient as one may easily arrive at a general multivariate

normal distribution through a suitable transformation of the one-dimensional standard normal distribution (see Sect. 2.3.4). The first step is to extend a one-dimensional standard-normally distributed random number $\mathfrak{X} \sim \mathcal{N}(0,1)$ to a multi-dimensional standard-normally distributed random vector $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. For this purpose, it is only necessary to have the random number generator produce a number of random numbers that equals the number of components within the random vector, and to arrange these random numbers as a vector. As the generated numbers are practically uncorrelated and have a variance equal to 1, the random vector constructed in this way already follows a multivariate standard-normal distribution, i.e. $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

2.3.4 Generation of correlated normally-distributed random numbers

The multinormal distribution is a building block for some of the algorithms devolved in chapter 3. Therefore, we include their generation here. It is well known that the multinormal distribution is fully characterized by its expected value $\boldsymbol{\mu} \in \mathbb{R}^n$ and its variance-covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ (cf. Eq.2.22). Since it is no problem to add $\boldsymbol{\mu}$ after generation we will assume that $\boldsymbol{\mu} = \mathbf{0}$ (see, e.g., Gentle 2003, p. 197).
Consider a vector $\boldsymbol{\mathfrak{Z}} = (\mathfrak{Z}_1, \cdots, \mathfrak{Z}_N)^T$ of independent standard-normally distributed random variables $\boldsymbol{\mathfrak{Z}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. As $\boldsymbol{\Sigma}$ is positive definite then there exists the Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$, where $\boldsymbol{R}$ is an $n \times n$ upper triangular matrix. Further, let

$$\boldsymbol{\mathfrak{X}} = \boldsymbol{R}^T\boldsymbol{\mathfrak{Z}}. \tag{2.26}$$

The random vector $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. This can be proven by

$$E\{(\boldsymbol{\mathfrak{X}} - E\{\boldsymbol{\mathfrak{X}}\})(\boldsymbol{\mathfrak{X}} - E\{\boldsymbol{\mathfrak{X}}\})^T\} = E\{\boldsymbol{\mathfrak{X}}\boldsymbol{\mathfrak{X}}^T\} \tag{2.27}$$

Substituting the transformation 2.26 into 2.27 produces:

$$\begin{aligned} E\{\boldsymbol{\mathfrak{X}}\boldsymbol{\mathfrak{X}}^T\} =& E\{\boldsymbol{R}^T\boldsymbol{\mathfrak{Z}}(\boldsymbol{R}^T\boldsymbol{\mathfrak{Z}})^T\} \\ E\{\boldsymbol{\mathfrak{X}}\boldsymbol{\mathfrak{X}}^T\} =& E\{\boldsymbol{R}^T\boldsymbol{\mathfrak{Z}}\boldsymbol{\mathfrak{Z}}^T\boldsymbol{R}\} \\ E\{\boldsymbol{\mathfrak{X}}\boldsymbol{\mathfrak{X}}^T\} =& \boldsymbol{R}^T \underbrace{E\{\boldsymbol{\mathfrak{Z}}\boldsymbol{\mathfrak{Z}}^T\}}_{\boldsymbol{I}} \boldsymbol{R} \\ =& \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{\Sigma} \end{aligned} \tag{2.28}$$

Accordingly, $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ holds. Now, a $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$-distributed random vector $\boldsymbol{\mathfrak{Y}}$ is generated. This follows from the transformation:

$$\boldsymbol{\mathfrak{Y}} = \boldsymbol{\mu} + \boldsymbol{\mathfrak{X}}. \tag{2.29}$$

This yields the following algorithm:

Algorithmus 2.4 (Generation of $\mathcal{N}(\mu, \Sigma)$-distributed random vectors)

Purpose:	To generate a sample $\boldsymbol{\mathfrak{X}}$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
Input:	Expected vector $\boldsymbol{\mu}$ and the symmetrical positive definite matrix $\boldsymbol{\Sigma}$
Output:	Realisations of $\boldsymbol{\mathfrak{Y}}$

1. Compute the Cholesky decomposition, that is: $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$.
2. Generate a realization of an independently and normally-distributed random vector $\boldsymbol{\mathfrak{Z}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
3. Compute the transformed $\boldsymbol{\mathfrak{X}} = \boldsymbol{R}^T\boldsymbol{\mathfrak{Z}}$, $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$
4. Compute transformed realizations according to $\boldsymbol{\mathfrak{Y}} = \boldsymbol{\mu} + \boldsymbol{\mathfrak{X}}$.
5. The vector $\boldsymbol{\mathfrak{Y}}$ is $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$-distributed.

2.4 Basic Monte Carlo Integration

The generation of random numbers is of great importance for the evaluation of integrals by means of Monte Carlo methods. As far as the applications in chapter 3 are concerned, a simple Monte Carlo integration method

will be discussed.

The original Monte Carlo approach was developed as a method using random number generation to compute integrals (KALOS and WHITLOCK 1986; LIU 2001). Let $\mathfrak{Z}$ be a real-valued random variable, $p(z)$ ($z \in \mathbb{R}$) its pdf, and $\chi \subset \mathbb{R}$ the domain of integration. Then, the integral

$$\int_{\chi} f(z)p(z)dz = E\{f(\mathfrak{Z})\} \tag{2.30}$$

represents the expected value of an arbitrary random function $f(\mathfrak{Z})$ over $p(z)$. Considering a large number of identically and independently distributed (i.i.d.) samples $s_z^{(1)}, \ldots, s_z^{(M)}$ with pdf $p(z)$, the expectation may be estimated by:

$$\widehat{E}\{f(\mathfrak{Z})\} = \frac{1}{M}\sum_{i=1}^{M} f(s_z^{(i)}) \; . \tag{2.31}$$

This procedure is called Monte Carlo integration and may be extended to a multivariate approach involving a real-valued random m-vector $\boldsymbol{\mathfrak{Z}}$ and a real-valued random n-vector function $\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})$. In this case, the expectation vector is estimated by

$$\widehat{E}\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{f}(\boldsymbol{s}_z^{(i)}) \; , \tag{2.32}$$

where $\boldsymbol{s}_z^{(i)}$ denote i.i.d. samples, each with joint pdf $p(\boldsymbol{z})$ ($\boldsymbol{z} \in \mathbb{R}^m$).
The covariance information $\Sigma\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}$, is obtained from the second central moment of $\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})$, that is:

$$\begin{aligned}\Sigma\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\} &= E\Big\{\Big(\boldsymbol{f}(\boldsymbol{\mathfrak{Z}}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)\Big(\boldsymbol{f}(\boldsymbol{\mathfrak{Z}}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)^T\Big\} \\ &= \int_{\mathcal{X}}\Big(\boldsymbol{f}(\boldsymbol{\mathfrak{Z}}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)\Big(\boldsymbol{f}(\boldsymbol{\mathfrak{Z}}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)^T p(\boldsymbol{z})d\boldsymbol{z}. \end{aligned} \tag{2.33}$$

The entries of (2.33) can be numerically determined by using Monte Carlo integration via drawing M vectors of samples $\boldsymbol{s}_z^{(i)}$ from $p(\boldsymbol{z})$ and computing:

$$\widehat{\Sigma}\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\} = \frac{1}{M}\sum_{i=1}^{M}\Big(\boldsymbol{f}(\boldsymbol{s}_z^{(i)}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)\Big(\boldsymbol{f}(\boldsymbol{s}_z^{(i)}) - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)^T \; . \tag{2.34}$$

In contrast to the standard error propagation procedure mentioned above, which is restricted to the computation of the variance $\Sigma\{\boldsymbol{F}\boldsymbol{\mathfrak{Z}}\}$ of a linear relation $\boldsymbol{\mathfrak{F}} = \boldsymbol{F}\boldsymbol{\mathfrak{Z}}$ (or to a linearized function), the random vector $\boldsymbol{s}_{\boldsymbol{z}}^{(i)}$ is here transformed directly according to $\boldsymbol{s}_{\boldsymbol{f}}^{(i)} = \boldsymbol{f}(\boldsymbol{s}_z^{(i)})$, and the variance is obtained by:

$$\widehat{\Sigma}\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\} = \frac{1}{M}\sum_{i=1}^{M}\Big(\boldsymbol{s}_{\boldsymbol{f}}^{(i)} - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)\Big(\boldsymbol{s}_{\boldsymbol{f}}^{(i)} - E\{\boldsymbol{f}(\boldsymbol{\mathfrak{Z}})\}\Big)^T, \tag{2.35}$$

i.e. through averaging the dyadic products of the centralized random vectors $\boldsymbol{s}_{\boldsymbol{f}}^{(i)}$.

In Sect. 3.4 the convergence of the Monte Carlo integration with respect to the number of generated samples will be discussed. Furthermore a procedure for improving the rate of convergence will be presented in Sect. 3.5.

3 Monte Carlo method for estimation of the VCM

The calculation of the inverse of a large positive definite matrix is a common task in many geodetic related problem. In particular, there are applications where it is desired to obtain the least squares estimates of the parameters of a large linear model. Moreover, to obtain the variances or the covariances of the least squares estimated parameters, one needs not only the solution of the normal equations, but also the full inverse matrix of the normal equations, possibly multiplied by an estimated variance factor. We assume n to be the number of observation and m to be the number of unknown parameters then it takes $\mathcal{O}(nm^2)$ floating-point operations to set up the normal equation matrix and $\mathcal{O}(\frac{1}{2}m^3)$ floating-point operations to calculate its inverse (see, e.g., Schuh 2001). In case of insufficient RAM the computing time of the inverse increases significantly.

Moreover, not only the covariance matrix of the estimated parameters are of interest but also the linear as well as the non-linear functions of the parameters with their corresponding accuracies. For example the adjustment of satellite data results in harmonic coefficients to represent the Earth's gravity field. The calculation of the covariance matrix of other functionals (e.g., geoid heights, gravity anomalies) of the gravity field from those coefficients can be done by error propagation (cf. Eq. 2.17). If the quantities of interest are derived by nonlinear transformations from the estimated parameters, Monte Carlo simulation may then be the only feasible method. For example, by nonlinear transformations in Koch (2005), the generated random vectors are converted to random values of the square roots of degree variances, of mean squares of geoid undulations and geoid anomalies.

A research area at the focus of this thesis, occurs with the simulation and application of the developed Monte Carlo methods on the new satellite missions such as GOCE. However, it is expressly emphasized that the methods presented in this chapter are universally applicable to any adjustment problems, whereas the analysis of GOCE data in chapter 5 solely serves as an example without limiting the generality of this procedure.
The current chapter is organized as follows: In Sect. 3.1 the terminology of the linear model and the method of least squares to estimate the unknown parameters and their covariance matrix is elaborated. Sect. 3.2 deals with the use of Gibbs sampler for the computation of the covariance matrices. In Sect. 3.2, Monte Carlo variance/covariance matrix (MCVCM)-Algorithms for random sample generation are presented. That section is subdivided into four parts: first, a direct application to observation equations is discussed; second, the decorrelation of observation equations is presented; third, a simple approach with respect to normal equations is given; and fourth, an algorithm for combined heterogeneous systems is developed. The evaluation of the accuracy of Monte Carlo integration as one of the crucial aspects of this study is described in Sect. 3.4. For this purpose, confidence regions for propagated variances are derived, and an answer to the important question of the relation between accuracy and the number of samples is given. In Sect. 3.5, a stepwise estimation process by conditioning is shown to take advantage of prior information of the preconditioner to increase the accuracy of the estimation process. Finally, the efficiency estimations of the MCVCM-Algorithms are introduced in Sect. 3.6 for the purpose of parallelization of the Monte Carlo algorithms.

3.1 The linear Gauss-Markov model

3.1.1 Definition

Based on a random vector $\mathfrak{L}$ containing n observations, we would like to estimate m unknown parameters, which are summarized in a random vector $\mathfrak{X}$. The relation between the observed quantities and the unknown parameters will be described by a function $f(\mathfrak{X})$. In the linear case of the Gauss-Markov-Model (GMM) this relation can be formulated as follows (see, e.g., Koch 1999, p. 153):

$$E\{\mathfrak{L}\} = \boldsymbol{A}\boldsymbol{\xi}, \qquad \text{with} \qquad \Sigma\{\mathfrak{L}\} = \sigma^2 \boldsymbol{P}^{-1}. \tag{3.1}$$

The function $E\{\mathfrak{L}\} = \boldsymbol{A}\boldsymbol{\xi}$, where $\boldsymbol{\xi} = E\{\mathfrak{X}\}$ is the true values of the parameters, expresses the expectations $\boldsymbol{E}\{\mathfrak{L}\}$ of the observations $\mathfrak{L}$ as a linear combination of the unknown m parameters by means of matrix of coefficients $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, also called the design matrix, which will be assumed to have full column rank. This relation is also called the functional model. An important aspect of the GMM is the stochastic model which is described by the positive definite variance/covariance matrix $\boldsymbol{\Sigma}\{\mathfrak{L}\} = \sigma^2 \boldsymbol{P}^{-1}$. This matrix can be calculated by means of the known weight matrix $\boldsymbol{P}$ of the observation and by a unknown variance factor σ^2, which describes the general level of variance between all observations. In the sense of least square adjustment the number of observations n is always larger than the number of unknown parameters m ($n > m$). In this case the system

of equations $\boldsymbol{\mathfrak{L}} = \boldsymbol{A}\boldsymbol{\mathfrak{X}}$ becomes inconsistent. In order to solve this problem, the best linear unbiased estimator (BLUE) has to be determined.

3.1.2 Best linear unbiased estimation

We look for a linear estimator

$$\boldsymbol{\mathfrak{X}} = \boldsymbol{B}\boldsymbol{\mathfrak{L}}, \tag{3.2}$$

which is unbiased

$$\boldsymbol{\xi} = E\{\boldsymbol{\mathfrak{X}}\} = E\{\boldsymbol{B}\boldsymbol{\mathfrak{L}}\} = \boldsymbol{B}E\{\boldsymbol{\mathfrak{L}}\} = \boldsymbol{B}\boldsymbol{A}\boldsymbol{\xi}. \tag{3.3}$$

Therefore $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}$ must hold. In addition the best estimator denoted by $\widehat{\boldsymbol{B}}$ must have minimal variance

$$\Sigma\{\widehat{\boldsymbol{B}}\Sigma\{L\}\widehat{\boldsymbol{B}}^T\} \qquad \rightarrow \qquad \min \tag{3.4}$$

This means, that also each linear transformed vector $\boldsymbol{\mathfrak{F}} = \boldsymbol{F}\boldsymbol{\mathfrak{X}}$ has a minimal variance. As Meissl (1982) shows, are this conditions sufficient to determine $\widehat{\boldsymbol{B}}$ uniquely by

$$\widehat{\boldsymbol{B}} = \left(\boldsymbol{A}^T\Sigma\{\boldsymbol{\mathfrak{L}}\}^{-1}\boldsymbol{A}^T\right)^{-1}\boldsymbol{A}^T\Sigma\{\boldsymbol{\mathfrak{L}}\}^{-1}, \tag{3.5}$$

where $\Sigma\{\boldsymbol{\mathfrak{L}}\}^{-1}$ describes the metric of the vector space. In the following $\Sigma\{\boldsymbol{\mathfrak{L}}\}^{-1}$ will be denoted by $\boldsymbol{\Sigma}$.

3.2 Generation of samples based on MCMC methods

We have discussed in in Sect. 2.4 the important role of MC methods in evaluating integrals. The most critical step in developing an efficient MC algorithm is the sampling from an appropriate pdf $p(\boldsymbol{x})$. When directly generating independent samples from $p(\boldsymbol{x})$ is not possible, one have to produce statistically dependent samples based on the idea of *Markov Chain Monte Carlo* (MCMC) sampling (Liu 2001).
Markov Chain Monte Carlo (MCMC) methods have gained enormous popularity beyond mathematical statistics over the last few years. A general discussion of the MCMC methods is given in, e.g., Robert and Casella 1999, and a practical use of MCMC for sampling of solutions to inverse problems can be found in Mosegaard and Tarantola (1995) and Tarantola (2005). Other instances of the use of MCMC sampling for the Bayesian image reconstruction can be found in Koch (2006). Comprehensive accounts of MCMC methods and their applications may also be found in Gelman et al. (2004). The purpose of the next section is to give a brief overview of the commonly used MCMC sampling algorithm, the Gibbs sampler. In Sect. 3.2.2, the Monte Carlo estimation of the inverse of the normal equation matrix by generation of random sample based on the Gibbs sampler will be outlined.

3.2.1 Gibbs sampler

Gibbs sampling (Geman and Geman 1984) typically involves a partitioning of the random vector $\boldsymbol{\mathfrak{X}}$ into multiple blocks $\boldsymbol{\mathfrak{X}} = (\boldsymbol{\mathfrak{X}}_1, \cdots, \boldsymbol{\mathfrak{X}}_m)$. The density is then defined as the product of the conditional density of each block given the data and the remaining parameters. In each iteration step, each component of the sample is generated from the corresponding conditional density. To compute the 0^{th} iteration initial values must be given.
The density $p(x), \mathfrak{X} \in \mathcal{S} \subseteq \mathbb{R}^p$, from which the sample components are generated, is assumed to be known. The conditional density of the k^{th} block is denoted by $p(x_k|x_{-k}) = p(x_k|x_1, \cdots, x_{k-1}, x_{k+1}, \cdots, x_m)$. The Gibbs sampling algorithm comprises the following steps (see Koch 2000, p. 205):

Algorithmus 3.1

Purpose:	To generate a sample $\mathbf{x}$ using Gibbs sampler
Input:	Initial values of $\mathbf{x} = \mathbf{0}$
Output:	Realisations $\mathbf{x}$ of the random vector $\boldsymbol{\mathfrak{X}}$

1. Specify starting values $\mathbf{x}^{(0)} = (\mathbf{x}_1^{(0)}, \cdots, \mathbf{x}_m^{(0)})$ and set $i = 0$
2. Sample or update in turn
   - $\mathfrak{X}_1^{(i+1)}$ from $p(\mathbf{x}_1|\mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)}, \cdots, \mathbf{x}_m^{(i)})$
   - $\mathfrak{X}_2^{(i+1)}$ from $p(\mathbf{x}_2|\mathbf{x}_1^{(i+1)}, \mathbf{x}_3^{(i)}, \cdots, \mathbf{x}_m^{(i)})$
   - $\mathfrak{X}_3^{(i+1)}$ from $p(\mathbf{x}_3|\mathbf{x}_1^{(i+1)}, \mathbf{x}_2^{(i+1)}, \mathbf{x}_4^{(i)}, \cdots, \mathbf{x}_m^{(i)})$
   - $\vdots$
   - $\mathfrak{X}_m^{(i+1)}$ from $p(\mathbf{x}_m|\mathbf{x}_1^{(i+1)}, \mathbf{x}_2^{(i+1)}, \cdots, \mathbf{x}_{m-2}^{(i+1)}, \mathbf{x}_{m-1}^{(i+1)})$
3. Set $i = i + 1$ and go to step 2.

In order to compute large covariance matrices, HARVILLE (1999) suggested the Alg. 3.1 whose main steps will be given in the next section.

3.2.2 Computation of the covariance matrix based on Gibbs sampler

The Gibbs sampling method for estimating the elements of the inverse normal equation matrix is discussed in detail in HARVILLE (1999) and GUNDLICH et al. (2003), and was applied to matrix inversion of huge normal equation systems in the context of spherical harmonic analysis.

The first step of this procedure consists in a partitioning of the $m \times m$ normal equation matrix $\boldsymbol{N}$ and of its inverse $\boldsymbol{\Sigma}$ into 2×2 submatrices or blocks. This can be written as:

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \quad \text{with sizes} \quad \begin{bmatrix} r \times r & r \times (m-r) \\ (m-r) \times r & (m-r) \times (m-r) \end{bmatrix}. \tag{3.6}$$

Let the sample $\mathbf{x}$, which is to be generated, be normally distributed with expectation vector $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$, that is, $\mathfrak{X} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Then $\mathbf{x}$ is partitioned into subvectors $\mathbf{x}_1$ and $\mathbf{x}_2$, where $\mathbf{x}_1$ is of dimension $r \times 1$ and $\mathbf{x}_2$ dimension $m - r \times 1$. Then the distribution of $\mathfrak{X}_1$ conditional on $\mathfrak{X}_2 = \mathbf{x}_2$ is:

$$(\mathfrak{X}_1|\mathfrak{X}_2 = \mathbf{x}_2) \sim \mathcal{N}\left(\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\mathbf{x}_2, \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right). \tag{3.7}$$

Proof
The conditional density function $p(\mathbf{x}_1|\mathbf{x}_2)$ can be rewritten from the relation 2.8 as:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \frac{p(\mathbf{x})}{p(\mathbf{x}_2)}, \tag{3.8}$$

where $p(\mathbf{x}_2)$ is the marginal pdf of $\mathfrak{X}_2$. If $\mathfrak{X}_2 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{22})$, then its pdf is given by:

$$p(\mathbf{x}_2) = \frac{1}{(2\pi)^{(m-r)/2}(\det \boldsymbol{\Sigma}_{22})^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}_2^T\boldsymbol{\Sigma}_{22}^{-1}\mathbf{x}_2\right). \tag{3.9}$$

Substituting the results Eqs. (3.9) and (2.22) into Eq. (3.8) yields:

$$\begin{aligned} p(\mathbf{x}_1|\mathbf{x}_2) &= \frac{\frac{1}{(2\pi)^{m/2}(\det \boldsymbol{\Sigma})^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right)}{\frac{1}{(2\pi)^{(m-r)/2}(\det \boldsymbol{\Sigma}_{22})^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}_2^T{\boldsymbol{\Sigma}_{22}}^{-1}\mathbf{x}_2\right)} \\ &= \frac{1}{(2\pi)^{r/2}(\det \boldsymbol{\Sigma})^{1/2}(\det \boldsymbol{\Sigma}_{22})^{-1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \mathbf{x}_2^T\boldsymbol{\Sigma}_{22}^{-1}\mathbf{x}_2\right)\right\} \\ &= \frac{1}{(2\pi)^{r/2}(\det \boldsymbol{\Sigma})^{1/2}(\det \boldsymbol{\Sigma}_{22})^{-1/2}} \exp\{H\}. \end{aligned} \tag{3.10}$$

The exponent in Eq. (3.10) may be rewritten as:

$$H = -\frac{1}{2}\begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}\boldsymbol{\Sigma}^{-1}\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} - \mathbf{x}_2^T\boldsymbol{\Sigma}_{22}^{-1}\mathbf{x}_2. \tag{3.11}$$

One sets up the inverse of the block matrix $\mathbf{\Sigma}$ (see, e.g., MEYER 2000; p. 123):

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \mathbf{F} & -\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \\ -\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F} & \mathbf{\Sigma}_{22}^{-1} + \mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \end{bmatrix}, \tag{3.12}$$

where $\boldsymbol{F} = (\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21})^{-1}$ and $\mathbf{\Sigma}_{22}$ are both nonsingular[1]. Substituting this in Eq. 3.11 leads to:

$$\begin{aligned} H &= -\frac{1}{2}\begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{bmatrix}\begin{bmatrix} \mathbf{F} & -\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \\ -\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F} & \mathbf{\Sigma}_{22}^{-1} + \mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \end{bmatrix}\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} - \mathbf{x}_2^T\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2 \\ &= -\frac{1}{2}\left(\mathbf{x}_1^T\mathbf{F}\mathbf{x}_1 - \mathbf{x}_2^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F}\mathbf{x}_1 - \mathbf{x}_1^T\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2 + \mathbf{x}_2^T\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2 + \right. \\ &\quad \left. \mathbf{x}_2^T\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\mathbf{F}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2 - \mathbf{x}_2^T\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2\right) \\ &= -\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2\right)^T\mathbf{F}\left(\mathbf{x}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2\right), \end{aligned} \tag{3.13}$$

Further, the following condition holds:

$$\det\mathbf{\Sigma} = \det\begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix}. \tag{3.14}$$

Since $\mathbf{\Sigma}_{11}$ and $\mathbf{\Sigma}_{22}$ are square matrices, then holds (see, e.g., MEYER 2000; p. 475):

$$\begin{aligned} \det\mathbf{\Sigma} &= \det(\mathbf{\Sigma}_{22})\det\left(\mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\right) \qquad \text{if} \qquad \mathbf{\Sigma}_{22}^{-1} \qquad \text{exists} \\ &= \det\mathbf{\Sigma}_{22}\det\boldsymbol{F}^{-1}. \end{aligned} \tag{3.15}$$

From Eq. 3.15 it follows that:

$$\det\boldsymbol{F}^{-1} = \det\mathbf{\Sigma}(\det\mathbf{\Sigma}_{22})^{-1}. \tag{3.16}$$

Substituting the results Eqs. (3.16) and (3.13) into Eq. (3.10) yields:

$$p(\mathbf{x}_1|\mathbf{x}_2) = \frac{1}{(2\pi)^{r/2}(\det\mathbf{F})^{-1/2}}\exp\left\{\left(\mathbf{x}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2\right)^T\mathbf{F}\left(\mathbf{x}_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2\right)\right\}. \tag{3.17}$$

This leads to:

$$(\boldsymbol{\mathfrak{X}}_1|\boldsymbol{\mathfrak{X}}_2 = \mathbf{x_2}) \sim \mathcal{N}\left(\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2, \mathbf{F}^{-1}\right); \tag{3.18}$$

That is, the conditional distribution of $\mathbf{x}_1$ given $\mathbf{x}_2$ is $\mathcal{N}\left(\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{x}_2, \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\right)$ (cf. KOCH 1999, p. 121). □

With the matrix identities (cf. KOCH 1999, p. 33)

$$\boldsymbol{N}\mathbf{\Sigma} = \begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \tag{3.19}$$

we get the equations:

$$\begin{aligned} &\boldsymbol{N}_{11}\mathbf{\Sigma}_{11} + \boldsymbol{N}_{12}\Sigma_{21} = \boldsymbol{I} \qquad (1) \qquad \boldsymbol{N}_{11}\mathbf{\Sigma}_{12} + \boldsymbol{N}_{12}\Sigma_{22} = \mathbf{0} \qquad (2) \\ &\boldsymbol{N}_{21}\mathbf{\Sigma}_{11} + \boldsymbol{N}_{22}\Sigma_{21} = \mathbf{0} \qquad (3) \qquad \boldsymbol{N}_{21}\mathbf{\Sigma}_{12} + \boldsymbol{N}_{22}\Sigma_{22} = \boldsymbol{I} \qquad (4). \end{aligned} \tag{3.20}$$

From (1) in Eq. (3.20) we obtain

$$\boldsymbol{N}_{11}^{-1} = \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} \tag{3.21}$$

and from (2) we get

$$\boldsymbol{N}_{12} = -\boldsymbol{N}_{11}\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} \qquad \Longleftrightarrow \qquad \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1} = -\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12} \tag{3.22}$$

Substituting Eqs. (3.21) and (3.22) into Eq. (3.18) results in:

$$(\boldsymbol{\mathfrak{X}}_1|\boldsymbol{\mathfrak{X}}_2 = \mathbf{x}_2) \sim \mathcal{N}\left(-\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\mathbf{x}_2, \boldsymbol{N}_{11}^{-1}\right). \tag{3.23}$$

[1]The matrix $\boldsymbol{F}$ is called the *Schur complement* of $\mathbf{\Sigma}_{22}$

As a simplification, we will use for the following the notation $\mathbf{x}_1|\mathbf{x}_2$ instead of $(\boldsymbol{\mathfrak{X}}_1|\boldsymbol{\mathfrak{X}}_2 = \mathbf{x}_2)$.

The normal equation matrix $\boldsymbol{N}$ is divided into $r \times r$ blocks. The Alg. 3.1 is applied to generate the samples $\mathbf{x}^i$ according to Eq. (3.23) that is:

$$\mathbf{x}_l|\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_{l-1}^i, \mathbf{x}_{l+1}^{i-1}, \cdots, \mathbf{x}_r^{i-1} \sim \mathcal{N}\left(-\boldsymbol{N}_{ll}^{-1}\left(\sum_{j<l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i)} + \sum_{j>l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i-1)}\right), \boldsymbol{N}_{ll}^{-1}\right). \tag{3.24}$$

It is common practice to generate values $\mathbf{z}$ as realizations of the random variable $\boldsymbol{\mathfrak{Z}}$ from the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. In our case, however, the $n \times 1$ samples $\mathbf{x}^{(i)}$ must follow a multivariate normal distribution, i.e. $\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ must hold (see Sect. 2.3.4). The transformation then reads (cf. Koch 2000, p. 187):

$$\begin{aligned}\mathbf{x}_l^i &= \mathbf{R}_{ll}\mathbf{z}_l^i + \mu_l^i \\ &= \mathbf{R}_{ll}\mathbf{z}_l^i - \boldsymbol{N}_{ll}^{-1}\left(\sum_{j<l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i)} + \sum_{j>l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i-1)}\right),\end{aligned} \tag{3.25}$$

where $\boldsymbol{R}_{ll}$ denotes a upper triangular matrix, which is obtained from the Cholesky decomposition of the inverse of each block $\boldsymbol{N}_{ll}^{-1} = \boldsymbol{R}_{ll}^T\boldsymbol{R}_{ll}$.
Fig.3.1 demonstrates the computation steps of the Eq. 3.25. For example, suppose that the third sample of the i^{th} iteration ($\mathbf{x}_3^{(i)}$) shall be computed. At first, the product of the Cholesky-decomposed main diagonal block $\boldsymbol{R}_{33}$ with the corresponding part of the random vector $\mathbf{z}_3^{(i)}$ is computed. From this, the product of the inverted main diagonal block $\boldsymbol{N}_{33}$ with the bracketed expression $\left(\sum_{j<l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i)} + \sum_{j>l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i-1)}\right)$ (here combined within the vector $\mathbf{h}_3^{(i)}$) is subtracted. This expression is arises from of the products of the secondary diagonal blocks, which have the row index 3, with the corresponding parts of the vectors $\mathbf{x}$. From these values, however, only the parts $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$ are computed within the i-th iteration step. Therefore, for all the parts with an index larger than the current index, the samples of the previous iteration step ($\mathbf{x}_4^{(i)}$ and $\mathbf{x}_5^{(i)}$) are used.

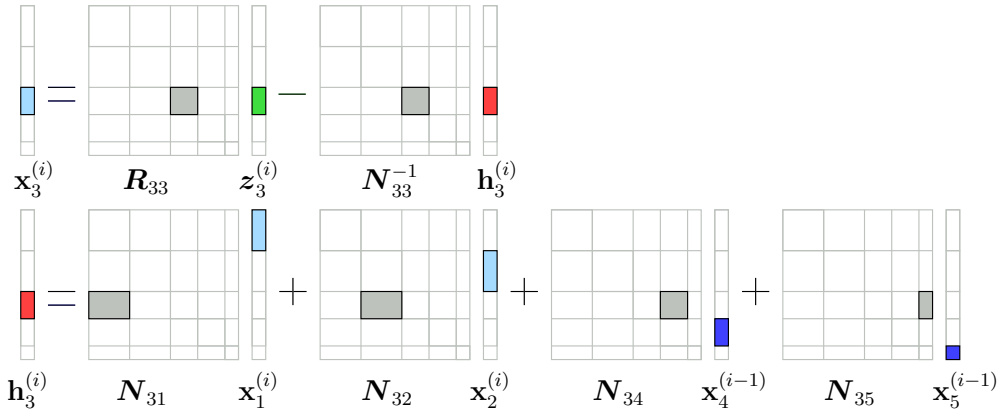


Fig. 3.1: Principle of the Gibbs-Sampler

Algorithm 3.1 reveals that the computation of the samples by means of Gibbs sampling is performed within an iterative process. To reduce the effect of the starting initial values $\mathbf{x}^{(0)}$, one generally discard the first S (*burn-in* period) values of the generated sequence. The *burn-in* period increases the efficiency of the estimation enormously. The length of this *burn-in period* is influenced by the starting values of the samples $\mathbf{x}^{(0)}$. A detailed discussion of suitable specifications for the *burn-in* period is found for instance in Johnson (1996).

Another problem that arises, is the dependence of the iterations in each generated sequence. As a remedy, once approximate convergence has been reached, is whether to *thin* the sequences by keeping every s^{th} simulation draw and discarding the rest (Gelman et al. 2004, pp. 287–305). Koch et al. (2004) proposed to use only every 5^{th} sample for an efficiently computing the covariance matrix. The following algorithm illustrates a generic implementation:

Algorithmus 3.2 (Version based on Gibbs sampler)

Purpose:	To estimate the $\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}$		
Input:	$\boldsymbol{N}$	...	normal equations
	$\boldsymbol{N}_{ij}$	...	blocks of the normal equations
	M	...	number of samples
	S	...	length of the burn-in period
	s	...	value to thin the samples
Output:	$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \boldsymbol{N}^{-1}$	...	VCM of parameters

1. Set arbitrary starting values for $\mathbf{x}$, for instance $\mathbf{x} = \mathbf{0}$
2. Divide the normal equation $\boldsymbol{N}$ into $r \times r$ blocks.
3. Reiterate the following computation for every $i = 1 \dots M$
   Generate samples $\mathbf{z}^{(i)}$ from $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
   `FOR l = 1, ..., r`
   - Invert the diagonal block: $\boldsymbol{N}_{ll}^{-1} = INV(\boldsymbol{N}_{ll})$.
   - Factorize $\boldsymbol{N}_{ll}^{-1} = \boldsymbol{R}_{ll}\boldsymbol{R}_{ll}^T$.
   - Transform random vector $\boldsymbol{z}_l^{(i)}$ to $\mathbf{x}_l^{(i)}$ by
     $\mathbf{x}_l^{(i)} = \boldsymbol{R}_{ll}\mathbf{z}_l^{(i)} - \boldsymbol{N}_{ll}^{-1}\left(\sum_{j<l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i)} + \sum_{j>l}\boldsymbol{N}_{lj}\mathbf{x}_j^{(i-1)}\right)$.

   `END l`
4. Discard the first S samples (the burn-in period)
5. Pick up every s^{th} sample from the remaining $M - S$ samples (i.e. thin the sequences).
6. Compute the VCM : $\boldsymbol{N}^{-1} = \frac{1}{M-S}\sum_{i=S+1}^{M}\mathbf{x}^{(i)}\mathbf{x}^{(i)T}$

Highly correlated unknown parameters lead to strong correlations between the generated samples $\mathbf{x}^{(i)}$. By clustering the samples of correlated parameters in the same subvectors $\mathbf{x}_l$, one can diminish the correlation (Harville 1999). The other reason for dividing the samples $\mathbf{x}^{(k)}$ into subvectors is to reduce the variance of the estimate $\boldsymbol{N}^{-1}$ in the last step of Alg. 3.2. This technique called *estimation by conditioning* and gives a more accurate estimate of the VCM of the unknown parameters (see Sect. 3.5.2).

It should be mentioned that this approach has two important characteristics. Firstly, it allows a very condensed representation of the variance/covariance information in terms of random samples (compression rate: 1:10 to 1:100), and secondly, it is optimally suited for parallel implementation. Koch et al. (2004) modified the algorithm 3.2 for parallel computation, which leads to an enormous decrease in computation time. Furthermore, this modified algorithm caused lower correlations between the generated samples.
The only shortcoming of the Gibbs sampler is the difficulty in adapting this algorithm to iterative solvers, which avoids the costly computation of the normal equations. Therefore, new generation method based on Monte Carlo Integration will be developed in the following section to overcome this lack.

