ABSTRACT

PETROSKE, KATRINA ELISE. Efficient Methods for Image Reconstruction and Uncertainty Quantification with Application to Photo-acoustic Tomography. (Under the direction of Arvind Krishna Saibaba.)

Inverse problems arise in various scientific applications such as biomedical and geophysical imaging applications. A significant amount of effort has focused on developing efficient and robust methods to compute solutions to inverse problems by reconstructing parameters of interest. In addition to parameter reconstruction, there is a critical need to be able to obtain valuable uncertainty information (e.g., solution variances, samples, and credible intervals) to assess the reliability of computed solutions and to aid in decision-making. To demonstrate the techniques developed in this work we use a model problem, photo-acoustic tomography (PAT), an imaging modality that is used in breast and brain imaging. The goal of PAT is to recover the spatial distribution of the optical properties, such as the absorption coefficient, from ultrasound measurements. The PAT reconstruction process can be mathematically formulated as two coupled inverse problems involving partial differential equations. We take a two-step approach to solving PAT: the first step is a linear inverse problem for which we take a Bayesian approach and the second step is non-linear for which we take a deterministic approach.

In the first part of the thesis, we focus on uncertainty quantification for linear inverse problems with Gaussian posterior distributions. We exploit Krylov subspace methods to develop and analyze new techniques for large-scale uncertainty quantification in inverse problems by exploring the posterior distribution. In particular, we use the generalized Golub-Kahan bidiagonalization to derive an approximation of the posterior covariance matrix, and we provide theoretical results that quantify the accuracy of the approximate posterior covariance matrix and of the resulting posterior distribution. Then, we describe efficient methods that use the approximation to compute measures of uncertainty, such as the Kullback-Liebler divergence and optimality criteria, which are important in the context of optimal experimental design. Additionally, we present two methods that use the preconditioned Lanczos algorithm to efficiently generate samples from the posterior distribution. Numerical examples, including a model problem from PAT, demonstrate the effectiveness of the described approaches.

In the second part of the thesis, we focus on the non-linear inverse problem, also known

as Quantitative photo-acoustic tomography (QPAT). We take a deterministic approach and formulate QPAT a nonlinear PDE-constrained optimization problem. We develop Newton and Gauss-Newton solvers for QPAT in which the search directions are computed inexactly using the preconditioned Conjugate Gradient method. We study various aspects of the solvers such as the type of regularization used, choice of preconditioner, choice of stopping criteria, and the behavior as the number of sources is increased. The performance of the solvers is demonstrated through a synthetic model problem from QPAT.

Efficient Methods for Image Reconstruction and Uncertainty Quantification with Application to Photo-acoustic Tomography

by
Katrina Elise Petroske

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2020

APPROVED BY:

Alen Alexanderian	Ralph Smith
Mohammad Pour-Ghaz	Arvind Krishna Saibaba Chair of Advisory Committee

DEDICATION

To everyone that has influenced me along the way.

BIOGRAPHY

The author has forged a long (rather bumpy) path and, to the amazement of those that know them, reached this point in their life. Along the way they have done a lot of stuff and learned a lot of things.

ACKNOWLEDGEMENTS

This work was funded by NSF DMS 1720398, OP: Collaborative Research: Novel Feature-Based, Randomized Methods for Large-Scale Inversion.

I would also like to acknowledge my collaborator Dr. Julianne Chung from Virginia Polytechnic Institute and State University (Virginia Tech).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

Inverse problems are ubiquitous in sciences and engineering: examples include applications in astrophysics (recovering a true image from a blurry image), signal processing and acoustics (recovering the true signal from a noisy signal), medical imaging (magnetic resonance imaging, computed tomography, projection radiography), geophysics (groundwater contaminant source identification, seismic tomography, electrical resistance tomography, hydraulic tomography), nuclear engineering (detecting defects in spent nuclear fuel rods), and many more. As the name suggests, an inverse problem is the "inverse" of another problem called the forward problem. In a forward problem, parameters are inputs to a model, typically described using partial differential equations (PDEs), that is used to generate predictions (the output). In an inverse problem, the inputs are noisy data, which are used to recover the outputs which are parameters of interest. In Figure 1.1, we provide a diagram that demonstrates how the forward and inverse problem are related.

