

Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Christopher Wang

March 25th, 2025

Pixels, Priors, and Precision: Bayesian Approaches to Image Reconstruction

By

Christopher Wang

Julianne Chung
Adviser

Mathematics

Julianne Chung
Adviser

Elizabeth Newman
Committee Member

Levon Nurbekyan
Committee Member

Kevin McAlister
Committee Member

2025

Pixels, Priors, and Precision: Bayesian Approaches to Image Reconstruction

By

Christopher Wang

Julianne Chung
Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Mathematics

2025

Abstract

Pixels, Priors, and Precision: Bayesian Approaches to Image Reconstruction
By Christopher Wang

With a growing reliance on data analytics to draw insights and conclusions from real-world phenomena, ensuring accuracy no longer suffices — computational methods are essential for handling large-scale datasets and characterizing uncertainty, enabling reconstructions that are not only precise but also trustworthy. This thesis seeks to examine the existing regularization methods, Bayesian frameworks, and computational tools used to stabilize the ill-posedness of inverse problems, in the context of seismic imaging. Specifically, methods for spectral filtering in a Tikhonov formulation (e.g. General Cross Validation minimization), prior specification, and Markov Chain Monte Carlo methods for exploring complex non-Gaussian distributions are applied to synthetic seismic and tomographic data, with the goal of gauging fidelity, reliability, and computational feasibility. Numerical experiments demonstrate that these approaches — especially when combined with iterative solvers — succeed in mitigating noise and recovering high-fidelity reconstructions, while remaining computationally feasible. Furthermore, the comparison of hierarchical sampling schemes and fixed parameter methods reveals how hyperparameter inference can refine the solution space, yielding significantly better estimates at the cost of computational complexity.

Pixels, Priors, and Precision: Bayesian Approaches to Image Reconstruction

By

Christopher Wang

Julianne Chung
Adviser

A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements for the degree of
Bachelor of Science with Honors

Mathematics

2025

Acknowledgments

My journey in math begins around two years ago, when I transferred to Emory for the remainder of my undergraduate degree. In that stretch of time, I have encountered a countably infinite number of challenges that would not have been surmountable without the help of others.

In particular, I extend my deepest gratitude to Professor Julianne Chung. She has not only served as an incredible teacher—one who imparted an incredible amount of knowledge and wisdom onto me—but also a mentor in shaping my decision to continue pursuing math after graduation. Although our weekly meetings were centered around the progression of my thesis, her commitment to her students' well-being resulted in long discussions about how I was doing and how to navigate large-decisions in life. I truly would not be in this position if I did not enroll in her undergraduate Numerical Analysis course the Spring of 2024.

I am equally thankful for Professor Elizabeth Newman, who has provided relentless support to me during my time at Emory. A quote that encapsulates her demeanor towards students is "that's why we use the whiteboard." Indeed, as I unknowingly violated numerous mathematical laws in my proofs, I knew it was ok to make a mistakes. That those errors incurred were an opportunity to learn, not a permanent mark on my ability to do math.

I would also like to thank Professor Levon Nurbekyan, who demonstrated an incredible dedication to my peers and I in Non-Linear Optimization. The culture he helped to foster in the classroom is why I have been able to establish such strong friendships with those in my section. His enthusiasm to work with students is unparalleled.

Finally, among the faculty at Emory, I would like to thank Professor Kevin McAlister. Although I never had the opportunity to enroll in a course he taught, our conversations about working with Emory students on various data projects have challenged my perspectives on pedagogy and leadership. Those moments underscore his abilities as a teacher to consistently push his students to think deeper.

As I transition into a Ph.D. program, the professors whom I have mentioned exemplify

the traits and characteristics I wish to embody. I hope to have the same impact on others, as they have had on me.

Among those who I spent 2 years in the "trenches" with, I would like to thank Tristan Bishop, a dear friend who inspired me to pursue math my first semester at Emory. I would also like to thank Sophia Xiao, Kon Kon, Andrew Kalish, and Ken Nakatsu—all of these people made the late-night work sessions at Woodruff Memorial Library enjoyable. Gratitude must also be given to my friends Michael Solomon and Joseph Woods, who have kept me accountable in the gym.

Saving the best for last, words cannot describe the thanks and gratitude I have for my family and my fiance. These past years have not been easy, and the support they provided me has been invaluable—from long calls home asking for advice to hugs when the world felt exceedingly difficult. Thank you.

The list of people whom I've acknowledged is long, but is a testament to the impacts of those who've worked and spent time with. Thank you everyone, again.

Contents

1	Introduction	1
1.1	Forward and Inverse Problems	1
1.2	Applications of Inverse Problems	4
1.3	Computational Challenges to Inverse Problems	6
2	Computational Inverse Problems	10
2.1	Regularization Methods	10
2.1.1	Regularization Parameter Selection Methods	14
2.2	Bayesian Framework Formalization	15
2.2.1	Different Priors	17
2.2.2	Conjugate Gradient	22
3	Computational Methods for UQ and Sampling	24
3.1	Randomized GCV Curve Estimation	25
3.2	MAP Estimation	28
3.3	Edge Preserving Prior	29
3.4	Fixed Parameter Sampling	31
3.5	Hierarchical Gibbs	32
4	Numerical Experiments	36
4.1	Experiment #1: Regularization Parameter Selection	37

4.1.1	GCV: A Deeper Dive	40
4.2	Experiment #2: Different Priors	43
4.3	Experiment #3: MCMC Methods	48
4.3.1	Computational Complexity	53
5	Conclusions and Future Work	55
5.1	Future Work	56
5.1.1	Real-World Data and Varying Sensor Placement	57
5.1.2	Preconditioning and Other Iterative Methods	57
5.1.3	Boundary Conditions and Priors	58
	Bibliography	59

List of Figures

1.1	Seismic Imaging: Sensor and Receiver Placement	5
1.2	Naive Reconstruction with 5% Level of Noise Added	6
1.3	Picard Plot of 16x16 pixel example	9
2.1	Comparison of Naive and Regularized Solutions	13
2.2	Adjusted Picard Plot of 16x16 pixel example	13
4.1	Over and Under-Regularization for Inverse Problems	38
4.2	Different Regularization Parameter Selection Methods in Subduction Imaging	39
4.3	Randomized GCV Curve Minimization	41
4.4	GCV Curve Approximation Using Randomized Methods	42
4.5	Three Zone True Image	43
4.6	Three Phases: Different Regularization Parameter Selection Methods	43
4.7	Numerical Experiment #2 True Images	44
4.8	Different Priors in Subduction Imaging	45
4.9	Different Priors in Smooth Imaging	46
4.10	Different Priors in C.T. Imaging	47
4.11	MCMC V.S. Fixed Parameter Methods with a GMRF Prior	49
4.12	Markov Chain Diagnostics using the GMRF Prior	49
4.13	MCMC V.S. Fixed Parameter Methods with an Identity Prior	51
4.14	Markov Chain Diagnostics using the Identity Prior	51

4.15 MCMC V.S. Fixed Parameter Methods with an Edge Preserving Prior . . .	52
4.16 Markov Chain Diagnostics using the Edge-Preserving Prior	52

List of Tables

4.1	Fixed Parameters Run Times	53
4.2	MCMC Run Times	53

List of Algorithms

1	Iterative Edge-Preserving Reconstruction using IGMRF Priors	29
2	Hierarchical Gibbs	34

Chapter 1

Introduction

1.1 Forward and Inverse Problems

Inverse problems are a fascinating and growing field in Computational Mathematics that deal with the estimation of directly unobservable phenomena from an aggregation of data by leveraging knowledge of the connection between the two. This definition, vague and general by construction, is emblematic of its wide-spread prevalence to fields ranging from Geographic Sciences to Medical Imaging, and, more recently, to Machine Learning. To build intuition into the Inverse Problem, we begin by examining its counterpart, the Forward Problem.

Most forward problems arise when predicting the outcome of a process given input parameters and a known mapping that defines how these inputs influence the result. At a basic level, the desired output is retrieved by adhering to the mapping. While forward problems can incorporate stochasticity, meaning the outcomes may not be deterministic, there is an inherent sense of replicability that exists – plugging in the same parameters should result in a similar output. This notion captures a critical property of these classes of problems — their well-posed nature. Introduced by Hadamard, Well-Posed Problems satisfy the following criteria 1) the problem has a solution, 2) the solution is unique, and 3) the solution depends

continuously on the input parameters [7].

In contrast, Inverse Problems are ill-posed, meaning at least one of the three criteria are violated [8, 20]. In this setting, given an output and some knowledge of how an input is mapped, the goal is to recover the input. In theory, this may be simple, but in practical applications, physical constraints complicate the matter severely. For instance, multiple distinct inputs might produce the same observed output (leading to non-uniqueness), or small fluctuations in the observed data may yield disproportionately large variations in the inferred inputs (undermining stability).

Having established general intuition into the formulation of Inverse Problems, a formal presentation follows from [1]. Let $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where \mathcal{X} and \mathcal{Y} are functional spaces that describe a physical model of interest, and define the forward operator $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$, which is a known mapping. Then an Inverse Problem can be stated as:

$$y = \mathcal{F}(x) \tag{1.1}$$

in which the goal becomes to determine the function x , given y and \mathcal{F} . In practice, however, y as a function is not given explicitly, and instead, takes the form of a series of recorded observations or measurements. Furthermore, \mathcal{F} may not be known, requiring further assumptions or approximations [2]. A commonly used example to illustrate this is Computed Tomography (CT) technology, in which a limited number of x-rays are taken to construct an accurate representation of the brain – in particular, a finite amount of measurements are recorded, but a model of x-ray energy decay is known [11]. These physical constraints are reflected by numerically discretizing the functions x and y , and incorporating error into 1.1, yielding the following:

$$\mathbf{b} = \mathcal{F}(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{1.2}$$

where $\mathbf{b} \in \mathbb{R}^M$ is a vector containing measurements, $\boldsymbol{\epsilon} \in \mathbb{R}^N$ is a random vector that

captures the error or noise in each measurement, and $\mathbf{x} \in \mathbb{R}^N$ is the solution. We now make two important assumptions: 1) the forward operator is linear and 2) the entries of $\boldsymbol{\epsilon}$ are independent and identically distributed standard normal random variables. The above equation then further simplifies to

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \quad (1.3)$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$. Then $\mathbf{b} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}, \sigma^2 \mathbf{I}_M)$ — indeed this is true as generally, given $\mathbf{v} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbf{w} = \mathbf{c} + \mathbf{B}\mathbf{v}$, then $\mathbf{w} \sim \mathcal{N}(\mathbf{c} + \mathbf{B}\boldsymbol{\mu}, \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\top)$. The proof of this property, which follows from exercise 1.6 in Computational Uncertainty by Bardsley, is found below, where

$$\begin{aligned} \mathbb{E}[\mathbf{w}] &= \mathbb{E}[\mathbf{c} + \mathbf{B}\mathbf{v}] \\ &= \mathbf{B}\mathbb{E}[\mathbf{v}] + \mathbb{E}[\mathbf{c}], \text{ by linearity} \\ &= \mathbf{B}\boldsymbol{\mu} + \mathbf{c}, \text{ since } \mathbf{c} \text{ is constant and } \mathbb{E}[\mathbf{v}] = \boldsymbol{\mu} \end{aligned} \quad (1.4)$$

and

$$\begin{aligned} \text{Cov}[\mathbf{w}] &:= \mathbb{E}[(\mathbf{w} - \mathbb{E}[\mathbf{w}])(\mathbf{w} - \mathbb{E}[\mathbf{w}])^\top] \\ &= \mathbb{E}[(\mathbf{B}\mathbf{v} + \mathbf{c} - (\mathbf{B}\boldsymbol{\mu} + \mathbf{c}))(\mathbf{B}\mathbf{v} + \mathbf{c} - (\mathbf{B}\boldsymbol{\mu} + \mathbf{c}))^\top] \\ &= \mathbb{E}[(\mathbf{B}\mathbf{v} - \mathbf{B}\boldsymbol{\mu})(\mathbf{B}\mathbf{v} - \mathbf{B}\boldsymbol{\mu})^\top] \\ &= \mathbb{E}[\mathbf{B}(\mathbf{v} - \boldsymbol{\mu})(\mathbf{v} - \boldsymbol{\mu})^\top \mathbf{B}^\top] \\ &= \mathbf{B}\mathbb{E}[(\mathbf{v} - \boldsymbol{\mu})(\mathbf{v} - \boldsymbol{\mu})^\top] \mathbf{B}^\top, \text{ by linearity} \\ &= \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\top, \text{ by definition of variance. } [2] \end{aligned} \quad (1.5)$$

As a naive solution, one can solve the linear system, $\mathbf{b} = \mathbf{A}\mathbf{x}$, for \mathbf{x} directly using methods and techniques such as matrix inversion — if \mathbf{A}^{-1} exists — iterative methods to solve the

corresponding least-squares problem — $\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2$ — and many more. However, such a solution is often erroneous, and will be discussed further. Thus, the general goal of an inverse problem is to reconstruct or recover parameters that accurately characterize a function from a set of observations.

1.2 Applications of Inverse Problems

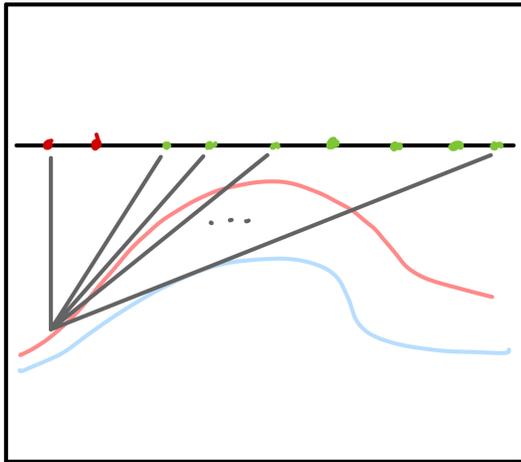
With an increasing reliance on data to extract meaningful insights, the framework of Inverse Problems has become a powerful tool to model physical phenomena across a myriad of disciplines.

In biomedical engineering, medical imaging technologies such as magnetic resonance imaging (MRI) and CT scans solve inverse problems to create an image of the desired cross section [11]. For the latter, the function of interest can be understood as a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ whose input is the cross-section being reconstructed and output is the density at a given point. The key idea being that the different tissues that comprise the cross-section have different densities, and thus can be used to create an image. Mathematically, measurements of the energy decay from x-rays fired into varying tissue density function as line integral measurements, which can then be inverted to solve for the original function [13]. The measurements are stored in \mathbf{b} , and \mathbf{A} encodes the aforementioned transformation of the function of interest. In image-de-blurring, the blurry image $\mathbf{B} \in \mathbb{R}^{m \times n}$ is first vectorized into $\mathbf{b} \in \mathbb{R}^{mn}$. With an understanding of how the image is blurred, encoded in the forward operator $\mathbf{A} \in \mathbb{R}^{mn \times mn}$, one is able to recover $\mathbf{x} \in \mathbb{R}^{mn}$, the unblurred image — again, the formulation is a linear inverse problem, $\mathbf{b} = \mathbf{Ax}$ [9].

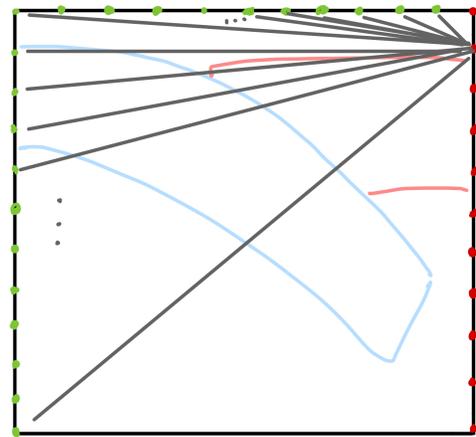
Faulty Measurements: Inverse Problems and Seismology

A large focus of this thesis is centered around the applications of Inverse Problems to geophysics and seismology – the study of imaging planetary bodies through the propagation of

seismic waves [15]. This has critical applications across many different fields such as civil engineering for structural stability evaluation and sinkhole detection, and archaeology for investigating historical sites. Similar to CT scans, the measurements can be thought of as line integrals that measure and denote how long it takes a wave to propagate through varying densities of materials [15]. This can be seen in diagrams below, where the red dots denote wave-generators or source placements, and the green dots denote receiver placements. The light red and blue lines indicate the boundaries of different material compositions.



(a) Example Seismic Imaging Problem



(b) IR Tools Problem

Figure 1.1: Seismic Imaging: Sensor and Receiver Placement

In example 1.1a, the waves generated by sources often reflect off of various densities of material and then propagate back towards a receiver [5]. Collecting these measurements allows one to reconstruct the boundaries of and distinguish between large sections of dirt versus clay, for example. Generally these sources are placed above ground as physical restrictions limit their ability to be placed underground. Example 1.1b illustrates the set-up for a seismic imaging problem in IR tools [6]— the placements of the receivers and sources are subterranean, with the receivers populating the top and left boundaries and the sources on the right side. Users have the ability to specify certain parameters such as the number of sources and receivers, what underground image to re-create, and whether or not to explicitly form the matrix \mathbf{A} . Important to note is that once the number of sources and receivers are



Figure 1.2: Naive Reconstruction with 5% Level of Noise Added

fixed, the sources are equally spaced across the left and upper boundaries. To generate the example, we call the function `PRseismic` with the outputs \mathbf{b} , \mathbf{x}_{true} , and \mathbf{A} . While noise is not explicitly added to \mathbf{b} , a simple function can be utilized to introduce Gaussian Noise to the measurements. Using this function, we are able to gauge how accurate and efficient our numerical experiments are by having the true solution to compare against.