3.3 Generation of random samples based on MC integration

In this chapter, special attention to least squares problems with the combination of heterogeneous types of observations is given. One part of the information is to be fully accessible in form of observation equations, whereas other parts are only provided in condensed form as normal equations. The goal is to estimate the variance information of the combined system. The approach will be developed step by step, beginning with a standard Gauss-Markov model with the full design matrix available. Then the approach is presented for the case that only the condensed information of the normal equations is available. Finally, combined systems are treated, where both types of information are assembled in an optimal way. It should be mentioned that the development of the following Monte Carlo algorithms is tailored for approximating large VCM in gravity field modeling, especially GOCE, and facilitate the related error propagation computations.

3.3.1 Gauss-Markov model, observation equations

As a starting point, the standard Gauss-Markov model (see 3.1) is considered as a linear transformation, defined by

$$\boldsymbol{\xi} = \boldsymbol{B}\boldsymbol{\lambda} \tag{3.26}$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$ denote *true values*, and the linear transformation matrix $\boldsymbol{B}$ (cf. Eq. (3.5)) is chosen as the inverse operator of $\boldsymbol{\lambda} = \boldsymbol{A}\boldsymbol{\xi}$ with The random counterparts of $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$ are denoted by the random vectors $\boldsymbol{\mathfrak{X}}$ and $\boldsymbol{\mathfrak{L}}$. The random vector

$$\boldsymbol{\mathfrak{L}} \sim \mathcal{N}(\boldsymbol{\lambda}, \boldsymbol{\Sigma}) \tag{3.27}$$

has the characteristic of an unbiased, normally distributed quantity with expectation $\boldsymbol{\lambda}$ and VCM $\boldsymbol{\Sigma}$. The characteristics of the linearly transformed quantity $\boldsymbol{\mathfrak{X}}$ can be derived by propagation of the expectation and the variances by:

$$\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\boldsymbol{B}\boldsymbol{\lambda}, \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T) \,. \tag{3.28}$$

Applying Eq. (3.5) to Eq. (3.28) yields:

$$\boldsymbol{\mathfrak{X}} \sim \mathcal{N}(\boldsymbol{\xi}, (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}) \,, \tag{3.29}$$

where $\boldsymbol{\xi}$ may also be viewed as the solution of the symmetric linear (*normal equation*) system

$$\left(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\right)\ \boldsymbol{\xi} = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\lambda} \,. \tag{3.30}$$

Direct computation of the VCM through

$$\boldsymbol{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \left(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\right)^{-1} = \boldsymbol{N}^{-1} \tag{3.31}$$

requires inverting the normal equation matrix $\boldsymbol{N}$. For large matrices this is a very costly task and the computational complexity increases with the power of three of the number of unknowns. Therefore, the application of Monte Carlo integration as a tool to circumvent the critical inversion step is described in the following. The first step is to generate sample vectors $\boldsymbol{s}_{\boldsymbol{\ell}}^{(1)}, \ldots, \boldsymbol{s}_{\boldsymbol{\ell}}^{(M)}$ with the same probability distribution as the random vector $\boldsymbol{\mathfrak{L}}$. These M samples are subsequently transformed through

$$\boldsymbol{s}_{\boldsymbol{x}}^{(i)} = \boldsymbol{B}\ \boldsymbol{s}_{\boldsymbol{\ell}}^{(i)} \,, \quad i = 1, \ldots, M \,, \tag{3.32}$$

into M vectors as realizations of $\boldsymbol{\mathfrak{X}}$. These transformed samples $\boldsymbol{s}_{\boldsymbol{x}}^{(i)}$ can then be used to estimate the VCM of $\boldsymbol{\mathfrak{X}}$ by applying Eq. (2.35). The transformation of the samples $\boldsymbol{s}_{\boldsymbol{\ell}}^{(i)}$ into the samples $\boldsymbol{s}_{\boldsymbol{x}}^{(i)}$ defined by Eq. (3.32) corresponds to the solution of Eq. (3.30), i.e. of

$$(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})\ \boldsymbol{s}_{\boldsymbol{x}}^{(i)} = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{s}_{\boldsymbol{\ell}}^{(i)}, \tag{3.33}$$

which can be accomplished by means of direct or iterative solution techniques (cf., e.g., Poder and Tschering 1973; Schuh 1984).

In order to simplify the computation, one can also work with centered quantities. This requires generating realizations $\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(1)}, \ldots, \boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(M)}$ from $\boldsymbol{\Delta\mathfrak{L}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and transforming them by

$$\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \boldsymbol{B}\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(i)}. \tag{3.34}$$

The transformed samples are distributed according to $\boldsymbol{\Delta\mathfrak{X}} \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1})$. Consequently, the VCM can be calculated by application of Eq. (2.35), yielding

$$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)T} \tag{3.35}$$

as the mean values of the sum of dyadic products of $\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)}$ which were generated by transformation in Eq. 3.34.

Proof
This can be proven immediately by variance propagation starting with the linear expression for $\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)}$ derived from (3.33)

$$\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)} \; . \tag{3.36}$$

Substituting this into (3.35) and reordering the summation yields:

$$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\left(\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)T}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1} \; . \tag{3.37}$$

Defining $\widehat{\boldsymbol{\Sigma}}$ by Eq. (2.35)

$$\widehat{\boldsymbol{\Sigma}} = \widehat{\Sigma}\{\Delta\boldsymbol{\ell}\} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)T} \tag{3.38}$$

one obtains

$$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}. \tag{3.39}$$

The central limit theorem (cf. Sect. 3.4) ensures that $\widehat{\boldsymbol{\Sigma}}$ converges to $\boldsymbol{\Sigma}$ if the number of samples M increases to infinity ($M \to \infty$), and therefore $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$ converges to $\Sigma\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}$. □

All the necessary steps for implementing this Monte Carlo estimation procedure are summarized in Alg. 3.3.

Algorithmus 3.3 (Version based on observation equations)

Purpose:	To estimate $\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}$		
Input:	$\boldsymbol{A}$	...	design matrix
	$\boldsymbol{\Sigma}$	...	VCM of observations
Output:	$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}$	...	VCM of parameters

1. Generate samples $\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}$, $i = 1, \ldots, M$ from $\Delta\boldsymbol{\mathfrak{L}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$.
2. Transform the sample vectors $\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}$ into $\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)}$ by solving the linear equation system $(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})\ \boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)} = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}$ for M right-hand sides.
3. Estimate the variance/covariance matrix by $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)}\boldsymbol{s}_{\Delta\boldsymbol{x}}^{(i)T}$.

Since samples are evaluated independently, Alg. 3.3 is ideally suited for parallel implementation (see section 3.6).
The crucial point of the whole procedure is how to generate the samples $\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}$ with the random characteristics of $\Delta\boldsymbol{\mathfrak{L}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ [2]. Assuming independency of the random quantities, the joint PDF may be factorized into univariate PDFs, and the VCM degenerates to a diagonal matrix. The realizations of the different components $\left(\boldsymbol{s}_{\Delta\boldsymbol{\ell}}^{(i)}\right)_j$ may therefore be independently drawn from the univariate normal distribution.
However, the more complex situation, where the observations are correlated and the VCM $\boldsymbol{\Sigma}$ is a dense system, must be considered too. For instance, the observations might stem from a pre-adjustment, or the measurement process might result in a time series with a colored noise characteristic. Since the VCM must, by definition, be set up as a symmetric and positive definite matrix, it may be decomposed into two triangular matrices $\boldsymbol{R}$ with $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$ by a rank-preserving Cholesky factorization. This procedure is implemented as follows:

[2] It should be mentioned that one of the great advantages of the MC approach is that it works with arbitrary PDFs. The choice of normal distribution for this study is justified by the asymptotic behavior of the normal distribution with respect to large numbers of samples.

Algorithmus 3.4

Purpose:	To generate samples $\boldsymbol{\Delta\mathfrak{L}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$		
Input:	$\boldsymbol{\Sigma}$	...	variance/covariance matrix
Output:	$\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(i)}$	...	realizations of $\boldsymbol{\Delta\mathfrak{L}}$

1. Factorize $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$.
2. Generate samples $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$, $i = 1, \ldots, M$, from $\boldsymbol{\mathfrak{E}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
3. Transform the random vector $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ to $\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(i)}$ by $\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(i)} = \boldsymbol{R}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$.

The resulting samples $\boldsymbol{s}_{\boldsymbol{\Delta\ell}}^{(i)}$ from auxiliary Alg. 3.4 belong to $\boldsymbol{\Delta\mathfrak{L}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. This can be immediately proven by variance propagation (see the proof of Alg. 3.3). Algorithms 3.3 and 3.4 summarize all the necessary steps to estimate the VCM of $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$ by Monte Carlo integration directly from the observation equations

$$\boldsymbol{A\mathfrak{X}} = \boldsymbol{\mathfrak{L}} + \boldsymbol{\mathcal{V}} \qquad \boldsymbol{\mathfrak{L}} \sim \boldsymbol{\mathcal{N}}(\boldsymbol{A\xi}, \boldsymbol{\Sigma}) \; . \tag{3.40}$$

with respect to the random vectors $\boldsymbol{\mathfrak{L}}$ of the observations, the residuals $\boldsymbol{\mathcal{V}}$ and the parameters $\boldsymbol{\mathfrak{X}}$. This is the empirical counterpart to $\boldsymbol{A\xi} = \boldsymbol{\lambda}$ with respect to the true values $\boldsymbol{\xi}$ and $\boldsymbol{\lambda}$. Assuming the Markov condition $E\{\boldsymbol{\mathcal{V}}\} = \mathbf{0}$ implies $E\{\boldsymbol{\mathfrak{L}}\} = \boldsymbol{\lambda}$ and $E\{\boldsymbol{\mathfrak{X}}\} = \boldsymbol{\xi}$, respectively. This straightforward algorithm may be optimized by introducing an additional transformation to fully decorrelate and homogenize the observation equations, followed by solving the simplified Gauss-Markov model

$$\bar{\boldsymbol{A}}\boldsymbol{x} = \bar{\boldsymbol{\ell}} + \bar{\boldsymbol{v}} \qquad \bar{\boldsymbol{\mathfrak{L}}} \sim \boldsymbol{\mathcal{N}}(\bar{\boldsymbol{A}}\boldsymbol{\xi}, \boldsymbol{I}) \; , \tag{3.41}$$

where $\boldsymbol{I}$ denotes the unit matrix. The next section focuses on this decorrelation process.

3.3.2 Gauss-Markov model, decorrelated observation equations

The decorrelation of the (fully correlated) observation equations, as defined by (3.40), is performed by applying a linear transformation, i.e.

$$\bar{\boldsymbol{\ell}} = \boldsymbol{F}\,\boldsymbol{\ell} \; , \tag{3.42}$$

on the observations where $\boldsymbol{F}$ represents a regular (i.e. invertible) matrix. The same transformation can be applied to the entire system of observation equations. This system is denoted by

$$\bar{\boldsymbol{A}}x = \bar{\boldsymbol{\ell}} + \bar{\boldsymbol{v}} \; , \tag{3.43}$$

with

$$\bar{\boldsymbol{A}} = \boldsymbol{F}\boldsymbol{A} \quad \text{and} \quad \bar{\boldsymbol{v}} = \boldsymbol{F}\boldsymbol{v} \; . \tag{3.44}$$

The VCM $\Sigma\{\bar{\boldsymbol{\mathfrak{L}}}\}$ of the filtered observations $\bar{\boldsymbol{\ell}}$ can be derived by variance propagation with

$$\Sigma\{\bar{\boldsymbol{\mathfrak{L}}}\} = \boldsymbol{F}\,\boldsymbol{\Sigma}\,\boldsymbol{F}^T \; . \tag{3.45}$$

If the Cholesky factor $\boldsymbol{R}$ of the VCM $\boldsymbol{\Sigma}$, defined by $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$, is used as a filter matrix with $\boldsymbol{F} = (\boldsymbol{R}^{-1})^T$, it can be seen immediately that

$$\begin{aligned} \Sigma\{\bar{\boldsymbol{\mathfrak{L}}}\} &= \boldsymbol{F}\boldsymbol{\Sigma}\boldsymbol{F}^T = \boldsymbol{F}\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{F}^T = \\ &= (\boldsymbol{R}^{-1})^T\boldsymbol{R}^T\boldsymbol{R}(\boldsymbol{R}^{-1}) = \boldsymbol{I}, \end{aligned} \tag{3.46}$$

thus the VCM $\Sigma\{\bar{\boldsymbol{\mathfrak{L}}}\}$ degenerates to the unit matrix $\boldsymbol{I}$. Therefore, the transformed, i.e. uncorrelated and homogenized observation equations are given by

$$\bar{\boldsymbol{A}}\boldsymbol{x} = \bar{\boldsymbol{\ell}} + \bar{\boldsymbol{v}}, \tag{3.47}$$

where $\bar{\boldsymbol{\ell}}$ are realizations of the random vector $\bar{\boldsymbol{\mathfrak{L}}} \sim \boldsymbol{\mathcal{N}}(\bar{\boldsymbol{A}}\boldsymbol{\xi}, \boldsymbol{I})$. Consequently, inserting this decorrelation step $\bar{\boldsymbol{A}} = (\boldsymbol{R}^{-1})^T\boldsymbol{A}$ into the Monte Carlo integration, Alg. 3.3 and 3.4 can be reformulated. The whole procedure is summarized in Alg. 3.5.

Algorithmus 3.5 (Version based on decorrelation)

Purpose:	To estimate $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$		
Input:	$\boldsymbol{A}$	...	design matrix
	$\boldsymbol{\Sigma}$	...	VCM of observations
Output:	$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}$	...	VCM of parameters

1. Decorrelation step: Transform the design matrix $\boldsymbol{A}$ into $\bar{\boldsymbol{A}}$.
2. Generate samples $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$, $i = 1, \ldots, M$ from $\boldsymbol{\Delta\mathfrak{E}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
3. Transform the random vectors $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ into $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$ by solving the linear system $(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}})\ \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \bar{\boldsymbol{A}}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ for M right-hand sides.
4. Estimate the VCM by $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)T}$.

The results of this algorithm can be proven by substitution and variance propagation (see proof of Alg. 3.3). The cost of the decorrelation step, characterized by the factorization step $\boldsymbol{\Sigma} = \boldsymbol{R}^T\boldsymbol{R}$ and the filter step $\bar{\boldsymbol{A}} = (\boldsymbol{R}^{-1})^T\boldsymbol{A}$, mainly depends on the structure of the VCM $\boldsymbol{\Sigma}$. In many applications, the VCM is defined as a diagonal matrix or a band matrix (such as the VCM for observations in geodetic networks, or the VCM of velocities derived from point observations). Then these steps are easy to compute. In reality, though, time series are often superposed by colored noise. Under the assumption of stationary noise and regularly distributed data the VCM is a Toplitz matrix. The computation with the Toeplitz matrix $\boldsymbol{\Sigma}$ degenerates to a convolution and can be performed efficiently in the frequency domain as well as in the time domain. In both cases, efficient tailored strategies can be constructed to perform the operation $(\boldsymbol{R}^{-1})^T\boldsymbol{A}$ for arbitrary matrices $\boldsymbol{A}$. This can be accomplished by applying FFT techniques, thus carrying out the discrete convolution in the frequency domain, or by implementing discrete linear ARMA (Auto Regressive Moving Average) filters, thus performing the computation in the time domain (Schuh 2003b).

The proposed algorithm has large potential for parallel implementation. The main steps, the decorrelation procedure as well as the solution process, can be split up horizontally or vertically into single threads and processed in parallel (cf. Schuh 2003a; Koch et al. 2004).

3.3.3 Gauss-Markov model, normal equations

In this section, the standard approach with direct application of this strategy to the normal equations will be discussed. In this case, the normal equations $\boldsymbol{N}\boldsymbol{x} = \boldsymbol{n}$ are given, and the VCM of the unknown parameters, defined by $\Sigma\{\boldsymbol{\mathfrak{X}}\} = \boldsymbol{N}^{-1}\sigma^2$ will be computed. For simplicity, we assume hereafter that $\sigma^2 = 1$, or that the variance factor is already included in the normal equation system. The focus is on the computation of the inverse $\boldsymbol{N}^{-1}$.

It is now intended to process large combined systems, where parts of the measurement information is condensed in the form of normal equations, but affects only relatively small groups of parameters. In fact, it may be assumed that the factorization of these systems with less than 10,000 to 20,000 parameters can be accomplished without much effort. Consequently, it appears to be rather inefficient to use the Monte Carlo integration under these circumstances. However, as it will be seen in the following section, this approach is nevertheless very useful when Monte Carlo integration is applied to a combined system. In such a system, large parts of the parameters are defined directly by observation equations, and normal equations yield significant additional information only for few parameters. Therefore, samples have to be generated also for smaller systems. Of course, this can be done by the application of the Gibbs sampler which was presented in Sect. 3.2, but also a very simple factorization approach is convenient.

Following this idea, the normal equations $\boldsymbol{N}$ are split up into two triangular matrices $\boldsymbol{S}$ by a Cholesky factorization $\boldsymbol{N} = \boldsymbol{S}^T\boldsymbol{S}$. The estimation of $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$ can be performed in various ways by Monte Carlo integration. In the following a first straightforward approach is given. First, samples $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ from $\boldsymbol{\mathfrak{E}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ are generated. These samples are transformed by $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)} = \boldsymbol{S}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ to obtain $\mathcal{N}(\boldsymbol{0}, \boldsymbol{N})$-distributed samples. The solution process $\boldsymbol{N}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$ yields samples $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$, which belongs to the distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{N}^{-1})$ and which can be used to estimate $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$ via Monte Carlo integration (cf. Alg. 3.6).

Algorithmus 3.6 (Version based on normal equations)

Purpose:	To estimate $\widehat{\Sigma}\{\mathfrak{X}\}$		
Input:	$\boldsymbol{N}$	...	normal equation
Output:	$\widehat{\Sigma}\{\mathfrak{X}\} = \boldsymbol{N}^{-1}$	...	VCM of parameters

1. Factorize $\boldsymbol{N} = \boldsymbol{S}^T\boldsymbol{S}$.
2. Generate samples $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$, $i = 1, \ldots, M$ from $\mathfrak{E} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
3. Transform the random vector $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ to $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$ by $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)} = \boldsymbol{S}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$.
4. Transform the random vector $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$ to $\boldsymbol{s}_{\boldsymbol{x}}^{(i)}$ by solving the linear system $\boldsymbol{N}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$.
5. Estimate the variance/covariance matrix by $\widehat{\Sigma}\{\mathfrak{X}\} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)T}$.

Comment: A step-by-step analysis of this Algorithm 3.6 immediately reveals the bottleneck as the computation of $\boldsymbol{N}^{-1}\boldsymbol{N}\boldsymbol{N}^{-1}$. Of course, the straightforward way of representing $\boldsymbol{N}^{-1}$ as $\boldsymbol{N}^{-1} = \boldsymbol{S}^{-1}(\boldsymbol{S}^T)^{-1}$ may be used to condense steps 3 and 4 of Alg. 3.6. This allows the direct computation of $\boldsymbol{s}_{\boldsymbol{x}}^{(i)}$ from $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ by solving the triangular system $\boldsymbol{S}\ s_{\boldsymbol{x}}^{(i)} = \boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ within a back substitution step. If only the VCM of a condensed normal equation system is of interest, this shortcut is convenient. However our goal is to combine different groups of data. With respect to this task Alg. 3.7 constitutes the basic tool to introduce condensed normal equations into a combined heterogeneous model.

3.3.4 Gauss-Markov model, combined heterogeneous systems

As mentioned before, it is intended to apply the Monte Carlo simulation strategy to estimate the VCM of the combined system defined by two groups of observations. The equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{\ell} + \boldsymbol{v} \qquad \mathfrak{L} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{\xi}, \boldsymbol{\Sigma}) \tag{3.48}$$

describe one part of the observations, for which it is desirable to avoid the explicit computation of the normal equations, and the system

$$\boldsymbol{N}\boldsymbol{x} = \boldsymbol{n} \tag{3.49}$$

contains the information of the second group of observations. The combination of these mutually uncorrelated systems results in the sum of normal equations (see e.g. Koch 1999, p. 177)

$$\left(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A} + \boldsymbol{N}\right)\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\ell} + \boldsymbol{n}. \tag{3.50}$$

The VCM of the combined system is defined by

$$\Sigma\{\mathfrak{X}\} = \left(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A} + \boldsymbol{N}\right)^{-1}. \tag{3.51}$$

To exploit this formula by a Monte Carlo approach, the same strategy as before can be followed: generate samples, transform these samples by solving the combined system, and estimate the VCM by an average process. As the single steps have already been prepared in Algs. 3.5 and 3.6, these algorithms simply need to be combined into one computational stream.

Algorithmus 3.7 (Version based on combined system)

Purpose:	To estimate $\widehat{\Sigma}\{\mathfrak{X}\}$		
Input:	$\boldsymbol{N}$	...	normal equation
	$\boldsymbol{A}$	...	design matrix
	$\boldsymbol{\Sigma}$	...	VCM of observations
Output:	$\widehat{\Sigma}\{\mathfrak{X}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A} + \boldsymbol{N})^{-1}$	...	VCM of parameters

1. Decorrelation step: Transform the design matrix $\boldsymbol{A}$ to $\bar{\boldsymbol{A}}$.
2. Generate samples $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$, $i = 1, \ldots, M$ from $\boldsymbol{\Delta\mathfrak{E}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
3. Factorize $\boldsymbol{N} = \boldsymbol{S}^T\boldsymbol{S}$.
4. Generate samples $\boldsymbol{s}_{\boldsymbol{g}}^{(i)}$, $i = 1, \ldots, M$ from $\boldsymbol{\mathfrak{G}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
5. Transform the random vector $\boldsymbol{s}_{\boldsymbol{g}}^{(i)}$ to $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$ by $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)} = \boldsymbol{S}^T\boldsymbol{s}_{\boldsymbol{g}}^{(i)}$.
6. Transform the random vectors $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ and $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$ to $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$ by solving the linear system
   $(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \boldsymbol{N})\ \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \bar{\boldsymbol{A}}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)} + \boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$
   for M right-hand sides.
7. Estimate the variance/covariance matrix by $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)T}$.

To prove this algorithm, use variance propagation taking into account that the $\boldsymbol{s}_{\boldsymbol{e}}^{(i)}$ are independent from the $\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}$.
The main computational work in Alg. 3.7 is done in step 6, where the samples $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$ are determined by solving the linear system

$$(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \boldsymbol{N})\ \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} = \bar{\boldsymbol{A}}^T\boldsymbol{s}_{\boldsymbol{e}}^{(i)} + \boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)} \tag{3.52}$$

for M different combined samples. Regarding the GOCE mission, these are very large systems with 100 millions of observations and about 60,000 unknowns, resulting from the extremely sensitive SGG observations. In contrast, the SST observations are collected in normal equations which are only sensitive to about 20,000 unknowns.
To take this into account, the normal equation system $\boldsymbol{N}\boldsymbol{x} = \boldsymbol{n}$ is split up into

$$\begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{0} \end{bmatrix}, \tag{3.53}$$

and the quantities $\boldsymbol{s}_{\boldsymbol{g}}^{(i)}$ are transformed by

$$\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)} = \begin{bmatrix} \left(\boldsymbol{s}_{\boldsymbol{\Delta n}}^{(i)}\right)_1 \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{11}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_{\boldsymbol{g}}^{(i)} \tag{3.54}$$

As far as the computational effort is concerned, it is easy to solely perform the Cholesky reduction of the small block $\boldsymbol{N}_{11}$ of the normal equation system. In the case of singular normal equations, a regularization by means of a rank preserving factorization (cf. Gill et al. 1981, p. 173) with a suitable choice of the null space has to be performed. It should be mentioned that the factorized matrix is only required to generate the samples, but not within the solution process, where the combined normal equations (Eq. 3.52) are involved. For the solution of this large system, parallel iterative procedures have been proven to be useful (cf. as an example the PCGMA algorithm in Schuh 1996).

3.4 Accuracy of the Monte Carlo integration

In Sect. 2.4, two types of integrals were defined to be solved by Monte Carlo integration. The first type (2.30) is related to the expectation, the second type (2.33) to the variance of a random vector. Both integrals may be estimated through Monte Carlo integration by averaging samples Eq. (2.32), or by averaging dyadic products of random samples Eq. (2.35). The fundamental theorem of Monte Carlo estimation (see, e.g., Leonard and Hsu 1999, p. 275) guarantees that the estimated value $\hat{E}\{f_j(\boldsymbol{\mathfrak{X}})\}$ for each individual function $f_j(\boldsymbol{\mathfrak{X}})$ converges to the true value $E\{f_j(\boldsymbol{\mathfrak{X}})\}$ almost surely, that is

$$P\left\{\lim_{M\to\infty}\hat{E}\{f_j(\boldsymbol{\mathfrak{X}})\} = E\{f_j(\boldsymbol{\mathfrak{X}})\}\right\} = 1\ , \tag{3.55}$$

if the number of samples M increases to infinity, i.e. if $M \to \infty$. If the number of samples is finite, an upper bound for the error $\boldsymbol{\varepsilon}_j$ with

$$\boldsymbol{\varepsilon}_j = \hat{E}\{f_j(\boldsymbol{\mathfrak{X}})\} - E\{f_j(\boldsymbol{\mathfrak{X}})\}, \tag{3.56}$$

which is defined as the difference between the estimated value $\hat{E}\{f_j(\mathfrak{X})\}$ and the true value $E\{f_j(\mathfrak{X})\}$, can be obtained by Chebychev's inequality (cf. KALOS and WHITLOCK 1986, p. 25)

$$P\left\{\varepsilon_j^2 \geq \frac{\sigma^2_{\hat{E}\{f_j(\mathfrak{X})\}}}{\alpha}\right\} \leq \alpha, \tag{3.57}$$

where $\sigma^2_{\hat{E}\{f_j(\mathfrak{X})\}}$ characterizes the variances of the estimated values $\hat{E}\{f_j(\mathfrak{X})\}$. The positive number α represents the error probability ($0 \leq \alpha < 1$).

Comment
This formulation is often denoted as the fundamental theorem of Monte Carlo integration. But we have to be very careful and distinguish between the variance of the estimated value $\hat{E}\{f_j(\mathfrak{X})\}$ denoted by $\sigma^2_{\hat{E}\{f_j(\mathfrak{X})\}}$ and the variance of the random function $f_j(\mathfrak{X})$, denoted by $\sigma^2_{f_j(\mathfrak{X})}$.

Recall that each component $f_j(\mathfrak{X})$ of the vector function $\boldsymbol{f}(\mathfrak{X})$ is sampled by $s^{(i)}_{f_j}$ with variance $\sigma^2_{f_j(\mathfrak{X})}$. The averaging process

$$\hat{E}\{f_j(\mathfrak{X})\} = \frac{1}{M}\sum_{i=1}^{M} s^{(i)}_{f_j}, \tag{3.58}$$

allows to estimate the function value with a higher accuracy, because the variance of the average can be determined by variance propagation with

$$\sigma^2_{\hat{E}\{f_j(\mathfrak{X})\}} = \frac{1}{M}\,\sigma^2_{f_j(\mathfrak{X})}\,. \tag{3.59}$$

Substituting the variance of the estimated value $\sigma^2_{\hat{E}\{f_j(\mathfrak{X})\}}$ by the variance of the samples $\sigma^2_{f_j(\mathfrak{X})}$ Chebyshev's inequality (Eq. 3.57) can be rewritten as

$$P\left\{|\varepsilon_j| \geq \sqrt{\frac{1}{\alpha}}\,\sqrt{\frac{1}{M}}\;\;\sigma_{f_j(\mathfrak{X})}\right\} \leq \alpha\,. \tag{3.60}$$

In order to increase the accuracy of the estimation process each of the two factors $M^{-\frac{1}{2}}$ and $\sigma_{f_j(\mathfrak{X})}$ defining the error bound can be improved. As a very simple attempt in doing so, the factor $M^{-\frac{1}{2}}$ could be decreased by increasing the number of samples M. However, it must be kept in mind that the square root decreases very slowly, or, in other words, the computational effort increases quadratically.
As a first impression, let the error probability be fixed at 1% ($\alpha = 0.01$) and evaluate this factor, resulting in $\alpha^{-\frac{1}{2}} = 10$. Here one can see immediately that Chebyshev's inequality is only a rough estimation. If normal distributed values for the samples $s^{(i)}_{f_j}$ are introduced, or if the central limit theorem for large numbers of samples is applied, the limit distribution for the estimated value $\hat{E}\{f_j(\mathfrak{X})\}$ is known as the normal distribution. Therefore, the factor $\alpha^{-\frac{1}{2}}$ may be replaced by the quantile values of standard normal distribution, which are $K^{\mathcal{N}(0,1)}_{\alpha/2}$ and $K^{\mathcal{N}(0,1)}_{1-\alpha/2}$, respectively. The corresponding two-sided confidence region is defined as

$$P\left\{K^{\mathcal{N}(0,1)}_{\alpha/2} \leq \frac{\varepsilon_j}{\sigma_{\hat{E}\{f_j(\mathfrak{X})\}}} \leq K^{\mathcal{N}(0,1)}_{1-\alpha/2}\right\} = 1-\alpha\,. \tag{3.61}$$

If $\sigma_{\hat{E}\{f_j(\mathfrak{X})\}}$ is substituted by Eq. (3.59) with the variance of the samples $\sigma_{f_j(\mathfrak{X})}$, then this results in

$$P\left\{|\varepsilon_j| \leq \sqrt{\frac{1}{M}} K^{\mathcal{N}(0,1)}_{1-\alpha/2}\;\sigma_{f_j(\mathfrak{X})}\right\} = 1-\alpha\,. \tag{3.62}$$

If the error level is again set to $\alpha = 0.01$ the quantile value $K^{\mathcal{N}(0,1)}_{1-\alpha/2}$ results in 2.57. Therefore, the application of the normal distribution corresponds to an enhancement of the estimate of assessment of accuracy by a factor of four, or a reduction of samples by a factor of 16 with respect to Chebyshev's inequality.
The third factor of the accuracy bound is determined by the variance of the samples. In the literature (e.g. KALOS and WHITLOCK 1986p. 92 and LIU 2001p. 26), different strategies are described to reduce this variance.

Before elaborating on these strategies, (cf. Sect. 3.5.2) the estimation of variance as the second kind of Monte Carlo integration is now discussed. As explained in Sect. 2.4, the variance is estimated by

$$\widehat{\Sigma}\{\mathfrak{X}\} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)} \boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)T} \ , \tag{3.63}$$

where $\boldsymbol{s}_{\boldsymbol{\Delta x}}$ characterizes a sample with expectation zero. An individual variance $\hat{\sigma}^2_{\mathfrak{X}_j}$ is then computed by

$$\hat{\sigma}^2_{\mathfrak{X}_j} = \left(\widehat{\Sigma}\{\mathfrak{X}\}\right)_{jj} = \frac{1}{M}\sum_{i=1}^{M} \left(\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\right)_j \left(\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\right)_j \ . \tag{3.64}$$

The error of this estimation, that is

$$\varepsilon_{\hat{\sigma}^2_j} = \hat{\sigma}^2_{\mathfrak{X}_j} - \sigma^2_{\mathfrak{X}_j}, \tag{3.65}$$

is bounded by Chebyshev's inequality

$$P\left\{ \varepsilon^2_{\hat{\sigma}^2_j} \geq \frac{\sigma^2_{\hat{\sigma}^2_{\mathfrak{X}_j}}}{\alpha} \right\} \leq \alpha \ . \tag{3.66}$$

To use this bound either the variance $\sigma^2_{\hat{\sigma}^2_{\mathfrak{X}_j}}$ of the estimated variance $\hat{\sigma}^2_{\mathfrak{X}_j}$ is determined, or the PDF of the estimated quantity is evaluated. With respect to Eq. (3.64) the computation is obviously based on the squared sum of independent samples. Another approach evaluating the uncertainty of the estimation of $\hat{\sigma}^2_{\mathfrak{X}_j}$ in Eq. (3.64) is to analyze the distribution of the squared sum of independent values. Scaling these random variables by the assumed variances $\sigma^2_{\mathfrak{X}_j}$ and supposing normal distributed values, the evaluated sum is χ^2-distributed with M degrees of freedom, denoted by χ^2_M. The two sided confidence region of $\hat{\sigma}^2_{\mathfrak{X}_j}$ is defined by

$$P\left\{ K^{\chi^2_M}_{\alpha/2} \leq M\frac{\hat{\sigma}^2_{\mathfrak{X}_j}}{\sigma^2_{\mathfrak{X}_j}} \leq K^{\chi^2_M}_{1-\alpha/2} \right\} = 1-\alpha \ , \tag{3.67}$$

where $K^{\chi^2_M}_{\alpha/2}$ and $K^{\chi^2_M}_{1-\alpha/2}$ denote the quantiles of the χ^2_M distribution with respect to the error probability $\alpha/2$ and $1-\alpha/2$, respectively. The reformulation of this confidence region yields

$$P\left\{ \frac{1}{M}\,\sigma^2_{\mathfrak{X}_j} K^{\chi^2_M}_{\alpha/2} \leq \hat{\sigma}^2_{\mathfrak{X}_j} \leq \frac{1}{M}\,\sigma^2_{\mathfrak{X}_j} K^{\chi^2_M}_{1-\alpha/2} \right\} = 1-\alpha \ . \tag{3.68}$$

Introducing the error of the estimation $\varepsilon_{\hat{\sigma}^2_j}$ from Eq. (3.65) the relation

$$P\left\{ \frac{1}{M}\,\sigma^2_{\mathfrak{X}_j} K^{\chi^2_M}_{\alpha/2} - \sigma^2_{\mathfrak{X}_j} \leq \varepsilon_{\hat{\sigma}^2_j} \leq \frac{1}{M}\,\sigma^2_{\mathfrak{X}_j} K^{\chi^2_M}_{1-\alpha/2} - \sigma^2_{\mathfrak{X}_j} \right\} = 1-\alpha \tag{3.69}$$

$$P\left\{ \left(\frac{1}{M}\,K^{\chi^2_M}_{\alpha/2} - 1\right)\sigma^2_{\mathfrak{X}_j} \leq \varepsilon_{\hat{\sigma}^2_j} \leq \left(\frac{1}{M}\,K^{\chi^2_M}_{1-\alpha/2} - 1\right)\sigma^2_{\mathfrak{X}_j} \right\} = 1-\alpha \ . \tag{3.70}$$

is obtained, which provides limits for the estimation error. To bring this result into agreement with the confidence region of the expectation in Eq. (3.62), the asymptotic behavior of the χ^2_M distribution is used by replacing it by a normal distribution with expectation M and variance $2M$ (Johnson and Kotz 1970a; p. 176). The quantile $K^{\chi^2_M}_{\beta}$ of the χ^2_M distribution can be approximated by

$$K^{\chi^2_M}_{\beta} \approx M + K^{\mathcal{N}(0,1)}_{\beta}\sqrt{2M} \tag{3.71}$$

using the quantile $K^{\mathcal{N}(0,1)}_{\beta}$ of the normal distribution. Numerical simulations strongly evidenced that the error is of the order $\mathcal{O}(\frac{1}{M})$. Therefore, this approximation is sufficient for the given application. Using this approximation, the confidence region for the error of the estimated variance is given by

$$P\left\{ |\varepsilon_{\hat{\sigma}^2_j}| \leq \sqrt{\frac{2}{M}}\; K^{\mathcal{N}(0,1)}_{1-\alpha/2}\; \sigma^2_{\mathfrak{X}_j} \right\} = 1-\alpha \ , \tag{3.72}$$

and consists of three terms again. The first term $(2/M)^{\frac{1}{2}}$ states that the accuracy improves with the number of samples, the second term, that the quantile value fixes the probability level, and the last term, that the variance $\sigma^2_{\mathfrak{x}_j}$ acts as scaling factor. In contrast to Eq. (3.62), the performance of the variance estimation is reduced by a factor of $\sqrt{2}$. It should be also mentioned that these error bounds are valid for the variances and not for the standard deviations. As a consequence of Eq. (3.72), the test value

$$T = \frac{\hat{\sigma}^2_{\mathfrak{x}_j} - \sigma^2_{\mathfrak{x}_j}}{\sigma^2_{\mathfrak{x}_j}} \sqrt{\frac{M}{2}} \;\sim\; \mathcal{N}(0,1)\;, \tag{3.73}$$

which reflects a relative error of the estimated variance, is standard normal distributed.
This discussion shows that the relative error $(\hat{\sigma}^2_{\mathfrak{x}_j} - \sigma^2_{\mathfrak{x}_j})/\sigma^2_{\mathfrak{x}_j}$ of the simulation depends primarily on the number of samples. However, it must be taken into account that, as the number of samples increases, the error bound decreases very slowly. Especially for large systems, one may run into severe time problems. Therefore, other strategies are necessary to reduce the error bound, and this can only be accomplished by reducing the variance of the Monte Carlo estimator. As far as relative errors are concerned, it is immediately observed that a multiplicative transformation (scaling) has no effect on the relative error. Thus, additive forms should be preferred.