Finding the solution to an inverse problem presents many challenges. Inverse problems are often ill-posed, meaning that they either have no solution, no unique solution, or the solution does not depend continuously on the data. One approach to solving inverse

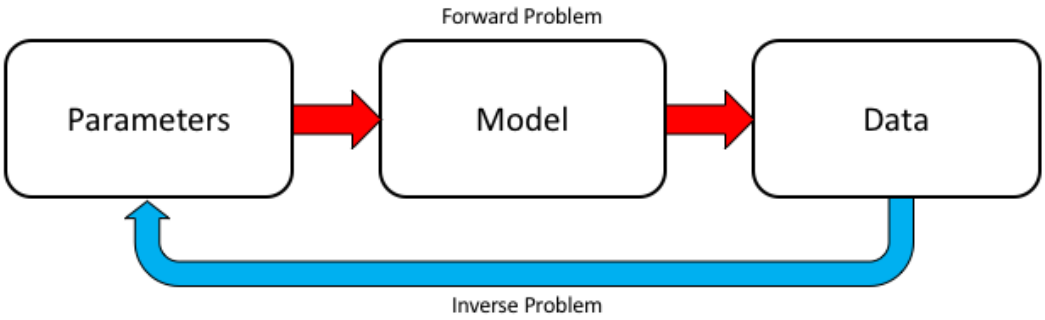


Figure 1.1 A diagram that shows the relationship between a forward problem and an inverse problem.

problems is deterministic, in which a single estimate of the parameter(s) is provided by solving an optimization problem. However, there may be uncertainties associate with the solution of the inverse problem and these can be handled using the Bayesian approach, where the solution of the inverse problem is the probability distribution of the parameter(s) of interest [69], since it has the capability of quantifying the uncertainty. Uncertainty quantification (UQ) for inverse problems is much more computationally intensive than generating a single estimate of the parameters. In this thesis we adopt both approaches for solving inverse problems.

In this thesis we develop and demonstrate efficient methods and algorithms to deal with some of the challenges posed by inverse problems. In order to motivate our work, we use the Photo-acoustic tomography (PAT) image reconstruction inverse problem as a model application. In this chapter we describe the model PAT problem and outline the thesis. In Section 1.1, we describe the physics behind PAT, some applications PAT is used in, and briefly discuss the image reconstruction process. In Section 1.2, we an outline of the contents and contributions of each chapter in the thesis.

1.1 Background for PAT

Photo-acoustic tomography (PAT) is a multi-scale imaging modality that works by illuminating a medium with multiple wavelengths of near infra-red light. The light is either absorbed or diffuses through the medium as it travels. The light that is absorbed transfers energy to the medium causing it to heat up. The heating causes thermal expansion, which in turn causes a pressure wave (the photo-acoustic effect). The pressure wave propagates as an acoustic wave and is measured by ultrasound detectors.

PAT is used in many *in vivo* imaging applications for anatomical, functional, molecular, and metabolic purposes in the microscopic scale [137]. PAT has been used to image mitochondrion (300 nm) [38] and blood vessels in the human hand (50-350 μm) [136]. PAT has, also been used in breast [119] and brain imaging [137]. Since PAT uses near infra-red light, unlike traditional imaging modalities (such as computed tomography, which use x-rays, and magnetic resonance imaging, that uses magnetic fields), it can be safely used on a wider variety of patients/subjects. For more technical engineering details about PAT systems and further applications the reader is directed to [130]. Outside of the life sciences PAT has been used as a non-destructive way to image the underdrawings of paintings [121].

For these reasons developing efficient methods for the reconstruction of PAT images is important and the inverse problems associated with PAT offer interesting mathematical challenges.

1.1.1 Mathematics of PAT

The physics behind PAT can be modeled by two separate, but coupled, partial differential equations (PDEs). Figure 1.2 shows the relation between the physics of the PAT process and the PDEs used to describe the physics.

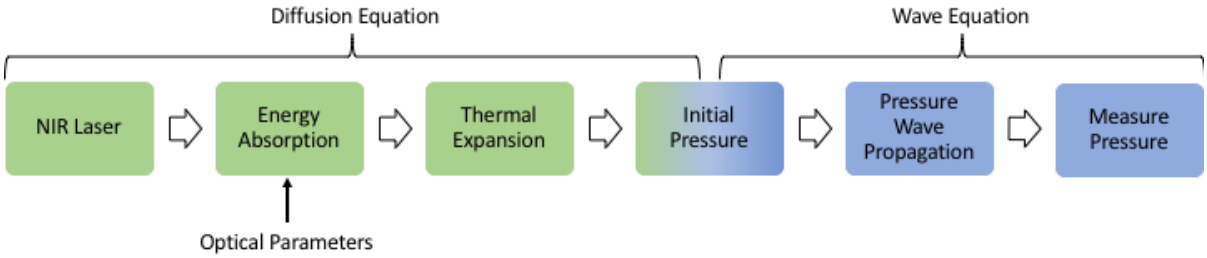


Figure 1.2 A diagram that shows the relation between the physics of the PAT process and the PDEs used to describe the physics.