1.3 Computational Challenges to Inverse Problems

Despite the widespread utility of Inverse Problems, there exist substantial challenges that need to be resolved. Many of the problems that follow – examined through the lens of Seismic Tomography – are applicable to the examples mentioned in the section before.

We begin by examining a small example of seismic imaging – a 16x16 pixel image of plate subduction – generated using `PRseismic` within the MatLab package `IR Tools`. Figure 1.2, shown above, displays both the “true image” of the earth below ground level and the result of the naive reconstruction. For this given problem, a noise level of 5% is added into the measurement vector \mathbf{b} . Given the relatively small size of the matrix, $\mathbf{A} \in \mathbb{R}^{512 \times 256}$, the naive solution was calculated via the normal equations, $\mathbf{x}_{\text{naive}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$. The issue presented underscores the statistical ill-posed nature of inverse problems — without regularization, the recovered solution is dominated by noise [2]. This is further illuminated, in exercise 1.3 of Computational Uncertainty, by decomposing $\mathbf{x}_{\text{naive}}$ as a product of the pseudo-inverse of \mathbf{A} and \mathbf{b} . Using the truncated SVD outer-product definition for $\mathbf{A}^\dagger =$

$\sum_{i=1}^r \mathbf{v}_i \sigma_i^{-1} \mathbf{u}_i^\top$, the naive solution can be expressed as follows,

$$\begin{aligned}
\mathbf{x}_{\text{naive}} &= \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \right) \mathbf{v}_i \\
&= \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top (\mathbf{A}\mathbf{x} + \boldsymbol{\epsilon})}{\sigma_i} \right) \mathbf{v}_i \\
&= \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top (\mathbf{A}\mathbf{x})}{\sigma_i} \right) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i \\
&= \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top (\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \mathbf{x})}{\sigma_i} \right) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i, \text{ plugging in the SVD of } \mathbf{A} \\
&= \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top (\sum_{i=1}^r \mathbf{u}_i (\mathbf{v}_i^\top \mathbf{x}) \sigma_i)}{\sigma_i} \right) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i \\
&= \sum_{i=1}^r (\mathbf{v}_i^\top \mathbf{x}) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i, \text{ since } \mathbf{u}_i\text{s are } \perp \quad [2]. \tag{1.6}
\end{aligned}$$

Since the error is stochastic, it is necessary to take the expectation of the difference between the naive and true solution — in other words the mean-squared error of the naive estimator — in order to numerically quantify the solution quality. Then, following the derivation of

exercise 1.6,

$$\begin{aligned}
\text{MSE}(\mathbf{x}_{\text{naive}}) &:= \mathbb{E} \left[\|\mathbf{x}_{\text{naive}} - \mathbf{x}\|^2 \right] \\
&= \mathbb{E} \left[\left\| \sum_{i=1}^r (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i - \sum_{i=1}^n (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i \right\|^2 \right] \\
&= \mathbb{E} \left[\left\| \sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i - \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i \right\|^2 \right] \\
&= \mathbb{E} \left[\left\| \sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i \right\|^2 + \left\| \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i \right\|^2 \right], \text{ by pythagorean law} \\
&= \mathbb{E} \left[\sum_{i=1}^r \left\| \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i \right\|^2 + \sum_{i=r+1}^n \left\| (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i \right\|^2 \right], \text{ by pythagorean law} \\
&= \mathbb{E} \left[\sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right)^2 \|\mathbf{v}_i\|^2 + \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x})^2 \|\mathbf{v}_i\|^2 \right] \\
&= \mathbb{E} \left[\sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right)^2 (1)^2 + \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x})^2 (1)^2 \right], \text{ since } \mathbf{v}_i\text{s are normal} \\
&= \mathbb{E} \left[\sum_{i=1}^r \left(\frac{\mathbf{u}_i^T \boldsymbol{\epsilon}}{\sigma_i} \right)^2 \right] + \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x})^2, \text{ by linearity and } \mathbf{v}_i^T \mathbf{x} \text{ are constant} \\
&= \sum_{i=1}^r \left(\frac{\mathbb{E}[(\mathbf{u}_i^T \boldsymbol{\epsilon})^2]}{\sigma_i^2} \right) + \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x})^2, \text{ by eqns 1.4 and 1.5, } \mathbf{u}_i^T \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2) \\
&= \sum_{i=1}^r \left(\frac{\sigma}{\sigma_i} \right)^2 + \sum_{i=r+1}^n (\mathbf{v}_i^T \mathbf{x})^2, \quad [2] \tag{1.7}
\end{aligned}$$

shows that as the singular values of \mathbf{A} tend to 0, the effect of the noisy error term, $\boldsymbol{\epsilon}$, increases significantly. This is visually demonstrated in figure 1.3, which depicts the Picard Plot of the previously generated example. Indeed, the discrete Picard condition is satisfied as the SVD coefficients do not decay faster on average compared to the singular values, indicating that the problem is ill-posed [8]. In particular, the stability condition of Hadamard's criterion is violated as the recovered solution is sensitive to minor perturbations of data. By imposing a regularization term, however, the effects of this problem can be mitigated.

While this small example underscores the fundamental ill-posed nature of inverse prob-

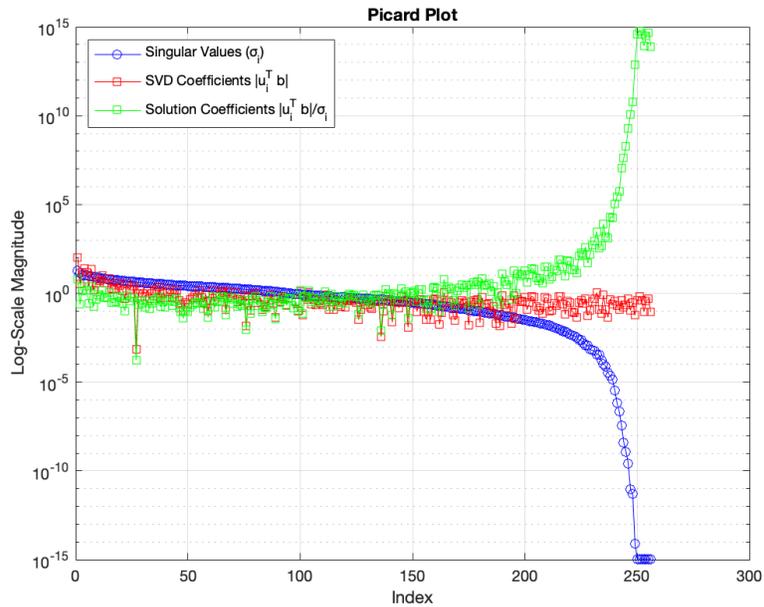


Figure 1.3: Picard Plot of 16x16 pixel example

lems, these challenges are further compounded when examining incredibly large linear systems — ones where the matrix \mathbf{A} cannot be constructed explicitly. In such instances, the process for solving the associated linear system through methods such as QR factorization is computationally infeasible. Furthermore, the methods for regularization parameter selection become constrained as most rely on the singular value decomposition of a matrix, meaning methods that exploit randomness are required. A natural extension then is gauging the accuracy of the recovered and regularized solution. However, the methods for uncertainty quantification [16, 2] rely heavily on not just finding a singular estimate but drawing samples from complex distributions, increasing the computational burden yet again. As such, a balance between accuracy and efficiency are always in interplay when solving an inverse problem.

Chapter 2

Computational Inverse Problems

2.1 Regularization Methods

Broadly speaking, incorporating regularization into an inverse problem helps to facilitate the solving process by imposing additional constraints and/or structure on the unknowns. In the context of inverse problems, including an appropriate regularization term can drastically improve the recovered solution as it can enforce stability or smoothness, making the problem better-posed [8]. These methods are known as spectral filtering methods and, as the name suggests, remove or mitigate the effects of the highly oscillatory singular vectors found in equations 1.6 and 1.7. The two main methods utilized are the truncated SVD and Tikonov regularization; the former identifies an index k such that $k < r$, where r is the rank of the matrix \mathbf{A} , and utilizes the first k vectors of the SVD to approximate the aforementioned matrix. This can be seen in modifying equation 1.6 in which, instead of summing from 1 to r , one sums from 1 to k instead, yielding

$$\begin{aligned} \mathbf{x}_\alpha &= \sum_{i=1}^k \left(\frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \right) \mathbf{v}_i \\ &= \sum_{i=1}^k (\mathbf{v}_i^T \mathbf{x}) \mathbf{v}_i + \sum_{i=1}^k \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i. \end{aligned} \tag{2.1}$$

Important to note is that k is selected to truncate the singular values that decay rapidly, meaning the effects of noise are mitigated.

For Tikonov regularization, the inclusion of a regularization parameter proportionally scales the singular values [2]. Tikhonov regularization can also be obtained by incorporating an additional term in the objective function of the minimization problem, yielding the corresponding least-squares formulation

$$\mathbf{x}_\alpha = \min_{\mathbf{x}} (\|\mathbf{b} - \mathbf{Ax}\|_2 + \alpha\|\mathbf{Lx}\|_2), \quad (2.2)$$

where α denotes the regularization parameter and $\mathbf{L} \in \mathbb{R}^{N \times N}$ is the regularization matrix that incorporates prior knowledge. From this lens, one can see that as α increases, more weight is placed on minimizing the latter term $\|\mathbf{Lx}\|$ in equation 2.2 as opposed to the former, penalizing solutions that over-fit the observations. Important to note is that while the additional regularized term is with respect to the 2-norm, other norms may be utilized to recover solutions with different characteristics i.e.; the 1-norm for sparse vectors. Furthermore, \mathbf{L} is generally the identity matrix, unless there is prior knowledge/assumptions of the problem [2]. For example, the discretized Laplacian operator \mathbf{L} is often used for imaging problems, where a given pixel value is generally correlated with surrounding pixel values.

The solution to this minimization problem, exercise 2.1 in Computational Uncertainty,

can be expressed as follows:

$$\begin{aligned}
\mathbf{x}_\alpha &= (\mathbf{A}^\top \mathbf{A} + \alpha \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b} \\
&= (\mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top + \alpha \mathbf{I})^{-1} \mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top \mathbf{b} \\
&= (\mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^\top + \alpha \mathbf{V} \mathbf{V}^\top)^{-1} \mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top \mathbf{b} \\
&= (\mathbf{V} (\boldsymbol{\Sigma}^2 \mathbf{V}^\top + \alpha \mathbf{I}) \mathbf{V}^\top)^{-1} \mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top \mathbf{b} \\
&= \mathbf{V} (\boldsymbol{\Sigma}^2 + \alpha \mathbf{I})^{-1} \mathbf{V}^\top \mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top \mathbf{b} \\
&= \mathbf{V} (\boldsymbol{\Sigma}^2 + \alpha \mathbf{I})^{-1} \boldsymbol{\Sigma}^2 \boldsymbol{\Sigma}^{-1} \mathbf{U}^\top \mathbf{b} \\
&= \mathbf{V} \boldsymbol{\Phi}_\alpha \boldsymbol{\Sigma}^{-1} \mathbf{U}^\top \mathbf{b},
\end{aligned} \tag{2.3}$$

where $\boldsymbol{\Phi}_\alpha = \text{diag}(\phi_1, \dots, \phi_r, 0 \dots, 0) \in \mathbb{R}^{N \times N}$ and $\phi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha}$. Note that, since $\mathbf{A}^\top \mathbf{A}$ is symmetric positive semi-definite and $\alpha > 0$, $\mathbf{A}^\top \mathbf{A} + \alpha \mathbf{I}$ is necessarily non-singular. Now, applying similar techniques to those used to derive equations 1.6 and 1.7, one can see that

$$\begin{aligned}
\mathbf{x}_\alpha &= \sum_{i=1}^r \phi_i (\mathbf{v}_i^\top \mathbf{x}) \mathbf{v}_i + \sum_{i=1}^r \phi_i \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i \\
&= \sum_{i=1}^r \left(\frac{\sigma_i^2}{\sigma_i^2 + \alpha} \right) (\mathbf{v}_i^\top \mathbf{x}) \mathbf{v}_i + \sum_{i=1}^r \left(\frac{\sigma_i^2}{\sigma_i^2 + \alpha} \right) \left(\frac{\mathbf{u}_i^\top \boldsymbol{\epsilon}}{\sigma_i} \right) \mathbf{v}_i,
\end{aligned} \tag{2.4}$$

and

$$\begin{aligned}
\text{MSE}(\mathbf{x}_\alpha) &= \sum_{i=1}^r \phi_i^2 \left(\frac{\sigma}{\sigma_i} \right)^2 + \sum_{i=1}^r (\phi_i - 1)^2 (\mathbf{v}_i^\top \mathbf{x})^2 + \sum_{i=r+1}^n (\mathbf{v}_i^\top \mathbf{x})^2 \\
&= \sum_{i=1}^r \left(\frac{\sigma_i^2}{\sigma_i^2 + \alpha} \right)^2 \left(\frac{\sigma}{\sigma_i} \right)^2 + \sum_{i=1}^r \left(\left(\frac{\sigma_i^2}{\sigma_i^2 + \alpha} \right) - 1 \right)^2 (\mathbf{v}_i^\top \mathbf{x})^2 + \sum_{i=r+1}^n (\mathbf{v}_i^\top \mathbf{x})^2,
\end{aligned} \tag{2.5}$$

where the regularization parameter effectively scales the oscillatory effects of the quickly decaying singular values [2].

Going back to the previous example, we can see the effects of regularization on the test problem below in figure 2.1. Indeed it is obvious that the regularized solution is a

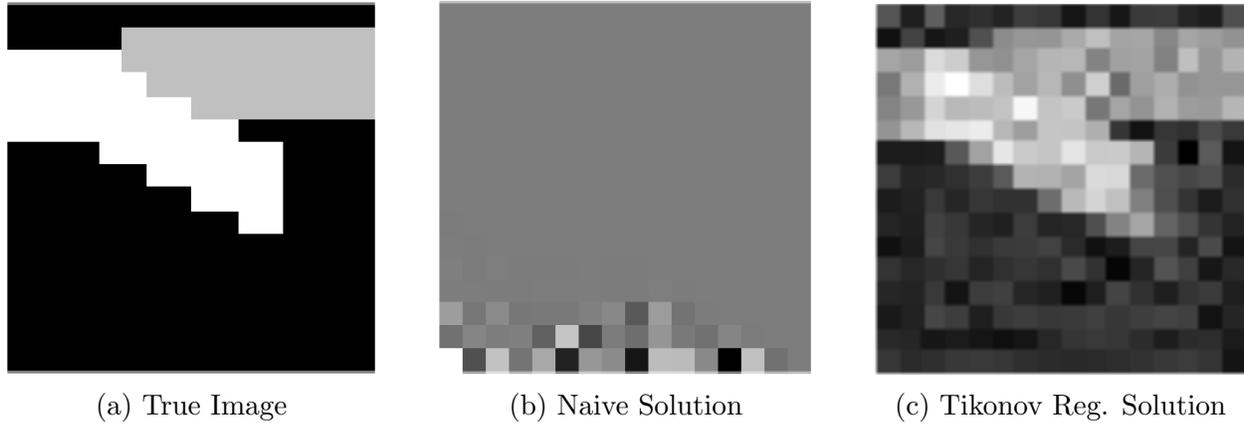


Figure 2.1: Comparison of Naive and Regularized Solutions

considerably better reconstruction of the image compared to the naive approach. This is further corroborated by the adjusted Picard plot below in which, with regularization, the

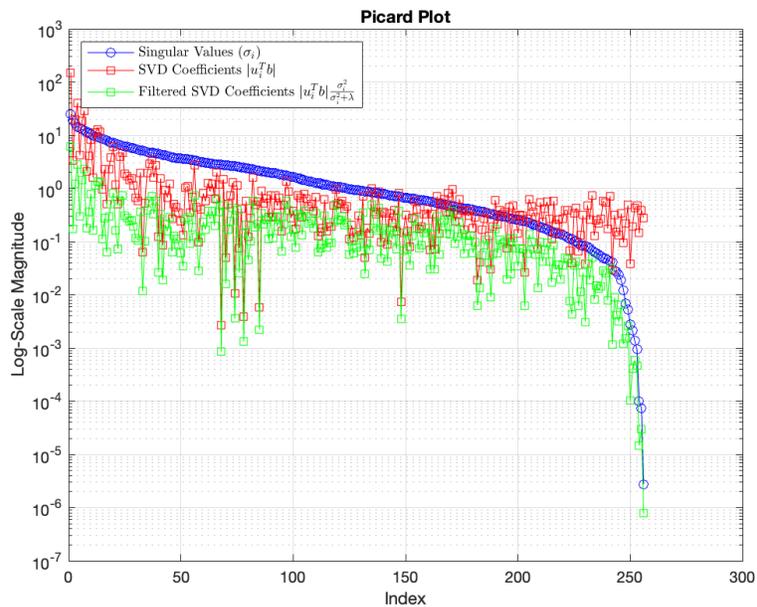


Figure 2.2: Adjusted Picard Plot of 16x16 pixel example

discrete Picard condition is no longer satisfied as the decay rate of the filtered SVD Coefficients matches that of the Singular Values. While the variance is controlled, the issue of bias becomes present, a natural effect of the bias-variance trade-off.