3.5 Variance reduction of the generated samples

3.5.1 Variance reduction techniques

In Sect. 3.4 we have seen that the accuracy of the Monte Carlo Integration primarily depends on the amount of generated random samples (cf. Eq. 3.73). The computing time increases proportionally with the number of generated samples. The general aim of the variance reductions techniques is to reduce errors and to get short computing times at the same time, without increasing the number of random samples. There are various methods to achieve this aim, the most common ones are:

- Stratified sampling: In this technique, the interval is divided in subintervals, and the estimate (Eqs. 2.32 and 2.35) of the basic Monte Carlo integration is applied to each subinterval separately. The variance of this method might be considerably smaller than the variance of the basic Monte Carlo simulation.

- Control variates: Another way to reducing the variance of the the estimators (Eqs. 2.32 and 2.35) is offered by the control variates. Instead of considering only realizations of the PDF, one introduce an additional, simple function $g(x)$, with a known expectation μ_g:

  $$\tilde{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\} \;=\; \hat{E}\{\boldsymbol{f}(\mathbf{x})\} - a(\boldsymbol{g}(\mathbf{x}) - \boldsymbol{\mu}_g).$$

  The variance of $\tilde{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\}$ is less than the variance of the original response if $\hat{E}\{\boldsymbol{f}(\mathbf{x})\}$ and g are positively correlated.

- Antithetic variates: In the antithetic variates one introduce an additional estimator $\bar{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\}$, with the same PDF as $\hat{E}\{\boldsymbol{f}(\mathbf{x})\}$. The variance reads

  $$V\left\{\frac{\bar{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\} + \hat{E}\{\boldsymbol{f}(\mathbf{x})\}}{2}\right\} \;=\; \frac{V\{\hat{E}\{\boldsymbol{f}(\mathbf{x})\}\}}{4} + \frac{V\{\bar{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\}\}}{4} + \frac{C\{\bar{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\}, \hat{E}\{\boldsymbol{f}(\mathbf{x})\}\}}{2}.$$

  If $\bar{\hat{E}}\{\boldsymbol{f}(\mathbf{x})\}$ and $\hat{E}\{\boldsymbol{f}(\mathbf{x})\}$ are negatively correlated, the variance may be considerably smaller than the variance of the estimates (2.32) and (2.35), respectively.

- Importance sampling: Another technique commonly used for reducing variance in Monte Carlo methods is importance sampling. This method is different from the basic Monte Carlo method presented in Sect. 2.4. Instead of sampling from $p(\mathbf{x})$ one generate samples from another PDF $h(\mathbf{x})$, and computes the estimate of (2.32) and (2.35) using averages of $\boldsymbol{f}(\mathbf{x})p(\mathbf{x})/h(\mathbf{x})$ instead of $\boldsymbol{f}(\mathbf{x})$. That is:

  $$E\left\{\frac{\boldsymbol{f}(\mathfrak{X})}{h(\mathbf{x})}\right\} = \int_{\chi} \boldsymbol{f}(\mathbf{x}) \frac{p(\mathbf{x})}{h(\mathbf{x})} h(\mathbf{x}) d\mathbf{x}.$$

  The function $h(\mathbf{x})$ is called the importance sampling distribution function.

- Conditional expectations: Conditional expectations is a very well-known variance reduction technique. Suppose one have generated independent samples $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}$ from the PDF $p(\mathbf{x})$ using the basic Monte Carlo method presented in Sect. 2.4. The estimator in (2.32) can be used to evaluate the integral. Furthermore, suppose that $\mathbf{x}$ divided into two parts and that the conditional expectation $E\{\boldsymbol{f}(\mathbf{x})|\mathbf{x}_2\}$ can be evaluated analytically. The alternative estimator of (2.32) is:

$$\tilde{\boldsymbol{I}} = \frac{1}{M}\Big(E\left\{\boldsymbol{f}(\mathbf{x})|\mathbf{x}_2^{(1)}\right\} + \cdots + E\left\{f(\mathbf{x})|\mathbf{x}_2^{(M)}\right\}\Big). \tag{3.74}$$

  Both estimator (2.32 and 3.74) are unbiased. Moreover,

$$V\{\hat{\boldsymbol{I}}\} = \frac{V\{\boldsymbol{f}(\mathbf{x})\}}{M} \geq \frac{V\{E\{\boldsymbol{f}(\mathbf{x})|\mathbf{x}_2\}\}}{M} = V\{\tilde{\boldsymbol{I}}\}, \tag{3.75}$$

  where $\hat{\boldsymbol{I}}$ denotes the estimator (2.32).

A detailed description of the mentioned techniques as well as other techniques can be found in Rubinstein (1981; p. 121 – 142) and in Robert and Casella (1999; p. 80 – 119). In the next chapter we would like to reduce the variance of the estimate Eq. (3.63) by applying a blocking technique (see, e.g., Liu 2001), i. e. by clustering the samples $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$ of correlated unknown parameters. This technique allows the *estimation by conditioning* which reduces the variance of the estimate Eq. (3.63). For more details see Harville (1999) and Gundlich et al. (2003).

3.5.2 Stepwise estimation by conditioning

An efficient estimator can be found by conditioning on subvectors. Let

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{bmatrix} \tag{3.76}$$

define a 2×2 block-structured matrix $\boldsymbol{N}$, and its inverse, which corresponds to the VCM, be given by

$$\boldsymbol{\Sigma} = \boldsymbol{N}^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} . \tag{3.77}$$

Applying the inversion process on block structured matrices (cf. Meyer 2000, p. 123) yields the relations

$$\boldsymbol{\Sigma}_{11} = \boldsymbol{N}_{11}^{-1} + \boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22}\boldsymbol{N}_{21}\boldsymbol{N}_{11}^{-1} \tag{3.78}$$

and

$$\boldsymbol{\Sigma}_{12} = -\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22} . \tag{3.79}$$

The main idea of the strategy of *estimation by conditioning* is that the inverse of $\boldsymbol{N}_{11}$ can be computed in a strict way, while the second term in Eq. (3.78) can be estimated by Monte Carlo integration.
The second diagonal block in the inverse matrix $\boldsymbol{\Sigma}_{22}$ may be estimated through Monte Carlo integration by (cf. Alg. 3.7)

$$\widehat{\boldsymbol{\Sigma}}_{22} = \frac{1}{M}\sum_{i=1}^{M} \left(\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\right)_2 \left(\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\right)_2^T, \tag{3.80}$$

where $\left(\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}\right)_2$ denotes the second (remaining) part of the samples $\boldsymbol{s}_{\boldsymbol{\Delta x}}^{(i)}$. A look at Eq. (3.78) reveals that each summand of is positive definite, resulting in the inequality

$$\boldsymbol{t}^T\left(\boldsymbol{\Sigma}_{11} - \boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22}\boldsymbol{N}_{21}\boldsymbol{N}_{11}^{-1}\right)\boldsymbol{t} \geq 0, \tag{3.81}$$

for arbitrary vectors $\boldsymbol{t}$. This states that each diagonal element of the matrix $\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22}\boldsymbol{N}_{21}\boldsymbol{N}_{11}^{-1}$ is smaller than or equal to the diagonal elements in $\boldsymbol{\Sigma}_{11}$, that is,

$$(\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22}\boldsymbol{N}_{21}\boldsymbol{N}_{11}^{-1})_{ii} \leq (\boldsymbol{\Sigma}_{11})_{ii}. \tag{3.82}$$

We see from Eq. (3.78) that the first summand $\boldsymbol{N}_{11}^{-1}$ is known (deterministic) so that only the second term has to be treated by the MC approach. As stated in Eq. (3.82) this term cannot be larger than $(\boldsymbol{\Sigma}_{11})_{ii}$, but corresponds to the third term $\sigma_{f_j(\boldsymbol{\mathfrak{X}})}$ in Eq. (3.60), which defines the accuracy of the MC estimator. Eq. (3.72) states that the error of the MC estimator is directly proportional to the variance $\sigma^2_{\boldsymbol{\mathfrak{X}}_j}$, which has to be estimated, or in other words, the relative accuracy of the MC estimation is constant. Now, if only the value $(\boldsymbol{N}_{11}^{-1}\boldsymbol{N}_{12}\boldsymbol{\Sigma}_{22}\boldsymbol{N}_{21}\boldsymbol{N}_{11}^{-1})_{ii}$ has to be estimated by the MC approach, and main parts of the variance are defined by the known term $\boldsymbol{N}_{11}^{-1}$, then the accuracy of the entire estimation process can be increased. Especially, the regular distribution of satellite observations produces diagonal block-dominant matrices (cf. SCHUH 1996). The block $\boldsymbol{N}_{11}^{-1}$ contains main parts of $\boldsymbol{\Sigma}_{11}$, and the estimation by conditioning substantially increases the accuracy (see the test simulations in Sect. 6.1). Instructions for implementing this procedure for an arbitrary $p \times p$ block system are provided in Alg. 3.8.

Algorithmus 3.8 (Version based on estimation by conditioning)

Purpose:	To estimate $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$		
Input:	$\boldsymbol{N}_{ij}$	...	blocks of the normal equations
	$(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}})_k$	...	subvectors of the samples
Output:	$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = \boldsymbol{N}^{-1}$	...	VCM of parameters

For $k = 1, \cdots, p$

1. Invert the diagonal block $\boldsymbol{N}_{kk}$
   $\boldsymbol{N}_{kk}^{-1} = INV(\boldsymbol{N}_{kk})$.
2. Transform random vector $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}}$ to $\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k$ by
   $\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k = -\boldsymbol{N}_{kk}^{-1}\Big(\sum_{\binom{j=1}{j\neq k}}^{p} \boldsymbol{N}_{kj}\ (\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}})_j\Big)$.
3. Compute the VCM of the diagonal block
   $\left(\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}\right)_{kk} = \boldsymbol{N}_{kk}^{-1} + \frac{1}{M}\sum_{i=1}^{M}\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k^T$.
4. Compute the VCM of the blocks in the row
   $\left(\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}\right)_{kj} = \frac{1}{M}\sum_{i=1}^{M}\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}})_j^T$
   $j = k{+}1, \ldots, p.$

END k

In Alg. 3.8, the normal equation matrix is subdivided into $p \times p$ blocks in such a way that all blocks on the diagonal are squared and regular. If the matrix N is symmetric and positive definite, the inverses of these diagonal blocks exist. Now, let k refer to a particular diagonal block and apply Eqs. (3.78) and (3.79), respectively. The index 1 in Eq. (3.78) is equivalent to k, and the index 2 corresponds to the remaining subset of indices $j = 1, \ldots, p$ with $j \neq k$.
This estimation by conditioning can be used to speed up the convergence behavior of all Monte Carlo integrations. This method does not work only in connection with normal equations (Alg. 3.6), but can be adapted to observation equations. Of course, in this case, the diagonal blocks of the normal equations need to be computed. This is often done automatically to find an appropriate preconditioner for the iterative procedure. Thus, this computation is not critical. Special attention is necessary regarding step 2 of Alg. 3.8, where the samples $\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k$ are generated. Here, all the off-diagonal blocks $\boldsymbol{N}_{kj}$ of the normal equations are involved. Due to the huge computational effort necessitated by this step, it is recommended instead that the off-diagonal blocks are evaluated direct from the solution of the system $\boldsymbol{N\Delta x} = \boldsymbol{n}$ by

$$\sum_{\binom{j=1}{j\neq k}}^{p} \boldsymbol{N}_{kj}\ (\boldsymbol{\Delta}x)_j = (\boldsymbol{n})_k - \boldsymbol{N}_{kk}(\boldsymbol{\Delta}x)_k\ , \tag{3.83}$$

where $(\boldsymbol{\Delta}x)_j$ denotes the j^{th} group of components in the vector $\boldsymbol{\Delta}x$. This transformation can be used in all algorithms. Therefore, the estimation by conditioning may be integrated into Alg. 3.6. Additional computations are required to compute

$$\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta \bar{x}}}\right)_k = (s^{(i)}_{\boldsymbol{\Delta x}})_k - \boldsymbol{N}_{kk}^{-1}(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}})_k \tag{3.84}$$

and to use the last steps of Alg. 3.8 in order to increase the accuracy of the estimation. With the same strategy, Alg. 3.5 can also be modified by computing the samples $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}$ with

$$\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}\right)_k = (s^{(i)}_{\boldsymbol{\Delta x}})_k - \boldsymbol{N}^{-1}_{kk}\bar{\boldsymbol{A}}_{\boldsymbol{k}}{}^T(\boldsymbol{s}^{(i)}_{\boldsymbol{e}})_k, \tag{3.85}$$

where $\bar{\boldsymbol{A}}_{\boldsymbol{k}}$ represents the design matrix of the k^{th} group of observations and $(\boldsymbol{s}^{(i)}_{\boldsymbol{e}})_k$ the samples of the same group of observations. In this paper, combined adjustment constitutes the main goal. Therefore, Alg. 3.7 is rewritten in combination with the estimation by conditioning. This results in Alg. 3.9.

Algorithmus 3.9 (Version based on estimation by conditioning)

Purpose:	To estimate $\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\}$ from combined system		
Input:	$\boldsymbol{N}$	...	normal equation
	$\boldsymbol{N}_{\oplus}$	...	block-diagonal preconditioner of $(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \boldsymbol{N})$
	$\boldsymbol{A}$	...	design matrix
	$\boldsymbol{\Sigma}$	...	VCM of observations
Output:	$\widehat{\Sigma}\{\boldsymbol{\mathfrak{X}}\} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A} + \boldsymbol{N})^{-1}$	...	VCM of parameters

1. Decorrelation step: Transform the design matrix $\boldsymbol{A}$ to $\bar{\boldsymbol{A}}$.
2. Generate samples $\boldsymbol{s}^{(i)}_{\boldsymbol{e}}$, $i = 1, \ldots, M$ from $\boldsymbol{\Delta\mathfrak{E}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
3. Factorize $\boldsymbol{N} = \boldsymbol{S}^T\boldsymbol{S}$.
4. Generate samples $\boldsymbol{s}^{(i)}_{\boldsymbol{g}}$, $i = 1, \ldots, M$ from $\boldsymbol{\mathfrak{G}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.
5. Transform the random vector $\boldsymbol{s}^{(i)}_{\boldsymbol{g}}$ to $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}}$ by
   $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}} = \boldsymbol{S}^T\boldsymbol{s}^{(i)}_{\boldsymbol{g}}$.
6. Transform the random vectors $\boldsymbol{s}^{(i)}_{\boldsymbol{e}}$ and $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}}$ to $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}}$ by solving the linear system
   $(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \boldsymbol{N})\ \boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}} = \bar{\boldsymbol{A}}^T\boldsymbol{s}^{(i)}_{\boldsymbol{e}} + \boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}}$
   for M right-hand sides.
7. Transform random vector $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}}$ to $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}$
   $k = 1, \ldots, p$
   $\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}\right)_k = (s^{(i)}_{\boldsymbol{\Delta x}})_k - (\boldsymbol{N}_{\oplus kk})^{-1}\Big(\bar{\boldsymbol{A}}_{\boldsymbol{k}}{}^T(\boldsymbol{s}^{(i)}_{\boldsymbol{e}})_k + (\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}})_k\Big)$.
8. Compute the VCM of the diagonal block,
   $k = 1, \ldots, p$
   $\left(\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}\right)_{kk} = \boldsymbol{N}^{-1}_{kk} + \frac{1}{M}\sum_{i=1}^{M}\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}\right)_k\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}\right)^T_k$
9. Compute the VCM of the not diagonal blocks,
   $k = 1, \ldots, p,\ j = k{+}1, \ldots, p$
   $\left(\widehat{\boldsymbol{\Sigma}}\{\boldsymbol{\mathfrak{X}}\}\right)_{kj} = \frac{1}{M}\sum_{i=1}^{M}\left(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta}\bar{\boldsymbol{x}}}\right)_k(\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta x}})^T_j$.

Algorithm 3.9 can be easily modified if, for example, only normal equations or observation equations $\boldsymbol{A}$ are available. In the case that only observation equations are available, all steps for computing $\boldsymbol{S}$, $\boldsymbol{s}^{(i)}_{\boldsymbol{g}}$ and $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}}$, and for adding the terms $\boldsymbol{s}^{(i)}_{\boldsymbol{\Delta n}}$ and $\boldsymbol{N}$ can be neglected. The same cancellation strategy works for a pure normal equation system. In addition, it should be mentioned that Algorithm 3.9 can be adapted to process more than one normal or observation equation system. Thus, Alg. 3.9 is flexible and efficient since it is readily tailored to solve large equation systems.

3.6 Efficiency estimations of the Monte Carlo variance/covariance matrix algorithms

In this section the estimations for the mathematical expectation of the time, serial efficiency and parallel speedup will be presented. All three parameters define the quality of the parallel algorithms.

In order to estimate how the Monte Carlo algorithms depend on different computer architectures, we have considered two models:

a) A serial model with time T_1 required to complete the operations

b) A multiprocessor configuration consisting of p processors (cluster). Every processor of the cluster performs its own instructions on the data in its own memory.

3.6.1 Serial efficiency

Before we analyze how the algorithm 3.9 can be parallelized, we study its efficiency in order to be sure that the processor speed is exploited optimally. This is only possible, if we attribute most operations to matrix multiplications. These are core operations in optimized libraries for linear algebra (ATLAS 2007). When we use these libraries, we have to make sure that the participating matrices exceed a critical size, for instance 100×100, because only then the optimization begin to have impact. As we can see, the algorithm 3.9 consists of the main part of matrix multiplications. The size of the individual matrices can be selected in such a way that it exceeds the minimum dimension of 100×100.

3.6.2 Parallel speedup

With the inherent parallelism of the Monte Carlo methods we have got the possibility to calculate each realization (sample) of the random variable on a different processor. There is no need for communication between the processors during the time of calculating the realizations. The only need for communication occurs at the end when the averaged value is to be calculated.

The *speedup* S, also called *parallel speedup*, is defined as the running time of execution of the program on one processor divided by the running time T_p of the parallel execution of the program on p-Processors (Gropp et al. 1994).

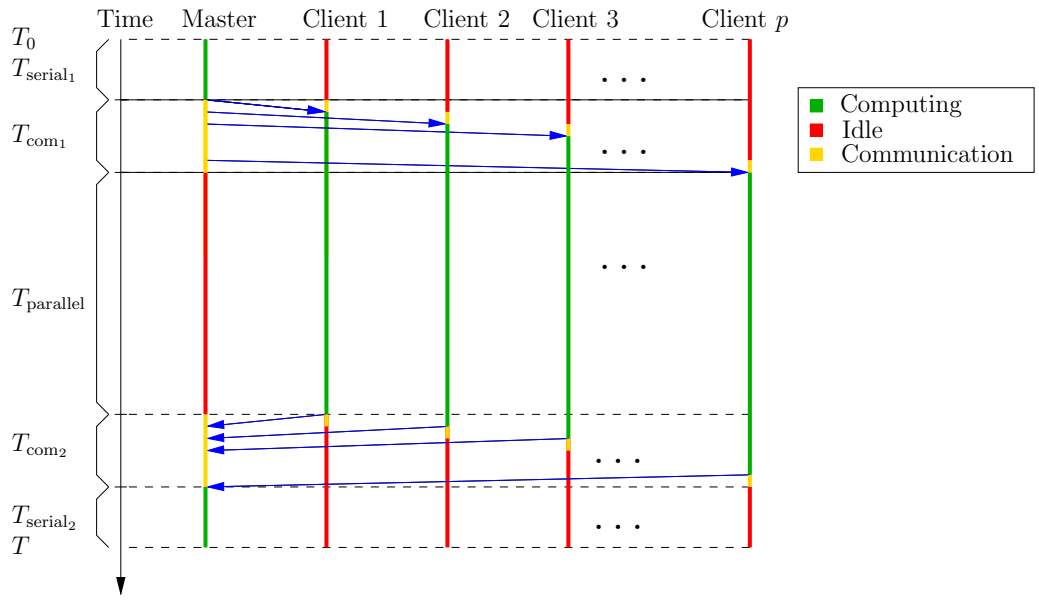


Fig. 3.2: Principle to parallelize Monte Carlo Variance/Covariance Matrix (MCVCM) algorithms

The computing time of a parallel program can be divided into parallelizable parts T_{parallel} and unparallelizable parts T_{seriell}. Furthermore the communication time T_{com} between the master and the clients has to be incorporated (cf. Fig. 3.2). Then *Amdahl's law* (Dongarra et al. 1990, p. 57) can be used to give a maximum speedup factor

$$S = \frac{T_1}{T_p} = \frac{T_1}{T_{\text{com}} + T_{\text{serial}} + \frac{T_{\text{parallel}}}{p}}. \tag{3.86}$$

Test simulations (see Sect. 6.1) were serial executed. They lead us to the following assumptions: The serial part of the computation requires approximately 10% of the total processing time while the parallelizable parts

of the program need about 90%. One has to take into account that the communication time has to be added to the excecution time T_1, if the program is parallelized. A pessimistic estimate for the communication time is 5% of the excecution time. This yields with Eq. (3.86) a maximum possible *speedup* S as:

$$S = \frac{100\%}{5\% + 10\% + 0\%} = 6.7. \tag{3.87}$$

The expected speedup for 1 to 100 processors, under the assumption of a constant communication time of $T_{\text{com}} = 5\%$, is depicted in Fig. 3.3. With these assumptions the best speedup factor of 5.8 is achieved, by using about 40 processors.

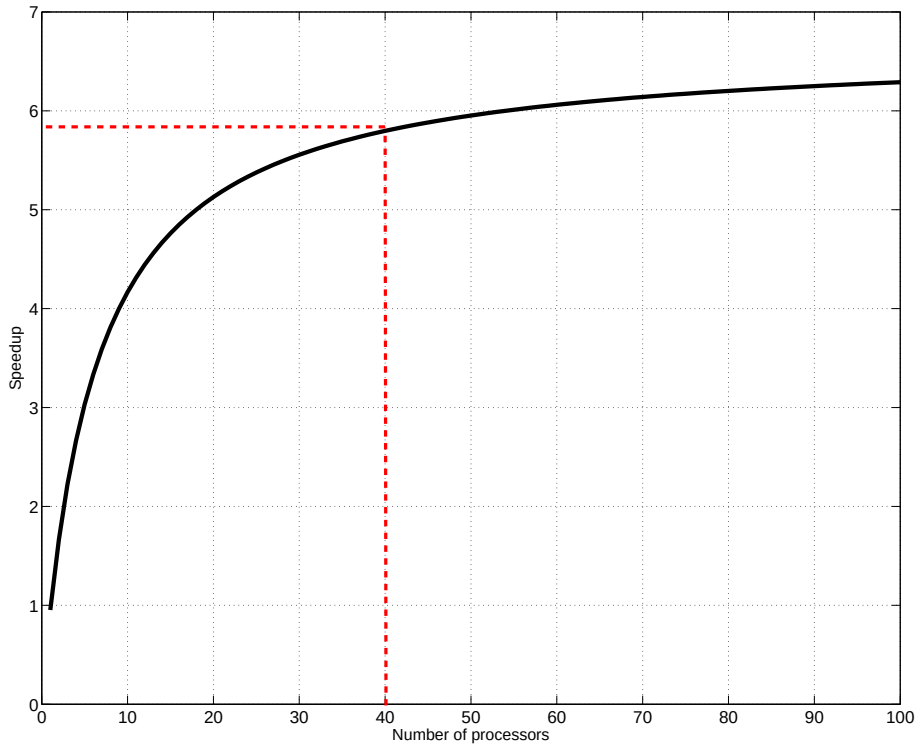


Fig. 3.3: Test simulation of the parallel speedup

The MCVCM algorithms can be easily applied in parallel computing. In fact, that was the motivation for developing the Monte Carlo methods for computing and propagating large covariance matrices.

4 Monte Carlo method for estimation of the variance components

4.1 The problem

One of the most important goals of current satellite missions is the reliable and accurate estimation of the Earth's gravity model. This model is based on various observables, which are strongly heterogeneous with respect to the type of sensor used and measurement accuracy. For instance, if a global modeling of the Earth's gravity field is desired, then data of different satellites (CHAMP, GRACE, GOCE) and ground stations, altimetry data, multi-arc data, etc. are analyzed. Some of the challenges to be taken into account when such heterogeneous data sources are combined into one model are:

- The measurement principles and sensors are different, e.g. satellite observations (SGG and SST), terrestrial and airborne gravimetry.
- The apriori accuracies of the various types of observables (if known) differ considerably.
- Even the variances of observations for one sensor type might be different, e.g. for GPS orbits (see Mayer-Gürr et al. (2005)).
- The normal equation system is often ill-conditioned.

In order to find appropriate weighting factors for the different data sets, on the one hand, and to regularize the inverse problem by determination of the regularization parameter on the other hand, we apply in this chapter the computational algorithm for VCE, which is presented in Koch and Kusche (2002) and Kusche (2003). In order to find a tailored VCE-algorithm, which can be integrated into PCGMA, this algorithm will be modified. How the modified algorithm is incorporated into the PCGMA algorithm, is shown later on in Sect. 5.4.2.

In Sect. 4.2 the general linear model with unknown variance components is presented. In this section, the unknown parameters and the unknown variance factors were computed rigorously by means of Alg. 4.1. The expensive repeated computation of the trace will be substituted in Sect. 4.3 by applying Monte Carlo methods (Alg. 4.2). Finally, the more general case of combining observation with normal equations will be outlined in Alg. 4.3.

4.2 Estimation of variance components

If multiple data types are given for an estimation of the parameters $\boldsymbol{x}$ and these observation groups are disjunctive, then the observation equations in the linear model (3.48) with a only unknown variance factor σ_0^2 for all the observations is extended to a model in which variance components are introduced:

$$\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{l}_i + \boldsymbol{v}_i \qquad \text{with} \qquad \Sigma\{\boldsymbol{\mathfrak{L}}\} = \sum_{i=1}^{o} \sigma_i^2 \boldsymbol{Q}_i \qquad \text{and} \qquad i \in \{1,\ldots,o\}\,, \tag{4.1}$$

where $\boldsymbol{Q}_i$ are (semi-)positive definite cofactor matrices, and σ_i^2 with $i \in \{1,\cdots,o\}$ are the unknown variance components to be estimated, in a joint estimation procedure together with the unknown parameters $\boldsymbol{x}$. If the model 4.1 is referred to gravity filed modeling, it is a almost always a series expansion of the gravity potential up to a maximal degree/order of spherical harmonic coefficients. In many cases, the normal equation system of the model 4.1 (e.g. for GOCE) tend to be ill-posed. Then the model 4.1 is extended to:

$$\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_o \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{l}_1 + \boldsymbol{v}_1 \\ \boldsymbol{l}_2 + \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{l}_o + \boldsymbol{v}_o \\ \boldsymbol{\mu} + \boldsymbol{v}_\mu \end{bmatrix}, \quad \text{with} \quad \Sigma\{\boldsymbol{\mathfrak{L}}, \boldsymbol{\mu}\} = \sum_{i=1}^{o} \sigma_i^2 \boldsymbol{Q}_i \; + \; \sigma_\mu^2 \boldsymbol{Q}_\mu. \tag{4.2}$$

$\boldsymbol{\mu}$ represents an $m \times 1$ vector of prior information for the unknown parameters, σ_μ^2 the corresponding variance factor which may be interpreted as regularization parameters, and $\boldsymbol{Q}_\mu$ an $m \times m$ prior positive definite matrix of the parameters. The observations $\boldsymbol{l}_i$ and the prior information $\boldsymbol{\mu}$ of the unknown parameters $\boldsymbol{x}$ in Eq. (4.2) may be assumed as stochastically independent quantities (see Koch and Kusche (2002)). The matrix $\Sigma\{\boldsymbol{\mathfrak{L}}, \boldsymbol{\mu}\}$ in Eq. (4.2) is generally block diagonal, and can be rewritten as follows:

$$\Sigma\{\mathfrak{L}, \boldsymbol{\mu}\} = \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 \boldsymbol{P}_1^{-1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 \boldsymbol{P}_2^{-1} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \sigma_o^2 \boldsymbol{P}_o^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \sigma_\mu^2 \boldsymbol{P}_\mu^{-1} \end{bmatrix}, \tag{4.3}$$

where $\boldsymbol{P}_i$ are the weight matrices of the corresponding types of observations and $\boldsymbol{P}_\mu$ is the weight matrix of the prior information.

The estimation of variance components in Eq. (4.2) has had a long history in geodesy and has been developed for a variety of applications. The most important estimation methods can be summarized as follows:

- In Rao and Kleffe (1988), the minimum norm quadratic unbiased estimation (MINQUE) method is derived. Besides that it is a quadratic and unbiased estimation, it fulfills the demand for minimal norm.
- Another method is the best invariant quadratic unbiased estimation (BIQUE) presented by Koch (1999) and Crocetto et al. (2000). Under the assumption that the observations are normally distributed, BIQUE and MINQUE are identical (Grafarend 1978).
- Variance components may also be estimated by the maximum likelihod method as explained in Koch (1986). An iterated maximum likelihood (IML) procedure for the Gauss-Markov model has been derived for normally distributed observations.
- A Bayesian estimation as well as Bayesian confidence intervals for variance components were suggested in Koch (1987) and Koch (2000)

The system of linear equations for the estimation of variance components can be given as follows (see, e.g., Koch 2000, p. 141):

$$\underbrace{\begin{bmatrix} r_1 & 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & r_o & 0 \\ 0 & 0 & \dots & 0 & r_\mu \end{bmatrix}}_{\boldsymbol{S}} \underbrace{\begin{bmatrix} \hat{\sigma}_1^2 \\ \vdots \\ \hat{\sigma}_o^2 \\ \hat{\sigma}_\mu^2 \end{bmatrix}}_{\hat{\boldsymbol{\sigma}}} = \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_o \\ q_\mu \end{bmatrix}}_{\boldsymbol{q}}, \tag{4.4}$$

where the variance components are combined into $\hat{\boldsymbol{\sigma}} = (\hat{\sigma}_1^2, \hat{\sigma}_1^2, \cdots, \hat{\sigma}_o^2, \hat{\sigma}_\mu^2)$ and coefficient matrix $\boldsymbol{S}$ and the vector $\boldsymbol{q}$ are known, given a set of start values of $\hat{\boldsymbol{\sigma}}$. Since the observations are assumed to be normally distributed and because of the variance components model 4.4 have a block diagonal, the methods listed above all lead to the same estimate (Xu et al. 2006). The variance components in 4.4 are iteratively solved, then the well-known alternative non-negative estimate of the variance components at the k^{th} iteration read as follows (see, e.g., Koch 2000, p. 146):

$$\hat{\sigma_i}^{2(k+1)} = \frac{q_i}{r_i} = \frac{\boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i}{r_i} \quad \text{with} \quad i \in \{1, \dots, o\} \qquad \text{and} \qquad \hat{\sigma}_\mu^{2(k+1)} = \frac{q_\mu}{r_\mu} = \frac{\boldsymbol{v}_\mu^T \boldsymbol{P}_\mu \boldsymbol{v}_\mu}{r_\mu}, \tag{4.5}$$

where

$$\boldsymbol{v}_i = \boldsymbol{A}_i \hat{\boldsymbol{x}} - \boldsymbol{l}_i \qquad \text{and} \qquad \boldsymbol{v}_\mu = \hat{\boldsymbol{x}} - \boldsymbol{\mu} \tag{4.6}$$

is the vector of residuals of the i^{th} data set and the vector of the regularization set, respectively. $\hat{\boldsymbol{x}}$ in Eq. (4.6) is the desired estimates for the unknown parameters of the normal equations

$$\underbrace{\left(\sum_{i=1}^{o} \frac{1}{\sigma_i^2} \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i + \frac{1}{\sigma_\mu^2} \boldsymbol{P}_\mu \right)}_{\boldsymbol{N}} \hat{\boldsymbol{x}} = \sum_{i=1}^{o} \frac{1}{\sigma_i^2} \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{l}_i + \frac{1}{\sigma_\mu^2} \boldsymbol{P}_\mu \boldsymbol{\mu} \tag{4.7}$$

of the GMM 4.2.

The partial redundancy of the i^{th} group r_i as well as the partial redundancy of regularization part group r_μ in eq. (4.5) are given as:

$$r_i = \operatorname{tr}\left(\boldsymbol{W}(\sigma_i^2\boldsymbol{Q}_i)\right) \quad \text{with} \quad i \in \{1,\cdots,o\} \qquad \text{and} \quad r_\mu = \operatorname{tr}\left(\boldsymbol{W}(\sigma_\mu^2\boldsymbol{Q}_\mu)\right), \tag{4.8}$$

where the matrix $\boldsymbol{W}$ is given as:

$$\boldsymbol{W} = \boldsymbol{\Sigma}_0^{-1} \; - \; \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Pi}_{\mathcal{S}(\boldsymbol{A})}^{\boldsymbol{\Sigma}_0^{-1}}\boldsymbol{\Sigma}_0^{-1}, \tag{4.9}$$

where

$$\Pi_{\mathcal{S}(\boldsymbol{A})}^{\boldsymbol{\Sigma}_0^{-1}} = \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T \tag{4.10}$$

denotes here the projector of column space of $\boldsymbol{A}$ with the metric $\boldsymbol{\Sigma}_0^{-1}$. Here $\boldsymbol{\Sigma}_0^{-1}$ stands for $\boldsymbol{\Sigma}$ in Eq. (4.3) with an approximate or initial values of variance components $\sigma_i^{2(0)}$ with $i \in \{1,\cdots,o\}$ and $\sigma_\mu^{2(0)}$.