PDEs Governing PAT

The first PDE that describes the physical process is a diffusion approximation to the radiation transport equation:

$$\begin{aligned} -\nabla \cdot (D\nabla u) + \mu u &= f, \quad x \in \Omega, \\ u + 2AD\nabla u \cdot \mathbf{n} &= 0, \quad x \in \partial\Omega, \end{aligned}$$

where μ is the absorption coefficient, u is the photon density, Ω is the domain, $\partial\Omega$ is the boundary of the domain, D is diffusion coefficient, and f is the source. This PDE describes the light illuminating the medium and the absorption of the light.

The second PDE used to model the underlying physics is a wave equation, which describes the propagation of the initial pressure wave:

$$\begin{aligned} \frac{1}{c^2} p_{tt} - \Delta p &= 0, \quad (t, x) \in [0, \infty) \times \Omega \\ p(0, x) &= H(x), \quad x \in \Omega, \\ p_t(0, x) &= 0, \quad x \in \Omega, \end{aligned}$$

where p is the pressure, c is the wave speed, and $H(x)$ is the initial pressure.

The two PDEs are coupled through the initial pressure, $H(x)$, in that

$$H(x) = \Upsilon(x)\psi(x) = \Upsilon(x)\mu(x)u(x), \tag{1.1}$$

where $\Upsilon(x)$ is the Grüneisen parameter, which is a measure of photoacoustic efficiency measure, and $\psi(x)$ is the absorbed energy. In Figure 1.3 we show a diagram of the wave equation and diffusion equation are coupled by the initial pressure.

Reconstruction of Photo-acoustic Images

In order to reconstruct the underlying optical properties in the medium we must use the collected data, i.e., the pressure measured by the ultrasound detectors, to first solve an inverse problem based on a wave equation to find the initial pressure wave, then we use the initial pressure wave to solve an inverse problem to find the optical properties, typically the absorption field. This is a two-step reconstruction process. An alternative approach called one-step PAT reconstructs the underlying optical properties directly from

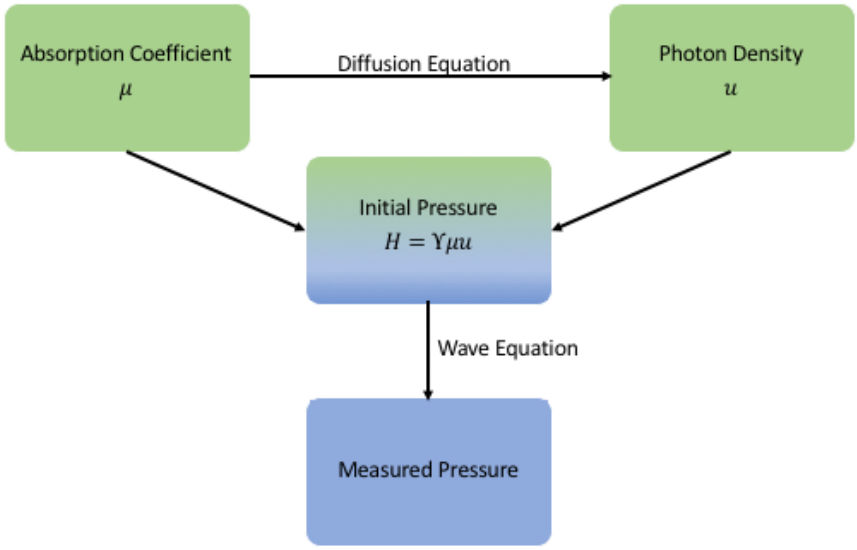


Figure 1.3 A diagram showing how the diffusion equation and wave equation are coupled through the initial pressure.

the measured pressure; we do not consider it here and refer the reader to [122, 67].