2.1.1 Regularization Parameter Selection Methods

This presents the need for apt regularization parameter selection that balances both over-fitting and under-fitting the given observations. However, the methods for doing so are dependent on the available information. Indeed, if the variance of the noise is known a priori, methods such as the discrepancy principle or the unbiased predictive risk estimator incorporate this into the estimation of the regularization parameter [2]; the former selects α that minimizes the unbiased predictive risk equation $\mathbb{E}(\|\mathbf{Ax}_\alpha - \mathbf{Ax}\|^2)$ and the latter chooses α such that the sum of squared residuals $\|\mathbf{Ax}_\alpha - \mathbf{b}\|^2$ is equal to the noise level, $M\sigma^2$. Under the discrepancy principle, a root-finding problem can then be formulated as follows,

$$\begin{aligned}
D(\alpha) &:= \|\mathbf{Ax}_\alpha - \mathbf{b}\|^2 - m\sigma^2 & (2.6) \\
&= \left\| \mathbf{A} \sum_{i=1}^r \phi_i^\alpha \left(\frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \right) \mathbf{v}_i - \sum_{i=1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \left\| \sum_{i=1}^r \phi_i^\alpha \left(\frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \right) \mathbf{U}\Sigma\mathbf{V}^\top \mathbf{v}_i - \sum_{i=1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \left\| \sum_{i=1}^r \phi_i^\alpha \left(\frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \right) \sigma_i \mathbf{u}_i - \sum_{i=1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \left\| \sum_{i=1}^r \phi_i^\alpha (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i - \sum_{i=1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \left\| \sum_{i=1}^r \phi_i^\alpha (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i - \sum_{i=1}^r (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i - \sum_{i=r+1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \left\| \sum_{i=1}^r (\phi_i^\alpha - 1) (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i - \sum_{i=r+1}^m (\mathbf{u}_i^\top \mathbf{b}) \mathbf{u}_i \right\|^2 - m\sigma^2 \\
&= \sum_{i=1}^r [(\phi_i^\alpha - 1) (\mathbf{u}_i^\top \mathbf{b})]^2 - \sum_{i=r+1}^m [\mathbf{u}_i^\top \mathbf{b}]^2 - m\sigma^2 & (2.7)
\end{aligned}$$

where the goal is to compute the root of $D(\alpha)$, which gives an estimate of the regularization parameter α .

In the case where the noise-level is unknown, the General Cross Validation (GCV) method and the L-curve criterion can be utilized to select the parameter. The broad idea for GCV is to approximate leave one out cross-validation (LOOCV), in which α is selected to minimize the predictive error, $V(\alpha) = \frac{1}{m} \sum_{i=1}^m \left([\mathbf{A}\mathbf{x}_\alpha^{[i]}]_i - \mathbf{b}_i \right)$ [2]. In this situation,

$$\begin{aligned} V(\alpha) &= \frac{1}{m} \sum_{i=1}^m \left([\mathbf{A}\mathbf{x}_\alpha^{[i]}]_i - \mathbf{b}_i \right) \\ &= \frac{1}{m} \sum_{i=1}^m \left(\frac{[\mathbf{A}\mathbf{x}_\alpha]_i - \mathbf{b}_i}{1 - [\mathbf{A}\mathbf{A}_\alpha]_{ii}} \right)^2 \\ G(\alpha) &\approx \frac{m \|\mathbf{A}\mathbf{x}_\alpha - \mathbf{b}\|^2}{[\text{tr}(\mathbf{I} - \mathbf{A}\mathbf{A}_\alpha)]^2} \end{aligned} \quad (2.8)$$

$$\begin{aligned} &= \frac{\sum_{i=1}^r [(\phi_i^\alpha - 1) (\mathbf{u}_i^\top \mathbf{b})]^2 - \sum_{i=r+1}^m [\mathbf{u}_i^\top \mathbf{b}]^2}{m - \sum_{i=1}^r \frac{\sigma_i^2}{\sigma_i^2 + \alpha}} \quad [2] \end{aligned} \quad (2.9)$$

where $G(\alpha)$ is the GCV function to be minimized.

The L-curve method is a visual heuristic in which the goal is to exploit the inverse relation of α on $\log \|\mathbf{x}_\alpha\|^2$ — the residual logarithm of the squared norm of the regularized solution — and $\log \|\mathbf{A}\mathbf{x}_\alpha - \mathbf{b}\|^2$ — the logarithm of the residual error squared of the regularized solution [2]. α is chosen to be the value that balances both terms, i.e., corresponding to the elbow of the L-curve; however this maybe difficult to discern visually. Thus, we do not further consider this approach. In summary, the methods of parameter selection rely heavily on the information available, specifically if the noise level is known or can be reasonably estimated [4].

2.2 Bayesian Framework Formalization

So far, the assumption has been made that \mathbf{x} is an unknown but fixed value — a common frequentist perspective. We now relax this assumption by modeling \mathbf{x} as a random variable

and consider a Bayesian framework [3, 10, 2]. Broadly, this formulation centers around Bayes' Theorem — $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ — and takes the form of

$$\begin{aligned} p(\theta|D) &= \frac{p(D|\theta)p(\theta)}{p(D)} \\ &\propto p(D|\theta)p(\theta), \end{aligned} \quad (2.8)$$

in which $p(\theta|D)$ is the posterior distribution, $p(D|\theta)$ is the likelihood distribution, and $p(\theta)$ is the prior distribution. Note, the purpose of the prior distribution is to incorporate additional information about the distribution from which the unknown is from. In the context of inverse problems, it is expressed as

$$p(\mathbf{x}|\mathbf{b}, \lambda, \delta) = \frac{p(\mathbf{b}|\mathbf{x}, \lambda)p(\mathbf{x}|\delta)}{p(\mathbf{b}|\delta, \lambda)} \propto p(\mathbf{b}|\mathbf{x}, \lambda)p(\mathbf{x}|\delta), \quad (2.10)$$

where

$$p(\mathbf{b}|\mathbf{x}, \lambda) \propto \lambda^{\frac{M}{2}} \exp\left(-\frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|\right) \quad (2.11)$$

and

$$p(\mathbf{x}|\delta) \propto \delta^{\frac{N}{2}} \exp\left(-\frac{\delta}{2} \mathbf{x}^\top \mathbf{L}\mathbf{x}\right). \quad (2.12)$$

Note, then that samples drawn from the posterior are sampled from a Gaussian random vector of the form

$$\mathbf{x}|\mathbf{b}, \lambda, \delta \sim \mathcal{N}((\lambda\mathbf{A}^\top \mathbf{A} + \delta\mathbf{L})^{-1}\lambda\mathbf{A}^\top \mathbf{b}, (\lambda\mathbf{A}^\top \mathbf{A} + \delta\mathbf{L})^{-1}), \quad (2.13)$$

where $(\lambda\mathbf{A}^\top \mathbf{A} + \delta\mathbf{L})^{-1}$ is the covariance matrix. Later in this section, proofs for the invertibility of the covariance matrix are provided. In these equations, $\lambda = \frac{1}{\sigma^2}$, $\delta > 0$ is a

parameter that tunes the prior, and both are related by $\alpha = \frac{\delta}{\lambda}$, the regularization parameter. This Bayesian approach provides a natural way to quantify uncertainty in the solution through the posterior distribution, while also enabling the incorporation of prior knowledge to stabilize ill-posed problems [2].

Having established the posterior distribution, we begin to derive a point estimate to characterize the recovered solution, known as the maximum a posteriori (MAP) estimator. Similar to the maximum likelihood estimator, the general principle is to choose \mathbf{x} that maximizes the probability of the observed data, with an additional regularization component. That is

$$\mathbf{x}_{\text{MAP}} = \arg \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{b}, \lambda, \delta) \quad (2.14)$$

or equivalently,

$$\begin{aligned} \mathbf{x}_{\text{MAP}} &= \arg \min_{\mathbf{x}} -\ln[p(\mathbf{x}|\mathbf{b}, \lambda, \delta)] \\ &= \arg \min_{\mathbf{x}} -\ln \left[\exp \left(-\frac{\lambda}{2} \|\mathbf{Ax} - \mathbf{b}\| \right) \exp \left(-\frac{\delta}{2} \mathbf{x}^\top \mathbf{Lx} \right) \right] \\ &= \arg \min_{\mathbf{x}} \left(\ln \left[\exp \left(\frac{\lambda}{2} \|\mathbf{Ax} - \mathbf{b}\| \right) \right] \ln \left[\exp \left(-\frac{\delta}{2} \mathbf{x}^\top \mathbf{Lx} \right) \right] \right) \\ &= \arg \min_{\mathbf{x}} \left[\frac{\lambda}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 + \frac{\delta}{2} \mathbf{x}^\top \mathbf{Lx} \right]. \end{aligned} \quad (2.15)$$

2.2.1 Different Priors

It was briefly mentioned during the regularization parameter section that additional knowledge of a given problem can be incorporated into \mathbf{L} by varying its structure. Another common and applicable prior used for imaging problems is the Gaussian Markov Random Field, which incorporates the connection between spatially-linked objects [2]. That is, a pixel generally is not drastically different from the ones surround it in an image. A simple formulation for the one-dimensional case follows, in which, contrary to the I.I.D. assumption, a data-point

x_i is Gaussian after conditioning on the points before and after. That is,

$$x_i | \mathbf{x}_{\partial i} \sim \mathcal{N} \left(\frac{x_{i-1} + x_{i+1}}{2}, (2\delta)^{-1} \right),$$

where $\mathbf{x}_{\partial i} = \{x_j | j \in \partial_i\}$, and $\partial_i = \{i-1, i+1\}$ is referred to as the neighborhood. An average of the two values are taken to impose the assumption that the estimated entry is in-between. The structure of \mathbf{L}_{1D} , the covariance matrix, then becomes

$$[\mathbf{L}]_{ij} = \delta \begin{cases} n_i, & i = j \\ -1, & j \in \partial_i \\ 0, & \text{otherwise} \end{cases} \quad \text{or visually, } \mathbf{L} = \delta \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{bmatrix}_{N \times N},$$

where n_i is the cardinality of the neighborhood. For entries x_1 and x_N , zero-boundary conditions are assumed; that is $x_0 = 0$ and $x_{N+1} = 0$. In the two-dimensional imaging case, $x_{ij} | \mathbf{x}_{\partial ij} \sim \mathcal{N}((x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1})/4, (4\delta)^{-1})$, where the neighborhood is $\partial_{ij} = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$, indicating that the pixels left/right and up/down adjacent are the ones that reveal information about x_{ij} . Expressing this relationship results in the following matrix, $\mathbf{L}_{2D} = \mathbf{I}_n \otimes \mathbf{L}_{1D} + \mathbf{L}_{1D} \otimes \mathbf{I}_n$, as, completing the derivation for exercise 4.3 in Computational Uncertainty,

$$\mathbf{L}_{2D} = \begin{bmatrix} \mathbf{1L}_{1D} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1L}_{1D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1L}_{1D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1L}_{1D} \end{bmatrix} + \begin{bmatrix} \mathbf{2I} & -\mathbf{1I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{1I} & \mathbf{2I} & -\mathbf{1I} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{1I} & \mathbf{2I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{2I} \end{bmatrix} \quad [2].$$

Since the block matrices are the same size we can add each entry wise, resulting in

$$\mathbf{L}_{2D} = \begin{bmatrix} \mathbf{L}_{1D} + 2\mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I} & \mathbf{L}_{1D} + 2\mathbf{I} & -\mathbf{I} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{L}_{1D} + 2\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{L}_{1D} + 2\mathbf{I} \end{bmatrix}.$$

Note that

$$\mathbf{L}_{1D} + 2\mathbf{I} = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \cdots & 0 \\ 0 & -1 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 4 \end{bmatrix}_{N \times N},$$

where the diagonal entries equaling 4 captures the effect of the additional spatial correlations, and the sub and super diagonal capture the dependence on the left/right adjacent pixels. For simplicity, the image recovered is assumed to be square, as is the case for the examples in the numerical example section. Now, applying the vec operator to an underlying image \mathbf{X} stacks its columns vertically, meaning the difference in indices between the top and bottom pixel differ by N . That is, in vectorized format, $x_{i-1,j} = x_{i-N}$ and $x_{i+1,j} = x_{i+N}$, where x_{i-N} and x_{i+N} are the $i + N$ and $i - N$ entries of the vectorized image. This corresponds exactly to the block structure found in \mathbf{L}_{2D} , where the $-\mathbf{I}$, along the sub and super diagonal ensures the n -spacing is maintained. More intuitively, the two Kronecker products function to apply the smoothing effect with respect to the four correlated pixels.

Another common prior is the edge-preserving prior which takes the form of

$$\mathbf{L} = \mathbf{D}_h^\top \Lambda_h \mathbf{D}_h + \mathbf{D}_v^\top \Lambda_v \mathbf{D}_v, \quad [2] \tag{2.16}$$

where $\mathbf{D}_h = \mathbf{I} \otimes \mathbf{D}$, $\mathbf{D}_v = \mathbf{D} \otimes \mathbf{I}$, and $\mathbf{\Lambda} \in \mathbb{R}^{(N-1)N \times (N-1)N}$ is a diagonal weight matrix that accounts for the change in pixel values. In both \mathbf{D}_h and \mathbf{D}_v , \mathbf{D} is the discrete gradient operator, which has the following structure

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix}_{N-1 \times N} .$$

In this instance, the diagonal entries of $\mathbf{\Lambda}$ encode the effect of the prior – more information as to how it is determined will follow. Under these assumptions, one can show equivalently that the new prior indeed acts as follows. We begin by examining the horizontal interaction by unfolding the Kronecker product and partitioning $\mathbf{\Lambda}$ into a block diagonal matrix of size $N \times N$ with block sizes $(N-1) \times (N-1)$,

$$\mathbf{D}_h = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D} \end{bmatrix}_{(N-1)N \times N^2} \quad \text{and} \quad \mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Lambda}_N \end{bmatrix} .$$

Note the transpose of \mathbf{D}_h is equivalent to taking the transpose of each block and then taking the transpose of the elements. Multiplying out the three matrices yields

$$\mathbf{D}_h^\top \mathbf{\Lambda} \mathbf{D} = \begin{bmatrix} \mathbf{D}^\top \mathbf{\Lambda}_1 \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^\top \mathbf{\Lambda}_2 \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}^\top \mathbf{\Lambda}_N \mathbf{D} \end{bmatrix} ,$$

where each diagonal block is a tri-diagonal matrix. Indeed, this can be seen as

$$\mathbf{D}^\top \mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{N \times N},$$

and since the diagonal matrix Λ_i only scales the entries, preserves the overall structure. Overall, the size of the resulting matrix is $N^2 \times N^2$.

For the vertical component, a similar argument follows, in which the decomposition yields a matrix of the following form

$$\mathbf{D}_v^\top \Lambda \mathbf{D}_v = \begin{bmatrix} \Lambda_1 & -\Lambda_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\Lambda_1 & \Lambda_1 + \Lambda_2 & -\Lambda_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\Lambda_2 & \Lambda_2 + \Lambda_3 & -\Lambda_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Lambda_{n-1} \end{bmatrix}_{N^2 \times N^2},$$

where Λ_i are $N \times N$ diagonal blocks. Note, the similar form compared to the GMRF prior, indicating that the transformation impacts the same index of points as intended.

To conclude this section, we examine the specific properties of the linear operator formed by $\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L}$, where \mathbf{L} is the prior precision (e.g., inverse of the covariance) matrix for a problem, and both δ and λ are strictly positive constants. Since these values do not affect the proofs to follow, they will be omitted. Under certain assumptions, we want to prove that the resulting matrix is positive definite (PD). To denoise, we begin by evaluating the properties of the matrix $\mathbf{A}^\top \mathbf{A}$, and then pivot to analyzing the different formulations for \mathbf{L} . Indeed, $\mathbf{A}^\top \mathbf{A}$, is trivially guaranteed to be positive semi-definite (PSD) as taking any vector \mathbf{x} in \mathbb{R}^n , $\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \|\mathbf{A} \mathbf{x}\|_2^2 \geq 0$. Positive definiteness is not guaranteed however, unless \mathbf{A} is full

column rank as, extending the previous proof, $\|\mathbf{Ax}\|_2^2 = 0$ iff $\mathbf{x} = \mathbf{0}$ as the $null(\mathbf{A}) = \{\mathbf{0}\}$. Thus, if $\mathbf{x} \neq \mathbf{0}$, $\|\mathbf{Ax}\|_2^2 > 0$. Note, the forward operator depends on the different methods or recording data, and as such, \mathbf{A} is not assumed to be PD.