One can reduce Eq. (4.8) for the computation of the partial redundancies, if matrix $\boldsymbol{W}$ in Eq.(4.9) can be written as:

$$\begin{aligned}\boldsymbol{W} = &\begin{bmatrix} \frac{1}{\sigma_1^2}\boldsymbol{P}_1 & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \dots & \dots & \dots & \dots & \dots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_o^2}\boldsymbol{P}_o & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu \end{bmatrix} - \begin{bmatrix} \frac{1}{\sigma_1^2}\boldsymbol{P}_1 & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \dots & \dots & \dots & \dots & \dots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_o^2}\boldsymbol{P}_o & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_1 \\ \dots \\ \boldsymbol{A}_o \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{N}^{-1} \\ &\begin{bmatrix} \boldsymbol{A}_1^T & \dots & \boldsymbol{A}_o^T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2}\boldsymbol{P}_1 & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \dots & \dots & \dots & \dots & \dots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_o^2}\boldsymbol{P}_o & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu \end{bmatrix}.\end{aligned} \tag{4.11}$$

The expanding of Eq. (4.11) yields:

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{\sigma_1^2}\boldsymbol{P}_1 - \frac{1}{\sigma_1^2}\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{N}^{-1}\boldsymbol{A}_1^T\left(\frac{1}{\sigma_1^2}\boldsymbol{P}_1\right) & \dots & \frac{1}{\sigma_1^2}\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{N}^{-1}\boldsymbol{A}_o^T\left(\frac{1}{\sigma_o^2}\boldsymbol{P}_o\right) & \frac{1}{\sigma_1^2}\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{N}^{-1}\left(\frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\right) \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{\sigma_o^2}\boldsymbol{P}_o\boldsymbol{A}_o\boldsymbol{N}^{-1}\boldsymbol{A}_1^T\left(\frac{1}{\sigma_1^2}\boldsymbol{P}_1\right) & \dots & \frac{1}{\sigma_o^2}\boldsymbol{P}_o - \frac{1}{\sigma_o^2}\boldsymbol{P}_o\boldsymbol{A}_o\boldsymbol{N}^{-1}\boldsymbol{A}_o^T\left(\frac{1}{\sigma_o^2}\boldsymbol{P}_o\right) & \frac{1}{\sigma_o^2}\boldsymbol{P}_o\boldsymbol{A}_o\boldsymbol{N}^{-1}\left(\frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\right) \\ \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\boldsymbol{A}_1^T\left(\frac{1}{\sigma_1^2}\boldsymbol{P}_1\right) & \dots & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\boldsymbol{A}_o^T\left(\frac{1}{\sigma_o^2}\boldsymbol{P}_o\right) & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu - \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\left(\frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\right) \end{bmatrix}. \tag{4.12}$$

The product $\boldsymbol{W}(\sigma_i^2\boldsymbol{Q}_i)$ in Eq. (4.8) is obtained for the j^{th} variance component by

$$\boldsymbol{W}(\sigma_j^2\boldsymbol{Q}_j) = \begin{bmatrix} \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{I}_j & \dots & \boldsymbol{0} \\ & & \vdots & & \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} & \dots & \frac{1}{\sigma_j^2}\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{N}^{-1}\boldsymbol{A}_j^T & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \dots & \frac{1}{\sigma_j^2}\boldsymbol{P}_j\boldsymbol{A}_j\boldsymbol{N}^{-1}\boldsymbol{A}_j^T & \dots & \boldsymbol{0} \\ & & \vdots & & \\ \boldsymbol{0} & \dots & \frac{1}{\sigma_j^2}\boldsymbol{P}_o\boldsymbol{A}_o\boldsymbol{N}^{-1}\boldsymbol{A}_j^T & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \frac{1}{\sigma_j^2}\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\boldsymbol{A}_j^T & \dots & \boldsymbol{0} \end{bmatrix}, \tag{4.13}$$

as well as the product $\boldsymbol{W}(\sigma_\mu^2\boldsymbol{Q}_\mu)$ is obtained by

$$\boldsymbol{W}(\sigma_\mu^2\boldsymbol{Q}_\mu) = \begin{bmatrix} \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ & & \vdots & & \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{I}_\mu \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_1^2}\boldsymbol{P}_1\boldsymbol{A}_1\boldsymbol{N}^{-1} \\ \vdots & \dots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_i^2}\boldsymbol{P}_j\boldsymbol{A}_j\boldsymbol{N}^{-1} \\ \vdots & \dots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_o^2}\boldsymbol{P}_o\boldsymbol{A}_o\boldsymbol{N}^{-1} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\boldsymbol{N}^{-1} \end{bmatrix}. \tag{4.14}$$

Therefore, the redundancy number r_i follows for Eq. 4.8 by

$$r_i = \operatorname{tr}\left(\boldsymbol{I}_i - \frac{1}{\sigma_i^2}\boldsymbol{P}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\right) \qquad \text{with} \qquad i \in \{1,\cdots,o\}. \tag{4.15}$$

Concerning the fact that $\operatorname{tr}(\boldsymbol{A}+\boldsymbol{B}) = \operatorname{tr}\boldsymbol{A} + \operatorname{tr}\boldsymbol{B}$, (cf. Koch 1999, p. 40), Eq. (4.15) simplifies to:

$$\begin{aligned} r_i =& \operatorname{tr}(\boldsymbol{I}_i) - \frac{1}{\sigma_i^2}\operatorname{tr}\left(\boldsymbol{P}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\right) \\ r_i =& n_i - \frac{1}{\sigma_i^2}\operatorname{tr}\left(\boldsymbol{P}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\right) \end{aligned} \tag{4.16}$$

and

$$\begin{aligned} r_\mu =& \operatorname{tr}(\boldsymbol{I}_\mu) - \frac{1}{\sigma_\mu^2}\operatorname{tr}\left(\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\right) \\ r_\mu =& u - \frac{1}{\sigma_\mu^2}\operatorname{tr}\left(\boldsymbol{P}_\mu\boldsymbol{N}^{-1}\right), \end{aligned} \tag{4.17}$$

where $\boldsymbol{I}_\mu$ is the $u \times u$ identity matrix (u here is number of unknown parameter).
The weight matrix of the observation group $\boldsymbol{P}_i$ in Eq. (4.16) can be substituted by the lower and upper symmetrical triangular matrices (computed by a Cholesky decomposition $\boldsymbol{P}_i = \boldsymbol{G}_i^T\boldsymbol{G}_i$). Note that the Cholesky decomposition is also applied to $\boldsymbol{P}_\mu = \boldsymbol{G}_\mu^T\boldsymbol{G}_\mu$ in equation Eq. (4.17). This yields to:

$$r_i = n_i - \frac{1}{\sigma_i^2}\operatorname{tr}\left(\boldsymbol{G}_i^T\boldsymbol{G}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\right) \qquad \text{and} \qquad r_\mu = u - \frac{1}{\sigma_\mu^2}\operatorname{tr}\left(\boldsymbol{G}_\mu^T\boldsymbol{G}_\mu\boldsymbol{N}^{-1}\right) \tag{4.18}$$

and because of $\operatorname{tr}(\boldsymbol{AB}) = \operatorname{tr}(\boldsymbol{BA})$ (cf. Koch 1999,S. 40) the partial redundancies for the computation of the variance components is then (cf. Eq. 21 in Koch and Kusche 2002)

$$r_i = n_i - \frac{1}{\sigma_i^2}\operatorname{tr}\left(\boldsymbol{G}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\boldsymbol{G}_i^T\right) \qquad \text{and} \qquad r_\mu = u - \frac{1}{\sigma_\mu^2}\operatorname{tr}\left(\boldsymbol{G}_\mu\boldsymbol{N}^{-1}\boldsymbol{G}_\mu^T\right). \tag{4.19}$$

The iterative estimation of the variance components $\hat{\boldsymbol{\sigma}} = \{\sigma_i^2,\cdots,\sigma_o^2,\sigma_\mu^2\}$ is carried out by the iterative procedure presented in algorithm 4.1:

Algorithmus 4.1 (VCE-Algorithm based on observation equations)

Purpose:	To compute the variance components		
Input:	$\boldsymbol{A}_i$ for $i \in \{1,\dots o\}$	...	design matrix
	$\boldsymbol{P}_i$ for $i \in \{1,\dots o\}$	...	weight matrix
	$\boldsymbol{l}_i$ for $i \in \{1,\dots o\}$	...	observation vector
	$\sigma_i^{(0)}$ for $i \in \{1,\dots o\}$	...	start values for the variances of each group
	$\boldsymbol{P}_\mu$	...	regularization matrix
	$\sigma_\mu^{(0)}$	...	start values for the variance of the regularization set
Output:	$\hat{\boldsymbol{x}}$	...	estimates for the unknown parameters
	σ_i^2 for $i \in \{1,\dots o\}$	...	variance factors of each group
	σ_μ^2	...	variance factor of the regularization matrix

1 Start with apriori values $\sigma_i^{2(0)}$ for the variances of each observation group for $i \in \{1, ...o\}$ and for $\sigma_\mu^{2(0)}$ and set $k = 1$

2 Compute the joint normal equation matrix:
$\boldsymbol{N}^{(k)} = \left(\frac{1}{\sigma_1^{2(k)}} \boldsymbol{A}_1^T \boldsymbol{P}_1 \boldsymbol{A}_1 + \ldots + \frac{1}{\sigma_o^{2(k)}} \boldsymbol{A}_o^T \boldsymbol{P}_o \boldsymbol{A}_o + \frac{1}{\sigma_\mu^{2(k)}} \boldsymbol{P}_\mu \right)$
and the joint right-hand side:
$\boldsymbol{b}^{(k)} = \left(\frac{1}{\sigma_1^{2(k)}} \boldsymbol{A}_1^T \boldsymbol{P}_1 \boldsymbol{l}_1 + \ldots + \frac{1}{\sigma_o^{2(k)}} \boldsymbol{A}_o^T \boldsymbol{P}_o \boldsymbol{l}_o + \frac{1}{\sigma_\mu^{2(k)}} \boldsymbol{\mu} \right)$

3 Estimate the parameters $\hat{\boldsymbol{x}}^{(k)}$ by solving the normal equations:
$\boldsymbol{N}^{(k)} \hat{\boldsymbol{x}}^{(k)} = \boldsymbol{b}^{(k)}$

4 Compute the residual vectors of each group: $\boldsymbol{v}_i{}^{(k)} = \boldsymbol{A}_i \hat{\boldsymbol{x}}^{(k)} - l_i$ for $i \in \{1, ...o\}$ and $\boldsymbol{v}_\mu{}^{(k)} = \hat{\boldsymbol{x}}^{(k)} - \boldsymbol{\mu}$

5 Compute the partial redundancies $r_i^{(k)}$ and $r_\mu^{(k)}$ of each group:
$r_i^{(k)} = n_i - \frac{1}{\sigma_i^{(k)2}} \mathrm{tr} \left(\boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{N}^{-1(k)} \boldsymbol{A}_i^T \boldsymbol{G_i}^T \right)$ and $r_\mu^{(k)} = u - \frac{1}{\sigma_\mu^{(k)2}} \mathrm{tr} \left(\boldsymbol{G}_\mu \boldsymbol{N}^{-1(k)} \boldsymbol{G}_\mu^T \right)$
and set $k = k + 1$

6 Compute the $\sigma_i^{2(k+1)} = \frac{\hat{\boldsymbol{v}}_i^{(k)T} \boldsymbol{P}_i \hat{\boldsymbol{v}}_i^{(k)}}{r_i^{(k)}}$ for $i \in \{1, ...o\}$ and $\sigma_\mu^{2(k+1)} = \frac{\hat{\boldsymbol{v}}_\mu^{(k)T} \boldsymbol{P}_\mu \hat{\boldsymbol{v}}_\mu^{(k)}}{r_\mu^{(k)}}$, respectively.

7 Set update values for the variance factors and repeat steps 2 to 6 until a point of convergence is reached.

Inspection of Alg. 4.1 reveals a few problems:

- It can be seen from step 5 of Alg. 4.1 that a rigorous computation of the variance components requires a repeated computation of the trace of the inverse of the combined normal equation matrix $\boldsymbol{N}^{(k)}$, where k denotes the iteration step. This computation requires a great deal of working memory and processing time for very large systems. Furthermore, within an iterative solution procedure (such as PCGMA), the expensive assembling of the combined normal equation matrix $\boldsymbol{N}^{(k)}$ for the heterogeneous observation types (steps 2 and 5 of Alg. 4.1) is avoided. Consequently, the computation of the partial redundancies is not possible without modification. This problem is solved by a stochastic trace estimation which will be introduced in Sect. 4.3.1. Alg. 4.1.

- As far as the computation of the partial redundancies in step 5 is concerned, some of the matrices are not given. The assembling of the weight matrix $\boldsymbol{P}_i$ of the corresponding observation group is not always possible due to the high dimension of the adjustment problem with millions of observations. A solution to this problem is achieved by applying suitable ARMA filters (see Sect. 3.3.2).

- The computation of the partial redundancies in step 5 requires the presence of the design matrices $A_i, i \in \{1, \cdots, o\}$ of all observation groups, which is not always available. For instance, an adjustment of SGG and SST data within the GOCE mission by means of PCGMA is performed by combining the observation equations for the SGG data with the normal equation system for the SST data. The latter is obtained via an external interface (see Boxhammer 2006). Furthermore, both systems are usually of different dimensions. In Sect. 4.3.2 the solution of this problem is outlined .

4.3 Monte Carlo variance component estimation

4.3.1 Monte Carlo trace estimator

It becomes evident from Eq. (4.18) that an iterative estimation of the trace term

$$t = \mathrm{tr} \left(\boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{N}^{-1} \boldsymbol{A}_i^T \boldsymbol{G}_i^T \right) \tag{4.20}$$

requires the expensive computation of the inverse $\boldsymbol{N}^{-1}$ of the combined normal equation system $\boldsymbol{N}^{-1}$. To avoid this computation, Koch and Kusche (2002) and Kusche (2003) demonstrated an alternative Monte Carlo approach, which is based on a substitution of the rigorous determination of the term t in Eq. (4.20) by a stochastic trace estimator. This method is called Monte Carlo variance component estimation (MCVCE).
Let $\boldsymbol{B}$ a positive definite $n \times n$ matrix and $\boldsymbol{\mathfrak{Z}}$ a random $n \times 1$ vector of n independent samples from a random vector with $E(\boldsymbol{\mathfrak{Z}}) = \boldsymbol{0}$ and $\Sigma \{\boldsymbol{\mathfrak{Z}}\} = \boldsymbol{I}$. Then,

$$E\{\boldsymbol{\mathfrak{Z}}^T\boldsymbol{B}\boldsymbol{\mathfrak{Z}}\} = \operatorname{tr}\boldsymbol{B}, \tag{4.21}$$

holds (see KOCH 1999, p. 134). In GIRARD (1989), q pseudo-random numbers following a multivariate normal distribution $\boldsymbol{\mathfrak{Z}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ were generated, and the trace of the matrix $\boldsymbol{B}$ was approximated by the unbiased estimation:

$$\operatorname{tr}\boldsymbol{B} = \frac{1}{q}\sum_{i=1}^{q}\boldsymbol{z}^T\boldsymbol{B}\boldsymbol{z}. \tag{4.22}$$

HUTCHINSON (1990) proved that the estimator Eq. (4.22) has minimal variance if $\boldsymbol{\mathfrak{Z}}$ follows a multivariate discrete distribution, which takes the values -1 or $+1$ with probability $\frac{1}{2}$. This reads:

$$\mathfrak{Z} \sim \mathcal{U}\begin{cases} -1 & \text{with probability } \frac{1}{2} \\ +1 & \text{with probability } \frac{1}{2} \end{cases} \tag{4.23}$$

GOLUB and VON MATT (1997) suggested for an accurate determining the trace of the matrix $\boldsymbol{B}$ just one sample vector $\boldsymbol{z}$. KOCH and KUSCHE (2002) and KUSCHE (2003) reported similar conclusions in the context of the choice of weighting factors and regularization parameters for GOCE data. The tests which has been carried out in Sect. 6.2 validate this conclusions. When the stochastic trace estimator in Eq. (4.22) is used, the term t in Eq. (4.20) may be substituted by:

$$\bar{t} = \boldsymbol{z}_i^T\left(\boldsymbol{G}_i\boldsymbol{A}_i\boldsymbol{N}^{-1}\boldsymbol{A}_i^T\boldsymbol{G}_i^T\right)\boldsymbol{z}_i. \tag{4.24}$$

Now let the auxiliary parameter vector $\boldsymbol{\alpha}_i$ in Eq. (4.24) define:

$$\boldsymbol{\alpha}_i = \boldsymbol{N}^{-1}\boldsymbol{A}_i^T\boldsymbol{G}_i^T\boldsymbol{z}_i \qquad \Leftrightarrow \qquad \boldsymbol{\alpha}_i\boldsymbol{N} = \boldsymbol{A}_i^T\boldsymbol{G}_i^T\boldsymbol{z}_i. \tag{4.25}$$

Then, the substitute form of the first part of Eq. (4.18) is obtained as:

$$r_i = n_i - \frac{1}{\sigma_i^2}\boldsymbol{z}^T\boldsymbol{G}_i\boldsymbol{A}_i\boldsymbol{\alpha}_i. \tag{4.26}$$

The partial redundancy r_μ used for determining the regularization parameter $\frac{1}{\sigma_\mu^2}$ then equals the difference between the total number of observations n (the sum of the individual numbers n_i of observations within each group and the dimension $n_\mu = m$ of the apriori parameters $\boldsymbol{\mu}$) and the number m of parameters:

$$\begin{aligned} r &= n - m \\ \sum_{i=1}^{o} r_i + r_\mu &= \sum_{i=1}^{o} n_i + n_\mu - m \qquad \Leftrightarrow \qquad r_\mu = \sum_{i=1}^{o} n_i - \sum_{i=1}^{o} r_i. \end{aligned} \tag{4.27}$$

The Monte Carlo trace estimator is incorporated into Alg. 4.1. The computational steps of the MCVCE algorithm are summarized in Alg. 4.2:

Algorithmus 4.2 (MCVCE-Algorithm: version based only on observation equations)

Purpose:	To compute the variance components		
Input:	$\boldsymbol{A}_i$ for $i \in \{1, \dots o\}$	...	design matrix
	$\boldsymbol{P}_i$ for $i \in \{1, \dots o\}$	...	weight matrix
	$\boldsymbol{l}_i$ for $i \in \{1, \dots o\}$	...	observation vector
	$\sigma_i^{(0)}$ for $i \in \{1, \dots o\}$	...	start values for the variances for all groups
	$\boldsymbol{P}_\mu$	...	regularization matrix
	$\sigma_\mu^{(0)}$	...	start value for the variance of the regularization set
Output:	$\hat{\boldsymbol{x}}$	...	estimates for the unknown parameters
	σ_i^2 for $i \in \{1, \dots o\}$, σ_μ^2	...	variance components

1 Start with apriori values for the variances of the observation groups $\sigma_i^{2\,(0)}$ for $i \in \{1, ...o\}$ and for $\sigma_\mu^{2\,(0)}$ and set $k = 1$

2 Generate the random vector $\boldsymbol{z}_i$ for $i \in \{1, \cdots, o\}$ from the distribution given in Sect. 4.3.1

3 Compute the joint normal equation:

$$\boldsymbol{N}^{(k)} = \sum_{i=1}^{o} \left(\frac{1}{\sigma_i^{2\,(k)}} \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_+ \frac{1}{\sigma_\mu^{2\,(k)}} \boldsymbol{P}_\mu \right)$$

and the joint right-hand side:

$$\boldsymbol{b}^{(k)} = \sum_{i=1}^{o} \left(\frac{1}{\sigma_i^{2\,(k)}} \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{l}_i + \frac{1}{\sigma_\mu^{2\,(k)}} \boldsymbol{\mu} \right)$$

4 Estimate the parameter $\hat{\boldsymbol{x}}^{(k)}$ as well as $\boldsymbol{\alpha}_i, i \in \{1, \cdots, o\}$ by solving the normal equation for multiple right-hand sides:
$\boldsymbol{N}^{(k)} \left(\hat{\boldsymbol{x}}^{(k)} \;\; \hat{\boldsymbol{\alpha}}_1{}^{(k)} \;\; \ldots \;\; \hat{\boldsymbol{\alpha}}_o{}^{(k)} \right) = \left(\boldsymbol{b}^{(k)} \;\; \boldsymbol{A}_1^T \boldsymbol{G}_1^T \boldsymbol{z}_1 \;\; \ldots \;\; \boldsymbol{A}_o^T \boldsymbol{G}_o^T \boldsymbol{z}_o \right)$

5 Compute the residual vectors of groups $\boldsymbol{v}_i{}^{(k)} = \boldsymbol{A}_i \hat{\boldsymbol{x}}^{(k)} - l_i$ for $i \in \{1, ...o\}$ and $\boldsymbol{v}_\mu{}^{(k)} = \hat{\boldsymbol{x}}^{(k)} - \boldsymbol{\mu}$

6 Compute the partial redundancies of groups $r_i^{(k)}$ and $r_\mu^{(k)}$:
$r_i^{(k)} = n_i - \frac{1}{\sigma_i^{(k)2}} \left(\boldsymbol{z}_i^T \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{\alpha}_i \right)$ and $r_\mu^{(k)} = \sum_{i=1}^{o} n_i + u - \sum_{i=1}^{o} r_i^{(k)}$
and set $k = k + 1$

7 Compute update values for the variance factors:

$$\sigma_i^{2(k+1)} = \frac{\hat{\boldsymbol{v}}_i^{(k)T} \boldsymbol{P}_i \hat{\boldsymbol{v}}_i^{(k)}}{r_i^{(k)}} \quad \text{for} \quad i \in \{1, ...o\} \quad \text{and} \quad \sigma_\mu^{2(k+1)} = \frac{\hat{\boldsymbol{v}}_\mu^{(k)T} \boldsymbol{P}_\mu \hat{\boldsymbol{v}}_\mu^{(k)}}{r_\mu^{(k)}}$$

8 Substitute the update values for the variance factors into step 2 and repeat steps 2 to 7 until the termination criteria is satisfied.

The main difference between Alg. 4.1 and Alg. 4.2 is the step for computing the partial redundancies. While this step cannot be performed in Alg. 4.1 without determination of the inverse of the combined normal equation matrix and without calculation of the expensive matrix-matrix multiplication (see step 5 of Alg. 4.1), computation of the partial redundancy of observation group i in Alg. 4.2 is only based on matrix-vector multiplications. The auxiliary parameter vector $\boldsymbol{\alpha}_i$ in Alg. 4.2 is estimated by solving the normal equation system for multiple right-hand sides (see step 4 of Alg. 4.2). This estimation may be done either directly of iteratively. In doing so, the MCVCE algorithm is built into PCGMA in Sect. 5.4.2.

4.3.2 Combination of observation equations and normal equations

Design matrix of some observation groups

Alg. 4.2 requires the presence of design matrices $\boldsymbol{A}_i, i \in \{1, \cdots, o\}$ for the different groups of observations, which is not the case for every adjustment model. For instance, the processing of GOCE data (see Sect. 5.4.2) by means of PCGMA is based on a combination of SGG and SST. The SST data are available only as a normal equation matrix $\boldsymbol{N}_{\text{sst}}$ and the corresponding right-hand side $\boldsymbol{b}_{\text{sst}}$ (which are usually processed from external interfaces), while the normal equations of the SGG data are assembled without computing the normal equation matrix (see Schuh 1996). Therefore, the variance components cannot be computed without modifying Alg. 4.2.

In order to compute the partial redundancies with respect to the groups of observations, which are available only through a normal equation matrix and a right-hand side, the trace of the matrix $\left(\boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{N}^{-1} \boldsymbol{A}_i^T \boldsymbol{G}_i^T \right)$ in Eq. (4.20) may be rewritten as:

$$\begin{aligned} \boldsymbol{t} &= \operatorname{tr} \left(\boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{N}^{-1} \boldsymbol{A}_i^T \boldsymbol{G}_i^T \right) \\ \boldsymbol{t} &= \operatorname{tr} \left(\boldsymbol{A}_i^T \boldsymbol{G}_i^T \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{N}^{-1} \right) \\ \boldsymbol{t} &= \operatorname{tr} \left(\boldsymbol{N}_i \boldsymbol{N}^{-1} \right). \end{aligned} \tag{4.28}$$

Applying again the stochastic trace estimator (Eq. 4.22) (see Sect. 4.3.1) to Eq. (4.28) leads to the following substitute equation for Eq. (4.28):

$$\bar{\boldsymbol{t}} = \boldsymbol{w}_i^T \left(\boldsymbol{N}_i \boldsymbol{N}^{-1} \right) \boldsymbol{w}_i. \tag{4.29}$$

The vector $\boldsymbol{w}$ in Eq. (4.29) is a realization of discrete distribution $\mathcal{U}$ given by 4.23, from which the vector $\boldsymbol{z}$ was generated (see Eq. (4.24)). They only differ in dimension. $\boldsymbol{w}$ is a $u \times 1$-vector, where u is the dimension of the $i-$th normal equation matrix.

The computational formula for the partial redundancy of the $i-$th group of observations (which are present only through the normal equation matrix $\boldsymbol{N}_i$) is then given by:

$$r_i = n_i - \frac{1}{\sigma_i^2} \left(\boldsymbol{w}_i^T \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{w}_i \right). \tag{4.30}$$

Defining the auxiliary parameter vector $\boldsymbol{\beta}_i$ in Eq. (4.30) as the solution of the normal equation system for multiple right-hand sides $\boldsymbol{w}_i$ as:

$$\boldsymbol{\beta}_i \boldsymbol{N} = \boldsymbol{w}, \tag{4.31}$$

yields for Eq. (4.30):

$$r_i = n_i - \frac{1}{\sigma_i^2} \boldsymbol{w}_i^T \boldsymbol{N}_i \boldsymbol{\beta}_i. \tag{4.32}$$

Residual vector of some observation groups
If the design matrix for the $i-$th group cannot be assembled, then the computation of the residual vector in step 5 of Alg. 4.2 by means of the equation $\boldsymbol{v}_i = \boldsymbol{A}_i \hat{\boldsymbol{x}} - \boldsymbol{l}_i$ is not possible. The residual vector $\boldsymbol{v}_i$ of the $i-$th group of observations is required for computing the square sum of residuals $\boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i$, which in turn is necessary for computing an update value for the variance factor. An alternative equation for computing the square sum of residuals is derived as follows:

$$\begin{aligned} \boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i &= \left(\boldsymbol{A}_i \hat{\boldsymbol{x}} - \boldsymbol{l}_i \right)^T \boldsymbol{P}_i \left(\boldsymbol{A}_i \hat{\boldsymbol{x}} - \boldsymbol{l}_i \right) \\ &= \left(\hat{\boldsymbol{x}}^T \boldsymbol{A}_i^T - \boldsymbol{l}_i^T \right) \boldsymbol{P}_i \left(\boldsymbol{A}_i \hat{\boldsymbol{x}} - \boldsymbol{l}_i \right) \\ &= \hat{\boldsymbol{x}}^T \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \hat{\boldsymbol{x}} - 2 \hat{\boldsymbol{x}}^T \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{l}_i + \boldsymbol{l}_i^T \boldsymbol{P}_i \boldsymbol{l}_i. \end{aligned} \tag{4.33}$$

Substitution of $\boldsymbol{N}_i$ for $\boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i$ and $\boldsymbol{b}_i$ for $\boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{l}_i$ in Eq. (4.33) yields:

$$\boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i = \hat{\boldsymbol{x}}^T \boldsymbol{N}_i \hat{\boldsymbol{x}} - 2 \hat{\boldsymbol{x}}^T \boldsymbol{b}_i + \boldsymbol{l}_i^T \boldsymbol{P}_i \boldsymbol{l}_i. \tag{4.34}$$

Substituting the above mentioned modifications:

1. Decorrelated groups of observations $\bar{\boldsymbol{A}}_i$ instead of $\boldsymbol{A}$, $\boldsymbol{P}_i = \boldsymbol{G}_i^T \boldsymbol{G}_i = \boldsymbol{I}$;
2. The alternative equation Eq. (4.32) for computing the partial redundancy of the $i-$th group of normal equations (given by $\boldsymbol{N}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{l}_i^T \boldsymbol{P}_i \boldsymbol{l}_i$);
3. The alternative equation Eq. (4.34) for computing the square sum of residuals of the group of normal equations;

into Alg. 4.2, leads to the following algorithm:

Algorithmus 4.3 (MCVCE-Algorithm: version based on combination of observation and normal equations)

Purpose:	To compute the variance components		
Input:	$\bar{\boldsymbol{A}}$	...	decorrelated design matrix
	$\bar{\boldsymbol{l}}$	...	deorrelated observation vector
	$\boldsymbol{N}_i$ for $i \in \{2, \dots o\}$	...	normal equations
	$\boldsymbol{b}_i$ for $i \in \{2, \dots o\}$	...	right-hand sides
	$\boldsymbol{l}_i^T \boldsymbol{P}_i \boldsymbol{l}_i$ for $i \in \{2, \dots o\}$	...	square sum of observations
	n_i for $i \in \{1, \dots o\}$	...	number of observations
	$\sigma_i^{(0)}$ for $i \in \{1, \dots o\}$	...	start values for the variances for all groups
	$\boldsymbol{P}_\mu$	...	weight matrix for the regularization matrix
	$\sigma_\mu^{(0)}$	...	start value for the variance of the regularization set
Output:	$\hat{\boldsymbol{x}}$	...	estimates for the unknown parameters
	σ_i^2 for $i \in \{1, \dots o\}$ and σ_μ^2	...	variance components

1 Generate the random vector $\boldsymbol{z}$ of dimension $n_1 \times 1$ and the vector $\boldsymbol{w}_i$ for $i \in \{2, \cdots, o\}$ of dimension $n_i \times 1$ from the distribution given in Sect. 4.3.1. Set $k = 1$.

2 Compute the joint normal equation as well as the joint right-hand side:

$$\boldsymbol{N}^{(k)} = \frac{1}{\sigma_1^{2(k)}} \bar{\boldsymbol{A}}^T \bar{\boldsymbol{A}} + \sum_{i=2}^{o} \left(\frac{1}{\sigma_i^{2(k)}} \boldsymbol{N}_i \right) + \frac{1}{\sigma_\mu^{2(k)}} \boldsymbol{P}_\mu \quad , \quad \boldsymbol{b}^{(k)} = \frac{1}{\sigma_1^{2(k)}} \bar{\boldsymbol{A}}^T \bar{\boldsymbol{l}} + \sum_{i=2}^{o} \left(\frac{1}{\sigma_i^{2(k)}} \boldsymbol{b}_i \right) + \frac{1}{\sigma_\mu^{2(k)}} \boldsymbol{\mu}$$

3 Estimate the parameter $\hat{\boldsymbol{x}}^{(k)}$, $\boldsymbol{\alpha}$ as well as $\beta_i, i \in \{2, \cdots, o\}$ by solving the normal equations for multiple right-hand sides:

$$\boldsymbol{N}^{(k)} \left(\hat{\boldsymbol{x}}^{(k)} \ \hat{\boldsymbol{\alpha}}^{(k)} \ \hat{\boldsymbol{\beta}}_1^{(k)} \ \dots \ \hat{\boldsymbol{\beta}}_o^{(k)} \right) = \left(\boldsymbol{b}^{(k)} \ \bar{\boldsymbol{A}}^T \boldsymbol{z} \ \boldsymbol{w}_1 \ \dots \ \boldsymbol{w}_o \right)$$

4 Compute the square sum of residuals of the observation groups, the groups of normal equations and the regularization:

$$\begin{aligned}
\left(\bar{\boldsymbol{v}}_1^T \bar{\boldsymbol{v}}_1 \right)^{(k)} &= \left(\bar{\boldsymbol{A}} \hat{\boldsymbol{x}}^{(k)} - \bar{l} \right)^T \left(\bar{\boldsymbol{A}} \hat{\boldsymbol{x}}^{(k)} - \bar{l} \right) && \dots \quad \text{for the groups of observation equations} \\
\left(\boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i \right)^{(k)} &= \hat{\boldsymbol{x}}^T \boldsymbol{N}_i \hat{\boldsymbol{x}} - 2 \hat{\boldsymbol{x}}^T \boldsymbol{b}_i + \boldsymbol{l}_i^T \boldsymbol{P}_i \boldsymbol{l}_i && \dots \quad \text{for the groups of normal equations} \\
\left(\boldsymbol{v}_\mu^T \boldsymbol{P}_\mu \boldsymbol{v}_\mu \right)^{(k)} &= \left(\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{\mu} \right)^T \left(\hat{\boldsymbol{x}}^{(k)} - \boldsymbol{\mu} \right) && \dots \quad \text{for the regularization parameter}
\end{aligned}$$

5 Compute the partial redundancies of groups $r_1^{(k)}$, $r_i^{(k)}$ for $i \in \{2, \dots o\}$ and $r_\mu^{(k)}$:

$$\begin{aligned}
r_1^{(k)} &= n_1 - \frac{1}{\sigma_1^{(k)2}} \left(\boldsymbol{z}^T \bar{\boldsymbol{A}} \boldsymbol{\alpha} \right) && \dots \quad \text{for the groups of observation equations} \\
r_i^{(k)} &= n_i - \frac{1}{\sigma_i^{(k)2}} \boldsymbol{w}_i^T \boldsymbol{N}_i \boldsymbol{\beta}_i && \dots \quad \text{for the groups of normal equations} \\
r_\mu^{(k)} &= \sum_{i=1}^{o} n_i + u - \sum_{i=1}^{o} r_i^{(k)} && \dots \quad \text{for the regularization parameter}
\end{aligned}$$

and set $k = k + 1$

6 Compute update values for the variance factors:

$$\sigma_1^{2(k+1)} = \frac{\left(\bar{\boldsymbol{v}}_1^T \bar{\boldsymbol{v}}_1 \right)^{(k)}}{r_1^{(k)}}, \sigma_i^{2(k+1)} = \frac{\left(\boldsymbol{v}_i^T \boldsymbol{P}_i \boldsymbol{v}_i \right)^{(k)}}{r_i^{(k)}} \quad \text{for} \quad i \in \{2, \cdots o\} \quad \text{and} \quad \sigma_\mu^{2(k+1)} = \frac{\left(\boldsymbol{v}_\mu^T \boldsymbol{P}_\mu \boldsymbol{v}_\mu \right)^{(k)}}{r_\mu^{(k)}}$$

7 Substitute the update values for the variance factors into step 2 and repeat steps 3 to 6 until the convergence.

5 Integration of Monte Carlo methods into PCGMA

5.1 Solution strategies for GOCE data

According to ESA (1999) the main goal of the GOCE mission is the determination of a gravity field model which surpasses any existing model in resolution with a half-wavelength of about $100km$ and accuracy ($1-2mgal$ for geoid anomalies and $1cm$ for geoid heights, respectively). For this purpose, GOCE will be equipped with a gradiometer, a Global Positioning System (GPS) receiver, and a laser retro-reflector. The gradiometer is based on a highly sensitive configuration of six accelerometers whose differential observables correspond to the second derivatives of the gravitational potential. This measurement principle is called satellite gravity gradiometry or SGG. To determine the long wavelengths accurately, SGG data are combined with high-low satellite-to-satellite tracking (SST) data. The expected resolution corresponds to a spherical harmonic series up to degree and order 240, which results in about 60,000 unknown parameters. This results in very large equation systems, which do not fit into the work memory of a current personal computer. To be more specific, the computation of the normal equation matrix takes approximately 400 days considering the current computation speed and its storage requires about 25 GByte.

The processing of such a large volume of data promises a challenging task. Therefore, our research group at the IGG in Bonn have designed a tuning software ("tuning machine") which operates on the basis of the iterative PCGMA. This tool will be used to determine the optimal gravity model and will be realized in the GOCE HPF as well as in GT GOCE-GRAND II (Geotechnologien GOCE-GRavitationsfeld-ANalyse Deutschland II, Rummel 2005). Others gravity processing tools are presented in Sect. 1.1.