The first step in the PAT reconstruction process is known as the PAT step or the qualitative step. Many authors have explored methods for computing the solution, the initial pressure, for this step in the PAT inverse problem using explicit inversion, series solution, and time reversal [4, 5, 75, 77, 134, 132, 133]. If one assumes that the wave speed, c, is constant the data can be represented by the spherical Radon transform. Doing so allows this step of the reconstruction process to simplify to the inversion of the spherical Radon transform [30, 43, 42, 78, 76, 135]. In this work we use the spherical Radon transform approach to represent the forward problem.

The second step in the process, called Quantitative Photo-acoustic Tomography (QPAT), uses the initial pressure reconstructed from the first step as data. In this work we assume the Grüneisen parameter (1.1) takes the value 1, so the data is the absorbed energy. Authors have primarily taken Gauss-Newton and quasi-Newton approaches to solve this inverse problem [48, 35, 10, 47, 81, 46, 114].

1.2 Outline of Thesis

In this section, we provide an overview of the thesis and how each chapter is connected to the PAT inverse problem, as well as highlight the main contributions in each chapter.

Rather than dedicating a separate chapter to reviewing background material, we review the necessary background within each chapter. It should be noted that Chapters 2 and 3 expand on a paper currently undergoing revision at Numerical Linear Algebra with Applications journal; a pre-print is available [103].

Chapter 2: Efficient Computing of Uncertainty Quantification Measures in Large-Scale Bayesian Linear Inverse Problems

The first step of the PAT reconstruction process can be modeled as a linear inverse problem, where the data collected by the ultrasound detectors is used to recover the initial pressure. We focus on not only solving the inverse problem, but on quantifying the uncertainty associated with the reconstruction as well. In this chapter we consider large scale Bayesian inverse problems with linear forward operators and Gaussian priors, where the covariance matrices are large and dense, making them difficult to work with explicitly. For this particular setup a generalized hybrid Golub-Kahan (genHyBR) method [32] was developed. The genHyBR method is used to find the maximum a posteriori (MAP) estimate and an approximation of the posterior covariance matrix, which together determine the approximate posterior distribution.

Contribution: In this chapter we use the iterates of the genHyBR method to approximate the posterior covariance matrix. We derive and prove theoretical bounds for the accuracy of the approximate posterior covariance and bounds for the approximate posterior distribution. Additionally, we develop methods for approximating the optimality criteria, that play an important role in optimal experimental design, using the genHyBR iterates. We also develop error bounds to monitor the accuracy of our estimators and demonstrate the benefits of the approach on model problems including PAT.

Chapter 3: Sampling from Gaussian Posterior Distributions

A popular method for visualization and uncertainty quantification is to generate samples (conditional realizations) from the posterior distribution, which provides realizations of solutions and can be used for quantifying uncertainty. Krylov subspace methods can be used to efficiently sample from Gaussian posterior distributions, which is the case for PAT. In particular, a Lanczos process can be used to generate samples from a posterior distribution. However, if the covariance matrix is large and dense, traditional Krylov subspace methods can be computationally infeasible.

Contribution: In this chapter, we propose two different sampling methods that use preconditioned Lanczos methods to sample realizations from the Gaussian posterior distribution arising from the PAT problem. The first method generates samples from approximate posterior distribution; we use the genHyBR method to approximate the posterior. The second method generates approximate samples from the exact posterior distribution. Sampling from the approximate posterior distribution has a lower computational cost while sampling from the exact posterior distribution provides better accuracy. We demonstrate both methods on model problems, including PAT.

Chapter 4: Efficient Newton-based Approaches to Solve Deterministic Quantitative Photo-acoustic Tomography

In this chapter, we model QPAT as a deterministic, PDE constrained optimization problem. We assume the initial pressure has been reconstructed and use it as the data in order to reconstruct the absorption coefficient. Current methods for the nonlinear QPAT image reconstruction process use only gradient information or approximations of the Hessian [35, 10, 47, 81, 46]. However, in this chapter we investigate the use of Newton-based methods that have the potential to reduce the computational cost over the existing methods.

Contribution: We solve the regularized QPAT inverse problem by taking an optimize-then-discretize approach. We use Lagrangian-based adjoint methods to derive the optimality conditions and then we solve the discretized inverse problem with preconditioned inexact-Newton-CG and Gauss-Newton methods. Additionally, we investigate different regularization types and preconditioners for efficiently computing the Newton step. We show that while the use of the full Hessian has a higher cost per iteration compared to the Gauss-Newton Hessian, it has an overall lower computational cost.

