Randomized Algorithms for Solving Large Scale Nonlinear Least Squares Problems

by

Farbod Roosta-Khorasani

B.Sc., The University of British Columbia, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2015

Abstract

This thesis presents key contributions towards devising highly efficient stochastic reconstruction algorithms for solving large scale inverse problems, where a large data set is available and the underlying physical systems is complex, e.g., modeled by partial differential equations (PDEs).

We begin by developing stochastic and deterministic dimensionality reduction methods to transform the original large dimensional data set into the one with much smaller dimensions for which the computations are more manageable. We then incorporate such methods in our efficient stochastic reconstruction algorithms.

In the presence of corrupted or missing data, many of such dimensionality reduction methods cannot be efficiently used. To alleviate this issue, in the context of PDE inverse problems, we develop and mathematically justify new techniques for replacing (or filling) the corrupted (or missing) parts of the data set. Our data replacement/completion methods are motivated by theory in Sobolev spaces, regarding the properties of weak solutions along the domain boundary.

All of the stochastic dimensionality reduction techniques can be reformulated as Monte-Carlo (MC) methods for estimating the trace of a symmetric positive semi-definite (SPSD) matrix. In the next part of the present thesis, we present some probabilistic analysis of such randomized trace estimators and prove various computable and informative conditions for the sample size required for such Monte-Carlo methods in order to achieve a prescribed probabilistic relative accuracy.

Although computationally efficient, a major drawback of any (randomized) approximation algorithm is the introduction of "uncertainty" in the overall procedure, which could cast doubt on the credibility of the obtained results. The last part of this thesis consist of uncertainty quantification of stochastic steps of our approximation algorithms presented earlier. As a result, we present highly efficient variants of our original algorithms where the degree of uncertainty can easily be quantified and adjusted, if needed.

The uncertainty quantification presented in the last part of the thesis is an application of our novel results regarding the maximal and minimal tail probabilities of non-negative linear combinations of gamma random variables which can be considered independently of the rest of this thesis.

Preface

Parts of this thesis have been published in four co-authored papers plus a submitted (and arXived) one. As a rule, the main contributor appears first in the author lists of these papers. My supervisor has actively participated in the writing phase of all publications.

Chapter 3 (including parts of Chapter 2) has been published as [119], and a version of Chapter 4 appeared as [118]. These papers significantly expand the line of research begun in [46] in several directions. I am responsible for several of the new ideas and for the theory in [118], as well as for the entire implementation (which was written from scratch based in part on a previous effort by Kees van den Doel) and for carrying out all the numerical tests.

Chapter 5 has been published as [116], while a paper described in Chapters 6 and 7 has appeared as [117]. I conceived the ideas that have led to this part of the research work, and these have been refined through discussions with my collaborators. I formulated and proved all the theorems in [116] and [117], and have also implemented and carried out all the numerical examples in these papers.

Chapter 8 corresponds to [19]. My contributions consist of discussions, editing, and preparation of some of the numerical examples.

Table of Contents

List of Tables

List of Figures

List of Acronyms

- LS/NLS: Least Squares/Nonlinear Least Squares
- SS: Simultaneous Sources
- RS: Random Subset
- PDE: Partial Differential Equation
- TSVD: Truncated Singular Value Decomposition
- ML/MAP: Maximum Likelihood/Maximum a Posteriori
- GN: Gauss Newton
- PCG: Preconditioned Conjugate Gradient
- DOT: Diffuse Optical Tomography
- QPAT: Quantitative photo-Acoustic Tomography
- DC Resistivity: Direct Current resistivity
- EIT: Electrical Impedance Tomography
- i.i.d: Independent and Identically Distributed
- PDF: Probability Density Function
- CDF: Cumulative Distribution Function
- MC: Monte-Carlo
- QMC: Quasi MC
- SPSD: Symmetric Positive Semi-Definite

Notation

- Bold lower case letters: vectors
- Regular upper case letters: matrices or scalar random variables
- A^T: matrix transpose
- $tr(A)$: matrix trace
- $\mathcal{N}(0, \Sigma)$: normal distribution with mean 0 and covariance matrix Σ
- $\mathbb{I}$: identity matrix
- $\mathbb{R}^{m \times n}$: $m \times n$ real matrix
- $\mathbf{e}_k$: k^{th} column of identity matrix
- $\mathbf{1}$: vector of all 1's
- exp: exponential function
- $\mathbb{E}$: Expectation
- Var: Variance
- Pr: Probability
- $X \sim Gamma(\alpha, \beta)$: gamma distributed r.v parametrized by shape α and rate β
- f_X: probability density function of r.v X
- F_X: cumulative distribution function of r.v X
- $\|\mathbf{v}\|_2$, $\|A\|_2$: vector or matrix two norm

- $\|A\|_F$: matrix Frobenius norm
- ∇: gradient
- $\nabla\cdot$: divergence
- Δ: Laplacian
- Δ_S: Laplace-Beltrami operator
- $\delta(\mathbf{x}-\mathbf{x}_0)$: Dirac delta distribution centered at $\mathbf{x}_0$
- Ω: PDE domain
- $\partial\Omega$: boundary of the PDE domain
- $\bar{\Gamma}$: closure of Γ w.r.t subspace topology
- X°: interior of the set X w.r.t subspace topology
- C^k: space of k times continuously differentiable functions
- C^β: space of Hölder continuous functions with the exponent β
- L_p: space of L_p function
- $\|f\|_{L_p}$: L_p norm of the function f
- Lip: space of Lipschitz functions
- H^k: Sobolev space of H^k function
- $\mathbf{u}$: solution of the discretized PDE
- $\mathbf{q}$: discretized input to the PDE (i.e., right hand side)
- $\mathbf{d}_i$: measurement vector for the i^{th} experiment
- $\boldsymbol{\eta}_i$: noise incurred at the i^{th} experiment
- $\mathbf{f}_i$: forward operator for the i^{th} experiment

- J_i: Jacobian of $\mathbf{f}_i$
- P_i: projection matrix for the i^{th} experiment
- ϕ: misfit
- $\widehat{\phi}$: approximated misfit
- s: total number of experiments or dimension of SPSD matrix in Chapters 5 and 7
- k: iteration counter for GN algorithm
- n_k: sample size at k^{th} iteration for GN algorithm
- n: sample size for MC trace estimators or total number of i.i.d gamma r.v's in Chapter 6
- r: Maximum PCG iterations or rank of the matrix
- $\mathbf{m}$: model to be recovered
- $\delta\mathbf{m}$: update direction used in each GN iteration
- ρ: stopping criterion tolerance
- ε: trace estimation relative accuracy tolerance
- δ: probability of failure of ε-accurate trace estimation
- α: Tikhonov regularization parameter

Acknowledgments

Starting from the first year of grad school, I have had the honor of being supervised by one of the greatest numerical analysts of our time, Prof. Uri Ascher. Without his guidance, insight and, most importantly, patience, this thesis would not have been possible. On a personal level, he helped my family and myself in times of hardship and I am forever indebted to him.

I would also like to give special thanks to my knowledgeable and brilliant supervisory committee members, Prof. Chen Greif and Prof. Eldad Haber. They have always been my greatest supporters, and have never hesitated to graciously offer their help and expert advice along the way. I will always be grateful for their kindness.

I am also thankful to Prof. Gábor J. Székely, Dr. Adriano De Cezaro, Prof. Michael Friedlander, Dr. Kees van den Doel, and Dr. Ives Macêdo for all their help and many fruitful discussions.

I am also grateful to all the CS administrative staff, especially Joyce Poon, for ever so diligently taking care of all the paperwork throughout the years and making the process run smoothly.

A special thanks to my wonderful friends and colleagues Yariv Dror Mizrahi, Kai Rothauge and Iain Moyles for creating many good memories during these years.

Finally and most importantly, I would like to thank my biggest cheerleader, my wife, Jill. She never stopped believing in me, even when I was the closest to giving up. She stood by my side from day one and endured all the pain, anxiety, and frustration that is involved in finishing a PhD, right along with me. Jill, I love you!

To

My Queen, Jill,

&

My Princess, Rosa.

Chapter 1

Introduction

Inverse problems arise often in many applications in science and engineering. The term "inverse problem" is generally understood as the problem of finding a specific physical property, or properties, of the medium under investigation, using indirect measurements. This is a highly important field of applied mathematics and scientific computing, as to a great extent, it forms the backbone of modern science and engineering. Examples of inverse problems can be found in various fields within medical imaging (e.g., [10, 12, 28, 100, 136]) and several areas of geophysics including mineral and oil exploration (e.g., [20, 35, 102, 120]). For many of these problems, in theory, having many measurements is crucial for obtaining credible reconstructions of the sought physical property, i.e., the model. For others where there is no theory, it is a widely accepted working assumption that having more data can only help (at worst not hurt) the quality of the recovered model. As a consequence, there has been an exponential growth in the ability to acquire large amounts of measurements (i.e., many data set) in short periods of time. The availability of "big data", in turn, has given rise to some new rather serious challenges regarding the potentially high computational cost of solving such large scale inverse problems. As the ability to gather larger amounts of data increases, the need to devise algorithms to efficiently solve such problems becomes more important. Here is where randomized algorithms have shown great success in reducing the computational costs of solving such large scale problems. More specifically, dimensionality reduction algorithms transform the original large dimensional problem into a smaller size problem where the effective solution methods can be used. The challenge is to devise methods which yield credible reconstructions but at much lower costs. The main purpose of this thesis is to propose, study and analyze various such highly efficient reconstruction algorithms in the context of large scale least squares problems. Henceforth, the terms "model" and "parameter function" are interchangeably used to refer to the sought

physical property or properties of the medium under investigation.

1.1 Large Scale Data Fitting Problems

Inverse problems can often be regarded as data fitting problems where the objective is to recover an unknown parameter function such that the misfit (i.e., the distance, in some norm, between predicted and observed data) is to within a desirable tolerance, which is mostly dictated by some prior knowledge on measurement noise.

Generally speaking (and after possibly discretization of the continuous problem), consider the system

$$\mathbf{d}_i = \mathbf{f}_i(\mathbf{m}) + \boldsymbol{\eta}_i,\ i = 1, 2, \ldots, s, \tag{1.1}$$

where $\mathbf{d}_i \in \mathbb{R}^l$ is the measured data obtained in the i^{th} experiment, $\mathbf{f}_i = \mathbf{f}_i(\mathbf{m})$ is the known forward operator (or data predictor) for the i^{th} experiment arising from the underlying physical system, $\mathbf{m} \in \mathbb{R}^{l_m}$ is the sought-after parameter vector[1], and $\boldsymbol{\eta}_i$ is the noise incurred in the i^{th} experiment. The total number of experiments, or the size of the data sets, is assumed large: $s \gg 1$; this is what is implied by "large scale" or "large dimensional problem". The goal of data fitting is to find (or infer) the unknown model, $\mathbf{m}$, from the measurements $\mathbf{d}_i$, $i = 1, 2, \ldots, s$, such that

$$\sum_{i=1}^{s} \|\mathbf{f}_i(\mathbf{m}) - \mathbf{d}_i\| \leq \rho,$$

where ρ is usually related to noise, and the chosen norm can be problem-dependent. Generally, this problem can be ill-posed. Various approaches, including different regularization techniques, have been proposed to alleviate this ill-posedness; see, e.g., [9, 52, 135]. Most regularization methods consist of incorporating some *a priori* information on $\mathbf{m}$. Such information may be in the form of expected physical properties of the model in terms, for example, of constraints on the size, value or the smoothness.

In the presence of large amounts of measurements, i.e., $s \gg 1$, and when computing $\mathbf{f}_i$, for each i, is expensive, the mere evaluation of the misfit function may become computationally prohibitive. As such any reconstruction algorithm involving (1.1) becomes intractable. The

[1] The parameter vector $\mathbf{m}$ often arises from a parameter function in several space variables projected onto a discrete grid and reshaped into a vector.

goal of this thesis is to devise reconstruction methods to alleviate this problem and recover a credible model, $\mathbf{m}$, efficiently.

1.1.1 Assumptions on the Forward Operator

In this thesis, we consider a special class of data fitting problems where the forward operators, $\mathbf{f}_i$ in (1.1), satisfy the following assumptions.

(A.1) The forward operators, $\mathbf{f}_i$, have the form

$$\mathbf{f}_i(\mathbf{m}) = \mathbf{f}(\mathbf{m}, \mathbf{q}_i), \quad i = 1, \ldots, s, \tag{1.2}$$

where $\mathbf{q}_i$ is the input in the i^{th} experiment. In other words, the i^{th} measurement, $\mathbf{d}_i$, is made after injecting the i^{th} input (or source) $\mathbf{q}_i$ into the system. Thus, for an input $\mathbf{q}_i$, $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$ predicts the i^{th} measurement, given the underlying model $\mathbf{m}$.

(A.2) For all sources, we have $\mathbf{q}_i \in \mathbb{R}^{l_q}, \forall i$, and $\mathbf{f}$ is linear in $\mathbf{q}$, i.e., $\mathbf{f}(\mathbf{m}, w_1\mathbf{q}_1 + w_2\mathbf{q}_2) = w_1\mathbf{f}(\mathbf{m}, \mathbf{q}_1) + w_2\mathbf{f}(\mathbf{m}, \mathbf{q}_2)$. Alternatively, we write $\mathbf{f}(\mathbf{m}, \mathbf{q}) = G(\mathbf{m})\mathbf{q}$, where $G \in \mathbb{R}^{l \times l_q}$ is a matrix that depends, potentially non-linearly, on the sought $\mathbf{m}$.

(A.3) Evaluating $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$ for each input, $\mathbf{q}_i$, is computationally expensive and is, in fact, the bottleneck of computations.

1.1.2 A Practical Example

An important class of inverse problems for which Assumptions (A.1) - (A.3) are often valid, is that of large scale partial differential equation (PDE) inverse problems with many measurements. Such nonlinear parameter function estimation problems involving PDE constraints arise often in science and engineering. The main objective in solving such inverse problems is to find a specific model which appears as part of the underlying PDE. For several instances of these PDE-constrained inverse problems, large amounts of measurements are gathered in order to obtain reasonable and credible reconstructions of the sought model. Examples of such problems include electromagnetic data inversion in mining exploration (e.g., [48, 69, 108, 110]), seismic data inversion in oil exploration (e.g., [56, 81, 115]), diffuse optical tomography (DOT)

(e.g., [11, 29]), quantitative photo-acoustic tomography (QPAT) (e.g., [63, 140]), direct current (DC) resistivity (e.g., [46, 71, 72, 111, 128]), and electrical impedance tomography (EIT) (e.g., [33, 39, 47]). For such applications, it has been suggested that many well-placed experiments yield practical advantage in order to obtain reconstructions of acceptable quality.

Mathematical Model for Forward Operators

In the class of PDE-constrained inverse problems, upon discretization of the continuous problem, the sought model, $\mathbf{m}$, is a discretization of the function $m(\mathbf{x})$ in two or three space dimensions. Furthermore, the forward operator involves an approximate solution of a PDE, or more generally, a system of PDEs. We write this in discretized form as

$$L(\mathbf{m})\mathbf{u}_i = \mathbf{q}_i, \quad i = 1, \ldots, s, \tag{1.3}$$

where $\mathbf{u}_i \in \mathbb{R}^{l_u}$ is the i^{th} field, $\mathbf{q}_i \in \mathbb{R}^{l_u}$ is the i^{th} source, and L is a square matrix discretizing the PDE plus appropriate side conditions. Furthermore, there are given projection matrices P_i such that

$$\mathbf{f}_i(\mathbf{m}) = \mathbf{f}(\mathbf{m}, \mathbf{q}_i) = P_i\mathbf{u}_i = P_iL^{-1}(\mathbf{m})\mathbf{q}_i \tag{1.4}$$

predicts the ith data set. In other words, the matrix P_i projects the field, $\mathbf{u}_i$, onto the locations in the domain where the i^{th} measurements are made. Note that the notation (1.3) reflects an assumption of linearity in $\mathbf{u}$ but not in $\mathbf{m}$. Assumptions (A.1) & (A.3) can be justified for the forward operator (1.4). However, if P_i's are different for each i, then the linearity assumption (A.2) does not hold. On the other hand, if the locations where the measurements are made do not change from one experiment to another, i.e., $P = P_i, \forall i$, then we get

$$\mathbf{f}(\mathbf{m}, \mathbf{q}_i) = PL^{-1}(\mathbf{m})\mathbf{q}_i, \tag{1.5}$$

and the linearity assumption (A.2) of $\mathbf{f}(\mathbf{m}, \mathbf{q})$ in $\mathbf{q}$ is satisfied. It should be noted that, under certain circumstances, if the P_i's are different across experiments, there are methods to transform the existing data set into the one where all sources share the same receivers. Different

such methods are discussed in [70, 96] as well as Chapter 4 of this thesis.

In the sequel, the cost of any reconstruction algorithm used on a PDE constrained inverse problem is measured by the total count of PDE solves, $L(\mathbf{m})^{-1}\mathbf{q}$, as solving this linear system for each $\mathbf{q}$ is assumed to be the bottleneck of the computations.

1.1.3 Assumptions on the Noise

The developments of the methods and algorithms presented in this thesis are done under one of the following assumptions on the noise. In what follows $\mathcal{N}$ denotes the normal distribution.

(N.1) The noise is independent and identically distributed (i.i.d) as $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \Sigma), \forall i$, where $\Sigma \in \mathbb{R}^{l\times l}$ is the symmetric positive definite covariance matrix.

(N.2) The noise is independent but *not* necessarily identically distributed, satisfying instead $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma_i^2\mathbb{I}), i = 1, 2, \ldots, s$, where $\sigma_i > 0$ are the standard deviations.

Henceforth, for notational simplicity, most of the algorithms and methods are presented for the special case of Assumption (N.1) with $\Sigma = \sigma\mathbb{I}$. However, all of these methods and algorithms can be readily extended to the more general cases in a completely straightforward manner.

1.2 Least Squares Formulation & Optimization

If we may assume that the noise satisfies[2] Assumption (N.1) with $\Sigma = \sigma\mathbb{I}$, the standard *maximum likelihood* (ML) approach, [123], leads to minimizing the ordinary LS misfit function

$$\phi(\mathbf{m}) := \sum_{i=1}^{s} \|\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\|_2^2 = \|F(\mathbf{m}) - D\|_F^2, \tag{1.6}$$

where $F(\mathbf{m})$ and D are $l \times s$ matrices whose i^{th} columns are, respectively, $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$ and $\mathbf{d}_i$, and $\|\cdot\|_F$ stands for the Frobenius norm. Hence, we obtain a misfit function for which the data fitting can be done in ℓ_2 sense. However, since the above inverse problem is typically ill-posed, a

[2] For notational simplicity, we do not distinguish between a random vector (e.g., noise) and its realization, as they are clear within the context in which they are used.

regularization functional, $R(\mathbf{m})$, is often added to the above objective, thus minimizing instead

$$\phi_{R,\alpha}(\mathbf{m}) := \phi(\mathbf{m}) + \alpha R(\mathbf{m}), \tag{1.7}$$

where α is a regularization parameter [9, 52, 135]. In general, this regularization term can be chosen using a priori knowledge of the desired model. The objective functional (1.7) coincides with the *maximum a posteriori (MAP)* formulation, [123]. Injection of the regularization (i.e., a priori knowledge), $R(\mathbf{m})$, on the sought-after model can also be done by formulating the problem as

$$\min_{\mathbf{m}} \; R(\mathbf{m}) \quad \text{s.t. } \phi(\mathbf{m}) \leq \rho \tag{1.8}$$

where ρ acts as the regularization parameter[3]. Note that the "meaning" of the regularization parameter ρ in (1.8) is more intuitive than α in (1.7), as ρ usually relates to noise and the maximum discrepancy between the measured and the predicted data. As such, determining ρ could be easier than α. Implicit regularization also exists in which there is no explicit term $R(\mathbf{m})$ in the objective [77, 78, 113, 114, 131, 133]. Various optimization techniques can be used on the (regularized) objective to decrease the value of the above misfit, (1.6), to a desired level (determined, e.g., by a given tolerance which depends on the noise level), thus recovering the sought-after model.

Let us suppose for now that the forward operators $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$, each involving a PDE solution, are given as in (1.5): see Appendix A and Section 3.3 for a specific instance, used for our numerical experiments. Next, consider the problem of reducing the value the misfit function $\phi(\mathbf{m})$ defined in (1.6) (what follows can be easily extended for the regularized objective function $\phi_{R,\alpha}(\mathbf{m})$ defined in (1.7)). With the sensitivity matrices

$$J_i(\mathbf{m}) = \frac{\partial \mathbf{f}_i}{\partial \mathbf{m}}, \quad i = 1, \ldots, s, \tag{1.9}$$

[3]Though for the rest of this thesis, we will not consider algorithms for solving the contained problem (1.8), the discussions regarding stopping criterion in the following chapters are directly relevant in any such algorithm.

we have the gradient

$$\nabla\phi(\mathbf{m}) = 2\sum_{i=1}^{s} J_i^T(\mathbf{f}_i(\mathbf{m}) - \mathbf{d}_i). \tag{1.10}$$

An iterative method such as modified Gauss-Newton (GN), L-BFGS, or nonlinear conjugate gradient ([41, 57, 109]) is typically designed to decrease the value of the objective function using repeated calculations of the gradient. Although the methods and issues under consideration here do not require a specific optimization method we employ variants of the GN method throughout this thesis, thus achieving a context in which to focus our attention on the new aspects of this work and enabling comparison to past efforts. In particular, the way in which the GN method is modified is important more generally; see Appendix A.3.

The GN iteration for (1.6) (or (1.7)) at the k^{th} iteration with the current iterate $\mathbf{m} = \mathbf{m}_k$, calculates the correction as the solution of the linear system

$$\left(\sum_{i=1}^{s} J_i^T J_i\right)\delta\mathbf{m} = -\nabla_{\mathbf{m}}\phi, \tag{1.11a}$$

followed by the update

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\delta\mathbf{m}. \tag{1.11b}$$

Here the step length, α_k, $0 < \alpha_k \leq 1$, is determined by a weak line search (using, say, the Armijo algorithm starting with $\alpha_k = 1$) ensuring sufficient decrease in $\phi(\mathbf{m}_{k+1})$ as compared to $\phi(\mathbf{m}_k)$.

Several nontrivial modifications are required to adapt this prototype method for our purposes, and these are described in context in Appendix A.3, resulting in a method we refer to as *stabilized GN*. This method replaces the solution of (1.11a) by r preconditioned conjugate gradient (PCG) inner iterations, which costs $2r$ solutions of the forward problem per iteration, for a moderate integer value r. Thus, if K outer iterations are required to obtain an acceptable solution then the total work estimate (in terms of the number of PDE solves) is approximated *from below* by

$$\text{Work Estimate} = 2(r+1)Ks. \tag{1.12}$$

This indicates that for $s \gg 1$, the computational costs can be rather prohibitive. In this thesis, we design and propose algorithms for lowering the above "Work Estimate" by reducing the size of the data set s used in each of the K iterations.

Note that an alternative method to GN such as L-BFGS would require only $r = 1$ in (1.12). However, the number of such iterations would be significantly higher. This point again does not affect the issues addressed here and is not pursued further.

1.2.1 Generalized Least Squares Formulation

Our assumption regarding the noise distribution leading to the ordinary LS misfit function (1.6), although standard, is quite simplistic. Under the more general assumptions (N.1) or (N.2) on the noise, described in Section 1.1.3, we can extend the ordinary LS misfit (1.6) to obtain generalized LS formulations. More specifically, under Assumption (N.1), the ML approach leads to minimizing the ℓ_2 misfit function

$$\phi_{(1)}(\mathbf{m}) := \sum_{i=1}^{s} \|C^{-1}\big(\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\big)\|_2^2 = \|C^{-1}\big(F(\mathbf{m}) - D\big)\|_F^2, \tag{1.13}$$

where $C \in \mathbb{R}^{l\times l}$ is any invertible matrix such that $\Sigma = CC^T$ (e.g., C can be the Cholesky factor of Σ). The matrices F and D are as in (1.6).

Similarly, Under Assumption (N.2), the ML approach yields the *weighted* LS misfit function

$$\phi_{(2)}(\mathbf{m}) := \sum_{i=1}^{s} \frac{1}{\sigma_i^2}\|\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i\|_2^2 = \|\big(F(\mathbf{m}) - D\big)C^{-1}\|_F^2. \tag{1.14}$$

where $C \in \mathbb{R}^{s\times s}$ denotes the diagonal matrix whose i^{th} diagonal element is σ_i.

Although the developments of methods and algorithms in this thesis is done using the simple misfit (1.6), they can be almost verbatim applied to the above more general misfits (1.13) and (1.14).

1.3 Thesis Overview and Outline

This thesis is organized into nine chapters. Following the present introductory chapter, in Chapter 2, we will review dimensionality reduction methods, both stochastic and deterministic,

to transform the original high dimensional problem, into a smaller and manageable size one. This is done either with approximating the misfit function (in the stochastic case) or approximating the data matrix (in the deterministic case). The common denominator in many of these dimensionality reduction methods is that they form fewer experiments by some combination of the original experiments, called *simultaneous sources* (SS). This smaller and newly formed set of experiments is then used in optimization iterations. The method of SS is only applicable when the linearity assumption (A.2) is justified. Under such assumption, the stochastic variants of SS methods provide accurate approximations to the misfit. However, in the absence of the linearity assumption (A.2), an alternative, more general and yet less accurate, approximation method named *random subset* (RS) can be used and will also be discussed in Chapter 2. Part of this chapter is taken from Roosta-Khorasani, van Den Doel and Ascher [119].

Efficient, practical and stochastic reconstruction algorithms based on these dimensionality reduction methods are presented in Chapter 3. Such dimensionality reduction methods always involve (random) sampling of the original measurements and as the iterations progress, this sample size might be required to grow. For these algorithms, novel stochastic mechanisms for controlling the growth of the number of such samples are proposed and justified. Our algorithms employ some variants of stabilized GN method, though other iterative methods can easily be incorporated as well. In addition to using such approximation methods in each GN iteration, we identify and justify two different purposes for using these approximations in our algorithm. Furthermore we show that these different purposes may well require different estimation methods. We show that if the linearity assumption (A.2) is justified, the reconstruction algorithms based on the SS methods are significantly more efficient than their counterpart using the RS method. The comparison among different variants and the overall efficacy of these reconstruction algorithms are demonstrated in the context of the famous DC resistivity problem. We present in details our methods for solving such inverse problems. These methods involve incorporation of a priori information such as piecewise smoothness, bounds on the sought conductivity surface, or even a piecewise constant solution. This chapter has appeared in [119].

Reconstruction algorithms based on the efficient SS methods, presented in Chapter 3, are only applicable if the linearity assumption (A.2) is valid. In situations where Assumption (A.2) is violated, such as missing or highly corrupted data, among all algorithmic variants described

in Chapter 3, only the one based on RS method can be used. However, as shown in Chapter 3, an algorithm employing the RS method requires more evaluations of the computationally expensive forward operators, $\mathbf{f}_i$'s, in order to obtain a credible reconstruction. Luckily, under certain circumstances, it is possible to transform the problem, by constructing a new set of measurements, for which Assumption (A.2) is restored and thus SS algorithms presented in Chapter 3 can be used. Such transformations, described in details in Chapter 4, are done by means of an approximation using an appropriately restricted gradient or Laplacian regularization, filling for the missing (or replacing the corrupted) data. Our data completion/replacement methods are motivated by theory in Sobolev spaces regarding the properties of weak solutions along the domain boundary. Results using the method of SS with the newly formed data set are then compared to those obtained by a more general but slower RS method which requires no modifications. This chapter has appeared as as Roosta-Khorasani, van den Doel and Ascher [118].

All of our randomized reconstruction algorithms presented in this thesis rely heavily upon some fundamental aspects such as dimensionality reduction methods, discussed in Chapter 2. This, within the context of LS formulations, amounts to randomized algorithms for estimating the trace of an implicit matrix using Monte Carlo (MC) methods. Chapter 5 represents a comprehensive study of the theory of MC implicit matrix trace estimators. Such a method approximates the trace of an SPSD matrix A by an average of n expressions of the form $\mathbf{w}^T(A\mathbf{w})$, with random vectors $\mathbf{w}$ drawn from an appropriate distribution. In Chapter 5, we prove, discuss and experiment with bounds on the number of realizations n required in order to guarantee a probabilistic bound on the relative error of the trace estimation upon employing Rademacher (Hutchinson), Gaussian and uniform unit vector (with and without replacement) probability distributions, discussed in Section 2.1.1. In total, one necessary and six sufficient bounds are proved, improving upon and extending similar estimates obtained in the seminal work of Avron and Toledo [22] in several dimensions. We first improve their bound on n for the Hutchinson method, dropping a term that relates to $rank(A)$ (hence proving a conjecture in [22]) and making the bound comparable with that for the Gaussian estimator. We further prove new sufficient bounds for the Hutchinson, Gaussian and the unit vector estimators, as well as a necessary bound for the Gaussian estimator, which depend more specifically on properties

of the matrix A. As such they may suggest for what type of matrices one distribution or another provides a particularly effective or relatively ineffective stochastic estimation method. This chapter has appeared as Roosta-Khorasani and Ascher [116].

Chapter 6 is a precursor of Chapter 7. Specifically, the theorems proved in Chapter 7 are applications of more general and novel results regarding extremal tail probabilities (i.e., maximum and minimum of the tail probabilities) of linear combinations of gamma distributed random variables, which are presented and proved in Chapter 6. Many distributions, such as chi-squared of arbitrary degree, exponential, and Erlang are special instances of gamma distribution. As such these results have a wide range of applications in statistics, engineering, insurance, actuarial science and reliability. These results have appeared as Roosta-Khorasani, Székely and Ascher [117] and can be considered independently of the rest of this thesis.

The main advantage of an efficient randomized reconstruction algorithms presented in Chapter 3 is the reduction of computational costs. However, a major drawback of any such algorithm is the introduction of "uncertainty" in the overall procedure. The presence of uncertainty in the approximation steps could cast doubt on the credibility of the obtained results. Hence, it may be useful to have means which allow one to adjust the cost and accuracy of such algorithms in a quantifiable way, and find a balance that is suitable to particular objectives and computational resources. In Chapter 7, eight variants of randomized algorithms in Chapter 3 are presented where the uncertainties in the major stochastic steps are quantified. This is done by incorporating similar conditions as those presented in Chapter 5 in our stochastic algorithms. However, the sufficient bounds derived in Chapter 5 are typically not tight enough to be practically useful. As such, in Chapter 7 and for the special case of Gaussian trace estimator, we prove *tight necessary* and *sufficient* conditions on the sample size for MC trace estimators. We show that these conditions are practically computable and yield small sample sizes, and hence, all variants of our proposed algorithm with uncertainty quantification are very practical and highly efficient. This chapter has appeared in [117].

The discussion regarding the probabilistic stopping criterion in Chapter 7 lead us to observe that issues discussed there can also arise in several other domains of numerical computations. Namely, in practical applications a precise value for a tolerance used in the related stopping criterion is rarely known; rather, only some possibly vague idea of the desired quality of the

numerical approximation is at hand. There are situations where treating such a tolerance as a "holy" constant can result in erroneous conclusions regarding the relative performance of different algorithms or the produced outcome of one such algorithm. Highlighting such situations and finding ways to alleviate these issues are important. This is taken up in Chapter 8 where we discuss three case studies from different areas of numerical computation, where uncertainty in the error tolerance value is revealed in different ways. Within the context of large scale problems considered in this thesis, we then concentrate on a probabilistic relaxation of the given tolerance. A version of this chapter has been submitted for publication as Ascher and Roosta-Khorasani [19].

Each of Chapters 3, 4, 5, and 7 of this thesis, includes a summary, conclusions and future work section related to that specific line of research or project. In Chapter 9, an overall summary is given and a few directions regarding possible future research, not mentioned in earlier chapters, are presented.

This thesis contains an appendix as well. In Appendix A, certain implementation details are given which are used throughout the thesis. Such details include discretization of the EIT/DC resistivity problem in two and three dimensions, injection of a priori knowledge on the sought parameter function via transformation functions in the original PDE, the overall discussion of a (stabilized) GN algorithm for minimization of the least squares objective, a short MATLAB code which is employed in Chapter 7 to compute the Monte-Carlo sample sizes used in matrix trace estimators, and finally the details of implementation and discretization of the total variation functional used in several numerical examples in this thesis.

Chapter 2

Dimensionality Reduction

Inverse problems of the form (1.1) for which the forward operators satisfy Assumptions (A.1) & (A.3), can be very expensive to solve numerically. This is so especially when $s \gg 1$ and many experiments, involving different combinations of sources and receivers, are employed in order to obtain reconstructions of acceptable quality. For example, the mere evaluation of the misfit function (the distance between predicted and observed data), $\phi(\mathbf{m})$ in (1.6), requires evaluation of all $\mathbf{f}(\mathbf{m}, \mathbf{q}_i), i = 1, \ldots, s$. In this chapter, we develop and assess dimensionality reduction methods, both stochastic and deterministic, to replace the original large data set by a smaller set of potentially modified measurements for which the computations are more manageable. Such dimensionality reduction methods always involve random or deterministic sampling of the experiments. In this chapter various such sampling techniques are discussed.

In problems where, in addition to (A.1) & (A.3), Assumption (A.2) also holds, efficient[4] dimensionality reduction methods consisting of stochastically or deterministically combining the experiments can be employed. In the stochastic case, this yields an unbiased estimator (i.e., approximation) of the misfit function. However, in the deterministic case experiments are approximated by projecting the original data set onto a smaller space where a newly formed and smaller set of experiments capture the essence of the original data set. Since in both of these approaches, the approximation is done through the mixing of the experiments, the resulting method, originating from the geophysics community, is generally named the method of simultaneous sources (SS) [25, 76].

However, in situations where Assumption (A.2) is violated, the SS method is no longer applicable. In such scenarios, an alternative approximation method can be used which essentially

[4]In the rest of this thesis, "efficiency" is measured with respect to the total number of evaluations of the computationally expensive forward operator, $\mathbf{f}(\mathbf{m}, \mathbf{q})$. For example, in PDE inverse problems, the efficiency is measured with respect to the number of PDE solves.

boils down to selecting, uniformly at random, a subset of experiments, and this selection is done without any mixing [46]. Such a method, in what follows, is called a random subset (RS) method. It will be shown that RS method also provides an unbiased estimator of the misfit and can be applied in a wider variety of situations, compared to SS method.

2.1 Stochastic Approximation to Misfit

Randomized algorithms that rely on efficiently approximating the misfit function $\phi(\mathbf{m})$ have been proposed and studied in [8, 22, 46, 71, 105, 134]. In effect, they draw upon estimating the trace of an implicit[5] SPSD matrix. To see this, consider the misfit (1.6) and let $B = B(\mathbf{m}) := F(\mathbf{m}) - D$. It can be shown that

$$\phi(\mathbf{m}) = \|B\|_F^2 = tr(B^T B) = \mathbb{E}(\|B\mathbf{w}\|_2^2), \tag{2.1}$$

where $\mathbf{w}$ is a random vector drawn from any distribution satisfying

$$\mathbb{E}(\mathbf{w}\mathbf{w}^T) = \mathbb{I}, \tag{2.2}$$

$tr(A)$ denotes the trace of the matrix A, $\mathbb{E}$ denotes the expectation and $\mathbb{I} \in \mathbb{R}^{s\times s}$ is the identity matrix. Hence, approximating the misfit function $\phi(\mathbf{m})$ in (1.6) is equivalent to approximating the corresponding matrix trace (or equivalently, approximating the above expectation). The standard approach for doing this is based on a Monte-Carlo method, where one generates n random vector realizations, $\mathbf{w}_j$, from any such suitable probability distribution and computes the empirical mean

$$\widehat{\phi}(\mathbf{m}, n) := \frac{1}{n}\sum_{j=1}^{n} \|B(\mathbf{m})\mathbf{w}_j\|_2^2 \approx \phi(\mathbf{m}). \tag{2.3}$$

Note that $\widehat{\phi}(\mathbf{m}, n)$ is an *unbiased estimator* of $\phi(\mathbf{m})$, as we have $\phi(\mathbf{m}) = \mathbb{E}(\widehat{\phi}(\mathbf{m}, n))$. Under Assumptions (A.1)-(A.3), if $n \ll s$ then this procedure yields a very efficient algorithm for

[5] By "implicit matrix" we mean that the matrix of interest is not available explicitly: only information in the form of matrix-vector products for any appropriate vector is available.

approximating the misfit (1.6), because

$$\sum_{i=1}^{s} \mathbf{f}(\mathbf{m}, \mathbf{q}_i) w_i = \mathbf{f}(\mathbf{m}, \sum_{i=1}^{s} \mathbf{q}_i w_i), \tag{2.4}$$

which can be computed with a single evaluation of $\mathbf{f}$ per realization of the random vector $\mathbf{w} = (w_1, \ldots, w_s)^T$.

In practice, one can choose any distribution for which (2.2) is satisfied. Some popular choices of distributions for $\mathbf{w}$ are described in details in Section 2.1.1.

2.1.1 Selecting a Sampling Method

There are a few possible choices of probability distributions for $\mathbf{w}$, among which the most popular ones are as follows.

(i) The Rademacher distribution [83] where the components of $\mathbf{w}$ are independent and identically distributed (i.i.d) with $Pr(w_i = 1) = Pr(w_i = -1) = \frac{1}{2}$ (referred to in what follows as *Hutchinson* estimator, in deference to [22, 86]).

(ii) The standard normal distribution, $\mathcal{N}(0, \mathbb{I})$, is another possible choice and is henceforth referred to as *Gaussian* estimator.

(iii) The unit vector distribution (in deference to [22]). Here, the vectors $\mathbf{w}_i$ in (2.3) are uniformly drawn from the columns of the scaled identity matrix, $\sqrt{s}\mathbb{I}$. Drawing these vectors can be done with or without replacement. Such estimator is called the *random subset* method.

Distributions (i) and (ii) give rise to popular methods of simultaneous random sources [53, 71, 81, 90, 115, 126]. The methods of SS, when the linearity assumption (A.2) holds, yield very efficient estimators, as shown in (2.4). It can also be easily shown that, for a given sample size n, the variance of the Hutchinson estimator is smaller than that of the Gaussian estimator. However, relying solely on variance analysis can be misleading in determining the relative merit of each of these estimators; this is discussed in more details in Chapter 5.

For an approximation using the unit vector distribution (iii), the linearity assumption (A.2) is no longer necessary: it boils down to selecting a random subset of the given experiments at

each iteration, rather than their weighted combination. Within the context of reconstruction algorithms for inverse problems, this estimator was first introduced in [46]. In the absence of Assumption (A.2), such RS estimator is the only one that can be applied. However, as will be shown in Chapters 3 and 4, when the methods of SS apply, they provide a much more efficient and accurate[6] approximation to the misfit, compared to RS method.

The objective is to be able to generate as few realizations of $\mathbf{w}$ as possible for achieving acceptable approximations to the misfit function. Estimates on how large n must be, for a given distribution, to achieve a prescribed accuracy in a probabilistic sense are derived in Chapter 5.

2.1.2 Approximation with Generalized Noise Assumption

The stochastic approximation methods described in Section 2.1 can be similarly applied for the more general misfit functions, described in Section 1.2.1, under the noise assumptions (N.1) or (N.2). More specifically, the Monte-Carlo approximation, $\widehat{\phi}_{(1)}(\mathbf{m}, n)$, of $\phi_{(1)}(\mathbf{m})$ in (1.13) is precisely as in (2.3) but with $B(\mathbf{m}) := C^{-1}\big(F(\mathbf{m}) - D\big)$. Similarly, with $B(\mathbf{m}) = (F(\mathbf{m}) - D)C^{-1}$, we can again apply (2.3) to obtain a similar Monte-Carlo approximation, $\widehat{\phi}_{(2)}(\mathbf{m}, n)$, of $\phi_{(2)}(\mathbf{m})$ in (1.14).

Now, if $n \ll s$ then the unbiased estimators $\widehat{\phi}_{(1)}(\mathbf{m}, n)$ and $\widehat{\phi}_{(2)}(\mathbf{m}, n)$ are obtained with a similar efficiency as $\widehat{\phi}(\mathbf{m}, n)$. In the sequel, for notational simplicity, we just concentrate on $\phi(\mathbf{m})$ and $\widehat{\phi}(\mathbf{m}, n)$, but all the results hold almost verbatim also for (1.13) and (1.14).

2.2 Deterministic Approximation to Data

An alternative to stochastically approximating the misfit, is to abandon randomization altogether, and instead select the mixing weights deterministically. Deterministic approaches for reducing the size of the original large data set have been proposed in [62, 68], which in effect are data compression approaches. These compression schemes remove redundancy in data, not through eliminating redundant data, but instead through some mixing of redundant data. Similar deterministic SS method to compress the data may be obtained upon applying truncated

[6] A less efficient estimator is the one for which more realizations of $\mathbf{w}$ are required to achieve a desirable accuracy with the same likelihood. A less accurate estimator is the one which, given the same sample size, is less likely to achieve a desirable accuracy.

singular value decomposition (TSVD) to the data re-cast as the $l \times s$ matrix D in (1.6), where in our context we have $s \gg l$. More specifically, as a pre-processing step, one can calculate the SVD decomposition as $D = U\Sigma V^T$, where $U \in \mathbb{R}^{l\times l}, V \in \mathbb{R}^{s\times l}$ are the unitary matrices and $\Sigma \in \mathbb{R}^{l\times l}$ is the diagonal matrix of singular values. Now, one can effectively obtain an approximation to the original D as $\widehat{D} = D\widehat{V} \in \mathbb{R}^{s\times n}$, where $\widehat{V}$ is a matrix consisting of the first n columns of V. As such, we can replace the original misfit with

$$\widetilde{\phi}(\mathbf{m}, n) := \frac{1}{n}\sum_{j=1}^{n} \|B(\mathbf{m})\mathbf{v}_j\|_2^2, \tag{2.5}$$

where $\mathbf{v}_j$ is the j^{th} column of V. It should be noted that unlike $\widehat{\phi}(\mathbf{m}, n)$ in (2.3), the new misfit $\widetilde{\phi}(\mathbf{m}, n)$ is not an unbiased estimator of the original misfit, $\phi(\mathbf{m})$, as here D is approximated and not $\phi(\mathbf{m})$.

If n is large enough, this approach should bring out the essence of what is in the data, especially when the current iterate is far from the solution of the inverse problem. This approach can also be seen as denoising the original data as it involves removing the components corresponding to small singular values. A plot of the singular values for a typical experiment (in the context of a DC resistivity problem) is depicted in Figure 2.1. The quick drop in the

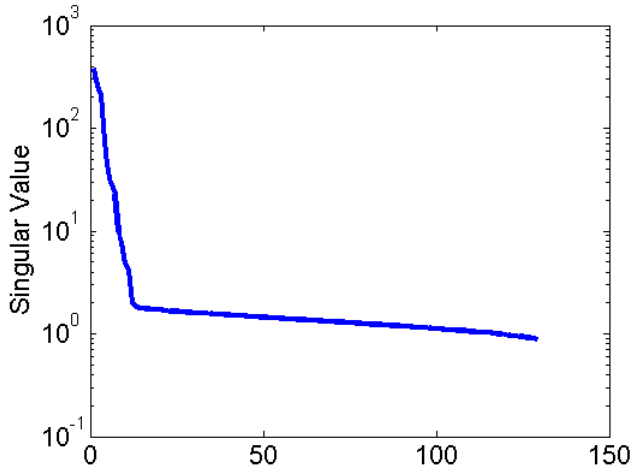


Figure 2.1: The singular values of the data used in Example 3.2 of Section 3.3.

singular values suggests that just a few singular vectors (the first columns of the orthogonal matrix U) represent the entire data well. This simple method is suitable when both dimensions

of the data matrix D are not too large. The SVD is performed only once prior to the inversion computations. Then, in the kth iteration of an optimization algorithm (such as stabilized GN in this thesis), the first few columns of V, corresponding to the largest singular values, provide fixed and deterministic weights for this SS method. Methods for choosing the number of such columns is discussed in Chapter 3.

2.3 GN Iteration on the Approximate Function

For the approximations (2.3) (or (2.5)), it is easy to return to a form like (1.6) and define sensitivity matrices $\widehat{J}_i = \widehat{J}_i(\mathbf{m}, n)$ and gradient $\nabla_{\mathbf{m}}\widehat{\phi} = \nabla_{\mathbf{m}}\widehat{\phi}(\mathbf{m}, n)$ analogously to (1.9) and (1.10), respectively. The GN iteration for (2.3) (or (2.5)) at a current iterate $\mathbf{m} = \mathbf{m}_k$ with n_k random weight vectors $\mathbf{w}_j$ in (2.3) (or deterministic weights $\mathbf{v}_j$ in (2.5)) calculates the correction as the solution of the linear system

$$\left(\sum_{i=1}^{n_k} \widehat{J}_i^T \widehat{J}_i\right) \delta\mathbf{m} = -\nabla_{\mathbf{m}}\widehat{\phi}, \tag{2.6a}$$

followed by the update

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \delta\mathbf{m}. \tag{2.6b}$$

Here, as in (1.11b), the step length, α_k, $0 < \alpha_k \leq 1$, is determined by a weak line search, ensuring sufficient decrease in approximation $\widehat{\phi}(\mathbf{m}_{k+1}, n)$ as compared to $\widehat{\phi}(\mathbf{m}_k, n)$.

Again, applying stabilized GN, as described in Appendix A.3, we see that, for K outer GN iterations, the total work estimate (in terms of the number of forward operator simulations) is approximated *from below* by

$$\text{Work Estimate} = 2(r+1)\sum_{k=1}^{K} n_k, \tag{2.7}$$

which indicates how keeping n_k small is important; see [46]. Comparing (2.7) with (1.12) shows that if $n_k \ll s$, $\forall k$, then the computational complexity is greatly reduced.

In Chapter 3, stochastic reconstruction algorithms are proposed which heavily rely on the

dimensionality reduction methods presented in Chapter 2. We also present randomized methods for controlling the sample size n_k used in these algorithms.

Chapter 3

Stochastic Reconstruction Algorithms

In this chapter, we present our stochastic algorithms for approximately minimizing (1.6) or (1.7), and discuss its novel elements. Here, we continue to make Assumptions (A.1) - (A.3); relaxation of the linearity assumption (A.2) is done in chapter (4). All these algorithms rely heavily on the dimensionality reduction techniques and sampling methods described in the previous chapter. Under Assumption (A.2), we have, as described in Chapter 2, four methods for sampling the original data set, which may be fused and compared.

As discussed earlier, the GN iteration (1.11) is computationally prohibitive. Consequently, as an alternative, one can consider the GN iteration (2.6) performed on the modified objective. If $n_k \ll s$, then these iterations can be performed more efficiently. In what follows, we assume for simplicity that the iterations are performed on the approximation (2.3) of the misfit (1.6) using dynamic regularization (or iterative regularization [46, 78, 132]) where the regularization is performed implicitly. We then incorporate the deterministic approximation (2.5) as well. Extension of the resulting algorithms to the case (1.7) is straightforward. Hence, the update direction, $\delta\mathbf{m}_k$, is calculated using the approximate misfit, $\widehat{\phi}(\mathbf{m}_k, n_k)$, defined in (2.3) where n_k is the sample size used for this approximation in the k^{th} iteration. However, since the iterations are performed on the modified objective function, the value of the original misfit might not necessarily be reduced. As such, any recovered model might not fit the original data appropriately. Thus, in each iteration, we need to check or assess whether the value of the original objective is also decreased using this new iterate. The challenge is to do this as well as check for termination of the iteration process with a minimal number of evaluations of the prohibitively expensive original misfit function (1.6).

The papers cited in Section 2.1.1 appear to assume one purpose for the approximate evaluation of the misfit function $\phi(\mathbf{m})$, and that is solely in (1.11a). In contrast, in Section 3.1, we identify two additional purposes for this task, and furthermore we show that these different purposes may well require different estimation methods. An additional fourth purpose will be introduced in Chapter 4 and further modified in Chapter 7.

The question of selecting the sample size n_k is addressed in Section 3.2. We propose two new algorithms which allow n_k to be very small for small k, and potentially significantly increase as the iteration approaches a solution. Algorithm 1 in Section 3.2.1 has the advantage of being simple, and it generates an exponentially increasing sequence of n_k values. Algorithm 2 in Section 3.2.2 uses cross validation in a manner similar to but not the same as that proposed in [46], and it generates a potentially more moderately increasing sequence of n_k values. The latter algorithm is particularly useful when s is "too large" in the sense that even near a satisfactory solution for the given inverse problem, far fewer than s experiments are required to satisfy the given error tolerances, a situation we qualitatively refer to as *embarrassing redundancy*. Within the context of these two algorithms, we compare the resulting weighting methods of Section 2.1.1 against the more generally applicable random subset method proposed in [46], and find that the three simultaneous sources methods are roughly comparable and are better than the random subset method by a factor of roughly 2 or more.

The computational work in Section 3.3 is done in the context of a DC resistivity problem. This is a simpler forward problem than low-frequency Maxwell's equations, and yet it reflects a similar spirit and general behaviour, allowing us to concentrate on the issues in focus here. A description of the implementation details is given in Appendices A.1, A.2, and A.3.

3.1 Two Additional Reasons for Unbiased Estimators

As described in Chapter 2, the original expensive misfit can be replaced by a computationally cheaper one, either stochastically or deterministically. One purpose of forming such a modified objective function is to be used in the iterations (2.6). Here we identify and justify two additional reasons for which stochastic approximate misfit (i.e., unbiased estimators) is used. A fourth purpose will be introduced in Chapter 4 and further modified in Chapter 7.

3.1.1 Cross Validation

It is desirable that after every iteration of any optimization method (such as GN), the value of the misfit (1.6) (or the regularized objective (1.7)) decreases (perhaps sufficiently). The mechanisms such as line-search are used to enforce such desired property. More specifically, it is desired that at the k^{th} iteration and after the update, we get

$$\phi(\mathbf{m}_{k+1}) \leq \kappa\phi(\mathbf{m}_k), \tag{3.1}$$

for some $\kappa \leq 1$, which indicates sufficient decrease in the misfit (or the objective $\phi_{R,\alpha}$ in the case of (1.7)). unfortunately, as argued before, such a test using the evaluation of the entire misfit is computationally prohibitive. However, since $\widehat{\phi}(\mathbf{m}_{k+1}, n_k)$ is an unbiased estimator of $\phi(\mathbf{m}_{k+1})$ with $n_k \ll s$, we can approximate the assessment of the updated iterate in terms of sufficient decrease in the objective function using a control set of random combinations of measurements. More specifically, at the k^{th} iteration with the new iterate $\mathbf{m}_{k+1}$, we test whether the condition

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \kappa\widehat{\phi}(\mathbf{m}_k, n_k) \tag{3.2}$$

(cf. (2.3)) holds for some $\kappa \leq 1$; The condition (3.2) is an independent, unbiased indicator of (3.1), and the success of (3.2) is an indicator that (3.1) is likely to be satisfied as well. However, for now, the test (3.2) is only left as a heuristic indicator of (3.1). As such, for the rest of this chapter, the sample size n_k used in (3.2) is chosen heuristically, but in Chapter (7), we will make this choice mathematically rigorous where the uncertainty in the test (3.2) is quantified. For example, we will develop tools to assess the probability of the success of (3.1), given the success of (3.2).