But not only an efficient solution is necessary to assess a reliable GOCE gravity field. In addition, the optimal choice of relative weighting factors between the SST data which mainly provide low-frequency information and SGG data which exploit the high frequency information is of interest. Therefore, the algorithms of variance components estimation presented in Sect.4.3 will be performed during GT GOCE GRAND II and GOCE HPF. Now it is necessary to integrate and implement these procedures into PCGMA.

In order to impart the information of the entire gravity model the variance/covariance information of the estimated gravity field parameters must be provided. However, because of huge storage requirements and the incapability of the PCGMA to compute the full VCM, this variance/covariance information would not be available for many users. As an alternative way to compute this information, the Monte Carlo algorithms in Sect. 3.2 will be applied. In this context the PCGMA package will be redesigned to implement the possibility of calculating the VCM.

In the next section, the two data types of GOCE observations, SGG and SST, as well as the regularization part are shortly described. In Sect. 5.3, the computational steps of the PCGMA algorithm are summarized. The integration of Monte Carlo algorithms 3.9 and 4.3 (which are discussed in Chaps. 3 and 4, respectively) into PCGMA are treated in Sect. 5.4.

5.2 Functional and stochastic model

The main goal of all strategies for determining the gravity field is the computation of the spherical harmonic coefficients $\bar{C}_{lm}$ and $\bar{S}_{lm}$ from the series expansion of the gravity potential:

$$V(r,\theta,\lambda) = \frac{GM}{R}\sum_{l=0}^{\infty}\left(\frac{R}{r}\right)^{l+1}\sum_{m=0}^{l}\bar{P}_{lm}(\cos\theta)[\bar{C}_{lm}\cos m\lambda + \bar{S}_{lm}\sin m\lambda], \tag{5.1}$$

where GM denotes the geocentric constant as the product of the gravitational constant and the Earth mass, R the Earth radius, r the distance to the Earth's center of a point whose gravitational potential is to be determined, λ, θ the geographic longitude and latitude of that point, $\bar{P}_{lm}$ the fully normalized Legendre functions and l, m the degree and order of the series expansion. In practice, the series (Eq. 5.1) is expanded only up to a maximal degree $l_{\max}$. The higher the number $l_{\max}$, the better this series fits the actual shape of the gravitational potential, but in contrast the numerical stability decreases because of the downward continuation effect.

5.2.1 SGG observations

The observation equations for SGG data comprise measurements of the second derivatives of the gravitational potential (Eq. 5.1). Four of the components of the SGG tensor will be recorded with high accuracy: T_{xx}, T_{yy}, T_{zz} and T_{xz}, where x, y, z refer to a body-fixed reference frame with x-axis aligned to the satellite's orbit ,z pointing to the Earth and y-axis perpendicular to the (x, z) plane. A detailed derivation of the functional SGG model may be found, for instance, in RUMMEL et al. (1993). For the *time-wise* approach it is assumed that the orbit of the GOCE satellite is known (e.g., determined from high-low SST observations). Therefore, the spherical harmonic coefficients are the only parameters within the SGG observation equations to be estimated, and these parameters are in a linear relationship with the SGG observations (see KLEES et al. 2000).

If all of the SGG data are combined in a vector $\boldsymbol{l}$ with corresponding covariance matrix $\boldsymbol{\Sigma}$, then the SGG observation equations may be written as (cf. Eq. 3.48):

$$\boldsymbol{l} + \boldsymbol{v} = \boldsymbol{A}\boldsymbol{x}, \tag{5.2}$$

where $\boldsymbol{A}$ is the design matrix of the SGG observations, $\boldsymbol{v}$ the SGG residual vector, and $\boldsymbol{x} = \left(\bar{C}_{lm}, \bar{S}_{lm}\right)$ the parameter vector. The accuracy of the SGG observations will be considerably downgraded by a number of error sources such as instrument errors, satellite errors, processing errors, etc. A detailed description of these error sources is found in ESA (1999). These errors result in colored noise, i.e. they cause the covariance matrix to be fully populated:

$$\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{P}^{-1}, \tag{5.3}$$

with the apriori variance factor σ^2 and the weight matrix of the SGG observations $\boldsymbol{P}^{-1}$. SCHUH (1996) demonstrated that the SGG observations may be viewed as a time series (according to the time-wise approach) and thus may be decorrelated by means of digital filters. This is done by applying ARMA filters to the observation equations (Eq. 5.2), which corresponds to a linear transformation (see Sect. 3.3.2). This leads from the model (Eq. 5.2) with Eq. (5.3) to the transformed observation equations:

$$\bar{\boldsymbol{l}} + \bar{\boldsymbol{v}} = \bar{\boldsymbol{A}}\boldsymbol{x} \qquad \text{with} \qquad \bar{\boldsymbol{\Sigma}} = \sigma^2 \boldsymbol{I}. \tag{5.4}$$

ARMA filters were integrated into PCGMA (see BOXHAMMER 2006). The redundant equation system (Eq. 5.4) may then be solved according to a Gauss-Markov model (see Sect. 3.1). The normal equation system of the SGG observation reads:

$$\bar{\boldsymbol{A}}^T \bar{\boldsymbol{A}} \hat{\boldsymbol{x}} = \bar{\boldsymbol{A}}^T \bar{\boldsymbol{l}}. \tag{5.5}$$

Efficient techniques for solving the normal equation system (Eq. 5.5) that avoid assemblance of the normal equation matrix $\bar{\boldsymbol{A}}^T \bar{\boldsymbol{A}}$ will be treated together with PCGMA (Sect. 5.3).

5.2.2 SST observations

The GOCE SST data consist in the code and phase observations of the on board GPS receiver. Their main task is to cover the long and medium wavelength of the potential, unaccessible to SGG. In the last years, various approaches have been developed for determining this part of the spectrum by SST observations. These approaches essentially differ in their functional or stochastic model, resulting in different linear or non-linear observation equations. One of the most popular methods is based on the principle of energy preservation within a closed system. This approach, which has been considered already in the early years of satellite geodesy (see REIGBER 1969), requires that observations are given densely. However, this requirement is satisfied since the CHAMP and GRACE satellite missions (see, e.g., ILK and LÖCHER 2003, FÖLDVARY et al. 2005, KUSCHE and VAN LOON 2004, and LÖCHER and ILK 2005). In MAYER-GÜRR (2006), a detailed description of these and other methods for assembling the observation equations with respect to different satellite observables is given.
The linear or linearized model reads:

$$\boldsymbol{l}_{\mathrm{sst}} + \boldsymbol{v}_{\mathrm{sst}} = \boldsymbol{A}_{\mathrm{sst}}\boldsymbol{x} \qquad \text{with} \qquad \boldsymbol{\Sigma}_{\mathrm{sst}} = \sigma^2 \boldsymbol{P}_{\mathrm{sst}}^{-1}. \tag{5.6}$$

Formally, the method of least squares leads to the normal equation system:

$$\boldsymbol{N}_{\mathrm{sst}}\hat{\boldsymbol{x}} = \boldsymbol{b}_{\mathrm{sst}} \qquad \text{with} \qquad \boldsymbol{N}_{\mathrm{sst}} = \boldsymbol{A}_{\mathrm{sst}}^T \boldsymbol{P}_{\mathrm{sst}} \boldsymbol{A}_{\mathrm{sst}} \qquad \text{and} \qquad \boldsymbol{b}_{\mathrm{sst}} = \boldsymbol{A}_{\mathrm{sst}}^T \boldsymbol{P}_{\mathrm{sst}} \boldsymbol{l}_{\mathrm{sst}}. \tag{5.7}$$

The SST observations are available as a normal equation system, which comprises the normal equation matrix $\boldsymbol{N}_{\mathrm{sst}}$ and the corresponding right-hand side $\boldsymbol{b}_{\mathrm{sst}}$. Therefore, the setting up of the underlying observation equations will not be explained in this thesis.

5.2.3 Regularization

In the course of GOCE data processing, the SGG data (resolved up to degree 240) are combined with SST data (resolved up to degree 100) for a joint adjustment. The combined normal equation system has a weak condition for various reasons: A first problem is that the observations are made at the height of the satellite and not on the Earth's surface, which is also called the downward continuation problem. A second factor contributing to the weak condition is a result of the problem that, due to the geometry of the satellite's orbit, no observations are collected over the polar regions. Among others, these causes lead to the problem that the harmonic coefficients cannot be estimated accurately from the data alone. Therefore, a regularization of the combined normal equation system is essential. For instance, the so-called spherical cap regularization method (Metzler and Pail 2005) was proposed to stabilize the system in polar regions by filling the polar caps with data from an analytic model. In comparison to other regularization methods this approach has the advantage that the regularization part affects only the polar regions (Metzler 2007).

Tikhonov regularization
Another popular method is Tikhonov regularization for which Hansen (1997) gives a detailed overview. The key principle of this approach consists in the minimization of the quadratic function:

$$J_\lambda(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{l}\|^2 + \lambda \|\boldsymbol{L}\boldsymbol{x}\|^2, \tag{5.8}$$

where λ in denotes the regularization parameter. The function $\|\boldsymbol{L}\boldsymbol{x}\|^2$ is called the discrete smoothing norm, where $\boldsymbol{L}$ is a semi positive definite matrix. $\boldsymbol{L}$ is in generally the discrete approximation of a derivation operator. Thus the unique solution of the minimization problem (Eq. 5.8) is:

$$\hat{\boldsymbol{x}}_{\boldsymbol{P}_{\boldsymbol{\mu}},\lambda} = \left(\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} + \lambda \boldsymbol{L}^T\boldsymbol{L}\right)^{-1} \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{l}, \tag{5.9}$$

where $\boldsymbol{L}^T\boldsymbol{L} = \boldsymbol{P}_\mu$ denotes here the regularization matrix. Usually, a positive definite matrix $\boldsymbol{P}_\mu$ (multiplied by a regularization factor λ) is added to the complete normal equation matrix as can be seen from Eq.(5.9).

One distinguishes different ways for specifying the regularization matrix $\boldsymbol{P}_\mu$. In *ordinary ridge regression*, $\boldsymbol{P}_\mu$ is set equal to the unit matrix $\boldsymbol{I}$. Another option is to specify $\boldsymbol{P}_\mu$ according to a geopotential degree variance model. To do this, one adds a well-conditioned VCM of the prior gravity field solution to the normal system (cf. Sect.4.2). If this VCM is derived from *Kaula's rule of thumb* for the signal variances (Kaula 1966)

$$\boldsymbol{P}_\mu(i,j) = \begin{cases} 10^{10} \cdot l^4 & \text{if} \quad i = j \\ 0 & \text{else,} \end{cases} \tag{5.10}$$

the resulting method is known then as Kaula stabilization.

Depending on the differential operator $\boldsymbol{L}$ in $\lambda\|\boldsymbol{L}\boldsymbol{x}\|^2$, the first or the higher orders of the derivative of the gravitational potential is minimized. At this point the reader is referred to the extensive study by Bouman (1998), Ditmar et al. (2003) and Kusche and Klees (2002) for choosing the regularization matrix in gravitation field determination.

Choice of the optimal regularization parameter

The main problem with applying regularization methods lies in the determination of a suitable regularization parameter. This parameter controls the influence of the additional information and restrictions, respectively, on the estimation process. Is often critically pointed out that on the one hand the influence of the additional information (inserted into the normal equations by virtue of the apriori parameter vector $\boldsymbol{\mu}$ according to Eq. (5.9)) is too strong when the regularization parameter is granted a value too generous. On the other hand, if the regularization part is weak within the normal equation matrix, the influence of the measurement errors is unchanged and a part of the parameter vector for the spherical harmonic coefficients becomes unusable.

There already exists a wide range of procedures for optimal determination of the regularization parameter which have been excellently summarized, for example, by Kusche (2002). In this thesis the regularization parameter is estimated by means of the variance component estimation (see algorithms described in chapter 4). Other methods for selecting regularization parameters shall be described briefly.

L-curve
L-curve based procedures are suitable for a graphical determination of the regularization parameter (Hansen 1997). The idea is to generate a double-logarithmic plot of the weighted residual sum of squares with respect to the weighted residual sum of squares of the prior information for different values of the regularization parameter. The form of the resulting curve resembles an L, and the optimal regularization parameter is read off at the corner point of this curve.

Generalized cross validation
The generalized Cross-Validation (GCV) method does not depend on a priori knowledge about the noise variance. The idea behind GCV is to find the parameter λ which minimizes the functional:

$$\lambda_{\text{gcv}} = \frac{n\|\boldsymbol{A}\boldsymbol{x}_\lambda - \boldsymbol{l}\|^2_{\boldsymbol{P}}}{\Big(\text{tr}(\boldsymbol{I} - \boldsymbol{Q}^\lambda)\Big)} \tag{5.11}$$

with $\boldsymbol{x}_\lambda$ being the solution obtained from the whole data. $\boldsymbol{Q}^\lambda$ is the so-called influence matrix (see, e.g., Kusche 2002, p. 50):

$$\boldsymbol{Q}^\lambda = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} + \lambda\boldsymbol{P}_\mu)^{-1}\boldsymbol{A}^T\boldsymbol{P}. \tag{5.12}$$

To avoid the computation of the trace in Eq. 5.11 for large-scale problems like GOCE, Kusche and Klees (2002) suggested to use the Monte Carlo trace estimator which has been used in Sect. 4.3.

5.3 The algorithm PCGMA

In the previous chapter the normal equation systems were described for different types of data. Combining the heterogeneous normal equation systems Eqs. (5.5), (5.7), and the regularization matrix yields (see, e.g., Koch 1999, p. 177):

$$\left(\frac{1}{\sigma_1^2}\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \frac{1}{\sigma_2^2}\boldsymbol{N}_{\text{sst}} + \frac{1}{\sigma_\mu^2}\boldsymbol{P}_\mu\right)\hat{\boldsymbol{x}} = \left(\frac{1}{\sigma_1^2}\bar{\boldsymbol{A}}^T\bar{\boldsymbol{l}} + \frac{1}{\sigma_2^2}\boldsymbol{b}_{\text{sst}} + \frac{1}{\sigma_\mu^2}\boldsymbol{\mu}\right). \tag{5.13}$$

After multiplying Eq. (5.13) with σ_1^2, defining

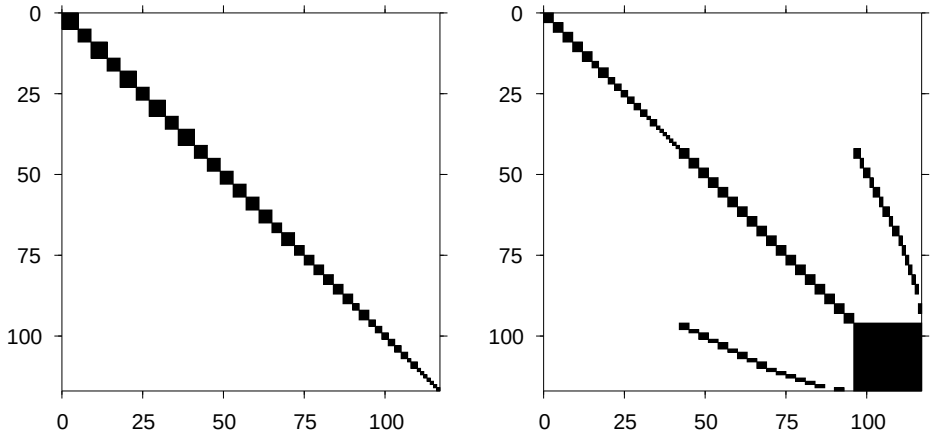
$$\omega = \frac{\sigma_1^2}{\sigma_2^2} \quad \text{as weight parameter} \quad \text{and} \quad \lambda = \frac{\sigma_1^2}{\sigma_\mu^2} \quad \text{as regularization parameter,} \tag{5.14}$$

Eq. (5.13) takes the following form:

$$\left(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \omega\boldsymbol{N}_{\text{sst}} + \lambda\boldsymbol{P}_\mu\right)\hat{\boldsymbol{x}} = \bar{\boldsymbol{A}}^T_{\text{sgg}}\bar{\boldsymbol{l}}_{\text{sgg}} + \omega\boldsymbol{b}_{\text{sst}} + \lambda\boldsymbol{\mu}. \tag{5.15}$$

Schuh (1996) proposed the procedure of preconditioned conjugate gradient multiple adjustment (PCGMA), which is based on the method of conjugate gradients with preconditioning with a modification by Schwarz (1970). This procedure allows the solving of a combined normal equation systems from uncorrelated groups of observations on the basis of both normal equations (the SST group) and observation equations (the SGG group). An essential part of the PCGMA algorithm consists in the selection of a suitable preconditioning matrix $\boldsymbol{N}_{\oplus}$. The preconditioning matrix improves the condition of the normal equation matrix, thus the convergence rate of the solver. Usually, the preconditioning matrix is specified such that it is as similar to the unknown normal equation matrix as possible, but may be computed and stored easily.

For example, a suitable preconditioning matrix for SGG data is given by the normal equation matrix with a block-diagonal structure. Colombo (1981) demonstrated that the normal equation matrix has a block-diagonal structure if the data are gridded regularly on a sphere. Although these conditions are not strictly valid for SGG data, one may assume that the normal equation matrix has a block-dominant structure. If only the diagonal blocks of the normal equation matrix are taken, then we obtain an approximation of the normal equation matrix which is a suitable preconditioner within the CG algorithm. The structure of a preconditioning matrix is depicted in Fig. 5.1(a). The blocks of the preconditioning matrix refer to those spherical harmonic coefficients that belong to one order of the series expansion. For the combined datasets (SGG+SST) an order-wise arrangement of the unknown coefficients causes the normal equation matrix to exhibit a pattern that prevents the exploitation of its sparse population in the course of solving the approximated normal equation system. To overcome this disadvantage Schuh (1996) suggested to use the kite scheme (see Fig. 5.1(b)). The original kite scheme was subject to the restrictions that the minimal degree of the series expansion had to equal 2 and that the maximal degree had to be constant for each dataset and each order. This limitation could be successfully removed with the free kite numbering scheme (Boxhammer 2006). The computational steps of the PCGMA procedure are summarized in Alg. 5.1.



(a) Block structure (SGG) (b) Kite structure

Fig. 5.1: Structure of the preconditioning matrix

Due to the large number of observations it is hardly possible to store the complete design matrix within the working memory of the computer. Therefore, it is necessary to divide the design matrix $\bar{\boldsymbol{A}}$ and the observation vector $\boldsymbol{l}$ into individual and independent blocks

$$\bar{\boldsymbol{A}}^T = \left[\bar{\boldsymbol{A}}_1^T \;\; \bar{\boldsymbol{A}}_2^T \;\; \cdots \;\; \bar{\boldsymbol{A}}_S^T\right], \qquad \bar{\boldsymbol{l}}^T = \left[\bar{\boldsymbol{l}}_1^T \;\; \bar{\boldsymbol{l}}_2^T \;\; \cdots \;\; \bar{\boldsymbol{l}}_S^T\right]. \tag{5.16}$$

The normal equation matrix $\boldsymbol{N}$ and right-hand side $\boldsymbol{n}$ of the joint normal equation system are then obtained by cumulating the products $\bar{\boldsymbol{A}}_s^T\bar{\boldsymbol{A}}_s$ and $\bar{\boldsymbol{A}}_s^T\bar{\boldsymbol{l}}_s$ for $s \in \{1, \cdots, S\}$ individual blocks. This procedure requires that the data are uncorrelated. Within the PCGMA algorithm, the decorrelation of the SGG data is performed by means of an ARMA filter (see also Chap. 3). In Alg. 5.1 the computation of the normal equation matrix $\boldsymbol{N}$ is skipped. Instead, only matrix-vector products are computed. Thus, the vector of residuals $\boldsymbol{r}^{(0)}$, which results from the SGG-, SST-, and the regularization part in the initialization step, is given by:

$$\boldsymbol{r}^{(0)} = \sum_{s=1}^{S}\left(\bar{\boldsymbol{A}}_s^T(\bar{\boldsymbol{A}}_s\boldsymbol{x}^{(0)} - \boldsymbol{\ell}_s)^{(0)}\right) + \omega\ \left(\boldsymbol{N}_{\text{sst}}\,\boldsymbol{x}^{(0)} - \boldsymbol{n}_{\text{sst}}\right) + \lambda\ \left(\boldsymbol{P}_\mu\boldsymbol{x}^{(0)} - \boldsymbol{\mu}\right), \tag{5.17}$$

where ω denotes the weight factor between the two groups of observations, and λ the regularization parameter, which may be estimated by applying the algorithms described in Chap. 4. The necessary modifications of the algorithm will be explained in the following section.

Within each iteration step of Alg. 5.1 the solution vector $\boldsymbol{x}^{(i)}$ of the preceding step is updated by the product of the absolute value of q and the search direction $\boldsymbol{\Pi}$ (step 12 of Alg. 5.1):

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + q\,\boldsymbol{\Pi}^{(i)}, \tag{5.18}$$

where $\boldsymbol{\Pi}$ defines the relaxation direction. The scaling factor q represents the distance from $\boldsymbol{x}^{(i)}$ in the direction $\boldsymbol{\Pi}^{(i)}$. The step length q in Eq. (5.18) must satisfy the condition that the square sum of the residuals is minimal along the search direction. This requires a repeated computation of the vectors $\boldsymbol{r}^{(i)}$, $\boldsymbol{\rho}^{(i)}$, $\boldsymbol{\Pi}^{(i)}$, and $\boldsymbol{h}$ within the iteration step. The residual vector $\boldsymbol{h}$ is obtained analogously to Eq. (5.17) by

$$\boldsymbol{h} = \sum_{s=1}^{S}\left(\bar{\boldsymbol{A}}_s^T(\bar{\boldsymbol{A}}_s\boldsymbol{\Pi})\right) + \omega\,\boldsymbol{N}_{\text{sst}}\,\boldsymbol{\Pi} + \lambda\,\boldsymbol{P}_\mu\,\boldsymbol{\Pi}. \tag{5.19}$$

However, when the vectors $\boldsymbol{r}^{(i)}$ in Eq. (5.17) and $\boldsymbol{h}^{(i)}$ in Eq. (5.19) are computed, the differing dimensions of the combined systems must be taken into account. The number of parameters of the SGG part is much larger than that of the SST part. In Boxhammer (2006) this problem was solved by rearranging the unknown parameters according to the free kite numbering scheme.

Algorithmus 5.1 (PCGMA)

Purpose:	To solve the equation system (5.13)	
Input:	$\bar{\boldsymbol{A}}_s$	design matrix of SGG-Data set, $s \in 1,\ldots,S$
	ℓ	observations of SGG-data set
	$\boldsymbol{N}_{\text{sst}}$	normal matrix of SST-Data set
	$\boldsymbol{b}_{\text{sst}}$	right hand side of SST-Data set
	$\boldsymbol{x}^{(0)}$	intitial solution
	ω	start value of the weight factor between the data sets
	I	number of iterations
	$\boldsymbol{N}_\oplus$	preconditioning matrix
	$\boldsymbol{P}_\mu$	regularization matrix
	λ	start value of the regularization parameter
	$\boldsymbol{\mu}$	apriori values of the parameter vector
Output:	$\boldsymbol{x}$	vector of the parameters
	$\boldsymbol{r}^T\rho$	square sum of the residuals
	$\boldsymbol{v}^T\boldsymbol{v}$	square sum of the residuals of the observations

Initialization

$$
\begin{aligned}
1.\quad & \boldsymbol{r}^{(0)} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T \underbrace{(\bar{\boldsymbol{A}}_s\boldsymbol{x}^{(0)} - \boldsymbol{\ell}_s)}_{\boldsymbol{v}_s^{(0)}}\Big) \\
& + \omega\,\Big(\boldsymbol{N}_{\mathsf{sst}}\,\boldsymbol{x}^{(0)} - \boldsymbol{n}_{\mathsf{sst}}\Big) + \lambda\,\Big(\boldsymbol{P}_\mu\boldsymbol{x}^{(0)} - \boldsymbol{\mu}\Big) \\
2.\quad & \boldsymbol{v}^{(0)} = \left[\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \ldots, \boldsymbol{v}_S^T\right]^T \\
3.\quad & \boldsymbol{\rho}^{(0)} = \mathsf{solve}(\boldsymbol{N}_\oplus, \boldsymbol{r}^{(0)}) \\
4.\quad & \boldsymbol{\Pi}^{(0)} = -\boldsymbol{\rho}^{(0)}
\end{aligned}
$$

Iteration steps $i = 0, 1, \ldots, I$

$$
\left.
\begin{aligned}
5.\quad & e = \frac{\boldsymbol{r}^{(i)T}\boldsymbol{\rho}^{(i)}}{\boldsymbol{r}^{(i-1)T}\boldsymbol{\rho}^{(i-1)}} \\
6.\quad & \boldsymbol{\Pi}^{(i)} = -\boldsymbol{\rho}^{(i)} + e\,\boldsymbol{\Pi}^{(i-1)}
\end{aligned}
\right\} i > 0
$$

$$
\begin{aligned}
7.\quad & \boldsymbol{h} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T \underbrace{(\bar{\boldsymbol{A}}_s\boldsymbol{\Pi})}_{\boldsymbol{g}_s}\Big) + \omega\,\boldsymbol{N}_{\mathsf{sst}}\,\boldsymbol{\Pi} + \lambda\,\boldsymbol{P}_\mu\,\boldsymbol{\Pi} \\
8.\quad & \boldsymbol{g}^{(i)} = \left(\boldsymbol{g}_1^T, \boldsymbol{g}_2^T, \ldots, \boldsymbol{g}_S^T\right)^T \\
9.\quad & q = \frac{\boldsymbol{r}^{(i)T}\boldsymbol{\rho}^i}{\boldsymbol{\Pi}^{(i)T}\boldsymbol{h}} \\
10.\quad & \boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + q\,\boldsymbol{\Pi}^{(i)} \\
11.\quad & \boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} + q\,\boldsymbol{h} \\
12.\quad & \boldsymbol{\rho}^{(i+1)} = \mathsf{solve}(\boldsymbol{N}_\oplus, \boldsymbol{r}^{(i+1)}) \\
13.\quad & \boldsymbol{v}^{(i+1)} = \boldsymbol{v}^{(i)} + q\,\boldsymbol{g}^{(i)}
\end{aligned}
$$

Where

$\boldsymbol{r}$	...	is the $n \times 1$ vector of residuals of the normal equations
$\boldsymbol{v}$	...	is the $n \times 1$ vector of the residuals of the SGG observations
$\boldsymbol{\Pi}$	...	is the $m \times 1$ vector of the relaxation direction in the the preconditioning system
$\boldsymbol{\rho}$	...	is the $m \times 1$ vector of the residuals direction in the the preconditioning system
$\boldsymbol{h}$	...	is the $m \times 1$ vector of auxiliary parameters
e, q	...	are the 1×1 scaling factors to determine the relaxation direction

5.4 Integration of Monte Carlo methods into PCGMA

In addition to the parameters, the knowledge about the quality of the parameter estimates is of high relevance. Information regarding the quality is usually provided from the computation of VCM of the coefficients determined within the adjustment process. This VCM may then be used to obtain the accuracies of quantities (such as geoid heights and geoid anomalies) that are derived from particular coefficients via error propagation. In Sect. 5.4.1 an extension of Alg. 5.1 by algorithmic steps for computing the full VCM of the parameters (as described in Chap. 3) will be discussed. Beside the estimation of the spherical harmonic coefficients $\boldsymbol{x}$, the estimation of ω and λ shall be accomplished within Alg. 5.1. Section 5.4.2 deals with the question which modifications of Alg. 5.1 are necessary to allow an estimation of these variance components.

5.4.1 Integration of the MCVCM algorithm

The computation of the VCM of the spherical harmonic coefficients is traditionally carried out by inversion of the combined normal equation matrix Eq. (3.31):

$$\Sigma\left\{\widehat{\mathfrak{x}}\right\} = \sigma^2\boldsymbol{N}^{-1}. \tag{5.20}$$

As the computation of the inverse itself is avoided within the PCGMA algorithm, the inverse of a sparse preconditioning matrix (see Sect. 5.3) is used as an approximation of the VCM. The inverting of a sparse preconditioning matrix requires very little computational and time effort. Abwerzger (1999) showed that such an approximation of the exact VCM of the parameters is sufficiently accurate.

As an alternative to using sparse matrices we will now discuss a method for computing the VCM by means of algorithms explained in Sect. 3 (see also Alkhatib and Schuh 2007). Figure 5.2 illustrates the computational steps necessary for estimating the VCM of a combined equation system (consisting of SST normal equations and SGG observation equations) by means of the PCGMA algorithm.

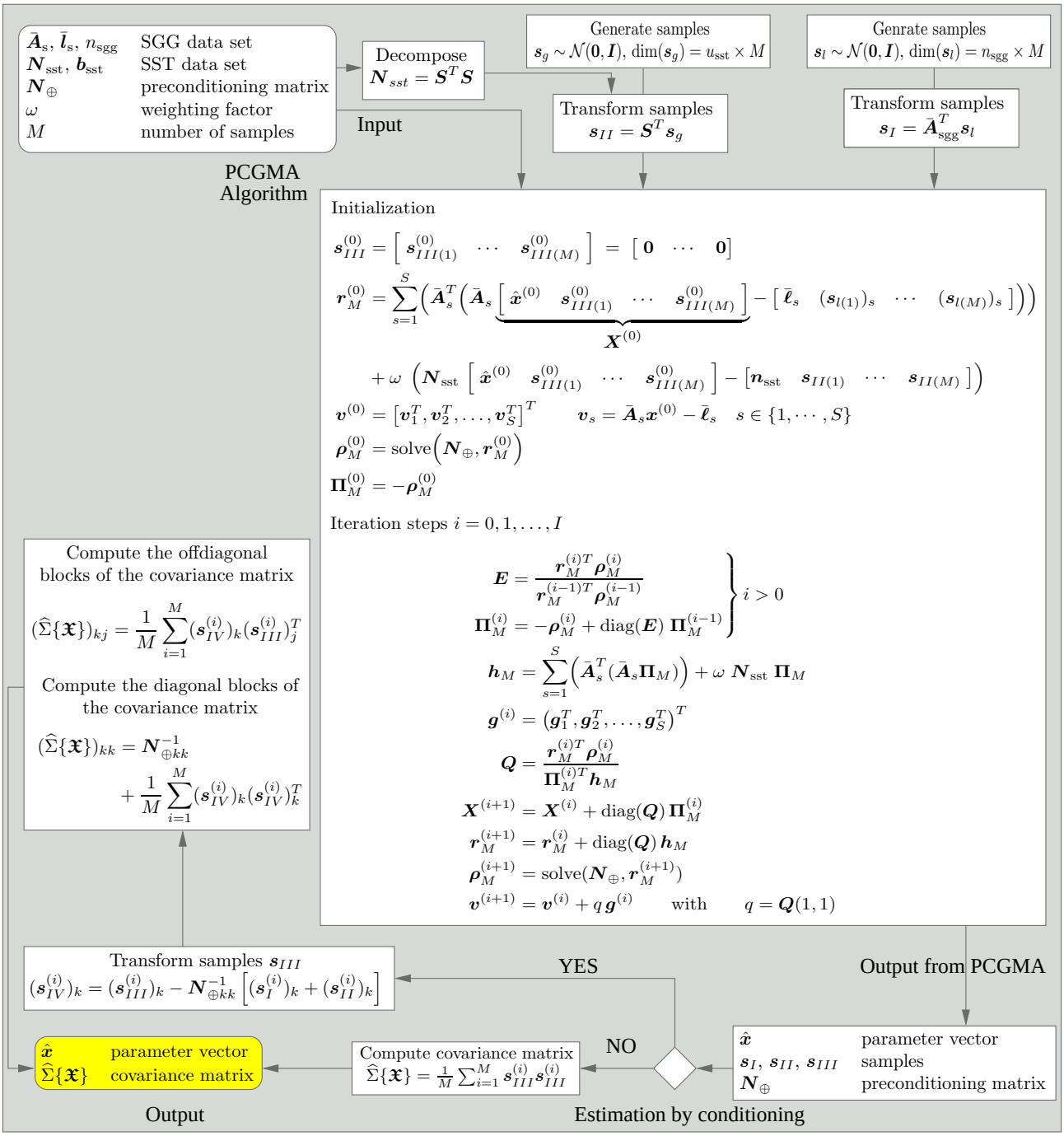


Fig. 5.2: Interation of the VCMC algorithm

For a clearer presentation of the extension of the Alg. 5.1 by Alg. 3.9 (discussed in Sect. 3.5.2), the regularization

terms in Eq. (5.13) in the current section will be neglected.

Comment: If the VCM of the parameters should be computed for the case of a combined equation system comprising a regularization matrix, then the regularization matrix (denoted by $\boldsymbol{P}_\mu$) is taken into account as an additional normal equation matrix within Alg. 3.9.

First, a specific number of samples is generated for each given dataset from standard-normal distributed random vectors (see Sect. 2.3.3). The groups of generated samples differ only by their dimension: for the SGG dataset $\boldsymbol{s}_l \in \mathbb{R}^{n\times M}$ samples are generated, and for the SST dataset $\boldsymbol{s}_g \in \mathbb{R}^{m\times M}$ samples are produced, where n denotes the length of the SGG observation vector and m the total number of parameters. In a second step, the generated samples $\boldsymbol{s}_g$ and $\boldsymbol{s}_l$ are transformed. The transformation steps involved in this operation have been explained in Sect. 3.5.2 and then incorporated into Alg. 3.9. For simplicity the transformed samples will be denoted as $\boldsymbol{s}_I$ and $\boldsymbol{s}_{II}$, respectively.

The combined equation system is solved by the PCGMA algorithm for additional multiple M right-hand sides, consisting of the summation of the $\boldsymbol{s}_I$ and $\boldsymbol{s}_{II}$ (Fig. 5.3).

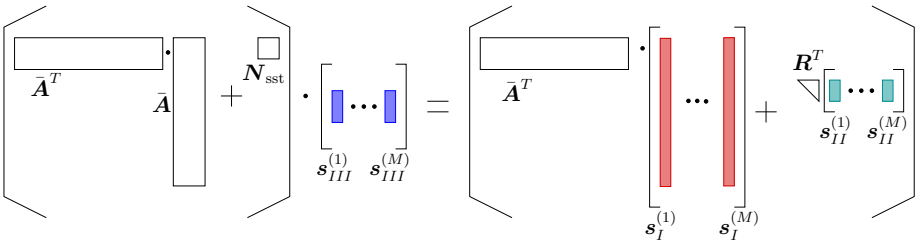


Fig. 5.3: Generation of MCVCM-samples

Beside the estimated parameters also the $\boldsymbol{s}_{III}$ must be provided by PCGMA as output for computing the VCM of the parameters. To achieve this, the algorithm 5.1 must be modified. As for the modified version of the PCGMA algorithm in the previous section, a number of changes are necessary now. To begin with, the initial values of the residual vectors $\boldsymbol{r}^{(0)j}$ in Eq. (5.17) are substituted by the following matrix $\boldsymbol{r}_M^{(0)j} \in \mathbb{R}^{m\times M+1}$:

$$\begin{aligned}\boldsymbol{r}_M^{(0)} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\Big(\bar{\boldsymbol{A}}_s \underbrace{\left[\hat{\boldsymbol{x}}^{(0)}\ \boldsymbol{s}_{III(1)}^{(0)}\ \cdots\ \boldsymbol{s}_{III(M)}^{(0)}\right]}_{\boldsymbol{X}^{(0)}} - \left[\bar{\boldsymbol{\ell}}_s\ (\boldsymbol{s}_{l(1)})_s\ \cdots\ (\boldsymbol{s}_{l(M)})_s\right]\Big)\Big) \\ + \omega\ \Big(\boldsymbol{N}_{\text{sst}}\left[\hat{\boldsymbol{x}}^{(0)}\ \boldsymbol{s}_{III(1)}^{(0)}\ \cdots\ \boldsymbol{s}_{III(M)}^{(0)}\right] - \left[\boldsymbol{n}_{\text{sst}}\ \boldsymbol{s}_{II(1)}\ \cdots\ \boldsymbol{s}_{II(M)}\right]\Big).\end{aligned} \tag{5.21}$$

Thus, the parameter matrix $\boldsymbol{X} \in \mathbb{R}^{m\times M+1}$ is jointly constructed from the parameter vector $\boldsymbol{x}$ and $\boldsymbol{s}_{III} \in \mathbb{R}^{m\times M}$. The computation of the matrix $\boldsymbol{X}^{(i+1)}$ with respect to the new iteration step $i+1$ is performed according to Eq. (5.27). This computation step is described by Fig. 5.4.