Chapter 5: Conclusion

Finally, in Chapter 5, we present overall conclusions and contributions, and highlight a few avenue for future research.

CHAPTER

2

EFFICIENT COMPUTATION OF UNCERTAINTY QUANTIFICATION MEASURES IN LARGE-SCALE BAYESIAN LINEAR INVERSE PROBLEMS

Inverse problems arise in various scientific applications, and a significant amount of effort has focused on developing efficient and robust methods to compute approximate solutions. However, as these numerical solutions are increasingly being used for data analysis and to aid in decision-making, there is a critical need to be able to obtain valuable uncertainty information (e.g., solution variances, samples, and credible intervals) to assess the reliability of computed solutions. Tools for inverse uncertainty quantification (UQ) often build upon the Bayesian framework from statistical inverse problems. Great

overviews and introductions can be found in, e.g., [19, 115, 117, 70, 25].

Unfortunately, for very large inverse problems, UQ using the Bayesian approach is prohibitively expensive from a computational standpoint. This is partly because the posterior covariance matrices are so large that constructing, storing, and working with them directly is not computationally feasible. For these scenarios, a generalized hybrid Golub-Kahan based method was proposed in [33] to efficiently compute the maximum a posteriori (MAP) estimate and to select a precision parameter automatically. We go beyond computing reconstructions (e.g., MAP estimates) and develop efficient methods for inverse UQ. In this chapter we focus on methods that use the approximate posterior distribution to compute measures of uncertainty. In Chapter 3 we develop methods for generating samples from the approximate and true posterior distributions that can then be used to compute measures of uncertainty.

Overview of Chapter and Main Contributions

In this chapter, we use efficient Krylov subspace iterative methods, previously used for solving the weighted least squares problem, for inverse UQ. This motivates new algorithms and analyses, which is the central focus of this chapter. Section 2.1 describes the setup of the linear Bayesian inverse problems and assumptions we are making. Sections 2.2 and 2.3 describe the generalized Hybrid Golub-Kahan (genHyBR) based method, which is used to solve the MAP estimate and estimate the precision parameter. The iterates from the genHyBR process are used to efficiently construct an approximation to the posterior covariance matrix resulting in an approximate posterior distribution. In Section 2.4, we discuss how the approximate posterior distribution can be used in Optimal Experimental Design. Lastly, in Section 2.5 we present numerical examples that demonstrate the performance of the theoretical bounds found in Section 2.3 and Section 2.4.
The main contributions are as follows:

- We develop error bounds for monitoring the accuracy of the approximate posterior covariance matrix based on the generalized hybrid Golub-Kahan [33] iterates.
- We relate the error in the approximate posterior covariance matrix to the error in the approximate posterior distribution.
- We show how to efficiently compute measures of uncertainty, such as the Kullback-Leibler divergence, from the posterior distribution to the prior distribution.

- We develop error bounds for the accuracy of approximated optimality criteria used in optimal experimental design.

2.1 Setup of the Linear Bayesian Inverse Problem

Throughout this chapter, we consider a linear inverse problem of the form

$$\mathbf{d} = \mathbf{A}\mathbf{s} + \boldsymbol{\delta}, \tag{2.1}$$

where the terms are defined as

- $\mathbf{d} \in \mathbb{R}^m$ are the observed data,
- $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a given matrix, forward operator, or parameter-to-observable map.

We model the the unknown parameters, $\mathbf{s}$, and the measurement error, $\boldsymbol{\delta}$, as independent Gaussian random vectors; that is

- $\boldsymbol{\delta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\text{noise}})$, where $\boldsymbol{\delta} \in \mathbb{R}^m$ is the measurement error (noise), $\boldsymbol{\Gamma}_{\text{noise}} \in \mathbb{R}^{m \times m}$ is a covariance matrix, e.g. a symmetric positive definite matrix,
- $\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}, \lambda^{-2}\mathbf{Q})$, where $\mathbf{s} \in \mathbb{R}^n$ are the unknown parameters we want to reconstruct, $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean or expected value of the parameters, λ is a non-zero precision parameter, and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a covariance matrix.

Note that the precision parameter λ can be fixed or an unknown to be estimated as part of the inversion, discussed and is discussed in Section 2.2.2.