3.1.2 Stopping Criterion and Uncertainty Check

The usual stopping criterion for terminating the iterative process for data fitting (cf. Section 1.1) is to check, after the update in the k^{th} iteration, whether

$$\phi(\mathbf{m}_{k+1}) \leq \rho, \tag{3.3}$$

for a given tolerance ρ, with $\phi(\mathbf{m}_{k+1})$ not being much smaller than ρ. This is done either to avoid under-fitting/over-fitting of the noise, or as part of the explicit constraint such as in (1.8). For instance, consider the simplest case where for all experiments there is a Gaussian noise distribution for which the (same) standard deviation σ is known. Thus $D = D^* + \sigma\mathcal{N}$, where $D^* = F(\mathbf{m}^*)$, with $\mathcal{N}$ an $l \times s$ matrix of i.i.d Gaussians. We wish to terminate the algorithm when (1.6) falls below some multiple $\eta \gtrapprox 1$ of the noise level squared, i.e. $\sigma^2\|\mathcal{N}\|_F^2$. Since the noise is not known, following the celebrated Morozov *discrepancy principle* [52, 91, 107, 135], we replace $\|\mathcal{N}\|_F^2$ by its expected value, sl, obtaining

$$\rho = \eta\sigma^2 sl.$$

Unfortunately, however, the mere calculation of $\phi(\mathbf{m}_{k+1})$ requires s evaluations of the computationally expensive forward operators. We therefore wish to perform this check as rarely as possible. Fortunately, as discussed before, we have in $\widehat{\phi}(\mathbf{m}_{k+1}, n_k)$ a good, unbiased estimator of $\phi(\mathbf{m}_{k+1})$ with $n_k \ll s$. Thus, in the course of an iteration we can perform the relatively inexpensive *uncertainty check* whether

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \rho. \tag{3.4}$$

This is like the stopping criterion, but in expectation. If (3.4) is satisfied, it is an indication that (3.3) is likely to be satisfied as well, so we check the expensive (3.3) only then. Similarly to the condition (3.2), for the rest of this chapter, the sample size n_k used in (3.4) is chosen heuristically, but its selection is made mathematically rigorous in Chapter 7.

Note that, for uncertainty check and cross validation steps, since we want an unbiased estimator of the objective, the approximation should not be constructed deterministically, as described in Section 2.1.

3.2 Adaptive Selection of Sample Size

In this section we describe two algorithms for determining the sample size n_k in the k^{th} stabilized GN iteration. Algorithm 1 adapts n_k in a brute force manner. Algorithm 2 uses a cross

validation technique to avoid situations in which n_k grows too rapidly or becomes larger than necessary.

3.2.1 Sample Size Selection Using Uncertainty Checks

While the management strategy of n_k in this algorithm is simply to increase it so long as (3.3) is not met, its novelty lies in the fusion of different strategies for selecting the weight matrices at different stages of each iteration. Our algorithm consists of three main steps: (i) *data fitting* – a stabilized GN outer iteration (2.6); (ii) *uncertainty check* – a check for condition (3.4); and (iii) depending on the outcome of the uncertainty check, perform either *sample size adjustment* or *stopping criterion* check for termination.

Algorithm 1 Solve inverse problem using uncertainty check

Given: sources $Q = [\mathbf{q}_1\mathbf{q}_2\cdots\mathbf{q}_s]$, measurements $D = [\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_s]$, stopping criterion level ρ (i.e. the desired misfit) and initial guess $\mathbf{m}_0$.
Initialize: $\mathbf{m} = \mathbf{m}_0$, $n_0 = 1$.
for $k = 0, 1, 2, \cdots$ until termination **do**
- Choose n_k wight vectors stochastically (or deterministically) as described in Section 2.1 (or Section 2.2).
- **Fitting**: Perform one stabilized GN iteration approximating (2.6), with $n = n_k$.
- Choose n_k wight vectors stochastically as described in Section 2.1.
- **Uncertainty Check**: Compute (3.4) using $\mathbf{m}_{k+1}$ and the above n_k wight vectors.
if Uncertainty Check holds **then**
- **Stopping Criterion**: Compute (3.3) with $\mathbf{m}_{k+1}$. **Terminate** if it holds.
else
- **Sample Size Increase**: Increase n_{k+1}, for example set $n_{k+1} = \min(2n_k, s)$.
end if
end for

The exponential growth of the sample size in Algorithm 1 can be theoretically appealing, as such a schedule (unlike keeping n_k fixed) enables the general convergence theory of [60]. However, in cases where there is embarrassing redundancy in the set of experiments, it may not be desirable for the sample size to grow so rapidly and in an unchecked manner, as we could end up using far more experiments than what is actually needed. Some mechanism is required to control the growth of sample size, and one such is proposed next.

3.2.2 Adaptive Selection of Sample Size Using Cross Validation

For monitoring the growth of n_k more closely, one strategy is to compare the objective function ϕ at the current iterate to its value in the previous iterate, effectively checking for the test (3.1), and increase the sample size if there is no sufficient decrease. Unfortunately, evaluating the test (3.1) exactly defeats the purpose (in Section 3.3 typically the total cost of the reconstruction algorithm is small multiples of just one evaluation of ϕ). Fortunately, however, using the cross validation test (3.2), described in Section 3.1.1, we can get a handle of how the objective function is likely to behave. In other words, the role of the cross validation step within an iteration is to assess whether the true objective function at the current iterate has (sufficiently) decreased compared to the previous one. If this test fails, we deem that the current sample size is not sufficiently large to yield an update that decreases the original objective, and the fitting step needs to be repeated using a larger sample size. A method of this sort, based on "cross validation", is proposed in [46] together with a Random Subset method. Here we generalize and adapt this technique in the present context.

Thus, the following algorithm involves the steps of Algorithm 1, with an additional check for a sufficient decrease in the estimate (2.3) using another, independently selected weight matrix. Only in case that this test is violated, we increase the sample size.

Algorithm 2 Solve inverse problem using uncertainty check and cross validation

Given: sources $Q = [\mathbf{q}_1\mathbf{q}_2\cdots\mathbf{q}_s]$, measurements $D = [\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_s]$, stopping criterion level ρ (i.e. the desired misfit) and initial guess $\mathbf{m}_0$.
Initialize: $\mathbf{m} = \mathbf{m}_0$, $n_0 = 1$.
for $k = 0, 1, 2, \cdots$ until termination **do**
- Choose n_k wight vectors stochastically (or deterministically) as described in Section 2.1 (or Section 2.2).
- **Fitting**: Perform one stabilized GN iteration approximating (2.6), with $n = n_k$.
- Choose n_k wight vectors stochastically as described in Section 2.1.
if $\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \kappa\widehat{\phi}(\mathbf{m}_k, n_k)$, i.e., **Cross Validation** is satisfied **then**
- **Uncertainty Check**: Compute (3.4) using $\mathbf{m}_{k+1}$ and the above n_k wight vectors.
if Uncertainty Check holds **then**
- **Stopping Criterion**: Compute (3.3) with $\mathbf{m}_{k+1}$. **Terminate** if it holds.
end if
else
- **Sample Size Increase**: Increase n_{k+1}, for example set $n_{k+1} = \min(2n_k, s)$.
end if
end for

Note that our use of the term "cross validation" does not necessarily coincide with its usual meaning in statistics. But the procedure retains the sense of a control set and this name is convenient. The performance of Algorithm 2 is not automatically better than that of Algorithm 1. Indeed, it is possible to generate examples where cross validation is not necessary, as the computations in Section 3.3 demonstrate. However, it provides an important safety mechanism.

3.3 Numerical Experiments

3.3.1 The EIT/DC Resistivity Inverse Problem

Our experiments are performed in the context of solving the EIT/DC resistivity problem (e.g., [33, 39, 46, 47, 71, 72, 111, 128]). We have made this choice since exploiting many data sets currently appears to be particularly popular in exploration geophysics, and our examples, in this thesis, can be viewed as mimicking a DC resistivity setup. Note that the PDE model for EIT is identical to that of DC resistivity and the main difference is in experimental setup.

Consider a linear PDE of the form

$$\nabla \cdot (\mu(\mathbf{x})\nabla u) = q(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{3.5a}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, and μ is a conductivity function which may be rough (e.g., discontinuous) but is bounded away from 0: there is a constant $\mu_0 > 0$ such that $\mu(\mathbf{x}) \geq \mu_0$, $\forall \mathbf{x} \in \Omega$. This elliptic PDE is subject to the homogeneous Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0, \quad \mathbf{x} \in \partial\Omega. \tag{3.5b}$$

For Ω, we will consider a unit square or a unit cube. The inverse problem is to recover μ in Ω from sets of measurements of u on the domain's boundary for different sources q. This is a notoriously difficult problem in practice, so it may be useful to inject some a priori information on μ, when such is available, via a parametrization of $\mu(\mathbf{x})$ in terms of $m(\mathbf{x})$ using an appropriate

transfer function ψ as $\mu(\mathbf{x}) = \psi(m(\mathbf{x}))$. For example, ψ can be chosen so as to ensure that the conductivity stays positive and bounded away from 0, as well as to incorporate bounds, which are often known in practice, on the sought conductivity function. Some possible choices of function ψ are described in Appendix A.2.

3.3.2 Numerical Experiments Setup

The experimental setting we use is as follows: for each experiment i there is a positive unit point source at $\mathbf{x}_1^i$ and a negative sink at $\mathbf{x}_2^i$, where $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$ denote two locations on the boundary $\partial\Omega$. Hence in (3.5) we must consider sources of the form $\mathbf{q}_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_1^i) - \delta(\mathbf{x} - \mathbf{x}_2^i)$, i.e., a difference of two δ-functions.

For our experiments in 2D, when we place a source on the left boundary, we place the corresponding sink on the right boundary in every possible combination. Hence, having p locations on the left boundary for the source would result in $s = p^2$ experiments. The receivers are located at the top and bottom boundaries. No source or receiver is placed at the corners.

In 3D we use an arrangement whereby four boreholes are located at the four edges of the cube, and source and sink pairs are put at opposing boreholes in every combination, except that there are no sources on the point of intersection of boreholes and the surface, i.e., at the top four corners, since these four nodes are part of the surface where data values are gathered.

In the sequel we generate data $\mathbf{d}_i$ by using a chosen true model (or ground truth) and a source-receiver configuration as described above. Since the field u from (3.5) is only determined up to a constant, only voltage differences are meaningful. Hence we subtract for each i the average of the boundary potential values from all field values at the locations where data is measured. As a result each row of the projection matrix P has zero sum. This is followed by peppering these values with additive Gaussian noise to create the data $\mathbf{d}_i$ used in our experiments. Specifically, for an additive noise of 3%, say, denoting the "clean data" $l \times s$ matrix by D^*, we reshape this matrix into a vector $\mathbf{d}^*$ of length sl, calculate the standard deviation $\mathtt{sd} = .03\|\mathbf{d}^*\|/\sqrt{sl}$, and define $D = D^* + \mathtt{sd} * \mathtt{randn}(\mathtt{l}, \mathtt{s})$ using MATLAB's random generator function `randn`.

For all numerical experiments, the "true field" is calculated on a grid that is twice as fine as the one used to reconstruct the model. For the 2D examples, the reconstruction is done on

a uniform grid of size 64^2 with $s = 961$ experiments in the setup described above, and we used $\eta = 1.2$. For our 3D examples, the reconstruction is done on a uniform grid of size 17^3 with $s = 512$ experiments, and we set $\eta = 1.5$.

In Section 3.3.3 below, for the first three examples we use the transfer function (A.5) with $\mu_{\max} = 1.2 \max \mu(\mathbf{x})$, and $\mu_{\min} = .83 \min \mu(\mathbf{x})$. In the ensuing calculations we then "forget" what the exact $\mu(\mathbf{x})$ is. Further, we set the PCG iteration limit to $r = 20$, and the PCG tolerance to 10^{-3}. The initial guess is $\mathbf{m}_0 = \mathbf{0}$. Our last example is carried out using the level set method (A.6). Here we can set $r = 5$, significantly lower than above. The initial guess for the level set examples is displayed in Figure 3.1.

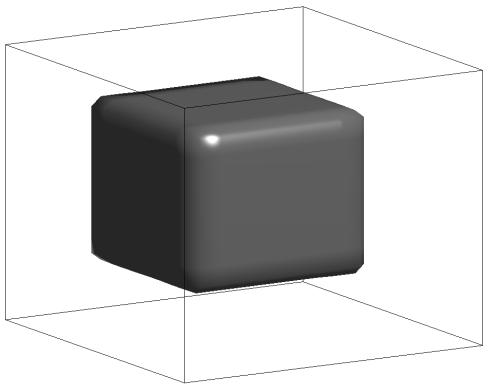


Figure 3.1: Example 3.4 – initial guess for the level set method.

In addition to displaying the log conductivities (i.e., $\log(\mu)$) for each reconstruction, we also show the log-log plot of misfit on the entire data (i.e. $\|F(\mathbf{m}) - D\|_F$) vs. PDE count. A table of total PDE counts (not including what extra is required for the plots) for each method is displayed. In this table, as a point of reference, we also include the total PDE count using the "plain vanilla" stabilized Gauss-Newton method which employs the entire set of experiments at every iteration.

We emphasize that, much as the rows in the work-unit table are easier to examine in order to determine which method is more efficient, it is important to also consult the corresponding data misfit plots, especially when the comparison is between relatively close quantities. This is so because one evaluation of the stopping criterion consumes a significant fraction of the total PDE count in each case, so an extra check that can randomly occur for a given experiment in

one method and not another may affect the work table far more than the misfit figures. In particular, the performance of the Hutchinson vs. Gauss estimators was found to be comparable in almost all experiments below.

Finally, before we turn to the numerical results let us comment on the expected general quality of such reconstructions. The quantifiers "good" and "acceptable" are relative concepts here. Our 3D experiments mimic DC geophysics surveys, where a reconstruction is considered good and acceptable if it generally looks like the true model, even remotely so. This is very different from the meaning of similar quantifiers in image denoising, for instance.

3.3.3 Numerical Experiments Comparing Eight Method Variants

In each of the four examples below we apply Algorithm 1 and Algorithm 2 with $\kappa = 1$; smaller values of κ would result in more aggressive increases of the sample size between one stabilized GN iteration and the next.

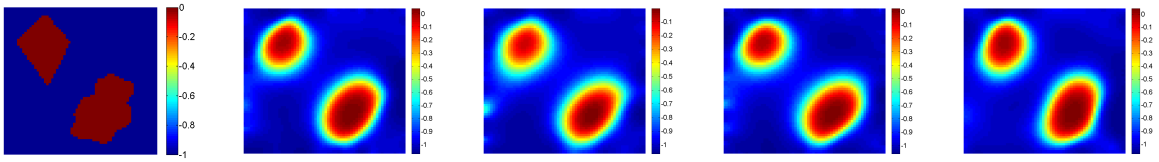
Furthermore, for convenience of cross reference, we gather all resulting eight work counts in Table 3.1 below. The corresponding entries of this table should be read together with the misfit plots for each example, though.

Example	**Alg**	Vanilla	Rand. Sub.	Hutch.	Gauss.	TSVD
3.1	1	86,490	3,788	1,561	1,431	2,239
	2		3,190	2,279	1,618	2,295
3.2	1	128,774	5,961	3,293	3,535	3,507
	2		3,921	2,762	2,247	2,985
3.3	1	36,864	6,266	1,166	1,176	1,882
	2		11,983	3,049	2,121	2,991
3.4	1	45,056	1,498	1,370	978	1,560
	2		2,264	1,239	896	1,656

Table 3.1: Work in terms of number of PDE solves for Examples 3.1–3.4. The "Vanilla" count is independent of the algorithms described in Section 3.2.

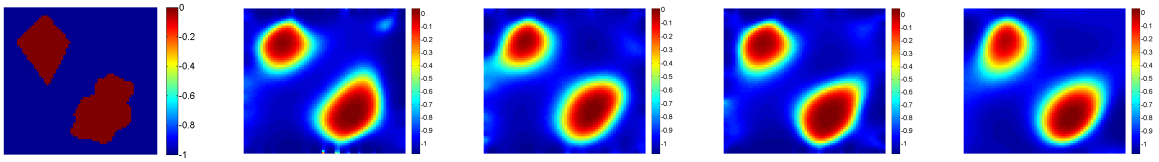
Example 3.1. *In this example, we place two target objects of conductivity $\mu_I = 1$ in a background of conductivity $\mu_{II} = 0.1$, and 3% noise is added to the data: see Figure 3.2(a). The reconstructions in Figures 3.2 and 3.3 are comparable.*

From Table 3.1 we see that all our methods offer vast improvements over the plain Vanilla method. Furthermore, the Random Subset method reduces the objective (i.e., misfit) function



(a) True model (b) Rand. Sub. (c) Gaussian (d) Hutchinson (e) TSVD

Figure 3.2: Example 3.1 – reconstructed log conductivity using Algorithm 1 and the four methods of Section 2.1.1.

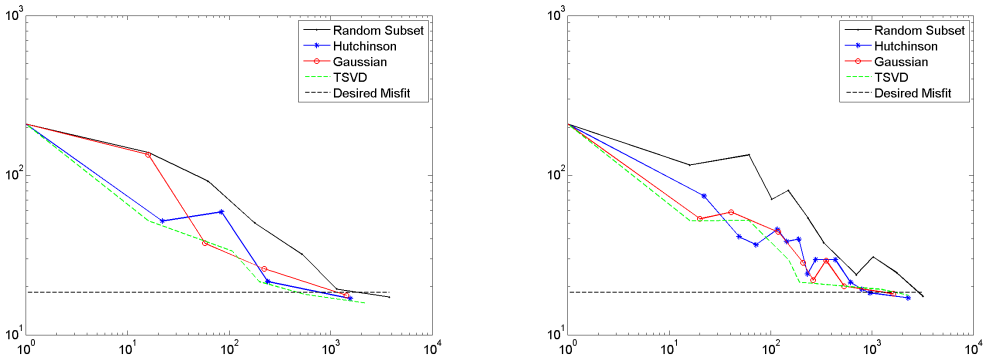


(a) True model (b) Rand. Sub. (c) Gaussian (d) Hutchinson (e) TSVD

Figure 3.3: Example 3.1 – reconstructed log conductivity using Algorithm 2 and the four methods of Section 2.1.1.

at a slower rate, requiring roughly twice as many PDE solves compared to the other methods of Section 2.1.1. Consulting also Figure 3.4, observe in addition that although the final PDE count for TSVD is slightly larger than for Hutchinson and Gaussian, it reduces the misfit at a faster, though comparable, rate. In fact, if we were to stop the iterations at higher noise tolerances then the TSVD method would have outperformed all others. In repeated similar tests, we have observed that the performance of Hutchinson and Gaussian is comparable.

Finally, comparing the first two rows of Table 3.1 and the subplots of Figure 3.4, it is clear

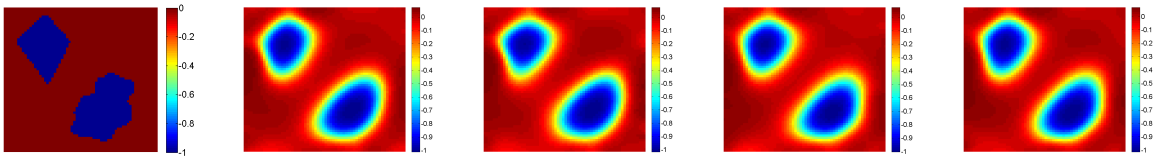


(a) Algorithm 1 (b) Algorithm 2

Figure 3.4: Data misfit vs. PDE count for Example 1.

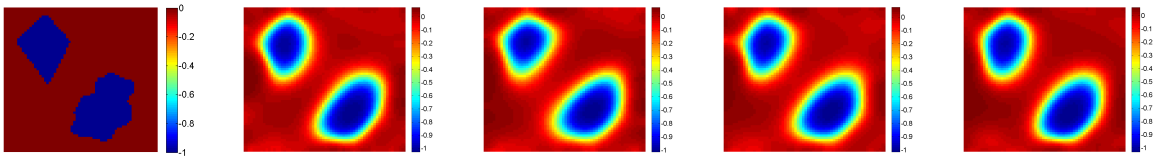
that the performance of Algorithms 1 and 2 is almost the same.

Example 3.2. *For this example, we merely swap the conductivities of the previous one, see Figure 3.5(a), and add the lower amount of* 1% *noise to the "exact data". The reconstruction results in Figures 3.5 and 3.6 are comparable. The performance indicators are gathered in Table 3.1 and Figure 3.7.*



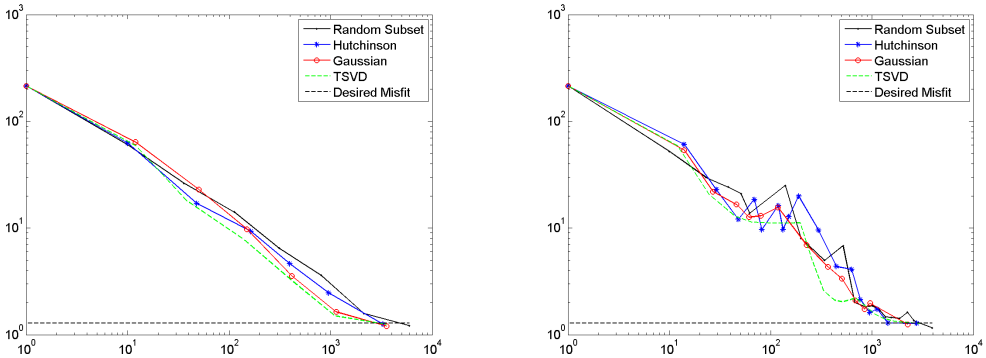
(a) True model (b) Rand. Sub. (c) Gaussian (d) Hutchinson (e) TSVD

Figure 3.5: Example 3.2 – reconstructed log conductivity using Algorithm 1 and the four methods of Section 2.1.1.



(a) True model (b) Rand. Sub. (c) Gaussian (d) Hutchinson (e) TSVD

Figure 3.6: Example 3.2 – reconstructed log conductivity using Algorithm 2 and the four methods of Section 2.1.1.



(a) Algorithm 1 (b) Algorithm 2

Figure 3.7: Data misfit vs. PDE count for Example 3.2.

Note that since in this example the noise is reduced compared to the previous one, more PDE solves are required. Similar observations to all those made for Example 3.1 apply here as well,

except that using the cross validation algorithm results in a notable reduction in PDE solves.

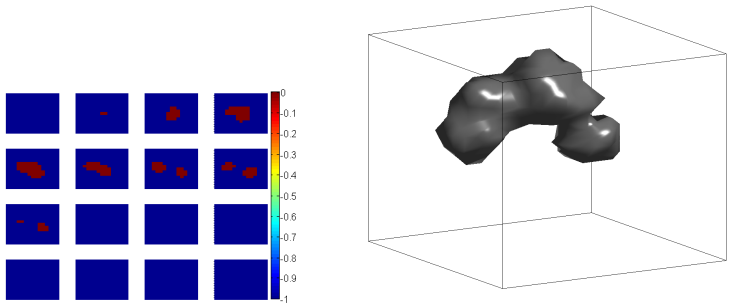
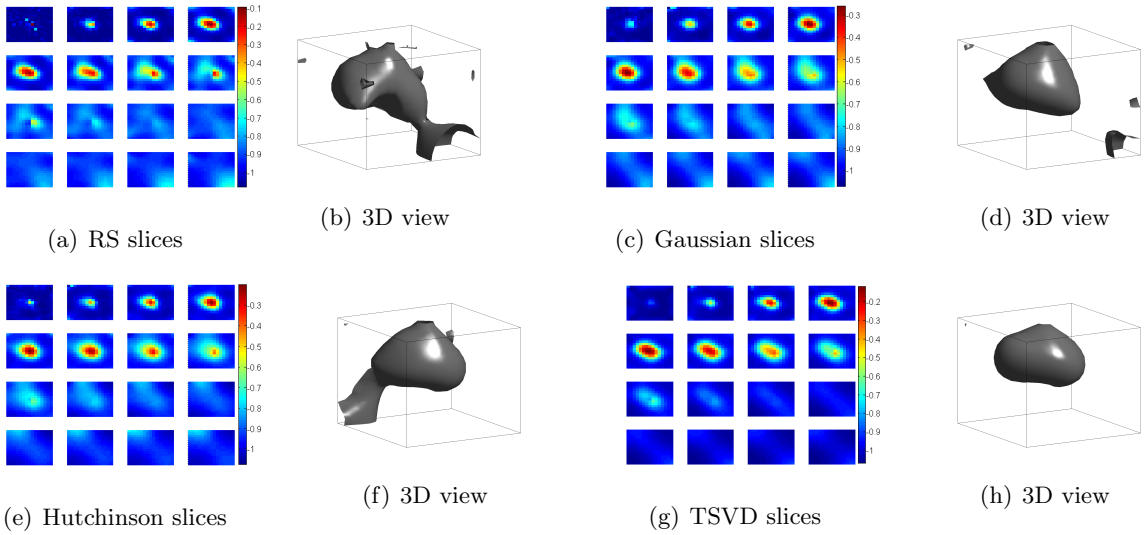


Figure 3.8: True Model for Examples 3.3 and 3.4. The left panel shows 2D equi-distant slices in the z direction from top to bottom, the right panel depicts the 3D volume.

Example 3.3. *In this 3D example, we place a target object of conductivity* $\mu_I = 1$ *in a background with conductivity* $\mu_{II} = 0.1$. *See Figure 3.8, whose caption also explains what other plots for 3D runs depict. A* 2% *noise is added to the "exact" data.*



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

Figure 3.9: Example 3.3 – reconstructed log conductivity for the 3D model using Algorithm 1 and (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.

The reconstruction quality for all eight method variants, see Figures 3.9 and 3.10, appears less clean than in our other examples; however, the methods are comparable in this regard, which allows us to concentrate on their comparative efficiency. It should be noted that no attempt was made here to "beautify" these results by post-processing, a practice not unheard of for hard geophysical inverse problems. Better reconstructions are obtained in the next example which

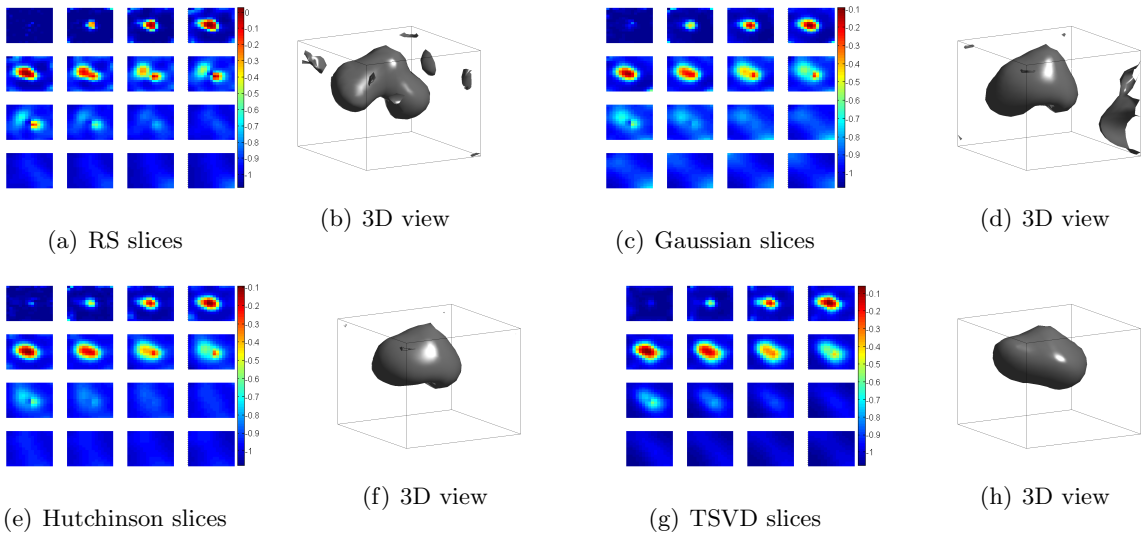
employs more a priori information and higher contrast.

In cases where more experiments are needed, the differences among the sampling methods are even more pronounced. This 3D example is one such case. All of the methods (excluding Vanilla) ended up using half of the experiments (i.e., $n_k \approx .5s$) before termination. Clearly, the Random Subset method is far outperformed by the other three, see Table 3.1 and Figure 3.13.

This is one example where Algorithm 1 achieves reconstructions of similar quality but more cheaply than Algorithm 2. This is so because in this case there is little embarrassing redundancy, i.e., larger sample sizes are needed to achieve the desired misfit, hence growing the sample size at a faster rate leads to an efficient algorithm. The sample size using cross validation grows more slowly, and relatively many GN iterations are performed using small sample sizes where each iteration decreases the misfit only slightly. These added iterations result in larger total PDE solve count.

Example 3.4. *This one is the same as Example 3.3, except that we assume that additional prior information is given, namely, that the sought model consists of piecewise constant regions with conductivity values μ_I and μ_{II}. This mimics a common situation in practice. So we reconstruct using the level set method* (A.6)*, which significantly improves the quality of the reconstructions: compare Figures 3.11 and 3.12 to Figures 3.9 and 3.10.*

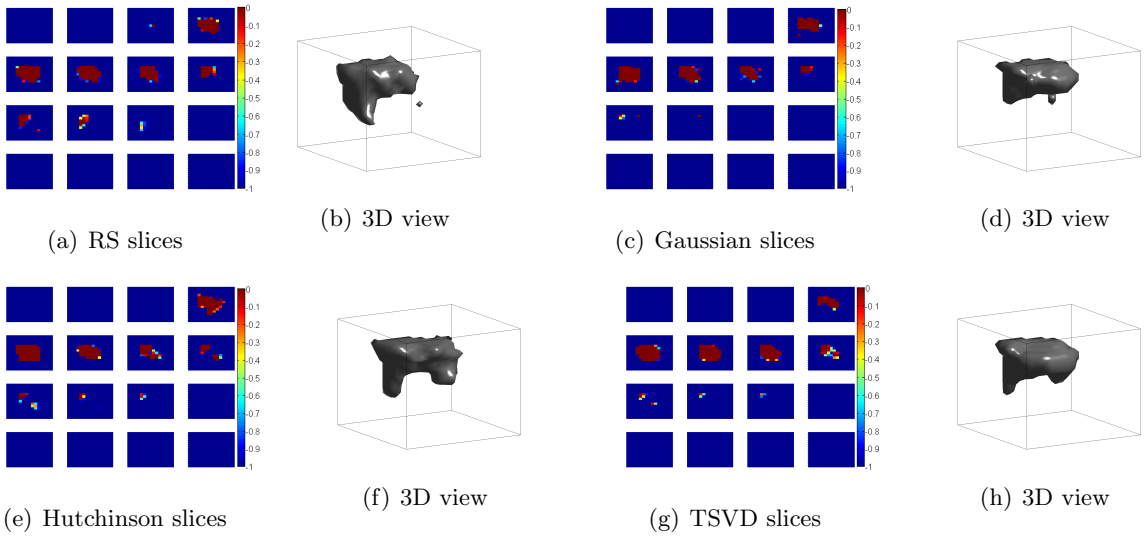
Here we observe less difference among the various methods. Specifically, in repeated experiments, the Random Subset method is no longer clearly the worst, see Table 3.1 and Figure 3.14. The numbers in the last row of Table 3.1 might be deceiving at first glance, as Random Subset seems to be worse than the rest; however, the graph of the misfit in Figure 3.14 reflects a more complete story. At some point in between the final PDE counts for Hutchinson and TSVD, the Random Subset misfit falls below the desired tolerance; however, the uncertainty check at that iterate results in a "false negative" which in turn does not trigger the stopping criterion. This demonstrates the importance of having a very good and reliable trace estimator in the uncertainty check. For all our eight algorithm variants and in all of our examples, we used the Hutchinson trace estimator for this purpose, as it has the smallest variance. And yet, one wrong estimate could result in additional, unnecessary GN iterations, leading to more PDE solves. False positives, on the other hand, trigger an unnecessary stopping criterion evaluation,



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

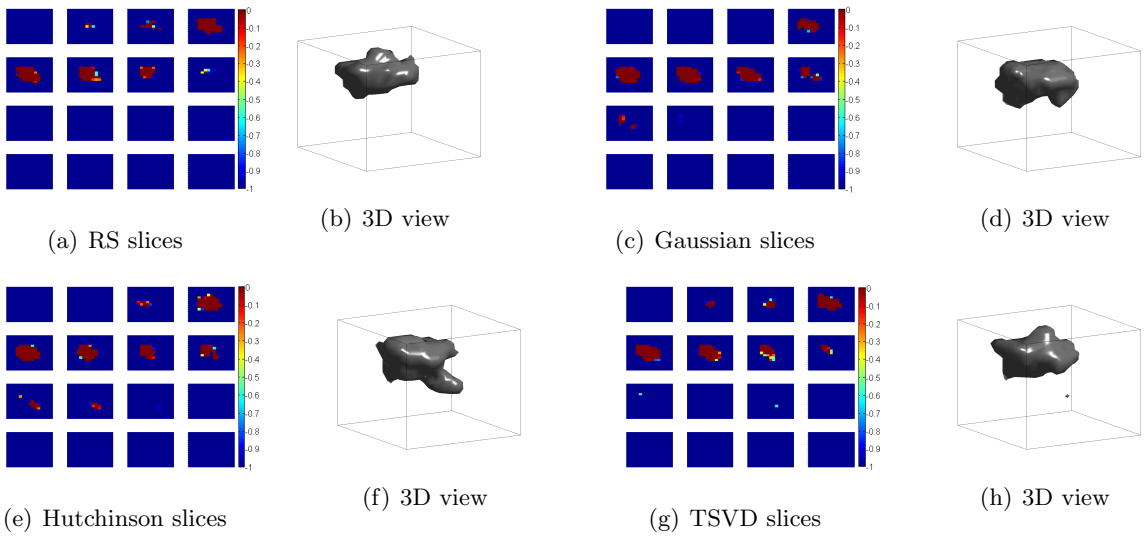
Figure 3.10: Example 3.3 – reconstructed log conductivity for the 3D model using Algorithm 2 and (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

Figure 3.11: Example 3.4 – reconstructed log conductivity for the 3D model using the level set method with Algorithm 1 and with (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.



(a) RS slices (b) 3D view (c) Gaussian slices (d) 3D view

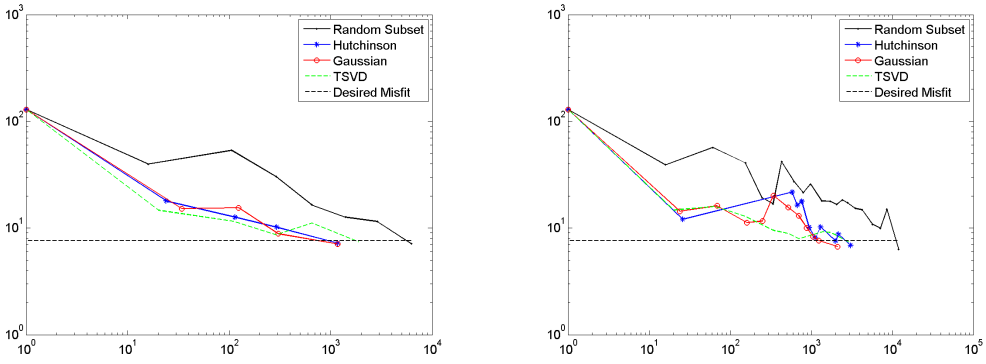
(e) Hutchinson slices (f) 3D view (g) TSVD slices (h) 3D view

Figure 3.12: Example 3.4 – reconstructed log conductivity for the 3D model using the level set method with Algorithm 2 and with (a,b) Random Subset, (c,d) Gaussian, (e,f) Hutchinson, and (g,h) TSDV.

which results in more PDE solves to calculate the misfit on the entire data set. For this example it was also observed that typically the Gaussian method outperforms Hutchinson by a factor of roughly 1.5.

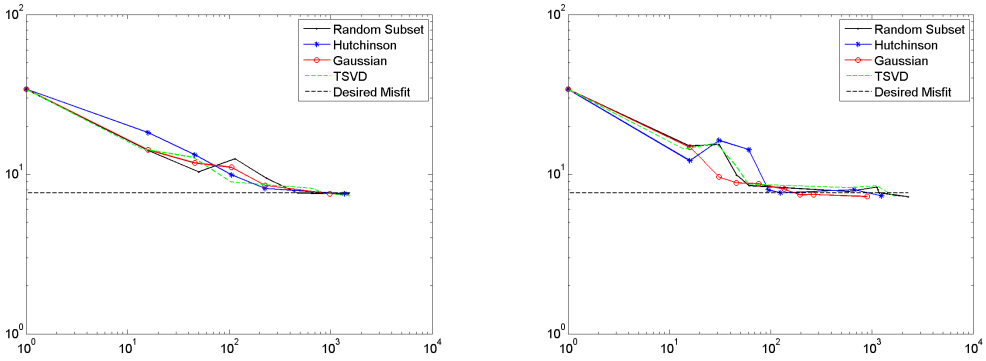
3.4 Conclusions

In this chapter we have developed and compared several highly efficient stochastic algorithms for the solution of inverse problems involving computationally expensive forward operators de-



(a) Algorithm 1 (b) Algorithm 2

Figure 3.13: Data misfit vs. PDE count for Example 3.3.



(a) Algorithm 1 (b) Algorithm 2

Figure 3.14: Data misfit vs. PDE count for Example 4.

scribed in Section 1.1 in the presence of many measurements or experiments s. Two algorithms for controlling the size $n_k \leq s$ of the data set in the k^{th} stabilized GN iteration have been proposed and tested. For each, four methods of sampling the original data set, three stochastic and one deterministic, discussed in Chapter 2, can be used, making for a total of eight algorithm variants. Our algorithms are known to converge under suitable circumstances because they satisfy the general conditions in [36, 60]. The numerical experiments are done specifically, in the context of DC resistivity.

It is important to emphasize that any of these algorithms is far better than a straightforward utilization of all experiments at each GN iteration. This is clearly borne out in Table 3.1. Note further that in order to facilitate a fair comparison we chose a fixed number of PCG inner iterations, ignoring the adaptive Algorithm 1 of [46], even though that algorithm can impact performance significantly. We also utilized for the sake of fair comparison a rather rigid (and expensive) stopping criterion; this will be eased off in future chapters. Further, we use the Hutchinson estimator for the uncertainty check in all methods, thus making them all stochastic. In particular, TSVD may not be used in (3.4) because it does not lead to an unbiased estimator for the objective function ϕ.

Inverse problems with many measurements arise in different applications which may have very different solution sensitivity to changes in the data (e.g., the full waveform inversion, although having other big difficulties in its solution process, is far less sensitive in this sense than DC resistivity). But in any case, it is an accepted working assumption that more data

can only help and not hurt the conditioning of the problem being solved. This then gives rise to the question whether our model reduction techniques may worsen the conditioning of the given problem. We have not observed any such effect in our experiments (and our "Vanilla" reconstructions in Section 3.3 are never better, or sharper, than the other, cheaper ones). In a sense it could be argued that a good model reduction algorithm actually covers approximately the same grounds as the full data problem, so it achieves a similar level of solution sensitivity to data.

As demonstrated in Examples 3.2 and 3.3, neither Algorithm 1 nor Algorithm 2 is always better than the other, and they often both perform well. Their relative performance depends on circumstances that can occasionally be distinguished before committing to calculations. Specifically, if there are relatively few data sets, as in Example 3.3, then Algorithm 1 is preferable, being both simpler and occasionally faster. On the other hand, if s is very large, the data having been massively calculated without much regard to experimental design considerations (as is often the case in geophysical exploration applications), then this may naturally lead to a case of embarrassing redundancy, and caution alone dictates using Algorithm 2.

The three methods of simultaneous sources, namely, Hutchinson, Gaussian and TSVD, are comparable (ignoring the cost of SVD computation), and no definitive answer can be given as to which is better for the model reduction. Further, especially when the level set method may not be used, we have found the methods of simultaneous sources to be consistently more efficient than the Random Subset method of [46], roughly by a factor of two or more. However, as mentioned before, SS methods can only be applied when the linearity assumption (A.2) is justified. In the absence of the Assumption (A.2), one is restricted to use the less efficient method of RS. Within the context of PDE constrained inverse problem (1.4), this means that the projection matrices P_i depend on i. That, in turn, raises the question whether the linearity assumption (A.2)can somehow be relaxed, thus allowing use of the faster methods of SS. This is the subject of Chapter 4.

Chapter 4

Data Completion

In Chapter 3, for the case where Assumptions (A.1) - (A.3) are valid, different methods of simultaneous sources are obtained by using different algorithms for this *model and data reduction* process. There, we have discussed and compared three such methods: (i) a Hutchinson random sampling, (ii) a Gaussian random sampling, and (iii) the deterministic truncated singular value decomposition (TSVD). We have found that, upon applying these methods, their performance was roughly comparable (although for just estimating the misfit function by (2.3), only the stochastic methods work well).

However, in situations where Assumption (A.2) is violated, none of the SS methods apply. Such situations arise, for example, when parts of measurements are missing or data is partially corrupted. In these cases, the random subset method can still be considered, where a random subset of the original experiments is selected at each iteration k, as the application of this method does not require the linearity assumption (A.2). However, as it was shown in Chapter 3, its performance is generally worse than the methods of simultaneous sources, roughly by a factor between 1 and 4, and on average about 2.[7] It is, in fact, possible to construct examples where a reconstruction algorithm using the RS method performs remarkably worse (much more than a factor of 4) than a similar SS based algorithm.

This brings us to the quest of the present Chapter, namely, to seek methods for the general case where Assumption (A.2) does not hold, which are as efficient as the simultaneous sources methods. The tool employed for this is to "fill in missing or replace corrupted data", thus restoring the linearity assumption (A.2). More specifically, the problem is transformed such that the original forward operators, $\mathbf{f}(\mathbf{m}, \mathbf{q}_i)$, are extended to the ones, which are linear in $\mathbf{q}$. For example, in PDE constrained inverse problems with the forward operators defined as

[7] The relative efficiency factor further increases if a less conservative criterion is used for algorithm termination, see Section 4.3.

in (1.4), i.e., P_i does depend on i, the goal is to replace P_i, for each i, by a common projection matrix P to the union of all receiver locations, $i = 1, \ldots, s$, effectively transforming the problem into the one with forward operators of the form (1.5). For the rest of the this chapter, we only consider the case of data completion, but application to data replacement is almost identical.

The prospect of such *data completion*, like that of casting a set of false teeth based on a few genuine ones, is not necessarily appealing, but is often necessary for reasons of computational efficiency. Moreover, applied mathematicians do a virtual data completion automatically when considering a Dirichlet-to-Neumann map, for instance, because such maps assume knowledge of the field u (see, e.g., (3.5)) or its normal derivative on the entire spatial domain boundary, or at least on a partial but continuous segment of it. Such knowledge of noiseless data at uncountably many locations is never the case in practice, where receivers are discretely located and some noise, including data measurement noise, is unavoidable. On the other hand, it can be argued that any practical data completion must inherently destroy some of the "integrity" of the statistical modeling underlying, for instance, the choice of iteration stopping criterion, because the resulting "generated noise" at the false points is not statistically independent of the genuine ones where data was collected.

Indeed, the problem of proper data completion is far from being a trivial one, and its inherent difficulties are often overlooked by practitioners. In this chapter we consider this problem in the context of the DC-resistivity problem (Section 4.1.3), with the sources and receivers for each data set located at segments of the boundary $\partial\Omega$ of the domain on which the forward PDE is defined. Forward operators are as defined in (1.4). Our data completion approach is to approximate or interpolate the given data directly in smooth segments of the boundary, while taking advantage of prior knowledge as to how the fields $\mathbf{u}_i$ must behave there. We emphasize that the sole purpose of our data completion algorithms is to allow the set of receivers to be shared among all experiments. This can be very different from traditional data completion efforts that have sought to obtain extended data throughout the physical domain's boundary or even in the entire physical domain. Our "statistical crime" with respect to noise independence is thus far smaller, although still existent.

We have tested several regularized approximations on the set of examples of Section 4.3, including several DCT [92], wavelet [101] and curvelet [49] approximations (for which we had

hoped to leverage the recent advances in compressive sensing and sparse ℓ_1 methods [50, 58]) as well as straightforward piecewise linear data interpolation. However, the latter is well-known not to be robust against noise, while the former methods are not suitable in the present context, as they are not built to best take advantage of the known solution properties. The methods which proved winners in the experimentation ultimately use a Tikhonov-type regularization in the context of our approximation, penalizing the discretized L_2 integral norm of the gradient or Laplacian of the fields restricted to the boundary segment surface. They are further described and theoretically justified in Section 4.2, providing a rare instance where theory correctly predicts and closely justifies the best practical methods. We believe that this approach applies to a more general class of PDE-based inverse problems.

In Section 4.1 we describe the inverse problem and the algorithm variants used for its solution. Several aspects arise with the prospect of data completion: which data – the original or the completed – to use for carrying out the iteration, which data for controlling the iterative process, what stopping criterion to use, and more. These aspects are addressed in Section 4.1.1. The resulting algorithm, based on Algorithm 2 of Chapter 3, is given in Section 4.1.2. The specific EIT/DC resistivity inverse problem described in Section 4.1.3 then leads to the data completion methods developed and proved in Section 4.2.

In Section 4.3 we apply the algorithm variants developed in the two previous sections to solve test problems with different receiver locations. The purpose is to investigate whether the SS algorithms based on completed data achieve results of similar quality at a cheaper price, as compared to the RS method applied to the original data. Overall, very encouraging results are obtained even when the original data receiver sets are rather sparse. Conclusions are offered in Section 4.4.

4.1 Stochastic Algorithms for Solving the Inverse Problem

The first two subsections below apply more generally than the third subsection. The latter settles on one application and leads naturally to Section 4.2.

Let us recall the acronyms for random subset (RS) and simultaneous sources (SS), used repeatedly in this section.

4.1.1 Algorithm Variants

To compare the performance of our model recovery methods with completed data, $\tilde{D}$, against corresponding ones with the original data, D, we use the framework of Algorithm 2 of Chapter 3. This algorithm consists of two stages within each GN iteration. The first stage produces a stabilized GN iterate, for which we use data denoted by $\hat{D}$. The second involves assessment of this iterate in terms of improvement and algorithm termination, using data $\bar{D}$. This second stage consists of evaluations of (2.3), in addition to (1.6). We consider three variants:

(i) $\hat{D} = D,\ \bar{D} = D$;

(ii) $\hat{D} = \tilde{D},\ \bar{D} = \tilde{D}$;

(iii) $\hat{D} = \tilde{D},\ \bar{D} = D$;

Note that only the RS method can be used in variant (i), whereas any of the SS methods as well as the RS method can be employed in variant (ii). In variant (iii) we can use a more accurate SS method for the stabilized GN stage and an RS method for the convergence checking stage, with the potential advantage that the evaluations of (2.3) do not use our "invented data". However, the disadvantage is that RS is potentially less suitable than Gaussian or Hutchinson precisely for tasks such as those in this second stage; see Chapter 3.

A major source of computational expense is the algorithm stopping criterion, which in Chapter 3 was taken to be (3.3), namely

$$\phi(\mathbf{m}_{k+1}) \leq \rho,$$

for a specified tolerance ρ. In Chapter 3, we deliberately employed this criterion in order to be able to make fair comparisons among different methods. However, the evaluation of ϕ for this purpose is very expensive when s is large, and in practice ρ is hardly ever known in a rigid sense. In any case, this evaluation should be carried out as rarely as possible. In Chapter 3, we addressed this by proposing a safety check, called "uncertainty check", which uses (2.3) as an unbiased estimator of $\phi(\mathbf{m})$ with $n_k \ll s$ realizations of a random vector from one of the distributions described in Section 2.1.1. Thus, in the course of an iteration we can perform the

relatively inexpensive uncertainty check (3.4), namely

$$\hat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \rho.$$

This is like the stopping criterion, but in expectation. If (3.4) is satisfied, it is an indication that (3.3) is likely to be satisfied as well, so we check the expensive (3.3) only then.

In the present chapter, we propose an alternative heuristic method of replacing (3.3) with another uncertainty check evaluation as in (3.4) with t_k realizations of the Rademacher random vector (NB the Hutchinson estimator has smaller variance than Gaussian). The sample size t_k can be heuristically set as

$$t_k = \min\big(s, \max\left(t_0, n_k\right)\big), \tag{4.1}$$

where $t_0 > 1$ is some preset minimal sample size for this purpose. Thus, for each algorithm variant (i), (ii) or (iii), we consider two stopping criteria, namely,

(a) the hard (3.3), and

(b) the more relaxed (3.4)+(4.1).

When using the original data D in the second stage of our general algorithm, as in variants (i) and (iii) above, since the linearity assumption (A.2) does not hold in the setting considered here, for efficiency reasons, one is restricted to the RS method as an unbiased estimator. However, when the completed data is used and, as a result, the linearity assumption (A.2) is restored, we can freely use the stochastic SS methods and leverage their rather better accuracy in order to estimate the true misfit $\phi(\mathbf{m})$. This is indeed an important advantage of data completion methods.

However, when using the completed data $\tilde{D}$ in the second stage of our general algorithm, as in variant (ii), an issue arises: when the data is completed, the given tolerance ρ loses its meaning and we need to take into account the effect of the additional data to calculate a new tolerance. Our proposed heuristic approach is to replace ρ with a new tolerance $\rho := (1 + c)\rho$, where c is the percentage of the data that needs to be completed expressed as a fraction. For example, if 30% of data is to be completed then we set $\rho := 1.3\rho$. Since the completed data after using (4.2) or (4.6) is smoothed and denoised, we only need to add a small fraction of the

initial tolerance to get the new one, and in our experience, $1 + c$ is deemed to be a satisfactory factor. We experiment with this less rigid stopping criterion in Section 4.3.

4.1.2 General Algorithm

Our general algorithm utilizes a stabilized Gauss-Newton (GN) method (see Chapter A.3 and [46]), where each iteration consists of two stages as described in Section 4.1.1. In addition to combining the elements described above, this algorithm also provides a schedule for selecting the sample size n_k in the k^{th} stabilized GN iteration. In Algorithm 3, variants (i), (ii) and (iii), and criteria (a) and (b), are as specified in Section 4.1.1.