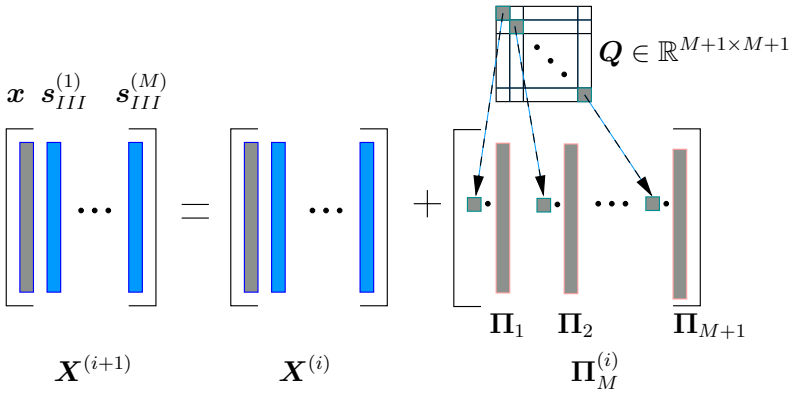


Fig. 5.4: Update of the paramete rmatrix $\boldsymbol{X}$ to generate $\boldsymbol{s}_{III}$

The computation of the matrix $\boldsymbol{Q} \in \mathbb{R}^{M+1 \times M+1}$ in Fig. 5.4 is done by means of the following equation:

$$\boldsymbol{Q} = \frac{\boldsymbol{r}_M^T \boldsymbol{\rho}_M}{\boldsymbol{\Pi}_M^T \boldsymbol{h}_M} \tag{5.22}$$

If the number M of the generated samples is large, then the assembling of this matrix within each iteration is computationally expensive. However, only the diagonal elements of the matrix $\boldsymbol{Q}$ are required, from which the update values for determining the search direction $q_1, \cdots, q_{M+1}$ are inferred. The computation of the diagonal elements of $\boldsymbol{Q}$ is explained by Fig. 5.5. Only the product of the i^{th} row of the matrix $\boldsymbol{r}_M^T$ with the corresponding i^{th} row of the matrix $\boldsymbol{\rho}_M$ needs to be computed, and this scalar product is divided by the product which results from multiplying the i^{th} rows of the matrices $\boldsymbol{\Pi}_M^T$ and $\boldsymbol{h}_M$. Fig. 5.5 illustrates the computation of the diagonal elements of $\boldsymbol{Q}$, which fixes the i^{th} search direction q_i. The computation of the matrix $\boldsymbol{E}$ within the modified

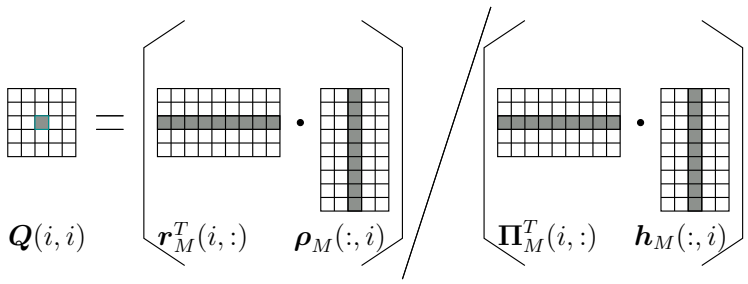


Fig. 5.5: Computation of the matrix $\boldsymbol{Q}$

PCGMA algorithm is carried out analogously to the computation of the matrix $\boldsymbol{Q}$ (see Fig. 5.2).

After I iterations the PCGMA algorithm achieved its convergence, then one obtains as output the parameter matrix $\boldsymbol{X}$, which comprises both the estimated spherical harmonic coefficients $\boldsymbol{x}$ and the M samples $\boldsymbol{s}_{III}$.

Computation of the dyadic products of the samples $\boldsymbol{s}_{III}$ (cf. Alg. 3.9 in Sect. 3.5.2) results in a quick but coarse estimation of the VCM of the parameters $\widehat{\Sigma}\{\boldsymbol{\mathfrak{x}}\}$. The more samples are generated, the more accurate this estimation becomes (compare with 3.73 in Sect. 3.4). Thus, a more accurate estimation of $\widehat{\Sigma}\{\boldsymbol{\mathfrak{x}}\}$ necessitates a higher number of samples, which in turn requires additional memory and arithmetic operations.

To keep the number of samples as small as possible the variance reduction technique *estimation by conditioning* was applied (see Sect. 3.5.2). This requires first that the $\boldsymbol{s}_{III}$ are transformed into $\boldsymbol{s}_{IV}$ as demonstrated by Fig. 5.6.

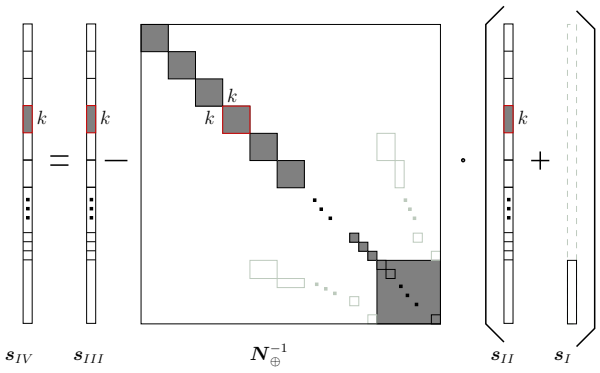


Fig. 5.6: Computation of the samples $\boldsymbol{s}_{IV}$

It can be seen from Fig. 5.6 that the computation of the transformed samples $\boldsymbol{s}_{IV}$ requires, beside the samples $\boldsymbol{s}_I$ and $\boldsymbol{s}_{II}$, the inverse of the main diagonal blocks of the kite matrix $\boldsymbol{N}_{\oplus}$. The samples $\boldsymbol{s}_I$ may be computed outside the PCGMA algorithm since the SST normal equation matrix is available as an external product (cf. Sect. 5.2.2). In contrast to these samples, the samples $\boldsymbol{s}_{II}$ must be obtained during the processing of the

SGG data. The $\boldsymbol{s}_{II}$ are then obtained from the initialization step for the SGG part in Eq. (5.21) as:

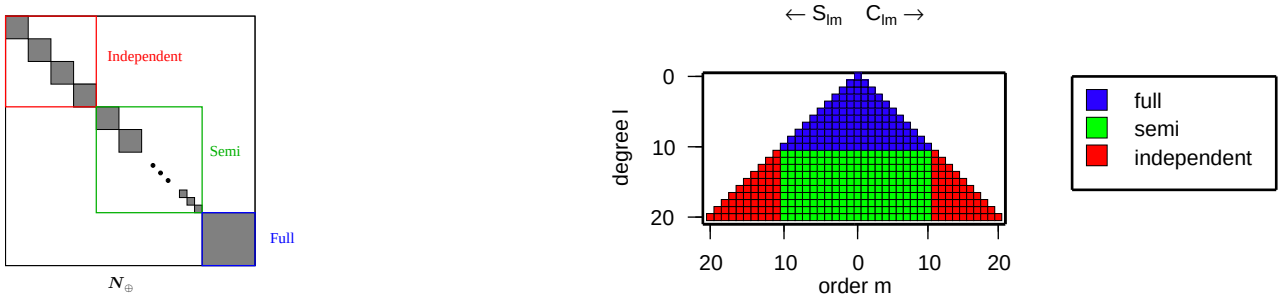
$$\left[\boldsymbol{r}_M^{(0)}\right]_{\mathrm{SGG}} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\Big(\underbrace{\bar{\boldsymbol{A}}_s\left[\,\hat{\boldsymbol{x}}^{(0)}\;\; \boldsymbol{s}_{III(1)}^{(0)}\;\cdots\;\boldsymbol{s}_{III(M)}^{(0)}\right]\boldsymbol{X}^{(0)}}_{\mathbf{0}} - \left[\,\bar{\boldsymbol{\ell}}_s\;\; (\boldsymbol{s}_{l(1)})_s\;\cdots\;(\boldsymbol{s}_{l(M)})_s\right]\Big)\Big). \tag{5.23}$$

Using null vectors as initial values for the parameter matrix $\boldsymbol{X}_M^{(0)}$ and multiplying the right-hand side of Eq. (5.23) by -1 yields:

$$\boldsymbol{s}_{II} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\left[\,(\boldsymbol{s}_{l(1)})_s\;\cdots\;(\boldsymbol{s}_{l(M)})_s\right]\Big). \tag{5.24}$$

Due to the different dimensions of the samples $\boldsymbol{s}_{II}$ and $\boldsymbol{s}_I$ resulting from PCGMA (see Fig. 5.6) the $\boldsymbol{s}_I$ are added to the last block of the samples $\boldsymbol{s}_{II}$. The dimension of this last block is equal to the dimension of the SST normal equation matrix. The samples $\boldsymbol{s}_{II}$ and $\boldsymbol{s}_{III}$ are divided into blocks. The partitioning of the samples is performed according to the dimension of the main diagonal blocks of the kite matrix. The kite in Fig. 5.6 is set up within the PCGMA algorithm in order to speed up the convergence of the PCGMA method (see also Sect. 5.3).

The following example will make this division of the samples clear. If SGG data resolved up to degree 20 are combined with SST data up to degree 10, then the blocks of the kite matrix are divided into three zones order by order (Boxhammer 2006). These zones are displayed in Fig. 5.7 only for the main diagonal blocks.



(a) kite matrix (b) Parameter triangles

Fig. 5.7: The partitioning of the kite-matrix and corresponding parameter triangles

The order of the parameters starts with the parameters of the *independent* zone. The blocks inside this zone are determined by SGG data only, and they take orders or degrees that are higher than the maximal resolution degree of the SST data:

$$\text{Zone Independent} = \begin{cases} m = 11:20 & C_{lm} = \{C_{l_{\mathrm{odd}}m}, C_{l_{\mathrm{even}}m}\} \qquad l \in \{11, 12, \cdots 20\} \\ m = 11:20 & S_{lm} = \{S_{l_{\mathrm{odd}}m}, S_{l_{\mathrm{even}}m}\} \qquad l \in \{11, 12, \cdots 20\}. \end{cases}$$

The *independent* zone is followed by the *semi* zone, which corresponds to parameters that have the same order as the SST dataset but higher degrees:

$$\text{Zone Semi} = \begin{cases} m = 0:10 & C_{lm} = \{C_{l_{\mathrm{odd}}m}, C_{l_{\mathrm{even}}m}\} \qquad l \in \{11, 12, \cdots 20\} \\ m = 1:10 & S_{lm} = \{S_{l_{\mathrm{odd}}m}, S_{l_{\mathrm{even}}m}\} \qquad l \in \{11, 12, \cdots 20\}. \end{cases}$$

The end of the kite matrix is occupied by the *full* zone for which the parameters are identical to the parameters of the SST data set:

$$\text{Zone Full} = \begin{cases} m = 0:10 & C_{lm} = \{C_{l_{\mathrm{odd}}m}, C_{l_{\mathrm{even}}m}\} \qquad l \in \{2, 3, \cdots 10\} \\ m = 1:10 & S_{lm} = \{S_{l_{\mathrm{odd}}m}, S_{l_{\mathrm{even}}m}\} \qquad l \in \{2, 3, \cdots 10\}. \end{cases}$$

The alternative estimation of the VCM consists then of two parts: the estimation of the main and the secondary diagonal blocks. The computational steps were explained in detail by Alg. 3.9, and visualized by Fig. 5.8. The

computation of the main diagonal blocks are showed at the top of Fig. 5.8 and the secondary diagonal blocks at the bottom. Suppose that the k^{th} main diagonal block of the VCM $(\widehat{\Sigma}\{\boldsymbol{\mathfrak{x}}\})_{kk}$ shall be computed. For this purpose, the dyadic product of the k^{th} block of $\boldsymbol{s}_{III}$ is computed, and the inverted main diagonal block of the kite matrix $(\boldsymbol{N}_{\oplus}^{-1})_{kk}$ is added to this. To obtain the j^{th} secondary diagonal block of the VCM $(\widehat{\Sigma}\{\boldsymbol{\mathfrak{x}}\})_{kj}$, the dyadic product of the k-th block of the samples $\boldsymbol{s}_{IV}$ with the k^{th} block of the samples $\boldsymbol{s}_{III}$ is computed.

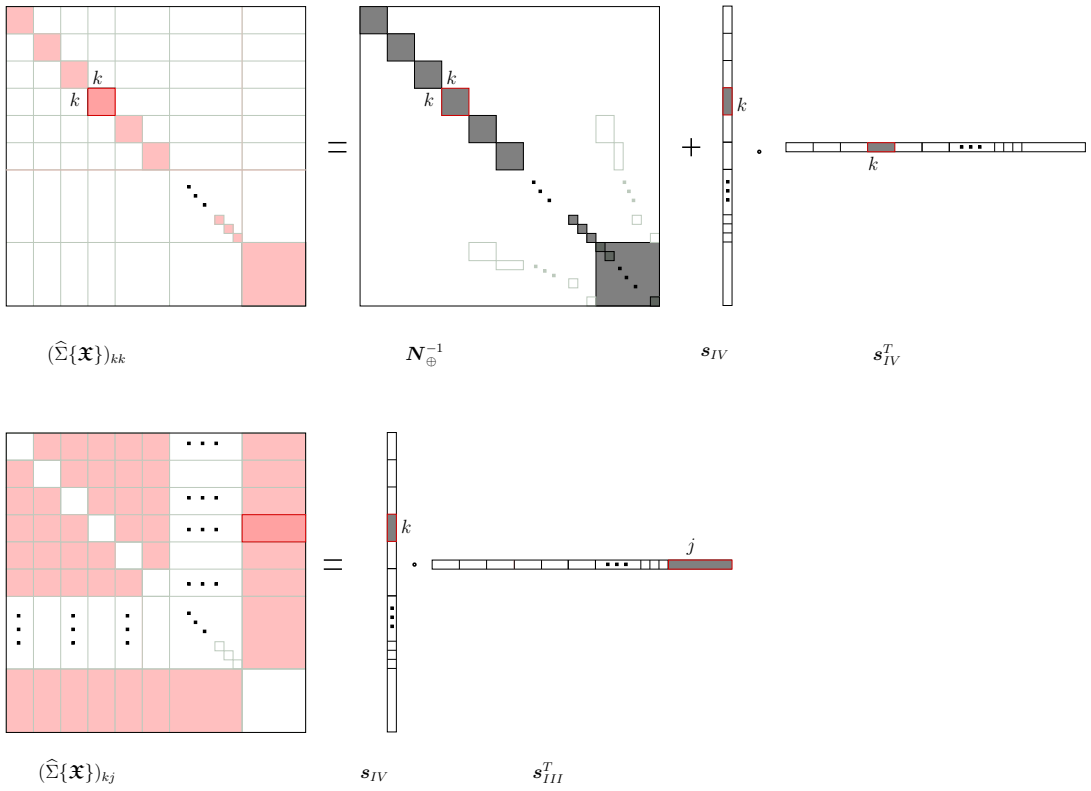


Fig. 5.8: Estimation of the VCM by means of *estimation by conditioning*

A great advantage of this procedure is that an accurate representation of the VCM of the parameters is achieved by using only a relatively small number of generated samples. This was demonstrated by a simulation in Sect. 6.1. However, a condition for this to be true is that the normal equation matrix has a dominating block diagonal structure. This condition is satisfied by GOCE data.

5.4.2 Integration of the MCVCE algorithm

The computation of the variance components $\hat{\boldsymbol{\sigma}} = \{\sigma_1^2, \sigma_2^2, \sigma_\mu^2\}$ is done by means of Alg. 4.3 from Sect. 4.3.2. It requires the solving of the combined normal equation system for multiple right-hand sides (see step 3 of Alg. 4.3):

$$\boldsymbol{N}^{(k)} \left(\hat{\boldsymbol{x}}^{(k)} \;\; \hat{\boldsymbol{\alpha}}^{(k)} \;\; \hat{\boldsymbol{\beta}}^{(k)} \right) = \left(\boldsymbol{b}^{(k)} \;\; \bar{\boldsymbol{A}}^T \boldsymbol{z} \;\; \boldsymbol{w} \right). \tag{5.25}$$

The number of additional right-hand sides in Eq. (5.25) is determined by the number of types of observations (in our case 2). For groups of observations that are available through corresponding observation equations, auxiliary parameter vectors $\boldsymbol{\alpha}$ are estimated in addition. The corresponding right-hand sides are given by the product $\boldsymbol{A}^T\boldsymbol{z}$ with $\boldsymbol{z} \sim \mathcal{U}$ from the distribution in Sect. 4.3.1. For data types that are given as normal equation systems, the auxiliary parameters $\boldsymbol{\beta}$ are estimated. The right-hand side is given by the vector $\boldsymbol{w}$, which follows the same distribution $\mathcal{U}$ as the random vector $\boldsymbol{z}$, but for which the parameter vector has dimension $m \times 1$.

Now Alg. 5.1 shall be modified to allow estimation of both the parameter vector $\boldsymbol{x}$ and the additional auxiliary parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in Eq. (5.25). For this purpose, the residual vector $\boldsymbol{r}_i$ is extended to the matrix $\boldsymbol{r}_M^{(0)j} \in \mathbb{R}^{m \times 3}$ within the initialization step:

$$\begin{aligned} \boldsymbol{r}_M^{(0)} = &\sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\Big(\bar{\boldsymbol{A}}_s \underbrace{\left[\hat{\boldsymbol{x}}^{(0)}\ \hat{\boldsymbol{\alpha}}^{(0)}\ \hat{\boldsymbol{\beta}}^{(0)}\right]}_{\boldsymbol{X}^{(0)}} - \left[\bar{\boldsymbol{\ell}}_s\ \boldsymbol{z}_s\ \boldsymbol{0}\right]\Big)\Big) \\ &+\omega\ \Big(\boldsymbol{N}_{\text{sst}}\left[\hat{\boldsymbol{x}}^{(0)}\ \hat{\boldsymbol{\alpha}}^{(0)}\ \hat{\boldsymbol{\beta}}^{(0)}\right] - \left[\boldsymbol{n}_{\text{sst}}\ \boldsymbol{0}\ \boldsymbol{w}\right]\Big) + \lambda\ \Big(\boldsymbol{P}_\mu\left[\hat{\boldsymbol{x}}^{(0)}\ \hat{\boldsymbol{\alpha}}^{(0)}\ \hat{\boldsymbol{\beta}}^{(0)}\right] - \left[\boldsymbol{\mu}\ \boldsymbol{0}\ \boldsymbol{0}\right]\Big). \end{aligned} \tag{5.26}$$

It is seen from Eq. (5.26) that the $m \times 3$ parameter matrix $\boldsymbol{X}$ consists of the individual parameter vectors $\boldsymbol{x}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The parameter matrix $\boldsymbol{X}^{(i+1)}$ of the new iteration step is computed in analogy to Eq. (5.18) by:

$$\boldsymbol{X}^{(i+1)} = \boldsymbol{X}^{(i)} + \text{diag}(\boldsymbol{Q})\,\boldsymbol{\Pi}_M^{(i)}. \tag{5.27}$$

In contrast to Eq. (5.18), $\boldsymbol{Q}$ in Eq. (5.27) is not a scalar anymore, but a 3×3 matrix, in which only its diagonal elements $\boldsymbol{Q}(k,k)$ (with $k = 1:3$) must be substituted for computing $\boldsymbol{X}^{(i+1)}$. Fig. 5.9 illustrates this computation step.

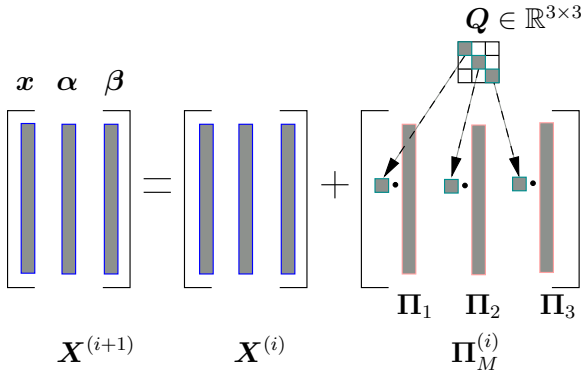


Fig. 5.9: Update of the parameter matrix $\boldsymbol{X}$ to estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

The vectors $\boldsymbol{h}$, $\boldsymbol{\rho}$, and $\boldsymbol{\Pi}$ in Alg. 5.1 are replaced by the matrices $\boldsymbol{h}_M$, $\boldsymbol{\rho}_M$, and $\boldsymbol{r}_M \in \mathbb{R}^{m\times3}$, whose dimensions are determined by the dimension of the parameter matrix $\boldsymbol{X}$, analogously to the matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$.
The modified PCGMA algorithm together with the MCVCE algorithm (Alg. 4.3) is displayed in Fig. 5.10, from which it is seen that two iterative procedures must be combined.

After successful convergence of the modified PCGMA algorithm one obtains the estimated parameter matrix $\boldsymbol{X}_i$ as the output, which is used in turn to compute update values for the variance components $\hat{\boldsymbol{\sigma}}$, and thus for the weight factor ω and the regularization parameter λ. To compute the partial redundancy of the SGG-part r_{sgg} (see Fig. 5.10), we need the right hand side $\bar{\boldsymbol{A}}^T\boldsymbol{z}$ which must be computed during the processing of the SGG data within PCGMA . The $\bar{\boldsymbol{A}}^T\boldsymbol{z}$ are obtained from the initialization step (regarding SGG observations in 5.26) as:

$$(\boldsymbol{r}_M^{(0)})_{r_{\text{sgg}}} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\Big(\bar{\boldsymbol{A}}_s \underbrace{\left[\hat{\boldsymbol{x}}^{(0)}\ \hat{\boldsymbol{\alpha}}^{(0)}\ \hat{\boldsymbol{\beta}}^{(0)}\right]}_{\boldsymbol{X}^{(0)}} - \left[\bar{\boldsymbol{\ell}}_s\ \boldsymbol{z}_s\ \boldsymbol{0}\right]\Big)\Big) \tag{5.28}$$

Using null vectors as initial values for the parameter matrix $\boldsymbol{X}^{(0)}$ and multiplying the right-hand side of Eq. (5.28) by -1 yields:

$$\bar{\boldsymbol{A}}^T\boldsymbol{z} = \sum_{s=1}^{S}\Big(\bar{\boldsymbol{A}}_s^T\boldsymbol{z}_s\Big). \tag{5.29}$$

Then the modified PCGMA algorithm is restarted again using the computed update value for ω and λ. Both iteration procedures are repeated until the estimates for the variance components do not change significantly anymore.

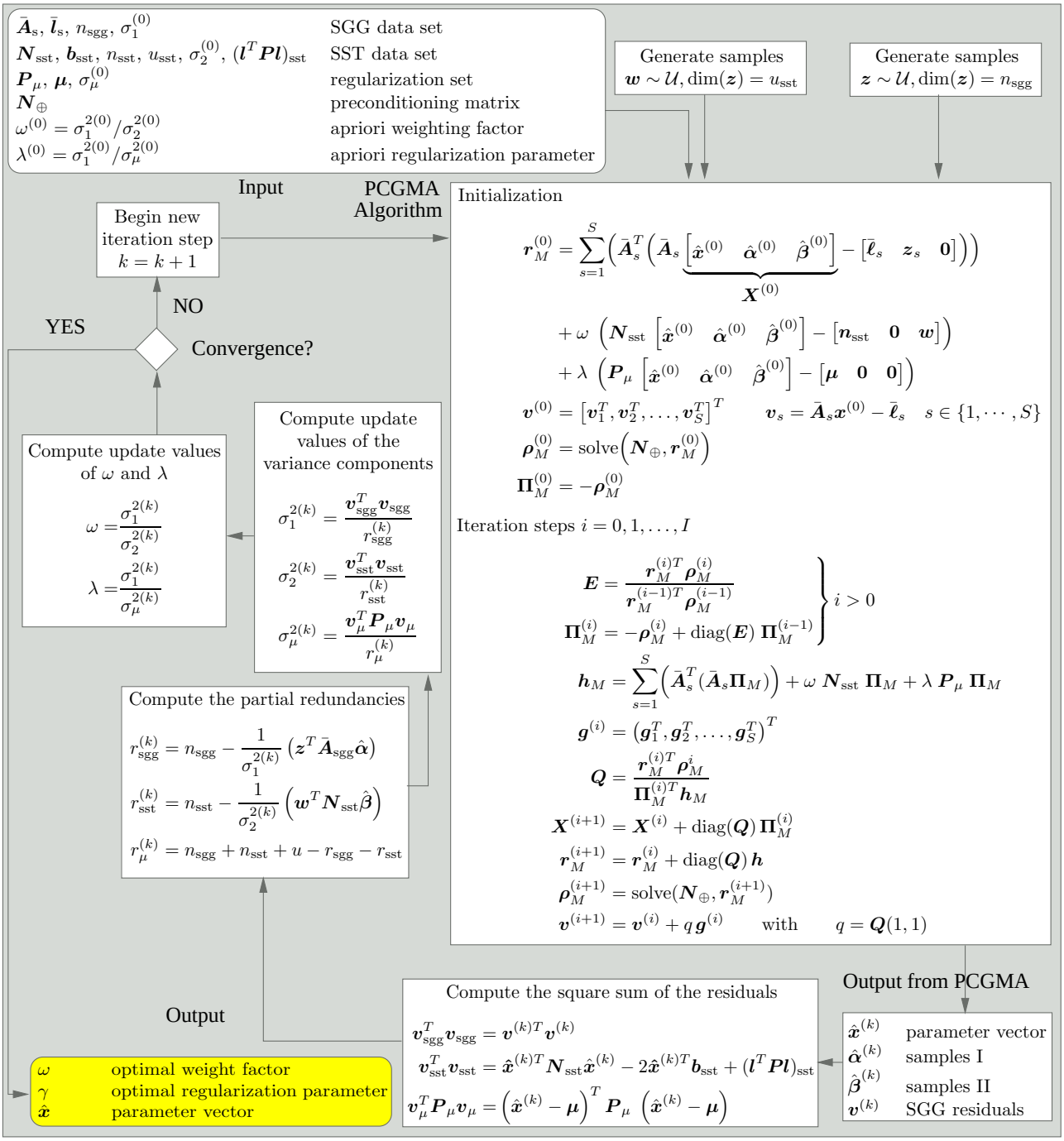


Fig. 5.10: Integration of the MCVCE-algorithm

6 Application of the Monte Carlo integration to simulated GOCE data

6.1 Scenario 1: Estimation of the variance/covariance information

6.1.1 Data sets

To demonstrate the capability of the developed algorithms in chapter 3, the results of numerical experiments using simulated GOCE data sets will be presented. The first data set was generated following the baseline of the GOCE mission ESA 1999. It consists of 1,480,000 simulated gradiometer measurements along the orbit of a satellite at 250 km altitude during a period of 23 days. The three diagonal components V_{xx}, V_{yy} and V_{zz} of the gravity tensor were used. The data interval was fixed at 4sec. These simulated observations are typically corrupted by coloured noise, which is generated by a power spectral density model with a white noise behavior of $3mE/\sqrt{Hz}$ in the measurement bandwidth between $0.005 - 0.1$ Hz and an $1/f$ behaviour in the low frequency range. Orbit and gradiometry data were derived from the EGM96 (Earth Gravitational Model, Lemoine et al. 1998) gravity model complete up to degree and order 60.
The second data set consists of an SST normal equation matrix, which resulted from a simulation of joint state vector and low-degree (to degree 60) gravity field estimation for the GPS-tracked, low-orbiting GOCE satellite. This normal equation matrix was simulated from a 30-day GOCE orbit solution from precise orbit determination (POD). The (x, y, z)-data covariance matrix was assumed to be a diagonal matrix with $\mathrm{diag}(\Sigma) = 1.75$ cm. The mathematical model is described in Mayer-Gürr et al. 2005.

6.1.2 Test simulation

The data sets were used to compute the numerical simulations. In the first part of these simulations, Algs. 3.5 to 3.7 were used to estimate the VCM by Monte Carlo integration in a straightforward way. Algorithm 3.5 was applied to compute the SGG-only solution directly from the observation equations, whereas the direct solution with the SST-normal equations was obtained by Alg. 3.6. For the combined SST/SGG solution Alg. 3.7 was used. In all simulations, it was aimed at guaranteeing at least one significant digit of the estimated variance information, which means that the relative error is to be bounded by

$$\frac{\hat{\sigma}^2_{\mathbf{x}_j} - \sigma^2_{\mathbf{x}_j}}{\sigma^2_{\mathbf{x}_j}} \leq 0.1 \; . \tag{6.1}$$

Applying Eq. (3.72) the number of necessary samples is given by

$$K^{N(0,1)}_{1-\alpha/2}\sqrt{\frac{2}{M}} \leq 0.1 \; . \tag{6.2}$$

Introducing the error probability $\alpha = 5\%$, the quantile $K^{N(0,1)}_{1-\alpha/2}$ is fixed at 1.96. The efficient number of samples M can be determined to be equal to 769. Therefore, 800 samples should guarantee one significant digit of the estimated variances. Numerous simulations were conducted to verify these statements, with special attention to the particular numerical characteristics of the given problem. According to theoretical considerations presented above, we proceed in the following way:

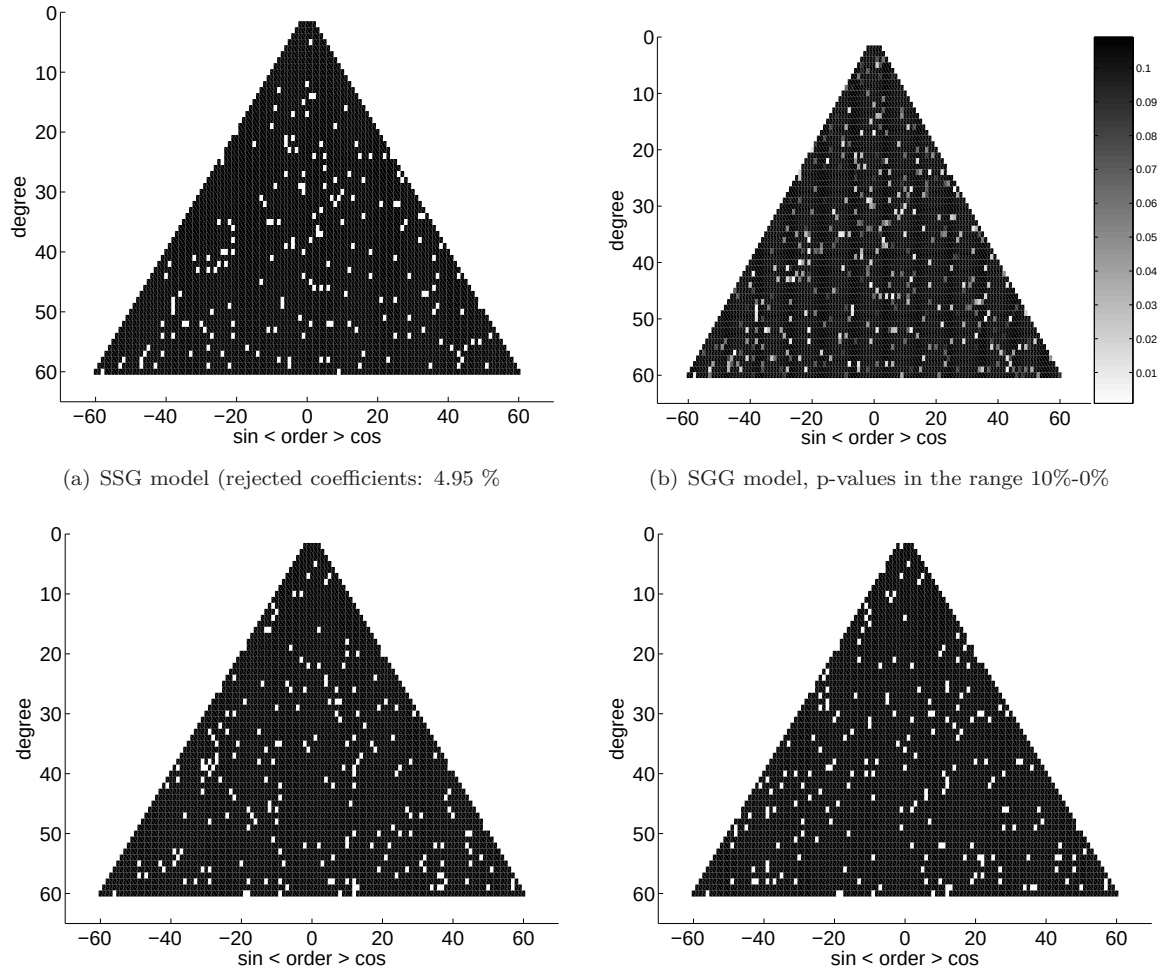
- We start by demonstrating the performance of the MC estimation of the coefficient's variances. We will provide an empirical proof of the error bound equation (Eq. 6.1).
- In a second step, we give a detailed analysis of the MC estimation process for two single coefficients. In addition to analyzing the numerical deviations of estimated variances from their true values, the distribution of the samples will be investigated and compared with its theoretical counterpart, the normal distribution. For this purpose, we generate 1000 MC estimates, each of which based on 800 samples.
- Our final investigation examines the performance of propagated quantities such as geoid height anomalies.

As a first example, Fig. 6.1 demonstrates the results of the hypothesis test of an SGG-only simulation with respect to the test value defined by Eq. (3.73):

$$H_0 : \hat{\sigma}^2_{\mathfrak{x}_j} = \sigma^2_{\mathfrak{x}_j}, \quad H_A : \hat{\sigma}^2_{\mathfrak{x}_j} \neq \sigma^2_{\mathfrak{x}_j}, \quad \text{with error probability:} \quad \alpha = 5\%. \tag{6.3}$$

where $\hat{\sigma}^2_{\mathfrak{x}_j}$ denotes the estimated variance of Monte Carlo integration and $\sigma^2_{\mathfrak{x}_j}$ stand for the determined variance of the rigorous inverse of the normal equation matrix (cf. Eq. 5.20)

Figure 6.1(a) shows the results of hypothesis (white – rejected null hypothesis, black – not rejected), and in Fig. 6.1(b) the corresponding p-values of the same sample are displayed. The p-value is defined as the probability contained in the tails of the distribution (under H_0) outside the observed test value T. Figure 6.1(b) focuses on the critical error probability region between 0% and 10%.