By taking a Bayesian approach to the problem we are no longer looking for a single deterministic solution; instead we are trying to find the posterior probability distribution, or the probability of $\mathbf{s}$ conditioned on the observed data $\mathbf{d}$. Recall Bayes' theorem, which states that the posterior probability distribution function (p.d.f.) is given by

$$\pi_{\text{post}} \equiv \pi(\mathbf{s}|\mathbf{d}) = \frac{\pi(\mathbf{d}|\mathbf{s})\pi(\mathbf{s})}{\pi(\mathbf{d})}\,. \tag{2.2}$$

It is also common to write $\pi_{\text{post}} \propto \pi(\mathbf{d}|\mathbf{s})\pi(\mathbf{s})$, where $\propto$ means proportional to, since the constant of proportionality is often not important in the application, as in our case.

Along with the posterior p.d.f. there are other values of interest, such as the maximum a posteriori (MAP) estimate that is defined as

$$\mathbf{s}_{\text{MAP}} = \arg\max_{\mathbf{s}\in\mathbf{R}^n} \pi(\mathbf{s}|\mathbf{d}), \tag{2.3}$$

which maximizes the posterior p.d.f. ~~probability~~ and the conditional mean (CM)

$$\mathbf{s}_{\text{CM}} = \int_{\mathbf{R}^n} \mathbf{s}\pi(\mathbf{s}|\mathbf{d})\, d\mathbf{s}, \tag{2.4}$$

which is also called the posterior mean [25]. The MAP estimate and CM help characterize the posterior p.d.f. and are particularly useful if the posterior is hard to visualize due to the high dimensionality of the problem or if the computational cost of solving the problem is prohibitive [69].

Since $\mathbf{s}$ and $\boldsymbol{\delta}$ are Gaussian random vectors, the posterior p.d.f. has the following representation,

$$\pi_{\text{post}} \propto \exp\left(-\frac{1}{2}\|\mathbf{A}\mathbf{s}-\mathbf{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{\lambda^2}{2}\|\mathbf{s}-\boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}\right), \tag{2.5}$$

where $\|\mathbf{x}\|_{\mathbf{M}} = \sqrt{\mathbf{x}^\top\mathbf{M}\mathbf{x}}$ is a vector norm for any symmetric positive definite matrix $\mathbf{M}$. Thus, the posterior distribution, π_{post}, is Gaussian with mean, $\mathbf{s}_{\text{post}}$, and covariance, $\boldsymbol{\Gamma}_{\text{post}}$, defined as

$$\boldsymbol{\Gamma}_{\text{post}} \equiv (\lambda^2\mathbf{Q}^{-1} + \mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{A})^{-1} \quad \text{and} \quad \mathbf{s}_{\text{post}} = \boldsymbol{\Gamma}_{\text{post}}(\mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{d} + \lambda^2\mathbf{Q}^{-1}\boldsymbol{\mu}) \tag{2.6}$$

and furthermore [69, Chapter 3.4] for linear inverse problems

$$\mathbf{s}_{\text{post}} = \mathbf{s}_{\text{MAP}} = \mathbf{s}_{\text{CM}}.$$

For the reminder of this chapter we will refer to $\mathbf{s}_{\text{post}}$ the MAP estimate.

For the problems of interest, computing the inverse and square root of $\boldsymbol{\Gamma}_{\text{noise}}$ are inexpensive (~~e.g.,~~ $\boldsymbol{\Gamma}_{\text{noise}}$ is a diagonal matrix), but explicit computation of $\mathbf{Q}$ (or its inverse or square root) may not be possible, this in turn makes computation and manipulation of $\boldsymbol{\Gamma}_{\text{post}}$ difficult. However, we assume that matrix-vector products (mat-vecs) involving $\mathbf{Q}$ can be done efficiently (e.g., in $\mathcal{O}(n\log n)$ operations rather than $\mathcal{O}(n^2)$ operations for an

$n \times n$ matrix). The covariance matrix $\mathbf{Q}$ can be described using covariance kernels we review a specific class of covariance kernels, that are popular for their desirable properties, which we use in this work.

2.1.1 Matérn Covariance Kernels

The Matérn class [96] is a class of stationary, isotropic, and positive definite kernels, given by the formula

$$C_\nu(r) = \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right) \tag{2.7}$$

where

- r is a measure of the distance between points,
- ν is a parameter that controls smoothness of the underlying process,
- Γ is the Gamma function,
- ℓ is a scaling parameter or the correlation length,
- $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν.