Beginning with the simplest prior, the identity, it is easy to see that any positive scaling of \mathbf{I} is indeed positive definite. Since \mathbf{I} is PD, the matrix formed by adding the forward operator and the prior is also PD. For the specific case of the identity, adding a positively scaled identity value shifts the entire spectrum of $\mathbf{A}^\top \mathbf{A}$ by a positive value, forcing all zero eigenvalues to be positive. More generally, if \mathbf{B}, \mathbf{C} is PSD and PD, respectively, $\forall \mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top (\mathbf{B} + \mathbf{C}) \mathbf{x} = \mathbf{x}^\top \mathbf{B} \mathbf{x} + \mathbf{x}^\top \mathbf{C} \mathbf{x}$, where $\mathbf{x}^\top \mathbf{B} \mathbf{x} \geq 0$ and $\mathbf{x}^\top \mathbf{C} \mathbf{x} > 0$ implies the sum is necessarily greater than 0.

For the GMRF prior, $\mathbf{L} = \mathbf{L}_{2D}$, we first note that \mathbf{L}_{1D} is positive definite. Indeed, as for any \mathbf{x} non-zero, we have that $\mathbf{x}^\top \mathbf{L}_{1D} \mathbf{x} = (x_1^2 - 2x_1x_2 + x_2^2 + x_2^2 - 2x_2x_3 + x_3^2 + \dots + x_{N-1}^2 - 2x_{N-1}x_N + x_N^2) = \sum_{i=1}^{N-1} (x_i - x_{i+1})^2 > 0$, since at least one entry is non-zero. Now, $\mathbf{I}_N \otimes \mathbf{L}_{1D}$ and $\mathbf{L}_{1D} \otimes \mathbf{I}_N$ are both symmetric — $(\mathbf{I}_N \otimes \mathbf{L}_{1D})^\top = (\mathbf{I}_N^\top \otimes \mathbf{L}_{1D}^\top) = (\mathbf{I}_N \otimes \mathbf{L}_{1D})^\top$, with a similar argument following the other Kronecker product — and furthermore, their eigenvalues are necessarily positive as the eigenvalues of a matrix formed through a Kronecker product of two matrices is the multiplication of their respective eigenvalues. Thus, the sum of these matrices is guaranteed to be PD. A proof for the edge-preserving prior follows in the upcoming section, as the formation of the matrix is more involved. A natural consequence of positive definiteness, is also invertibility, and thus, the covariance matrix in equation 2.13 is indeed invertible too.

2.2.2 Conjugate Gradient

Having established the positive definiteness of the posterior covariance matrix, an incredibly helpful tool at our disposal is the Conjugate Gradient (CG) method for solving linear least squares problems. That is, given the ability to multiply by a symmetric positive definite matrix (SPD) \mathbf{M} , one is able to solve systems of the form $\mathbf{M}\mathbf{x} = \mathbf{c}$. The CG method is

an iterative Krylov subspace method that constructs a sequence of vectors that minimizes the residual $\|\mathbf{x}_n - \mathbf{x}^*\|_{\mathbf{A}}$. In doing so, convergence is guaranteed in r iterations, where r is the number of distinct eigenvalues of \mathbf{M} [19]. Setting $\mathbf{c} = \mathbf{A}^\top \mathbf{b}$, and $\mathbf{M} = \lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L}$, we obtain an efficient iterative method to solve for \mathbf{x} , the desired solution. Typically for iterative methods, a good approximate solution can be obtained in significantly fewer than n iterations, thereby making these methods, which are $\mathcal{O}(n^2)$, cheaper than a direct factorization or inversion.

Chapter 3

Computational Methods for UQ and Sampling

To highlight the need for computationally efficient methods for uncertainty quantification, and the associated sampling required, we present the following motivating example: consider recovering a seismic image on a 256×256 grid. Unfolding the estimated quantities into a vector results in a vector of 65,536 unknown parameters that need to be estimated. Formulating and solving the inverse problem often entails factorizing large matrices, of the size $65,536 \times 65,536$ for this example – such as the ones discussed in chapter 2. Naïvely, if one were to compute a direct factorization (for instance, via LU decomposition), the computational complexity would grow on the order of n^3 unknowns, rendering even a moderate-scale example prohibitively expensive. This scaling challenge underscores why more efficient or approximate methods for solving large linear systems are required. When considering the need to draw samples from the relevant distributions for uncertainty quantification, this problem is further exemplified and quickly become computationally infeasible. The methods described below reduce the computational complexity.

3.1 Randomized GCV Curve Estimation

Oftentimes, a good initial guess of the regularization parameter is needed when performing computational UQ, so we begin by focusing on the problem of regularization parameter selection. Under this new framework, finding the minimum of the GCV curve using methods such as the SVD decomposition to extract the singular values no longer becomes feasible. Re-examining the formula for the GCV curve, equation 2.8, a component to its evaluation is determining the trace of the matrix $\mathbf{I} - \mathbf{A}\mathbf{A}_\alpha$. Then to find the true minimizer for α , a matrix-matrix multiplication is required. However, the large size of \mathbf{A} means matrix-matrix multiplications should be avoided as the cost is $\mathcal{O}(n^3)$. Hence, randomized methods to estimate the trace of $\mathbf{A}\mathbf{A}_\alpha$ [2] can be utilized to avoid these costs. We do so by noting the following property, $\mathbb{E}[\mathbf{v}^\top \mathbf{C}\mathbf{v}] = \text{tr}(\mathbf{C})$, if \mathbf{C} is symmetric and \mathbf{v} is a white-noise vector with unit-variance ie; $\mathbb{E}[\mathbf{v}] = \mathbf{0}$ and $\text{Cov}(\mathbf{v}) = \mathbf{I}$. Indeed, this is true as one can see that,

$$\text{tr}(\mathbf{v}^\top \mathbf{C}\mathbf{v}) = \text{tr}(\mathbf{C}\mathbf{v}\mathbf{v}^\top),$$

by the cyclic property of the trace operator. Then taking the expectation of both values,

$$\mathbb{E}[\text{tr}(\mathbf{v}^\top \mathbf{C}\mathbf{v})] = \mathbb{E}[\text{tr}(\mathbf{C}\mathbf{v}\mathbf{v}^\top)],$$

and by linearity of the trace and expectation operators, we can swap the order, yielding

$$\begin{aligned} \mathbb{E}[\text{tr}(\mathbf{v}^\top \mathbf{C}\mathbf{v})] &= \text{tr}(\mathbb{E}[\mathbf{C}\mathbf{v}\mathbf{v}^\top]) \\ &= \text{tr}(\mathbf{C}\mathbb{E}[\mathbf{v}\mathbf{v}^\top]) \\ &= \text{tr}(\mathbf{C}\mathbf{I}) = \text{tr}(\mathbf{C}). \end{aligned}$$

It is important, however, to account for the variance as well, which can be found by first determining

$$\begin{aligned}\mathbb{E}[(\mathbf{v}^\top \mathbf{C} \mathbf{v})^2] &= \mathbb{E} \left[\sum_{i,j} (\mathbf{C}_{ij} v_i v_j) \sum_{k,l} (\mathbf{C}_{kl} v_k v_l) \right] \\ &= \mathbb{E} \left[\sum_{i,j,k,l} (\mathbf{C}_{ij} v_i v_j v_k v_l) \right] \\ &= \sum_{i,j,k,l} (\mathbf{C}_{ij} \mathbf{C}_{kl} \mathbb{E}[v_i v_j v_k v_l]).\end{aligned}$$

We now proceed by cases:

1: Two pairs of distinct indices

Then the possible combinations of distinct indices are a) $i = j \neq k = l$, b) $i = k \neq j = l$, and c) $i = l \neq j = k$.

(a) Then the summation becomes $\sum_{i \neq k} (\mathbf{C}_{ii} \mathbf{C}_{kk} \mathbb{E}[v_i v_i v_k v_k]) = \sum_{i \neq k} (\mathbf{C}_{ii} \mathbf{C}_{kk} \mathbb{E}[v_i^2 v_k^2])$.

Note that the variance for an individual element is 1 and each are independent, yielding $\sum_{i \neq k} (\mathbf{C}_{ii} \mathbf{C}_{kk})$. Then this summation can be expressed equivalently as $(\sum_i \mathbf{C}_{ii}) (\sum_i \mathbf{C}_{kk}) - (\sum_i \mathbf{C}_{ii})^2 = tr(\mathbf{C})^2 - \sum_i (\mathbf{C}_{ii}^2)$

(b) Similarly, the summation is then $\sum_{i \neq k} (\mathbf{C}_{ij} \mathbf{C}_{ij} \mathbb{E}[v_i v_j v_i v_j])$, yielding $\sum_{i \neq j} (\mathbf{C}_{ij} \mathbf{C}_{ij})$.

Again this can be re-expressed as $(\sum_i \mathbf{C}_{ij}^2) - (\sum_i \mathbf{C}_{ii}^2) = \|\mathbf{C}\|_F^2 - (\sum_i \mathbf{C}_{ii}^2)$. Since \mathbf{C} is symmetric, the Frobenius norm reduces to $tr(\mathbf{C}^2)$.

(c) Lastly, the summation takes the form $\sum_{i \neq j} (\mathbf{C}_{ij} \mathbf{C}_{ji})$, but since \mathbf{C} is symmetric,

$\mathbf{C}_{ij} = \mathbf{C}_{ji}$, meaning the summation becomes the one in case 1b.

2: All indices are equal

Then the summation becomes $\sum_i (\mathbf{C}_{ii} \mathbf{C}_{ii} \mathbb{E}(v_i^4))$, yielding a final form of $\sum_i (\mathbf{C}_{ii}^2 \mathbb{E}(v_i^4))$.

Note the expectation yields the kurtosis of an individual element, and will be denoted as μ_4 . Since the elements are IID, then the summation becomes $\mu_4 \sum_i \mathbf{C}_{ii}^2$

3: All indices are different, or exactly three indices are different

Then the summation equals 0. Note, exactly three distinct indices implies there exists one pair.

Altogether, this summation reduces to

$$tr(\mathbf{C}^2) - \sum_i (\mathbf{C}_{ii}^2) + tr(\mathbf{C}^2) - \sum_i \mathbf{C}_{ii}^2 + tr(\mathbf{C}^2) - \sum_i \mathbf{C}_{ii}^2 + \mu_4 \sum_i \mathbf{C}_{ii}^2,$$

and after collecting terms, yields

$$tr(\mathbf{C})^2 + 2tr(\mathbf{C}^2) + (\mu_4 - 3) \sum_i \mathbf{C}_{ii}^2.$$

Finally, the variance results in the following expression

$$Var [\mathbf{v}^\top \mathbf{C} \mathbf{v}]^2 = 2tr(\mathbf{C}^2) + (\mu_4 - 3) \sum_i \mathbf{C}_{ii}^2. \quad (3.1)$$

Recall, the properties did not specify the distribution the white noise vector was sampled from. Indeed, to minimize the variance then comes down to choosing a distribution that minimizes the kurtosis for a given index. The distribution that minimizes this is the Rademacher Distribution [2] — a probability distribution characterized as -1 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$ — as the kurtosis must be greater than or equal to one, and the kurtosis for the Rademacher Distribution is

$$\frac{\mathbb{E}[X^4]}{\sigma^2} = \frac{\frac{1}{2}(1)^4 + \frac{1}{2}(-1)^4}{1^2} = 1.$$

Thus it attains the lower bound. Using this as the choice for the random vector yields a final variance value of

$$Var [\mathbf{v}^\top \mathbf{C} \mathbf{v}]^2 = 2tr(\mathbf{C}^2) - 2 \sum_i \mathbf{C}_{ii}^2. \quad (3.2)$$

To further drive down the variance, we can draw multiple IID samples and take a simple average. In fact, let V equal the variance in equation 3.2. Then a simple average with n samples of the randomized trace estimates yields $tr(\mathbf{C})$ in expectation, with an improved variance of $\frac{1}{n}V$. This motivates an important question of how many samples to draw to recover a more precise estimation of the true GCV curve, which will be investigated via numerical experiments.

Having established the theory, in practice, computing $\mathbf{A}_\alpha \mathbf{v}$ requires solving the system $(\mathbf{A}^\top \mathbf{A} + \alpha \mathbf{I})\mathbf{A}_\alpha \mathbf{v} = \mathbf{A}^\top \mathbf{v}$. To be done efficiently, an iterative method like CG must be utilized to approximate the result [2]. In this way, estimating the trace requires one matrix-vector multiplication and one run of CG. Similarly, to estimate \mathbf{x}_α , another run of CG must be utilized.

3.2 MAP Estimation

Recall, calculating the MAP from the posterior distribution in equation 2.13 requires minimizing the following function for \mathbf{x} .

$$\frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \frac{\delta}{2} \mathbf{x}^\top \mathbf{L}\mathbf{x}.$$

Since the function is smooth, we are able to differentiate it with respect to \mathbf{x} . Setting the gradient equal to $\mathbf{0}$ yields

$$\lambda \mathbf{A}^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) + \delta \mathbf{L} = \mathbf{0},$$

which means we must solve the following linear system for \mathbf{x}

$$(\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})\mathbf{x} = \lambda \mathbf{A}^\top \mathbf{b}. \tag{3.3}$$

In this instance, CG should be utilized to solve for \mathbf{x}_{MAP} to maintain computational efficiency.

3.3 Edge Preserving Prior

Having introduced the relevant tools, we begin formulating the construction of the edge-preserving matrix. Without knowing a priori the difference in gradients, we leverage the algorithm below [2] to estimate a suitable prior. We begin by defining the following function

$$\mathbf{\Lambda}(\mathbf{x}) = \text{diag} \left(1/\sqrt{(\mathbf{D}_h\mathbf{x})^2 + (\mathbf{D}_v\mathbf{x})^2 + \beta\mathbf{1}} \right) \quad [2]. \quad (3.4)$$

Then to estimate and update the prior, we utilize Algorithm 1.

Algorithm 1 Iterative Edge-Preserving Reconstruction using IGMRF Priors

Require: Maximum iterations maxIts

Ensure: Reconstructed image \mathbf{x}

- 1: **Initialize:** $\mathbf{\Lambda}_h^0 = \mathbf{\Lambda}_v^0$
 - 2: **for** $k = 1$ to maxIts **do**
 - 3: Define \mathbf{L}_{k-1} as in equation 2.16
 - 4: Compute the MAP estimator \mathbf{x}_{k-1} using \mathbf{L}_{k-1}
 - 5: Update $\mathbf{\Lambda}_h^{k-1} = \mathbf{\Lambda}_v^{k-1} = \mathbf{\Lambda}(\mathbf{x}_{k-1})$
 - 6: **end for**
 - 7: Form \mathbf{L}_k using the final $\mathbf{\Lambda}_h^{k-1}$ and $\mathbf{\Lambda}_v^{k-1}$
 - 8: **return** \mathbf{L}_k
-

Note that the function $\mathbf{\Lambda}(\mathbf{x})$ returns a diagonal matrix whose entries are always non-negative. Furthermore, the prior is always symmetric. Evaluating the quadratic form now, for any vector \mathbf{x} ,

$$\begin{aligned}\mathbf{x}^\top \mathbf{L}\mathbf{x} &= \mathbf{x}^\top (\mathbf{D}_h^\top \boldsymbol{\Lambda} \mathbf{D}_h + \mathbf{D}_v^\top \boldsymbol{\Lambda} \mathbf{D}_v) \mathbf{x} \\ &= (\mathbf{D}_h \mathbf{x})^\top \boldsymbol{\Lambda} (\mathbf{D}_h \mathbf{x}) + (\mathbf{D}_v \mathbf{x})^\top \boldsymbol{\Lambda} (\mathbf{D}_v \mathbf{x}).\end{aligned}$$

Since $\boldsymbol{\Lambda}$ is diagonal with entries $\lambda_i \geq 0$, we have

$$(\mathbf{D}_h \mathbf{x})^\top \boldsymbol{\Lambda} (\mathbf{D}_h \mathbf{x}) = \sum_i \lambda_i ((\mathbf{D}_h \mathbf{x})_i)^2 \geq 0,$$

and similarly,

$$(\mathbf{D}_v \mathbf{x})^\top \boldsymbol{\Lambda} (\mathbf{D}_v \mathbf{x}) = \sum_i \lambda_i ((\mathbf{D}_v \mathbf{x})_i)^2 \geq 0.$$

Thus,

$$\mathbf{x}^\top \mathbf{L}\mathbf{x} \geq 0 \quad \text{for all } \mathbf{x}.$$

Although, adding together two positive semi-definite matrices does not guarantee a positive definite matrix, we can append an additional $\epsilon \mathbf{I}$, $0 < \epsilon \ll 1$, forcing the structure we want.