Algorithm 3 Solve inverse problem using variant (i), (ii) or (iii), cross validation, and stopping criterion (a) or (b)

Given: sources Q, measurements $\hat{D}$, measurements $\bar{D}$, stopping tolerance ρ, decrease factor $\kappa < 1$, and initial guess $\mathbf{m}_0$.
Initialize: $\mathbf{m} = \mathbf{m}_0$, $n_0 = 1$.
for $k = 0, 1, 2, \ldots$ until termination **do**
- Choose n_k wight vectors stochastically as described in Section 2.1.
- **Fitting**: Perform one stabilized GN iteration, based on $\hat{D}$ and above weight, on (2.3).
- Choose two independent sets of n_k wight vectors stochastically as described in Section 2.1.

if $\hat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \kappa\hat{\phi}(\mathbf{m}_k, n_k)$, based on $\bar{D}$, using the above two sets of weights for each side of the inequality. i.e., **Cross Validation** holds **then**
- Choose n_k wight vectors stochastically as described in Section 2.1.
- **Uncertainty Check**: Compute (2.3) based on $\bar{D}$ using $\mathbf{m}_{k+1}$ and the above weights.

if (3.4) holds **then**
- **Stopping Criterion**:

if Option (a) selected and (3.3) holds **then**
terminate; otherwise set $n_{k+1} = n_k$.
else
Set $t_k = \min\left(s, \max\left(t_0, n_k\right)\right)$.
Choose t_k wight vectors stochastically as described in Section 2.1. **Terminate** if (3.4) holds using $\bar{D}$; otherwise set $n_{k+1} = n_k$.
end if
end if
else
- **Sample Size Increase**: for example, set $n_{k+1} = \min(2n_k, s)$.

end if
end for

4.1.3 The DC Resistivity Inverse Problem

For the forward problem, we consider the DC resistivity problem with a linear PDE of the form described in Section 3.3.1. In our numerical examples we again consider the simple domain $\Omega \subset \mathbb{R}^d$ to be the unit square or unit cube, and the sources $\mathbf{q}$ to be the differences of δ-functions; see details in Section 3.3. Since the receivers (where data values are measured) lie in $\partial\Omega$, in our data completion algorithms we approximate data along one of four edges in the 2D case or within one of six square faces in the 3D case. The setting of our experiments, which follows that used in Chapter 3, is more typical of DC resistivity than of the EIT problem.

For the inverse problem we introduce additional a priori information, when such is available, via a point-wise parametrization of $\mu(\mathbf{x})$ in terms of $m(\mathbf{x})$. For details of this, as well as the PDE discretization and the *stabilized* GN iteration used, we refer to Chapter 3, Appendix A and [46] and references therein.

4.2 Data Completion

Let $\Lambda_i \subset \partial\Omega$ denote the point set of receiver locations for the i^{th} experiment. Our goal here is to extend the data for each experiment to the union $\Lambda = \bigcup_i \Lambda_i \subseteq \partial\Omega$, the common measurement domain. To achieve this, we choose a suitable boundary patch $\Gamma \subseteq \partial\Omega$, such that $\Lambda \subset \bar{\Gamma}$, where $\bar{\Gamma}$ denotes the closure of Γ with respect to the boundary subspace topology. For example, one can choose Γ to be the interior of the convex hull (on $\partial\Omega$) of Λ. We also assume that Γ can be selected such that it is a simply connected open set. For each experiment i, we then construct an extension function v_i on $\bar{\Gamma}$ which approximates the measured data on Λ_i. The extension method can be viewed as an inverse problem, and we select a regularization based on knowledge of the function space that v_i (which represents the restriction of potential u_i to Γ) should live in. Once v_i is constructed, the extended data, $\tilde{\mathbf{d}}_i$, is obtained by restricting v_i to Λ, denoted in what follows by v_i^{Λ}. Specifically, for the receiver location $x_j \in \Lambda$, we set $[\tilde{\mathbf{d}}_i]_j = v_i(x_j)$, where $[\tilde{\mathbf{d}}_i]_j$ denotes the j^{th} component of vector $\tilde{\mathbf{d}}_i$ corresponding to x_j. Below we show that the trace of potential u_i to the boundary is indeed continuous, thus point values of the extension function v_i make sense.

In practice, the conductivity $\mu(\mathbf{x})$ in (3.5a) is often piecewise smooth with finite jump

discontinuities. As such one is faced with two scenarios leading to two approximation methods for finding v_i: (a) the discontinuities are some distance away from Γ; and (b) the discontinuities extend all the way to Γ. These cases result in a different a priori smoothness of the field v_i on Γ. Hence, in this section we treat these cases separately and propose an appropriate data completion algorithm for each.

Consider the problem (3.5). In what follows we assume that Ω is a bounded open domain and $\partial\Omega$ is Lipschitz. Furthermore, we assume that μ is continuous on a finite number of disjoint subdomains, $\Omega_j \subset \Omega$, such that $\bigcup_{j=1}^{N} \overline{\Omega}_j = \overline{\Omega}$ and $\partial\Omega_j \cap \overline{\Omega} \in C^{2,\alpha}$, for some $0 < \alpha \leq 1$, i.e., $\mu \in C^2(\overline{\Omega}_j),\ j = 1, \ldots, N$.[8] Moreover, assume that $q \in L_\infty(\Omega)$ and $q \in \mathrm{Lip}(\overline{\Omega}_j \cap \Omega)$, i.e., it is Lipschitz continuous in each subdomain; this assumption will be slightly weakened in Subsection 4.2.4.

Under these assumptions and for the Dirichlet problem with a $C^2(\partial\Omega)$ boundary condition, there is a constant γ, $0 < \gamma \leq 1$, such that $u \in C^{2,\gamma}(\overline{\Omega}_j)$ [88, Theorem 4.1]. In [98, Corollary 7.3], it is also shown that the solution on the entire domain is Hölder continuous, i.e., $u \in C^\beta(\overline{\Omega})$ for some β, $0 < \beta \leq 1$. Note that the mentioned theorems are stated for the Dirichlet problem, and here we assume a homogeneous Neumann boundary condition. However, in this case we have infinite smoothness in the normal direction at the boundary, i.e., C^∞ Neumann condition, and no additional complications arise; see for example [127]. So the results stated above would still hold for (3.5).

4.2.1 Discontinuities in Conductivity Are Away from Common Measurement Domain

This scenario corresponds to the case where the boundary patch Γ can be chosen such that $\Gamma \subset (\partial\Omega_j \cap \partial\Omega)$ for some j. Then we can expect a rather smooth field at Γ; precisely, $u \in C^{2,\gamma}(\overline{\Gamma})$. Thus, u belongs to the Sobolev space $H^2(\Gamma)$, and we can impose this knowledge in our continuous completion formulation. For the i^{th} experiment, we define our data completion function $v_i \in H^2(\Gamma) \cap C(\overline{\Gamma})$ as

$$v_i = \arg\min_{v} \ \frac{1}{2}\|v^{\Lambda_i} - \mathbf{d}_i\|_2^2 + \lambda \left\|\Delta_S v\right\|_{L_2(\Gamma)}^2, \tag{4.2}$$

[8] $\overline{X}$ denotes the closure of X with respect to the appropriate topology.

where Δ_S is the Laplace-Beltrami operator ([89, 124]) for the Laplacian on the boundary surface and v^{Λ_i} is the restriction of the continuous function v to the point set Λ_i. The regularization parameter λ depends on the amount of noise in our data; see Section 4.2.3.

We next discretize (4.2) using a mesh on Γ as specified in Section 4.3, and solve the resulting linear least squares problem using standard techniques.

Figure 4.1 shows an example of such data completion. The true field and the measured data correspond to an experiment described in Example 4.3 of Section 4.3. We only plot the profile of the field along the top boundary of the 2D domain. As can be observed, the approximation process imposes smoothness which results in an excellent completion of the missing data, despite the presence of noise at a fairly high level.

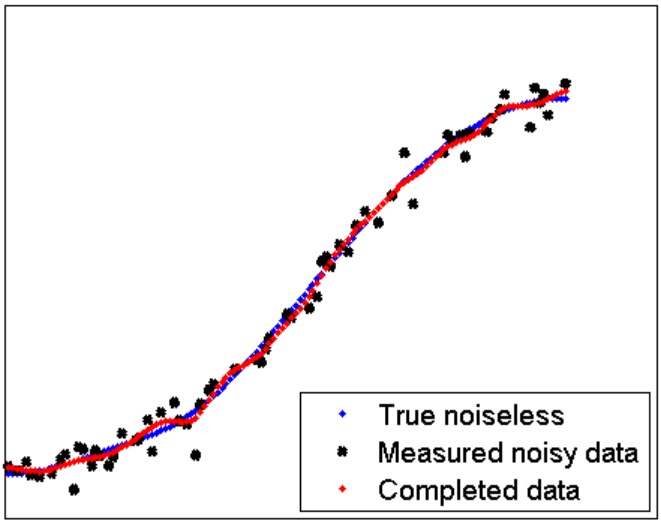


Figure 4.1: Completion using the regularization (4.2), for an experiment taken from Example 4.3 where 50% of the data requires completion and the noise level is 5%. Observe that even in the presence of significant noise, the data completion formulation (4.2) achieves a good quality field reconstruction.

We hasten to point out that the results in Figure 4.1, as well as those in Figure 4.2 below, pertain to differences in field values, i.e., the solutions of the forward problem u_i, and not those in the inverse problem solution shown, e.g., in Figure 4.5. The good quality approximations in Figures 4.1 and 4.2 generally form a necessary but not sufficient condition for success in the inverse problem solution.

4.2.2 Discontinuities in Conductivity Extend All the Way to Common Measurement Domain

This situation corresponds to the case in which Γ can only be chosen such that it intersects more than just one of the $(\partial\Omega \cap \partial\Omega_j)$'s. More precisely, assume that there is an index set $\mathcal{J} \subseteq \{1, 2, \cdots N\}$ with $|\mathcal{J}| = K \geq 2$ such that $\{\Gamma \cap (\partial\Omega \cap \partial\Omega_j)^\circ \ , \ j \in \mathcal{J}\}$ forms a set of disjoint subsets of Γ such that $\overline{\Gamma} = \bigcup_{j\in\mathcal{J}} \overline{\Gamma \cap (\partial\Omega \cap \partial\Omega_j)^\circ}$, where X° denotes the interior of the set X, and that the interior is with respect to the subspace topology on $\partial\Omega$. In such a case u, restricted to Γ, is no longer necessarily in $H^2(\Gamma)$. Hence, the smoothing term in (4.2) is no longer valid, as $\|\Delta_S u\|_{L_2(\Gamma)}$ might be undefined or infinite. However, as described above, we know that the solution is piecewise smooth and overall continuous, i.e., $u \in C^{2,\gamma}(\overline{\Omega}_j)$ and $u \in C^\beta(\overline{\Omega})$. The following theorem shows that the smoothness on Γ is not completely gone: we may lose one degree of regularity at worst.

Theorem 4.1. *Let U and $\{U_j |\ j = 1, 2, \ldots, K\}$ be open and bounded sets such that the U_j are pairwise disjoint and $\overline{U} = \bigcup_{j=1}^K \overline{U}_j$. Further, let $u \in C(\overline{U}) \cap H^1(U_j)\ \forall j$. Then $u \in H^1(U)$.*

Proof. It is easily seen that since $u \in C(\overline{U})$ and U is bounded, then $u \in L_2(U)$. Now, let $\phi \in C_0^\infty(U)$ be a test function and denote $\partial_i \equiv \frac{\partial}{\partial \mathbf{x}_i}$. Using the assumptions that the U_j's form a partition of U, u is continuous in $\overline{U}$, ϕ is compactly supported inside U, and the fact that the ∂U_j's have measure zero, we obtain

$$\begin{aligned}
\int_U u\partial_i\phi &= \int_{\overline{U}} u\partial_i\phi \\
&= \int_{\cup_{j=1}^K \overline{U_j}} u\partial_i\phi \\
&= \int_{(\cup_{j=1}^K U_j) \bigcup (\cup_{j=1}^K \partial U_j)} u\partial_i\phi \\
&= \int_{\cup_{j=1}^K U_j} u\partial_i\phi \\
&= \sum_{j=1}^K \int_{U_j} u\partial_i\phi \\
&= \sum_{j=1}^K \int_{\partial U_j} u\phi\nu_i^j - \sum_{j=1}^K \int_{U_j} \partial_i u\phi,
\end{aligned}$$

where ν_i^j is the i^{th} component of the outward unit surface normal to ∂U_j. Since $u \in H^1(U_j)\ \forall j$, the second part of the rightmost expression makes sense. Now, for two surfaces ∂U_m and ∂U_n such that $\partial U_m \cap \partial U_n \neq \emptyset$, we have $\nu_i^m(\mathbf{x}) = -\nu_i^n(\mathbf{x})\ \forall \mathbf{x} \in \partial U_m \cap \partial U_n$. This fact, and noting in addition that ϕ is compactly supported inside U, makes the first term in the right hand side vanish, i.e.,

$$\sum_{j=1}^{K} \int_{\partial U_j} u\phi\nu_i^j = 0.$$

We can now define the weak derivative of u with respect to $\mathbf{x}_i$ to be

$$v(\mathbf{x}) = \sum_{j=1}^{K} \partial_i u \mathcal{X}_{U_j}, \tag{4.3}$$

where $\mathcal{X}_{U_j}$ denotes the characteristic function of the set U_j. This yields

$$\int_U u\partial_i\phi = -\int_U v\phi. \tag{4.4}$$

Also

$$\|v\|_{L_2(U)} \leq \sum_{j=1}^{K} \|\partial_i u\|_{L_2(U_j)} < \infty, \tag{4.5}$$

and thus we conclude that $u \in H^1(U)$. □

If the assumptions stated at the beginning of this section hold then we can expect a field $u \in H^1(\Gamma) \cap C(\bar{\Gamma})$. This is obtained by invoking Theorem 4.1 with $U = \Gamma$ and $U_j = \Gamma \cap (\partial\Omega \cap \partial\Omega_j)^\circ$ for all $j \in \mathcal{J}$.

Now we can formulate the data completion method as

$$v_i = \arg\min_v \ \frac{1}{2}\|v^{\Lambda_i} - \mathbf{d}_i\|_2^2 + \lambda \left\|\nabla_S v\right\|_{L_2(\Gamma)}^2, \tag{4.6}$$

where v^{Λ_i} and λ are as in Section 4.2.1.

Figure 4.2 shows an example of data completion using the formulation (4.6), depicting the profile of v_i along the top boundary. The field in this example is continuous and only piecewise smooth. The approximation process imposes less smoothness along the boundary as compared to (4.2), and this results in an excellent completion of the missing data, despite a nontrivial

level of noise.

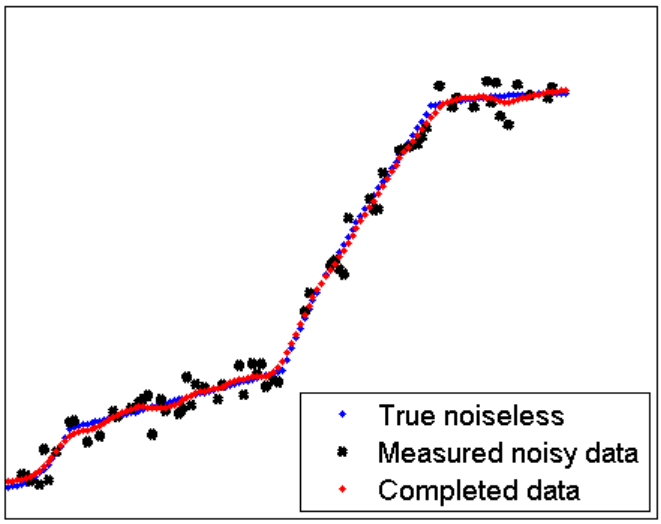


Figure 4.2: Completion using the regularization (4.6), for an experiment taken from Example 4.2 where 50% of the data requires completion and the noise level is 5%. Discontinuities in the conductivity extend to the measurement domain and their effect on the field profile along the boundary can be clearly observed. Despite the large amount of noise, data completion formulation (4.6) achieves a good reconstruction.

To carry out our data completion strategy, the problems (4.2) or (4.6) are discretized. This is followed by a straightforward linear least squares technique, which can be carried out very efficiently. Moreover, this is a preprocessing stage performed once, which is completed before the algorithm for solving the nonlinear inverse problem commences. Also, as the data completion for each experiment can be carried out independently of others, the preprocessing stage can be done in parallel if needed. Furthermore, the length of the vector of unknowns v_i is relatively small compared to those of u_i because only the boundary is involved. All in all the amount of work involved in the data completion step is dramatically less than one full evaluation of the misfit function (1.6).

4.2.3 Determining the Regularization Parameter

Let us write the discretization of (4.2) or (4.6) as

$$\min_{\mathbf{v}} \frac{1}{2}\|\hat{P}_i\mathbf{v} - \mathbf{d}_i\|_2^2 + \lambda\|L\mathbf{v}\|_2^2, \tag{4.7}$$

where L is the discretization of the surface gradient or Laplacian operator, $\mathbf{v}$ is a vector whose length is the size of the discretized Γ, $\hat{P}_i$ is the projection matrix from the discretization of Γ to Λ_i, and $\mathbf{d}_i$ is the i^{th} original measurement vector.

Determining λ in this context is a textbook problem; see, e.g., [135]. Viewing it as a parameter, we have a linear least squares problem for $\mathbf{v}$ in (4.7), whose solution can be denoted $\mathbf{v}(\lambda)$ as

$$\mathbf{v}_i(\lambda) = (\hat{P}_i^T\hat{P}_i + \lambda L^T L)^{-1}\hat{P}_i^T\mathbf{u}_i$$

Now, in the simplest case, which we assume in our experiments, the noise level for the i^{th} experiment, η_i, is known, so one can use the discrepancy principle to pick λ such that

$$\left\|\hat{P}_i\mathbf{v}(\lambda) - \mathbf{d}_i\right\|_2^2 \leq \eta_i. \tag{4.8}$$

Numerically, this is done by setting equality in (4.8) and solving the resulting nonlinear equation for λ using a standard root finding technique.

If the noise level is not known, one can use the generalized cross validation (GCV) method ([65]) or the L-curve method ([79]). For example, GCV function can be written as

$$GCV(\lambda) = \frac{\|\hat{P}_i\mathbf{v} - \mathbf{u}_i\|_2^2}{tr(\mathbb{I} - \hat{P}_i(\hat{P}_i^T\hat{P}_i + \lambda L^T L)^{-1}\hat{P}_i^T)^2},$$

where tr denotes the standard matrix trace. Now the best λ is the minimizer of $GCV(\lambda)$. We need not dwell on this longer here.

4.2.4 Point Sources and Boundaries with Corners

In the numerical examples of Section 4.3, as in Section 3.3 and following [46], we use delta function combinations as the sources $q_i(\mathbf{x})$, in a manner that is typical in exploration geophysics

(namely, DC resistivity as well as low-frequency electromagnetic experiments), less so in EIT. However, these are clearly not honest L_∞ functions. Moreover, our domains Ω are a square or a cube and as such they have corners.

However, the theory developed above, and the data completion methods that it generates, can be extended to our experimental setting because we have control over the experimental setup. The desired effect is obtained by simply separating the location of each source from any of the receivers, and avoiding domain corners altogether.

Thus, consider in (3.5a) a source function of the form

$$q(\mathbf{x}) = \hat{q}(\mathbf{x}) + \delta(\mathbf{x} - \mathbf{x}^*) - \delta(\mathbf{x} - \mathbf{x}^{**}),$$

where $\hat{q}$ satisfies the assumptions previously made on q. Then we select $\mathbf{x}^*$ and $\mathbf{x}^{**}$ such that there are two open balls $B(\mathbf{x}^*, r)$ and $B(\mathbf{x}^{**}, r)$ of radius $r > 0$ each and centered at $\mathbf{x}^*$ and $\mathbf{x}^{**}$, respectively, where (i) no domain corner belongs to $B(\mathbf{x}^*, r) \cup B(\mathbf{x}^{**}, r)$, and (ii) $(B(\mathbf{x}^*, r) \cup B(\mathbf{x}^{**}, r)) \cap \Gamma$ is empty. Now, in our elliptic PDE problem the lower smoothness effect of either a domain corner or a delta function is local! In particular, the contribution of the point source to the flux $\mu\nabla u$ is the integral of $\delta(\mathbf{x} - \mathbf{x}^*) - \delta(\mathbf{x} - \mathbf{x}^{**})$, and this is smooth outside the union of the two balls.

4.3 Numerical Experiments

The PDE problem used in our experiments is described in Sections 4.1.3 and 3.3. The experimental setting is also very similar to that in Section 3.3.2. Here again, in 2D, the receivers are located at the top and bottom boundaries (except the corners). As such, the completion steps (4.2) or (4.6) are carried out separately for the top and bottom 1D manifold of boundaries. In 3D, since the receivers are placed on the top surface, hence completion is done on the corresponding 2D manifold.

For all of our numerical experiments, the "true field" is calculated on a grid that is twice as fine as the one used to reconstruct the model. For the 2D examples, the reconstruction is done on a uniform grid of size 129^2 with $s = 961$ experiments in the setup described above. For the

3D examples, we set $s = 512$ and employ a uniform grid of size 33^3, except for Example 4.3 where the grid size is 17^3.

In the numerical examples considered below, we use true models with piecewise constant levels, with the conductivities bounded away from 0. For further discussion of such models within the context of EIT, see [64].

Numerical examples are presented for both cases described in Sections 4.2.1 and 4.2.2. For all of our numerical examples except Examples 4.5 and 4.6, we use the transfer function (A.5) with $\mu_{\max} = 1.2 \max \mu(\mathbf{x})$, and $\mu_{\min} = \frac{1}{1.2} \min \mu(\mathbf{x})$. In the ensuing calculations we then "forget" what the exact $\mu(\mathbf{x})$ is. Further, in the stabilized GN iteration we employ preconditioned conjugate gradient (PCG) inner iterations, setting as described in Section A.3 the PCG iteration limit to $r = 20$, and the PCG tolerance to 10^{-3}. The initial guess is $\mathbf{m}_0 = \mathbf{0}$. Examples 4.5 and 4.6 are carried out using the level set method (A.6). Here we can set $r = 5$, significantly lower than above. The initial guess for the level set example is a cube with rounded corners inside Ω as in Figure 3.1.

For Examples 4.1, 4.2, 4.3 and 4.5, in addition to displaying the log conductivities (i.e., $\log(\mu)$) for each reconstruction, we also show the log-log plot of misfit on the entire data (i.e., $\|F(\mathbf{m}) - D\|_F$) vs. PDE count. A table of total PDE counts (not including what extra is required for the plots) for each method is displayed. In order to simulate the situation where sources do not share the same receivers, we first generate the data fully on the entire domain of measurement and then knock out at random some percentage of the generated data. This setting roughly corresponds to an EMG experiment with faulty receivers.

For each example, we use Algorithm 1 with one of the variants (i), (ii) or (iii) paired with one of the stopping criteria (a) or (b). For instance, when using variant (ii) with the soft stopping criterion (b), we denote the resulting algorithm by (ii, b). For the relaxed stopping rule (b) we (conservatively) set $t_0 = 100$ in (4.1). A computation using RS applied to the original data, using variant (i,x), is compared to one using SS applied to the completed data through variant (ii,x) or (iii,x), where x stands for a or b.

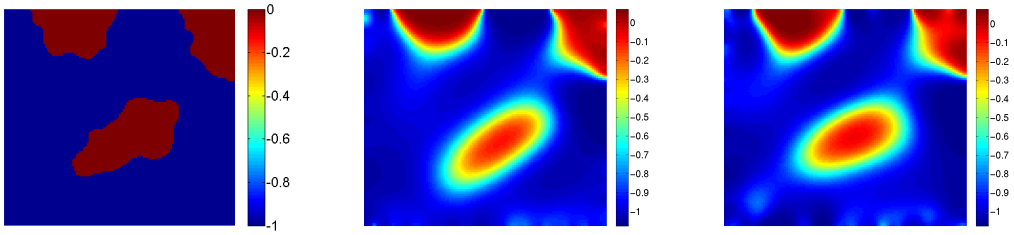
For convenience of cross reference, we gather all resulting seven algorithm comparisons and corresponding work counts in Table 4.1 below. For Examples 4.1, 4.2, 4.3 and 4.5, the

corresponding entries of this table should be read together with the misfit plots for each example.

Example	Algorithm	Random Subset	Data Completion
4.1	(i,a) | (iii,a)	3,647	1,716
4.2	(i,a) | (iii,a)	6,279	1,754
4.3	(i,a) | (iii,a)	3,887	1,704
4.4	(i,b) | (ii,b)	4,004	579
4.5	(i,a) | (iii,a)	3,671	935
4.6	(i,b) | (ii,b)	1,016	390
4.7	(i,b) | (ii,b)	4,847	1,217

Table 4.1: Algorithm and work in terms of number of PDE solves, comparing RS against data completion using Gaussian SS.

Example 4.1. *In this example, we place two target objects of conductivity* $\mu_I = 1$ *in a background of conductivity* $\mu_{II} = 0.1$*, and* 5% *noise is added to the data as described above. Also,* 25% *of the data requires completion. The discontinuities in the conductivity are touching the measurement domain, so we use* (4.6) *to complete the data. The hard stopping criterion* (a) *is employed, and iteration control is done using the original data, i.e., variants (i, a) and (iii, a) are compared: see the first entry of Table 4.1 and Figure 4.6(a).*



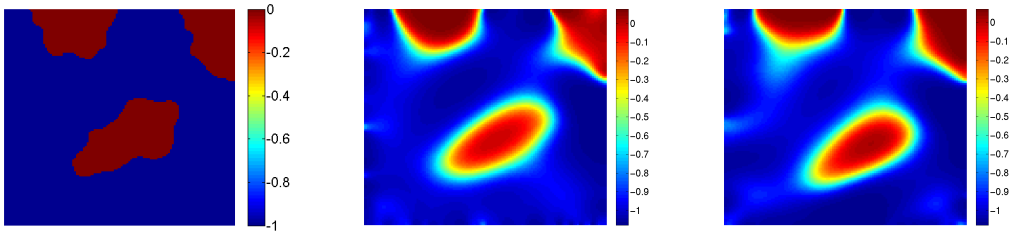
(a) True model (b) Random Subset (c) Data Completion

Figure 4.3: Example 4.1 – reconstructed log conductivity with 25% data missing and 5% noise. Regularization (4.6) has been used to complete the data.

The corresponding reconstructions are depicted in Figure 4.3. It can be seen that roughly the same quality reconstruction is obtained using the data completion method at less than half the price.

Example 4.2. *This example is the same as Example 4.1, except that* 50% *of the data is missing and requires completion. The same algorithm variants as in Example 4.1 are compared. The*

reconstructions are depicted in Figure 4.4, and comparative computational results are recorded in Table 4.1 and Figure 4.6(b).

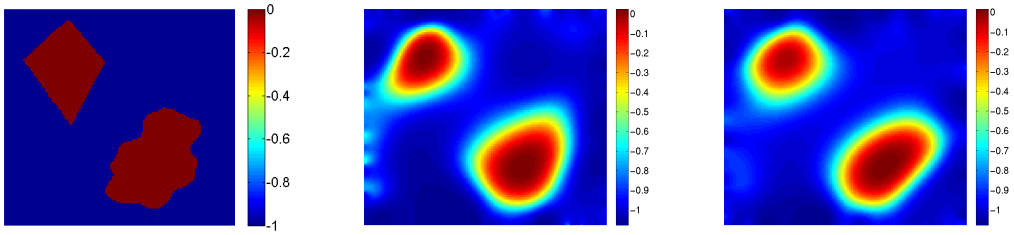


(a) True model (b) Random Subset (c) Data Completion

Figure 4.4: Example 4.2 – reconstructed log conductivity with 50% data missing and 5% noise. Regularization (4.6) has been used to complete the data.

Similar observations to those in Example 4.1 generally apply here as well, despite the smaller amount of original data.

Example 4.3. *This is the same as Example 4.2 in terms of noise and the amount of missing data, except that the discontinuities in the conductivity are some distance away from the common measurement domain, so we use* (4.2) *to complete the data. The same algorithm variants as in the previous two examples are compared, thus isolating the effect of a smoother data approximant.*

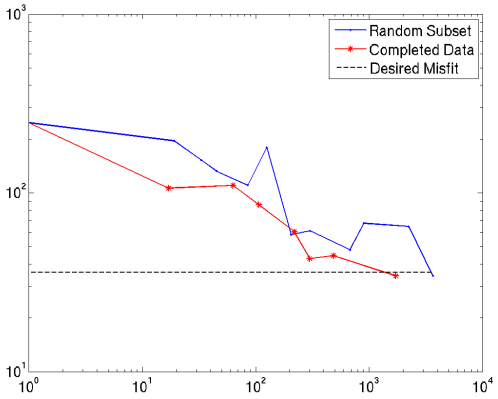


(a) True model (b) Random Subset (c) Data Completion

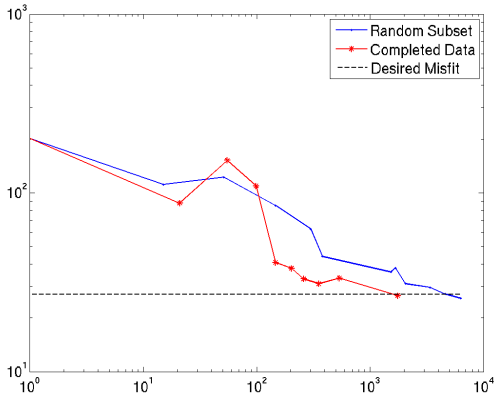
Figure 4.5: Example 4.3 – reconstructed log conductivity with 50% data missing and 5% noise. Regularization (4.2) has been used to complete the data.

Results are recorded in Figure 4.5, the third entry of Table 4.1 and Figure 4.6(c).

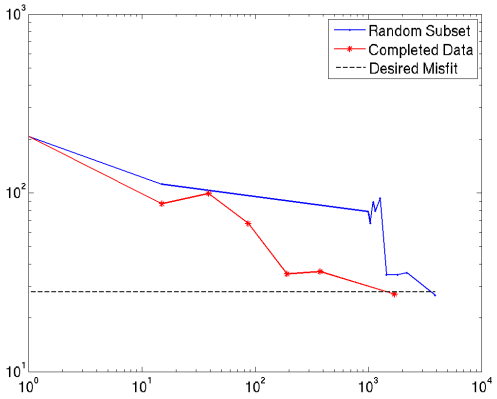
Figures 4.3, 4.4 and 4.5 in conjunction with Figure 4.6 as well as Table 4.1, reflect superiority of the SS method combined with data completion over the RS method with the original data. From the first three entries of Table 4.1, we see that the SS reconstruction with completed data



(a) Example 4.1



(b) Example 4.2



(c) Example 4.3

Figure 4.6: Data misfit vs. PDE count for Examples 1, 2 and 3.

can be done more efficiently by a factor of more than two. The quality of reconstruction is also very good. Note that the graph of the misfit for Data Completion lies mostly under that of Random Subset. This means that, given a fixed number of PDE solves, we obtain a lower (thus better) misfit for the former than for the latter.

Next, we consider examples in 3D.

Example 4.4. *In this example, the discontinuities in the true, piecewise constant conductivity extend all the way to the common measurement domain, see Figure 4.7. We therefore use* (4.6) *to complete the data. The target object has the conductivity* $\mu_I = 1$ *in a background with conductivity* $\mu_{II} = 0.1$. *We add* 2% *noise and knock out* 50% *of the data. Furthermore, we consider the relaxed stopping criterion* (b). *With the original data (hence using RS), the*

variant (i, b) is employed, and this is compared against the variant (ii, b) with SS applied to the completed data. For the latter case, the stopping tolerance is adjusted as discussed in Section 4.1.1.

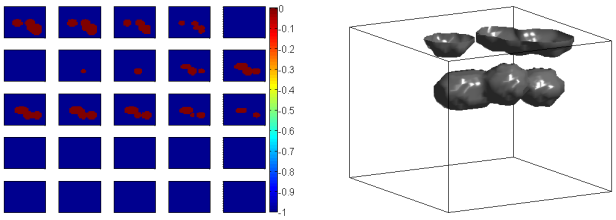
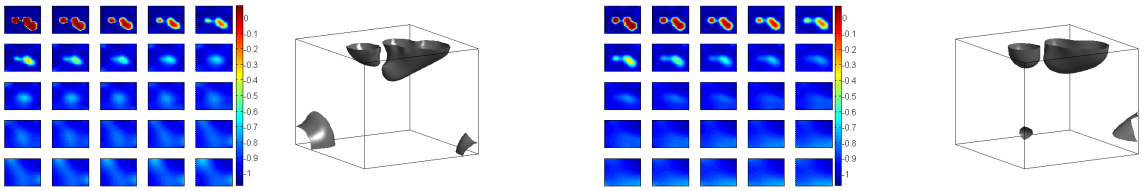


Figure 4.7: True Model for Example 4.4.



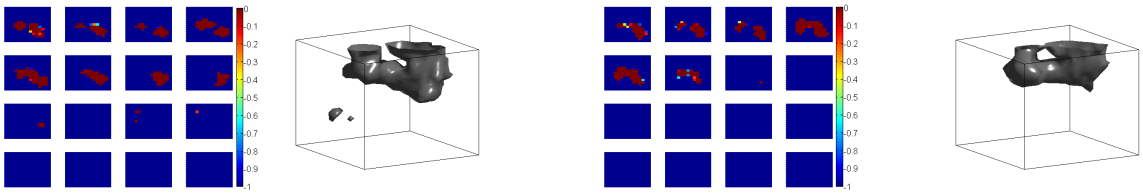
(a) RS slices (b) 3D view (c) DC slices (d) 3D view

Figure 4.8: Example 4.4 – reconstructed log conductivity for the 3D model with (a,b) Random Subset, (c,d) Data Completion for the case of 2% noise and 50% of data missing. Regularization (4.6) has been used to complete the data.

Reconstruction results are depicted in Figure 4.8, and work estimates are gathered in the 4th entry of Table 4.1. It can be seen that the results using data completion, obtained at about 1/7th the cost, are comparable to those obtained with RS applied to the original data.

Example 4.5. *The underlying model in this example is the same as that in Example 4.4 except that, since we intend to plot the misfit on the entire data at every GN iteration, we decrease the reconstruction mesh resolution to* 17^3*. Also,* 30% *of the data requires completion, and we use the level set transfer function* (A.6) *to reconstruct the model. With the original data, we use the variant (i, a), while the variant (iii, a) is used with the completed data. The reconstruction results are recorded in Figure 4.9, and performance indicators appear in Figure 4.10 as well as Table 4.1.*

The algorithm proposed here produces a better reconstruction than RS on the original data. A relative efficiency observation can be made from Table 4.1, where a factor of roughly 4 *is*



(a) RS slices (b) 3D view (c) DC slices (d) 3D view

Figure 4.9: Example 4.5 – reconstructed log conductivity for the 3D model using the level set method with (a,b) Random Subset, (c,d) Data Completion for the case of 2% noise and 30% of data missing. Regularization (4.6) has been used to complete the data.

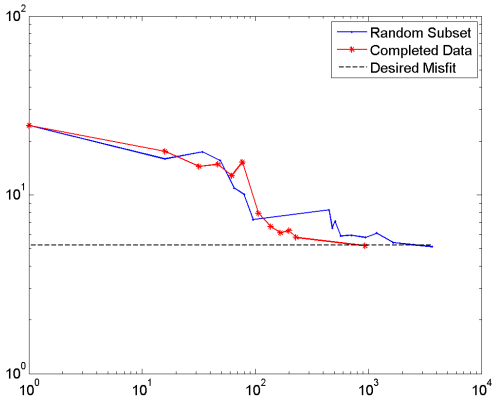


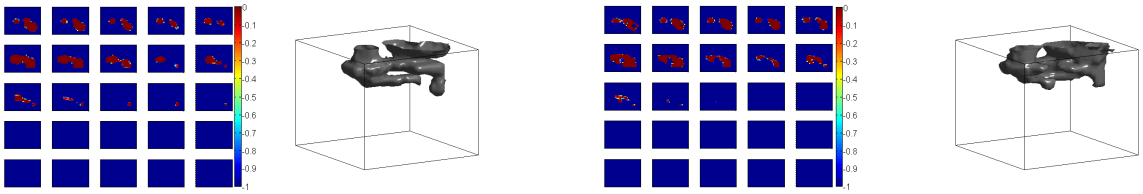
Figure 4.10: Data misfit vs. PDE count for Example 4.5.

revealed.

Example 4.6. *This is exactly the same as Example 4.4, except that we use the level set transfer function* (A.6) *to reconstruct the model. The same variants of Algorithm 1 as in Example 4.4 are employed.*

It is evident from Figure 4.11 that employing the level set formulation allows a significantly better quality reconstruction than in Example 4.4. This is expected, as much stronger assumptions on the true model are utilized. It was shown in [131] as well as Chapter 3 that using level set functions can greatly reduce the total amount of work, and this is observed here as well.

Whereas in all previous examples convergence of the modified GN iterations from a zero initial guess was fast and uneventful, typically requiring fewer than 10 iterations, the level set result of this example depends on $\mathbf{m}_0$ *in a more erratic manner. This reflects the underlying uncertainty of the inversion, with the initial guess* $\mathbf{m}_0$ *playing the role of a prior.*



(a) RS slices (b) 3D view (c) DC slices (d) 3D view

Figure 4.11: Example 4.6 – reconstructed log conductivity for the 3D model using the level set method with (a,b) Random Subset, (c,d) Data Completion for the case of 2% noise and 50% of data missing. Regularization (4.6) has been used to complete the data.

It can be clearly seen from the results of Examples 4.4, 4.5 and 4.6 that Algorithm 1 does a great job recovering the model using the completed data plus the SS method as compared to RS with the original data. This is so both in terms of total work and the quality of the recovered model. Note that for all reconstructions, the conductive object placed deeper than the ones closer to the surface is not recovered well. This is due to the fact that we only measure on the surface and the information coming from this deep conductive object is majorized by that coming from the objects closer to the surface.

Example 4.7. *In this 3D example, we examine the performance of our data completion approach for more severe cases of missing data. For this example, we place a target object of conductivity* $\mu_I = 1$ *in a background with conductivity* $\mu_{II} = 0.1$*, see Figure 4.12, and* 2% *noise is added to the "exact" data. Then we knock out* 70% *of the data and use* (4.2) *to complete it. The algorithm variants employed are the same as in Examples 4.4 and 4.6.*

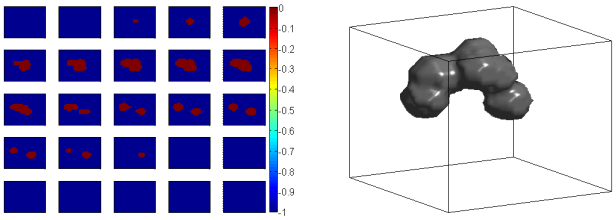
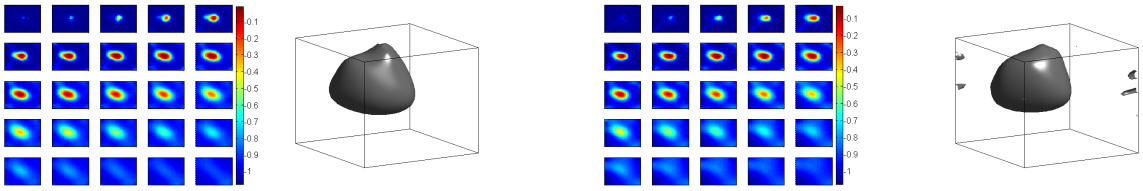


Figure 4.12: True Model for Example 4.7.

Results are gathered in Figures 4.13 as well as Table 4.1. The data completion plus simultaneous sources algorithm again does well, with an efficiency factor ≈ 4.



(a) RS slices (b) 3D view (c) DC slices (d) 3D view

Figure 4.13: Example 4.7 – reconstructed log conductivity for the 3D model with (a,b) Random Subset, (c,d) Data Completion for the case of 2% noise and 70% data missing. Regularization (4.2) has been used to complete the data.

4.4 Conclusions

This chapter is a sequel to Chapter 3 in which we studied the case that the linearity assumption (A.2) holds. In the context of PDE constrained inverse problem, this translates to the case where sources share the same receivers. Here we have focused on the very practical case where arise more often in practice, i.e., the linearity assumption (A.2) is violated. Such scenarios arise, for example, where there are parts of data missing or heavily corrupted. For PDE constrained inverse problems, this case corresponds to the situation where, unlike Chapter 3, sources do not share the same receivers. In this chapter, we assumed that the experimental setting is "suitable" enough to allow for the use of our proposed data completion techniques based on appropriate regularization. Our data completion methods are motivated by theory in Sobolev spaces, [54], regarding the properties of weak solutions along the domain boundary. The resulting completed data allows an efficient use of the methods developed in Chapter 3 as well as utilization of a relaxed stopping criterion. Our approach shows great success in cases of moderate data completion, say up to 60-70%. In such cases we have demonstrated that, utilizing some variant of Algorithm 3, an execution speedup factor of at least 2 and often much more can be achieved while obtaining excellent reconstructions.

It needs to be emphasized that a blind employment of some interpolation/approximation method would not take into account available a priori information about the sought signal. In contrast, the method developed in this chapter, while being very simple, is in fact built upon such a priori information, and is theoretically justified.

Note that with the methods of Section 4.2 we have also replaced the original data with new,

approximate data. Alternatively we could keep the original data, and just add the missing data sampled from v_i at appropriate locations. The potential advantage of doing this is that fewer changes are made to the original problem, so it would seem plausible that the data extension will produce results that are close to the more expensive inversion without using the simultaneous sources method, at least when there are only a few missing receivers. However, we found in practice that this method yields similar or worse reconstructions for moderate or large amounts of missing data as compared to the methods of Section 4.2.

For severe cases of missing data, say 80% or more, we do not recommend data completion in the present context as a safe approach. With so much completion the bias in the completed field could overwhelm the given observed data, and the recovered model may not be correct. In such cases, one can use the RS method applied to the original data. A good initial guess for this method may still be obtained with the SS method applied to the completed data. Thus, one can always start with the most daring variant (ii, b) of Algorithm 3, and add a more conservative run of variant (i, b) on top if necessary.

If the forward problem is very diffusive and has a strong smoothing effect, as is the case for the DC-resistivity and EIT problems, then data completion can be attempted using a (hopefully) good guess of the sought model $\mathbf{m}$ by solving the forward problem and evaluating the solution wherever necessary [70]. The rationale here is that even relatively large changes in $m(\mathbf{x})$ produce only small changes in the fields $u_i(\mathbf{x})$. However, such a prior might prove dominant, hence risky, and the data produced in this way, unlike the original data, no longer have natural high frequency noise components. Indeed, a potential advantage of this approach is in using the difference between the original measured data and the calculated prior field at the same locations for estimating the noise level ϵ for a subsequent application of the Morozov discrepancy principle [52, 135].

In this chapter we have focused on data completion, using whenever possible the same computational setting as in Chapter 3, which is our base reference. Other approaches to reduce the overall computational costs are certainly possible. These include adapting the number of inner PCG iterations in the modified GN outer iteration (see [46]) and adaptive gridding for $\mathbf{m}(\mathbf{x})$ (see, e.g., [72] and references therein). Such techniques are essentially independent of the focus here. At the same time, they can be incorporated or fused together with our stochastic

algorithms, further improving efficiency: effective ways for doing this form a topic for future research.

The specific data completion techniques proposed in this chapter have been justified and used in our model DC resistivity problem. However, the overall idea can be extended to other PDE based inverse problems as well by studying the properties of the solution of the forward problem. One first needs to see what the PDE solutions are expected to behave like on the measurement domain, for example on a portion of the boundary, and then imposing this prior knowledge in the form of an appropriate regularizer in the data completion formulation. Following that, the rest can be similar to our approach here. Investigating such extensions to other PDE models is a subject for future studies.

Chapter 5

Matrix Trace Estimation

As shown in Section 2.1, stochastic approximations to the misfit are closely related to Monte-Carlo estimations of the trace of the corresponding implicit matrix. So far, in this thesis, all these estimators have been used rather heuristically and no attempt to better mathematically understanding them has been made. In this chapter, we present a rigorous mathematical analysis of Monte-Carlo methods for the estimation of the trace, $tr(A)$, of an implicitly given matrix A whose information is only available through matrix-vector products. Such a method approximates the trace by an average of n expressions of the form $\mathbf{w}^T(A\mathbf{w})$, with random vectors $\mathbf{w}$ drawn from an appropriate distribution. We prove, discuss and experiment with bounds on the number of realizations n required in order to guarantee a probabilistic bound on the relative error of the trace estimation upon employing Rademacher (Hutchinson), Gaussian and uniform unit vector (with and without replacement) probability distributions, discussed in Section 2.1.1.

In total, one necessary and six sufficient bounds are proved, improving upon and extending similar estimates obtained in the seminal work of Avron and Toledo [22] in several dimensions. We first improve their bound on n for the Hutchinson method, dropping a term that relates to $rank(A)$ and making the bound comparable with that for the Gaussian estimator.

We further prove new sufficient bounds for the Hutchinson, Gaussian and the unit vector estimators, as well as a necessary bound for the Gaussian estimator, which depend more specifically on properties of the matrix A. As such they may suggest for what type of matrices one distribution or another provides a particularly effective or relatively ineffective stochastic estimation method.

By the novel results in this chapter, it is hoped to correct some existing misconceptions regarding the relative performance of different estimators that have resulted due to an unsatis-

factory state of the theory. The theory developed in the present chapter sheds light on several questions which had remained open for some time. Using these results, practitioners can better choose appropriate estimators for their applications.

5.1 Introduction

The need to estimate the trace of an implicit square matrix is of fundamental importance [126] and arises in many applications; see for instance [5, 6, 21–23, 43, 46, 66, 71, 86, 104, 121, 134, 138] and references therein. By "implicit" we mean that the matrix of interest is not available explicitly: only probes in the form of matrix-vector products for any appropriate vector are available. The standard approach for estimating the trace of such a matrix $A \in \mathbb{R}^{s\times s}$ is based on a Monte-Carlo method, where one generates n random vector realizations $\mathbf{w}_i$ from a suitable probability distribution $\mathcal{D}$ and computes

$$tr^n_{\mathcal{D}}(A) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_i^T A \mathbf{w}_i. \tag{5.1}$$

For the popular case where A is symmetric positive semi-definite (SPSD), the original method for estimating its trace, is due to Hutchinson [86] and uses the Rademacher distribution for $\mathcal{D}$.

Until the work by Avron and Toledo [22], the main analysis and comparison of such methods was based on the variance of one sample. It is known that compared to other methods the Hutchinson method has the smallest variance, and as such it has been extensively used in many applications. In [22] so-called (ε, δ) bounds are derived in which, using Chernoff-like analysis, a lower bound is obtained on the number of samples required to achieve a probabilistically guaranteed relative error of the estimated trace. More specifically, for a given pair (ε, δ) of small (say, < 1) positive values and an appropriate probability distribution $\mathcal{D}$, a lower bound on n is provided such that

$$Pr\left(|tr^n_{\mathcal{D}}(A) - tr(A)| \leq \varepsilon\ tr(A)\right) \geq 1 - \delta. \tag{5.2}$$

These authors further suggest that minimum-variance estimators may not be practically best, and conclude based on their analysis that the method with the best bound is the one using the

Gaussian distribution. Let us denote

$$c = c(\varepsilon, \delta) := \varepsilon^{-2} \ln(2/\delta), \tag{5.3a}$$

$$r = rank(A). \tag{5.3b}$$

Then [22] showed that, provided A is real SPSD, (5.2) holds for the Hutchinson method if $n \geq 6(c + \varepsilon^{-2} \ln r)$ and for the Gaussian distribution if $n \geq 20c$.

In the present chapter we continue to consider the same objective as in [22], and our first task is to improve on these bounds. Specifically, in Theorems 5.1 and 5.3, respectively, we show that (5.2) holds for the Hutchinson method if

$$n \geq 6c(\varepsilon, \delta), \tag{5.4}$$

and for the Gaussian distribution if

$$n \geq 8c(\varepsilon, \delta). \tag{5.5}$$

The bound (5.4) removes a previous factor involving the rank of the matrix A, conjectured in [22] to be indeed redundant. Note that these two bounds are astoundingly simple and general: they hold for any SPSD matrix, regardless of size or any other matrix property. Thus, we cannot expect them to be tight in practice for many specific instances of A that arise in applications. However, as was recently shown in [137], these two bounds are asymptotically tight.

Although practically useful, the bounds on n given in (5.4) and (5.5) do not provide insight into how different types of matrices are handled with each probability distribution. Our next contribution is to provide different bounds for the Gaussian and Hutchinson trace estimators which, though generally not computable for implicit matrices, do shed light on this question.

Furthermore, for the Gaussian estimator we prove a practically useful *necessary lower* bound on n, for a given pair (ε, δ).

A third probability distribution we consider was called the unit vector distribution in [22]. Here, the vectors $\mathbf{w}_i$ in (5.1) are uniformly drawn from the columns of a scaled identity matrix, $\sqrt{s}\mathbb{I}$, and A need not be SPSD. Such a distribution is used in obtaining the random subset

method discussed in Chapter 2. We slightly generalize the bound in [22], obtained for the case where the sampling is done with replacement. Our bound, although not as simply computed as (5.4) or (5.5), can be useful in determining which types of matrices this distribution works best on. We then give a tighter bound for the case where the sampling is done without replacement, suggesting that when the difference between the bounds is significant (which happens when n is large), a uniform random sampling of unit vectors without replacement may be a more advisable distribution to estimate the trace with.

This chapter is organized as follows. Section 5.2 gives two bounds for the Hutchinson method as advertised above, namely the improved bound (5.4) and a more involved but potentially more informative bound. Section 5.3 deals likewise with the Gaussian method and adds also a necessary lower bound, while Section 5.4 is devoted to the unit vector sampling methods.

In Section 5.5 we give some numerical examples verifying that the trends predicted by the theory are indeed realized. Conclusions and further thoughts are gathered in Section 5.6.

In what follows we use the notation $tr_H^n(A)$, $tr_G^n(A)$, $tr_{U_1}^n(A)$, and $tr_{U_2}^n(A)$ to refer, respectively, to the trace estimators using Hutchinson, Gaussian, and uniform unit vector with and without replacement, in lieu of the generic notation $tr_{\mathcal{D}}^n(A)$ in (5.1) and (5.2). We also denote for any given random vector of size n, $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})^T$. We restrict attention to real-valued matrices, although extensions to complex-valued ones are possible, and employ the 2-norm by default.

5.2 Hutchinson Estimator Bounds

In this section we consider the Hutchinson trace estimator, $tr_H^n(A)$, obtained by setting $\mathcal{D} = H$ in (5.1), where the components of the random vectors $\mathbf{w}_i$ are i.i.d Rademacher random variables (i.e., $Pr(w_{ij} = 1) = Pr(w_{ij} = -1) = \frac{1}{2}$).

5.2.1 Improving the Bound in [22]

Theorem 5.1. *Let A be an $s \times s$ SPSD matrix. Given a pair (ε, δ), the inequality* (5.2) *holds with $\mathcal{D} = H$ if n satisfies* (5.4).