A feature of Matérn family of kernels is that it includes other widely used kernels for certain values of ν.

- When $\nu \to \infty$ we get the squared exponential covariance function, for appropriate scaling of ℓ

$$\lim_{\nu \to \infty} C_\nu(r) = \exp\left(\frac{-r^2}{2\ell^2} \right)$$

- When $\nu = 1/2$ we get the exponential covariance function

$$C_{\nu=1/2}(r) = \exp\left(\frac{-r}{\ell} \right)$$

- When $\nu = p + 1/2$, where p is a positive integer we get that the covariance function is a product of an exponential and order p polynomial [97, 53],

$$C_{\nu=p+1/2}(r) = \exp\left(\frac{-\sqrt{2\nu} r}{\ell} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu} r}{\ell} \right)^{p-i}$$

If we have the set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, are points in the domain of interest that has dimension d, we can then define a covariance matrix

$$Q_{i,j} = C_\nu(\|\mathbf{x}_i - \mathbf{x}_j\|_2), \quad \text{for} \quad i, j = 1 \ldots n.$$

In Figure 2.1, we show some examples of Matérn kernels and realizations drawn from the distributions generated by the corresponding kernels. The figure illustrates how the value of ν controls the smoothness of the realizations and, in fact, the realizations are almost surely, $\lceil \nu \rceil - 1$ order continuously differentiable [105].

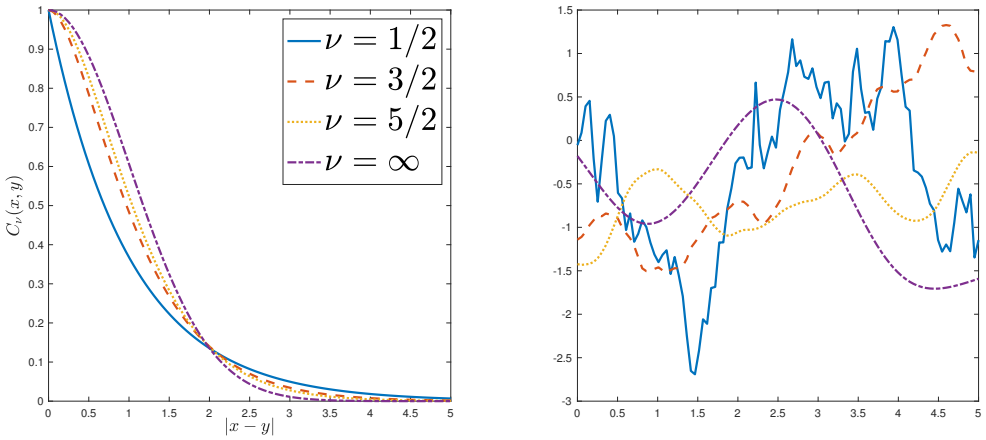


Figure 2.1 Examples of Matérn kernels for $\ell = 1$ and different values of ν (left) and realizations of the Gaussian distributions with zero mean and covariance defined be the respective kernels (right).

To summarize, the Matérn class of covariance kernels are useful because: ~~they have good theoretical properties,~~ they include other popular covariance kernels, the smoothness of the realizations can be controlled, they allow for areas of relative smoothness, but also can capture jumps. A drawback is that the covariance matrix, $\mathbf{Q}$, defined by Matérn kernels are dense and for large grid sizes making storage, inversion, and factoring difficult, but there are efficient ways to compute matrix vector products (mat-vecs) with $\mathbf{Q}$, which we now describe.

In general, the storage and computational cost for mat-vecs involving a dense $\mathbf{Q}$ is

$\mathcal{O}(n^2)$, recall that the forward operator $\mathbf{A} \in \mathbb{R}^{m \times n}$, using the naive approach of storing the entries explicitly. However, for covariance matrices defined by stationary or translation invariant covariance kernels (this includes Matérn kernels) with spatial points located on a regular, equispaced grid the cost per mat-vec can be reduced to $\mathcal{O}(n \log n)$, recall n is the dimension of the parameters. This is done be exploiting the connection between the Fast Fourier Transform (FFT) and matrices with Toeplitz structure [84]. For irregular grids the cost for approximate mat-vecs can be reduced to $\mathcal{O}(n \log n)$ using Hierarchical matrices [101] or $\mathcal{O}(n)$ using $\mathcal{H}^2$-matrices or the Fast Multipole Method (FMM) described in [3]. We will only use FFT based methods in this work.