A key insight is that when $\|\mathbf{D}_h \mathbf{x}\|^2 + \|\mathbf{D}_v \mathbf{x}\|^2$ at a particular pixel is large—that is, when there is a sharp difference or high gradient—then the corresponding diagonal entry in $\boldsymbol{\Lambda}(\mathbf{x})$ becomes relatively small (since it is governed by the reciprocal of the square root). This means that, for large gradients, the weight in the penalty term is reduced, mitigating the tendency to smooth out sudden transitions in the image. Conversely, in regions where gradients are small, $\boldsymbol{\Lambda}(\mathbf{x})$ assigns larger weights, imposing a stronger penalty on minor fluctuations. Consequently, the effect of $\boldsymbol{\Lambda}(\mathbf{x})$ is to allow edges and pronounced features to remain intact while still enforcing smoothness in flatter areas, thus preserving sharp boundaries in the reconstruction.

3.4 Fixed Parameter Sampling

We now turn our attention to the Fixed Parameter Sampling component of our framework, wherein we seek to efficiently draw samples from the posterior distribution without resorting to direct factorization of $\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L}$. As noted earlier, attempting a Cholesky on $\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L}$ scales as $O(n^3)$, which, in most practical settings, is computationally prohibitive. Instead, we rely on the Conjugate Gradient (CG) algorithm to stochastically construct samples that, in expectation, belong to the desired posterior. Concretely, we consider the following system:

$$(\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L}) \mathbf{x} | \mathbf{b}, \lambda, \delta = \lambda \mathbf{A}^\top \mathbf{b} + [\sqrt{\lambda} \mathbf{A}^\top, \sqrt{\delta} \mathbf{R}^\top] \boldsymbol{\epsilon}, \quad (3.5)$$

where $\boldsymbol{\epsilon}$ is sampled from a standard normal distribution and \mathbf{R} is a Cholesky factorization of \mathbf{L} . From a Bayesian perspective, this linear system is precisely the mean–covariance decomposition of the Gaussian posterior one obtains for a regularized inverse problem of the form

$$p(\mathbf{x} | \mathbf{b}, \delta, \lambda) \propto \exp\left(\frac{\lambda}{2} \|\mathbf{A} \mathbf{x} - \mathbf{b}\|^2 + \frac{\delta}{2} \mathbf{x}^\top \mathbf{L} \mathbf{x}\right).$$

Indeed, solving the above system for \mathbf{x} yields

$$\mathbf{x} = (\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})^{-1} \lambda \mathbf{A}^\top \mathbf{b} + (\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})^{-\frac{1}{2}} \boldsymbol{\eta}, \quad (3.6)$$

where $\boldsymbol{\eta}$ is a white-noise vector whose distribution is identical to $\boldsymbol{\epsilon}$. By construction, \mathbf{x} is then a valid draw from the posterior distribution with mean

$$(\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})^{-1} \lambda \mathbf{A}^\top \mathbf{b} \quad \text{and covariance} \quad (\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})^{-1}.$$

Of course, the direct factorization $(\lambda \mathbf{A}^\top \mathbf{A} + \delta \mathbf{L})^{-\frac{1}{2}}$ remains impractical for large-scale problems. However, computing a Cholesky Factorization of \mathbf{L} is feasible as the priors discussed

so far have high sparsity, and strong structure in the form of bands. This brings the cost down significantly.

In practice, we compute the regularization parameter α using a method such as Randomized GCV minimization. We then compute and fix λ and δ by running another CG iteration to solve $\lambda = 1/\text{var}(\mathbf{b} - \mathbf{A}\mathbf{x}_\alpha)$, and $\delta = \alpha\lambda$. From there, we perform multiple CG solves—each applied to a different realization of the random vector ϵ —to generate a set of samples [2]. Finally, we estimate pointwise variance across our grid (or pixels) from these sample to gain a fuller picture of our reconstruction. In the case of the edge-preserving prior, Algorithm 1 is run for 10 iterations, upon which the \mathbf{L} matrix is fixed. After fixing the prior, samples are computed the same way as the Identity and GMRF priors above.

3.5 Hierarchical Gibbs

We now relax the assumption that λ and δ are known. Instead, we treat them as hyper-parameters that follow certain hyperprior distributions. In this case, we consider gamma distributions,

$$p(\lambda) \propto \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda) \tag{3.7}$$

$$p(\delta) \propto \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta). \tag{3.8}$$

Then the Bayesian formulation becomes

$$p(\mathbf{x}, \lambda, \delta | \mathbf{b}) \propto p(\mathbf{b} | \mathbf{x}, \lambda) p(\mathbf{x} | \delta) p(\lambda) p(\delta). \tag{3.9}$$

Since the posterior now depends on the two hyper-priors, we are no longer able to drop the constants $\delta^{\frac{N}{2}}$ and $\lambda^{\frac{M}{2}}$, as they affect the shape of the distribution. The full posterior then

becomes

$$\begin{aligned}
p(\mathbf{x}, \lambda, \delta | \mathbf{b}) &\propto \lambda^{\frac{M}{2}} \exp\left(-\frac{\lambda}{2} \|\mathbf{Ax} - \mathbf{b}\|^2\right) \delta^{\frac{\bar{N}}{2}} \exp\left(-\frac{\delta}{2} \mathbf{x}^\top \mathbf{Lx}\right) \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda) \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta) \\
&\propto \lambda^{\frac{M}{2} + \alpha_\lambda - 1} \delta^{\frac{\bar{N}}{2} + \alpha_\delta - 1} \exp\left(-\frac{\lambda}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 - \frac{\delta}{2} \mathbf{x}^\top \mathbf{Lx} - \beta_\lambda \lambda - \beta_\delta \delta\right). \tag{3.10}
\end{aligned}$$

Similarly the full-conditionals for the new hyper-parameters are

$$p(\lambda | \mathbf{b}, \mathbf{x}, \delta) \propto p(\mathbf{b} | \mathbf{x}, \lambda) p(\lambda) \propto \lambda^{\frac{M}{2} + \alpha_\lambda - 1} \exp\left(\left[-\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 - \beta_\lambda\right] \lambda\right) \tag{3.11}$$

$$p(\delta | \mathbf{b}, \mathbf{x}, \delta) \propto p(\mathbf{x} | \delta) p(\delta) \propto \delta^{\frac{\bar{N}}{2} + \alpha_\delta - 1} \exp\left(\left[-\frac{1}{2} \mathbf{x}^\top \mathbf{Lx} - \beta_\delta\right] \delta\right) \tag{3.12}$$

Note, then that the probability distributions for each are

$$\lambda | \mathbf{b}, \mathbf{x}, \delta \sim \Gamma\left(\frac{M}{2} + \alpha_\lambda, \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|^2 + \beta_\lambda\right) \tag{3.13}$$

$$\lambda | \mathbf{b}, \mathbf{x}, \delta \sim \Gamma\left(\frac{\bar{N}}{2} + \alpha_\delta, \frac{1}{2} \mathbf{x}^\top \mathbf{Lx} + \beta_\delta\right). \tag{3.14}$$

Since neither of the full-conditional probability distributions for λ and δ depend on the latter, they are conditionally independent. Furthermore, by the chain rule of probability, the new posterior is given by

$$p(\mathbf{x}, \lambda, \delta | \mathbf{b}) = p(\mathbf{x} | \mathbf{b}, \lambda, \delta) p(\lambda, \delta | \mathbf{b}). \tag{3.15}$$

This is critical as drawing samples from the posterior can now be split into two components, sampling from λ and δ , followed by $\mathbf{x} | \mathbf{b}, \lambda, \delta$. Furthermore, this implies that for every independent coupled sample of δ_k and λ_k , the coupled sample \mathbf{x}_k , δ_k and λ_k , is an independent sample from the posterior distribution. Thus, it suffices to analyze the δ and λ chains when assessing convergence of the Markov Chain Monte Carlo (MCMC) method [2]. This natural partition gives way to MCMC simulations, specifically Hierarchical Gibbs (HG).

Intuitively, HG partitions a general probability distribution into smaller chains, with the

idea that sampling from the original chain is computationally infeasible, and instead one should sample from another. Formally, the algorithm for inverse problems [2] is as follows

Algorithm 2 Hierarchical Gibbs

Require: Number of samples N_{samples} ; initial guesses $\mathbf{x}^{(0)}, \lambda^{(0)}, \delta^{(0)}$

Ensure: Samples $\{\mathbf{x}^{(n)}, \lambda^{(n)}, \delta^{(n)}\}_{n=1}^{N_{\text{samples}}}$

1: **for** $n = 1$ to N_{samples} **do**

2: **Sample** $\lambda^{(n)}$

 Conditional on $\mathbf{x}^{(n-1)}$ and \mathbf{b} draw $\lambda^{(n-1)} \sim \Gamma\left(\frac{M}{2} + \alpha_\lambda, \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|^2 + \beta_\lambda\right)$

3: **Sample** $\delta^{(n-1)}$

 Conditional on $\mathbf{x}^{(n-1)}$ and \mathbf{b} , draw $\delta^{(n-1)} \sim \Gamma\left(\frac{\bar{N}}{2} + \alpha_\delta, \frac{1}{2}\mathbf{x}^\top \mathbf{Lx} + \beta_\delta\right)$

4: **Sample** $\mathbf{x}^{(n)}$

 Conditional on $\lambda^{(n)}, \delta^{(n)}$, draw $\mathbf{x}^{k-1} \sim \mathcal{N}\left((\lambda_k \mathbf{A}^\top \mathbf{A} + \delta_k \mathbf{L})^{-1} \lambda_k \mathbf{A}^\top \mathbf{b}, (\lambda_k \mathbf{A}^\top \mathbf{A} + \delta_k \mathbf{L})^{-1}\right)$,

5: If $k = N_{\text{samples}}$ stop; otherwise $k = k + 1$.

6: **end for**

7: **return** $\{\mathbf{x}^{(n)}, \lambda^{(n)}, \delta^{(n)}\}_{n=1}^{N_{\text{samples}}}$

On a high-level, the goal of the derived MCMC chain is to fully explore the target chain's distribution, in our case, the posterior. As such the Markov chain generated must satisfy properties of irreducibility – there is a non-zero probability of transitioning from any given state to another within a number steps – aperiodicity – the chain does not get stuck in a fixed cycle – and ergodicity – if a chain is run long enough, the proportion of time it stays in a given state is equal to the probability determined by the stationary distribution [14]. Intuitively this makes sense as to compute and quantify the uncertainty of our distribution, it is necessary that our samples are representative of it. With these assumptions, the samples drawn converge in distribution to the distribution desired. Thus, after removing the initial values, referred to as the burn-in, the remaining samples should represent and characterize

the posterior distribution. This method makes the computations of complex non-Gaussian distributions feasible [2].

To analyze the chain’s behavior, a sequence of samples from our Markov chain can be collected, and visualized [18, 2]. Trace plots, created by plotting each iteration’s value of a parameter along a horizontal axis, effectively creates a timeline of the chain’s walk through the parameter space. This single graphic makes it easy to see if the chain rapidly “forgets” its initial condition and if it is wandering around a seemingly stationary region, which is a sign of good convergence. Autocorrelation Function Graphs (ACFs) measure how correlated each sample is with samples from earlier steps. Concretely, for lag k , we compute the sample correlation between $\{\theta_i\}$ and $\{\theta_{i+k}\}$ over a range of i . If the autocorrelation drops quickly to zero, we can infer that consecutive draws are effectively independent; if it remains high, that implies our chain moves slowly and we may need more iterations for robust inference. Similarly, generating a simple histogram of the δ and λ values for a given chain can demonstrate insights into how broad or peaked our posterior beliefs are – the former indicating uncertainty in the parameter value and the latter, the opposite.

Chapter 4

Numerical Experiments

In this section, we provide an overarching view of the three main numerical experiments, illustrating how the interplay between regularization parameter selection, prior assumptions, and computational strategies critically shapes the fidelity and stability of our reconstructions.

In Experiment 1, we focus on different parameter selection methods—such as GCV minimization, the Discrepancy Principle, UPRE minimization, and hybrid techniques—to identify which approach best balances noise suppression and detail preservation. A well-chosen initial parameter value is critical not only for non-hierarchical uncertainty quantification but also for effectively initializing hyper-priors in hierarchical one.

Experiment 2 places a strong emphasis on how various priors — smoothing, edge-preserving, and non-informative ones — impact seismic image reconstruction. Discretion in prior selection is critical as each prior encodes distinct structural assumptions into the problem, often yielding markedly different outcomes.

Finally, Experiment 3 examines the performance of MCMC methods in comparison to their non-hierarchical counterparts, particularly with respect to uncertainty quantification. We evaluate the effectiveness of each approach in capturing posterior variability while considering computational feasibility.

4.1 Experiment #1: Regularization Parameter Selection

In this experiment, we investigate the effects of various regularization parameter selection methods on the recovered solution for image reconstruction. A motivation for selecting the optimal parameter is shown below in figure 4.1, in which extreme examples for the regularization constant found in equation 2.2 are leveraged to illuminate their importance. Through the PRSeismic function [6], a 128x128 pixel sized image of subduction was generated. The recorded measurements, stored in the returned \mathbf{b} vector, were then perturbed with white noise, at a level of 10%. An α value of 100,000 was selected to exemplify over-regularization and similarly, an α value of $10e - 6$ was selected to underscore under-regularization. Finally, the GMRF prior is utilized. The image displayed is the MAP estimate found using equation 3.3 from the distribution derived in 2.13. Using the factorization method described in 3.5, and solving the system using conjugate gradient, 100 samples were drawn from the posterior distribution and the variance of each pixel was calculated. These were then visualized as the reconstructions found in figure 4.1.

What becomes immediately apparent from these examples is the critical role of parameter selection in inverse problems. In both extreme cases, the reconstructed solution deviates significantly from the true image: for over-regularization (large α), the image appears overly smooth or “blurry,” while for under-regularization (large α), it is overwhelmed by noise. Less obvious, however, is the subtle but crucial effect of the regularization parameter on the variance of the recovered image. Recalling the background discussion, increasing α in Equation 2.2 prioritizes smoothness and places greater weight on penalizing large or rapidly changing features. As a result, the reconstructed solutions are driven toward lower-frequency content, ultimately reducing the influence of measurement noise — noise introduced — and yielding lower-variance solutions. However, this reduction in variance comes at the cost of sacrificing sharp details, thereby blurring the final reconstruction. Consequently, determin-

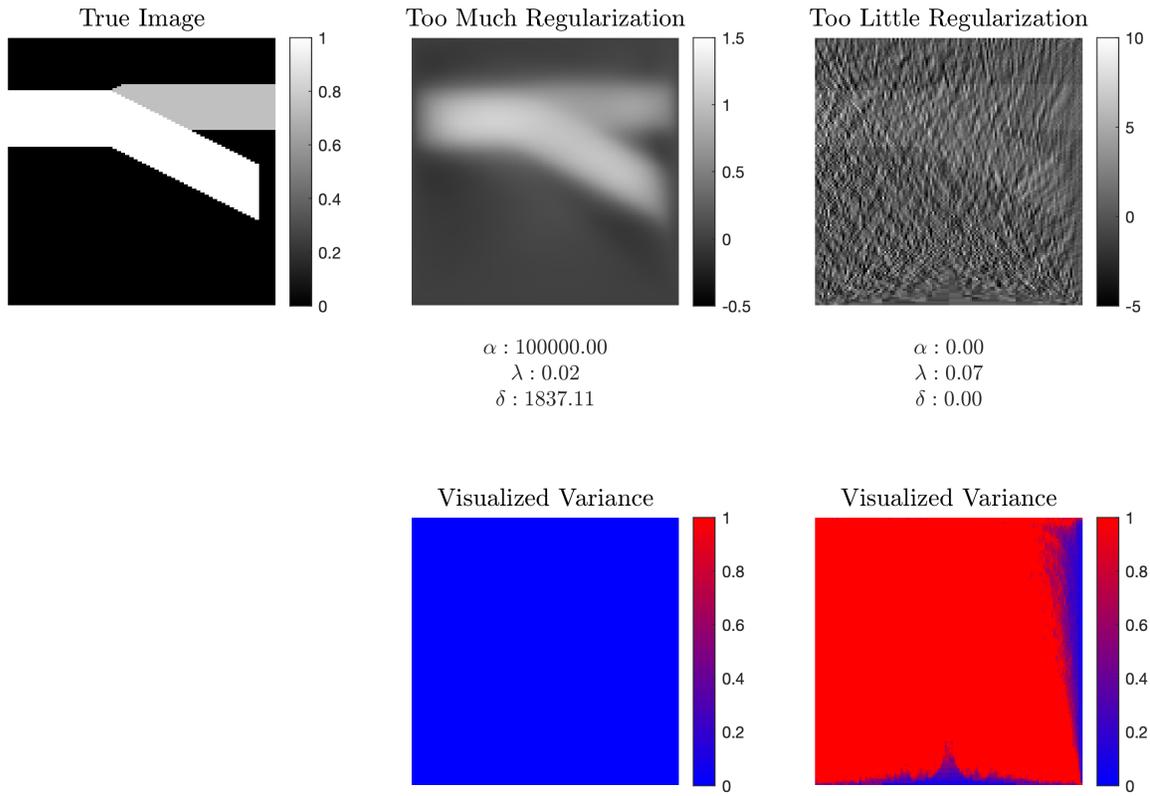


Figure 4.1: Over and Under-Regularization for Inverse Problems

ing an appropriate balance between noise suppression and detail preservation is essential, underscoring the importance of thoughtful parameter selection in achieving accurate and stable solutions.