Proof. Since A is SPSD, it can be diagonalized by a unitary similarity transformation as $A = U^T \Lambda U$. Consider n random vectors $\mathbf{w}_i,\ i = 1, \ldots, n$, whose components are i.i.d and drawn from the Rademacher distribution, and define $\mathbf{z}_i = U\mathbf{w}_i$ for each. We have

$$\begin{aligned}
Pr\left(tr_H^n(A) \leq (1-\varepsilon)tr(A)\right) &= Pr\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_i^T A \mathbf{w}_i \leq (1-\varepsilon)tr(A)\right) \\
&= Pr\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{z}_i^T \Lambda \mathbf{z}_i \leq (1-\varepsilon)tr(A)\right) \\
&= Pr\left(\sum_{i=1}^{n}\sum_{j=1}^{r} \lambda_j z_{ij}^2 \leq n(1-\varepsilon)tr(A)\right) \\
&= Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \sum_{i=1}^{n} z_{ij}^2 \leq n(1-\varepsilon)\right) \\
&\leq \exp\{tn(1-\varepsilon)\}\mathbb{E}\left(\exp\{\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \sum_{i=1}^{n} -tz_{ij}^2\}\right),
\end{aligned}$$

where the last inequality holds for any $t > 0$ by Markov's inequality.

Next, using the convexity of the exp function and the linearity of expectation, we obtain

$$\begin{aligned}
\mathbb{E}\left(\exp\{\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \sum_{i=1}^{n} -tz_{ij}^2\}\right) &\leq \sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \mathbb{E}\left(\exp\{\sum_{i=1}^{n} -tz_{ij}^2\}\right) \\
&= \sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \mathbb{E}\left(\prod_{i=1}^{n} \exp\{-tz_{ij}^2\}\right) \\
&= \sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} \prod_{i=1}^{n} \mathbb{E}\left(\exp\{-tz_{ij}^2\}\right),
\end{aligned}$$

where the last equality holds since, for a given j, z_{ij}'s are independent with respect to i.

Now, we want to have that

$$\exp\{tn(1-\varepsilon)\} \prod_{i=1}^{n} \mathbb{E}\left(\exp\{-tz_{ij}^2\}\right) \leq \delta/2.$$

For this we make use of the inequalities in the end of the proof of Lemma 5.1 of [2]. Following

inequalities (15)–(19) in [2] and letting $t = \varepsilon/(2(1+\varepsilon))$, we get

$$\exp\{tn(1-\varepsilon)\}\prod_{i=1}^{n}\mathbb{E}\left(\exp\{-tz_{ij}^2\}\right) < \exp\{-\frac{n}{2}(\frac{\varepsilon^2}{2}-\frac{\varepsilon^3}{3})\}.$$

Next, if n satisfies (5.4) then $\exp\{-\frac{n}{2}(\frac{\varepsilon^2}{2}-\frac{\varepsilon^3}{3})\} < \delta/2$, and thus it follows that

$$Pr\left(tr_H^n(A) \leq (1-\varepsilon)tr(A)\right) < \delta/2.$$

By a similar argument, making use of inequalities (11)–(14) in [2] with the same t as above, we also obtain with the same bound for n so that

$$Pr\left(tr_H^n(A) \geq (1+\varepsilon)tr(A)\right) \leq \delta/2.$$

So finally using the union bound yields the desired result. □

It can be seen that (5.4) is the same bound as the one in [22] with the important exception that the factor $r = rank(A)$ does not appear in the bound. Furthermore, the same bound on n holds for any SPSD matrix.

5.2.2 A Matrix-Dependent Bound

Here we derive another bound for the Hutchinson trace estimator which may shed light as to what type of matrices the Hutchinson method is best suited for.

For $k, j = 1, \ldots, s$, let us denote by $a_{k,j}$ the (k,j)th element of A and by $\mathbf{a}_j$ its jth column.

Theorem 5.2. *Let A be an $s \times s$ symmetric positive semi-definite matrix, and define*

$$\mathcal{K}_H^j := \frac{\|\mathbf{a}_j\|^2 - a_{j,j}^2}{a_{j,j}^2} = \sum_{k\neq j} a_{k,j}^2 \;/\; a_{j,j}^2, \quad \mathcal{K}_H := \max_j \mathcal{K}_H^j. \tag{5.6}$$

Given a pair of positive small values (ε, δ), the inequality (5.2) holds with $\mathcal{D} = H$ if

$$n > 2\mathcal{K}_H c(\varepsilon, \delta). \tag{5.7}$$

Proof. Elementary linear algebra implies that since A is SPSD, $a_{j,j} \geq 0$ for each j. Furthermore, if $a_{j,j} = 0$ then the jth row and column of A identically vanish, so we may assume below that $a_{j,j} > 0$ for all $j = 1, \ldots, s$. Note that

$$tr_H^n(A) - tr(A) = \frac{1}{n} \sum_{j=1}^{s} \sum_{i=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{s} a_{j,k} w_{ij} w_{ik}.$$

Hence

$$\begin{aligned} Pr\left(tr_H^n(A) \leq (1-\varepsilon) tr(A)\right) &= Pr\left(\sum_{j=1}^{s} \sum_{i=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{s} -a_{j,k} w_{ij} w_{ik} \geq n\varepsilon \; tr(A) \right) \\ &= Pr\left(\sum_{j=1}^{s} \frac{a_{j,j}}{tr(A)} \sum_{i=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{s} -\frac{a_{j,k}}{a_{j,j}} w_{ij} w_{ik} \geq n\varepsilon \right) \\ &\leq \exp\{-tn\varepsilon\} \mathbb{E}\left(\exp\{ \sum_{j=1}^{s} \frac{a_{j,j}}{tr(A)} \sum_{i=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{s} -\frac{a_{j,k} t}{a_{j,j}} w_{ij} w_{ik} \} \right), \end{aligned}$$

where the last inequality is again obtained for any $t > 0$ by using Markov's inequality. Now, again using the convexity of the exp function and the linearity of expectation, we obtain

$$\begin{aligned} Pr\left(tr_H^n(A) \leq (1-\varepsilon) tr(A)\right) &\leq \exp\{-tn\varepsilon\} \sum_{j=1}^{s} \frac{a_{j,j}}{tr(A)} \mathbb{E}\left(\exp\{ \sum_{i=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{s} -\frac{a_{j,k} t}{a_{j,j}} w_{ij} w_{ik} \} \right) \\ &= \exp\{-tn\varepsilon\} \sum_{j=1}^{s} \frac{a_{j,j}}{tr(A)} \prod_{i=1}^{n} \mathbb{E}\left(\exp\{ \sum_{\substack{k=1 \\ k \neq j}}^{s} -\frac{a_{j,k} t}{a_{j,j}} w_{ij} w_{ik} \} \right) \end{aligned}$$

by independence of $w_{ij} w_{ik}$ with respect to the index i.

Next, note that

$$\mathbb{E}\left(\exp\{ \sum_{\substack{k=1 \\ k \neq j}}^{s} \frac{a_{j,k} t}{a_{j,j}} w_{ik} \} \right) = \mathbb{E}\left(\exp\{ \sum_{\substack{k=1 \\ k \neq j}}^{s} -\frac{a_{j,k} t}{a_{j,j}} w_{ik} \} \right).$$

Furthermore, since $Pr(w_{ij} = -1) = Pr(w_{ij} = 1) = \frac{1}{2}$, and using the law of total expectation, we have

$$\begin{aligned}\mathbb{E}\left(\exp\{\sum_{\substack{k=1\\k\neq j}}^{s} -\frac{a_{j,k}t}{a_{j,j}}w_{ij}w_{ik}\}\right) &= \mathbb{E}\left(\exp\{\sum_{\substack{k=1\\k\neq j}}^{s} \frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right)\\ &= \prod_{\substack{k=1\\k\neq j}}^{s}\mathbb{E}\left(\exp\{\frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right),\end{aligned}$$

so

$$Pr\left(tr_H^n(A) \leq (1-\varepsilon)tr(A)\right) \leq \exp\{-tn\varepsilon\}\sum_{j=1}^{s}\frac{a_{j,j}}{tr(A)}\prod_{i=1}^{n}\prod_{\substack{k=1\\k\neq j}}^{s}\mathbb{E}\left(\exp\{\frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right).$$

We want to have the right hand side expression bounded by $\delta/2$.

Applying Hoeffding's lemma we get

$$\mathbb{E}\left(\exp\{\frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right) \leq \exp\{\frac{a_{j,k}^2t^2}{2a_{j,j}^2}\},$$

hence

$$\exp\{-tn\varepsilon\}\prod_{i=1}^{n}\prod_{\substack{k=1\\k\neq j}}^{s}\mathbb{E}\left(\exp\{\frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right) \leq \exp\{-tn\varepsilon + \mathcal{K}_H^j nt^2/2\} \tag{5.8a}$$

$$\leq \exp\{-tn\varepsilon + \mathcal{K}_H nt^2/2\}. \tag{5.8b}$$

The choice $t = \varepsilon/\mathcal{K}_H$ minimizes the right hand side. Now if (5.7) holds then

$$\exp(-tn\varepsilon)\prod_{i=1}^{n}\prod_{\substack{k=1\\k\neq j}}^{s}\mathbb{E}\left(\exp\{\frac{a_{j,k}t}{a_{j,j}}w_{ik}\}\right) \leq \delta/2,$$

hence we have

$$Pr(tr_H^n(A) \leq (1-\varepsilon)tr(A)) \leq \delta/2.$$

Similarly, we obtain that

$$Pr(tr_H^n(A) \geq (1+\varepsilon)tr(A)) \quad \leq \quad \delta/2,$$

and using the union bound finally gives desired result. □

Comparing (5.7) to (5.4), it is clear that the bound of the present subsection is only worthy of consideration if $\mathcal{K}_H < 3$. Note that Theorem 5.2 emphasizes the relative ℓ_2 energy of the off-diagonals: the matrix does not necessarily have to be diagonally dominant (i.e., where a similar relationship holds in the ℓ_1 norm) for the bound on n to be moderate. Furthermore, a matrix need not be "nearly" diagonal for this method to require small sample size. In fact a matrix can have off-diagonal elements of significant size that are far away from the main diagonal without automatically affecting the performance of the Hutchinson method. However, note also that our bound can be pessimistic, especially if the average value or the mode of $\mathcal{K}_H^j$ in (5.6) is far lower than its maximum, $\mathcal{K}_H$. This can be seen in the above proof where the estimate (5.8b) is obtained from (5.8a). Simulations in Section 5.5 show that the Hutchinson method can be a very efficient estimator even in the presence of large outliers, so long as the bulk of the distribution is concentrated near small values.

The case $\mathcal{K}_H = 0$ corresponds to a diagonal matrix, for which the Hutchinson method yields the trace with one shot, $n = 1$. In agreement with the bound (5.7), we expect the actual required n to grow when a sequence of otherwise similar matrices A is envisioned in which $\mathcal{K}_H$ grows away from 0, as the energy in the off-diagonal elements grows relatively to that in the diagonal elements.

5.3 Gaussian Estimator Bounds

In this section we consider the Gaussian trace estimator, $tr_G^n(A)$, obtained by setting $\mathcal{D} = G$ in (5.1), where the components of the random vectors $\mathbf{w}_i$ are i.i.d standard normal random variables. We give two sufficient and one necessary lower bounds for the number of Gaussian samples required to achieve an (ε, δ) trace estimate. The first sufficient bound (5.5) improves the result in [22] by a factor of 2.5. Our bound is only worse than (5.4) by a fraction, and

it is an upper limit of the potentially more informative (if less available) bound (5.10), which relates to the properties of the matrix A. The bound (5.10) provides an indication as to what matrices may be suitable candidates for the Gaussian method. Then we present a practically computable, necessary bound for the sample size n.

5.3.1 Sufficient Bounds

The proof of the following theorem closely follows the approach in [22].

Theorem 5.3. *Let A be an $s \times s$ SPSD matrix and denote its eigenvalues by $\lambda_1, \ldots, \lambda_s$. Further, define*

$$\mathcal{K}_G^j := \frac{\lambda_j}{tr(A)}, \quad \mathcal{K}_G := \max_j \mathcal{K}_G^j = \frac{\|A\|}{tr(A)}. \tag{5.9}$$

Then, given a pair of positive small values (ε, δ), the inequality (5.2) holds with $\mathcal{D} = G$ provided that (5.5) holds. This estimate also holds provided that

$$n > 8\mathcal{K}_G c(\varepsilon, \delta). \tag{5.10}$$

Proof. Since A is SPSD, we have $\|A\| \leq tr(A)$, so if (5.5) holds then so does (5.10). We next concentrate on proving the result assuming the tighter bound (5.10) on the actual n required in a given instance.

Writing as in the previous section $A = U^T \Lambda U$, consider n random vectors $\mathbf{w}_i$, $i = 1, \ldots, n$, whose components are i.i.d and drawn from the normal distribution, and define $\mathbf{z}_i = U\mathbf{w}_i$. Since U is orthogonal, the elements z_{ij} of $\mathbf{z}_i$ are i.i.d Gaussian random variables. We have as before,

$$\begin{aligned} Pr\left(tr_G^n(A) \leq (1-\varepsilon)tr(A)\right) &= Pr\left(\sum_{i=1}^{n}\sum_{j=1}^{r} \lambda_j z_{ij}^2 \leq n(1-\varepsilon)tr(A)\right) \\ &\leq \exp\{tn(1-\varepsilon)tr(A)\}\mathbb{E}\left(\exp\{\sum_{i=1}^{n}\sum_{j=1}^{r} -t\lambda_j z_{ij}^2\}\right) \\ &\leq \exp\{tn(1-\varepsilon)tr(A)\}\prod_{i=1}^{n}\prod_{j=1}^{r}\mathbb{E}\left(\exp\{-t\lambda_j z_{ij}^2\}\right). \end{aligned}$$

Here z_{ij}^2 is a χ^2 random variable of degree 1 (see [106]), and hence for the characteristics we have

$$\mathbb{E}\left(\exp\{-t\lambda_j z_{ij}^2\}\right) = (1+2\lambda_j t)^{-\frac{1}{2}}.$$

This yields the bound

$$Pr\left(tr_G^n(A) \leq (1-\varepsilon)tr(A)\right) \leq \exp\{tn(1-\varepsilon)tr(A)\}\prod_{j=1}^{r}(1+2\lambda_j t)^{-\frac{n}{2}}.$$

Next, it is easy to prove by elementary calculus that given any $0 < \alpha < 1$, the following holds for all $0 \leq x \leq \frac{1-\alpha}{\alpha}$,

$$\ln(1+x) \geq \alpha x. \tag{5.11}$$

Setting $\alpha = 1-\varepsilon/2$, then by (5.11) and for all $t \leq (1-\alpha)/(2\alpha\|A\|)$, we have that $(1+2\lambda_j t) > \exp\{2\alpha\lambda_j\}t$, so

$$\begin{aligned} Pr\left(tr_G^n(A) \leq (1-\varepsilon)tr(A)\right) &\leq \exp\{tn(1-\varepsilon)tr(A)\}\prod_{j=1}^{r}\exp(-n\alpha\lambda_j t) \\ &= \exp\{tn(1-\varepsilon-\alpha)tr(A)\}. \end{aligned}$$

We want the latter right hand side to be bounded by $\delta/2$, i.e., we want to have

$$n \geq \frac{\ln\left(2/\delta\right)}{(\alpha-(1-\varepsilon))tr(A)t} = \frac{2\varepsilon c(\varepsilon,\delta)}{tr(A)t},$$

where $t \leq (1-\alpha)/(2\alpha\|A\|)$. Now, setting

$$t = (1-\alpha)/(2\alpha\|A\|) = \varepsilon/(2(2-\varepsilon)\|A\|),$$

we obtain

$$n \geq 4(2-\varepsilon)c(\varepsilon,\delta)\mathcal{K}_G,$$

so if (5.10) holds then

$$Pr\left(tr_G^n(A) \leq (1-\varepsilon)tr(A)\right) \leq \delta/2.$$

Using a similar argument we also obtain

$$Pr\left(tr_G^n(A) \geq (1+\varepsilon)tr(A)\right) \leq \delta/2,$$

and subsequently the union bound yields the desire result. □

The matrix-dependent bound (5.10), proved to be sufficient in Theorem 5.3, provides additional information over (5.5) about the type of matrices for which the Gaussian estimator is (probabilistically) guaranteed to require only a small sample size: if the eigenvalues of an SPSD matrix are distributed such that the ratio $\|A\|/tr(A)$ is small (e.g., if they are all of approximately the same size), then the Gaussian estimator bound requires a small number of realizations. This observation is reaffirmed by looking at the variance of this estimator, namely $2\|A\|_F^2$. It is easy to show that among all the matrices with a fixed trace and rank, those with equal eigenvalues have the smallest Frobenius norm.

It is easy to see that the *stable rank* (see [130] and references therein) of any real rectangular matrix C which satisfies $A = C^TC$ equals $1/\mathcal{K}_G$. Thus, the bound constant in (5.10) is inversely proportional to this stable rank, suggesting that estimating the trace using the Gaussian distribution may become inefficient if the stable rank of the matrix is low. Furthermore, the ratio

$$eRank := 1/\mathcal{K}_G = tr(A)/\|A\|$$

is known as the *effective rank* of the matrix (see [51]), which is, similar to stable rank, a continuous relaxation and a stable quantity compared with the usual rank. Using the concept of effective rank, we can establish a connection between efficiency of the Gaussian estimator and the effective rank of matrices: Theorem 5.3 indicates that the *true* sample size, i.e., the minimum sample size for which (5.2) holds, is in fact in $O(1/eRank)$. Hence as the effective rank of a matrix grows larger, it becomes easier (i.e., smaller sample size is required) to estimate its trace, with the same probabilistic accuracy. Theorem 5.5 in Section 5.3.2 below establishes a different relationship between the inefficiency of the Gaussian estimator and a rank of a matrix.

As an example of an application of the above results, let us consider finding the minimum number of samples required to compute the rank of a projection matrix using the Gaussian

estimator [22, 26]. Recall that a projection matrix is SPSD with only 0 and 1 eigenvalues. Compared to the derivation in [22], here we use Theorem 5.3 directly to obtain a similar bound with a slightly better constant.

Corollary 5.4. *Let A be an $s \times s$ projection matrix with rank $r > 0$, and denote the rounding of any real scalar x to the nearest integer by $round(x)$. Then, given a positive small value δ, the estimate*

$$Pr\left(round(tr_G^n(A)) \neq r\right) \leq \delta \tag{5.12a}$$

holds if

$$n \geq 8\, r \ln\left(2/\delta\right). \tag{5.12b}$$

Proof. The result immediately follows using Theorem 5.3 upon setting $\varepsilon = 1/r, \|A\| = 1$ and $tr(A) = r$. □

5.3.2 A Necessary Bound

Below we provide a rank-dependent, almost tight necessary condition for the minimum sample size required to obtain (5.2). This bound is easily computable in case that $r = rank(A)$ is known.

Before we proceed, recall the definition of the regularized Gamma functions

$$P\left(\alpha, \beta\right) := \frac{\gamma\left(\alpha, \beta\right)}{\Gamma\left(\alpha\right)}, \quad Q\left(\alpha, \beta\right) := \frac{\Gamma\left(\alpha, \beta\right)}{\Gamma\left(\alpha\right)},$$

where $\gamma\left(\alpha, \beta\right), \Gamma\left(\alpha, \beta\right)$ and $\Gamma\left(\alpha\right)$ are, respectively, the lower incomplete, the upper incomplete and the complete Gamma functions, see [1]. We also have that $\Gamma\left(\alpha\right) = \Gamma\left(\alpha, \beta\right) + \gamma\left(\alpha, \beta\right)$. Further, define

$$\Phi_\theta(x) := P\left(\frac{x}{2}, \frac{\tau(1-\theta)x}{2}\right) + Q\left(\frac{x}{2}, \frac{\tau(1+\theta)x}{2}\right), \tag{5.13a}$$

where

$$\tau = \frac{\ln(1+\theta) - \ln(1-\theta)}{2\theta}. \tag{5.13b}$$

Theorem 5.5. *Let A be a rank-r SPSD $s \times s$ matrix, and let (ε, δ) be a tolerance pair. If the inequality (5.2) with $\mathcal{D} = G$ holds for some n, then necessarily*

$$\Phi_\varepsilon(nr) \leq \delta. \tag{5.14}$$

Proof. As in the proof of Theorem 5.3 we have

$$\begin{aligned} Pr\left(|tr_G^n(A) - tr(A)| \leq \varepsilon\ tr(A)\right) &= Pr\left(|\sum_{i=1}^{n}\sum_{j=1}^{r} \lambda_j z_{ij}^2 - ntr(A)| \leq \varepsilon n tr(A)\right) \\ &= Pr\left((1-\varepsilon) \leq \sum_{i=1}^{n}\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)\ n} z_{ij}^2 \leq (1+\varepsilon)\right). \end{aligned}$$

Next, applying Theorem 3 of [129] gives

$$Pr\left(|tr_G^n(A) - tr(A)| \leq \varepsilon\ tr(A)\right) \leq Pr\left(c(1-\varepsilon) \leq \frac{1}{nr}\mathcal{X}_{nr}^2 \leq c(1+\varepsilon)\right),$$

where $\mathcal{X}_M^2$ denotes a chi-squared random variable of degree M with the cumulative distribution function (CDF)

$$CDF_{\mathcal{X}_M^2}(x) = Pr\left(\mathcal{X}_M^2 \leq x\right) = \frac{\gamma\left(\frac{M}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{M}{2}\right)}.$$

A further straightforward manipulation yields the stated result. □

Using the condition (5.14), we can establish a connection between inefficiency of the Gaussian estimator and the rank of matrices: Theorem 5.5 indicates that the *true* sample size, i.e., the minimum sample size for which (5.2) holds, is in fact in $\Omega(1/r)$. Hence, as the rank of a matrix becomes smaller, it becomes harder (i.e., a larger sample size is necessarily required) to estimate its trace, with the same probabilistic accuracy.

Having a computable necessary condition is practically useful: given a pair of fixed sample size n and error tolerance ε, the failure probability δ cannot be smaller than $\delta_0 = \Phi_\varepsilon(nr)$.

Since our sufficient bounds are not tight, it is not possible to make a direct comparison between the Hutchinson and Gaussian methods based on them. However, using this necessary condition can help for certain matrices. Consider a low rank matrix with a rather small $\mathcal{K}_H$ in (5.7). For such a matrix and a given pair (ε, δ), the condition (5.14) will probabilistically necessitate a rather large n, while (5.7) may give a much smaller sufficient bound for n. In this situation, using Theorem 5.5, the Hutchinson method is indeed guaranteed to require a smaller sample size than the Gaussian method.

The condition in Theorem 5.5 is almost tight in the following sense. Note that in (5.13b), $\tau \approx 1$ for $\theta = \varepsilon$ sufficiently small. So,

$$1 - \Phi_\varepsilon(nr)$$

would be very close to

$$Pr\left((1-\varepsilon) \leq tr_G^n(A^*) \leq (1+\varepsilon)\right),$$

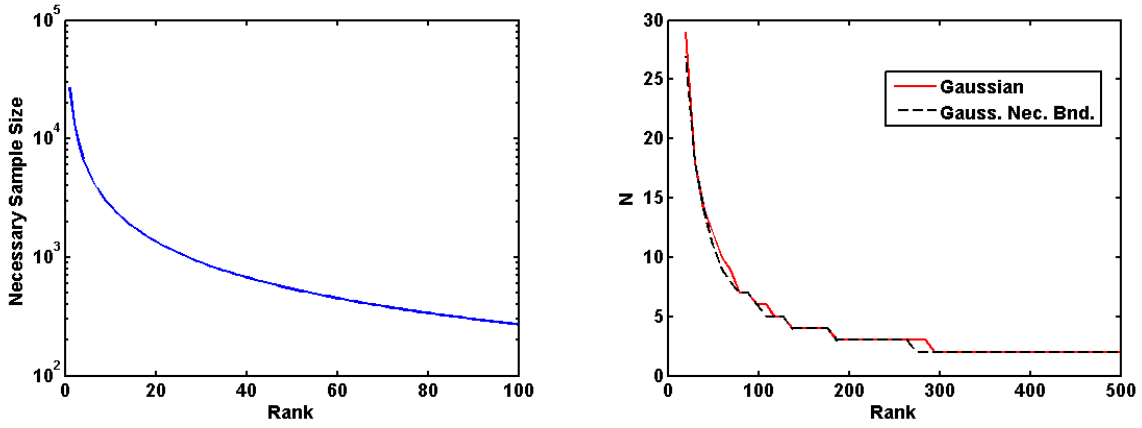
where A^* is an SPD matrix of the same rank as A whose eigenvalues are all equal to $1/r$. Next note that the condition (5.14) should hold for all matrices of the same rank; hence it is almost tight. Figures 5.1 and 5.4 demonstrate this effect.

Notice that for a very low rank matrix and a reasonable pair (ε, δ), the necessary n given by (5.14) could be even larger than the matrix size s, i.e., $n \geq s$, rendering the Gaussian method useless for such instances; see Figure 5.1.

Both of the conditions given in (5.5) and (5.14) are sharpened in Chapter 7, where tight (i.e., exact for some class of matrices) necessary and sufficient conditions are derived.

5.4 Random Unit Vector Bounds, with and without Replacement, for General Square Matrices

An alternative to the Hutchinson and Gaussian estimators is to draw the vectors $\mathbf{w}_i$ from among the s columns of the scaled identity matrix $\sqrt{s}\mathbb{I}$, i.e., we use a random subset of the vectors forming the scaled identity matrix. Note that if $\mathbf{w}_i$ is the ith (scaled) unit vector then



(a) $\varepsilon = \delta = 0.02$ (b) $n = 10,000,\ \varepsilon = \delta = 0.1$

Figure 5.1: Necessary bound for the Gaussian estimator: (a) the log-scale of n according to (5.14) as a function of $r = rank(A)$: larger ranks yield smaller necessary sample size. For very low rank matrices, the necessary bound grows significantly: for $s = 1000$ and $r \leq 30$, necessarily $n > s$ and the Gaussian method is practically useless; (b) tightness of the necessary bound demonstrated by an actual run as described for Example 5.4 in Section 5.5 where A has all eigenvalues equal.

$\mathbf{w}_i^T A \mathbf{w}_i = n a_{ii}$. Hence the trace can be recovered in $n = s$ deterministic steps upon setting in (5.1) $i = j,\ j = 1, 2, \ldots, s$. However, our hope is that for some matrices a good approximation for the trace can be recovered in $n \ll s$ such steps, with $\mathbf{w}_i$'s drawn as mentioned above.

There are typically two ways one can go about drawing such samples: with or without replacement. The first of these has been studied in [22]. However, in view of the exact procedure, we may expect to occasionally require smaller sample sizes by using the strategy of sampling without replacement. In this section we make this intuitive observation more rigorous.

In what follows, U_1 and U_2 refer to the uniform distribution of unit vectors with and without replacement, respectively. We first find expressions for the mean and variance of both strategies, obtaining a smaller variance for U_2.

Lemma 5.6. *Let A be an $s \times s$ matrix and let n denote the sample size. Then*

$$\begin{aligned}
\mathbb{E}\left(tr^n_{U_1}(A)\right) &= \mathbb{E}\left(tr^n_{U_2}(A)\right) = tr(A), && \text{(5.15a)}\\
Var\left(tr^n_{U_1}(A)\right) &= \frac{1}{n}\left(s\sum_{j=1}^{s} a_{jj}^2 - tr(A)^2\right), && \text{(5.15b)}\\
Var\left(tr^n_{U_2}(A)\right) &= \frac{(s-n)}{n(s-1)}\left(s\sum_{j=1}^{s} a_{jj}^2 - tr(A)^2\right),\ n \le s. && \text{(5.15c)}
\end{aligned}$$

Proof. The results for U_1 are proved in [22]. Let us next concentrate on U_2, and group the randomly selected unit vectors into an $s \times n$ matrix W. Then

$$\begin{aligned}
\mathbb{E}\left(tr^n_{U_2}(A)\right) &= \frac{1}{n}\mathbb{E}\left(tr\left(W^T A W\right)\right)\\
&= \frac{1}{n}\mathbb{E}\left(tr\left(A\, WW^T\right)\right)\\
&= \frac{1}{n} tr\left(A\,\mathbb{E}\left(WW^T\right)\right).
\end{aligned}$$

Let y_{ij} denote the (i,j)th element of the random matrix WW^T. Clearly, $y_{ij} = 0$ if $i \neq j$. It is also easily seen that y_{ii} can only take on the values 0 or s. We have

$$\mathbb{E}\left(y_{ii}\right) = sPr\left(y_{ii} = s\right) = s\frac{\binom{s-1}{n-1}}{\binom{s}{n}} = n,$$

so $\mathbb{E}(WW^T) = n \cdot \mathbb{I}$, where $\mathbb{I}$ stands for the identity matrix. This, in turn, gives $\mathbb{E}\left(tr^n_{U_2}(A)\right) = tr(A)$.

For the variance, we first calculate

$$\begin{aligned}
\mathbb{E}\left[\left(tr^n_{U_2}(A)\right)^2\right] &= \frac{1}{n^2}\mathbb{E}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{w}_i^T A \mathbf{w}_i\right)\left(\mathbf{w}_j^T A \mathbf{w}_j\right)\right)\\
= \frac{1}{n^2}\left(\sum_{i=1}^{n}\mathbb{E}\left[\left(\mathbf{w}_i^T A \mathbf{w}_i\right)^2\right] + \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\mathbb{E}\left[\left(\mathbf{w}_i^T A \mathbf{w}_i\right)\left(\mathbf{w}_j^T A \mathbf{w}_j\right)\right]\right) && \text{(5.16)}
\end{aligned}$$

Let $\mathbf{e}_j$ denote the j^{th} column of the scaled identity matrix, $\sqrt{s}I$. Using the law of total

expectation (i.e., the tower rule), we have for any two random vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ with $i \neq j$,

$$\begin{aligned}
\mathbb{E}\left[\left(\mathbf{w}_i^T A \mathbf{w}_i\right)\left(\mathbf{w}_j^T A \mathbf{w}_j\right)\right] &= \sum_{k=1}^{s} \mathbb{E}\left[\left(\mathbf{w}_i^T A \mathbf{w}_i\right)\left(\mathbf{w}_j^T A \mathbf{w}_j\right) | \mathbf{w}_i = \mathbf{e}_k\right] \cdot Pr\left(\mathbf{w}_i = \mathbf{e}_k\right) \\
&= \sum_{k=1}^{s} s a_{kk} \cdot \mathbb{E}\left[\left(\mathbf{w}_j^T A \mathbf{w}_j\right) | \mathbf{w}_i = \mathbf{e}_k\right] \cdot \frac{1}{s} \\
&= \sum_{k=1}^{s} a_{kk} \sum_{\substack{l=1 \\ l \neq k}}^{s} \mathbb{E}\left[\left(\mathbf{w}_j^T A \mathbf{w}_j\right) | \mathbf{w}_j = \mathbf{e}_l\right] \cdot Pr\left(\mathbf{w}_j = \mathbf{e}_l | \, \mathbf{w}_i = \mathbf{e}_k\right) \\
&= \sum_{k=1}^{s} a_{kk} \sum_{\substack{l=1 \\ l \neq k}}^{s} s a_{ll} \frac{1}{s-1} \\
&= \frac{s}{s-1} \sum_{k=1}^{s} \sum_{\substack{l=1 \\ k \neq l}}^{s} a_{kk} a_{ll} \\
&= \frac{s}{s-1} (tr(A)^2 - \sum_{j=1}^{s} a_{jj}^2).
\end{aligned}$$

Substituting this in (5.16) gives

$$\mathbb{E}\left[\left(tr_{U_2}^n(A)\right)^2\right] = \frac{1}{n^2}\left(sn \sum_{j=1}^{s} a_{jj}^2 + \frac{sn(n-1)}{s-1}\left(tr(A)^2 - \sum_{j=1}^{s} a_{jj}^2\right)\right).$$

Next, the variance is

$$Var\left(tr_{U_2}^n(A)\right) = \mathbb{E}\left[\left(tr_{U_2}^n(A)\right)^2\right] - \left[\mathbb{E}\left(tr_{U_2}^n(A)\right)\right]^2,$$

which gives (5.15c). □

Note that $Var\left(tr_{U_2}^n(A)\right) = \frac{s-n}{s-1} Var\left(tr_{U_1}^n(A)\right)$. The difference in variance between these sampling strategies is small for $n \ll s$, and they coincide if $n = 1$. Moreover, in case that the diagonal entries of the matrix are all equal, the variance for both sampling strategies vanishes.

We now turn to the analysis of the sample size required to ensure (5.2) and find a slight improvement over the bound given in [22] for U_1. A similar analysis for the case of sampling without replacement shows that the latter may generally be a somewhat better strategy.

Theorem 5.7. *Let A be a real $s \times s$ matrix, and denote*

$$\mathcal{K}_U^{(i,j)} = \frac{s}{tr(A)} \left| a_{ii} - a_{jj} \right|, \quad \mathcal{K}_U = \max_{\substack{1 \le i,j \le s \\ i \neq j}} \mathcal{K}_U^{(i,j)}. \tag{5.17}$$

Given a pair of positive small values (ε, δ), the inequality (5.2) holds with $\mathcal{D} = U_1$ if

$$n > \frac{\mathcal{K}_U^2}{2} c(\varepsilon, \delta) \equiv \mathcal{F}, \tag{5.18}$$

and with $\mathcal{D} = U_2$ if

$$n \geq \frac{s+1}{1 + \frac{s-1}{\mathcal{F}}}. \tag{5.19}$$

Proof. This proof is refreshingly short. Note first that every sample of these estimators takes on a Rayleigh value in $[s \min_j a_{jj}, s \max_j a_{jj}]$.

The proof of (5.18), for the case with replacement, uses Hoeffding's inequality in exactly the same way as the corresponding theorem in [22]. We obtain directly that if (5.18) is satisfied then (5.2) holds with $\mathcal{D} = U_1$.

For the case without replacement we use Serfling's inequality [125] to obtain

$$Pr\left(|tr_{U_2}^n(A) - tr(A)| \geq \varepsilon tr(A) \right) \leq 2 \exp\left\{ \frac{-2n\varepsilon^2}{(1 - f_{n-1}) \mathcal{K}_U^2} \right\},$$

where f_n is the sampling fraction defined as

$$f_n = \frac{n-1}{s-1}.$$

Now, for the inequality (5.2) to hold, we need

$$2 \exp\left\{ \frac{-2n\varepsilon^2}{(1 - f_{n-1}) \mathcal{K}_U^2} \right\} \leq \delta,$$

so we require that

$$\frac{n}{1 - f_{n-1}} \geq \mathcal{F}.$$

The stated result (5.19) is obtained following some straightforward algebraic manipulation. □

Looking at the bounds (5.18) for U_1 and (5.19) for U_2 and observing the expression (5.17) for $\mathcal{K}_U$, one can gain insight as to the type of matrices which are handled efficiently using this estimator: this would be the case if the diagonal elements of the matrix all have similar values. In the extreme case where they are all the same, we only need one sample. The corresponding expression in [22] does not reflect this result.

An illustration of the relative behaviour of the two bounds is given in Figure 5.2.

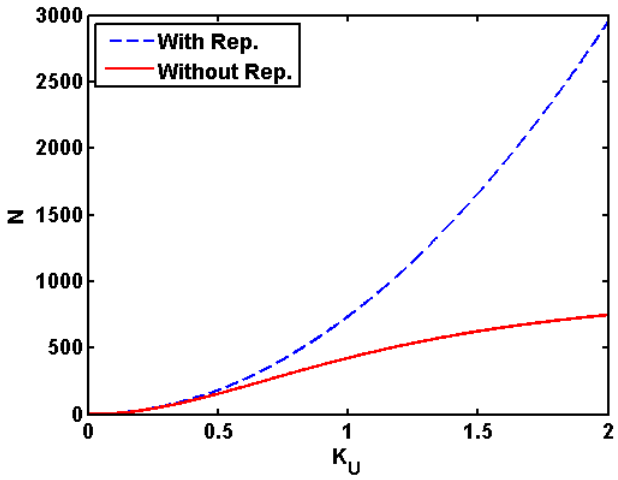


Figure 5.2: The behaviour of the bounds (5.18) and (5.19) with respect to the factor $K = \mathcal{K}_U$ for $s = 1000$ and $\varepsilon = \delta = 0.05$. The bound for U_2 is much more resilient to the distribution of the diagonal values than that of U_1. For very small values of $\mathcal{K}_U$, there is no major difference between the bounds.

5.5 Numerical Examples

In this section we experiment with several examples, comparing the performance of different methods with regards to various matrix properties and verifying that the bounds obtained in our theorems indeed agree with the numerical experiments.

Example 5.1. *In this example we do not consider δ at all. Rather, we check numerically for various values of ε what value of n is required to achieve a result respecting this relative tolerance. We have calculated maximum and average values for n over 100 trials for several special examples, verifying numerically the following considerations.*

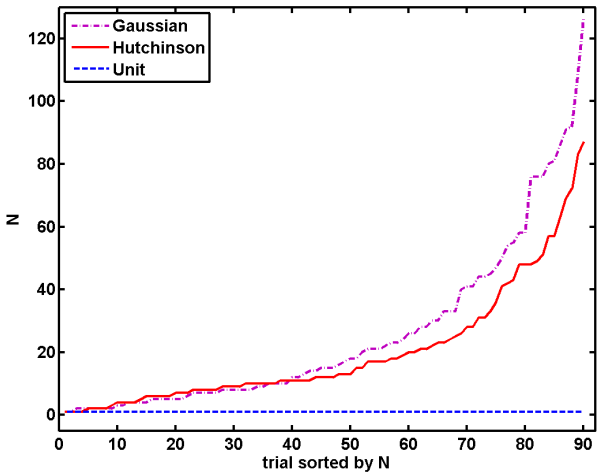


Figure 5.3: Example 5.1. For the matrix of all 1s with $s = 10,000$, the plot depicts the numbers of samples in 100 trials required to satisfy the relative tolerance $\varepsilon = .05$, sorted by increasing n. The average n for both Hutchinson and Gauss estimators was around 50, while for the uniform unit vector estimator always $n = 1$. Only the best 90 results (i.e., lowest resulting values of n) are shown for reasons of scaling. Clearly, the unit vector method is superior here.

- *The matrix of all* 1*s (in* MATLAB, `A=ones(s,s)`*) has been considered in [22]. Here* $tr(A) = s$, $\mathcal{K}_H = s - 1$, *and a very large* n *is often required if* ε *is small for both Hutchinson and Gauss methods. For the unit vector method, however,* $\mathcal{K}_U = 0$ *in* (5.17), *so the latter method converges in one iteration,* $n = 1$. *This fact yields an example where the unit vector estimator is far better than either Hutchinson or Gaussian estimators; see Figure 5.3.*

- *Another extreme example, where this time it is the Hutchinson estimator which requires only one sample whereas the other methods may require many more, is the case of a diagonal matrix* A. *For a diagonal matrix,* $\mathcal{K}_H = 0$, *and the result follows from Theorem 5.2.*

- *If* A *is a multiple of the identity then, since* $\mathcal{K}_U = \mathcal{K}_H = 0$, *only the Gaussian estimator from among the methods considered requires more than one sample; thus, it is worst.*

- *Examples where the unit vector estimator is consistently (and significantly) worst are obtained by defining* $A = Q^T DQ$ *for a diagonal matrix* D *with different positive elements which are of the same order of magnitude and a nontrivial orthogonal matrix* Q.

- *We have not been able to come up with a simple example of the above sort where the Gaus-*

sian estimator shines over both others, although we have seen many occasions in practice where it slightly outperforms the Hutchinson estimator with both being significantly better than the unit vector estimators.

Example 5.2. *Consider the matrix* $A = \mathbf{x}\mathbf{x}^T/\|\mathbf{x}\|^2$, *where* $\mathbf{x} \in \mathbb{R}^s$, *and for some* $\theta > 0$, $x_j = \exp(-j\theta)$, $1 \le j \le s$. *This extends the example of all 1s of Figure 5.3 (for which* $\theta = 0$*) to instances with rapidly decaying elements.*

It is easy to verify that

$$
\begin{aligned}
&tr(A) = 1, \quad r = 1, \quad \mathcal{K}_G = 1,\\
&\mathcal{K}_H^j = \|\mathbf{x}\|^2 x_j^{-2} - 1, \quad \mathcal{K}_H = \|\mathbf{x}\|^2 x_s^{-2} - 1,\\
&\mathcal{K}_U^{(i,j)} = \frac{s}{\|\mathbf{x}\|^2}|x_i^2 - x_j^2|, \quad \mathcal{K}_U = \frac{s}{\|\mathbf{x}\|^2}(x_1^2 - x_s^2),\\
&\|\mathbf{x}\|^2 = \frac{\exp(-2\theta) - \exp(-2(s+1)\theta)}{1 - \exp(-2\theta)}.
\end{aligned}
$$

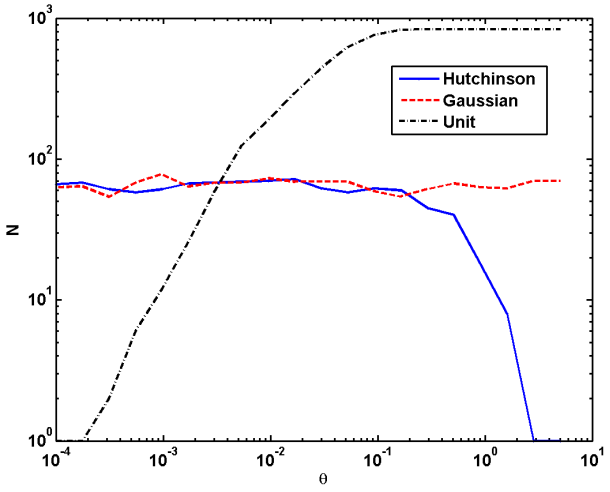


Figure 5.4: Example 5.2. For the rank-1 matrix arising from a rapidly-decaying vector with $s = 1000$, this log-log plot depicts the actual sample size n required for (5.2) to hold with $\varepsilon = \delta = 0.2$, vs. various values of θ. In the legend, "Unit" refers to the random sampling method without replacement.

Figure 5.4 displays the "actual sample size" n for a particular pair (ε, δ) *as a function of* θ *for the three distributions. The values n were obtained by running the code 100 times for each* θ *to calculate the empirical probability of success.*

In this example the distribution of $\mathcal{K}_H^j$ values gets progressively worse with heavier tail values as θ gets larger. However, recall that this matters in terms of the sufficient bounds (5.4) *and* (5.7) *only so long as $\mathcal{K}_H < 3$. Here the crossover point happens roughly when $\theta \sim 1/(2s)$. Indeed, for large values of θ the required sample size actually drops when using the Hutchinson method: Theorem 5.2, being only a sufficient condition, merely distinguishes types of matrices for which Hutchinson is expected to be* efficient*, while making no claim regarding those matrices for which it is an inefficient estimator.*

On the other hand, Theorem 5.5 clearly distinguishes the types of matrices for which the Gaussian method is expected to be inefficient*, because its condition is necessary rather than sufficient. Note that n (the red curve in Figure 5.4) does not change much as a function of θ, which agrees with the fact that the matrix rank stays fixed and low at $r = 1$.*

The unit vector estimator, unlike Hutchinson, deteriorates steadily as θ is increased, because this estimator ignores off-diagonal elements. However, for small enough values of θ the $\mathcal{K}_U^{(i,j)}$'s are spread tightly near zero, and the unit vector method, as predicted by Theorem 5.7, requires a very small sample size.

For Examples 5.3 and 5.5 below, given (ε, δ), we plot the probability of success, i.e., $Pr\left(|tr_{\mathcal{D}}^n(A) - tr(A)| \leq \varepsilon\ tr(A)\right)$ for increasing values of n, starting from $n = 1$. We stop when for a given n, the probability of success is greater than or equal to $1 - \delta$. In order to evaluate this for each n, we run the experiments 500 times and calculate the empirical probability.

In the figures below, 'With Rep.' and 'Without Rep.' refer to uniform unit sampling with and without replacement, respectively. In all cases, by default, $\varepsilon = \delta = .05$. We also provide distribution plots of the quantities $\mathcal{K}_H^j$, $\mathcal{K}_G^j$ and $\mathcal{K}_U^{(i,j)}$ appearing in (5.6), (5.9) and (5.17), respectively. These quantities are indicators for the performance of the Hutchinson, Gaussian and unit vector estimators, respectively, as evidenced not only by Theorems 5.2, 5.3 and 5.7, but also in Examples 5.1 and 5.2, and by the fact that the performance of the Gaussian and unit vector estimators is not affected by the energy of the off-diagonal matrix elements.

Example 5.3 (Data fitting with many experiments)**.** *A major source of applications where trace estimation is central arises in problems involving least squares data fitting with many experiments (cf. Chapter 1). In its simplest, linear form, we look for $\mathbf{m} \in \mathbb{R}^m$ so that the*

misfit function

$$\phi(\mathbf{m}) = \sum_{i=1}^{s} \|J_i\mathbf{m} - \mathbf{d}_i\|^2, \tag{5.20a}$$

for given data sets $\mathbf{d}_i \in \mathbb{R}^l$ *and sensitivity matrices* J_i*, is either minimized or reduced below some tolerance level. The* $l \times m$ *matrices* J_i *are very expensive to calculate and store, so this is avoided altogether, but evaluating* $J_i\mathbf{m}$ *for any suitable vector* $\mathbf{m}$ *is manageable. Moreover,* s *is large. Next, writing* (5.20a) *using the Frobenius norm as*

$$\phi(\mathbf{m}) = \|C\|_F^2, \tag{5.20b}$$

where C *is* $l \times s$ *with the* j*th column* $C_j = J_j\mathbf{m} - \mathbf{d}_j$*, and defining the SPSD matrix* $A = C^TC$*, we have*

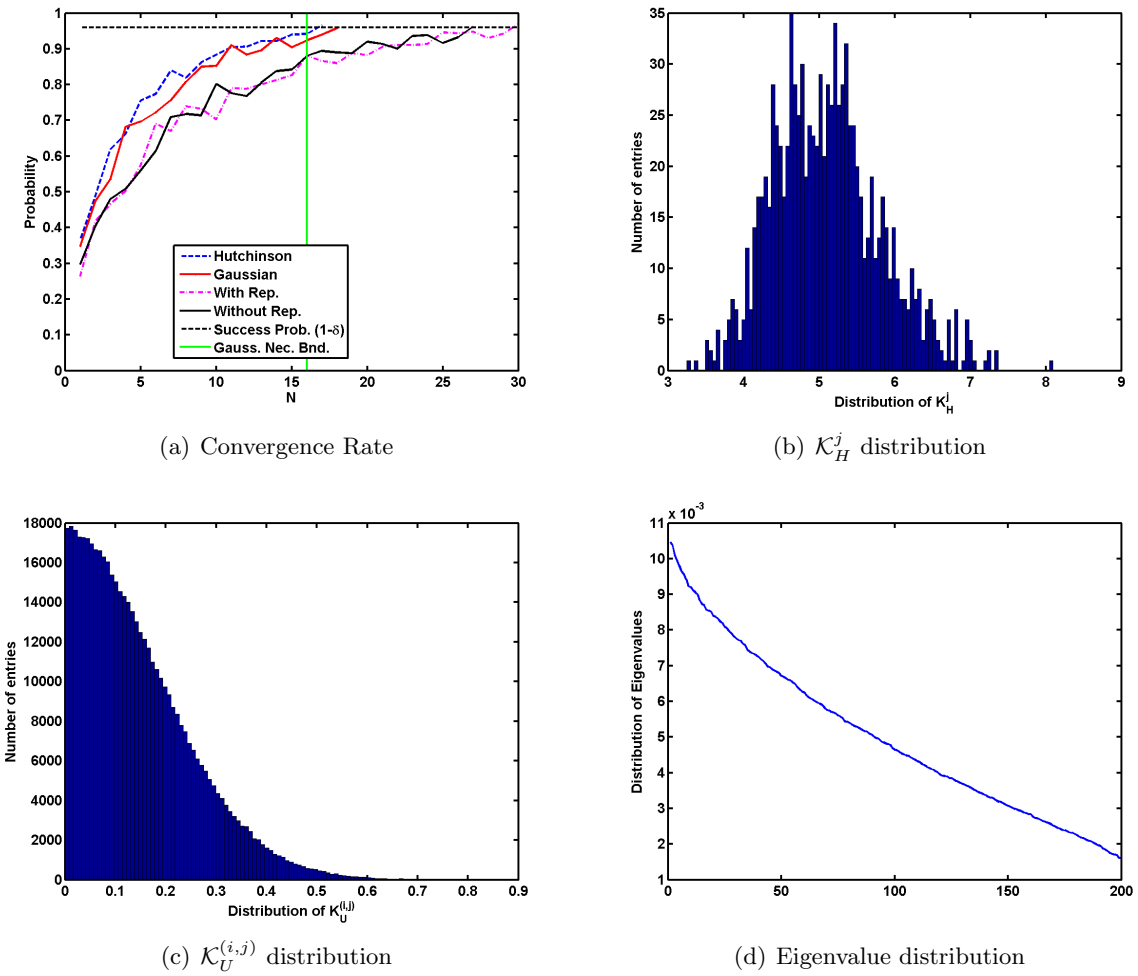
$$\phi(\mathbf{m}) = tr(A(\mathbf{m})). \tag{5.20c}$$

Cheap estimates of the misfit function $\phi(\mathbf{m})$ *are then sought by approximating the trace in* (5.20c) *using only* n *(rather than* s*) linear combinations of the columns of* C*, which naturally leads to expressions of the form* (5.1)*. Hutchinson and Gaussian estimators in a similar or more complex context were considered in [71, 134, 138].*

Drawing the $\mathbf{w}_i$ *as random unit vectors instead is a method proposed in [46] and compared to others in Chapter 3, where it is called "random subset": this latter method can have efficiency advantages that are beyond the scope of the presentation here. Typically,* $l \ll s$*, and thus the matrix* A *is dense and often has low rank.*

Furthermore, the signs of the entries in C *can be, at least to some extent, considered random. Hence we consider below matrices* $A = C^TC$ *whose entries are Gaussian random variables, obtained using the* MATLAB *command* `C = randn(l,s)`*. We use* $l = 200$ *and hence the rank is, almost surely,* $r = 200$*.*

It can be seen from Figure 5.5(a) that the Hutchinson and the Gaussian methods perform similarly here. The sample size required by both unit vector estimators is approximately twice that of the Gaussian and Hutchinson methods. This relative behaviour agrees with our observa-



(a) Convergence Rate

(b) $\mathcal{K}_H^j$ distribution

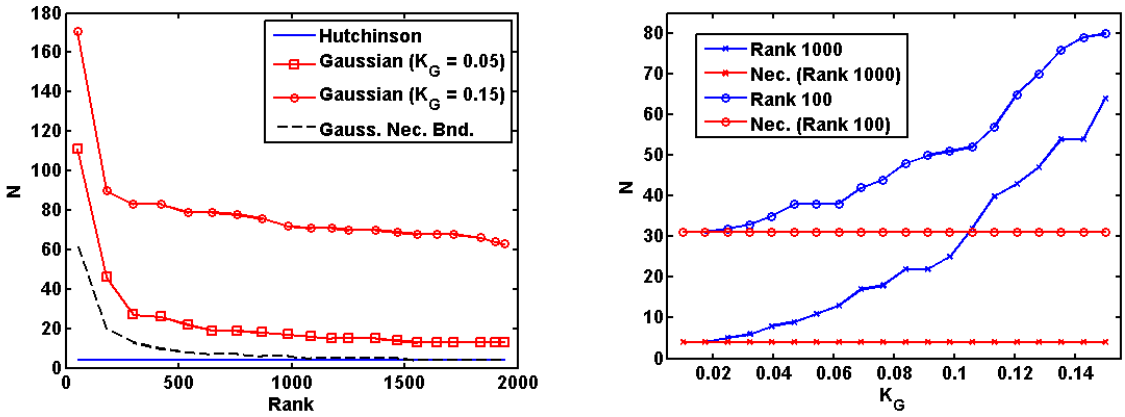
(c) $\mathcal{K}_U^{(i,j)}$ distribution

(d) Eigenvalue distribution

Figure 5.5: Example 5.3. A dense SPSD matrix A is constructed using MATLAB's `randn`. Here $s = 1000$, $r = 200$, $tr(A) = 1, \mathcal{K}_G = 0.0105$, $\mathcal{K}_H = 8.4669$ and $\mathcal{K}_U = 0.8553$. The method convergence plots in (a) are for $\varepsilon = \delta = .05$.

tions in the context of actual application as described above, see Chapter 3. From Figure 5.5(d), the eigenvalue distribution of the matrix is not very badly skewed, which helps the Gaussian method perform relatively well for this sort of matrix. On the other hand, by Figure 5.5(b) the relative ℓ_2 energies of the off-diagonals are far from being small, which is not favourable for the Hutchinson method. These two properties, in combination, result in the similar performance of the Hutchinson and Gaussian methods despite the relatively low rank. The contrast between $\mathcal{K}_U^{(i,j)}$'s is not too large according to Figure 5.5(c), hence a relatively decent performance of both unit vector (or, random sampling) methods is observed. There is no reason to insist on avoiding repetition here either.