(a) SSG model (rejected coefficients: 4.95 %

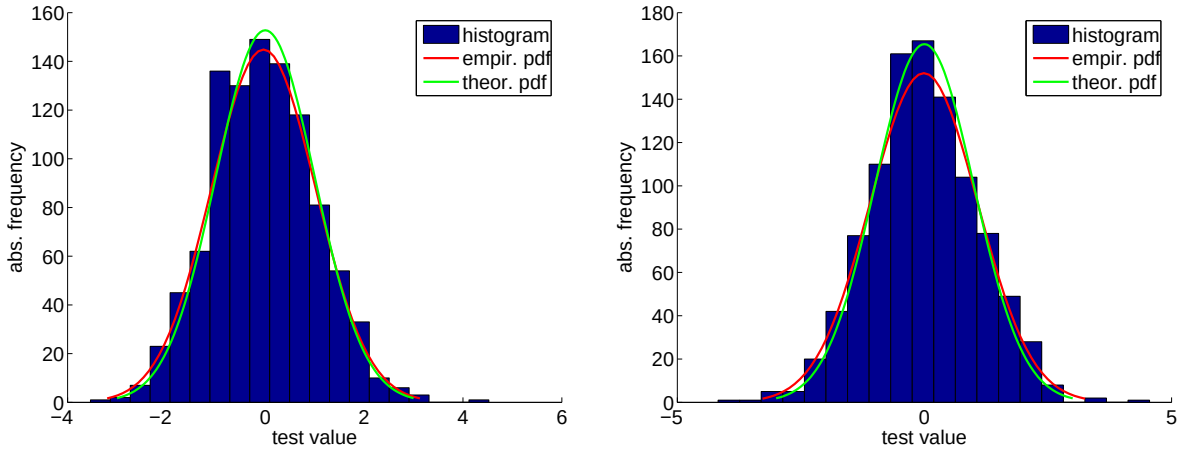
(b) SGG model, p-values in the range 10%-0%

(c) SST model (rejected coefficients: 5.27 %

(d) combined model, rejected coefficients: 5.38 %

Fig. 6.1: Accuracy of the MC estimation process using 800 samples; Hypotheses test based on Eq. (3.73).

This computation demonstrates that no clustering of rejected values occurs. It should also be mentioned that each MC realization with a different set of 800 samples reflects another pattern of rejected elements. In our current example, we have a rejection rate of 4.95% samples. The results of a purely SST solution and a combined SST/SGG are visualized by Figs. 6.1c and 6.1(d), reflect the random behavior of the rejected coefficients.
To underpin the overall performance, 1000 MC estimates were computed, each of wich were computed by MC integration with 800 samples. A histogram plot of the 1000 test values $T = \frac{\hat{\sigma}^2_{\mathfrak{x}_j} - \sigma^2_{\mathfrak{x}_j}}{\sigma^2_{\mathfrak{x}_j}} \sqrt{\frac{M}{2}}$, estimated using Eq. (3.73), is given in Fig. 6.2. In this test we look especially at the SGG-only solution, because in this computation we expect weak values for coefficients of low degree and order due to the band-limited behavior of the SGG measurements. Figure 2a shows the histogram with respect to a representative example of a weakly determined zonal low degree coefficient ($C_{10,0}$), 6.2(b) correspondingly with respect to a well-determined coefficient ($C_{47,30}$).



(a) Coefficient: $C_{10,0}$ mean -0.035, variance 1.11, skewness 0.14, kurtosis 3.09; Kolmogorov-Smirnov-test: p-value 0.04

(b) Coefficient: $C_{47,30}$ mean -0.011, variance 1.18, skewness 0.05, kurtosis 3.33; Kolmogorov-Smirnov-test: p-value 0.48:

Fig. 6.2: Histogram, empirical and theoretical PDF of the standard normal distributed test values T defined by Eq. (3.73).

It can be immediately seen that the estimated variances coincide closely with their theoretical distribution. Quantitatively, the p-value of the Kolmogorov-Smirnov-test indicates that the seemingly small deviations from the theoretical distribution are significant with respect to an error probability of 5%.
Evidently, the mean (-0.035) of the test value for coefficient $C_{10,0}$ contains a bias. Now, recall that this test value holds for an accuracy of one digit in the relative error, i.e. the bias must be multiplied by this accuracy (0.1) in order to determine the absolute bias, which is therefore equal to -0.0035. Considering the test scenario of 1000×800 samples, a confidence region of ± 0.0031 for the estimation of the relative variance is indicated by Eq. (3.72). This means that this bias is significant with respect to the error probability of 5%. An analysis including all coefficients shows that the range of bias is between -0.0146 and 0.0083. However, these extreme values are not clustered according to particular degrees or orders, but are arbitrarily distributed throughout all spherical harmonic coefficients.

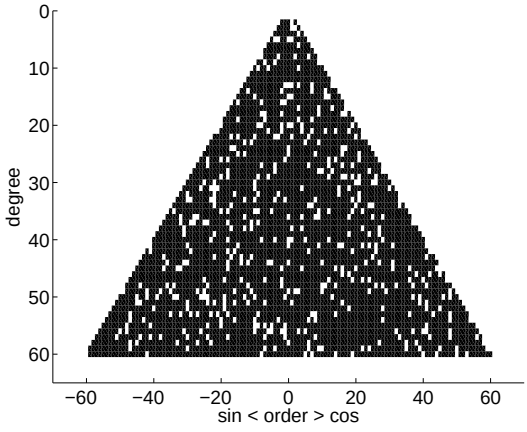


Fig. 6.3: Overall accuracy of the MC estimation process of the SGG model using 1000×800 samples; Hypotheses test based on Eq. (3.73)

To demonstrate this, we plot in Fig. 6.3 again the test value defined by Eq. (3.73), but now for 800,000 samples. We test again the hypothesis 6.3. This figure shows, on the one hand, the irregular distribution of the rejected null hypotheses, and on the other hand we can observe immediately that the number of rejected elements increases from the theoretical error probability of 5% to about 12%. This means that systematic effects (numerics, random number generators, etc.) occur, and we have to be very careful not to violate the "square root of M" law, where idealized assumptions are made, which are not easy to fulfill in practice. Table 1 gives an overview of the test statistics and shows that after 80,000 samples (corresponds to an accuracy of 2 digits),

the number of rejected elements increases. Therefore, we can conclude that the statistical behaviour of the test value (Eq. 3.73) is influenced by systematic effects.

	800	100×800	300×800	1000×800
SGG-only	4.97%	5.23%	7.32%	12.27%
SST-only	5.27%	5.46%	7.18%	12.27%
SGG+SST	5.38%	5.22%	7.21%	12.27%

Table 1: Percentage of rejected hypotheses tests based on Eq. (3.73) with respect to the number of samples

6.1.3 Results

Summarizing these results, it can be stated that for the given application, Monte Carlo integration is a suitable method to reproduce the variances with the expected accuracy of one or two digits in the relative error (Eq. 6.1.) Small yet significant biases occur and influence the third digit of the estimated variances. These biases are irregularly distributed , but for larger numbers of samples we have to be very careful with statements about the accuracy because of bias effects. The SST and the combined SST/SGG solution reflect the same behavior and the same range in bias.

A way to analyze the whole ensemble of spherical harmonic coefficients at once is to compare variances of geoid height anomalies for 250 arbitrary points. This test would respond sensitively to systematic effects, since systematic errors in the variances of the coefficients would immediately lead to errors in the variances of geoid height anomalies. Numerous numerical experiments did not provide any evidence of deviations from the above statements. The test values reflect the same behavior as for the single components. The bias of the test value lies approximately within the same range as before (between -0.0137 and 0.0067, which is now even a shorter interval).

As a summary of these simulations, it can be said that all proposed algorithms work and that the accuracy estimation (Eq. 3.72) is valid for all data sets and propagated functionals used in the experiments. A shortcoming of this procedure is due to the fact that the number of samples must be increased considerably, if a higher accuracy is aimed for. For one additional digit in accuracy the number of samples increases by a factor of 100, that is 80,000 samples to fix two digits of the estimated variances. However, as the experiments show, caution is advised when extending these technique to a large accuracy because of the biases in the solution.

As elaborated in Sect. 3.5.2, the stepwise estimation by conditioning by-passes this handicap. To demonstrate the performance of Alg. 3.9 the following test is performed. 800 samples are used to predict the variances of the geoid height anomalies of 250 randomly distributed points in a region between -60° and $+60^\circ$ latitude. A block-diagonal matrix with block size of at least 60 is used as a conditioner. This means that the spherical harmonic coefficients are arranged order by order, and this block-dominant system is matched with this 60×60 grid. Starting with the block of zonal coefficients, one can proceed order by order. If a block of a particular order can be placed within the actual grid, then it is included, otherwise the actual grid is closed, and the coefficients of this order are placed in the next block. Because of the block-dominant structure of the gravity field estimation process, it follows that the majority of the variances can be estimated purely by using the diagonal blocks. Nevertheless, the accuracy of this estimation can be further investigated by Monte Carlo integration.

To demonstrate the performance, Table 2 comprises the results for one individual point. It comprises between the rigorous computed values σ_H, estimation from the block-diagonal structure $\sigma_{H_{block}}$, stepwise estimation by conditioning $\sigma_{H_{cond}}$ and estimation by Monte Carlo integration $\sigma_{H_{MC}}$. The results are given for the point, for which the block estimate has the maximal error (based on 800 samples).

To maximize the visibility of the effects, the point with the largest error in the block estimate is used. Furthermore, the simple block estimation process is compared with the stepwise estimation by conditioning and the standard Monte Carlo integration approach, respectively. Note that in this representation standard deviations are used instead of variances. The experiments should, on the one hand, give the reader a better feel for the numbers. On the other hand, it should be remarked that they provide a proof of concept with respect to Monte Carlo integration and not with respect to specific mission scenarios.

unit: [m]	σ_H	$\sigma_{H_{block}}$	$\sigma_{H_{cond}}$	$\sigma_{H_{MC}}$
SGG-only	0.071554	0.071617	0.071533	0.071755
error		63	-21	201
SST-only	0.195057	0.199235	0.195295	0.189731
error		4177	237	-5327
SGG+SST	0.006704	0.006587	0.006719	0.006649
error		-117	15	-55

Table 2: Estimated standard deviations of geoid height anomalies in [m]

To gain some insight about the overall performance of the three variance estimation procedures, the mean standard deviations of all the 250 points are compared. The results, listed in Table 3, support the theoretical results. It comprises the means of absolute errors of the estimated block with the value ($\varepsilon_{\sigma_{H_{block}}}$), stepwise estimation by conditioning ($\varepsilon_{\sigma_{H_{cond}}}$), and estimation by Monte Carlo integration ($\varepsilon_{\sigma_{H_{MC}}}$) (based on 800 samples).
It is evident that for the step-wise approach of estimation by conditioning, the accuracy of variance estimation can be improved dramatically. For the block estimate we gain already 2-3 digits (error: 0.6%), and again we can improve these values by virtue of MC estimation by at least one third (800 samples imply a factor of $\sqrt{10}$). In addition it should be mentioned that, in general, also the standard Monte Carlo integration method leads to remarkably good results with the relative error below 2%.

unit: [m]	mean(σ_H)	mean$\lvert\varepsilon_{\sigma_{H_{block}}}\rvert$	mean$\lvert\varepsilon_{\sigma_{H_{cond}}}\rvert$	mean$\lvert\varepsilon_{\sigma_{H_{MC}}}\rvert$
SGG-only	0.064845	0.000015	0.000005	0.001306
SST-only	0.176710	0.001093	0.000126	0.003773
SGG+SST	0.005689	0.000024	0.000006	0.000114

Table 3: Mean error ε ($\varepsilon = \sigma_{\text{estimated}} - \sigma_{\text{true}}$) of the standard deviation of the geoid height anomalies of all 250 (randomly distributed) points

6.2 Scenario 2: Estimation of the variance component

6.2.1 Data sets

The numerical study is based on GOCE-simulation-data arranged by the European Space Agency (ESA, see De Sanctis et al. 2002). The same simulation setup has been used for the official completion test AR 2 of the HPF-Software, which was conducted at the beginning of 2006 and originate from ESA's end-to-end simulation. The simulated data set consists of:

- *Gradients*: Simulated gradients (main diagonal components of the gradient tensor) over a period of 60 days referred to the Gradiometer Reference Frame (GRF). The simulated gradients are based on the EGM96 (Lemoine et al. 1998) up to degree and order n,m=360 with a sampling rate of $s = 1$ sec.

- *Orbit*: The gradients are defined along a typical GOCE orbit, which is generated by numerical integration based on the EGM96 up to degree and order $l, m = 200$. Also considered in the orbit simulation are external force models as well as the DFAC (drag free and attitude control) system (ESA 1999).

- *Attitude*: The orientation of the satellite system (or the associated GRF) with respect to the inertial reference frame is realized by means of quaternions, which can be calculated from the information given by the star cameras and the gradiometer itself by integrating the angular velocities (Pail 2005). Naturally these quaternions are subjected to biased errors and high frequency noise. The error model in this case is chosen according to the assumed error characteristics in the ESA-Simulation.

- *Noise characteristics*: The noise behaviour of the gradiometer depends on the position of the GOCE satellite and is of periodical nature, which can be seen in the error spectrum (Schuh et al. 2006).

The simulated data set and its design parameter are summarized in Table 4

SGG test data			
positions	5 011 200	unknown parameter	40 397
observations	14 988 600	resolution	maximal degree/order 200
observation period	60 days	gravity model	EGM96
earth revolution	928	error charackteristics	colored noise
representation	T_{xx}, T_{yy}, T_{zz}		

Table 4: Design of parameters of the simulated SGG-data set

The SGG data will now be combined with SST data in order to estimate the harmonic coefficients of lower degree and order. The SST data are already available in terms of a normal equation system up to degree and order $l, m = 90$. With respect to the accuracy this presetting is sufficient to obtain a solid combined SGG and SST solution (Pail et al. 2007). The design parameters for the SST data set can be extracted from Table 5:

SST test data			
positions	5 011 200	unknown parameter	8277
observations	5 011 200	resolution	maximal degree/order 90
observation period	60 days	gravity model	EGM96
earth revolution	928	error charackteristics	white noise
available as $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}, \quad \boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{\ell}, \quad \boldsymbol{\ell}^T\boldsymbol{P}\boldsymbol{\ell}$ with $\boldsymbol{P} = \boldsymbol{I}$			

Table 5: Design of parameters of the simulated SST-data set

As a result of the specific GOCE configuration, in particular the sun synchronised orbit we have to expect instabilities within the normal equations due to the lack of data at polar regions. For that reason *Kaul rule's of thumb* (see Eq. 5.10) has been used to minimize these effects. In our scenario the combined SGG+SST data sets are regularized above degree and order $l \geq 90$. In this connection it should be mentioned that there already exist tailor-made regularization methods specifically for polar regions, which has been tested with the same simulation data set (Metzler and Pail 2005).

6.2.2 Test simulation

In order to achieve the best possible estimate of the harmonic coefficients, the optimal weighting factors between the combination of SGG data (Tabel 4) and SST data (Table 5) as well as the optimal regularization parameter are calculated according to the modified PCGMA-algorithm presented in Sect. 5.4.2. The evaluation of the data up to a maximum degree of $l = 200$ has been performed on the Jülich Multi Processor (JUMP)-Cluster in Juelich (JUMP 2006). 256 processors are used to execute the computations. The time of evaluation for a complete cycle of PCGMA averages around 7 hours. The start value of the weighting factor between the SGG- the SST-group is set to $\omega^{(0)} = 1$. As aforementioned, we used the Kaula regularization presented in Sect. 5.2.3. The regularization matrix $\boldsymbol{P}_\mu$ is a diagonal matrix with the variances of the harmonic coefficients given by Kaula's rule of thump (cf. 5.10). The corresponding prior information on the unknown parameters is the null vector $\boldsymbol{\mu} = \boldsymbol{0}$ and the start value of the regularization parameter is set to $\lambda^{(0)} = 1 \cdot 10^{-12}$. If $\boldsymbol{\mu} = \boldsymbol{0}$, the formula for the regularized solution is obtained:

$$\left(\bar{\boldsymbol{A}}^T\bar{\boldsymbol{A}} + \omega^{(0)}\boldsymbol{N}_{\text{sst}} + \lambda^{(0)}\boldsymbol{P}_\mu\right)\hat{\boldsymbol{x}}_\lambda = \bar{\boldsymbol{A}}^T_{\text{sgg}}\bar{\boldsymbol{l}}_{\text{sgg}} + \omega^{(0)}\boldsymbol{b}_{\text{sst}}. \tag{6.4}$$

As initialization values for all harmonic coefficients we used the null vector.

Convergence of the PCGMA-algorithm

Fig. 6.4 acts as an indicator for analysing the developing of iteration. To make it more comprehensive every step of iteration is depicted in this Figure. After each iteration a gravity field is determined (test model) and compared with the EGM96 model (true model). The blue lines in Fig. 6.4 are the mean values of the absolute differences between the coefficients ΔC_{lm} and ΔS_{lm} of the test- and true model:

$$\sigma_l^{\text{mean}} = \frac{1}{2l+1}\sum_{m=0}^{l}(\Delta C_{lm})^2 + (\Delta S_{lm})^2 \quad \text{for} \quad m \in \{1, \cdots, l\}. \tag{6.5}$$

The black lines represent the median of the absolute differences between the test- and the true model in each iteration step.

$$\sigma_l^{\text{median}} = \text{median}\left(|\Delta C_{lm}| \; + \; |\Delta S_{lm}|\right). \tag{6.6}$$

The red line is calculated according to *rule of thumb* (KAULA 1966) and indicates the order of magnitude of the coefficients with respect to their degree. The estimated standard deviation σ_l^{model} of the estimated paremter for every degree l is described by the green line with:

$$\sigma_l^{\text{test}} = \text{median}(\sigma_{C_{lm}} + \sigma_{S_{lm}}). \tag{6.7}$$

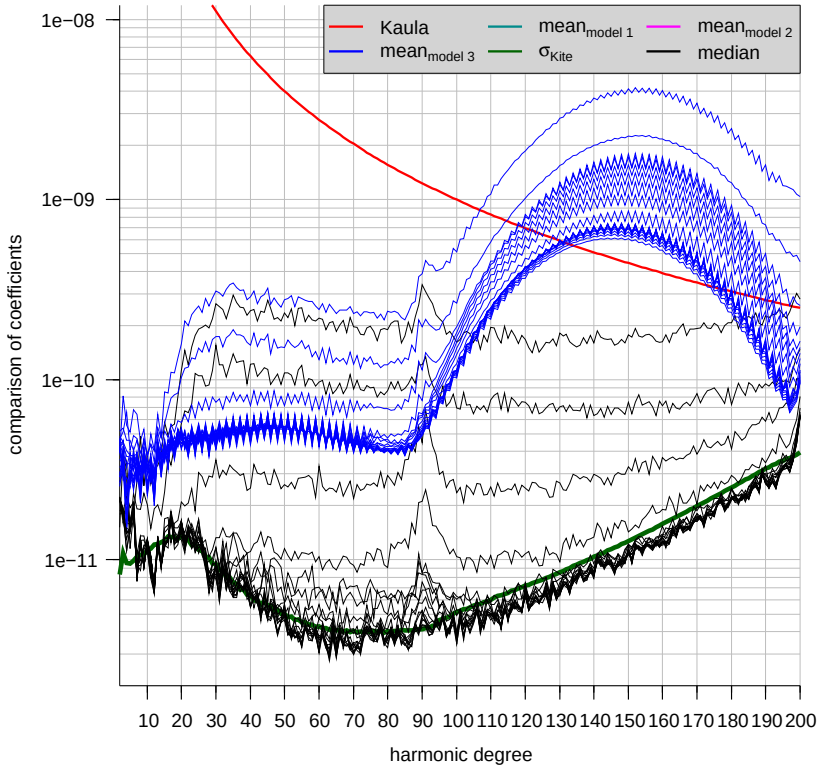


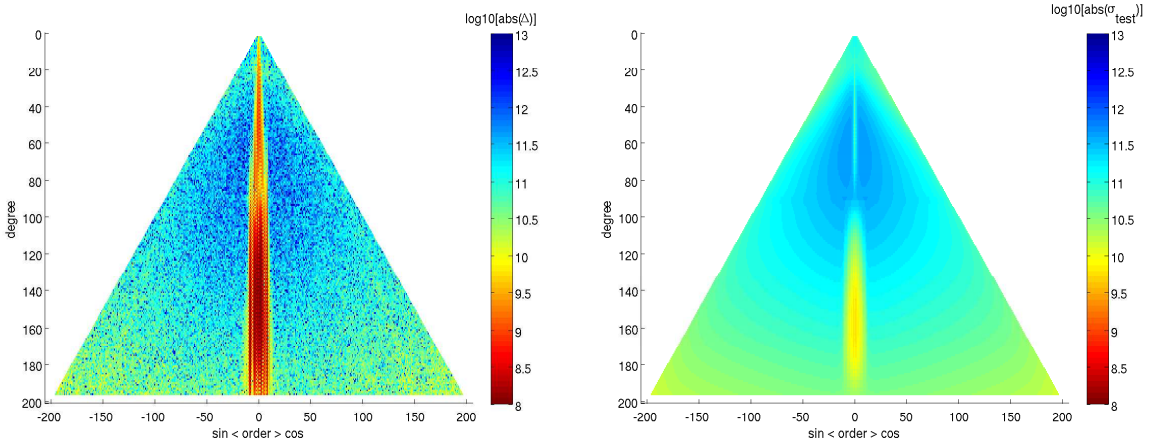
Fig. 6.4: Demonstration of convergence of PCGMA-algorithm

With ascending iterations the median curves starting to merge with the green curve. This issue does not hold for the mean value (blue curves). In contrast to the median the mean value is not a robust estimator. Fig. 6.5 explains the characteristics of the blue curves.
It shows the absolute differences of the coefficients between the test- and true model as well as the standard deviation of the test model. Especially the differences of the zonal and near zonal coefficients attract attention. These enormous variations probably occur due to the lack of data at polar regions. The standard deviation of the test model seems to be consistent with the absolute difference except for the zonal and near zonal coefficients, whose accuracies apparently are estimated to optimistically. Furthermore you can see a jump within the accuracies of the coefficients at degree $l = 90$.

Validation of MCVCE

After the first cycle of the PCGMA the update values for ω and λ according to Eq. (5.14) are determined by means of the MCVCE-algorithm. The calculation of the partial redundancies is carried out with the two random vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (see Fig. 5.10). The calculation of the partial redundancies of each data set only requires one random sample (see, e.g., KUSCHE 2003), which was explained in Sect. 4.3.1.

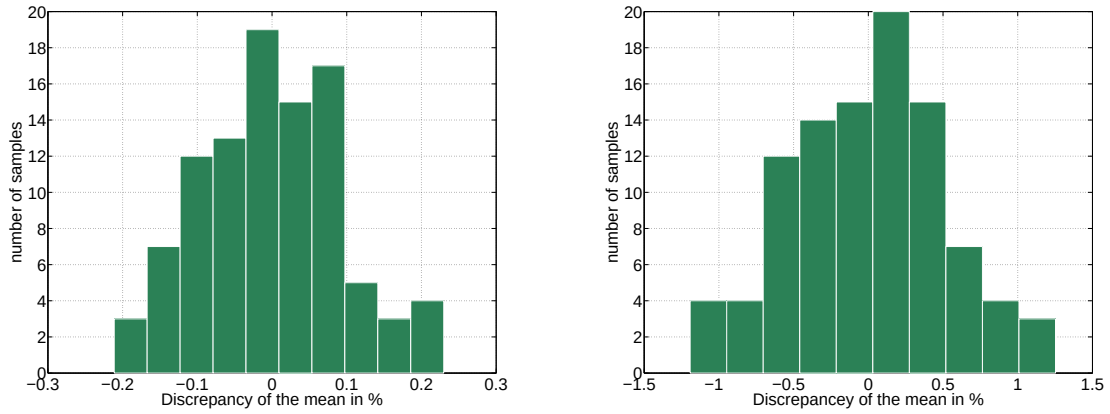
In order to validate this statement and to apply it on our simulation data set, we have estimated 100 realizations of the random vectors $\boldsymbol{\alpha}$ as well as of $\boldsymbol{\beta}$. The variations from the mean value for all determined partial



(a) absolute discrepancy (b) accuracy of the model 1

Fig. 6.5: Absolute discrepancy between the coefficients of the model 1 and the true model

redundancies of the SGG data r_{sgg} as well as of the SST data r_{sst} is shown in the histograms 6.6(a) and 6.6(b), respectively.



(a) partial redundancy r_{sgg} (b) partial redundancy r_{sst}

Fig. 6.6: Histogram of the estimation of the redundancy contribution

As we can see in Figure 6.6, all 100 random samples of the r_{sgg} and r_{sst} vary only about 0.3% and 1% from the mean value, respectively. We conclude that the computed partial redundancies according to step 6 of Alg. 4.3 can be used for accurate variance component estimate samples. The partial redundancies of the SST data set r_{sst} can be calculated using the total redundancy of the estimated model and the partial redundancies of the SGG data r_{sgg}:

$$r_{\text{sst}} = n_{\text{sgg}} + n_{\text{sst}} - m - r_{\text{sgg}}. \tag{6.8}$$

This result will be calibrate the estimation according to the Eq. 4.32 (cf. 6^{th} step of the Alg. 4.3). The difference between the two estimates (Eqs. 6.8 and 4.32) amounts in the last iteration of MCVCE to approximately 845. That is

$$\begin{aligned} \Delta r &= |r_{\text{sst}}^{(1)} - r_{\text{sst}}^{(2)}| \\ \Delta r &= |5\,010\,409.09 - 5\,011\,253.5| \approx 845, \end{aligned} \tag{6.9}$$

where $r_{\text{sst}}^{(1)}$ denotes the estimate with Eq. (6.8) and $r_{\text{sst}}^{(2)}$ denotes the estimate with Eq. (4.32). This value of difference is equal to a relative change of 0.00042 in the variance component σ_{sst}^2. Therefore the Monte Carlo trace estimator is suitable for calculating the partial redundancies and moreover meets the demands regarding

the accuracy for the variance component estimation.

Convergence of the MCVCE

After 4 cycles of iteration the MCVCE algorithm converges. The determined variance components: $\hat{\boldsymbol{\sigma}} = \{\sigma_1^2, \sigma_2^2, \sigma_\mu^2\}$ and the corresponding values ω and λ are stated in Table 6. The optimal estimated ω and λ are marked in blue.

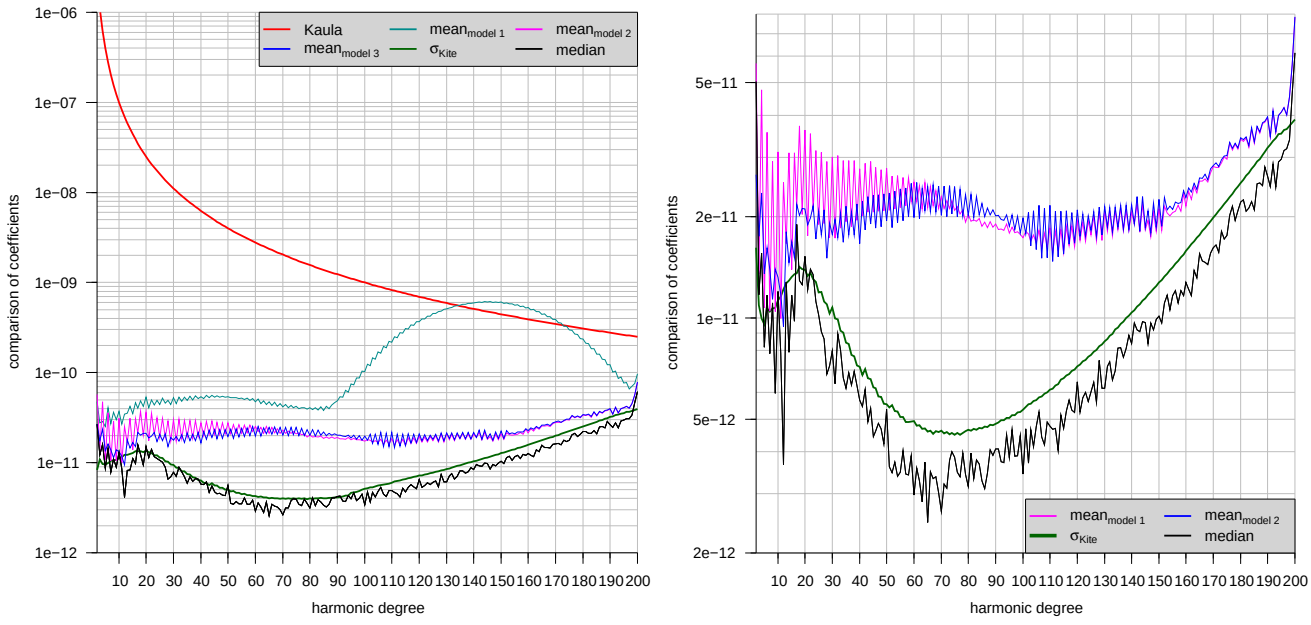
	zero step	first step	second step	third step	last step
ω	1	1.08457	1.17173	1.17173	1.17173
λ	$1 \cdot 10^{-12}$	0.93568	0.84595	0.84012	0.83962

Table 6: Change of the regularization parameter λ and of the weighting parameter ω during the iteration process of the MCVCE algorithm

In the following investigations 3 models will be presented. The first model (zero step) represents the solution, which arises from the initial values λ and ω taken from the first column in Table 6, $\lambda = 1 \cdot 10^{-12}$ means no regularization is applied. The second model uses λ and ω from the first step of iteration and in analogy the third model is based on the optimal values for ω and λ (last column of Table 6).

6.2.3 Results

The results of all three solutions will now be compared. For that purpose we calculate the degree variances of all three models and contrast them with the degree variances from the EGM96 in order to evaluate to what extent each solution deviates from the *true* solution (Fig. 6.7).



(a) all models

(b) model 2 and 3

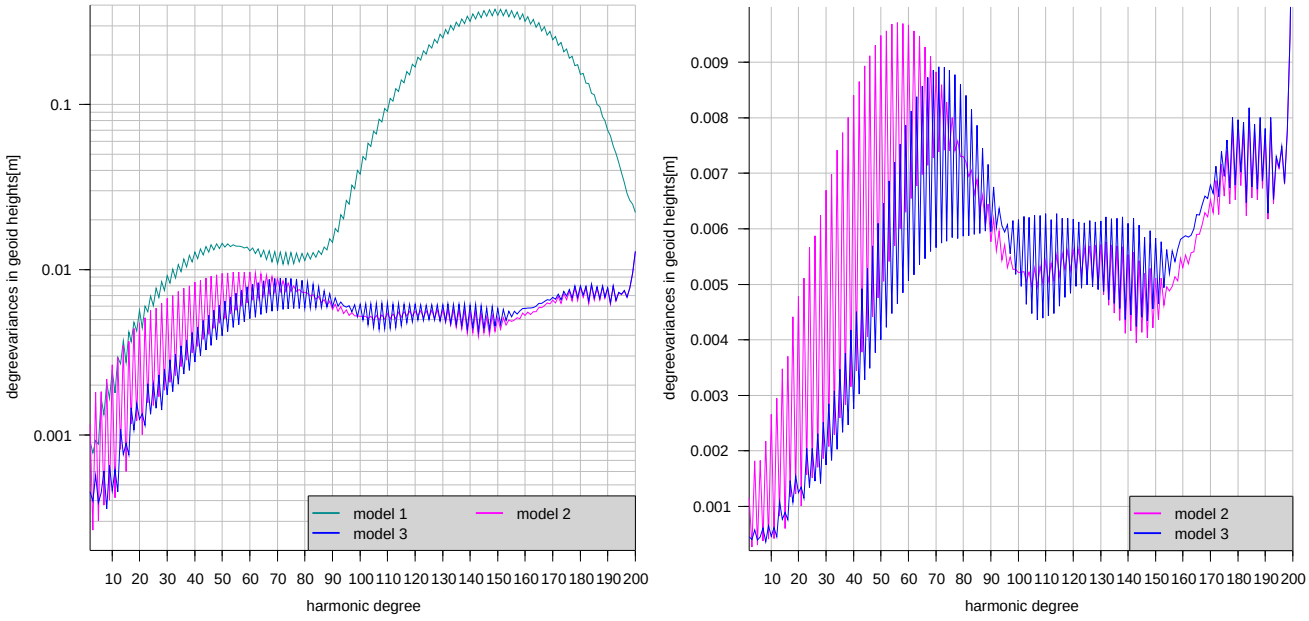
Fig. 6.7: Differences of the spherical harmonic coefficients absolute discrepancies over all orders of the same degree (median, mean) with respect to the size of the Kaula and the accuracy of the adjustment (median values)

As expected the three solutions does not differ significantly up to degree $l = 40$. Because of the ill-posed problematic, the errors of the first solution increase with increasing degree and eventually exceed Kaula's curve. This problematic will be antagonized by regularization in the second solution. The signal is attenuated so that it does not exceed a certain limit. In order to show the influence of the optimal determined λ and ω on the estimated coefficients, solution 2 and solution 3 are compared in Fig. 6.7(b). In this figure you can clearly see in how far the optimal weighting factors and regularization parameters smooth the signal, in particular with respect to the lower and medium degrees.

Another important comparison is the representation of the degree variances between the solutions in terms of geoid height anomalies:

$$\sigma_l(N) = R\sqrt{\sum_{m=0}^{l}(\Delta C_{lm}^2 + \Delta S_{lm}^2)}, \tag{6.10}$$

where R is the earth's radius. This can be realized with subject to the individual degrees (Fig. 6.8) as well as cumulative (Fig. 6.10). The geoid height anomalies, according to 6.10, are represented on the y-axis using a logarithmic scale. If now regularization is applied then the largest variations appear to be over degree $n \geq 90$ for the SGG data. This jump can also be found in Fig. 6.7 and probably is ascribed to a non-optimal determination of the regularization and weighting parameters between the combination of SGG and SST data. Using the optimal values for λ and ω this is removed for the most part.



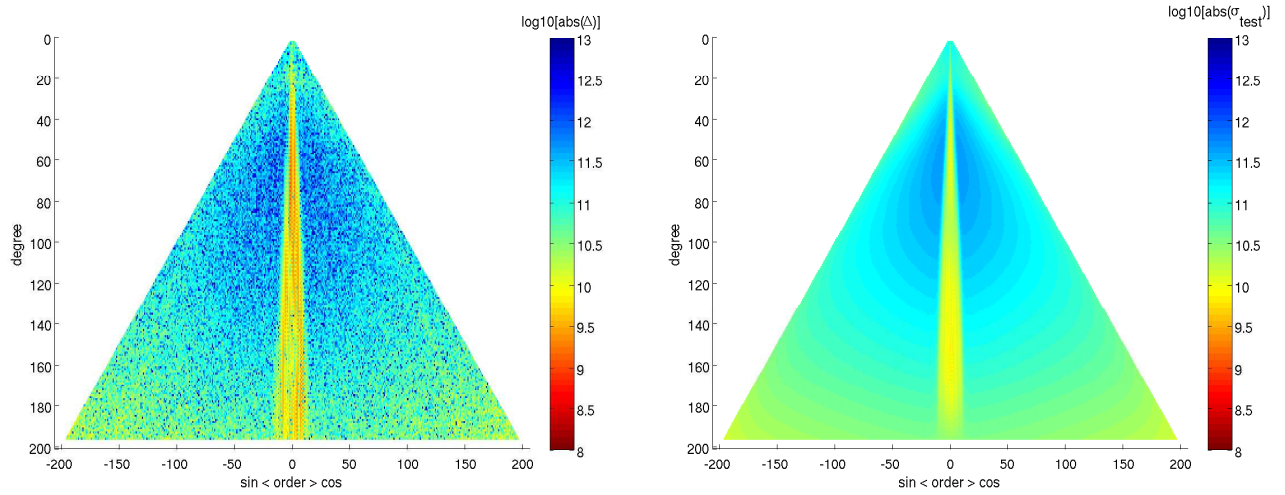
(a) all models (b) model 2 and 3

Fig. 6.8: Degree variances between the solutions in terms of geoid height anomalies

In this matter we compare the absolute errors with respect to the reference model EGM96 and the corresponding accuracies for model 1 (Fig. 6.5) and for model 3 (Fig. 6.9). Apparently the absolute errors and their accuracies seem to be more consistent than illustrated in (Fig. 6.9). Furthermore the jump in Fig. 6.5(b) is mostly removed.