Having introduced the detrimental effects of both over- and under-regularization, we now examine various regularization parameter selection methods and the reconstructed solutions they produce. This experiment follows a setup similar to the illustrative example: we introduce 10% noise in the measurements, use the conjugate gradient method to generate 100 samples from the posterior (defined using a GMRF prior), and calculate the variance across these samples. The key difference in this experiment lies in the choice of the initial regularization parameter α . We implement four strategies for parameter selection—GCV minimiza-

tion, UPRE minimization, the Discrepancy Principle, and Iterative Hybrid Methods—to investigate how each method influences both the recovered solutions and their associated uncertainties.

Shown below in figure 4.2, are the various reconstructions using the aforementioned methods. Most notable is the similarity in image reconstruction across the first four visualizations. This follows as the associated α values are similar — minimizing the randomized GCV function using Matlab’s built in `fminbnd` function yielded $\alpha \approx 162$, minimizing the randomized GCV function via simulated annealing yielded $\alpha \approx 168$, minimizing the UPRE yielded $\alpha \approx 197$, and using hybrid methods returned $\alpha \approx 380$. Note, the distinction between using different minimization methods for the GCV will be discussed further in the following subsection. The discrepancy principle, however, returns $\alpha \approx 1603$, significantly larger than the previous ones. Because the GMRF prior is already smoothing the image (thus absorbing some noise), the overall residual norm is driven down more than the discrepancy principle’s i.i.d. noise model expects. Consequently, to match the target noise level, the solver inflates the regularization parameter, leading to heavier smoothing than if the prior were non-informative. This high level of regularization in the discrepancy principle approach

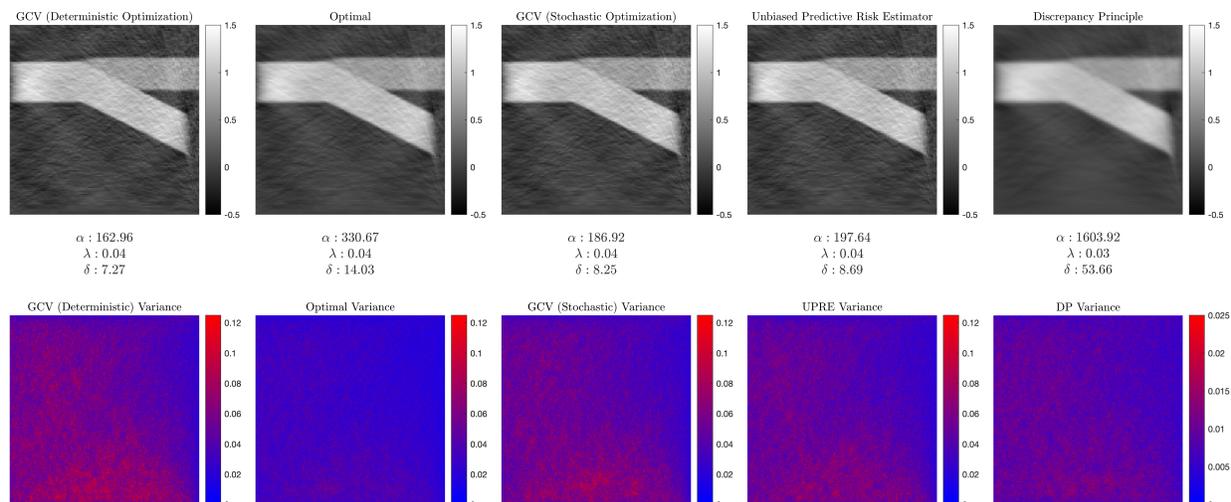


Figure 4.2: Different Regularization Parameter Selection Methods in Subduction Imaging

yields a visibly smoother reconstruction and, correspondingly, the lowest overall variance. In

Bayesian terms, a lower variance can be interpreted as higher certainty in the estimate. Here, the discrepancy principle enforces that the residual norm aligns with the known noise level, effectively distributing the noise uniformly across the measurement vector. By leveraging this explicit knowledge of the noise, the method can strongly penalize large fluctuations, thus substantially reducing noise. However, the trade-off is that the image may be oversmoothed, potentially masking finer details. Ultimately, whether this smoother result is viewed as an improvement depends on the relative importance of noise suppression versus the preservation of high-frequency features in the specific application. This will be underscored in a later example.

4.1.1 GCV: A Deeper Dive

Leveraging different iterative methods to estimate the randomized GCV function is motivated primarily by figure 4.3. This graph plots the true GCV for a small-scale problem – obtainable through direct evaluations – against the required randomized methods for making evaluation of the GCV function feasible. In particular, due to the generally large size of the matrix, $[\text{tr}(\mathbf{I} - \mathbf{A}\mathbf{A}_\alpha)]$ and $\|\mathbf{A}\mathbf{x}_\alpha - \mathbf{b}\|$ of equation 2.8 become difficult to evaluate outright, if not impossible given that an SVD of A is required. As such, any plethora of randomized trace methods can be implemented to retrieve an estimate of the former term, while a CG evaluation can make the latter accessible. Notably, incorporating these randomized samples can introduce noise into the objective, creating a landscape akin to the red curve in the figure. Deterministic optimization techniques, such as Nelder–Mead, often become trapped in local extrema under these conditions [17]. By contrast, stochastic methods like simulated annealing can better explore the noisy function space, as they inject randomness to escape local minima or maxima [17, 12]. Nonetheless, these methods are also sensitive to noise levels: averaging multiple randomized trace estimates can mitigate this issue, but if the underlying function—such as a harmonic or highly oscillatory one—presents a complex shape, stochastic methods may still face challenges [17]. Fortunately, for typical GCV applications,

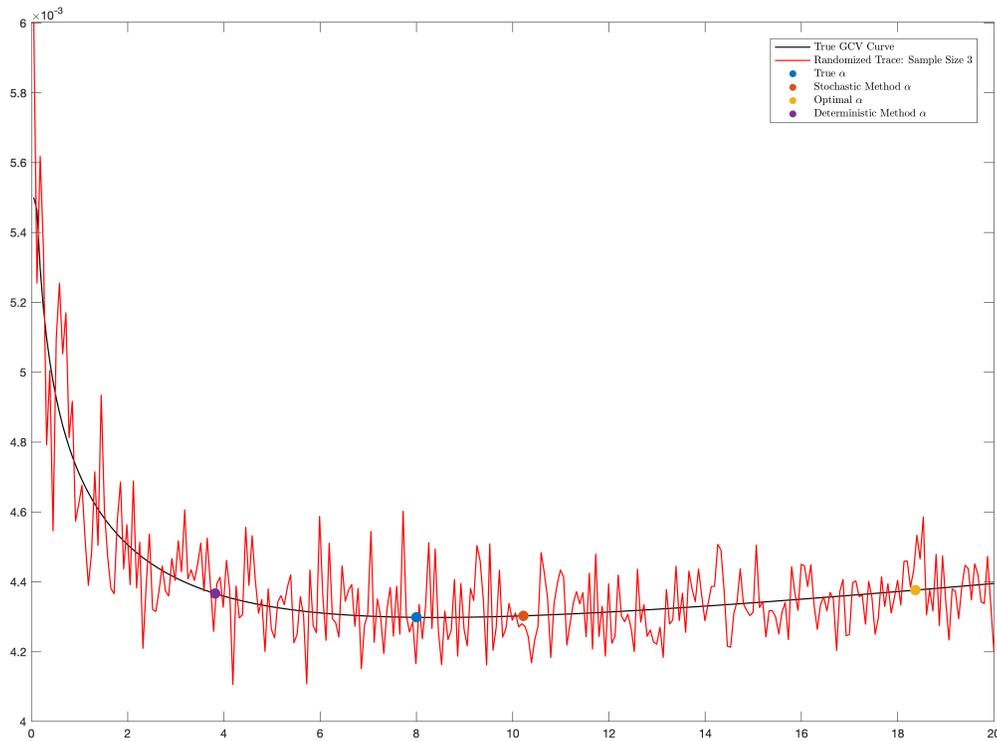


Figure 4.3: Randomized GCV Curve Minimization

the underlying function tends toward a parabolic shape, making it more amenable to robust optimization strategies.

Continuing in this vein, computing multiple randomized trace estimates and averaging across each iteration of the stochastic GCV function reduces the noisiness of the trace estimator. This averaging helps to mitigate the variability that naturally arises from using random samples. In practice, the inherent randomness can lead to fluctuations that might obscure the true behavior of the function, making it challenging to determine optimal regularization parameters. By averaging over several estimates, the overall variance is reduced, leading to a more stable and reliable approximation. Logically, as more samples are drawn and averaged, the approximation converges to the true function value, ensuring that the stochastic GCV approach accurately reflects the underlying model dynamics.

We introduce a different seismic example to highlight the difference in regularization

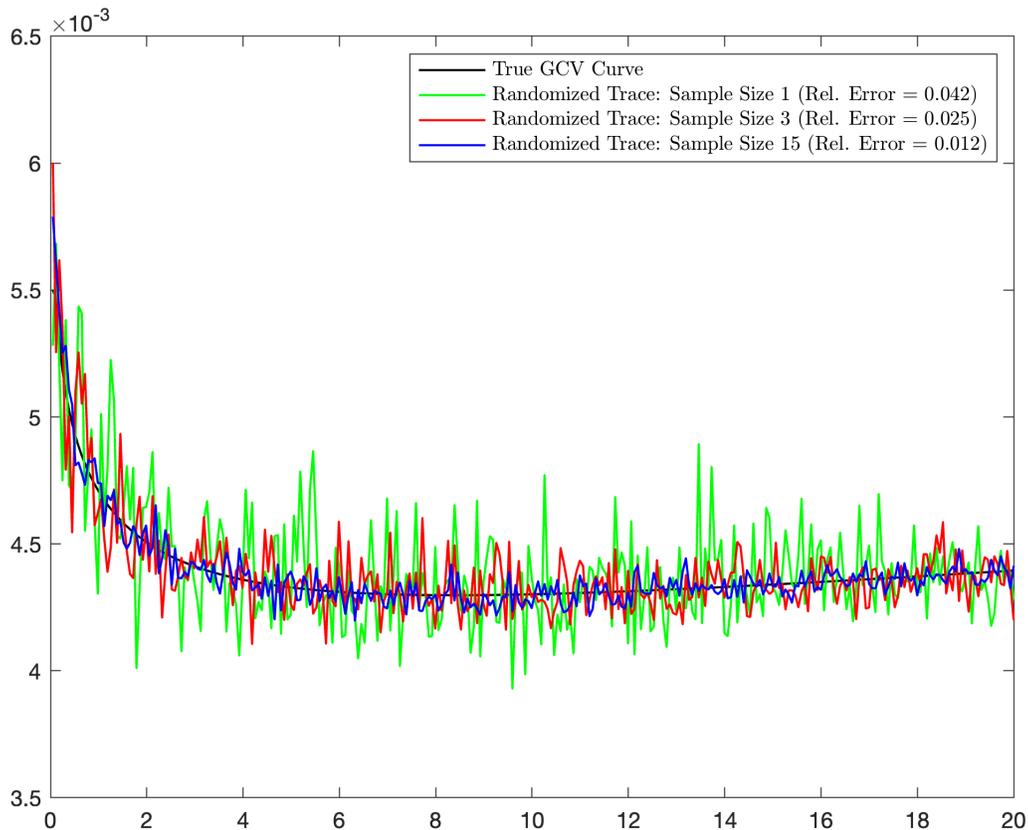


Figure 4.4: GCV Curve Approximation Using Randomized Methods

parameter selection pertaining to more complex images. Using `PRSeismic` [6], the threezone parameter is selected, yielding the following true image.

Implementing the same methods for parameter selection yields the results seen in figure 4.6. Notably, across all approaches, the recovered images appear less distinct and more visually ambiguous. In this setting, the role of variance becomes particularly important as lower variances in a reconstruction can signify regions that remain robust to noise and model uncertainties, making it easier to identify structural features that are consistently recovered. In contrast, areas with higher variance may indicate instability or insufficient information, underscoring the potential for errors. As such, the discrepancy principle provides a more robust insight into what the image looks like. Finally, the areas of higher variance along the center of the bottom is a result of the placement of sensors and receivers inherent

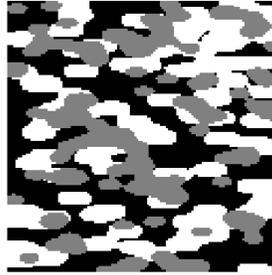


Figure 4.5: Three Zone True Image

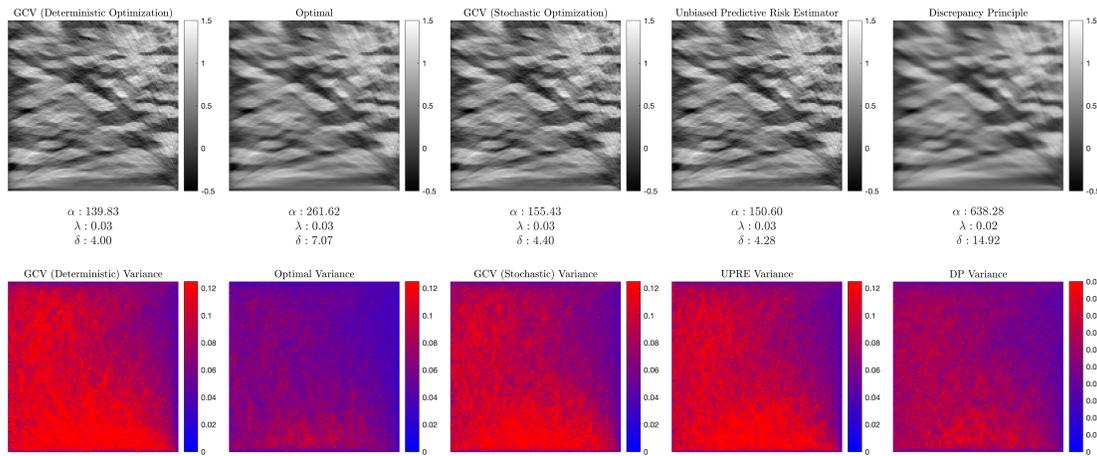


Figure 4.6: Three Phases: Different Regularization Parameter Selection Methods

to the PRseismic package, and the points of intersection between the two [6].

4.2 Experiment #2: Different Priors

In this experiment, we examine how different prior assumptions — specifically, a GMRF, identity, and edge-preserving prior — affect both the reconstructed image quality and the associated variance. For these priors, zero boundary conditions are utilized, meaning pixels past the image are assumed to be 0. By varying the problem setup, we gain insight into how each prior shapes the trade-off between noise suppression and feature preservation.

We begin by noting that the choice of prior influences the regularization operator L , which in turn affects the selection of the regularization parameter α . The selection method used — GCV minimization via simulated annealing — remains unchanged across the different

experiments. As in the previous one, the recorded measurements are perturbed with 10% white noise, and the variance is computed on a pixel-by-pixel basis from 100 samples drawn from the posterior. Presented below in figure below are the three different images that will be used to assess the effect of different priors on reconstruction.

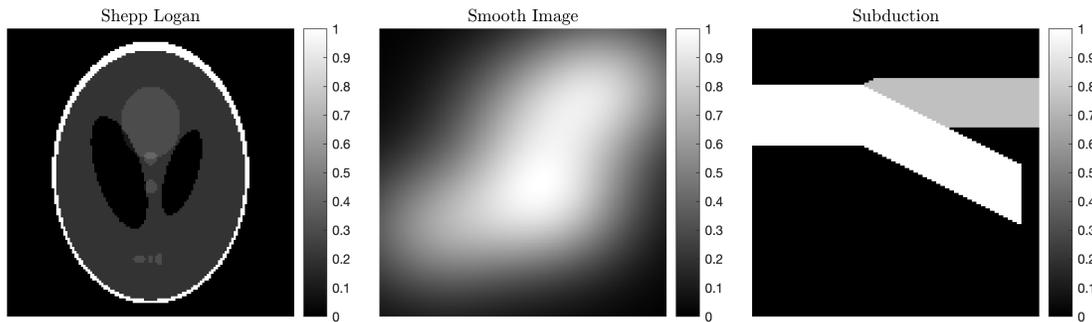


Figure 4.7: Numerical Experiment #2 True Images

These examples are 128x128 pixel images generated from the PRTomography, PRSpherical, and PRSeismic packages [6], respectively, within the Iterative Regularization Tools package.