Example 5.4 (Effect of rank and $\mathcal{K}_G$ on the Gaussian estimator)**.** *In this example we plot the actual sample size n required for* (5.2) *to hold. In order to evaluate* (5.2)*, we repeat the experiments 500 times and calculate the empirical probability. In all experiments, the sample sizes predicted by* (5.4) *and* (5.5) *were so pessimistic compared with the true n that we simply did not include them in the plots.*



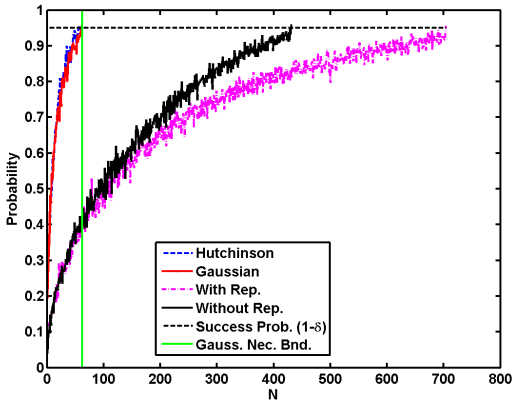
(a) `sprandn`, $s = 5,000$ (b) diagonal, $s = 10,000$

Figure 5.6: Example 5.4. The behaviour of the Gaussian method with respect to rank and $\mathcal{K}_G$. We set $\varepsilon = \delta = .05$ and display the necessary condition (5.14) as well.

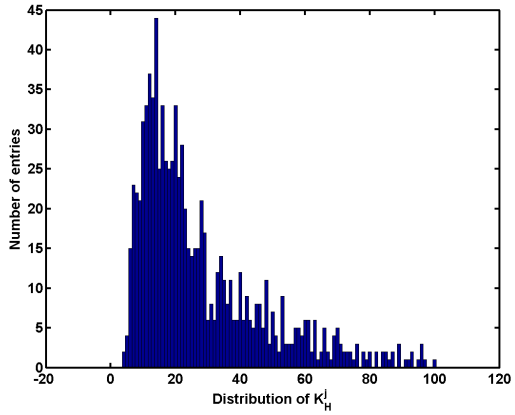
In order to concentrate only on rank and $\mathcal{K}_G$ variation, we make sure that in all experiments $\mathcal{K}_H \ll 1$. For the results displayed in Figure 5.6(a), where r is varied for each of two values of $\mathcal{K}_G$, this is achieved by playing with MATLAB*'s normal random generator function* `sprandn`*. For Figure 5.6(b), where $\mathcal{K}_G$ is varied for each of two values of r, diagonal matrices are utilized: we start with a uniform distribution of the eigenvalues and gradually make this distribution more skewed, resulting in an increased $\mathcal{K}_G$. The low $\mathcal{K}_H$ values cause the Hutchinson method to look very good, but that is not our focus here.*

It can be clearly seen from Figure 5.6(a) that as the matrix rank gets lower, the sample size required for the Gaussian method grows significantly. For a given rank, the matrix with a smaller $\mathcal{K}_G$ requires smaller sample size. From Figure 5.6(b) it can also be seen that for a fixed rank, the matrix with more skewed $\mathcal{K}_G^j$'s distribution (marked here by a larger $\mathcal{K}_G$) requires a larger sample size.

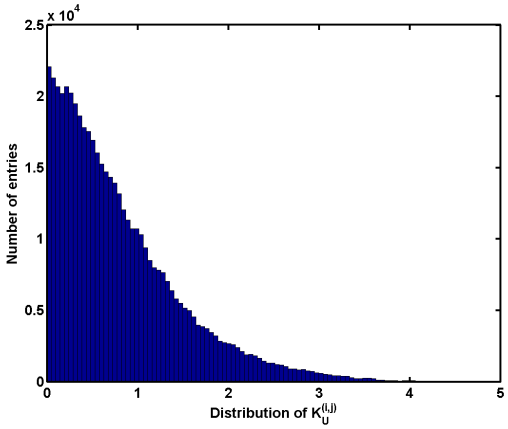
Example 5.5 (Method performance for different matrix properties)**.** *Next we consider a much*



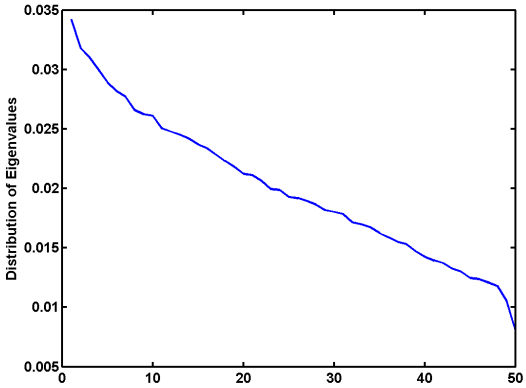
(a) Convergence Rate



(b) $\mathcal{K}_H^j$ distribution ($\mathcal{K}_H^j \leq 100$)



(c) $\mathcal{K}_U^{(i,j)}$ distribution

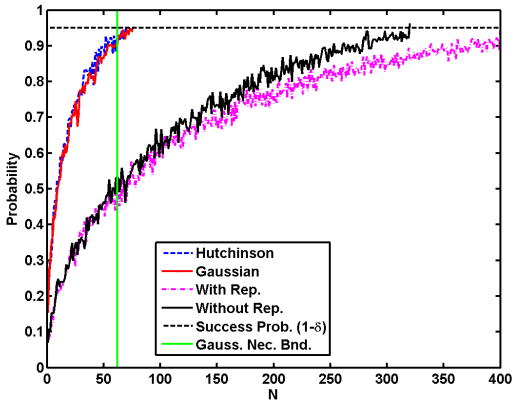


(d) Eigenvalue distribution

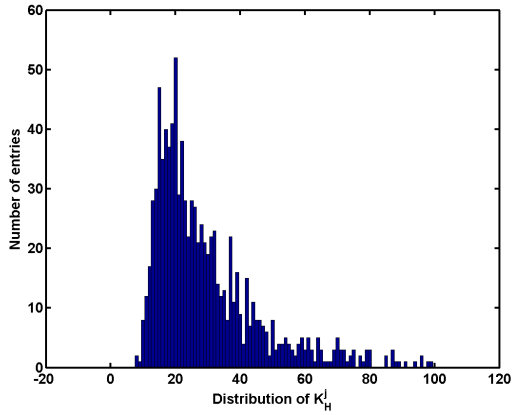
Figure 5.7: Example 5.5. A sparse matrix `(d = 0.1)` is formed using `sprandn`. Here $r = 50$, $\mathcal{K}_G = 0.0342$, $\mathcal{K}_H = 15977.194$ and $\mathcal{K}_U = 4.8350$.

more general setting than that in Example 5.4, and compare the performance of different methods with respect to various matrix properties. The matrix A is constructed as in Example 5.3, except that also a uniform distribution is used. Furthermore, a parameter d controlling denseness of the created matrix is utilized. This is achieved in MATLAB *using the commands* `C=sprandn(l,s,d)` *or* `C=sprand(l,s,d)`*. By changing* `l` *and* `d` *we can change the matrix properties $\mathcal{K}_H$, $\mathcal{K}_G$ and $\mathcal{K}_U$ while keeping the rank r fixed across experiments. We maintain $s = 1000$, $tr(A) = 1$ and $\varepsilon = \delta = .05$ throughout. In particular, the four figures related to this example are comparable to Figure 5.5 but for a lower rank.*

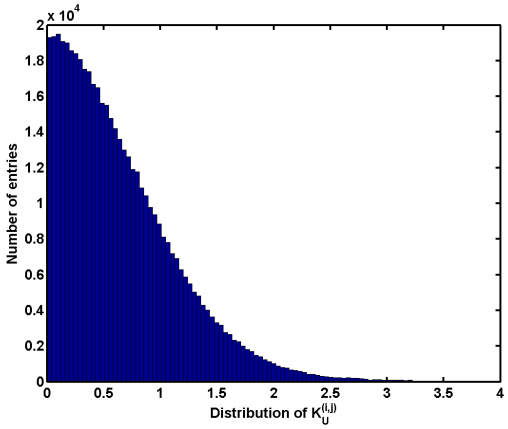
By comparing Figures 5.7 and 5.8, as well as 5.9 and 5.10, we can see how not only the values of $\mathcal{K}_H$, $\mathcal{K}_G$ and $\mathcal{K}_U$, but also the distribution of the quantities they maximize matters.



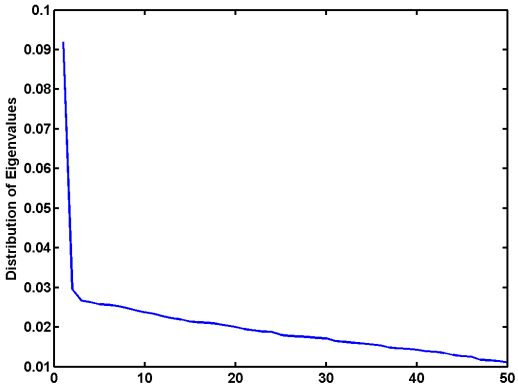
(a) Convergence Rate



(b) $\mathcal{K}_H^j$ distribution ($\mathcal{K}_H^j \leq 100$)



(c) $\mathcal{K}_U^{(i,j)}$ distribution

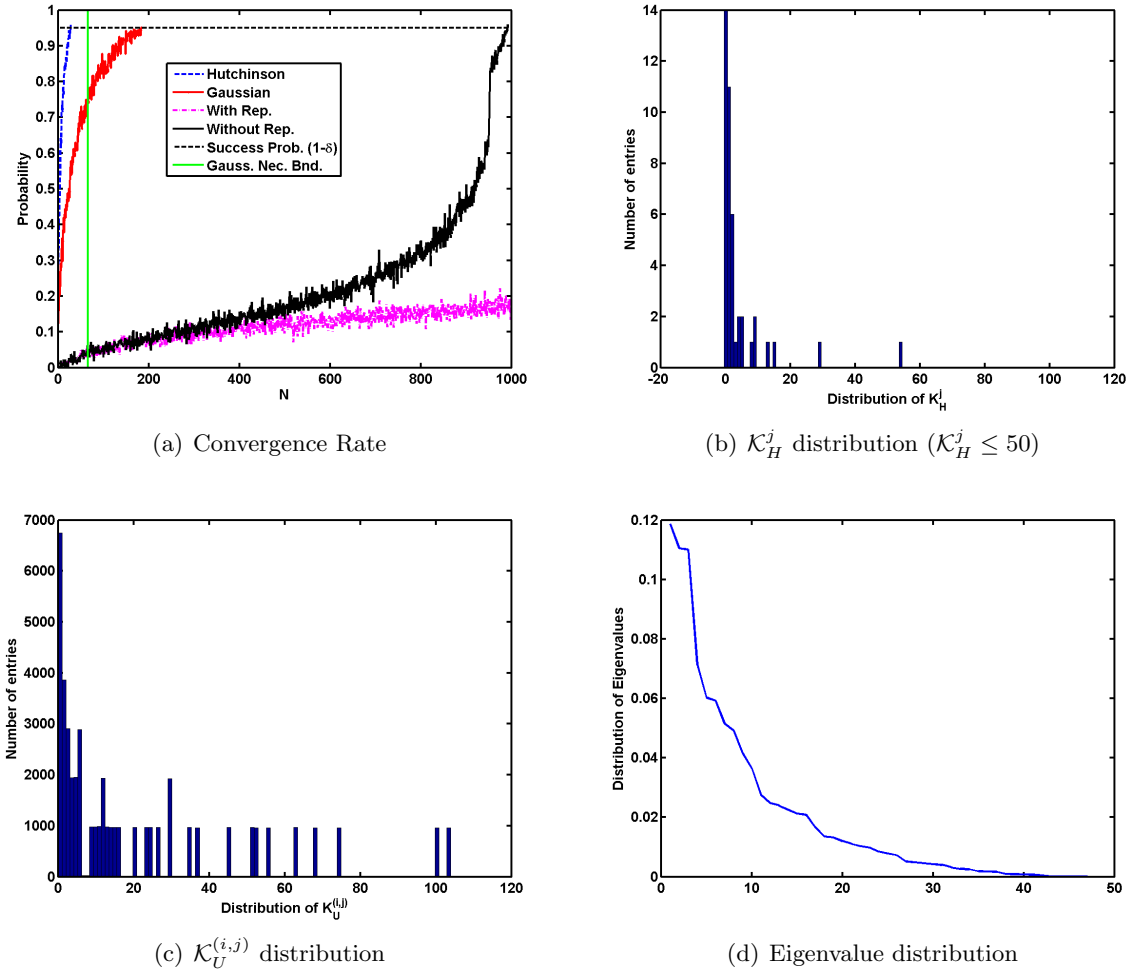


(d) Eigenvalue distribution

Figure 5.8: Example 5.5. A sparse matrix `(d = 0.1)` is formed using `sprand`. Here $r = 50, \mathcal{K}_G = 0.0919$, $\mathcal{K}_H = 11624.58$ and $\mathcal{K}_U = 3.8823$.

Note how the performance of both unit vector strategies is negatively affected with increasing average values of $\mathcal{K}_U^{(i,j)}$'s. From the eigenvalue (or $\mathcal{K}_G^j$) distribution of the matrix, it can also be seen that the Gaussian estimator is heavily affected by the skewness of the distribution of the eigenvalues (or $\mathcal{K}_G^j$'s): given the same r and s, as this eigenvalue distribution becomes increasingly uneven, the Gaussian method requires larger sample size.

Note that comparing the performance of the methods on different matrices solely based on their values $\mathcal{K}_H$, $\mathcal{K}_G$ or $\mathcal{K}_U$ can be misleading. This can be seen for instance by considering the performance of the Hutchinson method in Figures 5.7, 5.8, 5.9 and 5.10 and comparing their respective $\mathcal{K}_H^j$ distributions as well as $\mathcal{K}_H$ values. Indeed, none of our 6 sufficient bounds can be guaranteed to be generally tight. As remarked also earlier, this is an artifact of the generality



(a) Convergence Rate

(b) $\mathcal{K}_H^j$ distribution ($\mathcal{K}_H^j \leq 50$)

(c) $\mathcal{K}_U^{(i,j)}$ distribution

(d) Eigenvalue distribution

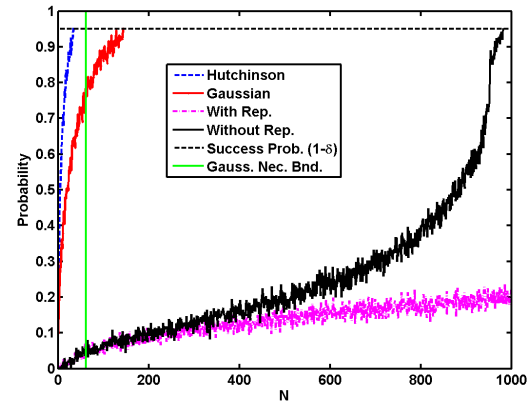
Figure 5.9: Example 5.5. A very sparse matrix `(d = 0.01)` is formed using `sprandn`. Here $r = 50$, $\mathcal{K}_G = 0.1186$, $\mathcal{K}_H = 8851.8$ and $\mathcal{K}_U = 103.9593$.

of the proved results.

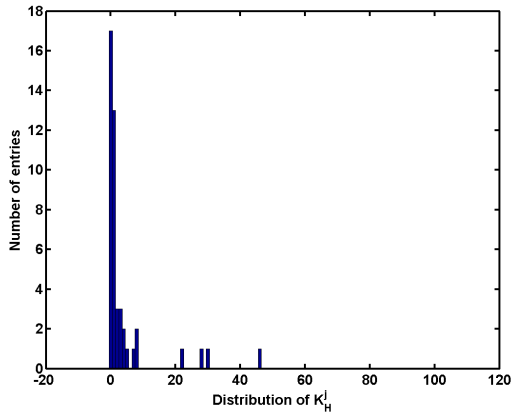
Note also that rank and eigenvalue distribution of a matrix have no direct effect on the performance of the Hutchinson method: by Figures 5.9 and 5.10 it appears to only depend on the $\mathcal{K}_H^j$ distribution. In these figures, one can observe that the Gaussian method is heavily affected by the low rank and the skewness of the eigenvalues. Thus, if the distribution of $\mathcal{K}_H^j$'s is favourable to the Hutchinson method and yet the eigenvalue distribution is rather skewed, we can expect a significant difference between the performance of the Gaussian and Hutchinson methods.

5.6 Conclusions

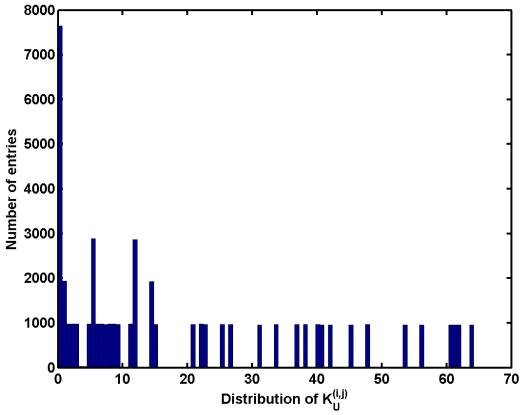
In this chapter, we have proved six sufficient bounds for the minimum sample size n required to reach, with probability $1-\delta$, an approximation for $tr(A)$ to within a relative tolerance ε. Two such bounds apply to each of the three estimators considered in Sections 5.2, 5.3 and 5.4, respectively. In Section 5.3 we have also proved a necessary bound for the Gaussian estimator. These bounds have all been verified numerically through many examples, some of which are summarized in Section 5.5.



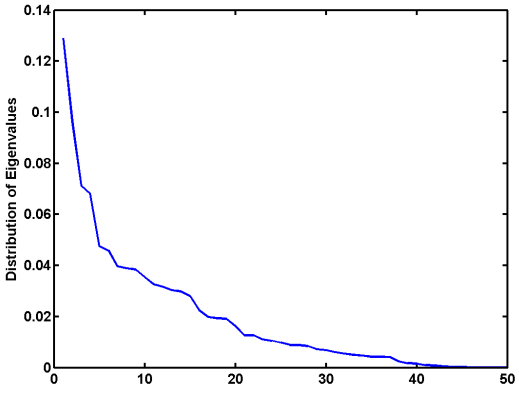
(a) Convergence Rate



(b) $\mathcal{K}_H^j$ distribution ($\mathcal{K}_H^j \leq 50$)



(c) $\mathcal{K}_U^{(i,j)}$ distribution



(d) Eigenvalue distribution

Figure 5.10: Example 5.5. A very sparse matrix `(d = 0.01)` is formed using `sprand`. Here $r = 50$, $\mathcal{K}_G = 0.1290$, $\mathcal{K}_H = 1611.34$ and $\mathcal{K}_U = 64.1707$.

Two of these bounds, namely, (5.4) for Hutchinson and (5.5) for Gaussian, are immediately computable without knowing anything else about the SPSD matrix A. In particular, they are

independent of the matrix size s. As such they may be very pessimistic. And yet, in some applications (for instance, in exploration geophysics) where s can be very large and ε need not be very small due to uncertainty, these bounds may indeed provide the comforting assurance that $n \ll s$ suffices (say, s is in the millions and n in the thousands). Generally, these two bounds have the same quality.

The underlying objective in this work, which is to seek a small n satisfying (5.2), is a natural one for many applications and follows that of other works. But when it comes to comparing different methods, it is by no means the only performance indicator. For example, variance can also be considered as a ground to compare different methods. However, one needs to exercise caution to avoid basing the entire comparison solely on variance: it is possible to generate examples where a linear combination of $\mathcal{X}^2$ random variables has smaller variance, yet higher tail probability.

The lower bound (5.14) that is available only for the Gaussian estimator may allow better prediction of the actual required n, in cases where the rank r is known. At the same time it also implies that the Gaussian estimator can be inferior in cases where r is small. The Hutchinson estimator does not enjoy a similar theory, but empirically does not suffer from the same disadvantage either.

The matrix-dependent quantities $\mathcal{K}_H$, $\mathcal{K}_G$ and $\mathcal{K}_U$, defined in (5.6), (5.9) and (5.17), respectively, are not easily computable for any given implicit matrix A. However, the results of Theorems 5.2, 5.3 and 5.7 that depend on them can be more indicative than the general bounds. In particular, examples where one method is clearly better than the others can be isolated in this way. At the same time, the sufficient conditions in Theorems 5.2, 5.3 and 5.7, merely distinguish the types of matrices for which the respective methods are expected to be efficient, and make no claims regarding those matrices for which they are inefficient estimators. This is in direct contrast with the necessary condition in Theorem 5.5.

It is certainly possible in some cases for the required n to go over s. In this connection, it is important to always remember the deterministic method which obtains $tr(A)$ in s applications of unit vectors: if n grows above s in a particular stochastic setting then it may be best to abandon ship and choose the safe, deterministic way.

Chapter 6

Extremal Probabilities of Linear Combinations of Gamma Random Variables

This chapter prepares us for Chapter 7, in the sense that two pivotal results, given in Theorems 6.1 and 6.2 below, are subsequently used there. However, the development here is significantly more general than what is needed for Chapter 7, and the results are novel. Hence we describe them in a separate chapter, as they can be considered independently of the rest of this thesis.

The gamma distribution forms an important family of distributions, and gamma random variables (r.v's) appear in many practical applications. For example, linear combinations (i.e., convolutions) of independent gamma r.v's often naturally arise in many applications in statistics, engineering, insurance, actuarial science and reliability. As such, in the literature, there has been extensive study of the stochastic properties of gamma r.v's and their convolutions. For examples of such theoretical studies as well as applications see [7, 30–32, 44, 61, 93–95, 99, 103, 129, 139, 141, 142] and references therein.

In what follows, let $X \sim Gamma(\alpha, \beta)$ denote a gamma distributed random variable (r.v) parametrized by shape α and rate β parameters with the probability density function (PDF)

$$f(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} & x \geq 0 \\ 0 & x < 0 \end{cases} . \tag{6.1}$$

An important stochastic property of gamma r.v is that of the monotonicity of regularized gamma function (see Section 5.3.2), i.e., cumulative distribution function (CDF) of gamma r.v,

with respect to different shape α and rate β parameters. Theorems 6.1 gives conditions which allow one to obtain certain important monotonicity results for the regularized gamma function (cf. (5.13)).

Theorem 6.1 (**Monotonicity of cumulative distribution function of gamma r.v**). *Given parameters $0 < \alpha_1 < \alpha_2$, let $X_i \sim Gamma(\alpha_i, \alpha_i)$, $i = 1, 2$, be independent r.v's, and define*

$$\Delta(x) := \Pr(X_2 < x) - \Pr(X_1 < x).$$

Then we have that

(i) there is a unique point $x(\alpha_1, \alpha_2)$ such that $\Delta(x) < 0$ for $0 < x < x(\alpha_1, \alpha_2)$ and $\Delta(x) > 0$ for $x > x(\alpha_1, \alpha_2)$,

(ii) $1 \leq x(\alpha_1, \alpha_2) \leq \frac{2\sqrt{\alpha_1(\alpha_2-\alpha_1)}+1}{2\sqrt{\alpha_1(\alpha_2-\alpha_1)}}$.

Another important stochastic property, is that of the maximum and minimum of tail probabilities of linear combinations of i.i.d gamma r.v's. More specifically, let $X_i \sim Gamma(\alpha, \beta)$ for $i = 1, 2, \ldots, n$, be n i.i.d gamma r.v's. Consider the following non-negative linear combinations of such r.v's

$$\sum_{i=1}^{n} \lambda_i X_i,$$

where $\lambda_i \geq 0$, $i = 1, 2, \ldots, n$, are real numbers. The goal is to find conditions allowing one to determine the maximum and minimum of tail probability

$$\Pr\left(\sum_{i=1}^{n} \lambda_i X_i < x\right),$$

with respect to the mixing weights λ_i, $i = 1, \ldots, n$ for various values of x. Theorem 6.2 describes these conditions.

Theorem 6.2 (**Extremal probabilities of linear combination of gamma r.v's**). *Given shape and rate parameters $\alpha, \beta > 0$, let $X_i \sim Gamma(\alpha, \beta)$, $i = 1, 2, \ldots, n$, be i.i.d gamma r.v's, and define*

$$\Theta := \{\boldsymbol{\lambda} = \left(\lambda_1, \lambda_2, \ldots, \lambda_n\right)^T \mid \lambda_i \geq 0 \; \forall i, \; \sum_{i=1}^{n} \lambda_i = 1\}.$$

Then we have

$$m_n(x) := \min_{\boldsymbol{\lambda}\in\Theta} \Pr\left(\sum_{i=1}^{n} \lambda_i X_i < x\right) = \begin{cases} \Pr\left(\frac{1}{n}\sum_{i=1}^{n} X_i < x\right), & x < \frac{\alpha}{\beta} \\ \Pr\left(X_1 < x\right), & x > \frac{2\alpha+1}{2\beta}, \end{cases}$$

$$M_n(x) := \max_{\boldsymbol{\lambda}\in\Theta} \Pr\left(\sum_{i=1}^{n} \lambda_i X_i < x\right) = \begin{cases} \Pr\left(X_1 < x\right), & x < \frac{\alpha}{\beta} \\ \Pr\left(\frac{1}{n}\sum_{i=1}^{n} X_i < x\right), & x > \frac{2\alpha+1}{2\beta} \end{cases}.$$

Results similar to Theorem 6.2 were obtained in [129] for the special case where the X_i's are chi-squared r.v's of degree 1 (corresponding to $\alpha = \beta = 1/2$). Theorem 6.2 extends those results to arbitrary gamma random variables, including chi-squared of arbitrary degree, exponential, Erlang, etc.

In what follows, for a gamma r.v X, we use the notation f_X for its PDF and F_X for its CDF.

The objective in the proof of Theorem 6.2 is to find the extrema (with respect to $\boldsymbol{\lambda} \in \Theta$) of the CDF of r.v $\sum_{i=1}^{n} \lambda_i X_i$. This is mainly achieved by perturbation arguments, employing a key identity which is derived using Laplace transforms. Using our perturbation arguments with this identity and employing Lemma 6.4, we obtain that at any extremum, we must have either $\lambda_1, \lambda_2 > 0$ and $\lambda_3 = \cdots = \lambda_n = 0$ or for some $i \leq n$ we must get $\lambda_1 = \cdots = \lambda_i > 0$ and $\lambda_{i+1} = \cdots = \lambda_n = 0$. (Note that this latter case covers the "corners" as well.). In the former case, Lemma 6.5 is used to distinguish between the minima and maxima for different values of x. These results along with Theorem 6.1 are then used to prove Theorem 6.2.

Three lemmas are used in the proofs of our two theorems. Lemma 6.3 describes some properties of the PDF of non-negative linear combinations of arbitrary gamma r.v's, such as analyticity and vanishing derivatives at zero. Lemma 6.4 describes the monotonicity property of the mode of the PDF of non-negative linear combinations of a *particular* set of gamma r.v's, which is useful for the proof of Theorem 6.2. Lemma 6.5 gives some properties regarding the mode of the PDF of convex combinations of two *particular* gamma r.v's, which is used in proving Theorem 6.1 and Theorem 6.2.

6.1 Lemmas

We next state and prove the lemmas summarized above.

Lemma 6.3 (Generalization of [129, Lemma A]). *Let $X_i \sim Gamma(\alpha_i, \beta_i)$, $i = 1, 2, \ldots, n$, be independent r.v's, where $\alpha_i, \beta_i > 0$ $\forall i$. Define $Y_n := \sum_{i=1}^{n} \lambda_i X_i$ for $\lambda_i > 0$, $\forall i$ and $\rho_j := \sum_{i=1}^{j} \alpha_i$. Then for the PDF of Y_n, f_{Y_n}, we have*

(i) $f_{Y_n} > 0$, $\forall x > 0$,

(ii) f_{Y_n} *is analytic on* $\mathbb{R}^+ = \{x | x > 0\}$,

(iii) $f_{Y_n}^{(k)}(0) = 0$, *if* $0 \leq k < \rho_n - 1$, *where* $f_{Y_n}^{(k)}$ *denotes the* k^{th} *derivative of* f_{Y_n}.

Proof. The proof is done by induction on n. For $n = 2$ we have

$$\begin{aligned} f_{Y_2}(x) &= \int_0^\infty f_{\lambda_1 X_1}(y) f_{\lambda_2 X_2}(x-y) dy \\ &= \int_0^x \frac{(\beta_1/\lambda_1)^{\alpha_1}}{\Gamma(\alpha_1)} y^{\alpha_1 - 1} e^{-\frac{\beta_1 y}{\lambda_1}} \frac{(\beta_2/\lambda_2)^{\alpha_2}}{\Gamma(\alpha_2)} (x-y)^{\alpha_2 - 1} e^{-\frac{\beta_2 (x-y)}{\lambda_2}} dy \\ &= \frac{(\beta_1/\lambda_1)^{\alpha_1} (\beta_2/\lambda_2)^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \int_0^x y^{\alpha_1 - 1} (x-y)^{\alpha_2 - 1} e^{-\frac{\beta_1 y}{\lambda_1} - \frac{\beta_2 (x-y)}{\lambda_2}} dy. \end{aligned}$$

Now the change of variable $y \to x \cos^2 \theta_1$ would yield

$$f_{Y_2}(x) = 2 \frac{(\beta_1/\lambda_1)^{\alpha_1} (\beta_2/\lambda_2)^{\alpha_2}}{\Gamma(\alpha_1)\Gamma(\alpha_2)} x^{(\alpha_1 + \alpha_2 - 1)} \int_0^{\frac{\pi}{2}} (\cos\theta_1)^{2\alpha_1 - 1} (\sin\theta_1)^{2\alpha_2 - 1} e^{-x(\frac{\beta_1 \cos^2\theta_1}{\lambda_1} + \frac{\beta_2 \sin^2\theta_1}{\lambda_2})} d\theta_1.$$

By induction on n, one can show that for arbitrary $n \geq 2$

$$f_{Y_n}(x) = 2^{n-1} \left(\prod_{i=1}^{n} \frac{(\beta_i/\lambda_i)^{\alpha_i}}{\Gamma(\alpha_i)} \right) x^{\rho_n - 1} \int_{D^{n-1}} P_n(\Theta_{n-1}) Q_n(\Theta_{n-1}) e^{-x R_n(\Theta_{n-1})} \mathbf{d}\Theta_{n-1}, \quad (6.2a)$$

where

$$P_n(\Theta_{n-1}) := \prod_{j=1}^{n-1} (\cos\theta_j)^{2\rho_j - 1}, \quad (6.2b)$$

$$Q_n(\Theta_{n-1}) := \prod_{j=1}^{n-1} (\sin\theta_j)^{2\alpha_{j+1} - 1}, \quad (6.2c)$$

the function $R_n(\Theta_{n-1})$ satisfies the following recurrence relation

$$R_n(\Theta_{n-1}) := \cos^2\theta_{n-1} R_{n-1}(\Theta_{n-2}) + \beta_n \lambda_n^{-1} \sin^2\theta_{n-1},\ \forall n \geq 2 \tag{6.2d}$$

$$R_1(\Theta_0) := \beta_1/\lambda_1, \tag{6.2e}$$

and $\mathbf{d}\Theta_{n-1}$ denotes the $n-1$ dimensional Lebesgue measure with the domain of integration

$$D^{n-1} := (0,\pi/2) \times (0,\pi/2) \times \ldots \times (0,\pi/2) = (0,\pi/2)^{n-1} \subset \mathbb{R}^{n-1}. \tag{6.2f}$$

Now the claims in Lemma 6.3 follow from (6.2). □

Lemma 6.4 (Generalization of [129, Lemma 1])**.** *Let $X_i \sim Gamma(\alpha_i, \alpha)$, $i = 1, 2, \ldots, n$, be independent r.v's, where $\alpha_i > 0\ \forall i$ and $\alpha > 0$. Also let $\psi \sim Gamma(1, \alpha)$ be another r.v independent of all X_i's. If $\sum_{i=1}^n \alpha_i > 1$, then the mode, $\bar{x}(\lambda)$, of the r.v $W(\lambda) = Y + \lambda\psi$ is strictly increasing in $\lambda > 0$, where $Y = \sum_{i=1}^n \lambda_i X_i$ with $\lambda_i > 0$, $\forall i$.*

Proof. The proof is almost identical to that of Lemma 1 in [129] and we give it here for completeness. By Lemma 6.3, $\bar{x}(\lambda) > 0$ for $\lambda \geq 0$. By the unimodality of $W(\lambda)$, for any $\lambda > \lambda_0 > 0$, it is enough to show that

$$J\big(\lambda, \bar{x}(\lambda_0)\big) := \left[\frac{d^2}{dx^2} \Pr\left(W(\lambda) \leq x\right)\right]_{x=\bar{x}(\lambda_0)} > 0. \tag{6.3}$$

Note that $J\big(\lambda_0, \bar{x}(\lambda_0)\big) = 0$ and since $\sum_{i=1}^n \alpha_i > 1$, by Lemma 6.3(iii), $f_Y(0) = 0$. So we have

$$\begin{aligned} J\big(\lambda, \bar{x}(\lambda_0)\big) &= \left[\frac{d}{dx}\int_0^x f_Y(x-z)\frac{\alpha}{\lambda}e^{-\frac{\alpha}{\lambda}z}dz\right]_{x=\bar{x}(\lambda_0)} \\ &= \left[\int_0^x \frac{d}{dx} f_Y\big(x-z\big)\frac{\alpha}{\lambda}e^{-\frac{\alpha}{\lambda}z}dz\right]_{x=\bar{x}(\lambda_0)} \\ &= \int_0^{\bar{x}(\lambda_0)} f_Y'(z)\frac{\alpha}{\lambda}e^{-\frac{\alpha}{\lambda}\left(\bar{x}(\lambda_0)-z\right)}dz. \end{aligned}$$

Therefore,

$$\int_0^{\bar{x}(\lambda_0)} f_Y'(z)e^{\frac{\alpha z}{\lambda_0}}dz = \frac{\lambda_0}{\alpha}e^{\frac{\alpha\bar{x}(\lambda_0)}{\lambda_0}} J\big(\lambda_0, \bar{x}(\lambda_0)\big) = 0.$$

Thus for $\lambda > \lambda_0 > 0$, we have

$$\begin{aligned}
\frac{\lambda}{\alpha} e^{\frac{\alpha \bar{x}(\lambda)}{\lambda}} J\big(\lambda, \bar{x}(\lambda_0)\big) &= \int_0^{\bar{x}(\lambda_0)} f_Y^{'}(z) e^{\frac{\alpha z}{\lambda}} dz \\
&= \int_0^{\bar{x}(\lambda_0)} f_Y^{'}(z) e^{\frac{\alpha z}{\lambda}} - f_Y^{'}(z) e^{\frac{\alpha z}{\lambda_0}} e^{\alpha \bar{x}(0)\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)} dz \\
&= \int_0^{\bar{x}(\lambda_0)} f_Y^{'}(z) \left(e^{\frac{\alpha z}{\lambda}} - e^{\frac{\alpha z}{\lambda_0} + \alpha \bar{x}(0)\left(\frac{1}{\lambda} - \frac{1}{\lambda_0}\right)} \right) dz \\
&= \int_0^{\bar{x}(\lambda_0)} f_Y^{'}(z) \left(e^{\frac{\alpha z}{\lambda}} - e^{\frac{\alpha z}{\lambda} + \Phi\left(z, \bar{x}(0)\right)} \right) dz,
\end{aligned}$$

where $\bar{x}(0) > 0$ is the mode of r.v Y and

$$\Phi\big(z, \bar{x}(0)\big) := \alpha\Big(z - \bar{x}(0)\Big)\Big(\frac{1}{\lambda_0} - \frac{1}{\lambda}\Big).$$

Now if $z < \bar{x}(0)$ then $\Phi\big(z, \bar{x}(0)\big) < 0$ and $f_Y^{'}(z) > 0$ so we get $J\big(\lambda, \bar{x}(\lambda_0)\big) > 0$. Similarly if $z > \bar{x}(0)$ then $\Phi\big(z, \bar{x}(0)\big) > 0$ and $f_Y^{'}(z) < 0$ and again we have $J\big(\lambda, \bar{x}(\lambda_0)\big) > 0$. □

Lemma 6.5 (Generalization of [129, Lemma 2])**.** *For some $\alpha_2 \geq \alpha_1 > 0$, let $\xi_1 \sim Gamma(1 + \alpha_1, \alpha_1)$ and $\xi_2 \sim Gamma(1 + \alpha_2, \alpha_2)$ be independent gamma r.v's. Also let $\bar{x} = \bar{x}(\lambda)$ denote the mode of the r.v $\xi(\lambda) = \lambda\xi_1 + (1 - \lambda)\xi_2$ for $0 \leq \lambda \leq 1$. Then*

(i) for a given λ, $\bar{x}(\lambda)$ is unique,

(ii) $1 \leq \bar{x}(\lambda) \leq \frac{2\sqrt{\alpha_1\alpha_2}+1}{2\sqrt{\alpha_1\alpha_2}}, \quad \forall 0 \leq \lambda \leq 1$, with $\bar{x}(0) = \bar{x}(1) = 1$ and, in case of $\alpha_i = \alpha_j = \alpha$, $\bar{x}(\frac{1}{2}) = \frac{2\alpha+1}{2\alpha}$, otherwise the inequalities are strict, and

(iii) there is a $\lambda^* \in \big(\frac{\sqrt{\alpha_1}}{\sqrt{\alpha_2}+\sqrt{\alpha_1}}, 1\big)$ such that the mode $\bar{x}(\lambda)$ is a strictly increasing function of λ on $(0, \lambda^*)$ and it is a strictly decreasing function on $(\lambda^*, 1)$ and, for $\alpha_1 = \alpha_2$, we have $\lambda^* = \frac{1}{2}$.

Proof. Uniqueness claim (i) has already been proven in [129, Theorem 4]. We prove (iii) since (ii) is implied from within the proof. For $0 < \lambda < 1$, the PDF of $\xi(\lambda)$ can be written as

$$f_{\xi(\lambda)}(x) = \int_0^x f_{\lambda\xi_1}(y) f_{(1-\lambda)\xi_2}(x - y) dy.$$

Since $f_{\lambda\xi_1}(0) = f_{(1-\lambda)\xi_2}(0) = 0$ we have

$$\begin{aligned}\frac{\partial}{\partial x} f_{\xi(\lambda)}(x) &= \int_0^x f_{\lambda\xi_1}(y) \frac{\partial}{\partial x} f_{(1-\lambda)\xi_2}(x-y)dy \\ &= -\int_0^x f_{\lambda\xi_1}(y) \frac{\partial}{\partial y} f_{(1-\lambda)\xi_2}(x-y)dy \\ &= \int_0^x \frac{\partial}{\partial y} \left(f_{\lambda\xi_1}(y)\right) f_{(1-\lambda)\xi_2}(x-y)dy\end{aligned}$$

where for the second equality we use the fact that $\frac{\partial}{\partial x} f(x-y) = -\frac{\partial}{\partial y} f(x-y)$, and for the third equality we used integration by parts. Let $\alpha = \alpha_1$ and $\alpha_2 = c\alpha$ for some $c \geq 1$. So now we have

$$\begin{aligned}&\frac{\partial}{\partial x} f_{\xi(\lambda)}(x) = \frac{(\frac{\alpha}{\lambda})^{1+\alpha}(\frac{c\alpha}{1-\lambda})^{1+\alpha c}}{\Gamma(1+\alpha)\Gamma(1+\alpha c)} \int_0^x \frac{\partial\left(y^\alpha e^{-\frac{\alpha y}{\lambda}}\right)}{\partial y}(x-y)^{\alpha c} e^{-\frac{c\alpha(x-y)}{1-\lambda}} dy \\ = \;& \frac{\alpha^{2+\alpha}(c\alpha)^{1+c\alpha}}{\Gamma(1+\alpha)\Gamma(1+c\alpha)} \lambda^{-2-\alpha} (1-\lambda)^{-1-\alpha c} e^{-\frac{c\alpha x}{(1-\lambda)}} \int_0^x (\lambda - y) y^{\alpha-1} (x-y)^{\alpha c} e^{-\alpha y\left(\frac{1}{\lambda} - \frac{c}{1-\lambda}\right)} dy \\ = \;& C(x,\lambda) A(x,\lambda),\end{aligned}$$

where

$$\begin{aligned}C(x,\lambda) &:= \frac{\alpha^{2+\alpha}(c\alpha)^{1+c\alpha}}{\Gamma(1+\alpha)\Gamma(1+c\alpha)} \lambda^{-2-\alpha} (1-\lambda)^{-1-\alpha c} e^{-\frac{c\alpha x}{(1-\lambda)}}, \\ A(x,\lambda) &:= \int_0^x (\lambda - y) y^{\alpha-1} (x-y)^{\alpha c} e^{-\phi(\lambda) y} dy, \\ \phi(\lambda) &:= \alpha\left(\frac{1}{\lambda} - \frac{c}{1-\lambda}\right).\end{aligned}$$

Now if $\bar{x}$ is the mode of $\xi(\lambda)$, then we have

$$\frac{\partial}{\partial x} f_{\xi(\lambda)}(\bar{x}) = C(\bar{x},\lambda) A(\bar{x},\lambda) = 0,$$

which implies that $A(\bar{x},\lambda) = 0$ since $C(\bar{x},\lambda) > 0$. Let us define the linear functional $L : \mathcal{G} \to \mathbb{R}$, where $\mathcal{G} = \{g : (0,\bar{x}) \to \mathbb{R} \mid \int_0^{\bar{x}} g(y) y^{\alpha-1} < \infty\}$, as

$$L(g) := \int_0^{\bar{x}} g(y) y^{\alpha-1} (\bar{x}-y)^{\alpha c} e^{-\phi(\lambda) y} dy.$$

We have

$$\begin{aligned}\frac{\partial}{\partial\lambda}A(x,\lambda) &= \int_0^x \left[1-\phi^{'}(\lambda)y(\lambda-y)\right] y^{\alpha-1}(x-y)^{\alpha c}e^{-\phi(\lambda)y}dy \\ &= \int_0^x \left[1-\lambda\phi^{'}(\lambda)y+\phi^{'}(\lambda)y^2\right] y^{\alpha-1}(x-y)^{\alpha c}e^{-\phi(\lambda)y}dy,\end{aligned}$$

so

$$\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} = L\left(1-\lambda\phi^{'}(\lambda)f+\phi^{'}(\lambda)f^2\right), \tag{6.4}$$

where $f\in\mathcal{G}$ is such that $f(y)=y$. On the other hand since $A(\bar{x},\lambda)=0$, we get

$$\begin{aligned}L(\lambda)=L(f) &= \int_0^{\bar{x}} y^{\alpha}(\bar{x}-y)^{\alpha c}e^{-\phi(\lambda)y}dy \\ &= \int_0^{\bar{x}} y^{\alpha}e^{-\phi(\lambda)y}d\left(-\frac{(\bar{x}-y)^{\alpha c+1}}{\alpha c+1}\right) \\ &= (\alpha c+1)^{-1}\int_0^{\bar{x}}(\bar{x}-y)^{\alpha c+1}\,d\left(y^{\alpha}e^{-\phi(\lambda)y}\right) \\ &= (\alpha c+1)^{-1}\int_0^{\bar{x}}(\bar{x}-y)^{\alpha c+1}(\alpha y^{\alpha-1}e^{-\phi(\lambda)y}-\phi(\lambda)y^{\alpha}e^{-\phi(\lambda)y})dy \\ &= (\alpha c+1)^{-1}\int_0^{\bar{x}}(\bar{x}-y)\,(\alpha-\phi(\lambda)y)\,y^{\alpha-1}\,(\bar{x}-y)^{\alpha c}\,e^{-\phi(\lambda)y}dy \\ &= (\alpha c+1)^{-1}L\Big(\,(\bar{x}-f)\,(\alpha-\phi(\lambda)f)\,\Big) \\ &= (\alpha c+1)^{-1}L\Big(\alpha\bar{x}-\alpha f-\phi(\lambda)\bar{x}f+\phi(\lambda)f^2\Big),\end{aligned}$$

where the second integral is Lebesgue-Stieltjes, and the third integral follows from Lebesgue-Stieltjes integration by parts. So, for $\lambda\in(0,\frac{1}{c+1})\cup(\frac{1}{c+1},1)$, we get

$$\begin{aligned}L(f^2) &= \frac{1}{\phi(\lambda)}\left[(\alpha c+1)L(f)-L\Big(\alpha\bar{x}-\alpha f-\phi(\lambda)\bar{x}f\Big)\right] \\ &= \frac{1}{\phi(\lambda)}\left[L\left((\alpha c+1)f-\frac{\alpha\bar{x}}{\lambda}f+\alpha f+\phi(\lambda)\bar{x}f\right)\right] \\ &= \frac{1}{\phi(\lambda)}\left[\left((\alpha c+1)+\alpha+\phi(\lambda)\bar{x}-\frac{\alpha\bar{x}}{\lambda}\right)L(f)\right] \\ &= \frac{1}{\phi(\lambda)}\left[\left((\alpha+\alpha c+1)+(\phi(\lambda)-\frac{\alpha}{\lambda})\bar{x}\right)L(f)\right] \\ &= \frac{1}{\phi(\lambda)}\left[\Big((1+c)\alpha+1-\frac{c\alpha\bar{x}}{1-\lambda}\Big)L(f)\right],\end{aligned}$$

where we used the fact that $L(\alpha\bar{x}) = \frac{\alpha\bar{x}}{\lambda}L(\lambda) = \frac{\alpha\bar{x}}{\lambda}L(f)$. Now substituting $L(f^2)$ in (6.4) yields

$$\begin{aligned}
\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} &= L\Big(\frac{1}{\lambda}f - \lambda\phi^{'}(\lambda)f + \phi^{'}(\lambda)f^2\Big) \\
&= \left(\frac{1}{\lambda} - \lambda\phi^{'}(\lambda) + \frac{\phi^{'}(\lambda)}{\phi(\lambda)}\Big[(1+c)\alpha + 1 - \frac{c\alpha\bar{x}}{1-\lambda}\Big]\right)L(f), \\
&= \left(\frac{1}{\lambda} - \phi^{'}(\lambda)\left(\lambda + \frac{1}{\phi(\lambda)}\left[(1+c)\alpha + 1 - \frac{c\alpha\bar{x}}{1-\lambda}\right]\right)\right)L(f) \\
&= \left(\frac{1}{\lambda} - \frac{\phi^{'}(\lambda)}{\phi(\lambda)}\left(\lambda\phi(\lambda) + (1+c)\alpha + 1 - \frac{c\alpha\bar{x}}{1-\lambda}\right)\right)L(f)
\end{aligned}$$

which after some tedious but routine computations gives

$$\left[\frac{\partial}{\partial\lambda}A(x,\lambda)\right]_{x=\bar{x}} = R(\lambda)\frac{\bar{x} - \Phi(\lambda)}{1-(c+1)\lambda}, \quad \lambda \in \Big(0, \frac{1}{1+c}\Big) \cup \Big(\frac{1}{1+c}, 1\Big)$$

where $R(\lambda) > 0$, for all $0 < \lambda < 1$, and

$$\Phi(\lambda) := \frac{\alpha + (1-2\alpha)\lambda + (\alpha - 1 + \alpha c)\lambda^2}{\alpha\Big((c+1)\lambda^2 - 2\lambda + 1\Big)}.$$

Since

$$\frac{d\Phi(\lambda)}{d\lambda} = \Big((1-c)\lambda^2 - 2\lambda + 1\Big)\Big/\Big(\alpha\big((c+1)\lambda^2 - 2\lambda + 1\big)\Big)^2,$$

we have that

$$\frac{d\Phi(\lambda)}{d\lambda} = 0 \ \text{ at } \lambda = \frac{1}{(1+\sqrt{c})}.$$

Note that the other root, $1/(1-\sqrt{c})$, falls outside of $(0,1)$ for any $c \geq 1$. It readily can be seen that $\Phi(\lambda)$ is increasing on $0 < \lambda < \frac{1}{1+\sqrt{c}}$ and decreasing on $\frac{1}{1+\sqrt{c}} < \lambda < 1$, and so

$$1 \leq \Phi(\lambda) \leq \frac{2\alpha\sqrt{c}+1}{2\alpha\sqrt{c}}, \quad \forall 0 \leq \lambda \leq 1.$$

The differentiability of $\bar{x}(\lambda)$ with respect to λ follows from implicit function theorem:

$$\frac{d\bar{x}(\lambda)}{d\lambda} = -\frac{\frac{\partial}{\partial\lambda}A(\bar{x},\lambda)}{\frac{\partial}{\partial\bar{x}}A(\bar{x},\lambda)},$$

and for that we need to show that $\frac{\partial A(\bar{x},\lambda)}{\partial \bar{x}} \neq 0$ for all $0 < \lambda < 1$. If we assume the contrary for some λ, we get

$$\begin{aligned} \alpha c A(\bar{x},\lambda) &= \alpha c \int_0^{\bar{x}} (\lambda - y) y^{\alpha-1} (\bar{x} - y)^{\alpha c} e^{-\phi(\lambda) y} dy = 0, \\ (\bar{x} - \lambda) \frac{\partial}{\partial \bar{x}} A(\bar{x},\lambda) &= \alpha c \int_0^{\bar{x}} (\lambda - y)(\bar{x} - \lambda) y^{\alpha-1} (\bar{x} - y)^{\alpha c - 1} e^{-\phi(\lambda) y} dy = 0, \end{aligned}$$

which is impossible since the integrand in the first equality is strictly larger than the one in the second equality: we can see this by looking at the two cases $0 < y < \lambda$ and $\lambda < y < \bar{x}$. From this we can also note that $\frac{\partial}{\partial \bar{x}} A(\bar{x},\lambda) < 0$ for all $0 < \lambda < 1$. To see this, first consider the case $\bar{x} > \lambda$, and it follows directly as above that $\frac{\partial}{\partial \bar{x}} A(\bar{x},\lambda) < [\alpha c/(\bar{x} - \lambda)] A(\bar{x},\lambda) = 0$. Now assume that $\bar{x} \leq \lambda$, but since the integrand in the first equality is strictly positive for all $0 < y < \bar{x}$, then $A(\bar{x},\lambda) > 0$ which is impossible. So we get

$$\frac{d\bar{x}(\lambda)}{d\lambda} = S(\lambda) \frac{\bar{x} - \Phi(\lambda)}{1 - (c+1)\lambda}, \quad \lambda \in [0,1] \tag{6.5}$$

where $S(\lambda) > 0$ for all $0 < \lambda < 1$. We also defined $\frac{d\bar{x}(\lambda)}{d\lambda}$ for $\lambda = 0, 1, \frac{1}{2}$ using l'Hôpital's rule (with one-sided differentiability for $\lambda = 0, 1$). It is easy to see that

$$\begin{aligned} &\bar{x}(0) = \bar{x}(1) = \Phi(0) = \Phi(1) = 1, \\ &\bar{x}\Big(\frac{1}{c+1}\Big) = \Phi\Big(\frac{1}{c+1}\Big) = \frac{(c+1)\alpha + 1}{(c+1)\alpha}. \end{aligned}$$

Next we show that $\bar{x}$ is strictly increasing on $(0, \frac{1}{c+1})$. We first show that on this interval, we must have $\bar{x}(\lambda) \geq \Phi(\lambda)$, otherwise there must exist a $\hat{\lambda} \in (0, \frac{1}{c+1})$ such that $\bar{x}(\hat{\lambda}) < \Phi(\hat{\lambda})$. But this contradicts $\bar{x}(\frac{1}{c+1}) = \Phi(\frac{1}{c+1})$ by (6.5), increasing property of Φ and continuity of $\bar{x}$. So $\bar{x}$ is non-decreasing on $(0, \frac{1}{c+1})$. We must also have that $\bar{x}(\lambda) > \Phi(\lambda)$ for $\lambda \in (0, \frac{1}{c+1})$, otherwise if there is a $\hat{\lambda} \in (0, \frac{1}{c+1})$ such that $\bar{x}(\hat{\lambda}) = \Phi(\hat{\lambda})$, then, by (6.5), it must be a saddle point of $\bar{x}$. But since Φ is strictly increasing and $\bar{x}$ is non-decreasing on this interval, this would imply that for an ε arbitrarily small, we must have $\bar{x}(\hat{\lambda} + \varepsilon) < \Phi(\hat{\lambda} + \varepsilon)$ but this would contradict the non-decreasing property of $\bar{x}$ on this interval by (6.5). The same reasoning shows that we must have $\bar{x}(\lambda) < \Phi(\lambda)$ on $(\frac{1}{c+1}, \lambda^*)$ (i.e. $\bar{x}$ is strictly increasing on $(\frac{1}{c+1}, \lambda^*)$) and $\bar{x}(\lambda) > \Phi(\lambda)$

on $(\lambda^*, 1)$ (i.e. $\bar{x}$ is strictly decreasing on $(\lambda^*, 1)$). Now we show that $\lambda^* \geq \frac{1}{1+\sqrt{c}}$. For $c = 1$ we have $\frac{1}{c+1} = \frac{1}{\sqrt{c}+1}$, hence $\lambda^* = \frac{1}{2}$. For $c > 1$, Since $\bar{x}(\lambda)$ is increasing for $0 < \lambda < \lambda^*$, decreasing for $\lambda^* < \lambda < 1$, and $\bar{x}(\lambda^*) = \Phi(\lambda^*)$, then by (6.5), this implies that λ^* is where the maximum of $\bar{x}(\lambda)$ occurs. Now if we assume that $\lambda^* < \frac{1}{1+\sqrt{c}}$, since Φ is increasing on $(0, \frac{1}{1+\sqrt{c}})$, this would contradict $\bar{x}(\lambda) > \Phi(\lambda)$ on $(\lambda^*, 1)$. Lemma 6.5 is proved. □

6.2 Proofs of Theorems 6.1 and 6.2

We now give the detailed proofs for our main theorems stated at the beginning of this chapter and used in Chapter 7.