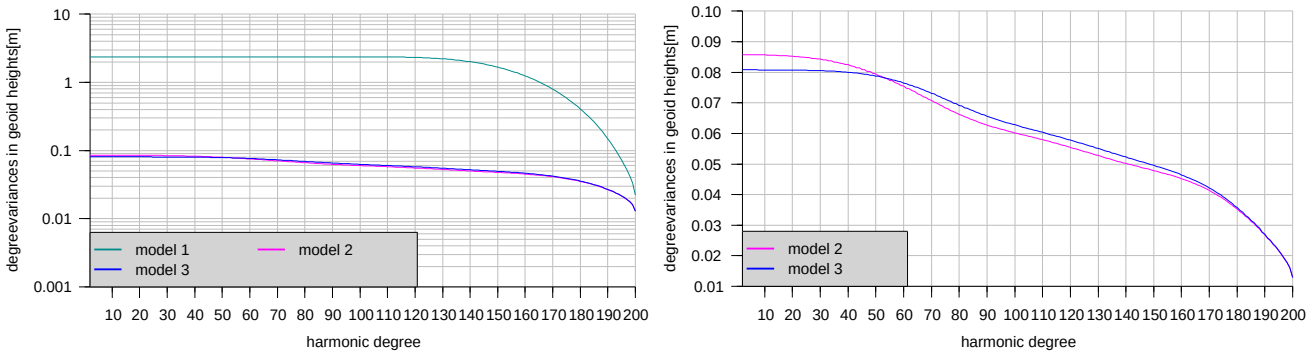
In addition the accumulated errors according to 6.10 (see Fig. (6.10)) have been reduced from 2.35m for the first model to 7cm for the third model. We now compare model 2 and model 3, the results are shown in Fig. 6.10(b). A minor improvement regarding global errors can be extracted from this Figure. The most important part in this case however is the during the iteration. Apparently this comparison shows that the MCVCE algorithm already gives reasonable results for the variance components after the second cycle of iteration.

The geographical assignment of the differences for all models with respect to the reference solution (EGM96) is depicted in Fig. 6.11 in terms of geoid undulations. The statistical data of the residues is listed in Table 7. The global recovery area was chosen to be from 0 to 360° longitude and -89.5 to 89.5° latitude and the gravity undulations was modeled by $0.5 \times 0.5°$ height anomalies. The second row in Table 7 excludes the polar gap bounding the area to $-83.5 < \Phi < 83.5°$. As you can see the standard deviation has improved significantly from 7,4cm to 2,5cm after the first iteration. With respect to the global scale the standard deviation has improved from 12,18m to 32,6cm. The maximum variations are located in polar regions, which is quite obvious due to the orbit setup of the satellite. Comparing the differences between model 3 and EGM96 (see Fig. 6.11(c)) you can see the biased error, which is periodically distributed over the whole earth. These errors are analyzed in detail in Schuh et al. (2006).



(a) absolute discrepancy (b) accuracy of the model 3

Fig. 6.9: Absolute discrepancy between the coefficients of the model 3 and the true model



(a) all models (b) model 2 and 3

Fig. 6.10: Cumulative Geoid Accuracy-Geoid height

global reconstruction				
	min	**max**	**mean**	σ
model 1	-178.145m	181.298m	0.505m	12.180m
	-0.893m	0.972m	-0.001m	0.074 m
model 2	-4.015m	3.755m	-0.055m	0.457m
	-0.155m	0.140m	0.000m	0.025 m
model 3	-3.499m	4.370m	-0.018m	0.326m
	-0.153m	0.139m	-0.000m	0.026 m

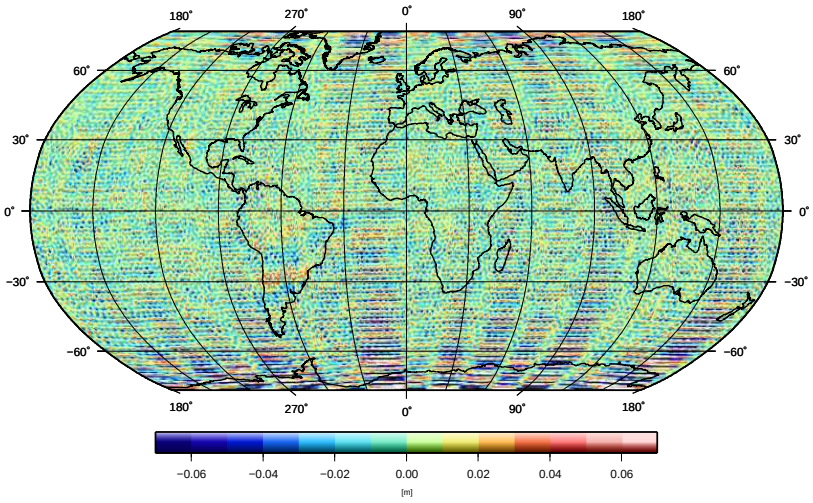
Table 7: Global reconstruction of second level information: discrepancies of geoid height anomalies between true model (EGM 96) and test model. The first row gives the discrepancies for the area from $-89.5 < \Phi < 89.5°$ and the second row excludes the polar gap bounding the area to $-83.5 < \Phi < 83.5°$

In order to get a better understanding how the fine structure of the calculated gravity models behave, the global calculation will be complemented with a calculation conducted over a bounded local area. The regional recovery area was chosen to be from $-10°$ to $60°$ longitude and $10°$ to $60°$ latitude. The results are given in Table 8. In this case the standard deviation has been improved from 3,2cm to 2,3cm.

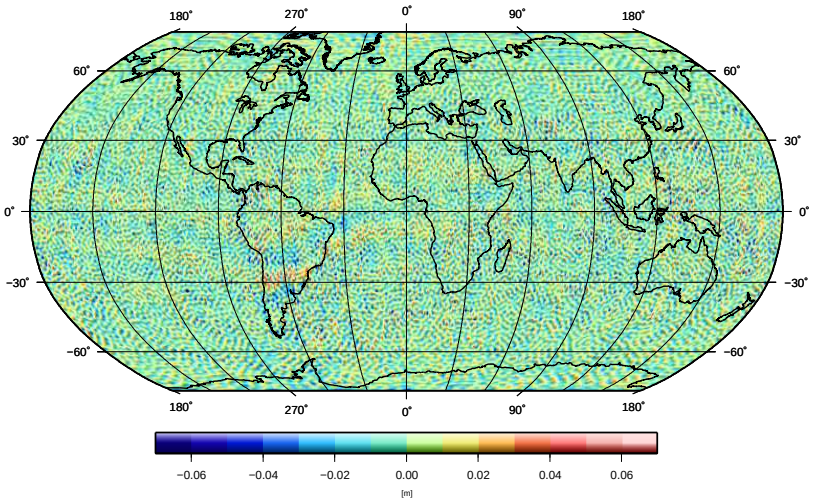
local reconstruction				
	min	max	mean	σ
model 1	-0.115m	0.118m	-0.001m	0.032m
model 2	-0.104m	0.116m	0.000m	0.023m
model 3	-0.104m	0.115m	-0.000m	0.023m

Table 8: Local reconstruction of second level information: discrepancies of geoid height anomalies between true model (EGM 96) and test model (without smoothing)

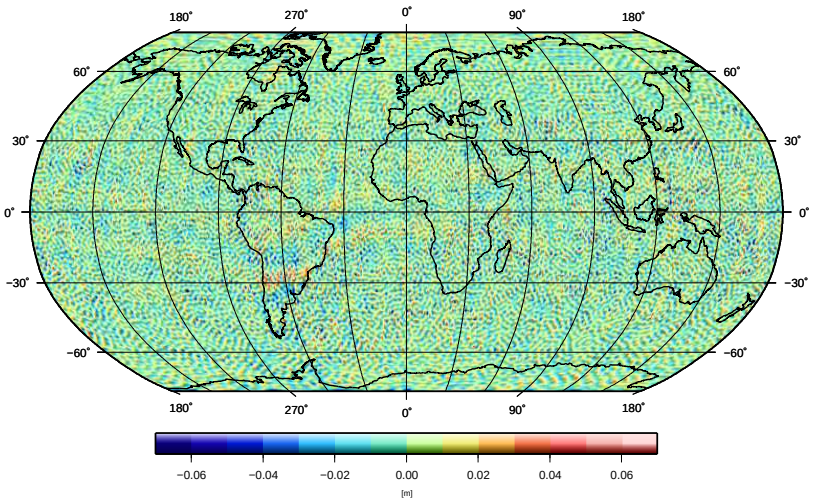
As a conclusion you can say that the MCVCE algorithm has been successfully implemented in the processing software PCGMA. At first the Monte Carlo trace estimator has been investigated and statistically validated. Comparing all three models with respect to the EGM96 it can be stated, that even after the first cycle of iteration the weighting and regularization parameters yield reasonable results regarding the variance components.



(a) model 1



(b) model 2



(c) model 3

Fig. 6.11: Comparison of the model discerpancies in term of geoid heights anomalies between the test and the EGM96 model

7 Summary and Conclusions

The concept of Monte Carlo methods covers many numerical methods in which random number are used to solve complex problems. For the one part, Monte Carlo methods are used within this work as an alternative tool to compute the inverse of the normal equation matrix of huge systems. For another part, these methods are used to find an approximation of trace terms of huge matrix products needed for the prediction of variance components, which would otherwise be computationally very intensive.

In chapter 3, the first Monte Carlo method used to obtain estimates of the inverse of the normal equation matrix since this matrix is too large to be inverted by using conventional or sparse algorithms. Monte Carlo integration allows a condensed representation of the variance/covariance information and fits well into efficient solution strategies for large-scale equation systems. The proposed variance/covariance estimation procedure is flexible and may be integrated into many types of solvers, e.g. sparse solvers, parallel direct solvers using distributed memory, or iterative solvers. Furthermore, it supports combined solution strategies for heterogeneous data sets. A shortcut of many simulation processes lies in the estimation of the accuracy. In Sect. 3.4, confidence regions for the estimated variances are deduced and the simulations support this concept also for ill-posed inverse problems. Because of the direct connection between confidence region and number of samples, the number of necessary samples at a fixed a priori accuracy and probability level can determined easily. However, the simulations also show that the standard Monte Carlo integration has limitations with respect to extreme accuracy conditions. If a relative accuracy of three or more digits is necessary, then biases have to be taken into account. In this case, estimation by conditioning gives an improvement. In contrast to a purely algebraic approach, such as via incomplete inverses, Monte Carlo estimation by conditioning provides a tool to further increase the accuracy of the VCM not only for the adjusted parameters, but also for all propagated quantities.

The second Monte Carlo technique in chapter 4 deals with an efficient way to implement the variance component estimation, which means, to estimate the optimal relative weight factors of different types of observations and the regularization parameter, simultaneously. Naturally, the choice of the optimal weights for different data sets is very important for obtaining reliable results. However, most of the traditional methods require intensive computations. The proposed Monte Carlo algorithm avoids most of this intensive computation and uses efficient estimators to compute group redundancies which are needed to compute the variance components. One needs only a single additional solution of the normal equation for every weighting parameter and another solution for every regularization parameter. The novelty of the proposed algorithm is its flexibility to work in iterative solvers as well as in direct solvers and in the possibility to combine different form of input data.

In a case study in chapter 5 these Monte Carlo techniques are applied to simulated GOCE data, where SGG and SST observations are combined for reconstructing the gravity field. The objective is to compute a high-resolution spherical harmonic model including a quality description of the estimated coefficients in terms of a full VCM as well as the optimal regularization and weighting parameters. A tailored version of Monte Carlo techniques is integrated into the iterative solver PCGMA. By means of these algorithms we are able to cover the whole processing chain (parameter-,covariance- and variance components estimation) in a uniform way based on the direct application of observation equations. Due to the sequential access to the observation equations, the numerically intensive parts of the algorithms can be parallelized very easily and are implemented as parallel programs and tested on the super computers JUMP in Juelich.

The tow numerical experiments in chapter 6 showed evidence that Monte Carlo methods work well and are efficient for estimating the variance/covariance information and for estimating variance components in a case study of the combination of heterogeneous simulated GOCE data. In the first test computations of the VCM is estimated for 3717 harmonic coefficients. We conclude that the development algorithms work and that the derived confidence interval of the estimates variances is valid for different data sets.
The second scenario collects the harmonic coefficients from $l, m = 200$ with 40 397 parameters. The spherical harmonic coefficients as well as the optimum weighting and regularization parameters are estimated by means of the modified PCGMA and MCVCE algorithms. It can be stated on the one hand, that without additional computational effort within PCGMA, one can compute reliable variance components. On the other hand, only a few iterations are necessary to achieve the convergence of the iterative MCVCE algorithm.

Thus the following objectives are accomplished:

The iterative solving algorithm PCGMA for the harmonic analysis was extended by two components, which now allow an integrated variance component estimation, as well as to provide variance/covariance information on the estimated parameters. Both model-specific options are essential components in the course of the GOCE-HPF project, where the PCGMA algorithm in terms of a "tuning machine" is implemented.
In future research we would like to apply this algorithm for rigorous solutions of high-resolution (up to degree an order $l, m = 720$) combination models (e.g. GRACE-GOCE combined with terrestrial data) . For this reason it is certainly necessary to enhance the efficiency of the procedure in order to reduce the number of samples.

The choice of the PDFs to generating samples plays the vital role within a Monte Carlo simulation. The choice of normal distribution for algorithms presented in chapter 3 is justified by the asymptotic behavior of the normal distribution with respect to large numbers of samples. In addition other PDFs for example, the discrete PDF proposed in chapter 4 to approximate the trace of a large-scale matrix satisfies a minimum variance criterion. However, the use of other PDFs for generating of Monte Carlo samples needs additional research in the future.

The significant biases in the estimated variances of the VCM occur and influence on the third digit of the estimated variances. In order to remove the effect of biases on the estimate of variance/covariance matrix, development of bias-corrected estimator are essential in future advanced research.

List of Figures

List of Tables

References

Abwerzger, G. (1999) Accuracy Analysis on High-Resolution Geopotential Models. Diplomarbeit, TU Graz.

Alkhatib, H. and W.-D. Schuh (2007) Integration of the Monte Carlo Covariance Estimation Strategy into Tailored Solution Procedures for Large-scaled Least Squares Problems. *J. Geodesy*, 81:53–66.

ATLAS (2007) The ATLAS Homepage. *http://math-atlas.sourceforge.net*.

Bauer, O. and J. Kusche (2006) LSQR Based Geopotential Recovery. In: *Gravity Field of the Earth (first IGFS 2006, Istanbul)*. Springer, Berlin - Heidelberg - New York, in print.

Bouman, J. (1998) Quality of Regularization Methods. Technical Report, Delft University press, Delft.

Box, G.E.P and M.E. Muller (1958) A Note on the Generation of Random Normal Deviates. *Annals Mathematical Statistics*, 29:610–611.

Boxhammer, C. (2006) *Effiziente Numerische Verfahren zur Sphärischen Harmonischen Analyse von Satellitendaten*. Dissertation, Institut für Theoretische Geodäsie der Universität Bonn.

Colombo, O.L. (1981) *Numerical Methods for Harmonic Analysis on the Sphere*. Reports of the Department of Geodetic Science. Ohio State University (OSU), Ohio. No. 310.

Crocetto, M., M. Gatti and P. Russo (2000) Simplified Formulae for the BIQUE Estimation of Variance Components in Disjunctive Observation Groups. *J. Geodesy*, 74:447–457.

De Sanctis, S., S. Cesare, E. Detoma, M. Dumontel, R. Floberghagen, M. Parisch, G. Sechi and A. Anselmi (2002) *Simulator Design Specification*. Technical Note, G0-SP-AI-0001, Alenia Spazio, Turin, Italy.

Ditmar, P., J. Kusche and R. Klees (2003) Computation of Spherical Harmonic Coefficients from Gravity Gradiometry Data to be Acquired by the GOCE Satellite: Regularization issues. *J. Geodesy*, 77:465–477.

Dongarra, Jack J., Iain S. Duff, Danny C. Sorensen and Henk Van Der Vorst (1990) *Solving Linear Systems on Vector and Shared Memory Computers*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Dudewicz, E. G. and S. N. Mishra (1988) *Modern Mathematical Statistics*. John Wiley and Sons, New York.

Eicker, A., T. Mayer-Guerr and K. H. Ilk (2005) Global Gravity Field Solutions Based on Simulation Scenario of GRACE SST Data and Regional Refinements by GOCE SGG Observations. In: Jekeli, C., L. Bastos and J. Fernandes (Ed.), *Gravity, Geoid and Space Missions*, Lecture Notes in Earth Sciences 129 (GGSM 2004, Porto). Springer, Berlin - Heidelberg - New York, 66–71.

ESA (1999) *The Four Candidate Earth Explorer Core Missions – Gravity field and Steady-State Ocean Circulation Mission*. ESA Report SP-1233(1), Granada.

ESA (2002) From Eötvös to mGal+. Technical Report, ESA-Project, ESA/ESTEC Contract No. 14287/00/NL/DC.

Feller, W. (1968) *An Intoduction to Probability Theory and Its Applications*. Wiley, New York, 4. edn.

Fishman, G. S. (2003) *Monte Carlo: Concepts, Algorithms and Applications*. Springer, New York, 2. edn.

Földvary, L., D. Svehla, C. Gerlach, T. Gruber, M. Wermuth, N. Sneeuw, B. Frommknecht, H. Oberndorfer, T. Peters, M. Rothacher, R. Rummel and P. Steinberger (2005) Gravity Model TUM-2Sp Based on the Energy Balance Approach and Kinematic CHAMP Orbits. In: Reigber, C., H. Luehr, P. Schwintzer and J. Wickert (Ed.), *Earth observation with CHAMP – Results from three years in orbit*. Springer, 13–18.

Gelman, A., J. B. Carlin, H. S. Stern and D. B. Rubin (2004) *Bayesian Data Analysis*. CHAMPMAN and HALL/CRC, Boca Raton, London, New York, Washington, D.C., 2. edn.

Geman, S. and D. Geman (1984) Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restorations of Images. *IEEE Transactions Patt Anal Machine Intell PAMI*, 6:721–741.

Gentle, J. E. (2003) *Random Number Generation and Monte Carlo Methods.* Wiley, Springer, 2. edn.

Gill, P.E., W. Murray and M. Wright (1981) *Practical Optimization.* Academic Press, London - New York - Toronto - Sydney - San Francisco, 4. edn.

Girard, DA. (1989) A fast Monte Carlo Cross-Validation Procedure for Large Least Squares Problems with Noisy Data. *Numer. Math.*, 56:1–23.

Glasserman, P. (2003) *Monte Carlo Methods in Financial Engineering.* Springer, New York - Berlin - Heidelberg -.

Golub, GH and U von Matt (1997) Generalized Cross-Validation for Large-Scale Problems. *J. Comput Graph Stat*, 6:1–34.

Grafarend, E. und D'Hone, A. (1978) *Gewichtsschätzung in Geodätischen Netzen*, Band 88 der Reihe A. Deutsche Geodätische Kommission, München.

Gropp, William, Ewing Lusk and Anthony Skjellum (1994) *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* MIT Press, Cambridge, MA.

Gundlich, B., K.R. Koch and J. Kusche (2003) Gibbs sampler for Computing and Propagating Large Covariance Matrices. *J. Geodesy*, 77:514–528.

Hansen, P.C. (1997) *Rank-Deficient and Discrete Ill-Posed Problems.* SIAM, Philadelphia.

Harville, DA (1999) Use of the Gibbs Sampler to Invert Llarge, possibly Sparse, Positive Definite Matrices. *Lin Alg Applic*, 289:203–224.

Hutchinson, MF (1990) A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. *Commun Statist Simulat Comput*, 19:433–450.

Ilk, K. H. and A. Löcher (2003) The Use of Energy Balance Relations for Validation of Gravity Field Models and Orbit Determination. In: Sansò, F. (Ed.), *A Window on the Future of Geodesy, IAG General Assembly, Sapporo, Japan*, Band 128 der Reihe International Association of Geodesy Symposia. Springer, 494–499.

Johnson, V. E, . (1996) Studing Convergence of Markov Chain Monte Carlo Algorithm Using Coupled Sample Pathes. *J. Am. Stat. Assoc.*, 91:154–166.

Johnson, N. L. and S. Kotz (1970a) *Distributions in statistics. Continuous univariate.* 1st Ed. Wiley, New York.

Johnson, N.L and S. Kotz (1970b) *Continuous Univariate Distributions- 1.* Houghton Mifflin, Boston.

Johnson, N.L and S. Kotz (1970c) *Continuous Univariate Distributions- 2.* Houghton Mifflin, Boston.

JUMP (2006) (JU)elich (M)ulti(P)rozessor (JUMP) Dokumentation. `http://jumpdoc.fz-juelich.de/`. Status: 2006.

Kalos, M. and P. Whitlock (1986) *Monte Carlo Methods. Vol. I: Basics.* Wiley & Sons, New York.

Kaula, W. M. (1966) *Theory of Satellite Geodesy.* Blaisdell, Waltham.

Klees, R., R. Koop, P. Visser and J. van den Ijssel (2000) Efficient Gravity Field Recovery from GOCE Gravity Gradient Observations. *J. Geodesy*, 74:561–571.

Koch, K.R. (1986) Maximum Likelihood Estimate of Variance Components;Ideas by A.J. Pope. *Bulletin Géodésique*, 60:329–338.

Koch, K.R. (1987) Bayesian Inference for Variance Components. *Manuscripta geodaetica*, 12:309–313.

Koch, K.R. (1999) *Parameter Estimation and Hypothesis Testing in Linear Models.* 2nd Ed. Springer, Berlin - Heidelberg - New York.

Koch, K.R. (2000) *Einführung in die Bayes-Statistik.* Springer, Berlin - Heidelberg - New York.

Koch, K.R. (2005) Determining the Maximum Degree of Harmonic Coefficients in Geopotential Models by Monte Carlo Methods. *Studia Geophysica et Geodaetica*, 49:259–275.

Koch, K.R. (2006) Bayesian Image Restoration by Markov Chain Monte Carlo Methods. *ZfV*, 130:318–327.

Koch, K.R. and J. Kusche (2002) Regularization of Geopotential Determination from Satellite Data by Variance Components. *J. Geodesy*, 76:259–268.

Koch, K.R., J. Kusche, C. Boxhammer and B. Gundlich (2004) Parallel Gibbs Sampling for Computing and Propagating Large Covariance Matrices. *ZfV*, 129:32–427.

Kusche, J. (2002) *Inverse Probleme bei der Gravitationsfeldbestimmung mittels SST- und SGG-Satellitenmissionen*, Band 548 der Reihe C. Deutsche Geodätische Kommission, München.

Kusche, J. (2003) A Monte-Carlo Technique for Weight Estimation in Satellite Geodesy. *J. Geodesy*, 76:641–652.

Kusche, J. and R. Klees (2002) Regularization of Gravity Field Estimation from Satellite Gravity Gradients. *J. Geodesy*, 76:359–368.

Kusche, K. and J. P. van Loon (2004) Statistical Assessment of CHAMP Data and Models Using the Energy Balance Approach. In: et. al., Reigber (Ed.), *Earth observation with CHAMP, results from three years in orbit.* Springer, Berlin - Heidelberg - New York.

Lemoine, F.G., D.E. Smith, L. Kunz, R. Smith, E.C. Pavlis, N.K. Pavlis, S.M. Klosko, D.S. Chinn, M.H. Torrence, R.G. Williamson, C.M. Cox, K.E. Rachlin, Y.M. Wang, R.H. Rapp and R.S. Nerem (1998) The Development of the NASA GSFC and NIMA Joint Geopotential Model. *Proceedings of the "International Symposium on Gravity, Geoid, and Marine Geodesy"*, Tokyo, Japan.

Leonard, T. and J. S.J. Hsu (1999) *Bayesian Methods: An Analysis For Statisticians and Interdisciplinary Reserarchers.* Cambridge University, Cambridge.

Liu, J.S. (2001) *Monte Carlo Strategies in Scientific Computing.* Springer, Berlin - Heidelberg - New York.

Löcher, A. and K. H. Ilk (2005) Energy Balance Relations for Validation of Gravity Field Models and Orbit Determinations Applied to the Results of the CHAMP Mission. In: Reigber, C., H. Lühr, P. Schwintzer and J. Wickert (Ed.), *Earth observation with CHAMP – Results from three years in orbit.* Springer, 53–58.

Mayer-Gürr, T. (2006) *Gravitationsfeldbestimmung aus der Analyse kurzer Bahnbögen am Beispiel der Satellitenmission CHAMP und GRACE.* Dissertation, Institut für Theoretische Geodäsie der Universität Bonn.

Mayer-Gürr, T., K.H. Ilk, A. Eicker and M. Feuchtinger (2005) ITG-CHAMP01: A CHAMP Gravity Field Model from Short Kinematical Arcs of a One-Year Observation Period. *J. Geodesy*, 78:462–480.

Meissl, P. (1982) *Least Squares Adjustment a Modern Approach.* mitteilungen der geodätischen Institute der Technischen Universität Graz. Folge 43.

Metzler, B. (2007) *Spherical Cap Regularization–A Spatially Restricted Regularization Method Tailored to the Polar Gap Problem.* Dissertation, Graz University of Technology, Institut of Navigation and Satellite Geodesy.

Metzler, B. and R. Pail (2005) GOCE data processing: The Spherical Cap Regularization Approach. *Stud. Geophys. Geod.*, 49:441–462.

Meyer, C. (2000) *Matrix Analysis and Applied Linear Algebra with Solutions Manual.* Society for Industrial and Applied Mathematics, SIAM, Philadelphia.

Migliaccio, F., M. Reguzzoni and F. Sansò (2004) Space-Wise Approach to Satellite Gravity Field Determination in the Presence of Coloured Noise. *J. Geodesy*, 78:304–313.

Mosegaard, K. and A. Tarantola (1995) Monte Carlo Sampling of Solutions to Inverse Problems. *J. Geophysical Research*, 10, No B7:431–447.

Mueller, F. and T. Mayer-Guerr (2004) Comparison of Downward Continuation Methods of Airborne Gravimetry Data. In: Sansò, F. (Ed.), *A windo of the future of Geodesy*, Lecture Notes in Earth Sciences 128 (GGSM 2003, Sapporo). Springer, Berlin - Heidelberg - New York, 254–258.

Pail, R. (2005) A parametric study on the impact of satellite attitude errors on GOCE gravity field recovery. *J. Geodesy*, 79:231–241.

Pail, R. and G. Plank (2002) Assessment of Three Numerical Solution Strategies for Gravity Field Recovery from GOCE Satellite Gravity Gradiometry Implemented on a Parallel Platform. *J. Geodesy*, 76:462–474.

Pail, R., B. Metzler, B. Lackner, T. Preimesberger, E. Höck, W.-D. Schuh, H. Alkhatib, Ch. Boxhammer, Ch. Siemes and M. Wermuth (2007) GOCE-Schwerefeldprozessierung: Software-Architektur und Simulationsergebnisse. *ZfV*, 132:(in print).

Pitman, J. (1993) *Probability.* Springer, New York, Beriln, Heidelberg.

Plank, Gernot (2004) *Numerical Solution strategies for the GOCE Mission by Using Cluster Technologies.* Dissertation, Institut für Navigation und Satellitengeodäsie der Technischen Universität Graz.

Poder, K. and C.C. Tscherning (1973) *Cholesky's Method on a Computer.* Internal Report No. 8, The Danish Geodetic Institute.

Rao, CR. and J. Kleffe (Ed.) (1988) *Estimation of Variance Components and Applications.* North-Holland, Amsterdam.

Reigber, C. (1969) *Zur Bestimmung des Gravitationsfeldes der Erde aus Satellitenbeobachtungen*, Band 137 der Reihe C. Deutsche Geodätische Kommission, München.

Reigber, C., H. Lühr, P. Schwintzer and J. Wickert (Ed.) (2004) *Earth Observation with CHAMP: Results from Three years in Orbit.* Springer, Berlin-Heidelberg.

Robert, C.P. and G. Casella (1999) *Monte Carlo Statistical Methods.* Springer, New York.

Rubinstein, R. Y. (1981) *Simulation and the Monte Carlo Method.* Wiley, New York, Chichester, Brisbane, Toronto.

Rummel, R. (2005) GOCE-GRavitationsfeld-ANalyse Deutschland II-GOCE-GRAND II: Joint Proposal to Research Theme 2 "Observation of the System Earth from Space" of the BMBF/DFG Research and Development Programms " GEOTECHNOLOGIEN". Technical Report, Institute for Astronomical and Physical Geodesy, Technical University Munich.

Rummel, R., F. Sansò, M. van Gelderen, M. Brovelli, R. Koop, F. Miggliaccio, E. Schrama and F. Scerdote (1993) *Spherical Harmonic Analysis of Satellite Gradiometry.* Netherlands Geodetic Commission, New Series, 39.

Rummel, R., R. Koop and Th. Gruber (2004) High Level Processing Facility for GOCE: Products and Processing Strategy. In: *Proceedings of Second International GOCE User Workshop "GOCE, The Geoid and Oceanography".* ESA-ESRIN, Frascati.

Schuh, W.-D. (1984) *Analyse und Konvergenzbeschleunigung der Methode der konjugierten Gradienten bei geodätischen Netzen.* Mitteilungen der Geodätischen Institute der TU, Graz. Folge 49.

Schuh, W.-D. (1996) *Tailored Numerical Solution Strategies for the Global Determination of the Earth's Gravity Field.* Mitteilungen der Geodätischen Institute der TU Graz, Graz. Folge 81.

Schuh, W.-D. (2001) *Numerische Verfahren zur geodätischen Optimierung.* Skriptum. Theoretische Geodäsie, Universität Bonn.

Schuh, W.-D. (2003a) *GOCE Gravity Field Determination - Simulation Studies.* Status Seminar Bavarian State Mapping Agency (BLVA), Munich, June 12-13, 2003. Observation of the System Earth from Space. Geotechnologien, Science Report, No.3, 152–155.

Schuh, W.-D. (2003b) The Processing of Band-Limited Measurements; Filtering Techniques in the Least Squares Context and in the Presence of Data Gaps. Beutler, G., M.R. Drinkwater, R. Rummel and R. von Steiger (Ed.), *Earth Gravity Field from Space - From Sensors to Earth Sciences*, Band 108, Space Science Reviews, 67–78. ISSI Workshop, Bern (March 11-15,2002).

Schuh, W.-D., C. Boxhammer and C. Siemes (2006) Correlations, Variances, Covariances-From GOCE Signal to GOCE Products. In: *Proc. 3rd GOCE User Workshop, Frascati, November 2006.* Noordwijk.

Schwarz, H.R. (1970) Die Methode der konjugierten Gradienten in der Ausgleichsrechnung. *ZfV*, 95:130–140.

Sneeuw, N. (2000) *A Semi-Analytical Approach to Gravity Field Analysis from Satellite Observations.* Reihe C, 527. Deutsche Geodätische Kommission, München.

Tapley, B., J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, B. Gunter, Z. Kang, P. Nagel, R. Pastor, T. Pekker, S. Poole and F. Wang (2005) GGM02 - An improved Earth Gravity Field Model from GRACE. *J. Geodesy*, 79:467–478.

Tarantola, Albert (2005) *Inverse Problem Theory and Methods for Model Parameter Estimation.* Reihe C, 527. SIAM, Philadelphia.

The Math Works, Inc. (2006a) *MATLAB The Language of Technical Computing*, 6. edn., 2006.

The Math Works, Inc. (2006b) *Statistica Toolbox for Use with MATLAB*, 3. edn., 2006.

Tscherning, C., O. Anderson, D. Arabelos, E. Garminati, R. Foresberg, A. Gardi, P. Knudsen, J. N. Larsen, R. Sabadini and G. Strykowski (1999) Refinement of the Current Observation Requirements for GOCE. Technical Report, ESA-Project, ESA/ESTEC Contract No. 12339/NL/GD, National Survey and Cadastre, Denmark.

van Loon, J. P. and J. Kusche (2005) Stochastic Model Validation of Satellite Gravity Data: A Test with CHAMP Pseudo-Observations. In: Jekeli, C., L. Bastos and J. Fernandes (Ed.), *Gravity, Geoid and Space Missions*, Lecture Notes in Earth Sciences 129 (GGSM 2004, Porto). Springer, Berlin - Heidelberg - New York, 24–29.

Xu, P., Y. Shen and Y. Fukuda (2006) Variance Component Estimation in Linear Inverse Ill-posed Models. *J. Geodesy*, 80:69–81.

Acknowledgment

This dissertation was completed during my engagement with the Institute of Geodesy and Geoinformation at the University of Bonn.

This research was financially supported by the BMBF Geotechnologien programs GOCE-GRAND I (Grant 03F0329C) and GOCE-GRAND II (Grant 03F0421B). Calculations were made possible through the computational services provided by the John von Neumann Computing Institute at Jülich (project 1827).

My heartfelt thanks are due to Prof. Dr. techn. W.–D. Schuh for his committed supervision and trust and the open working environment granted to me during my time at the institute. I would also like to thank PD. Dr.–Ing. J. Kusche for his continuous assistance as co-reviewer, as well as Prof. Dr.–Ing., Dr.–Ing. E.h. mult. K. R. Koch (em.) for his helpful feedback and discussion.

I thank all my colleagues at the institute for their cooperation and the collegial working atmosphere which also contributed to the success of this work. Particular thanks are due to Boris Kargoll for his numerous reviews of this thesis. Not least I thank my wife Taghrid and my children Khaled and Salma for their support and understanding throughout my research endeavors.

In der Schriftenreihe des Instituts für Geodäsie und Geoinformation
der Rheinischen Friedrich-Wilhelms-Universität Bonn sind erschienen:

Heft 7 2008	Hamza Alkhatib On Monte Carlo Methods
Heft 6 2008	Klaus Borchard Annäherungen an Städtebau und Raumentwicklung
Heft 5 2008	Jens Jähnke Zur Teilmarktbildung beim Landerwerb der öffentlichen Hand
Heft 4 2008	Atef Abd-Elhakee Makhloof The Use of Topographic Isostatic Mass Information
Heft 3 2008	Markus Vennebusch Singular Value Decomposition and Cluster Analysis
Heft 2 2007	Christian Beder Grouping Uncertain Oriented Projective Geometric Entities
Heft 1 2007	Klaus Börger Geodäsie und Quantenphysik

Vertrieb: Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation
· Bibliothek ·
Nußallee 17
53115 Bonn

Tel.: +49 (0)228 73-3566
Fax: +49 (0)228 73-2988

Internet: http://www.igg.uni-bonn.de

ISSN 1864-1113