We begin with the simplest case: subduction. From figure 4.8, we can see that the resulting MAP reconstructions across the three different priors are visually similar to the true image. The most noticeable difference is found in the GMRF Prior, in which the reconstruction exhibits artifacts near the left boundary, likely due to the imposition of the zero boundary conditions – the inherent smoothing of the GMRF prior creates a gradient that transitions from dark to light for lighter section of the sub-ducting plate. This is further corroborated both in the reconstructions using the identity and edge-preserving priors; in the former, no assumptions are made regarding the spatial correlation between neighboring pixels and therefore a naturally grainy image but artifact-free reconstruction is recovered; in the latter, the edge-preserving prior accounts for spatial correlation, however, the sharp transition from dark to light – which is analogous to a rapidly changing gradient – penalizes the left border, creating a sharp edge. In this simple problem, the difference in error norms among the three options corroborates our results. The differences in the variance reveal more nuanced details, largely due to the additional, or lack of, spatial structure imposed upon by

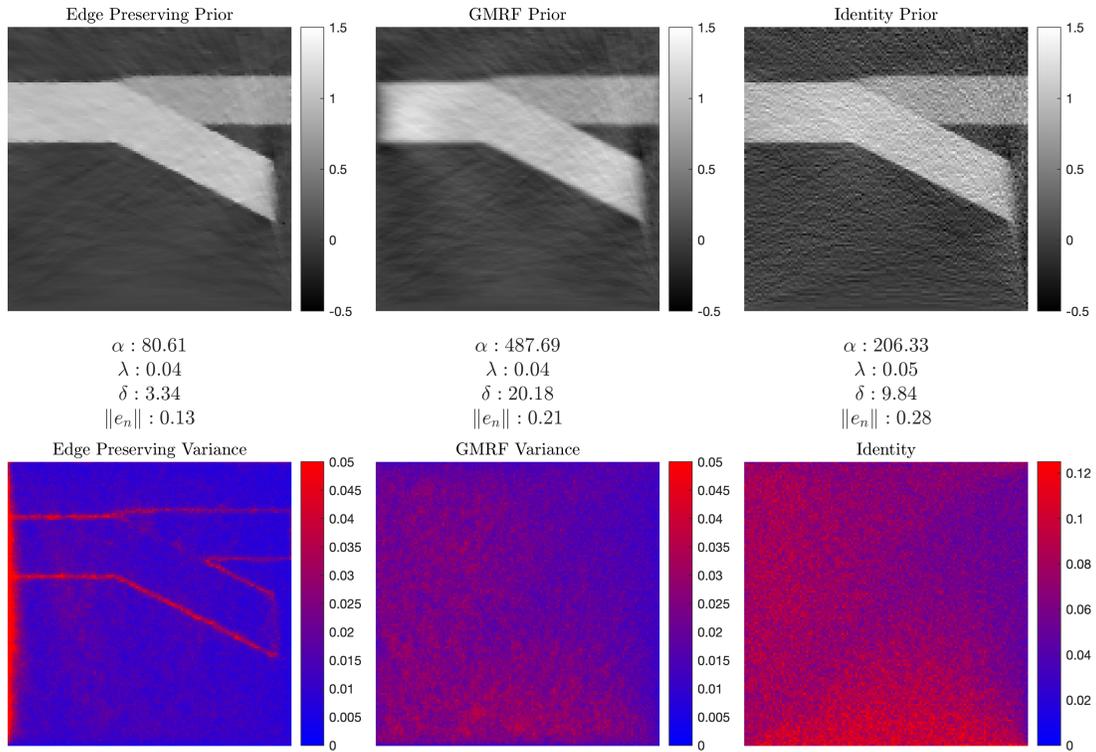


Figure 4.8: Different Priors in Subduction Imaging

the different priors. Specifically, by correlating neighboring pixels, the GMRF can reduce uncertainty in regions where features are relatively smooth, thereby producing lower variance in those areas. Conversely, abrupt transitions or boundary conditions may introduce higher variance, as the prior struggles to reconcile the spatial continuity with sharp changes in pixel intensity. In the case of the edge preserving prior, areas of relative uniformity are highlighted and coerced to be similar values, leading to lower variances within large structures such as the plate and the darkness around it. Again, the imposition of zero-boundary conditions manifests itself on the left border as that area exhibits very large variances. These appear less on the right one as those are where the majority of detectors are placed, meaning more information and measurements are recorded. In contrast to both, the identity prior lacks these spatial correlations, often leading to higher variances across the image.

In instances where an image is smooth, edge-preserving priors naturally perform poorly. Most evidently, in figure 4.9, artifacts are generated when the edge preserving prior is used

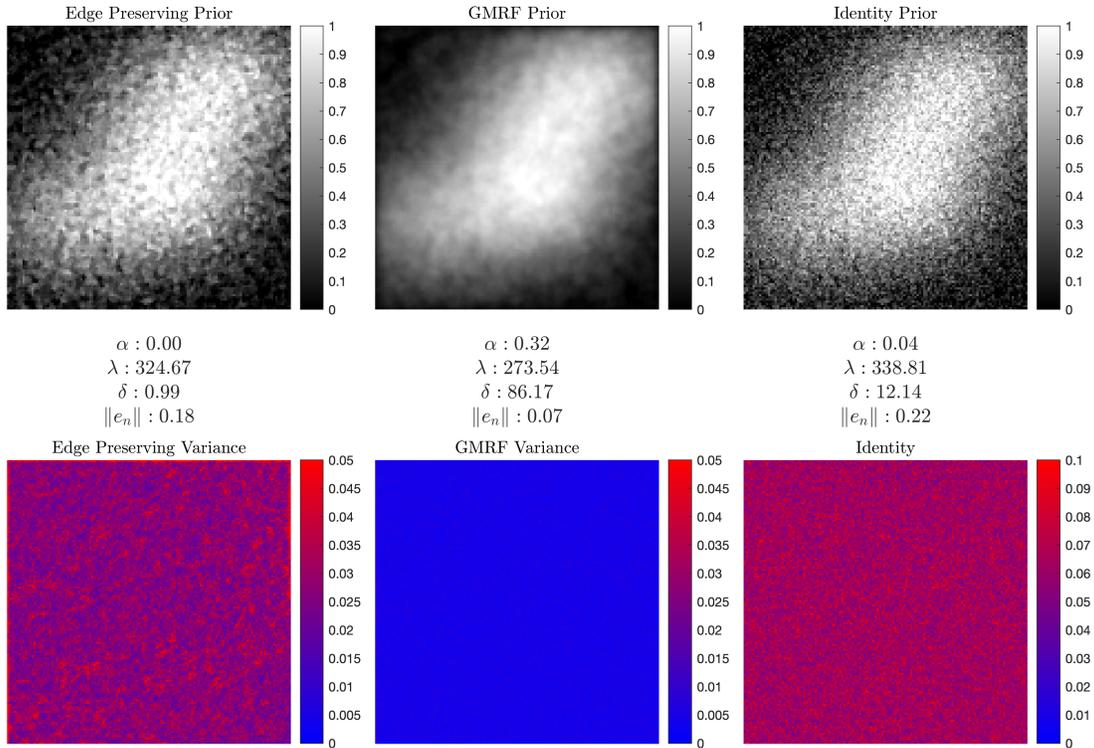


Figure 4.9: Different Priors in Smooth Imaging

– the bias towards sharper edges generates well defined circular clusters in the image that are not present in the original. A similar effect can be seen with the usage of the GMRF prior – however, the smoothness of the image is captured well. The reconstruction using an Identity Prior for this problem is again grainy, but remains an accurate reconstruction of the underlying smooth image. Correspondingly, one can see that the variance of the image generated with an edge-preserving prior is significantly worse compared to that of the GMRF prior.

While the error norm associated for both images is seemingly always the lowest, this is attributable to the fact that the nature of these examples are imaging problems, meaning spatial correlation between pixels is expected to some degree. However, as seen in the previous cases, unless a specific structure to the correlation is known – smoothness or edginess of an image — the identity imposes the least amount of artifacts. That is to say, an identity prior is best applied to situations where minimal information is known, reducing the impacts

of artifacts.

For the final example, we utilize a more realistic and impactful example, the Shepp Logan reconstruction. This image has become a benchmark for evaluating different regularization

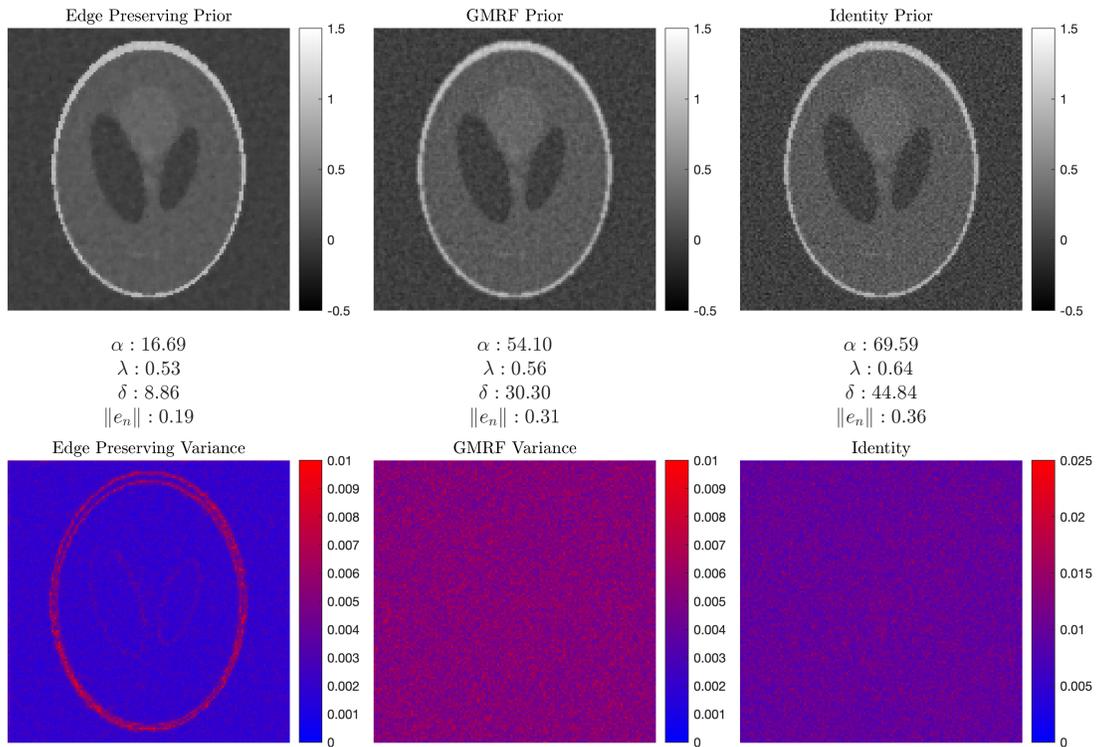


Figure 4.10: Different Priors in C.T. Imaging

parameter methods as it highlights the need for clear and artifact-free reconstructions. Furthermore, understanding where regions of uncertainty exist in the image is critical to determine its reliability. By inspection, all capture the three distinct ellipsoids within the image, although struggle in regards to the smaller and lighter ones. Although the error norms differ for each method, they are not good proxies of each reconstruction as not only is the true solution not normally known, but also, on a practical level, the images yield nearly visually indistinguishable results. While an image maybe grainier than another, likely contributing to a higher error, these patterns are consistent with the usage of different priors. Examining the variance, it is clear the edge-preserving prior offers more insight into where regions of accuracy can be trusted. Specifically, the borders between large transitions of light and dark

are highlighted, offering insight into the image itself. Specifically, since edge-preserving priors encode large gradient changes, as more samples are drawn from the posterior, the noise perturbs these values more, increasing the variance.

4.3 Experiment #3: MCMC Methods

In the final experiment, we relax the assumption that λ and δ are known. Instead, we assume that both parameters are distributed according to gamma distributions with $\alpha_\delta = \alpha_\lambda = 1$ and $\beta_\delta = \beta_\lambda = 10^{-4}$. A noise level of 10% is introduced to the data. For both the MCMC method and the fixed parameter approach, a total of 5000 iterations are drawn. In the case of the MCMC method, the initial values for α , δ , and λ are determined via GCV minimization and subsequently refined through the priors, whereas the fixed parameter method relies solely on GCV minimization for initialization. To ensure computational feasibility, the Preconditioned Conjugate Gradient (PCG) iterations are capped at 150. A burn-in period of 10% is applied, yielding 4500 effective samples from the posterior. Variance is computed by evaluating the pixel-wise variance across the remaining samples, and the mean is calculated in a similar manner, given the absence of a conventional method to compute a maximum a posteriori (MAP) estimate. The variance for the fixed parameter scenario is determined using the same approach.

With the experimental framework in place, we now turn to the reconstructed images and variance maps shown in figure 4.11, which contains the recovered solutions obtained through Hierarchical Gibbs Sampling and Fixed Parameter Sampling. By inspection, the recovered solutions using Hierarchical Gibbs with a GMRF prior and the fixed parameters analog yield very different images, with the former displaying more signs of blurring.

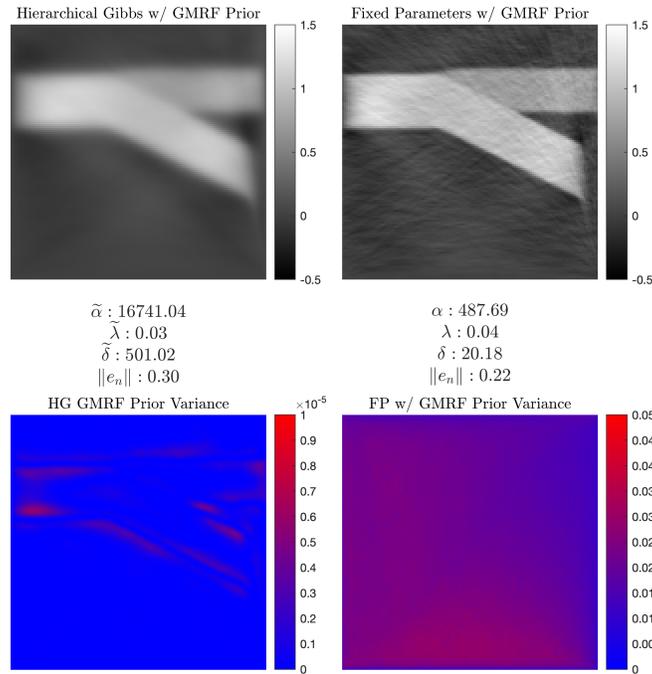


Figure 4.11: MCMC V.S. Fixed Parameter Methods with a GMRF Prior

This can mainly be attributed to the iterative nature of the HG sampler — the repeated sampling and exploration of the posterior distribution reinforces the spatially correlated prior, imposing a stronger smoothing effect compared to the fixed parameter method. Indeed this is corroborated by the δ values, where the HG sample’s median value is 501 compared to the 20 found via the methods in section 4. The over-smoothing effect does, however drastically reduce the variance of the image, with the majority of values being $\ll 10^{-5}$.

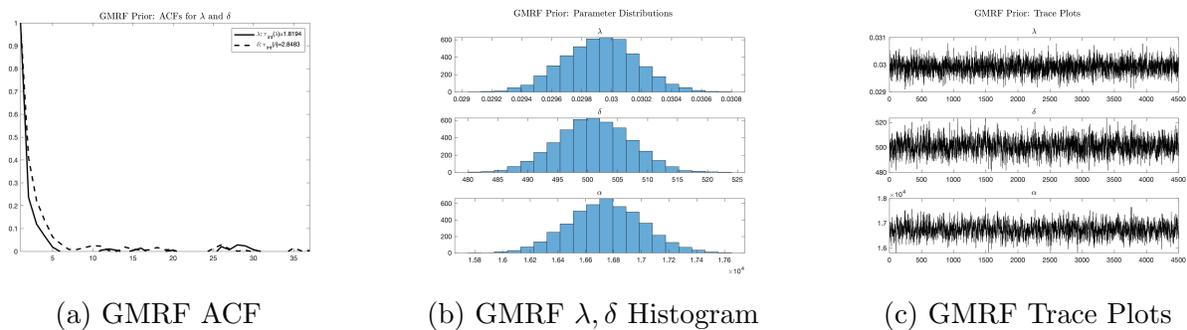


Figure 4.12: Markov Chain Diagnostics using the GMRF Prior

The three plots in figure 4.12 indicate that our HG sampler did what it intended to

do as the sharp drop off in the autocorrelation function graph (ACF) indicates that the samples generated afterwards are independent of one another. High-mixing is generally an indication that, given a long enough chain, the generated distribution will have converged to the stationary one. The uni-modal distributions found in the histograms indicate that the parameters computed via the hierarchical sampling are generally accurate. The medians of each distribution are listed on the reconstructed images. Finally, the trace plots appear to be aperiodic as there is no macroscopic sinusoidal trend.

The next experiment run utilizes an Identity prior as opposed to a GMRF one, and the results can be seen in figure 4.13. For the identity prior, since no assumptions are made on the pixels correlation, both images share a similar grainy appearance, with the samples drawn using MCMC methods a bit more so. This difference in overall smoothness of the image may result from the MCMC chain exploring a more uncertain space compared to sampling with fixed parameters, with pre-selected "ideal parameters". Again, with the increased exploration of the posterior distribution by relaxing the assumptions on the hyper-parameters, the variance is drastically lower than that using fixed parameter methods. Furthermore, in comparison with the GMRF priors, there is no additional artifacts along the left boundary of the image.