Proof of Theorem 6.1

For proving (i), we first show that $\Delta(x) = 0$ at exactly one point on $\mathbb{R}^+ = \{x | x > 0\}$ denoted by $x(\alpha_1, \alpha_2)$. Since $\alpha_2 > \alpha_1$, let $\alpha_2 = \alpha_1 + c$, for some $c > 0$. We have

$$\begin{aligned} \frac{d\Delta(x)}{dx} &= C(\alpha_2)x^{\alpha_2-1}e^{-\alpha_2 x} - C(\alpha_1)x^{\alpha_1-1}e^{-\alpha_1 x} \\ &= C(\alpha_2)x^{\alpha_1-1}e^{-\alpha_1 x}\left(x^c e^{-cx} - \frac{C(\alpha_1)}{C(\alpha_2)}\right) \end{aligned}$$

where $C(\alpha) = (\alpha)^\alpha/\Gamma(\alpha)$. The constant $C(\alpha_1)/C(\alpha_2)$ cannot be larger than $x^c e^{-cx}$, for all $x \in \mathbb{R}^+$, otherwise $d\Delta(x)/dx$ would be negative for all $x \in \mathbb{R}^+$, and this is impossible since $\Delta(0) = \Delta(\infty) = 0$. The function $x^c e^{-cx}$ is increasing on $(0, 1)$ and decreasing on $(1, \infty)$, and since $C(\alpha_1)/C(\alpha_2)$ is constant, there must exist an interval (a, b) containing $x = 1$ such that $d\Delta(x)/dx > 0$ for $x \in (a, b)$ and $d\Delta(x)/dx < 0$ for $x \in (0, a) \cup (b, \infty)$. Now since $\Delta(x)$ is continuous and $\Delta(0) = \Delta(\infty) = 0$, then there must exist a unique $x(\alpha_1, \alpha_2) \in (0, \infty)$ such that $\Delta(x)$ crosses zero (i.e., $\Delta(x) = 0$ at the unique point $x(\alpha_1, \alpha_2)$) and that $\Delta(x) < 0$ for $0 < x < x(\alpha_1, \alpha_2)$ and $\Delta(x) > 0$ for $x > x(\alpha_1, \alpha_2)$.

We now prove (ii). The desired inequality is equivalent to

$$\Delta(x) < 0, \quad \forall x < 1$$

and

$$\Delta(x) > 0, \quad \forall x > \big(2\sqrt{\alpha_1(\alpha_2-\alpha_1)}+1\big)/\big(2\sqrt{\alpha_1(\alpha_2-\alpha_1)}\big).$$

Without loss of generality consider $\alpha = \alpha_1$, and $\alpha_2 = (1+c)\alpha$, for $c = (\alpha_2 - \alpha)/\alpha$. Define $\tilde{X} \sim Gamma(c\alpha, c\alpha)$ and let

$$Y(t) := tX_1 + (1-t)\tilde{X}.$$

Note that

$$Y(1) = X_1$$

and

$$Y(\frac{1}{1+c}) = X_2,$$

so it suffices to show that the CDF of $Y(t)$ is increasing in $t \in [\frac{1}{1+c}, 1]$ for $x < 1$ and decreasing for $x > (2\alpha\sqrt{c}+1)/(2\alpha\sqrt{c})$. Now, we take the Laplace transform of $Y(t)$ as

$$\mathcal{L}[Y(t)](z) = \big(1+\frac{tz}{\alpha}\big)^{-\alpha}\big(1+\frac{(1-t)z}{c\alpha}\big)^{-c\alpha}, \quad Re(z) > \max\left\{-\alpha/t, -c\alpha/(1-t)\right\}.$$

The Laplace transform of F_Y is

$$\begin{aligned}\mathcal{L}[F_Y](z) &= \int_0^\infty e^{-zx}F_Y(x)dx \\ &= \frac{1}{z}\int_0^\infty e^{-zx}dF_Y(x) \\ &= \frac{1}{z}\mathcal{L}[Y](z).\end{aligned}$$

Note that in the second equality we applied integration by parts and the fact that $F_Y(0) = 0$. Defining

$$J(z) := \mathcal{L}[F_Y](z)$$

and differentiating with respect to t gives

$$\begin{aligned}\frac{dJ}{dt} &= J\frac{d}{dt}\left(\ln(J)\right)\\ &= J\frac{d}{dt}\left(-\ln(z)-\alpha\ln(1+\frac{tz}{\alpha})-c\alpha\ln\left(1+\frac{(1-t)z}{c\alpha}\right)\right)\\ &= \frac{z^2}{c\alpha}J\Big((1+c)t-1\Big)\left(1+\frac{tz}{\alpha}\right)^{-1}\left(1+\frac{(1-t)z}{c\alpha}\right)^{-1}.\end{aligned}$$

Taking the inverse transform yields

$$\frac{d}{dt}\Pr\left(Y(t)\le x\right)=\frac{(1+c)t-1}{c\alpha}\frac{d^2}{dx^2}\Pr\left(Y(t)+t\psi_1+\frac{1-t}{c}\psi_2<x\right),$$

where $\psi_i\sim Gamma(1,\alpha)$, $i=1,2$, are i.i.d gamma r.v's which are also independent of all X_1 and X_2. Now applying Lemma 6.5 yields the desired results. □

Proof of Theorem 6.2 It is enough to prove the theorem for the special case where $\alpha=\beta$ and the general statement follows from the scaling properties of gamma r.v.

Introduce the random variable

$$Y:=\sum_{i=1}^{n}\lambda_i X_i$$

with CDF $F_Y(x)=\Pr(Y<x)$. As in the proof of Theorem 6.1, define

$$J(z):=\mathcal{L}[F_Y](z)=\frac{1}{z}\mathcal{L}[Y](z),$$

where $\mathcal{L}[F_Y]$ and $\mathcal{L}[Y]$ denote the Laplace transform of F_Y and Y, respectively and

$$\mathcal{L}[Y](z)=\prod_{i=1}^{n}\left(1+\lambda_i z/\alpha\right)^{-\alpha},\quad Re(z)>-\alpha/\lambda_i,\ i=1,2,\ldots,n.$$

Now consider a vector $\boldsymbol{\lambda}\in\Theta$ for which $\lambda_i\lambda_j\neq 0$ for some $i\neq j$. We keep all λ_k, $k\neq i,j$ fixed and vary λ_i and λ_j under the condition that $\lambda_i+\lambda_j=const$. We may assume without loss of generality that $i=1$ and $j=2$. Vectors for which $\lambda_i=1$ for some i, i.e. the "corners"

of Θ, are considered at the end of this proof. Differentiating J, we get

$$\begin{aligned}\frac{dJ}{d\lambda_1} = J\frac{d}{d\lambda_1}(\ln J) &= J\frac{d}{d\lambda_1}\Big(-\ln(z) - \alpha\sum_{i=1}^{n}\ln(1+\frac{\lambda_i z}{\alpha})\Big) \\ &= J\alpha\frac{z^2}{\alpha^2}\frac{\lambda_1-\lambda_2}{(1+\frac{\lambda_1 z}{\alpha})(1+\frac{\lambda_2 z}{\alpha})} \\ &= \frac{1}{\alpha}(\lambda_1-\lambda_2)z\mathcal{L}[\lambda_1\psi_1](z)\mathcal{L}[\lambda_2\psi_2](z)\mathcal{L}[Y](z) \qquad (6.6)\end{aligned}$$

where $\psi_i \sim Gamma(1,\alpha)$, $i = 1, 2$ are i.i.d gamma r.v's which are also independent of all X_i's.

Letting

$$W(\lambda) := Y + \lambda_1\psi_1 + \lambda\psi_2$$

with the CDF $F_{W(\lambda)}(x)$, it can be shown that since $\lambda_1\lambda_2 \neq 0$, then by Lemma 6.3(iii), $F_{W(\lambda)}(0) = F'_{W(\lambda)}(0) = 0,\ \forall\lambda \geq 0$. Defining

$$L(Y,\lambda,x) := F''_{W(\lambda)} = \frac{d^2}{dx^2}\Pr\left(W(\lambda) < x\right) = \frac{d^2}{dx^2}\Pr\Big(Y + \lambda_1\psi_1 + \lambda\psi_2 < x\Big) \qquad (6.7)$$

and noting that

$$\mathcal{L}[W(\lambda)](z) = \mathcal{L}[\lambda_1\psi_1](z)\mathcal{L}[\lambda\psi_2](z)\mathcal{L}[Y](z),$$

we get

$$\begin{aligned}\mathcal{L}\big[L(Y,\lambda,.)\big](z) &= \int_0^\infty e^{-zx}L(Y,\lambda,x)dx \\ &= \int_0^\infty e^{-zx}F''_{W(\lambda)}(x)dx \\ &= z\int_0^\infty e^{-zx}F'_{W(\lambda)}(x)dx \\ &= z^2\int_0^\infty e^{-zx}F_{W(\lambda)}(x)dx \\ &= z\int_0^\infty e^{-zx}dF_{W(\lambda)}(x) \\ &= z\mathcal{L}\big[W(\lambda)\big](z) \\ &= z\mathcal{L}\big[\lambda_1\psi_1\big](z)\mathcal{L}\big[\lambda\psi_2\big](z)\mathcal{L}\big[Y\big](z).\end{aligned}$$

Inverting (6.6) yields

$$\frac{dF_Y(x)}{d\lambda_1} = \frac{1}{\alpha}(\lambda_1 - \lambda_2)L(Y, \lambda_2, x). \tag{6.8}$$

So a necessary condition for the extremum of $F_Y(x)$ is either $\lambda_1\lambda_2(\lambda_1-\lambda_2) = 0$ or $L(\lambda_2, x) = 0$. Since $\lambda_1\lambda_2 \neq 0$ then by Lemma 6.3, the PDF, $f_{W(\lambda)}(x)$, of the linear form $W(\lambda) = Y + \lambda_1\psi_1 + \lambda\psi_2$, for $\lambda > 0$, is differentiable everywhere and $f_{W(\lambda)}(0) = 0$. In addition, on the positive half-line, $f'_{W(\lambda)}(x) = 0$ holds at a unique point because $f_{W(\lambda)}(x)$ is a unimodal analytic function (its graph contains no line segment). The unimodality of $f_{W(\lambda)}(x)$ was already proven for all gamma random variables in [129, Theorem 4].

Now we can prove that, for any $x > 0$, if $F_Y(x)$ has an extremum then the nonzero λ_i's can take at most two different values. Suppose that $\lambda_1\lambda_2(\lambda_1 - \lambda_2) \neq 0$, then by (6.8) we have $L(Y, \lambda_2, x) = 0$. Now we show that, for every $\lambda_j \neq 0$, (6.8) implies that $\lambda_i = \lambda_1$ or $\lambda_i = \lambda_2$. For this, we assume the contrary that $\lambda_i \neq \lambda_1$, $\lambda_i \neq \lambda_2$, and by using the same reasoning that led to (6.8), we can show that

$$L(Y, \lambda_2, x) = L(Y, \lambda_j, x) = 0$$

for every $\lambda_j \neq 0$, i.e. the point $x > 0$ is simultaneously the mode of the PDF of $W_Y^{\lambda_2}$ and $W_Y^{\lambda_j}$ which contradicts Lemma 6.4. So we get that $\lambda_i = \lambda_1$ or $\lambda_2 = \lambda_j$. Thus the extrema of $F_Y(x)$ are taken for some $\lambda_1 = \lambda_2 = \ldots = \lambda_k$, $\lambda_{k+1} = \lambda_{k+2} = \ldots = \lambda_{k+m}$, and $\lambda_{k+m+1} = \lambda_{k+m+2} = \ldots = \lambda_n = 0$ where $k + m \leq n$, i.e.,

$$\text{extremum } \Pr\big(\sum_{i=1}^{n} \lambda_i X_i \leq x\big) = \text{extremum } \Pr\big(\frac{\lambda}{k}\sum_{i=1}^{k} X_i + \frac{1-\lambda}{m}\sum_{i=k+1}^{k+m} X_i \leq x\big).$$

Here without loss of generality we can assume $k \geq m \geq 1$. Now the same reasoning as in the end of the proof of [129, Theorem 1] shows an extremum is taken either at $k = m = 1$, or at $\lambda_1 = \lambda_2 = \ldots = ... = \lambda_{k+m}$. In the former case, by Lemma 6.5, for any $x \in (0, 1) \cup (\frac{2\alpha+1}{2\alpha}, \infty)$, the extremum can only be taken at $\lambda \in \{0, \frac{1}{2}, 1\}$. However, for any $x \in [1, \frac{2\alpha+1}{2\alpha}]$, in addition to $\lambda \in \{0, \frac{1}{2}, 1\}$, the extremum can be achieved for some λ^* such that $x = \bar{x}(\lambda^*)$ where $\bar{x}(\lambda)$ denotes the mode of the distribution of $\lambda X_1 + (1-\lambda)X_2 + \lambda\psi_1 + (1-\lambda)\psi_2$. But for such λ^* and x, using (6.8) and Lemma 6.5(iii) with $\alpha_1 = \alpha_2 = \alpha$, one can show that $\Pr(\lambda X_1 + (1-\lambda)X_2 \leq x)$ achieves a local maximum. Now including the case where $\lambda_1 = 1$ mentioned earlier in the proof,

we get

$$\begin{aligned} m_n(x) &= \min_{1\le d\le n} \Pr\left(\frac{1}{d}\sum_{i=1}^{d} X_i < x\right) \quad \forall x > 0, \\ M_n(x) &= \max_{1\le d\le n} \Pr\left(\frac{1}{d}\sum_{i=1}^{d} X_i < x\right) \quad \forall x \in \Big(0,1\Big) \cup \Big(\frac{2\alpha+1}{2\alpha}, \infty\Big), \end{aligned}$$

where $m_n(x)$ and $M_n(x)$ are defined in the statement of Theorem 6.2 in Section 7.1. Now applying Theorem 6.1 by considering the collection $\alpha_i = i\alpha,\ i = 1, 2, \ldots, n$, would yield the desired results. □

Chapter 7

Uncertainty Quantification of Stochastic Reconstruction Algorithms

In the present chapter, we continue to consider the stochastic algorithms, presented in Chapters 3 and 4, for efficiently solving the class of large scale non-linear least squares (NLS) problems described in Chapter 1. We will continue to make Assumptions (A.1) - (A.3) (Assumption (A.2) can be, if necessary, restored by employing similar techniques as in Chapter 4). In Chapters 3 and 4, practical and randomized reconstruction algorithms were discussed and their efficiency was demonstrated by various numerical examples. However, all randomized steps in these algorithms were left to heuristics and as such the amount of uncertainty in each stochastic step remained unchecked. One advantage of leaving these steps heuristic is the great simplicity in the design and high efficiency in the performance of such algorithms. However, the mere existence of uncertainty in the overall procedure can cast doubt on the credibility of the reconstructions. In many applications, one might be willing to sacrifice the simplicity and even compromise slightly on the efficiency in order to have a handle on the amount of uncertainty in the algorithm. Hence, it may be desirable to have means which allow one to adjust the cost and accuracy of such algorithms in a quantifiable way, and find a balance that is suitable to particular objectives and computational resources. Here, we propose eight variants of Algorithm 2 where the uncertainties in the major stochastic steps are quantified (adjustment of Algorithms 1 and 3 in a similar way is straightforward). Quantifying the uncertainty in these stochastic steps, again, involves approximating the NLS objective function using Monte-Carlo (MC) methods as discussed in Section 2.1. There, it was shown that such approximation is,

in fact, equivalent to estimating the trace of the corresponding SPSD matrices. In Chapter 5, these estimators were analyzed and conditions on the MC sample size (which translates to cost) to satisfy the prescribed probabilistic relative accuracy were given. However, these conditions, though asymptotically tight, are pessimistic and are typically not sufficiently tight to be practically useful. On the other hand, as discussed in Chapter 3, the objective is to be able to generate as few random samples as possible for achieving acceptable approximations to the objective function. Hence, in the present chapter, and for the case of the Gaussian estimator, we prove *tight necessary* and *sufficient* conditions on the MC sample size and we show that these conditions are practically computable and yield small sample sizes. They are then incorporated in our stochastic algorithm to quantify the uncertainty in each randomized step. The bounds we use are applications of the main results of Chapter 6 presented in Theorems 6.1 and 6.2.

This chapter is organized as follows. In Section 7.1, we develop and state theorems regarding the tight tail bounds promised above. In Section 7.2 we present our stochastic algorithm variants for approximately minimizing (1.6) or (1.7) and discuss their novel elements. Subsequently in Section 7.3, the efficiency of the proposed algorithm variants is numerically demonstrated. This is followed by conclusions and further thoughts in Section 7.4.

7.1 Tight Conditions on Sample Size for Gaussian MC Trace Estimators

Let the matrix $A = B^T B \in \mathbb{R}^{s\times s}$ be implicit SPSD, and denote its trace by $tr(A)$. As described in Chapter 5, the Gaussian Monte-Carlo estimator of $tr(A)$ is defined by (cf. (5.1) with $\mathcal{D} = G$)

$$tr_G^n(A) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_i^T A \mathbf{w}_i, \tag{7.1}$$

where $\mathbf{w}_j \in \mathbb{R}^s \sim \mathcal{N}(0, \mathbb{I})$.

Now, given a pair of small positive real numbers (ε, δ), consider finding an appropriate

sample size n such that

$$\Pr\left(tr_G^n(A) \geq (1-\varepsilon)tr(A)\right) \geq 1-\delta, \tag{7.2a}$$

$$\Pr\left(tr_G^n(A) \leq (1+\varepsilon)tr(A)\right) \geq 1-\delta. \tag{7.2b}$$

In Chapter 5 we showed that the inequalities (7.2) hold if

$$n > 8c, \quad \text{where } c = c(\varepsilon,\delta) = \varepsilon^{-2}\ln(1/\delta). \tag{7.3}$$

However, this bound on n can be rather pessimistic and yields sample sizes which may not be practically appealing. Theorems 7.1 and 7.2 and Corollary 7.3 below provide tighter and hopefully more useful bounds on n. For the proof of these, we make use of Theorems 6.1 and 6.2 of Chapter 6.

Let us define

$$Q(n) := \frac{1}{n}Q_n,$$

where $Q_n \sim \chi^2_n$ denotes a chi-squared r.v of degree n. Note that $Q(n) \sim Gamma(n/2, n/2)$, i.e., a gamma r.v, parametrized by shape $\alpha = n/2$ and rate $\beta = n/2$ parameters with PDF given as (6.1). In case of several i.i.d gamma r.v's of this sort, we refer to the j^{th} r.v by $Q_j(n)$.

Theorem 7.1 (**Necessary and sufficient condition for** (7.2a))**.** *Given an SPSD matrix A of rank r and tolerances (ε,δ) as above, the following hold:*

(i) ***Sufficient condition:*** *there exists some integer $n_0 \geq 1$ such that*

$$\Pr\left(Q(n_0) < (1-\varepsilon)\right) \leq \delta. \tag{7.4}$$

Furthermore, (7.2a) *holds for all $n \geq n_0$.*

(ii) ***Necessary condition:*** *if* (7.2a) *holds for some $n_0 \geq 1$, then for all $n \geq n_0$*

$$P^-_{\varepsilon,r}(n) := \Pr\left(Q(nr) < (1-\varepsilon)\right) \leq \delta. \tag{7.5}$$

(iii) ***Tightness:*** *if the r positive eigenvalues of A are all equal (NB this always happens if*

$r = 1$), then there is a positive integer n_0 satisfying (7.5), such that (7.2a) holds iff $n \geq n_0$.

Proof. Since A is SPSD, it can be diagonalized by a unitary similarity transformation as $A = U^T \Lambda U$, where Λ is the diagonal matrix of eigenvalues sorted in non-increasing order. Consider n random vectors $\mathbf{w}_i,\ i = 1, \ldots, n$, whose components are i.i.d and drawn from the standard normal distribution, and define $\mathbf{z}_i = U\mathbf{w}_i$ for each i. Note that since U is unitary, the entries of $\mathbf{z}_i$ are i.i.d standard normal variables, like the entries of $\mathbf{w}_i$. We have

$$\begin{aligned}
\frac{tr_G^n(A)}{tr(A)} &= \frac{1}{n\ tr(A)} \sum_{i=1}^{n} \mathbf{w}_i^T A \mathbf{w}_i \\
&= \frac{1}{n\ tr(A)} \sum_{i=1}^{n} \mathbf{z}_i^T \Lambda \mathbf{z}_i \\
&= \frac{1}{n\ tr(A)} \sum_{i=1}^{n} \sum_{j=1}^{r} \lambda_j z_{ij}^2 \\
&= \sum_{j=1}^{r} \frac{\lambda_j}{n\ tr(A)} \sum_{i=1}^{n} z_{ij}^2 \\
&= \sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n),
\end{aligned}$$

where the λ_j's appearing in the sums are positive eigenvalues of A. Now, noting that

$$\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} = 1,$$

Theorem 6.2 yields

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n) \leq (1-\varepsilon) \right) \leq \Pr\big(Q(n) \leq (1-\varepsilon)\big) = P_{\varepsilon,1}^{-}(n), \tag{7.6a}$$

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n) \leq (1-\varepsilon) \right) \geq \Pr\big(Q(nr) \leq (1-\varepsilon)\big) = P_{\varepsilon,r}^{-}(n). \tag{7.6b}$$

In addition, for any given $r > 0$ and $\varepsilon > 0$, the function $P_{\varepsilon,r}^{-}(n)$ is monotonically decreasing on integers $n \geq 1$. This can be seen by Theorem 6.1 using the sequence $\alpha_i = (n_0 + (i-1))r/2,\ i \geq 1$.

The claims now easily follow by combining (7.6) and this decreasing property. □

Theorem 7.2 (**Necessary and sufficient condition for** (7.2b))**.** *Given an SPSD matrix A of rank r and tolerances (ε, δ) as above, the following hold:*

(i) ***Sufficient condition:*** *if the inequality*

$$\Pr\big(Q(n_0) \leq (1+\varepsilon)\big) \geq 1-\delta \tag{7.7}$$

is satisfied for some $n_0 > \varepsilon^{-1}$, then (7.2b) *holds with $n = n_0$. Furthermore, there is always an $n_0 > \varepsilon^{-2}$ such that* (7.7) *is satisfied and, for such n_0, it follows that* (7.2b) *holds for all $n \geq n_0$.*

(ii) ***Necessary condition:*** *if* (7.2b) *holds for some $n_0 > \varepsilon^{-1}$, then*

$$P^{+}_{\varepsilon,r}(n) := \Pr\big(Q(nr) \leq (1+\varepsilon)\big) \geq 1-\delta, \tag{7.8}$$

with $n = n_0$. Furthermore, if $n_0 > \varepsilon^{-2}r^{-2}$, then (7.8) *holds for all $n \geq n_0$.*

(iii) ***Tightness:*** *if the r positive eigenvalues of A are all equal, then there is a smallest $n_0 > \varepsilon^{-2}r^{-2}$ satisfying* (7.8) *such that for any $n \geq n_0$,* (7.2b) *holds, and for any $\varepsilon^2 r^{-2} < n < n_0$,* (7.2b) *does not hold. If δ is small enough so that* (7.8) *does not hold for any $n \leq \varepsilon^2 r^{-2}$, then n_0 is both necessary and sufficient for* (7.2b)*.*

Proof. The same unitary diagonalization argument as in the proof of Theorem 7.1 shows that

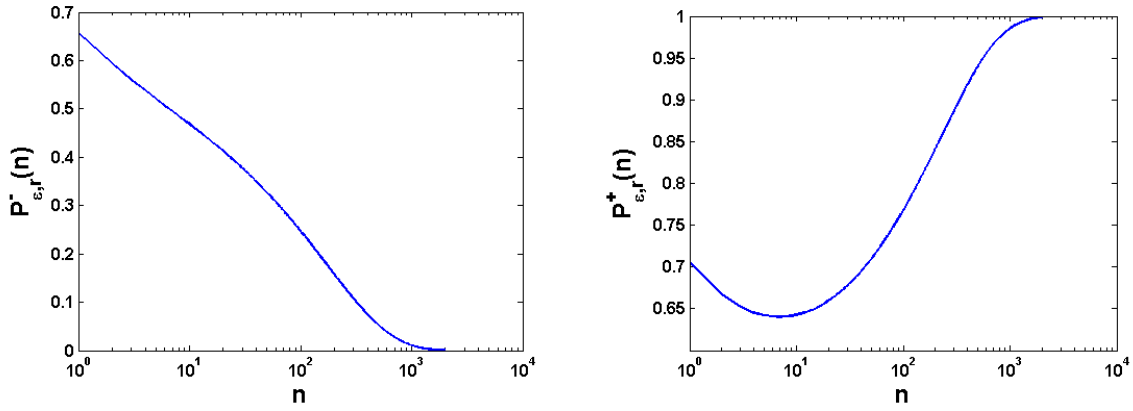
$$\Pr\Big(tr^n_G(A) < (1+\varepsilon)tr(A)\Big) = \Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n) < (1+\varepsilon)\right).$$

Now we see that if $n > \varepsilon^{-1}$, Theorem 6.2 with $\alpha = n/2$ yields

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n) \leq (1+\varepsilon)\right) \;\geq\; \Pr\big(Q(n) \leq (1+\varepsilon)\big) = P^{+}_{\varepsilon,1}(n), \tag{7.9a}$$

$$\Pr\left(\sum_{j=1}^{r} \frac{\lambda_j}{tr(A)} Q_j(n) \leq (1+\varepsilon)\right) \;\leq\; \Pr\big(Q(nr) \leq (1+\varepsilon)\big) = P^{+}_{\varepsilon,r}(n). \tag{7.9b}$$

In addition, for any given $r > 0$ and $\varepsilon > 0$, the function $P^{+}_{\varepsilon,r}(n)$ is monotonically increasing on integers $n > \varepsilon^{-2}r^{-2}$. This can be seen by Theorem 6.1 using the sequence $\alpha_i = (n_0 + (i-1))r/2,\ i \geq 1$. The claims now easily follow by combining (7.9) and this increasing property. □



(a) (b)

Figure 7.1: The curves of $P^{-}_{\varepsilon,r}(n)$ and $P^{+}_{\varepsilon,r}(n)$, defined in (7.5) and (7.8), for $\varepsilon = 0.1$ and $r = 1$: (a) $P^{-}_{\varepsilon,r}(n)$ decreases monotonically for all $n \geq 1$; (b) $P^{+}_{\varepsilon,r}(n)$ increases monotonically only for $n \geq n_0$, where $n_0 > 1$: according to Theorem 7.2, $n_0 = 100$ is safe, and this value does not disagree with the plot.

Remarks:

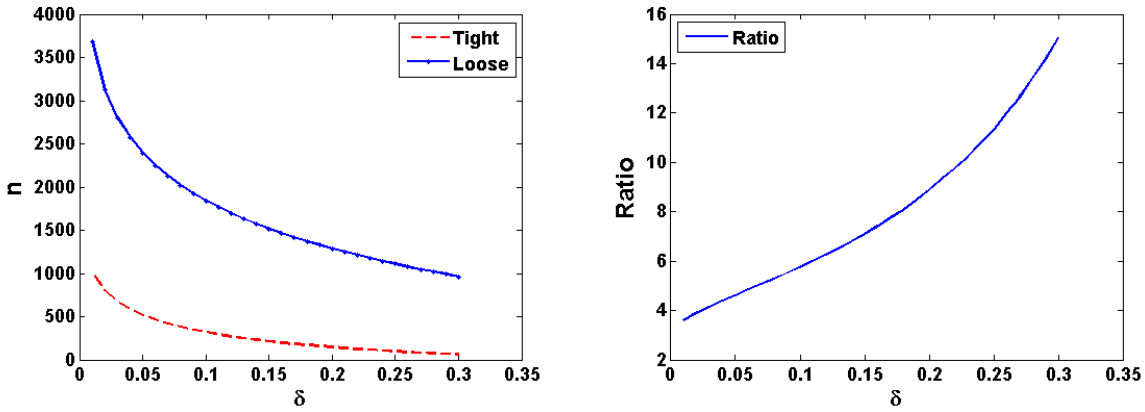
(i) Part (iii) of Theorem 7.2 states that if δ is not small enough, then n_0 might not be a necessary and sufficient sample size for the special matrices mentioned there, i.e., matrices with $\lambda_1 = \lambda_2 = \cdots = \lambda_r$. This can be seen from Figure 7.1(b): for $r = 1, \varepsilon = 0.1$, if $\delta = 0.33$, say, there is an integer $10 < n \leq 100$ such that (7.2b) holds, so $n = 101$ is no longer a necessary sample size (although it is still sufficient).

(ii) Simulations show that the sufficient sample size obtained using Theorems 7.1 and 7.2, amounts to bounds of the form $\mathcal{O}\left(c(\varepsilon,\delta)g(\delta)\right)$, where $g(\delta) < 1$ is a decreasing function of δ and $c(\varepsilon,\delta)$ is as defined in (7.3). As such, for larger values of δ, i.e., when larger uncertainty is allowed, one can obtain significantly smaller sample sizes than the one predicted by (7.3); see Figures 7.2 and 7.3. In other words, the difference between the above tighter conditions and (7.3) is increasingly more prominent as δ gets larger.

(iii) Note that the results in Theorems 7.1 and 7.2 are independent of the size of the matrix.

In fact, the first items (i) in both theorems do not require any a priori knowledge about the matrix, other than it being SPSD. In order to compute the necessary sample sizes, though, one is required to also know the rank of the matrix.

(iv) The conditions in our theorems, despite their potentially ominous look, are actually simple to compute. Appendix A.4 contains a short MATLAB code which calculates these necessary or sufficient sample sizes to satisfy the probabilistic accuracy guarantees (7.2), given a pair (ε, δ) (and the matrix rank r in case of necessary sample sizes). This code was used for generating Figures 7.2 and 7.3.



(a) (b)

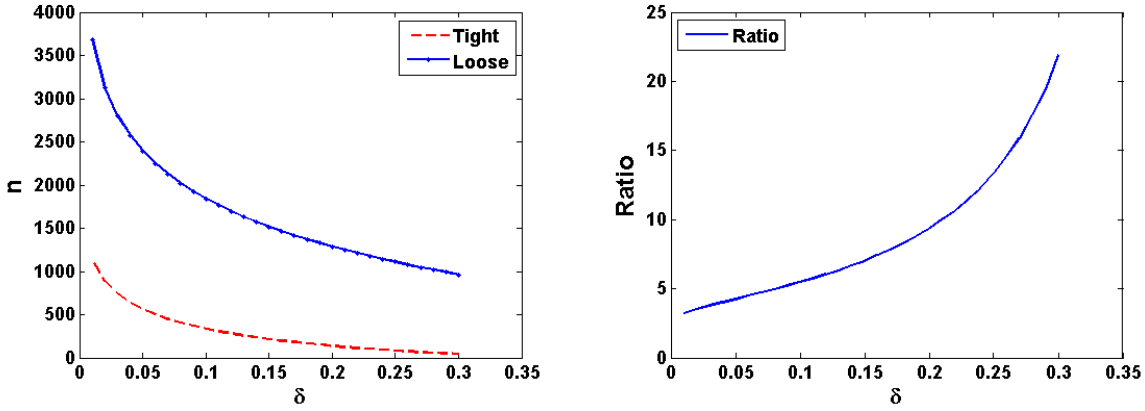
Figure 7.2: Comparing, as a function of δ, the sample size obtained from (7.4) and denoted by "tight", with that of (7.3) and denoted by "loose", for $\varepsilon = 0.1$ and $0.01 \leq \delta \leq 0.3$: (a) sufficient sample size, n, for (7.2a), (b) ratio of sufficient sample size obtained from (7.3) over that of (7.4). When δ is relaxed, our new bound is tighter than the older one by an order of magnitude.

Combining Theorems 7.1 and 7.2, we can easily state conditions on the sample size n for which the condition

$$\Pr\left(|tr_G^n(A) - tr(A)| \leq \varepsilon\ tr(A)\right) \geq 1 - \delta \tag{7.10}$$

holds. We have the following immediate corollary:

Corollary 7.3 (**Necessary and sufficient condition for** (7.10))**.** *Given an SPSD matrix A of rank r and tolerances (ε, δ) as above, the following hold:*



(a) (b)

Figure 7.3: Comparing, as a function of δ, the sample size obtained from (7.7) and denoted by "tight", with that of (7.3) and denoted by "loose", for $\varepsilon = 0.1$ and $0.01 \leq \delta \leq 0.3$: (a) sufficient sample size, n, for (7.2b), (b) ratio of sufficient sample size obtained from (7.3) over that of (7.7). When δ is relaxed, our new bound is tighter than the older one by an order of magnitude.

(i) ***Sufficient condition:*** *if the inequality*

$$\Pr\left((1-\varepsilon) \leq Q(n_0) \leq (1+\varepsilon)\right) \geq 1-\delta \tag{7.11}$$

is satisfied for some $n_0 > \varepsilon^{-1}$, *then* (7.10) *holds with* $n = n_0$. *Furthermore, there is always an* $n_0 > \varepsilon^{-2}$ *such that* (7.11) *is satisfied and, for such* n_0, *it follows that* (7.10) *holds for all* $n \geq n_0$.

(ii) ***Necessary condition:*** *if* (7.10) *holds for some* $n_0 > \varepsilon^{-1}$, *then*

$$\Pr\left((1-\varepsilon) \leq Q(nr) \leq (1+\varepsilon)\right) \geq 1-\delta, \tag{7.12}$$

with $n = n_0$. *Furthermore, if* $n_0 > \varepsilon^{-2}r^{-2}$, *then* (7.12) *holds for all* $n \geq n_0$.

(iii) ***Tightness:*** *if the* r *positive eigenvalues of* A *are all equal then there is a smallest* $n_0 > \varepsilon^{-2}r^{-2}$ *satisfying* (7.12) *such that for any* $n \geq n_0$, (7.10) *holds, and for any* $\varepsilon^{-2}r^{-2} < n < n_0$, (7.10) *does not hold. If* δ *is small enough so that* (7.12) *does not hold for any* $n \leq \varepsilon^{-2}r^{-2}$, *then* n_0 *is both necessary and sufficient for* (7.10).

Remark: The necessary condition in Corollary 7.3(ii) is only valid for $n > \varepsilon^{-1}$ (this is a consequence of the condition (7.12) being tight, as shown in part (iii)). In Section 5.3.2, an "almost tight" necessary condition is given that works for all $n \geq 1$.

7.2 Quantifying the Uncertainty in Randomized Algorithms

As described in Section 1.2, consider the problem of decreasing the value of the original objective (1.6) to a desired level (e.g., satisfying a given tolerance) to recover the sought model, $\mathbf{m}$. Namely, consider an iterative method such as modied Gauss-Newton (GN), using sensitivity matrices

$$J_i(\mathbf{m}) = \frac{\partial \mathbf{f}(\mathbf{m}, \mathbf{q}_i)}{\partial \mathbf{m}}, \quad i = 1, \ldots, s$$

and the gradient

$$\nabla \phi(\mathbf{m}) = 2 \sum_{i=1}^{s} J_i^T(\mathbf{m})(\mathbf{f}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i).$$

As in Chapter 3, what is special in our context here is that the update direction, $\delta \mathbf{m}_k$, is calculated using the approximate misfit, $\widehat{\phi}(\mathbf{m}_k, n_k)$, defined as described in (2.3) (n_k is the sample size used for this approximation in the k^{th} iteration). However, since the ultimate goal is to fit the original data, we need to assess whether the value of the original objective is also decreased using this new iterate. The challenge is to do this as well as check for termination of the iteration process with a minimal number of evaluations of the prohibitively expensive original misfit function ϕ.

In this section, we extend the algorithms introduced in Chapters 3 and 4. Variants of modified stochastic steps in the original algorithms are presented, and using Theorems 7.1 and 7.2, the uncertainties in these steps are quantified. More specifically, in Algorithm 2 introduced in Chapter 3, following a stabilized GN iteration on the approximated objective function using the approximated misfit, the iterate is updated, and some (or all) of the following steps are performed:

(i) *cross validation* (see Section 3.1.1) – approximate assessment of this iterate in terms of

sufficient decrease in the objective function using a control set of random combinations of measurements. More specifically, at the k^{th} iteration with the new iterate $\mathbf{m}_{k+1}$, we test whether the condition (3.2), namely

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \kappa \widehat{\phi}(\mathbf{m}_k, n_k)$$

(cf. (2.3)) holds for some $\kappa \leq 1$, employing an independent set of weight vectors used in both approximations of ϕ;

(ii) *uncertainty check* (see Section 3.1.2) – upon success of cross validation, an inexpensive plausible termination test is performed where, given a tolerance ρ, we check for the condition (3.4), namely

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_k) \leq \rho$$

using a fresh set of random weight vectors; and

(iii) *stopping criterion* (see Section 3.1.2) – upon success of the uncertainty check, an additional independent and potentially more rigorous termination test against the given tolerance ρ is performed (possibly using the original misfit function).

The role of the cross validation step within an iteration is to assess whether the true objective function at the current iterate has (sufficiently) decreased compared to the previous one. If this test fails, we deem that the current sample size is not sufficiently large to yield an update that decreases the original objective, and the fitting step needs to be repeated using a larger sample size, see [46]. In Chapter 3, this step was used heuristically, so the amount of uncertainty in such validation of the current iterate was not quantified. Consequently, there was no handle on the amount of false positives/negatives in such approximate evaluations (e.g., a sample size could be deemed too small while the stabilized GN iteration has in fact produced an acceptable iterate). In addition, in Chapter 3 the sample size for the uncertainty check was heuristically chosen. So this step was also performed with no control over the amount of uncertainty.

For the stopping criterion step in Chapter 3 as well as [46], the objective function was accurately evaluated using all s experiments, which is clearly a very expensive choice for an algorithm termination check. This was a judicious decision made in order to be able to have

a fairer comparison of the new and different methods proposed there. Replacement of this termination criterion by another independent heuristic "uncertainty check" is experimented with in Chapter 4.

In this section, we address the issues of quantifying the uncertainty in the validation, uncertainty check and stopping criterion steps within a nonlinear iteration. In what follows we continue to assume, for simplicity, that the iterations are performed on the objective (1.6) using dynamic regularization (or iterative regularization [45, 78, 132]) where the regularization is performed implicitly. Extension to the case (1.7) is straightforward. Throughout, we assume to be given a pair of positive and small probabilistic tolerance numbers, (ε, δ).

7.2.1 Cross Validation Step with Quantified Uncertainty

The condition (3.2) is an independent, unbiased indicator of (3.1), which indicates sufficient decrease in the objective. If (3.2) is satisfied then the current sample size, n_k, is considered sufficiently large to capture the original misfit well enough to produce a valid iterate, and the algorithm continues using the same sample size. Otherwise, the sample size is deemed insufficient and is increased. Using Theorems 7.1 and 7.2, we can now remove the heuristic characteristic as to *when* this sample size increase has been performed hitherto, and present two variants of (3.2) where the uncertainties in the validation step are quantified.

Assume we have a sample size n_c such that

$$Pr\left(\widehat{\phi}(\mathbf{m}_k, n_c) \leq (1+\varepsilon)\phi(\mathbf{m}_k)\right) \geq 1-\delta, \tag{7.13a}$$

$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \geq (1-\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{7.13b}$$

If in the procedure outlined above, after obtaining the updated iterate $\mathbf{m}_{k+1}$, we verify that

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq \kappa \left(\frac{1-\varepsilon}{1+\varepsilon}\right) \widehat{\phi}(\mathbf{m}_k, n_c), \tag{7.14}$$

then it follows from (7.13) that $\phi(\mathbf{m}_{k+1}) \leq \kappa\phi(\mathbf{m}_k)$ with a probability of, at least, $(1-\delta)^2$. In other words, success of (7.14) indicates that the updated iterate decreases the value of the original misfit (1.6) with a probability of, at least, $(1-\delta)^2$.

Alternatively, suppose that we have

$$Pr\left(\widehat{\phi}(\mathbf{m}_k, n_c) \geq (1-\varepsilon)\phi(\mathbf{m}_k)\right) \geq 1-\delta, \tag{7.15a}$$

$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq (1+\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{7.15b}$$

Now, if instead of (7.14) we check whether or not

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_c) \leq \kappa\left(\frac{1+\varepsilon}{1-\varepsilon}\right)\widehat{\phi}(\mathbf{m}_k, n_c), \tag{7.16}$$

then it follows from (7.15) that if the condition (7.16) is *not* satisfied, then $\phi(\mathbf{m}_{k+1}) > \kappa\phi(\mathbf{m}_k)$ with a probability of, at least, $(1-\delta)^2$. In other words, failure of (7.16) indicates that the updated iterate results in an insufficient decrease in the original misfit (1.6) with a probability of, at least, $(1-\delta)^2$.

We can replace (3.2) with either of the conditions (7.14) or (7.16) and use the conditions (7.4) or (7.7) to calculate the cross validation sample size, n_c. If the relevant check (7.14) or (7.16) fails, we deem the sample size used in the fitting step, n_k, to be too small to produce an iterate which decreases the original misfit (1.6), and consequently consider increasing the sample size, n_k. Note that since $\frac{1-\varepsilon}{1+\varepsilon} < 1 < \frac{1+\varepsilon}{1-\varepsilon}$, the condition (7.14) results in a more aggressive strategy for increasing the sample size used in the fitting step than the condition (7.16). Figure 7.8 in Section 7.3 demonstrates this within the context of an application.

Remarks:

(i) Larger values of ε result in more aggressive (or relaxed) descent requirement by the condition (7.14) (or (7.16)).

(ii) As the iterations progress and we get closer to the solution, the decrease in the original objective could be less than what is imposed by (7.14). As a result, if ε is too large, we might never successfully pass the cross validation test. One useful strategy to alleviate this is to start with a larger ε, decreasing it as we get closer to the solution. A similar strategy can be adopted for the case when the condition (7.16) is used as a cross validation: as the iterations get closer to the solution, one can make the condition (7.16) less relaxed by decreasing ε.

7.2.2 Uncertainty Check with Quantified Uncertainty and Efficient Stopping Criterion

The usual test for terminating the iterative process is to check for condition (3.3), namely

$$\phi(\mathbf{m}_{k+1}) \leq \rho,$$

for a given tolerance ρ. However, this can be very expensive in our current context; see Section 7.3 and Tables 7.1 and 7.2 for examples of a scenario where one misfit evaluation using the entire data set can be as expensive as the entire cost of an efficient, complete algorithm. In addition, if the exact value of the tolerance ρ is not known (which is usually the case in practice), one should be able to reflect such uncertainty in the stopping criterion and perform a softer version of (3.3). Hence, it could be useful to have an algorithm which allows one to adjust the cost and accuracy of such an evaluation in a quantifiable way, and find the balance that is suitable to particular objectives and computational resources.

Regardless of the issues of cost and accuracy, this evaluation should be carried out as rarely as possible and only when deemed timely. In Chapter 3, we addressed this by employing an "uncertainty check" (3.4) as described earlier in this section, heuristically. Using Theorems 7.1 and 7.2, we now devise variants of (3.4) with quantifiable uncertainty. Subsequently, again using Theorems 7.1 and 7.2, we present a much cheaper stopping criterion than (3.3) which, at the same time, reflects our uncertainty in the given tolerance.

Assume that we have a sample size n_u such that

$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \geq (1-\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta. \tag{7.17}$$

If the updated iterate, $\mathbf{m}_{k+1}$, successfully passes the cross validation test, then we check for

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1-\varepsilon)\rho. \tag{7.18}$$

If this holds too then it follows from (7.17) that $\phi(\mathbf{m}_{k+1}) \leq \rho$ with a probability of, at least, $(1-\delta)$. In other words, success of (7.18) indicates that the misfit is likely to be below the

tolerance with a probability of, at least, $(1-\delta)$.

Alternatively, suppose that

$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1+\varepsilon)\phi(\mathbf{m}_{k+1})\right) \geq 1-\delta, \tag{7.19}$$

and instead of (7.18) we check for

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_u) \leq (1+\varepsilon)\rho. \tag{7.20}$$

then it follows from (7.19) that if the condition (7.20) is *not* satisfied, then $\phi(\mathbf{m}_{k+1}) > \rho$ with a probability of, at least, $(1-\delta)$. In other words, failure of (7.20) indicates that using the updated iterate, the misfit is likely to be still above the desired tolerance with a probability of, at least, $(1-\delta)$.

We can replace (3.4) with the condition (7.18) (or (7.20)) and use the condition (7.4) (or (7.7)) to calculate the uncertainty check sample size, n_u. If the test (7.18) (or (7.20)) fails then we skip the stopping criterion check and continue iterating. Note that since $(1-\varepsilon) < 1 < (1+\varepsilon)$, the condition (7.18) results in fewer false positives than the condition (7.20). On the other hand, the condition (7.20) is expected to results in fewer false negatives than the condition (7.18). The choice of either alternative is dependent on one's requirements, resources and the application on hand.

The stopping criterion step can be performed in the same way as the uncertainty check but potentially with higher certainty in the outcome. In other words, for the stopping criterion we can choose a smaller δ, resulting in a larger sample size n_t satisfying $n_t > n_u$, and check for satisfaction of either

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1-\varepsilon)\rho, \tag{7.21a}$$

or

$$\widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1+\varepsilon)\rho. \tag{7.21b}$$

Clearly the condition (7.21b) is a softer than (7.21a): a successful (7.21b) is only necessary and not sufficient for concluding that (3.3) holds with the prescribed probability.

In practice, when the value of the stopping criterion threshold, ρ, is not *exactly* known (it is often crudely estimated using the measurements), one can reflect such uncertainty in ρ by choosing an appropriately large δ. Smaller values of δ reflect a higher certainty in ρ and a more rigid stopping criterion.

Remarks:

(i) If ε is large then using (7.21a), one might run the risk of over-fitting. Similarly, using (7.21b) with large ε, there is a risk of under-fitting. Thus, appropriate values of ε need to be considered in accordance with the application and one's computational resources and experience.

(ii) The same issues regarding large ε arise when employing the uncertainty check condition (7.18) (or (7.20)): large ε might increase the frequency of false negatives (or positives).

7.2.3 Algorithm

We now present an extension of Algorithm 2 for approximately solving NLS formulations of (1.6) or (1.7). By performing cross validation, uncertainty check and stopping criterion as descried in Section 7.2.1 and Section 7.2.2, we can devise 8 variants of Algorithm 4 below. Depending on the application, the variant of choice can be selected appropriately. More specifically, cross validation, uncertainty check and stopping criterion can, respectively, be chosen to be one of the following combinations (referring to their equation numbers):

(i) (7.14 - 7.18 - 7.21a)	(ii) (7.14 - 7.18 - 7.21b)
(iii) (7.14 - 7.20 - 7.21a)	(iv) (7.14 - 7.20 - 7.21b)
(v) (7.16 - 7.18 - 7.21a)	(vi) (7.16 - 7.18 - 7.21b)
(vii) (7.16 - 7.20 - 7.21a)	(viii) (7.16 - 7.20 - 7.21b)

Remark:

(i) The sample size, n_k, used in the fitting step of Algorithm 4 could in principle be determined by Corollary 7.3, using a pair of tolerances $(\varepsilon_f, \delta_f)$. If cross validation (7.14) (or (7.16)) fails, the tolerance pair $(\varepsilon_f, \delta_f)$ is reduced to obtain, in the next iteration, a

larger fitting sample size, n_{k+1}. This would give a sample size which yields a quantifiable approximation with a desired relative accuracy. However, in the presence of all the added safety steps described in this section, we have found in practice that Algorithm 4 is capable of producing a satisfying recovery, even with a significantly smaller n_k than the one predicted by Corollary 7.3. Thus, the "*how*" of the fitting sample size increase is left to heuristic (as opposed to its "*when*", which is quantified as described in Section 7.2.1).

(ii) In the algorithm below, we only consider fixed values (i.e., independent of k) for ε and δ. One can easily modify Algorithm 4 to incorporate non-stationary values which adapt to the iteration process, as mentioned in the closing remark of Section 7.2.1.

In Algorithm 4, when we draw vectors $\mathbf{w}_i$ for some purpose, we always draw them independently from the standard normal distribution.

7.3 Numerical Experiments

In this section, we numerically demonstrate the efficacy of Algorithm 4 by applying it to the important class of problems described in Section 1.1.2: large scale PDE constrained inverse problems with many measurements. We show below the capability of our method by applying it to such examples in the context of the DC resistivity/EIT problem (see Section 3.3.1), as in Chapters 3 and 4 as well as [46, 70, 71, 111].

We consider the forward operators as defined in (1.5) where the linearity assumption (A.2) is satisfied (i.e., the locations where data are measured do not change from one experiment to another, i.e., $P = P_i, \forall i$). Hence, we can use Algorithm 4 to efficiently recover $\mathbf{m}$ and be quantifiably confident in the recovered model. If the P_i's are different across experiments, it might be possible to use methods such as the ones introduced in Chapter 4 or [70] to extend the existing data set to one where all sources share the same receivers. Using these methods (when they apply!), one can effectively restore the linearity assumption (A.2) and transform the problem (1.4) to (1.5), for which Algorithm 4 can be employed.

Considering the inverse problem with the PDE model (3.5), below we give two examples, each having a piecewise constant "exact solution", or "true model", used to synthesize data:

Algorithm 4 Solve NLS formulation of (1.6) (or (1.7)) using uncertainty check, cross validation and cheap stopping criterion

Given: sources $\mathbf{q}_i$, $i = 1, \ldots, s$, measurements $\mathbf{d}_i$, $i = 1, \ldots, s$, stopping criterion level ρ, objective function sufficient decrease factor $\kappa \leq 1$, pairs of small numbers $(\varepsilon_c, \delta_c)$, $(\varepsilon_u, \delta_u)$, $(\varepsilon_t, \delta_t)$, and initial guess $\mathbf{m}_0$.

Initialize:

- $\mathbf{m} = \mathbf{m}_0$, $n_0 = 1$
- Calculate the cross validation sample size, n_c, as described in Section 7.2.1 with $(\varepsilon_c, \delta_c)$.
- Calculate the sample sizes for uncertainty check, n_u, and stopping criterion, n_t, as described in Section 7.2.2 with $(\varepsilon_u, \delta_u)$ and $(\varepsilon_t, \delta_t)$, respectively.

for $k = 0, 1, 2, \cdots$ until termination **do**

  Fitting:

  - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_k$.
  - Approximate the misfit term and potentially its gradient in (1.6) or (1.7) using (2.3) with the above weights and $n = n_k$.
  - Find an update for the objective function using the approximated misfit (2.3).

  Cross Validation:

  - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_c$.

  if (7.14) (or (7.16)) holds **then**

    Uncertainty Check:

    - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_u$.

    if (7.18) (or (7.20)) holds **then**

      Stopping Criterion:

      - Draw $\mathbf{w}_i$, $i = 1, \ldots, n_t$.

      if (7.21a) (or (7.21b)) holds **then**

        - Terminate

      end if

    end if

    - Set $n_{k+1} = n_k$.

  else

    - **Sample Size Increase**: for example, set $n_{k+1} = \min(2n_k, s)$.

  end if

end for

(E.1) in our simpler model a target object with conductivity $\mu_t = 1$ has been placed in a background medium with conductivity $\mu_b = 0.1$ (see Figure 7.4(a)); and

(E.2) in a slightly more complex setting a conductive object with conductivity $\mu_c = 0.01$, as well as a resistive one with conductivity $\mu_r = 1$, have been placed in a background medium with conductivity $\mu_b = 0.1$ (see Figure 7.6(a)). Note that the recovery of the model in Example (E.2) is more challenging than Example (E.1) since here the dynamic range of the conductivity is much larger.

Details of the numerical setup for the following examples are given in Section 3.3.2.

Example (E.1)

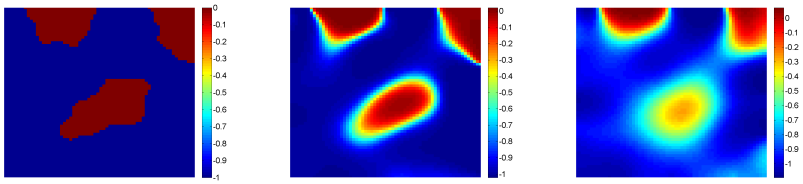
We carry out the 8 variants of Algorithm 4 for the parameter values $(\varepsilon_c, \delta_c) = (0.05, 0.3)$, $(\varepsilon_u, \delta_u) = (0.1, 0.3)$, $(\varepsilon_t, \delta_t) = (0.1, 0.1)$, and $\kappa = 1$. The resulting total count of PDE solves, which is the main computational cost of the iterative solution of such inverse problems, is reported in Tables 7.1 and 7.2. As a point of reference, we also include the total PDE count using the "plain vanilla" stabilized Gauss-Newton method which employs the entire set of s experiments at every iteration and misfit estimation task. The recovered conductivities are displayed in Figures 7.5 and 7.7, demonstrating that employing Algorithm 4 can drastically reduce the total work while obtaining equally acceptable reconstructions.

Vanilla	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
436,590	4,058	4,028	3,764	3,282	4,597	3,850	3,734	3,321

Table 7.1: Example (E.1). Work in terms of number of PDE solves for all variants of Algorithm 4, described in Section 7.2.3 and indicated here by (i)–(viii). The "vanilla" count is also given, as a reference.

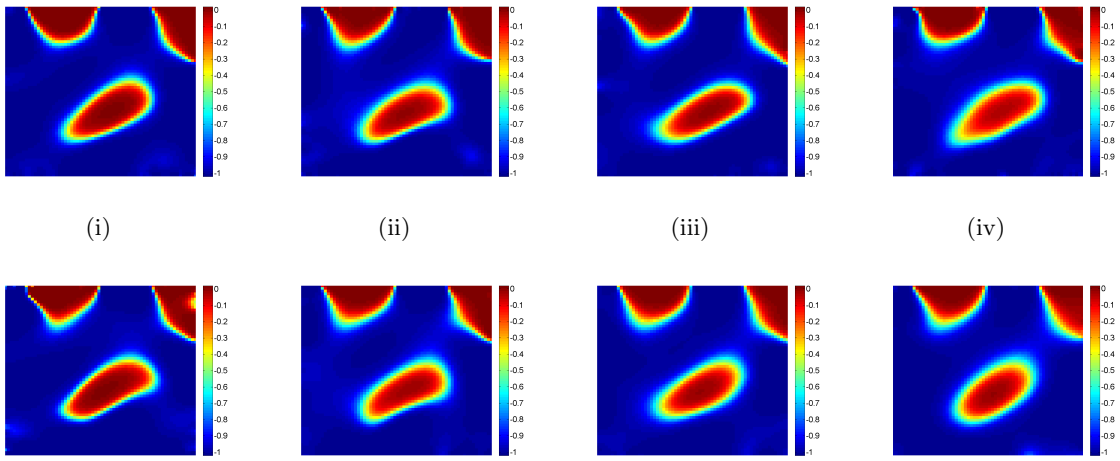
For the calculations displayed here we have employed *dynamical regularization* [45, 132]. In this method there is no explicit regularization term $R(\mathbf{m})$ in (1.7) and the regularization is done implicitly and iteratively.

The quality of reconstructions obtained by the various variants in Figure 7.5 is comparable to that of the "vanilla" with $s = 3,969$ in Figure 7.4(b). In contrast, employing only $s = 49$ data sets corresponding to similar experiments distributed over a coarser grid yields an inferior



(a) (b) (c)

Figure 7.4: Example (E.1). Plots of log-conductivity: (a) True model; (b) Vanilla recovery with $s = 3,969$; (c) Vanilla recovery with $s = 49$. The vanilla recovery using only 49 measurement sets is clearly inferior, showing that a large number of measurement sets can be crucial for better reconstructions.



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

Figure 7.5: Example (E.1). Plots of log-conductivity of the recovered model using the 8 variants of Algorithm 4, described in Section 7.2.3 and indicated here by (i)–(viii). The quality of reconstructions is generally comparable to that of plain vanilla with $s = 3,969$ and across variants.

reconstruction in Figure 7.4(c). The cost of this latter run is $5,684$ PDE solves, which is more expensive than our randomized algorithms for the much larger s. Furthermore, comparing Figures 7.4(b) and 7.5 to Figures 4.3 and 4.4 in Chapter 4, which shows similar results for $s = 961$ data sets, we again see a relative improvement in reconstruction quality. All of this goes to show that a large number of measurements s can be crucial for better reconstructions. Thus, it is not the case that one can dispense with a large portion of the measurements and still expect the same quality reconstructions. Hence, it is indeed useful to have algorithms such as Algorithms 1, 2, 3, or 4 that, while taking advantage of the entire available data, can efficiently carry out the computations and yet obtain credible reconstructions.

We have resisted the temptation to make comparisons between values of $\phi(\mathbf{m}_{k+1})$ and $\hat{\phi}(\mathbf{m}_{k+1})$ for various iterates. There are two major reasons for that. The first is that $\hat{\phi}$ values in bounds such as (7.14), (7.16), (7.18), (7.20) and (7.21) are different and are always compared against tolerances in context that are based on noise estimates. In addition, the sample sizes that we used for uncertainty check and stopping criteria, since they are given by Theorems 7.1 and 7.2, already determine how far the estimated misfit is from the true misfit. The second (and more important) reason is that in such a highly diffusive forward problem as DC resistivity, misfit values are typically far closer to one another than the resulting reconstructed models $\mathbf{m}$ are. A good misfit is merely a necessary condition, which can fall significantly short of being sufficient, for a good reconstruction; see [69] and Chapter 4.

Example (E.2)

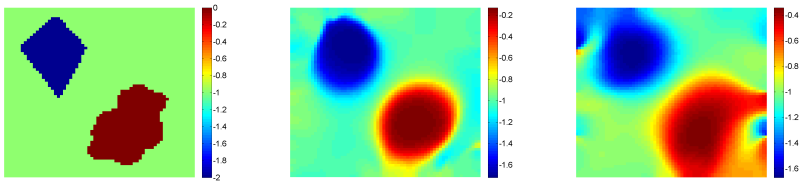
Here we have imposed prior knowledge on the "*discontinuous*" model in the form of total variation (TV) regularization [34, 38, 47]. Specifically, $R(\mathbf{m})$ in (1.7) is the discretization of the TV functional $\int_\Omega |\nabla m(\mathbf{x})|$. For implementation details of TV functional see Appendix A.5. For each recovery, the regularization parameter, α, has been chosen by trial and error within the range $[10^{-6}, 10^{-3}]$ to visually yield the best quality recovery.

Vanilla	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
476,280	5,631	5,057	5,011	3,990	6,364	4,618	4,344	4,195

Table 7.2: Example (E.2). Work in terms of number of PDE solves for all variants of Algorithm 4, described in Section 7.2.3 and indicated here by (i)–(viii). The "vanilla" count is also given, as a reference.

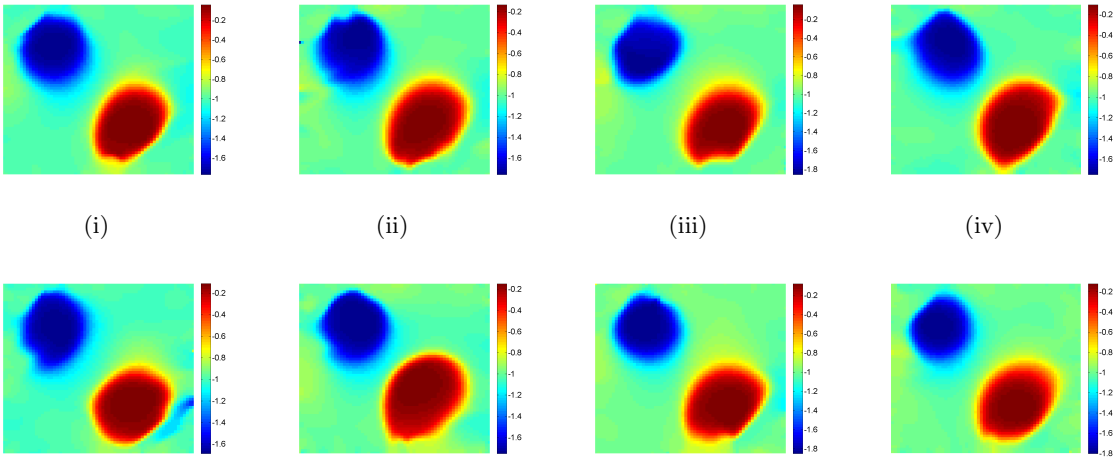
Table 7.2 and Figures 7.6 and 7.7 tell a similar story as in Example (E.1). The quality of reconstructions with $s = 3,969$ by the various variants, displayed in Figure 7.7, is comparable to that of the "vanilla" version in Figure 7.6(b), yet is obtained at only at a fraction (about 1%) of the cost. The "vanilla" solution for $s = 49$ displayed in Figure 7.6(c), costs $5,978$ PDE solves, which again is a higher cost for an inferior reconstruction compared to our Algorithm 4.

It is clear from Tables 7.1 and 7.2 that for most of these examples, variants (i)–(iv) which use the more aggressive cross validation (7.14) are at least as efficient as their respective counterparts, namely, variants (v)–(viii) which use (7.16). This suggests that, sometimes, a more



(a) (b) (c)

Figure 7.6: Example (E.2). Plots of log-conductivity: (a) True model; (b) Vanilla recovery with $s = 3,969$; (c) Vanilla recovery with $s = 49$. The vanilla recovery using only 49 measurement sets is clearly inferior, showing that a large number of measurement sets can be crucial for better reconstructions.



(i) (ii) (iii) (iv)

(v) (vi) (vii) (viii)

Figure 7.7: Example (E.2). Plots of log-conductivity of the recovered model using the 8 variants of Algorithm 4, described in Section 7.2.3 and indicated here by (i)–(viii). The quality of reconstructions is generally comparable to each other and that of plain vanilla with $s = 3,969$.

aggressive sample size increase strategy may be a better option; see also the numerical examples in Chapter 3. Notice that for all variants, the entire cost of the algorithm is comparable to one single evaluation of the misfit function $\phi(\mathbf{m})$ using the full data set!

7.4 Conclusions

In this chapter, we have proved tight necessary and sufficient conditions for the sample size, n, required to reach, with a probability of at least $1 - \delta$, (one-sided) approximations, using Gaussian estimator, for $tr(A)$ to within a relative tolerance ε. All of the sufficient conditions are computable in practice and do not assume any a priori knowledge about the matrix. If the

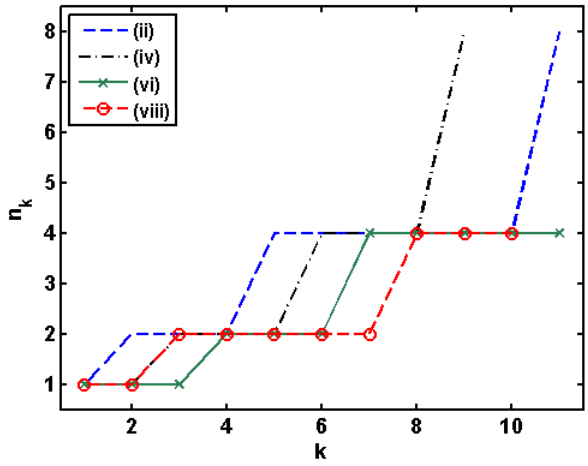


Figure 7.8: Example (E.2). Growth of the fitting sample size, n_k, as a function of the iteration k, upon using cross validation strategies (7.14) and (7.16). The graph shows the fitting sample size growth for variants (ii) and (vi) of Algorithm 4, as well as their counterparts, namely, variants (vi) and (viii). Observe that for variants (ii) and (iv) where (7.14) is used, the fitting sample size grows at a more aggressive rate than for variants (vi) and (viii) where (7.16) is used.

rank of the matrix is known then the necessary bounds can also be computed in practice.

Subsequently, using these conditions, we have presented eight variants of a general-purpose algorithm for solving an important class of large scale non-linear least squares problems. These algorithms can be viewed as an extended version of those in Chapters 3 and 4, where the uncertainty in most of the stochastic steps is quantified. Such uncertainty quantification allows one to have better control over the behavior of the algorithm and have more confidence in the recovered solution. The resulting algorithm is presented in Section 7.2.3.

Furthermore, we have demonstrated the performance of our algorithm using an important class of problems which arise often in practice, namely, PDE inverse problems with many measurements. By examining our algorithm in the context of the DC resistivity problem as an instance of such class of problems, we have shown that Algorithm 4 can recover solutions with remarkable efficiency. This efficiency is comparable to similar heuristic algorithms proposed in Chapters 3 and 4. The added advantage here is that with the uncertainty being quantified, the user can have more confidence in the approximate solution obtained by our algorithms.

Tables 7.1 and 7.2 show the amount of work (in PDE solves) of the 8 variants of our algorithm. Compared to a similar algorithm which uses the entire data set, an efficiency im-

provement by two orders of magnitude is observed. For most of the examples considered, the same tables also show that the more aggressive cross validation strategy (7.14) is, at least, as efficient as the more relaxed strategy (7.16). A thorough comparison of the behavior of these cross validation strategies (and all of the variants, in general) on different examples and model problems is left for future work.

Chapter 8

Algorithms That Satisfy a Stopping Criterion, Probably

Iterative numerical algorithms are typically equipped with a stopping criterion, where the iteration process is terminated when some error or misfit measure is deemed to be below a given tolerance. This is a useful setting for comparing algorithm performance, among other purposes.

However, in practical applications a precise value for such a tolerance is rarely known; rather, only some possibly vague idea of the desired quality of the numerical approximation is at hand. We discuss three case studies from different areas of numerical computation, where uncertainty in the error tolerance value and in the stopping criterion is revealed in different ways. This leads us to think of approaches to relax the notion of exactly satisfying a tolerance value.

We then concentrate on a *probabilistic* relaxation of the given tolerance. Relaxing the notion of an error tolerance in such a way allows the development of theory towards an uncertainty quantification of Monte Carlo methods (e.g., [2, 22, 84, 87, 138]). For example, this allows derivation of proven bounds on the sample size of certain Monte Carlo methods, as in Chapters 5 and 7. Such error relaxation was introduced in Chapter 7 and was incorporated in Algorithm 4. We show that Algorithm 4 becomes more efficient in a controlled way as the uncertainty in the tolerance increases, and we demonstrate this in the context of a class of inverse problems discussed in Section 1.1.2.

8.1 Introduction

A typical iterative algorithm in numerical analysis and scientific computing requires a stopping criterion. Such an algorithm involves a sequence of generated iterates or steps, an error tolerance, and a method to compute (or estimate) some quantity related to the error. If this error

quantity is below the tolerance then the iterative procedure is stopped and success is declared.

The actual manner in which the error in an iterate is estimated can vary all the way from being rather complex to being as simple as the normed difference between two consecutive iterates. Further, the "tolerance" may actually be a set of values involving combinations of absolute and relative error tolerances. There are several fine points to this, often application-dependent, that are typically incorporated in mathematical software packages (see for instance MATLAB's various packages for solving ordinary differential equation (ODE) or optimization problems). That makes some authors of introductory texts devote significant attention to the issue, while others attempt to ignore it as much as possible (cf. [14, 40, 80]). Let us choose here the middle way of considering a stopping criterion in a general form

$$\text{error\_estimate}(k) \leq \rho, \tag{8.1}$$

where k is the iteration or step counter, and $\rho > 0$ is the tolerance, assumed given.

But now we ask, *is ρ really given?!* Related to this, we can also ask, *to what extent is the stopping criterion adequate?*

- The numerical analyst would certainly *like* ρ to be given. That is because their job is to invent new algorithms, prove various assertions regarding convergence, stability, efficiency, and so on, and compare the new algorithm to other known ones for a similar task. For the latter aspect, a rigid deterministic tolerance for a trustworthy error estimate is indispensable.

  Indeed, in research areas such as image processing where criteria of the form (8.1) do not seem to capture certain essential features and the "eye norm" rules, a good comparison between competing algorithms can be far more delicate. Moreover, accurate comparisons of algorithms that require stochastic input can be tricky in terms of reproducing claimed experimental results.

- On the other hand, a practitioner who is the customer of numerical algorithms, applying them in the context of some complicated practical application that needs to be solved, will more often than not find it very hard to justify a particular choice of a precise value

for ρ in (8.1).

Our first task in what follows is to convince the reader that often in practice there is a significant uncertainty in the actual selection of a meaningful value for the error tolerance ρ, a value that must be satisfied. Furthermore, numerical analysts are also subconsciously aware of this fact of life, even though in most numerical analysis papers such a value is simply given, if at all, in the numerical examples section. Three typical yet different classes of problems and methods are considered in Section 8.2.

Once we are all convinced that there is usually a considerable uncertainty in the value of ρ (hence, we only know it "probably"), the next question is what to do with this notion. The answer varies, depending on the particular application and the situation at hand. In some cases, such as that of Section 8.2.1, the effective advice is to be more cautious, as mishaps can happen. In others, such as that of Section 8.2.2, we are simply led to acknowledge that the value of ρ may come from thin air (though one then concentrates on other aspects). But there are yet other classes of applications and algorithms, such as in Section 8.2.3, for which it makes sense to attempt to quantify the uncertainty in the error tolerance ρ using a probabilistic framework. We are not proposing here to propagate an entire probability distribution for ρ: that would be excessive in most situations. But we do show, by studying an instance extended to a wide class of problems, that employing such a framework can be practical and profitable.

Following Section 8.2.3 we therefore consider in Section 8.3 a particular manner of relaxing the notion of a deterministic error tolerance, introduced in Chapter 7, by allowing an estimate such as (8.1) to hold only within some given probability. Some numerical examples are given to illustrate these ideas. Conclusions and some additional general comments are offered in Section 8.4.

8.2 Case Studies

In this section we consider three classes of problems and associated algorithms, in an attempt to highlight the use of different tests of the form (8.1) and in particular the implied level of uncertainty in the choice of ρ.

8.2.1 Stopping Criterion in Initial Value ODE Solvers

Using a case study, we show in this section that numerical analysts, too, can be quick to not consider ρ as a "holy constant": we adapt to weaker conditions in different ways, depending on the situation and the advantage to be gained in relaxing the notion of an error tolerance.

Let us consider an initial value ODE system in "time" t, written as

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}), \quad 0 \le t \le b, \tag{8.2a}$$

$$\mathbf{u}(0) = \mathbf{v}_0, \tag{8.2b}$$

with $\mathbf{v}_0$ a given initial value vector. A typical adaptive algorithm proceeds to generate pairs $(t_i, \mathbf{v}_i),\ i = 0, 1, 2, \ldots, N$, in N consecutive steps, thus forming a mesh π such that

$$\pi : 0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = b,$$

and $\mathbf{v}_i \approx \mathbf{u}(t_i),\ i = 1, \ldots, N$.

Denoting the numerical solution on the mesh π by $\mathbf{v}^\pi$, and the restriction of the exact ODE solution to this mesh by $\mathbf{u}^\pi$, there are two general approaches for controlling the error in such an approximation.

- Given a tolerance value ρ, keep estimating the *global error* and refining the mesh (i.e., the gamut of step sizes) until roughly

$$\|\mathbf{v}^\pi - \mathbf{u}^\pi\|_\infty \le \rho. \tag{8.3}$$

  Details of such methods can be found, for instance, in [17, 37, 74, 82].

  In (8.3) we could replace the absolute tolerance by a combination of absolute and relative tolerances, perhaps even different ones for different ODE equations. But that aspect is not what we concentrate on in this chapter.

- However, most general-purpose ODE codes estimate a *local error* measure for (8.1) instead, and refine the step size locally. Such a procedure advances one step at a time, and estimates the next step size using local information related to the local truncation error,

or simply the difference between two approximate solutions for the next time level, one of which presumed to be significantly more accurate than the other.[9] For details see [17, 74, 75] and many references therein. In particular, the popular MATLAB codes `ode45` and `ode23s` use such a local error control.

The reason for employing local error control is that this allows for developing a much cheaper and yet more sensitive adaptive procedure, an advantage that cannot be had, for instance, for general boundary value ODE problems; see, e.g., [16].

But *does this always produce sensible results?!* The answer to this question is negative. A simple example to the contrary is the problem

$$\frac{du}{dt} = 100(u - \sin t) + \cos t, \quad u(0) = 0,\ b = 1.$$

Local truncation (or discretization) errors for this unstable initial value ODE propagate like $\exp(100t)$, a fact that is not reflected in the local behaviour of the exact solution $u(t) = \sin t$ on which the local error control is based. Thus, we may have a large error $\|\mathbf{v}^\pi - \mathbf{u}^\pi\|_\infty$ even if the local error estimate is bounded by ρ for a small value of ρ.

Local error control can be dangerous even for a stable ODE system

Still one can ask, *are we safe with local error control in case that we know that our ODE problem is stable?* Here, by "safe" we mean that the global error will not be much larger than the local truncation error in scaled form. The answer to this more subtle question turns out to be negative as well. The essential point is that the global error consists of an accumulation of contributions of local errors from previous time steps. If the ODE problem is asymptotically stable (typically, because it describes a damped motion) then local error contributions die away as time increases, often exponentially fast, so at some fixed time only the most recent local error contributions dominate in the sum of contributions that forms the global error. However, if the initial value ODE problem is merely marginally stable (which is the case for Hamiltonian

[9] Recall that the local truncation error at some time $t = t_i$ is the amount by which the exact solution $\mathbf{u}^\pi$ fails to satisfy the scheme that defines $\mathbf{v}^\pi$ at this point. Furthermore, if at t_i, using the known $\mathbf{v}_i$ and a guess for t_{i+1}, we apply one step of two different Runge-Kutta methods of orders 4 and 5, say, then the difference of the two results at t_{i+1} gives an estimate for the error in the lower order method over this mesh subinterval.

systems) then local error contributions propagate undamped, and their accumulation over many time steps can therefore be significantly larger than just one or a few such errors.[10]

For a simple concrete example, consider applying `ode45` with default tolerances to find the linear oscillator with a slowly varying frequency that satisfies the following initial value ODE for $p(t)$:

$$\begin{aligned} \frac{dq}{dt} &= \lambda^2 p, \quad q(0) = 1, \\ \frac{dp}{dt} &= -(1+t)^2 q, \quad p(0) = 0. \end{aligned}$$

Here $\lambda > 0$ is a given parameter. Thus, $\mathbf{u} = (q, p)^T$ in the notation of (8.2). This is a Hamiltonian system, with the Hamiltonian function given by

$$H(q, p, t) = \frac{1}{2}\left[((1+t)q)^2 + (\lambda p)^2\right].$$

Now, since the ODE is not autonomous, the Hamiltonian is not constant in time. However, the *adiabatic invariant*

$$J(q, p, t) = H(q, p, t)/(1+t)$$

(see, e.g., [18, 97]) is almost constant for large λ, satisfying

$$[J(t) - J(0)]/J(0) = \mathcal{O}(\lambda^{-1})$$

over the interval $[0, 1]$. This condition means in particular that for $\lambda \gg 1$ and the initial values given above, $J(1) = J(0) + \mathcal{O}(\lambda^{-1}) \approx J(0)$.

Figure 8.1 depicts two curves approximating the adiabatic invariant for $\lambda = 1000$. Displayed are the calculated curve using `ode45` with default tolerances (absolute=1.e-6, relative=1.e-3), as well as what is obtained upon using `ode45` with the stricter relative tolerance RelTol=1.e-6. From the figure it is clear that when using the looser tolerance, the resulting approximation for $J(1)$ differs from $J(0)$ by far more than what λ^{-1} =1.e-3 and RelTol=1.e-3 would indicate, while the stricter tolerance gives a qualitatively correct result, using the "eye norm". Annoyingly,

[10] The local error control basically seeks to equalize the magnitude of such local errors at different time steps.

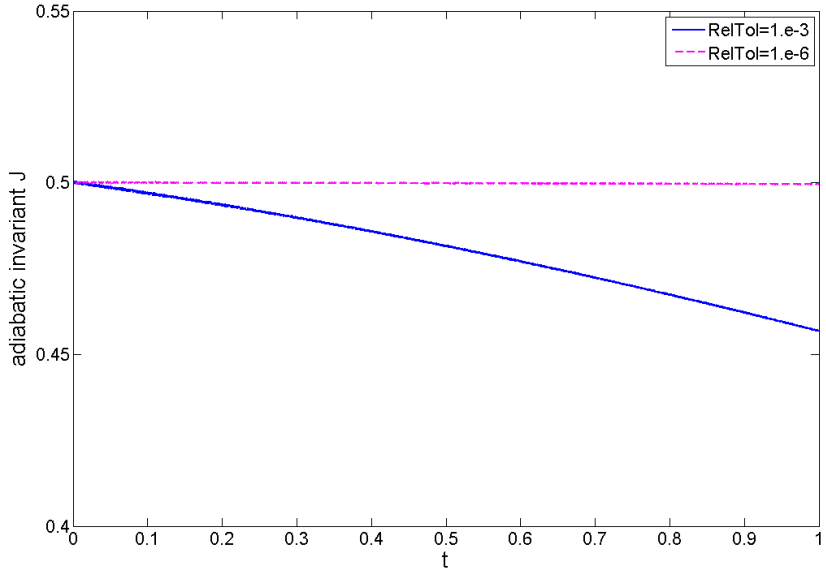


Figure 8.1: Adiabatic invariant approximations obtained using MATLAB's package `ode45` with default tolerances (solid blue) and stricter tolerances (dashed magenta).

the qualitatively incorrect result does not look like "noise": while not being physical, it looks downright plausible, and hence could be misleading for an unsuspecting user. Adding to the pain is the fact that this occurs for default tolerance values, an option that a vast majority of users would automatically select. ♦

A similar observation holds when trying to approximate the phase portrait or other properties of an autonomous Hamiltonian ODE system over a long time interval using `ode45` with default tolerances: this may produce qualitatively wrong results. See for instance Figures 16.12 and 16.13 in [14]: the Fermi-Pasta-Ulam problem solved there is described in detail in Chapter 1 of [73]. What we have just shown here is that the phenomenon can arise also for a very modest system of *two linear ODEs that do not satisfy any exact invariant.*

We hasten to add that the documentation of `ode45` (or other such codes) does not propose to deliver anything like (8.3). Rather, the tolerance is just a sort of a knob that is turned to control local error size. However, this does not explain the popularity of such codes despite their limited offers of assurance in terms of qualitatively correct results.

Our key point in the present section is the following: we propose that one reason for the popularity of ODE codes that use only local error control is that in applications one rarely

knows a precise value for ρ as used in (8.3) anyway. (Conversely, if such a global error tolerance value is known and is important then codes employing a global error control, and not `ode45`, should be used.) Opting for local error control over global error control can therefore be seen as one specific way of adjusting mathematical software in a deterministic sense to realistic uncertainties regarding the desired accuracy.

8.2.2 Stopping Criterion in Iterative Methods for Linear Systems

In this case study, extending basic textbook material, we argue not only that tolerance values used by numerical analysts are often determined solely for the purpose of the comparison of methods (rather than arising from an actual application), but also that this can have unexpected effects on such comparisons.

Consider the problem of finding $\mathbf{u}$ satisfying

$$A\mathbf{u} = \mathbf{b}, \tag{8.4}$$

where A is a given $s \times s$ symmetric positive definite matrix such that one can efficiently carry out matrix-vector products $A\mathbf{v}$ for any suitable vector $\mathbf{v}$, but decomposing the matrix directly (and occasionally, even looking at its elements) is too inefficient and as such is "prohibited". We relate to such a matrix as being given *implicitly*. The right hand side vector $\mathbf{b}$ is given as well.

An iterative method for solving (8.4) generates a sequence of iterates $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k, \ldots$ for a given initial guess $\mathbf{u}_0$. Denote by $\mathbf{r}_k = \mathbf{b} - A\mathbf{u}_k$ the residual in the kth iterate. The MINRES method, or its simpler version Orthomin(2), can be applied to reduce the residual norm so that

$$\|\mathbf{r}_k\|_2 \leq \rho \|\mathbf{r}_0\|_2 \tag{8.5}$$

in a number of iterations k that in general is at worst $\mathcal{O}\left(\sqrt{\kappa(A)}\right)$, where $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$ is the condition number of the matrix A. Below in Table 8.1 we refer to this method as MR. The more popular conjugate gradient (CG) method generally performs comparably in practice. We refer to [67] for the precise statements of convergence bounds and their proofs.

A well-known and simpler-looking family of *gradient descent* methods is given by

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{r}_k, \tag{8.6}$$

where the scalar $\alpha_k > 0$ is the step size. Such methods have recently come under intense scrutiny because of applications in stochastic programming and sparse solution recovery. Thus, it makes sense to evaluate and understand them in the simplest context of (8.4), even though it is commonly agreed that for the strict purpose of solving (8.4) iteratively, CG cannot be significantly beaten. Note that (8.6) can be viewed as forward Euler for the artificial time ODE

$$\frac{d\mathbf{u}}{dt} = -A\mathbf{u} + \mathbf{b}, \tag{8.7}$$

with "time" step size α_k. Next we consider two choices of this step size.

The steepest descent (SD) variant of (8.6) is obtained by the greedy (exact) line search for the function

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T A\mathbf{u} - \mathbf{b}^T\mathbf{u},$$

which gives

$$\alpha_k = \alpha_k^{SD} = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{r}_k^T A\mathbf{r}_k} \equiv \frac{(\mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{r}_k, A\mathbf{r}_k)} \equiv \frac{\|\mathbf{r}_k\|_2^2}{\|\mathbf{r}_k\|_A^2}.$$

However, SD is very slow, requiring k in (8.5) to be proportional to $\kappa(A)$; see, e.g., [3].[11]

A more enigmatic choice in (8.6) is the lagged steepest descent (LSD) step size

$$\alpha_k = \alpha_k^{LSD} = \frac{(\mathbf{r}_{k-1}, \mathbf{r}_{k-1})}{(\mathbf{r}_{k-1}, A\mathbf{r}_{k-1})}.$$

It was first proposed in [24] and practically used for instance in [27, 42]. To the best of our knowledge, there is no known a priori bound on how many iterations as a function of $\kappa(A)$ are

[11] The precise statement of error bounds for CG and SD in terms of the error $\mathbf{e}_k = \mathbf{u} - \mathbf{u}_k$ uses the A-norm, or "energy norm", and reads

$$\begin{aligned} \|\mathbf{e}_k\|_A &\leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^k \|\mathbf{e}_0\|_A, \quad \text{for CG,} \\ \|\mathbf{e}_k\|_A &\leq \left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^k \|\mathbf{e}_0\|_A, \quad \text{for SD.} \end{aligned}$$

See [67].

required to satisfy (8.5) with this method [24, 45, 59, 112].

We next compare these four methods in a typical fashion for a typical PDE example, where we consider the model Poisson problem

$$-\Delta u = 1, \quad 0 < x, y < 1,$$

subject to homogeneous Dirichlet BC, and discretized by the usual 5-point difference scheme on a $\sqrt{s} \times \sqrt{s}$ uniform mesh. Denote the reshaped vector of mesh unknowns by $\mathbf{u} \in \mathbb{R}^s$. The largest eigenvalue of the resulting matrix A in (8.4) is $\lambda_{\max} = 4h^{-2}(1 + \cos(\pi h))$, and the smallest is $\lambda_{\min} = 4h^{-2}(1 - \cos(\pi h))$, where $h = 1/(\sqrt{s} + 1)$. Hence by Taylor expansion of $\cos(\pi h)$, for $h \ll 1$ the condition number is essentially proportional to s:

$$\kappa(A) = \frac{\lambda_{\max}}{\lambda_{\min}} \approx \left(\frac{2}{\pi}\right)^2 s.$$

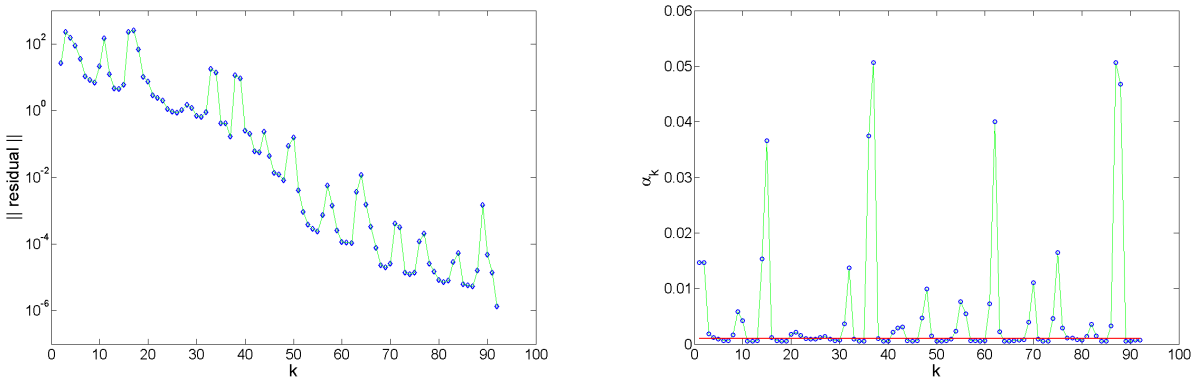
In Table 8.1 we list iteration counts required to satisfy (8.5) with $\rho = 10^{-7}$, starting with $\mathbf{u}_0 = \mathbf{0}$.

s	MR	CG	SD	LSD
7^2	9	9	196	45
15^2	26	26	820	91
31^2	54	55	3,337	261
63^2	107	109	13,427	632
127^2	212	216	53,800	1,249

Table 8.1: Iteration counts required to satisfy (8.5) for the Poisson problem with tolerance $\rho = 10^{-7}$ and different mesh sizes s.

But now, returning to the topic of the present chapter, we ask, *why insist on* $\rho = 10^{-7}$*?* Indeed, the usual observation that one draws from the columns of values for MR, CG and SD in a table such as Table 8.1, is that the first two grow like $\sqrt{\kappa(A)} \propto \sqrt{s}$ while the latter grows like $\kappa(A) \propto s$. The value of ρ, so long as it is not too large, does not matter at all!

And yet, this is not quite the case for the LSD iteration counts. These do not decrease in the same orderly fashion as the others, even though they are far better (in the sense of being significantly smaller) than those for SD. Indeed, this method is chaotic [45], and the residual



(a) Residuals

(b) Step sizes

Figure 8.2: Relative residuals and step sizes for solving the model Poisson problem using LSD on a 15×15 mesh. The red line in (b) is the forward Euler stability limit.

norm decreases rather non-monotonically, see Figure 8.2(a). Thus, the iteration counts in Table 8.1 correspond to the iteration number $k = k^*$ where the rough-looking relative residual norm first records a value below the tolerance ρ. Unlike the other three methods, here the particular value of the tolerance, picked out of nowhere, does play an unwanted role in the relative values, as a function of s, or $\kappa(A)$, of the listed iteration counts.

8.2.3 Data Fitting and Inverse Problems

In the previous two case studies we have encountered cases where the intuitive use of an error tolerance within a stopping criterion could differ widely (and wildly) from the notion that is embodied in (8.1) for the consumer of numerical analysts' products. We next consider a family of problems where the value of ρ in a particular criterion (8.1) is more directly relevant.

Suppose we are given observed data $\mathbf{d} \in \mathbb{R}^l$ and a *forward operator* $f_i(m)$, $i = 1, \ldots, l$, which provides predicted data for each instance of a distributed parameter function m. The (unknown) function m is defined in some domain Ω in physical space and possibly time. We are particularly interested here in problems where f involves the solution u in Ω of some linear PDE system, sampled in some way at the points where the observed data are provided; see Section 1.1.2. Further, for a given mesh π discretizing Ω, we consider a corresponding discretization (i.e., nodal representation) of m and u, as well as the differential operator. Reshaping these mesh functions into vectors we can write the resulting approximation of the forward operator as (1.5),

namely

$$\mathbf{f}(\mathbf{m}, \mathbf{q}) = P\mathbf{u} = PL^{-1}(\mathbf{m})\mathbf{q}, \tag{8.8}$$

where the right hand side vector $\mathbf{q}$ is commonly referred to as a source, L is a square matrix discretizing the PDE operator plus appropriate side conditions, $\mathbf{u} = L^{-1}(\mathbf{m})\mathbf{q}$ is the field (i.e., the PDE solution, here an interim quantity), and P is a projection matrix that projects the field to the locations where the data values $\mathbf{d}$ are given.

This setup is typical in the thriving research area of inverse problems; see, e.g., [52, 135]. A specific example is provided in Section 8.3.

The inverse problem is to find $\mathbf{m}$ such that the predicted and observed data agree to within noise $\boldsymbol{\eta}$: ideally,

$$\mathbf{d} = \mathbf{f}(\mathbf{m}, \mathbf{q}) + \boldsymbol{\eta}. \tag{8.9}$$

To obtain such a model $\mathbf{m}$ that satisfies (8.9) we need to estimate the *misfit function* $\phi(\mathbf{m})$, i.e., the normed difference between observed data $\mathbf{d}$ and predicted data $\mathbf{f}(\mathbf{m})$. An iterative algorithm is then designed to sufficiently reduce this misfit function. But, *which norm should we use to define the misfit function?*

It is customary to conveniently assume that the noise satisfies $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma I)$, i.e., that the noise is normally distributed with a scaled identity for the covariance matrix, where σ is the standard deviation. Then the maximum likelihood (ML) data misfit function is simply the squared ℓ_2-norm[12]

$$\phi(\mathbf{m}) = \|\mathbf{f}(\mathbf{m}) - \mathbf{d}\|_2^2. \tag{8.10}$$

In this case, the celebrated Morozov *discrepancy principle* yields the stopping criterion

$$\phi(\mathbf{m}) \leq \rho, \quad \text{where } \rho = \sigma^2 l, \tag{8.11}$$

see, e.g., [52, 91, 107]. So, here is a class of problems where we do have a meaningful and

[12] For a more general symmetric positive definite covariance matrix Σ, such that $\boldsymbol{\eta} \sim \mathcal{N}(0, \Sigma)$, we get weighted least squares, or an "energy norm", with the weight matrix Σ^{-1} for ϕ. But let's not go there in this chapter.

directly usable tolerance value!

Assuming that a known tolerance ρ must be satisfied as in (8.11) is often too rigid in practice, because realistic data do not quite satisfy the assumptions that have led to (8.11) and (8.10). Well-known techniques such as L-curve and GCV (see, e.g., [65, 79, 135]) are specifically designed to handle more general and practical cases where (8.11) cannot be used or justified. Also, if (8.11) is used then a typical algorithm would try to find $\mathbf{m}$ such that $\phi(\mathbf{m})$ is (smaller but) not much smaller than ρ, because having $\phi(\mathbf{m})$ too small would correspond to fitting the noise – an effect one wants to avoid. The latter argument and practice do not follow from (8.11).

Moreover, keeping the misfit function $\phi(\mathbf{m})$ in check does not necessarily imply a quality reconstruction (i.e., an acceptable approximation $\mathbf{m}$ for the "true solution" $\mathbf{m}^*$, which can be an elusive notion in itself). However, $\phi(\mathbf{m})$, and not direct approximations of $\|\mathbf{m}^* - \mathbf{m}\|$, is what one typically has to work with.[13] So any additional a priori information is often incorporated through some regularization.

Still, despite all the cautious comments in the preceding two paragraphs, we have in (8.11) in a sense a more meaningful practical expression for stopping an iterative algorithm than hitherto.

Typically there is a need to regularize the inverse problem, and often this is done by adding a regularization term to (8.10). Thus, one attempts to *approximately* solve the Tikhonov-type problem

$$\min_{\mathbf{m}} \ \phi(\mathbf{m}) + \alpha R(\mathbf{m}),$$

where $R(\mathbf{m}) \geq 0$ is a prior (we are thinking of some norm or semi-norm of $\mathbf{m}$), and $\alpha \geq 0$ is a regularization parameter.

A fourth case study is the one that this thesis concentrates on, namely, the extension of Case Study 8.2.3 to problems with many data sets to which the additional approximation using Monte-Carlo sampling is applied. Of course, our uncertainty in the error criterion and

[13] The situation here is different from that in Section 8.2.1, where the choice of local error criterion over a global one was made based on convenience and efficiency considerations. Here, although controlling $\phi(\mathbf{m})$ is merely a necessary and not sufficient condition for obtaining a quality reconstruction $\mathbf{m}$, it is usually all we have to work with.

specifically the error tolerance, if anything, increases even further here. On the other hand, unlike in the previous case studies where we only call for increased alertness and additional caution regarding the error tolerance, here we have the framework to quantify uncertainty and as such we can obtain more efficient algorithms for problems with more such uncertainty. Satisfying the tolerance only probably thus leads to cheaper computations in a disciplined manner.

8.3 Probabilistic Relaxation of a Stopping Criterion

The previous section details three different case studies which highlight the fact of life that in applications an error tolerance for stopping an algorithm is rarely known with absolute certainly. Thus, we can say that such a tolerance is only "probably" known. Yet in some situations, it is also possible to assign it a more precise meaning in terms of statistical probability. This holds true for the problems considered in this thesis, namely extensions of Case Study 8.2.3 to problems with many data. Thus, one can consider a way to relax (8.1), which is more systematic and also allows for further theoretical developments. Specifically, we consider satisfying a tolerance in a probabilistic sense, as proposed in Section 7.2.2.

Thus, according to (7.2), in the check for termination of our iterative algorithm at the next iterate $\mathbf{m}_{k+1}$, we consider replacing the condition (3.3), namely

$$\phi(\mathbf{m}_{k+1}) \leq \rho$$

by either (7.21a) or (7.21b), namely

$$\begin{aligned} \widehat{\phi}(\mathbf{m}_{k+1}, n_t) &\leq (1-\varepsilon)\rho, \quad \text{or} \\ \widehat{\phi}(\mathbf{m}_{k+1}, n_t) &\leq (1+\varepsilon)\rho, \end{aligned}$$

for a suitable $n = n_t$ that is governed by Theorems 7.1 or 7.2 with a prescribed pair (ε, δ). If (7.21a) holds, then it follows with a probability of at least $(1-\delta)$ that (3.3) holds. On the other hand, if (7.21b) does *not* hold, then we can conclude with a probability of at least $(1-\delta)$ that (3.3) is *not* satisfied. In other words, unlike (7.21a), a successful (7.21b) is only necessary

and not sufficient for concluding that (3.3) holds with the prescribed probability $1 - \delta$.

What are the connections among these three parameters, ρ, δ and ε?! The parameter ρ is the deterministic but not necessarily too trustworthy error tolerance appearing in (3.3), much like the tolerance in Section 8.2.1. Next, we can reflect the uncertainty in the value of ρ by choosing an appropriately large δ (≤ 1). Smaller values of δ reflect a higher certainty in ρ and a more rigid stopping criterion (translating into using a larger n_t). For instance, success of (7.21a) is equivalent to making a statement on the probability that a positive "test" result will be a "true" positive. This is formally given by the conditional probability statement

$$Pr\left(\phi(\mathbf{m}_{k+1}) \leq \rho \mid \widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1-\varepsilon)\rho\right) \geq 1 - \delta.$$

Note that, once the condition in this statement is given, the rest only involves ρ and δ. So the tolerance ρ is augmented by the probability parameter δ. The third parameter ε governs the false positives/negatives (i.e., the probability that the test will yield a positive/negative result, if in fact (3.3) is false/true), where a false positive is given by

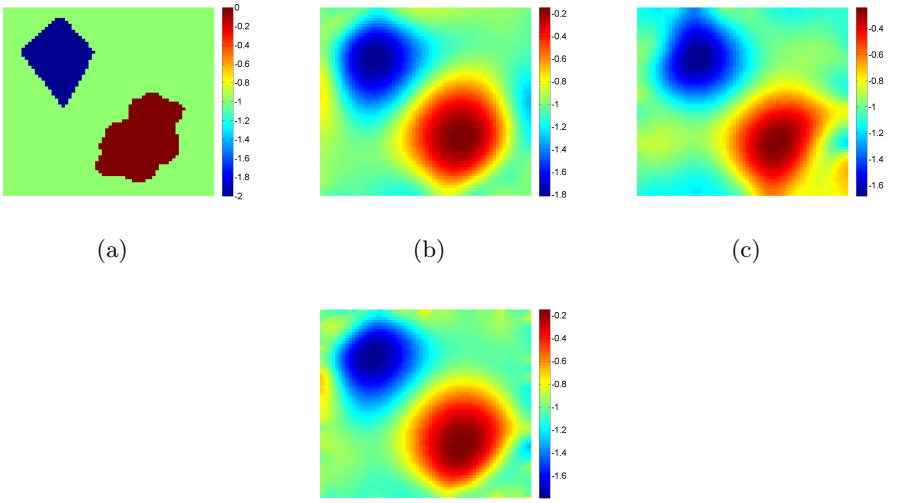
$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_t) \leq (1-\varepsilon)\rho \mid \phi(\mathbf{m}_{k+1}) > \rho\right),$$

while a false negative is

$$Pr\left(\widehat{\phi}(\mathbf{m}_{k+1}, n_t) > (1-\varepsilon)\rho \mid \phi(\mathbf{m}_{k+1}) \leq \rho\right).$$

Such probabilistic stopping criterion is incorporated in Algorithm 4 in Chapter 7 and, there, various numerical examples are given to illustrate these ideas on a concrete application. Here, employing Algorithm 4 again, we give some more examples with the same setup as that of Example (E.2) in Chapter (7), but instead of TV, we use dynamical regularization. Note again that the large dynamical range of the conductivities, together with the fact that the data is available only on less than half of the boundary, contribute to the difficulty in obtaining good quality reconstructions. The term "Vanilla" refers to using all s available data sets for each task during the algorithm. This costs 527,877 PDE solves[14] for $s = 3,969$ (b) and 5,733 PDE

[14]Fortunately, the matrix L does not depend on i in (1.4). Hence, if the problem is small enough that a direct



(a) (b) (c)

(d)

Figure 8.3: Plots of log-conductivity: (a) True model; (b) Vanilla recovery with $s = 3,969$; (c) Vanilla recovery with $s = 49$; (d) Monte Carlo recovery with $s = 3,969$. The vanilla recovery using only 49 measurement sets is clearly inferior, showing that a large number of measurement sets can be crucial for better reconstructions. The recovery using our algorithm, however, is comparable in quality to Vanilla with the same s. The quantifier values used in our algorithm were: $(\varepsilon_c, \delta_c) = (0.05, 0.3)$, $(\varepsilon_u, \delta_u) = (0.1, 0.3)$ and $(\varepsilon_t, \delta_t) = (0.1, 0.1)$.

solves for $s = 49$ (c). However, the quality of reconstruction using the smaller number of data sets is clearly inferior. On the other hand, using our algorithm yields a recovery (d) that is comparable to Vanilla but at the cost of only 5,142 PDE solves. The latter cost is about 1% that of Vanilla and is comparable in order of magnitude to that of evaluating $\phi(\mathbf{m})$ once!