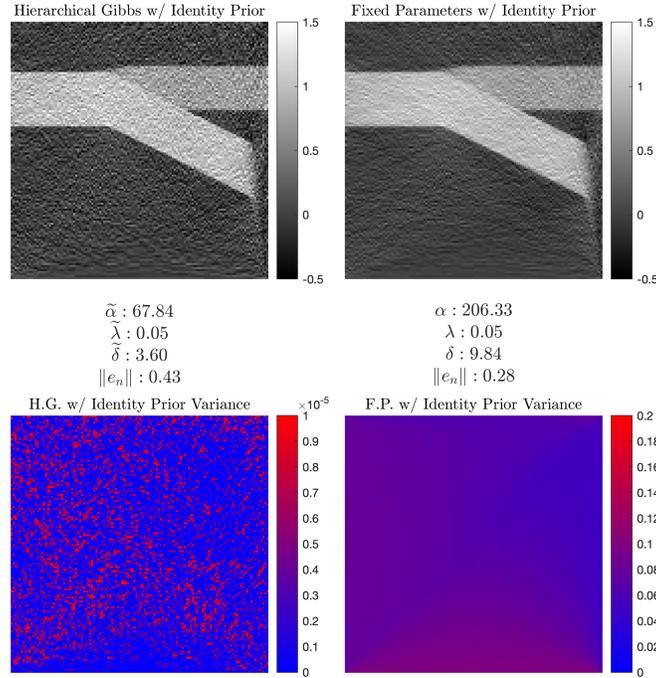


Figure 4.13: MCMC V.S. Fixed Parameter Methods with an Identity Prior

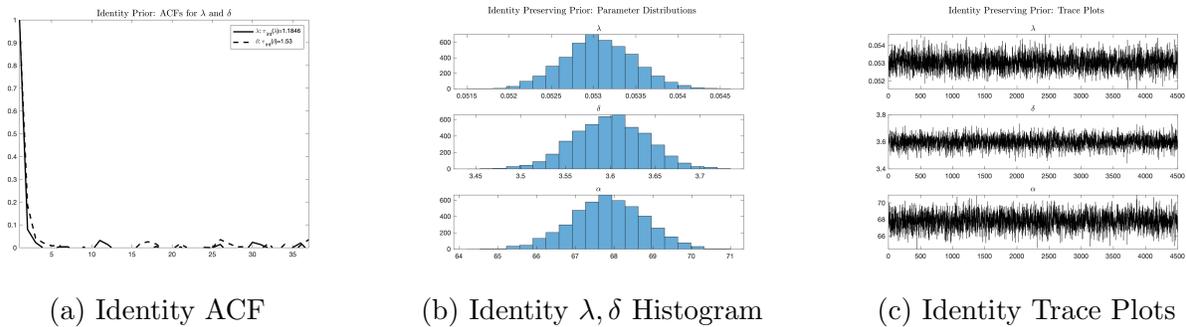


Figure 4.14: Markov Chain Diagnostics using the Identity Prior

Examining the three diagnostic plots in figure 4.14, a similar trend is found for the Identity Prior – the rapid drop off in the Autocorrelation Function graph indicates that the samples drawn from the identity chain are independent from the previous ones; the unimodal distribution of the histogram of sampled λ and δ values indicate that these parameters are relatively consistent. Finally, the Trace Plots also suggest that there is irreducibility as there are no discernible long term trends.

As a last example, an Edge-Preserving Prior when sampling from the posterior distribution. Note, that since the prior is determined iteratively using algorithm 1, it is updated to account for the change in parameter values. With this modification, the samples drawn allow for the exploration of the posterior distribution. Most notably, the HG sampling using an edge preserving prior generates the most accurate reconstruction of the true image – likely attributable to the imposition of correct prior information. Furthermore, it has the lowest overall variance compared to the other reconstructions.

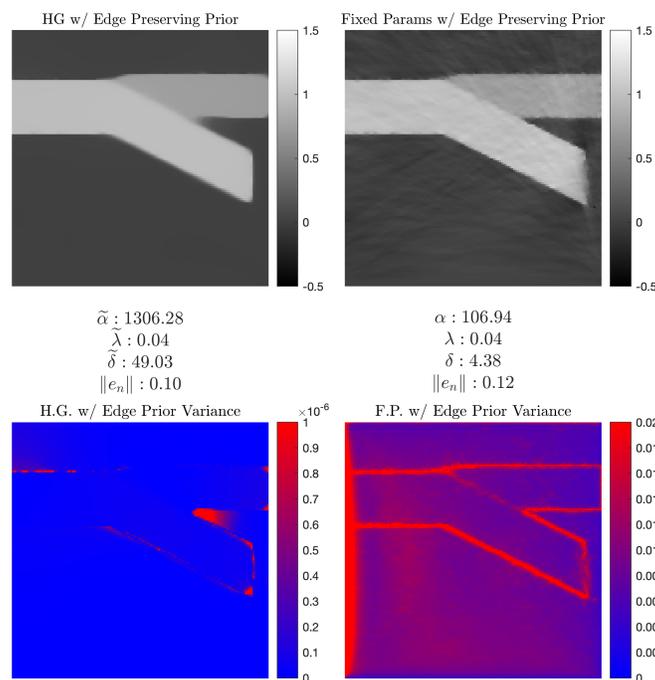


Figure 4.15: MCMC V.S. Fixed Parameter Methods with an Edge Preserving Prior

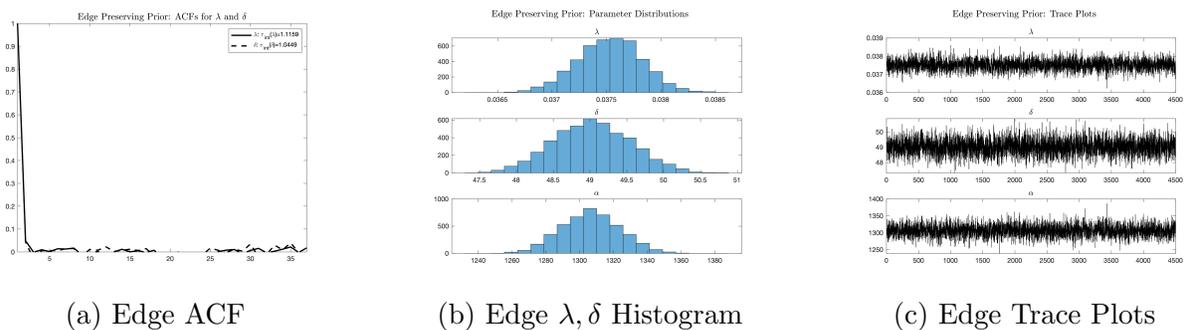


Figure 4.16: Markov Chain Diagnostics using the Edge-Preserving Prior

Examining the diagnostic plots again indicate that the MCMC chain converged correctly – indeed, the steep decrease in the ACF indicates large independence between samples, the shape of the histogram indicates confidence in the parameters being sampled, and the trace plot indicates aperiodicity.

4.3.1 Computational Complexity

Using the tic toc function in Matlab, the total time to compute and draw each sample using the different methods can be analyzed. Note, the discussion to follow leverages runtime as an analog to computational cost. Furthermore, runtime is compared in a relative sense as different levels of computing power can yield different results. The tables below includes the various runtimes for the different sampling methods.

Table 4.1: Fixed Parameters Run Times

Prior Type	Run Time
Identity	7 hours, 52 minutes
GMRF	8 hours, 32 minutes
Edge	8 hours, 44 minutes

Table 4.2: MCMC Run Times

Prior Type	Run Time
Identity	8 hours, 36 minutes
GMRF	8 hours, 54 minutes
Edge	8 hours, 53 minutes

Most notably, the run times across the board are similar. This makes sense as the most computationally heavy step comes from computing a sample using the CG method. The additional time to run using MCMC methods is most attributable to the two additional matrix-vector multiplications nested within each sample draw. While the overall cost for drawing one sample from the respective posterior distributions are similar, the main difference between the two methods lies in 1) the number of samples required to be drawn and 2) parallelization. For the former, to accurately recover estimates utilizing MCMC methods, the sample size required is around 5000-1000. Utilizing Fixed Parameters, the method has no hard iteration count – the overall variance does improve as more samples are drawn, but will converge to the true variance at some point. For the latter, the MCMC chain, as its name implies, requires a chain to be generated and therefore the computations cannot

be parallelized as they are independent after λ, δ have been sampled. However, after fixing λ and δ , such is the case for the Fixed Parameters method, samples drawn are independent and therefore can be parallelized. Thus, while both methods share a similar per-sample cost, the sequential nature of the MCMC chain and its higher sample requirements explain the differences in the noise level and overall computational strategy between the two approaches.

Chapter 5

Conclusions and Future Work

Guided by the broad issues outlined at the outset, this conclusion begins by reaffirming the driving aims of the thesis and clarifying their broader scholarly relevance. Recall that, in the context of large-scale inverse problems, two recurring difficulties stand out; first, ill-posedness disrupts the stability condition by amplifying small errors in measurements, leading to solutions swamped by noise unless carefully stabilized; second, computational intensity poses its own hurdle as even if a stable solution can be obtained through regularization, uncertainty quantification often demands high-dimensional sampling or iterative solvers, creating formidable demands on time and resources. These pressures are alleviated, in theory, by imposing suitable regularization to control noise amplification on the one hand, and by exploiting matrix structure and iterative methods on the other—yet the success of each depends heavily on context. Indeed, the choice of regularization approach and the nature of iterative methods can vary by problem size, noise characteristics, or domain constraints. Throughout this thesis, I have compared and contrasted such techniques primarily in the realm of seismic imaging, where partial sensor data and layered geological structures underscore both challenges. By examining how these algorithmic strategies interact with real or synthetic seismic data, the discussion that follows highlights why certain methods yield stable image reconstructions—and, crucially, how each technique manages or propagates uncertainty in

high-dimensional settings.

To illustrate these principles in practical settings, the thesis next turns to a series of carefully designed experiments that systematically assess how each regularization and sampling strategy performs under varying noise levels, problem sizes, and seismic-imaging constraints. In our first numerical experiment comparing different regularization parameter selection methods, we showed that randomized GCV based methods provided a strong balance between noise suppression and detail preservation, without needing knowledge of the noise level. This is in contrast to the the DP approach, where the assumptions on the GMRF prior and the method did not agree, resulting in over-regularization. Additionally, with additional computations of the randomized trace estimate, the GCV curve can be better approximated. Overall, Experiment 1 demonstrated that the choice of parameter-selection method can decisively impact the fidelity of reconstructed images. In the second numerical experiment, we demonstrated the importance of correct and representative prior selection in a myriad of context, underscoring how different priors can yield artifacts in reconstruction. Finally, Experiment 3 compared a hierarchical Gibbs sampler, which dynamically estimates hyper-parameters, against a fixed-parameter approach. The results showed that MCMC-based sampling gives a richer picture of posterior uncertainty—albeit at higher computational cost—while fixed-parameter sampling runs faster but can under-represent true variance if the chosen parameters are not accurate. Overall, it underscored the trade-off between in-depth uncertainty exploration and practical runtime constraints in large-scale inverse problems.

5.1 Future Work

The results presented here highlight both the promise and limitations of regularization-driven and Bayesian methodologies for large-scale inverse problems. Building on these insights, future investigations should explore enhancements to solver efficiency, the role of domain-

specific priors, and the adaptability of sensor setups in realistic seismic-imaging conditions.

5.1.1 Real-World Data and Varying Sensor Placement

A key next step is to extend these methods to real-world datasets. While synthetic experiments provide a controlled framework for evaluating inverse problem formulations, real-world seismic data introduces additional complexities such as correlated noise, measurement biases, and irregular sensor distributions. Indeed, physical limitations often impede the assumed uniform distribution of the sensors provided, introducing additional areas of uncertainty that might struggle to be explored by MCMC methods. Incorporating field-collected seismic waveforms would allow a direct assessment of the robustness of the proposed regularization techniques and Bayesian uncertainty quantification in practical settings. Additionally, the impact of sensor quality and placement should be further explored, particularly in scenarios where limited measurements induce under-determined systems. Investigating how the spatial configuration of sensors affects reconstruction quality will be critical for improving imaging in resource-constrained environments.

5.1.2 Preconditioning and Other Iterative Methods

A natural extension to the CG approach explored throughout this work is investigating preconditioners that can accelerate the solving process by reducing the number of iterations until convergence. Note that the construction of the preconditioner must be SPD as well, which is an additional restriction. Finding this preconditioner may be difficult, especially in contexts where the covariance matrix is dense. As such, exploring more advanced iterative methods such as randomized or inner-product free iterative approaches may be promising and have not been explored in the UQ context. Such methods for solving the systems that arise could be faster and determining an appropriate preconditioner may be easier. Additionally, although we have predominantly relied on forward operators that are SPD, certain problem variants — particularly those involving domain-specific scalings of — may deviate slightly

from strict symmetry in practice. In such cases, more advanced iterative methods can offer a more flexible pathway to accelerated convergence, especially when combined with carefully chosen preconditioners.

5.1.3 Boundary Conditions and Priors

Although the frameworks presented assume relatively straightforward boundary conditions (e.g., zero-boundaries), real-world applications demand more nuanced approaches to ensure accurate modeling. However, introducing complex boundary behavior may complicate theoretical guarantees — such as maintaining SPD properties — and thus merits additional scrutiny of how these boundary conditions interact with the forward operator and with the chosen priors. From a reconstruction perspective, boundary conditions exert notable influence on estimated solutions and their associated variances, potentially introducing artifacts if not carefully matched to the physical context. In parallel, prior formulations also shape how boundary effects propagate through an image. For instance, a GMRF prior with increased smoothing may inadvertently amplify or mask boundary artifacts. This raises broader questions about relaxing or modifying the default priors to accommodate regions of high uncertainty near domain edges. Investigating piecewise definitions of the GMRF — or other advanced priors — that scale their influence depending on proximity to boundaries could offer finer control of reconstruction quality. Ultimately, refining boundary condition models in tandem with domain-specific priors stands to bolster both the theoretical rigor and practical performance of large-scale seismic imaging methods.

Bibliography

- [1] Guillaume Bal. Introduction to inverse problems. <https://www.stat.uchicago.edu/~guillaumbal/PAPERS/IntroductionInverseProblems.pdf>, July 2019. University of Chicago, Chicago, IL 60637.
- [2] Johnathan M Bardsley. *Computational Uncertainty Quantification for Inverse Problems*, volume 19. SIAM, Philadelphia, 2018.
- [3] Daniela Calvetti and Erkki Somersalo. *Bayesian scientific computing*. Springer, 2023.
- [4] Julianne Chung and Silvia Gazzola. Computational methods for large-scale inverse problems: a survey on hybrid projection methods. *SIAM Review*, 66(2):205–284, 2024.
- [5] EarthScope Consortium. Seismic tomography—using earthquakes to image earth’s interior, n.d. Background document provided by IRIS animations. Supported by NSF Geodetic Facility for the Advancement of Geoscience (GAGE) and NSF Seismological Facility for the Advancement of Geoscience (SAGE). Available online at <http://www.iris.edu>.
- [6] Silvia Gazzola, Per Christian Hansen, and James G Nagy. IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81(3):773–811, 2019.
- [7] Jacques Hadamard. *Lectures on Cauchy’s Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, CT, 1923.

- [8] Per Christian Hansen. *Discrete inverse problems: Insight and algorithms*. SIAM, 2010.
- [9] Per Christian Hansen, James G. Nagy, and Dianne P. O’Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, Philadelphia, PA, 2006. ISBN 978-0-89871-618-4. doi: 10.1137/1.9780898718874. URL <https://doi.org/10.1137/1.9780898718874>.
- [10] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.
- [11] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Originally published by IEEE Press, 1988. Electronic version (C) 1999 by authors. Chapter 1.
- [12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi: 10.1126/science.220.4598.671. URL <https://www.science.org/doi/10.1126/science.220.4598.671>.
- [13] Johann Radon. On the determination of functions from their integral values along certain manifolds. *IEEE Transactions on Medical Imaging*, 5(4):170–176, December 1986. doi: 10.1109/TMI.1986.4307775. Translation of Radon’s 1917 article.
- [14] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Cambridge, MA, 12th edition, 2019. ISBN 9780128143469.
- [15] M. D. Sacchi. Introduction to inverse problems in exploration seismology. PIMS Summer School ’06 - Inversion and Imaging, Institute for Geophysical Research, University of Alberta, 2006. Available online: https://www.mathtube.org/sites/default/files/sacchi_pims2006.pdf.
- [16] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*. SIAM, 2024.

- [17] James C. Spall. *Stochastic Search and Optimization: Motivation and Supporting Results*, pages 1–33. John Wiley Sons, Ltd, 2003. ISBN 9780471722137. doi: <https://doi.org/10.1002/0471722138.ch1>.
- [18] Marco Taboga. Markov chain monte carlo (mcmc) diagnostics. Online appendix to "Lectures on Probability Theory and Mathematical Statistics", 2021. URL <https://www.statlect.com/fundamentals-of-statistics/Markov-Chain-Monte-Carlo-diagnostics>.
- [19] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
- [20] C Vogel. Computational methods for inverse problems. *Frontiers in Applied Mathematics/SIAM*, 2002.