8.3.1 TV and Stochastic Methods

This section is not directly related to the main theme of this chapter, but it arises from the present discussion and should have merit on its own (in addition to being mercifully short).

The specific example considered above is used also in Chapter 7, except that the objective function there includes a total variation (TV) regularization. This represents usage of additional a priori information (namely, that the true model is discontinuous with otherwise flat regions), whereas here an implicit ℓ_2-based regularization has been employed without such knowledge regarding the true solution. The results in Figures 7.6(b) and 7.7(vi) there correspond to

method can be used to construct G, i.e., perform one LU decomposition at each iteration k, then the task of solving half a million PDEs just for comparison sake becomes less daunting.

our Figures 8.3(b) and 8.3(d), respectively, and as expected, they look sharper in Chapter 7. On the other hand, a comparative glance at Figure 7.6(c) there vs the present Figure 8.3(c) reveals that the ℓ_1-based technique can be inferior to the ℓ_2-based one, even for recovering a piecewise constant solution! Essentially, even for this special solution form TV shines only with sufficiently good data, and here "sufficiently good" translates to "many data sets". This intuitively obvious observation does not appear to be as well-known today as it used to be [47].

8.4 Conclusions

Mathematical software packages typically offer a default option for the error tolerances used in their implementation. Users often select this default option without much further thinking, at times almost automatically. This in itself suggests that practical occasions where the practitioner does not really have a good hold of a precise tolerance value are abundant. However, since it is often convenient to assume having such a value, and convenience may breed complacency, surprises may arise. We have considered in Section 8.2 three case studies which highlight various aspects of this uncertainty in a tolerance value for a stopping criterion.

Recognizing that there can often be a significant uncertainty regarding the actual tolerance value and the stopping criterion, we have subsequently considered the relaxation of the setting into a probabilistic one, and demonstrated its benefit in the context of large scale problems considered in this thesis. The environment defined by probabilistic relative accuracy, such as (7.2), although well-known in other research areas, is relatively new (but not entirely untried) in the numerical analysis community. It allows, among other benefits, specifying an amount of trust in a given tolerance using two parameters that can be tuned, as well as the development of bounds on the sample size of certain Monte Carlo methods. In Section 8.3, following Chapter 7, we have applied this setting in the context of a particular inverse problem involving the solution of many PDEs, and we have obtained some uncertainty quantification for a rather efficient algorithm solving a large scale problem.

There are several aspects of our topic that remain untouched in this chapter. For instance, there is no discussion of the varying nature of the error quantity that is being measured (which strongly differs across the subsections of Section 8.2, from solution error through residual error

through data misfit error for an ill-posed problem to stochastic quantities that relate even less closely to the solution error). Also, we have not mentioned that complex algorithms often involve sub-tasks such as solving a linear system of equations iteratively, or employing generalized cross validation (GCV) to obtain a tolerance value, or invoking some nonlinear optimization routine, which themselves require some stopping criterion: thus, several occurrences of tolerances in one solution algorithm are common. In the probabilistic sections, we have made the choice of concentrating on bounding the sample size n and not, for example, on minimizing the variance as in [86].

What we have done here is to highlight an often ignored yet rather fundamental issue from different angles. Subsequently, we have pointed at and demonstrated a promising approach (or direction of thought) that is not currently common in the scientific computing community.

Chapter 9

Summary and Future Work

Efficiently solving large scale non-linear inverse problems of the form described in Section 1.1 is indeed a challenging problem. Large scale, within the context we aimed to study in this thesis, implies that we are given a very large number of measurement vectors, i.e., $s \gg 1$. For many instances of such problems, there are theoretical reasons for requiring large amounts of data for obtaining any credible reconstruction. For many others, it is an accepted working assumption that having more data can only help and not hurt the conditioning of the problem being solved. As such, methods for efficiently solving such problems are highly sought after in practice. In this thesis, we have proposed highly efficient randomized reconstruction algorithms for solving such problems. we have also demonstrated both the efficacy and the efficiency of the proposed algorithms in the context of an important class of such problems, namely PDE inverse problems with many measurements. As a specific instance, we used the famous and notoriously difficult DC resistivity problem.

Each chapter of this thesis contains conclusions and future research directions specic to that particular line of research; here we present an overall summary and some more topics for future research, not mentioned earlier.

9.1 Summary

In Chapter 2, various dimensionality reduction (i.e., approximation) methods were presented to deal with computational challenges arising from evaluating the misfit (1.6). All these methods consist of sampling the large dimensional data and creating a new set of lower dimensional data for which computations can be done more efficiently. Such sampling can be done either (i) stochastically or (ii) deterministically. We showed that stochastic sampling results in an unbiased estimator of the original misfit (1.6), and as such, the misfit itself is approximated. Main

examples of appropriate distributions for stochastic estimations were discussed: Rademacher, Gaussian and Random Subset. In cases where the underlying forward operators satisfy Assumptions (A.1) - (A.3), we showed that the stochastic methods using the Rademacher or Gaussian distribution result in the method of simultaneous sources (SS) and indeed yield very efficient approximations (recall our definition of efficiency, in footnotes 4 and 6, in Chapter 2). In situations where Assumption (A.2) is violated, however, the method of random subset (RS) is the only applicable estimator. On the other hand, our proposed method for deterministic sampling is based on TSVD approximation of the matrix consisting of all measurement vectors. As such, unlike the stochastic methods which approximate the misfit, the TSVD approach approximates the data matrix, D in (1.6). If Assumptions (A.1) - (A.3) hold, such deterministic method, similar to stochastic ones, yields another instance of SS methods.

In Chapter 3, continuing to make Assumptions (A.1) - (A.3), we developed and compared several highly efficient stochastic iterative reconstruction algorithms for approximately solving the (regularized) NLS formulation of aforementioned large scale data fitting problems. All these iterative algorithms involve employing the dimensionality reduction techniques discussed in Chapter 2. As such, at iteration k of our algorithms, the original s measurement vectors are sampled and a new, yet smaller, set of n_k measurement vectors with $n_k \ll s$ are formed. Two reconstruction algorithms for controlling the size n_k of the data set in the k^{th} iteration have been proposed and tested. We identified and justified three different purposes for such the dimensionality reduction methods within various steps of our proposed algorithms, namely fitting, cross validation and uncertainty check. Using the four methods of sampling the data (i.e., three stochastic and one deterministic introduced in Chapter 2), our two algorithms make for a total of eight algorithm variants. All of our algorithms are known to converge under suitable circumstances because they satisfy the general conditions in [36, 60].

Chapter 4 is a sequel to Chapter 3 in which we relax the linearity assumption (A.2) and propose methods that, where applicable, transform the problem into one where the linearity assumption (A.2) is restored. Hence, efficient dimensionality reduction methods introduced in Chapter 2 can be applied. In Chapter 4, we focus on a particular case where the linearity assumption (A.2) is violated due to missing or corrupted data. Such situations arise often in practice, and hence, it is desired to have methods to be able to apply variants of the efficient

reconstruction algorithms presented in Chapter 3. The transformation methods presented in Chapter 4 involve completion/replacement of the missing/corrupted portion of the data. These methods are presented in the context of EIT/DC resistivity as an important class of PDE inverse problems; however, we believe that similar ideas can be applied in many more instances of such problems. Our data completion/replacement methods are motivated by theory in Sobolev spaces, regarding the properties of weak solutions along the domain boundary. Our methods prove to be capable of effectively reconstructing the data in the presence of large amounts of missing or corrupted data. Variants of efficient reconstruction algorithms, presented in Chapter 3, are proposed and numerically verified. In addition to completion/replacement methods, a heuristic and efficient alternative to the rigid stopping criterion (3.3) is given.

All of our proposed randomized dimensionality reduction methods rely heavily upon the fundamental concept of estimating the trace of an implicit symmetric positive semi-definite matrices using Monte Carlo methods. As such the question of accuracy and efficiency of our stochastic approximation methods are tied with those of such trace estimators. In Chapter 5 this task is visited, and accuracy and efficiency of the randomized trace estimators are analyzed using a suitable and intuitive probabilistic framework. Under such probabilistic framework, one seeks conditions on the sample size required for these Monte-Carlo methods to probabilistically guarantee estimate's desired relative accuracy. In this chapter, conditions for all the distributions discussed in Section 2.1.1 are derived. In addition to practically computable conditions, we also provide some uncomputable, yet informative, conditions which shed light on questions regarding the type of matrices a particular distribution is best/least suited for. Part of the theory presented in Chapter 5 is, subsequently, further improved in Chapter 7.

Chapter 6 is a precursor of Chapter 7. Specifically, the improvements in theoretical studies of MC trace estimators presented in Chapter 7 are applications of more general results regarding the extremal probabilities (i.e., maxima and minima of the tail probabilities) of non-negative linear combinations (i.e., convolutions) of gamma random variables, which are proved in Chapter 6. In addition, in Chapter 6, we prove results regarding the monotonicity of the regularized gamma function. All these results are very general and have many applications in economics, actuarial science, insurance, reliability and engineering.

The main advantage of any efficient (randomized) approximation algorithm is the reduction

of computational costs. However, a major drawback of any such algorithm is the introduction of "uncertainty" in the overall procedure. The presence of uncertainty in the approximation steps could cast doubt on the credibility of the obtained results. Hence, it may be useful to have means which allow one to adjust the cost and accuracy of such algorithms in a quantifiable way, and find a balance that is suitable to particular objectives and computational resources. Chapter 7 offers the uncertainty quantification of the major stochastic steps of our reconstruction algorithms presented in Chapters 3 and 4. Such steps include the fitting, uncertainty check, cross validation and stopping criterion. This results in highly efficient variants of our original algorithms where the degree of uncertainty can easily be quantified and adjusted, if needed. Using the resulting algorithm, one could, in a quantifiable way, obtain a desirable balance between the amount of uncertainty and the computational complexity of the reconstruction algorithm. In order to achieve this, we make use of similar probabilistic analysis as in Chapter 5. However, the conditions presented in Chapter 5 are typically not sufficiently tight to be useful in many practical situations. In Chapter 7, using the results of Chapter 6, we improve upon some of the theory presented in Chapter 5. Specifically, in Chapter 7, we prove tight bounds for tail probabilities for such Monte-Carlo approximations employing the standard normal distribution. These tail bounds are then used to obtain necessary and sufficient bounds on the required sample size, and we demonstrate that these bounds can be practically small and computable. Numerical examples demonstrate the efficiency of our proposed uncertainty-quantified algorithm.

Numerical algorithms are typically equipped with a stopping criterion where the calculation is terminated when some error or misfit measure is deemed to be below a given tolerance. However, in practice such a tolerance is rarely known; rather, only some possibly vague idea of the desired quality of the numerical approximation is available. In Chapter 8, we discuss several case studies, from different areas of numerical analysis, where a rigid interpretation of error criterion and tolerance may result in erroneous outcomes and conclusions. We discuss, for instance, fast codes for initial value ODEs and DAEs, which heavily rely on the underlying philosophy that satisfying a tolerance for the global error is too rigid a task; such codes proceed to control just the local error. Another instance of soft error control is when a large, complicated model for fitting data is reduced, say by a Monte Carlo sampling method as in previous chapters. If calculating the misfit norm is in itself very expensive then the option of satisfying the stopping

criterion only in a probabilistic sense naturally arises. This leads one to think of devising approaches, where they are possible, to relax the notion of exactly satisfying a tolerance value. In Chapter 8, we discuss this in the context of large scale PDE inverse problems described in Section 1.1.2. Such probabilistic relaxation of the given tolerance in this context, allows, for instance, for the use of the proven bounds in Chapters 5 and 7.

This thesis also includes an appendix. In Appendix A, certain implementation details are given which are used throughout the thesis. Such details include discretization of the EIT/DC resistivity problem in two and three dimensions, injection of a priori knowledge on the sought parameter function via transformation functions in the original PDE, an overall discussion of the (stabilized) Gauss-Newton algorithm for minimization of a least squares objective, a short MATLAB code which is used in Chapter 7 to compute the Monte-Carlo sample sizes employed in matrix trace estimators, and finally the details of implementation and discretization of the total variation functional used in some of the numerical examples in this thesis.

9.2 Future Work

At the end of each of Chapters 3, 4, 5, and 7, some general directions for future research, related to the specific topics presented in the respective chapter, are discussed. In this section, we present few more directions and ideas for further research, which arose as a result of the work in this thesis. Time constraints did not allow for their full investigation in the present study and they are left for future work. Some of these ideas are presented in their most general form, while others are described more specifically.

9.2.1 Other Applications

The success of randomized approximation algorithms have only been thoroughly evaluated in a handful of applications. However, it is widely believed that the application range of such stochastic algorithms can be extended. There are many more important medical and geophysical applications where the study of efficient randomized approximation algorithms requires more concentrated effort. Such applications include large scale seismic data inversion in oil exploration and medical imaging such as quantitative photoacoustic tomography, among many

others. For many of these applications, one typically makes large amounts of measurements and, hence, the model recovery is computationally very challenging. In addition, there are unique challenges that arise as a result of the nature of each individual application. Within the context of approximation algorithms, these challenges need to be individually investigated. These might impose a wide range of difficulties, from a simple modification to the algorithms in this thesis to devising completely new approaches.

9.2.2 Quasi-Monte Carlo and Matrix Trace Estimation

As shown in this thesis, within the context of large scale non-linear least squares problems, efficiency in estimating the objective function (or the trace of the corresponding implicit matrix) directly translates to efficiency in solving such large scale problems. In this thesis, it was shown that, for such problems, naive randomized approximation techniques using simple Monte-Carlo methods can have great success in designing highly efficient algorithms. Such Monte-Carlo methods for estimating the trace of an implicit matrix was thoroughly studied in Chapters 5 and 7. As shown, the analysis is based on a probabilistic framework for which, given two small tolerances (ε, δ), one obtains sufficient conditions on sample size in order to guarantee that the probability of the relative accuracy being below ε is more than $1 - \delta$. However, it has been shown in [137] that using simple Monte-Carlo methods, the true sample size grows like $\Omega(\varepsilon^{-2})$. As such, for scenarios where an accurate estimation is required, such algorithms might be completely inefficient and computationally expensive. And yet, it might be possible to improve the bound $\Omega(\varepsilon^{-2})$ through the application of Quasi-Monte Carlo (QMC) methods, where careful design of a sequence of correlated samples yields more accurate approximations, at lower costs. The application of such QMC methods for efficiently solving large scale inverse problems has not been greatly studied in the literature. Hence, the analysis and the practical implementation of such new algorithms is an interesting topic for future research.

9.2.3 Randomized/Deterministic Preconditioners

In Chapter 5, it was shown that the "skewness" of eigenvalue distribution of a symmetric positive semi-definite matrix greatly affects the performance and efficiency of the Gaussian trace estimation. In other words, estimating the trace of a matrix whose eigenvalues are similar

can be done more efficiently (i.e., with smaller sample size) than that for which the discrepancy between the eigenvalues is large (i.e., eigenvalues are more skewed). A question arises whether it is possible to find a randomized preconditioning scheme to balance the skewed eigenvalues of a matrix while preserving the value of the trace. In other words, one may seek to find a random matrix P such that PAP^T has a more balanced eigenvalue distribution than A, yet we have $tr(A) = tr(PAP^T)$ (or $tr(A) = \mathbb{E}\left[tr(PAP^T)\right]$). Alternatively, one could look at deterministic constructions such as the following formulation

$$\begin{aligned} &\min_{P \in \mathcal{P}} \|PAP^T\|_2^2 \\ &\text{s.t. } tr(PAP^T) = tr(A) \end{aligned}$$

where $\mathcal{P}$ is an appropriate space of *non-orthogonal* matrices. Minimizing $\|PAP^T\|_2^2$ translates to minimizing the largest eigenvalue of PAP^T, which given the constraint for the sum of eigenvalues, forces the eigenvalue distribution to be less skewed. If such a preconditioner exists, it can, in addition, be adopted for preconditioning matrices in linear system solvers.

Bibliography

[1] M. Abramowitz. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables,*. Dover Publications, Incorporated, 1974.

[2] D. Achlioptas. Database-friendly random projections. In *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 01*, volume 20, pages 274–281, 2001.

[3] H. Akaike. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Stat. Math. Tokyo*, 11:1–16, 1959.

[4] A. Alessandrini and S. Vessella. Lipschitz stability for the inverse conductivity problem. *Adv. Appl. Math.*, 35:207–241, 2005.

[5] A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas. A-optimal design of experiments for infinite-dimensional bayesian linear inverse problems with regularized ℓ_0-sparsification. *arXiv:1308.4084*, 2013.

[6] A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas. A fast and scalable method for a-optimal design of experiments for infinite-dimensional bayesian nonlinear inverse problems. *arXiv:1410.5899*, 2014.

[7] L. Amiri, B. Khaledi, and F. J. Samaniego. On skewness and dispersion among convolutions of independent gamma random variables. *Probability in the Engineering and Informational Sciences*, 25(01):55–69, 2011.

[8] A. Aravkin, M. P. Friedlander, F. J. Herrmann, and T. van Leeuwen. Robust inver-

sion, dimensionality reduction, and randomized sampling. *Mathematical programming*, 134(1):101–125, 2012.

[9] G. Archer and DM. Titterington. On some bayesian/regularization methods for image restoration. *Image Processing, IEEE Transactions on*, 4(7):989–995, 1995.

[10] S. R. Arridge. Optical tomography in medical imaging. *Inverse problems*, 15(2):R41, 1999.

[11] S R Arridge. Optical tomography in medical imaging. *Inverse Problems*, 15(2):R41, 1999.

[12] S. R. Arridge and J. C. Hebden. Optical imaging in medicine: Ii. modelling and reconstruction. *Physics in Medicine and Biology*, 42(5):841, 1997.

[13] U. Ascher. *Numerical Methods for Evolutionary Differential Equations*. SIAM, Philadelphia, PA, 2008.

[14] U. Ascher and C. Greif. *First Course in Numerical Methods*. Computational Science and Engineering. SIAM, 2011.

[15] U. Ascher and E. Haber. A multigrid method for distributed parameter estimation problems. *J. ETNA*, 18:1–18, 2003.

[16] U. Ascher, R. Mattheij, and R. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Classics. SIAM, 1995.

[17] U. Ascher and L. Petzold. *Computer Methods for Ordinary Differential and Differential-Algebraic Equations*. SIAM, 1998.

[18] U. Ascher and S. Reich. The midpoint scheme and variants for hamiltonian systems: advantages and pitfalls. *SIAM J. Scient. Comput.*, 21:1045–1065, 1999.

[19] U. Ascher and F. Roosta-Khorasani. Algorithms that satisfy a stopping criterion, probably. 2014. Preprint, arXiv:1408.5946.

[20] R. C. Aster, B. Borchers, and C. H. Thurber. *Parameter estimation and inverse problems*. Academic Press, 2013.

[21] H. Avron. Counting triangles in large graphs using randomized matrix trace estimation. *Workshop on Large-scale Data Mining: Theory and Applications*, 2010.

[22] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *JACM*, 58(2), 2011. Article 8.

[23] Z. Bai, M. Fahey, and G. Golub. Some large scale matrix computation problems. *J. Comput. Appl. Math.*, 74:71–89, 1996.

[24] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA J. Num. Anal.*, 8:141–148, 1988.

[25] C. J. Beasley. A new look at marine simultaneous sources. *The Leading Edge*, 27(7):914–917, 2008.

[26] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Appl. Numer. Math.*, 57:12141229, 2007.

[27] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Scient. Comput.*, 31(2):890–912, 2008.

[28] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging*. CRC press, 2010.

[29] D.A. Boas, D.H. Brooks, E.L. Miller, C. A. DiMarzio, M. Kilmer, R.J. Gaudette, and Q. Zhang. Imaging the body with diffuse optical tomography. *Signal Processing Magazine, IEEE*, 18(6):57–75, 2001.

[30] M. E. Bock, P. Diaconis, F. W. Huffer, and M. D. Perlman. Inequalities for linear combinations of gamma random variables. *Canadian Journal of Statistic*, 15:387–395, 1987.

[31] P. J. Boland, E. El-Neweihi, and F. Proschan. Schur properties of convolutions of exponential and geometric random variables. *Journal of Multivariate Analysis*, 48(1):157–167, 1994.

[32] J. Bon and E. Pãltãnea. Ordering properties of convolutions of exponential random variables. *Lifetime Data Analysis*, 5(2):185–192, 1999.

[33] L. Borcea, J. G. Berryman, and G. C. Papanicolaou. High-contrast impedance tomography. *Inverse Problems*, 12:835–858, 1996.

[34] A. Borsic, B. M. Graham, A. Adler, and W. R. Lionheart. Total variation regularization in electrical impedance tomography. 2007.

[35] C. Bunks, F. M. Saleck, S. Zaleski, and G. Chavent. Multiscale seismic waveform inversion. *Geophysics*, 60(5):1457–1473, 1995.

[36] R. Byrd, G. Chin, W. Neveitt, and J. Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM J. Optimization*, 21(3):977–995, 2011.

[37] Y. Cao and L. Petzold. A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM J. Scient. Comput.*, 26:359–374, 2004.

[38] T. Chan and X. Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *J. Comp. Phys.*, 193:40–66, 2003.

[39] M. Cheney, D. Isaacson, and J. C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.

[40] G. Dahlquist and A. Bjorck. *Numerical Methods*. Prentice-Hall, 1974.

[41] Y. Dai. Nonlinear conjugate gradient methods. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[42] Y. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische. Math.*, 100:21–47, 2005.

[43] C. A. Deledalle, S. Vaiter, G. Peyré, and J. M. Fadili. Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *arXiv:1405.1164*, 2014.

[44] P. Diaconis and M. D. Perlman. Bounds for tail probabilities of weighted sums of independent gamma random variables. *Lecture Notes-Monograph Series*, pages 147–166, 1990.

[45] K. van den Doel and U. Ascher. The chaotic nature of faster gradient descent methods. *J. Scient. Comput.*, 48, 2011. DOI: 10.1007/s10915-011-9521-3.

[46] K. van den Doel and U. Ascher. Adaptive and stochastic algorithms for EIT and DC resistivity problems with piecewise constant solutions and many measurements. *SIAM J. Scient. Comput.*, 34:DOI: 10.1137/110826692, 2012.

[47] K. van den Doel, U. Ascher, and E. Haber. The lost honour of ℓ_2-based regularization. *Radon Series in Computational and Applied Math*, 2013. M. Cullen, M. Freitag, S. Kindermann and R. Scheinchl (Eds).

[48] O. Dorn, E. L. Miller, and C. M. Rappaport. A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets. *Inverse Problems*, 16, 2000. 1119-1156.

[49] L. Demanet E. Candes, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5(3):861–899, 2006.

[50] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[51] Y. C. Eldar and G. Kutyniok. *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[52] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer, Dordrecht, 1996.

[53] J. Krebs et al. Iterative inversion of data from simultaneous geophysical sources. *http://www.faqs.org/patents/app/20100018718*, 28/01/2010.

[54] L. C. Evans. Partial differential equations. 1998.

[55] C. Farquharson and D. Oldenburg. Non-linear inversion using general measures of data misfit and model structure. *Geophysics J.*, 134:213–227, 1998.

[56] A. Fichtner. *Full Seismic Waveform Modeling and Inversion*. Springer, 2011.

[57] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[58] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Springer, 2013.

[59] A. Friedlander, J. Martinez, B. Molina, and M. Raydan. Gradient method with retard and generalizations. *SIAM J. Num. Anal.*, 36:275–289, 1999.

[60] M. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Scient. Comput.*, 34(3), 2012.

[61] E. Furman and Z. Landsman. Tail variance premium with applications for elliptical portfolio of risks. *Astin Bulletin*, 36(2):433, 2006.

[62] F. Gao, A. Atle, P. Williamson, et al. Full waveform inversion using deterministic source encoding. In *2010 SEG Annual Meeting*. Society of Exploration Geophysicists, 2010.

[63] H. Gao, S. Osher, and H. Zhao. Quantitative photoacoustic tomography. In *Mathematical Modeling in Biomedical Imaging II*, pages 131–158. Springer, 2012.

[64] M. Gehrea, T. Kluth, A. Lipponen, B. Jin, A. Seppaenenb, J. Kaipio, and P. Maass. Sparsity reconstruction in electrical impedance tomography: An experimental evaluation. *J. Comput. Appl. Math.*, 236:2126–2136, 2012.

[65] G. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[66] G. H. Golub, M. Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[67] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, 1997.

[68] TM. Habashy, A. Abubakar, G. Pan, A. Belani, et al. Full-waveform seismic inversion using the source-receiver compression approach. In *2010 SEG Annual Meeting*. Society of Exploration Geophysicists, 2010.

[69] E. Haber, U. Ascher, and D. Oldenburg. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics*, 69:1216–1228, 2004.

[70] E. Haber and M. Chung. Simultaneous source for non-uniform data variance and missing data. 2012. submitted.

[71] E. Haber, M. Chung, and F. Herrmann. An effective method for parameter estimation with PDE constraints with multiple right-hand sides. *SIAM J. Optimization*, 22:739–757, 2012.

[72] E. Haber, S. Heldmann, and U. Ascher. Adaptive finite volume method for distributed non-smooth parameter identification. *Inverse Problems*, 23:1659–1676, 2007.

[73] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer, 2002.

[74] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer, 1993.

[75] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer, 1996.

[76] G. Hampson, J. Stefani, and F. Herkenhoff. Acquisition using simultaneous sources. *The Leading Edge*, 27(7):918–923, 2008.

[77] M. Hanke. Regularizing properties of a truncated newton-cg algorithm for nonlinear inverse problems. *Numer. Funct. Anal. Optim.*, 18:971–993, 1997.

[78] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, 1998.

[79] P. C. Hansen. *The L-curve and its use in the numerical treatment of inverse problems*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark, 1999.

[80] M. Heath. *Scientific Computing, An Introductory Survey*. McGraw-Hill, 2002. 2nd Ed.

[81] F. Herrmann, Y. Erlangga, and T. Lin. Compressive simultaneous full-waveform simulation. *Geophysics*, 74:A35, 2009.

[82] Desmond J. Higham. Global error versus tolerance for explicit runge-kutta methods. *IMA J. Numer. Anal*, 11:457–480, 1991.

[83] P. Hitczenko and S. Kwapień. On the rademacher series. In *Probability in Banach Spaces, 9*, pages 31–36. Springer, 1994.

[84] J. Holodnak and I. Ipsen. Randomized approximation of the gram matrix: Exact computation and probabilistic bounds. *SIAM J. Matrix Anal. Applic.*, 2014. to appear.

[85] Peter J Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[86] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *J. Comm. Stat. Simul.*, 19:433–450, 1990.

[87] I. Ipsen and T. Wentworth. The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems. *SIAM J. Matrix Anal. Applic.*, 2014. To appear.

[88] V. Isakov. *Inverse Problems for Partial Differential Equations*. Springer; 2nd edition, 2006.

[89] J. Jost and J Jost. *Riemannian geometry and geometric analysis*, volume 42005. Springer, 2008.

[90] A. Juditsky, G. Lan, A. Nemirovski, and A. Shapiro. Stochastic approximation approach to stochastic programming. *SIAM J. Optimization*, 19(4):1574–1609, 2009.

[91] J. Kaipo and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.

[92] S. A. Khayam. The discrete cosine transform (dct): theory and application. *Michigan State University*, 2003.

[93] S. Kochar and M. Xu. On the right spread order of convolutions of heterogeneous exponential random variables. *Journal of Multivariate Analysis*, 101(1):165–176, 2010.

[94] S. Kochar and M. Xu. The tail behavior of the convolutions of gamma random variables. *Journal of Statistical Planning and Inference*, 141(1):418–428, 2011.

[95] S. Kochar and M. Xu. Some unified results on comparing linear combinations of independent gamma random variables. *Probability in the Engineering and Informational Sciences*, 26(03):393–404, 2012.

[96] R. Kumar, C. Da Silva, O. Akalin, A. Y. Aravkin, H. Mansour, B. Recht, and F. J. Herrmann. Efficient matrix completion for seismic data reconstruction. Submitted to Geophysics on August 8, 2014., 08 2014.

[97] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge, 2004.

[98] Yan Yan Li, Michael Vogelius, and Communicated R. V. Kohn. Gradient estimates for solutions to divergence form elliptic equations with discontinuous coefficients. *Arch. Rational Mech. Anal*, 153:91–151, 2000.

[99] S. Lihong and Z. Xinsheng. Stochastic comparisons of order statistics from gamma distributions. *Journal of Multivariate Analysis*, 93(1):112–121, 2005.

[100] AK. Louis. Medical imaging: state of the art and future development. *Inverse Problems*, 8(5):709, 1992.

[101] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[102] W. Menke. *Geophysical data analysis: discrete inverse theory*. Academic press, 2012.

[103] M. L. Merkle and Petrović. On schur-convexity of some distribution functions. *Publications de l'Institut Mathématique*, 56(76):111–118, 1994.

[104] Y. Michel. Diagnostics on the cost-function in variational assimilations for meteorological models. *Nonlinear Processes in Geophysics*, 21(1):187–199, 2014.

[105] P. Moghaddam and F. Herrmann. Randomized full-waveform inversion: a dimensionality reduction approach. In *SEG Technical Program Expanded Abstracts*, volume 29, pages 977–982, 2010.

[106] A. Mood, F. A. Graybill, and D. C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill; 3rd edition, 1974.

[107] V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer, 1984.

[108] G. A. Newman and D. L. Alumbaugh. Frequency-domain modelling of airborne electromagnetic responses using staggered finite differences. *Geophys. Prospecting*, 43:1021–1042, 1995.

[109] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.

[110] D. Oldenburg, E. Haber, and R. Shekhtman. 3D inverseion of multi-source time domain electromagnetic data. *J. Geophysics*, 2013. To appear.

[111] A. Pidlisecky, E. Haber, and R. Knight. RESINVM3D: A MATLAB 3D Resistivity Inversion Package. *Geophysics*, 72(2):H1–H10, 2007.

[112] M. Raydan. *Convergence Properties of the Barzilai and Borwein Gradient Method*. PhD thesis, Rice University, Houston, Texas, 1991.

[113] A. Rieder. Inexact newton regularization using conjugate gradients as inner iteration. *SIAM J. Numer. Anal.*, 43:604–622, 2005.

[114] A. Rieder and A. Lechleiter. Towards a general convergence theory for inexact newton regularizations. *Numer. Math.*, 114(3):521–548, 2010.

[115] J. Rohmberg, R. Neelamani, C. Krohn, J. Krebs, M. Deffenbaugh, and J. Anderson. Efficient seismic forward modeling and acquisition using simultaneous random sources and sparsity. *Geophysics*, 75(6):WB15–WB27, 2010.

[116] F. Roosta-Khorasani and U. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 2014. DOI: 10.1007/s10208-014-9220-1.

[117] F. Roosta-Khorasani, G. Székely, and U. Ascher. Assessing stochastic algorithms for large scale nonlinear least squares problems using extremal probabilities of linear combinations of gamma random variables. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):61–90, 2015. DOI: 10.1137/14096311X.

[118] F. Roosta-Khorasani, K. van den Doel, and U. Ascher. Data completion and stochastic algorithms for PDE inversion problems with many measurements. *Electronic Transactions on Numerical Analysis*, 42:177–196, 2014.

[119] F. Roosta-Khorasani, K. van den Doel, and U. Ascher. Stochastic algorithms for inverse problems involving PDEs and many measurements. *SIAM J. Scientific Computing*, 36(5):S3–S22, 2014.

[120] B. H. Russell. *Introduction to seismic inversion methods*, volume 2. Society of Exploration Geophysicists, 1988.

[121] A. K. Saibaba and P. K. Kitanidis. Uncertainty quantification in geostatistical approach to inverse problems. *arXiv:1404.1263*, 2014.

[122] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge, 2001.

[123] L. L. Scharf. *Statistical signal processing*, volume 98. Addison-Wesley Reading, MA, 1991.

[124] F. Schmidt. The laplace-beltrami-operator on riemannian manifolds. *Technical Report, Computer Vision Group, Technische Universität München.*

[125] R. J. Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, 2:39–48, 1974.

[126] A. Shapiro, D. Dentcheva, and D. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. Piladelphia: SIAM, 2009.

[127] S. Shkoller. *Lecture Notes on Partial Differential Equations*. Department of Mathematics, University of California, Davis, June 2012.

[128] N. C. Smith and K. Vozoff. Two dimensional DC resistivity inversion for dipole dipole data. *IEEE Trans. on geoscience and remote sensing*, GE 22:21–28, 1984.

[129] G. J. Székely and N. K. Bakirov. Extremal probabilities for gaussian quadratic forms. *Probab. Theory Related Fields*, 126:184–202, 2003.

[130] J. Tropp. Column subset selection, matrix factorization, and eigenvalue optimization. *SODA*, pages 978–986, 2009. SIAM.

[131] K. van den Doel and U. M. Ascher. On level set regularization for highly ill-posed distributed parameter estimation problems. *J. Comp. Phys.*, 216:707–723, 2006.

[132] K. van den Doel and U. M. Ascher. Dynamic level set regularization for large distributed parameter estimation problems. *Inverse Problems*, 23:1271–1288, 2007.

[133] K. van den Doel and U. M. Ascher. Dynamic regularization, level set shape optimization, and computed myography. *Control and Optimization with Differential-Algebraic Constraints*, 23:315, 2012.

[134] T. van Leeuwen, S. Aravkin, and F. Herrmann. Seismic waveform inversion by stochastic optimization. *Hindawi Intl. J. Geophysics*, 2011:doi:10.1155/2011/689041, 2012.

[135] C. Vogel. *Computational methods for inverse problem*. SIAM, Philadelphia, 2002.

[136] w. Rundell and H. W. Engl. *Inverse problems in medical imaging and nondestructive testing*. Springer-Verlag New York, Inc., 1997.

[137] K. Wimmer, Y. Wu, and P. Zhang. Optimal query complexity for estimating the trace of a matrix. *arXiv preprint arXiv:1405.7112*, 2014.

[138] J. Young and D. Ridzal. An application of random projection to parameter estimation in partial differential equations. *SIAM J. Scient. Comput.*, 34:A2344–A2365, 2012.

[139] Y. Yu. Some stochastic inequalities for weighted sums. *Bernoulli*, 17(3):1044–1053, 2011.

[140] Z. Yuan and H. Jiang. Quantitative photoacoustic tomography: Recovery of optical absorption coefficient maps of heterogeneous media. *Applied physics letters*, 88(23):231101–231101, 2006.

[141] P. Zhao. Some new results on convolutions of heterogeneous gamma random variables. *Journal of Multivariate Analysis*, 102(5):958–976, 2011.

[142] P. Zhao and N. Balakrishnan. Mean residual life order of convolutions of heterogeneous exponential random variables. *Journal of Multivariate Analysis*, 100(8):1792–1801, 2009.

Appendix A

Implementation Details

Here we describe the forward problem that yields the operators $\mathbf{f}_i(\mathbf{m})$ of (1.4), and provide some details on the stabilized GN iteration used in our numerical experiments. We also provide details of discretization of the EIT/DC resistivity problem in two and three dimensions, injection of a priori knowledge on the sought parameter function via transformation functions in the original PDE, a short MATLAB code which is used in Chapter 7 to compute the Monte-Carlo sample sizes used in matrix trace estimators, and finally the details of implementation and discretization of the total variation functional used in numerical examples in this thesis.

There is nothing strictly new here, and yet some of the details are both tricky and very important for the success of an efficient code for computing reasonable reconstructions for this highly ill-posed problem. It is therefore convenient to gather all these details in one place for further reference.

A.1 Discretizing the Forward Problem

The PDE (3.5) is discretized on a staggered grid as described in [15] and in Section 3.1 of [13]. The domain is divided into uniform cells of side length h, and a cell-nodal discretization is employed, where the field unknowns $u_{i,j}$ (or $u_{i,j,k}$ in 3D) are perceived to be located at the cell corners (which are cell centers of the dual grid) while $\mu_{i+1/2,j+1/2}$ values are at cell centers (cell corners of the dual grid). For the finite volume derivation, the PDE (3.5a) is written first as

$$\mathbf{j} = \mu(\mathbf{x})\nabla u, \quad \mathbf{x} \in \Omega, \tag{A.1a}$$

$$\nabla \cdot \mathbf{j} = q(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{A.1b}$$

and then both first order PDEs are integrated prior to discretization. A subsequent, standard removal of the constant null-space then results in the discretized problem (1.3).

In 2D, let us assume here for notational simplicity that the source q is a δ-function centred at a point in the finite volume cell (i_*, j_*). The actual sources used in our experiments are combinations of such functions, as detailed in Section 3.3.2, and the discretization described below is correspondingly generalized in a straightforward manner. We write for the flux, $\mathbf{v} = (v^x, v^y)^T$, expressions such as $u_x = \mu^{-1} v^x$ at the eastern cell face $x = x_{i+1/2}$. Setting

$$\begin{aligned} \mu^{-1}(x_{i+1/2}, y) &\approx \mu^{-1}_{i+1/2,j} = \frac{1}{2}\big(\mu^{-1}_{i,j} + \mu^{-1}_{i+1,j}\big), \\ v^x_{i+1/2,j} &= h^{-1} \int_{y_{j-1/2}}^{y_{j+1/2}} v^x(x_{i+1/2}, y)\, dy, \end{aligned}$$

and integrating yields

$$v^x_{i+1/2,j} = \mu_{i+1/2,j} \frac{u_{i+1,j} - u_{i,j}}{h}.$$

Similar expressions are obtained at the other three faces of the cell. Then integrating (A.1b) over the cell yields

$$\begin{aligned} &\big[\mu_{i+1/2,j}(u_{i+1,j} - u_{i,j}) - \mu_{i-1/2,j}(u_{i,j} - u_{i-1,j}) \\ &+\mu_{i,j+1/2}(u_{i,j+1} - u_{i,j}) - \mu_{i,j-1/2}(u_{i,j} - u_{i,j-1})\big] \\ &= \begin{cases} 1 & if\ i = i_*\ and\ j = j_* \\ 0 & otherwise \end{cases}, \quad 1 \le i, j \le 1/h. \end{aligned} \tag{A.2}$$

Repeating the process in 3D (with an obvious notational extension for the source location)

yields the formula

$$\begin{aligned} & h\big[\mu_{i+1/2,j,k}(u_{i+1,j,k}-u_{i,j,k})-\mu_{i-1/2,j,k}(u_{i,j,k}-u_{i-1,j,k}) \\ + \;& \mu_{i,j+1/2,k}(u_{i,j+1,k}-u_{i,j,k})-\mu_{i,j-1/2,k}(u_{i,j,k}-u_{i,j-1,k}) \\ + \;& \mu_{i,j,k+1/2}(u_{i,j,k+1}-u_{i,j,k})-\mu_{i,j,k-1/2}(u_{i,j,k}-u_{i,j,k-1})\big] \\ & = \begin{cases} 1 & if\ i=i_*\ and\ j=j_*\ and\ k=k_* \\ 0 & otherwise \end{cases}, \quad 1\le i,j,k\le 1/h, \end{aligned} \tag{A.3}$$

where, e.g.,

$$\begin{aligned} \mu^{-1}_{i+1/2,j,k} &= \frac{1}{4}\big(\mu^{-1}_{i+1/2,j+1/2,k+1/2}+\mu^{-1}_{i+1/2,j+1/2,k-1/2} \\ &+ \mu^{-1}_{i+1/2,j-1/2,k+1/2}+\mu^{-1}_{i+1/2,j-1/2,k-1/2}\big). \end{aligned}$$

The derivation is entirely parallel to the 2D case, although note the extra factor h multiplying the left hand side in (A.3), which arises due to the special nature of the source q.

The boundary conditions are discretized by applying, say, (A.2) at $i = 1$ and utilizing (3.5b) to set $u_{0,j} = u_{2,j}$ and $\mu_{-1/2,j} = \mu_{1/2,j}$. Combining all this results in a linear system of the form (1.3) which is positive semi-definite and has a constant null space, as does the PDE problem (3.5). This null-space is removed using standard techniques.

The method employed for solving the resulting linear system does not directly affect our considerations in this thesis. For the sake of completeness, however, let us add that given the large number of right hand sides in problems such as (1.3) that must be solved, a direct method which involves one Cholesky decomposition followed by forward and backward substitution for each right hand side is highly recommended. If the program runs out of memory (on our system this happens in 3D for $h = 2^{-6}$) then we use a preconditioned conjugate gradient method with an incomplete Cholesky decomposition for a preconditioner.

A.2 Taking Advantage of Additional A Priori Information

In general, we wish to recover $\mu(\mathbf{x})$ based on measurements of the field $u(\mathbf{x})$ such that (3.5) approximately holds. Note that, since the measurements are made only at relatively few locations, e.g., the domain's boundary rather than every point in its interior, the matrices P_i (whether or not they are all equal) all have significantly more columns than rows. Moreover, this inverse problem is notoriously ill-posed and difficult in practice, especially for cases where μ has large-magnitude gradients. Below we introduce additional a priori information, when such is available, via a parametrization of $\mu(\mathbf{x})$ in terms of $m(\mathbf{x})$ (see also [47]). To this end let us define the transfer function

$$\psi(\tau) = \psi(\tau; \theta, \alpha_1, \alpha_2) = \alpha \tanh\left(\frac{\tau}{\alpha\theta}\right) + \frac{\alpha_1 + \alpha_2}{2}, \quad \alpha = \frac{\alpha_2 - \alpha_1}{2}. \tag{A.4}$$

This maps the real line into the interval (α_1, α_2) with the maximum slope θ^{-1} attained at $\tau = 0$.

1. In practice, often there are reasonably tight bounds available, say $\mu_{\min}$ and $\mu_{\max}$, such that $\mu_{\min} \le \mu(\mathbf{x}) \le \mu_{\max}$. Such information may be enforced using (A.4) by defining

$$\mu(\mathbf{x}) = \psi(m(\mathbf{x})), \quad \text{with } \psi(\tau) = \psi(\tau; 1, \mu_{\min}, \mu_{\max}). \tag{A.5}$$

2. Occasionally it is reasonable to assume that the sought conductivity function $\mu(\mathbf{x})$ takes only one of two values, μ_I or μ_{II}, at each $\mathbf{x}$. Viewing one of these as a background value, the problem is that of shape optimization. Such an assumption greatly stabilizes the inverse problem [4]. In [46, 131, 132] we considered an approximate level set function representation for the present problem. We write $\mu(\mathbf{x}) = \lim_{h\to 0} \mu(\mathbf{x}; h)$, where

$$\mu(\mathbf{x}; h) = \psi(m(\mathbf{x}); h, \mu_I, \mu_{II}). \tag{A.6}$$

   The function $\psi(\tau; h)$ depends on the resolution, or grid width h. It is a scaled and mollified version of the Heaviside step function, and its derivative magnitude is at most $O(\frac{|\mu_I - \mu_{II}|}{h})$.

Thus, as $h \to 0$ the sought function $m(\mathbf{x})$ satisfying

$$\begin{aligned} \nabla \cdot (\psi(m(\mathbf{x}))\nabla \mathbf{u}_i) &= \mathbf{q}_i, \quad i = 1, \ldots, s, \\ \frac{\partial \mathbf{u}_i}{\partial n}\big|_{\partial\Omega} &= 0, \end{aligned} \tag{A.7}$$

has bounded first derivatives, whereas $\mu(\mathbf{x})$ is generally discontinuous.

Establishing the relationship between μ and m completes the definition of the forward operators $\mathbf{f}_i(\mathbf{m})$ by (1.4).

A.3 Stabilized Gauss-Newton

Here we briefly describe the modifications made to the GN method (1.11), turning it into the stabilized GN method used in our experiments. The matrix at the left hand side of (1.11a) is singular in the usual case where $l < l_m$, and therefore this linear system requires regularization. Furthermore, $\delta\mathbf{m}$ also requires smoothing, because there is nothing in (1.11) to prevent it from forming a non-smooth grid function. These regularization tasks are achieved by applying only a small number of PCG iterations towards the solution of (1.11a), see [131, 133]. This dynamic regularization (or iterative regularization [78]) is also very efficient, and results in a *stabilized GN* iteration. An adaptive algorithm for determining a good number of such inner iterations is proposed in [46]. However, here we opt to keep this number fixed at r PCG iterations independently of n, in order to be able to compare other aspects of our algorithms more fairly. Further, the task of penalizing excessive non-smoothness in the correction $\delta\mathbf{m}$ is achieved by choosing as the preconditioner a discrete Laplacian with homogeneous Neumann boundary conditions. This corresponds to a penalty on $\int |\nabla m(\mathbf{x})|^2$ (i.e., least squares but *not* total variation).

The modified GN iteration described above is our outer iteration, and the entire regularization method is called *dynamical regularization* [77, 78, 113, 114, 131, 133]. The essential cost in terms of forward operator simulations comes through (1.11a) from multiplying J_i or J_i^T by a vector. Each such multiplication costs one forward operator simulation, hence $2rs$ simulations for the left hand side of (1.11a) (or $2rn_k$ in case of (2.6a)). The evaluation of the gradient

costs another $2s$ forward operator evaluations per outer iteration. Considering K GN outer iterations, this gives the work under-estimate formula (1.12). This still neglects, for clarity, the additional line search costs, although the additional forward operator simulations necessitated for determining α_k in (1.11b) have of course been counted and included in the work tallies reported in all the tables in this thesis.

A.4 MATLAB Code

Here we provide a short MATLAB code, promised in Section 7.1, to calculate the necessary or sufficient sample sizes to satisfy the probabilistic accuracy guarantees (7.2) for a SPSD matrix using the Gaussian trace estimator. This code can be easily modified to be used for (7.10) as well.

```
function [N1,N2] = getSampleSizes(epsilon,delta,maxN,r)
% INPUT:
% @ epsilon:  Accuracy of the estimation .
% @ delta:  Uncertainty of the estimation.
% @ r: Rank of the matrix (Use r = 1 for obtaining the sufficient sample sizes).
% @ maxN: Maximum allowable sample size
% OUTPUT:
% @ N1: The sufficient (or necessary) sample size for (7.2a).
% @ N2: The sufficient (or necessary) sample size for (7.2b).
Ns = 1:1:maxN;
P1 = gammainc(Ns*r*(1-epsilon)/2,Ns*r/2);
I1 = find(P1 <= delta,1,"first");
N1 = Ns(I1); % Necessary/Sufficient sample size obtained for (7.2a).
Ns = (floor(1/epsilon)+1):1:maxN;
P2 = gammainc(Ns*r*(1+epsilon)/2,Ns*r/2);
I2 = find(P2 >= 1-delta,1,"first");
N2 = Ns(I2); % Necessary/Sufficient sample size obtained for (7.2b).
end
```

A.5 Implementation of Total Variation Functional

For the Total Variation (TV) regularization, $R(\mathbf{m})$ in (1.7) is the discretization of the TV functional

$$TV(m) = \int_\Omega |\nabla m(\mathbf{x})|,$$

where Ω is the domain under investigation. Consider a 2D square domain, which is divided into uniform cells of side length h, resulting in N^2 cells. Let (x_i, x_j) be the center of the cell (i, j). A usual discretization of this TV integral, using this mesh, is obtained as

$$TV(m) \approx \sum_{i,j=1}^{N} h^2 \left(\left| \frac{\partial m(x_i, x_j)}{\partial x} \right| + \left| \frac{\partial m(x_i, x_j)}{\partial y} \right| \right),$$

where $\partial m(x_i, x_j)/\partial x$ is the value of $\partial m/\partial x$ at the center of the cell (i, j). The standard approach for obtaining $|\partial m(x_i, x_j)/\partial x|$ is by "averaging the square of the differences" among cell values. More specifically, for $i, j = 1, \cdots, N$, letting $m_{i,j}$ denote the grid value of $m(\mathbf{x})$ at cell (i, j), we get

$$\left| \frac{\partial m(x_i, x_j)}{\partial x} \right| \approx \frac{1}{h} \sqrt{\frac{(m_{i+1,j} - m_{i,j})^2 + (m_{i,j} - m_{i-1,j})^2}{2}}, \tag{A.8a}$$

$$\left| \frac{\partial m(x_i, x_j)}{\partial y} \right| \approx \frac{1}{h} \sqrt{\frac{(m_{i,j+1} - m_{i,j})^2 + (m_{i,j} - m_{i,j-1})^2}{2}}. \tag{A.8b}$$

Next, we form the vector $\mathbf{m}$ consisting of $m_{i,j}$ values. Let D_x and D_y be the matrices that implement the difference operations in (A.8) in x and y directions, respectively. Similarly, let A_x and A_y be the matrices that implement the averaging operations in (A.8) in x and y directions, respectively. Now we can write (A.8) in vectorized form as

$$R(\mathbf{m}) = \mathbf{1}^T \sqrt{A_x(D_x\mathbf{m})^2 + A_y(D_y\mathbf{m})^2},$$

where $\mathbf{1}$ is a vector of 1's, and the square and absolute value are taken pointwise.

One can introduce differentiability to $R(\mathbf{m})$ by

$$R_\varepsilon(\mathbf{m}) = \mathbf{1}^T \sqrt{A_x(D_x\mathbf{m})^2 + A_y(D_y\mathbf{m})^2 + \varepsilon\mathbf{1}},$$

for some $\varepsilon \ll 1$. An alternative is to use the Huber switching function [55, 85, 122]. Extensions of the above procedure to 3D is straightforward.