In the subsequent sections, we describe algorithms for finding the maximum a posteriori (MAP) estimate and the approximate posterior covariance matrix, using the Generalized Hybrid Golub-Kahan method that can be used with large, dense $\mathbf{Q}$. In the next chapter we will discuss how to use the MAP estimate and posterior covariance to draw samples from the posterior distribution.

2.2 Generalized Hybrid Golub-Kahan (genHyBR) Approach

In this section we will discuss how the genHyBR method [33] can be used to find the MAP estimate for the posterior distribution. The genHyBR method has two components: a generalized Golub-Kahan (genGK) method for computing Krylov subspaces and methods for estimating the precision parameter λ. We describe the genHyBR method in detail in this section to provide context for subsequent material. We start by a brief review of existing methods, then describe the underlying algorithm for the method, how the MAP estimate is found, and finally the hybrid approach we take.

2.2.1 Review of Existing Methods

There is a large body of work that focuses on finding the MAP estimate, $\mathbf{s}_{\text{post}}$, which in our case can be found by minimizing the negative log likelihood of (2.5), e.g.

$$\mathbf{s}_{\text{post}} = \underset{\mathbf{s} \in \mathbb{R}^n}{\arg\min} \frac{1}{2} \|\mathbf{A}\mathbf{s} - \mathbf{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{\lambda^2}{2} \|\mathbf{s} - \boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}} \tag{2.8}$$

which is equivalent to solving the normal equations

$$\left(\mathbf{A}^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{A} + \lambda^2 \mathbf{Q}^{-1}\right)\mathbf{s} = \mathbf{A}^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{d} + \lambda^2 \mathbf{Q}^{-1}\boldsymbol{\mu}. \tag{2.9}$$

Many of the existing methods solve a problem equivalent to (2.8), called a general-form Tikhonov problem, defined as

$$\mathbf{s}_{\text{post}} = \arg\min_{\mathbf{s}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{L}_{\boldsymbol{\Gamma}}(\mathbf{A}\mathbf{s} - \mathbf{d})\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}_{\mathbf{Q}}(\mathbf{s} - \boldsymbol{\mu})\|_2^2, \tag{2.10}$$

where $\mathbf{L}_{\boldsymbol{\Gamma}}$ and $\mathbf{L}_{\mathbf{Q}}$ are Cholesky factors of $\boldsymbol{\Gamma}_{\text{noise}}$ and $\mathbf{Q}$ respectively.

In iterative regularization, the precision parameter λ is set to zero and an iterative solver is used to minimize the data misfit term, the first term in (2.10), and terminate the iterations early [59]. This approach relies on the semiconvergence of iterative methods where the initial iterates appear to reduce the error, but at some later iteration the iterates will diverge from the true solution [59, 31]. A major drawback is that selecting a good stopping iteration is often a non trivial task [31].

One way of finding the precision parameter λ is to solve (2.10) multiple times with different a priori selected λ then, depending on chosen criteria, the solution corresponds to the "best" λ [59]. In Section 2.2.2 we will go into further detail about different ways of choosing λ. One can clearly see that this approach to finding the precision parameter is computationally intensive, making it infeasible for some problems.

Another way to solve (2.10) is to project the problem into a lower-dimensional subspace, search for a solution in this subspace. The basis vectors for the low dimension subspace are typically taken to be the basis vectors for a Krylov subspace associated with (2.9) [59]. Projection methods also display semiconvergence so the previously mentioned early termination methods apply to the projected problem as well.

This finally brings us to hybrid methods. A hybrid method combines projection methods with estimating λ during the iterative process. There have been hybrid iterative methods [85, 73, 14, 29, 31, 62, 52] developed to solve the standard-form Tikhonov problem, $\mathbf{L}_{\mathbf{Q}} = \mathbf{I}$, and the general problem [98, 72, 51, 63, 64], but they would involve factoring $\mathbf{Q}$. The genHyBR algorithm we use is also a hybrid method, which avoids this factorization.

Preliminaries

Since we cannot compute $\mathbf{Q}^{-1}$, we first make a change of variables

$$\mathbf{x} = \mathbf{Q}^{-1}(\mathbf{s} - \boldsymbol{\mu}), \qquad \mathbf{b} = \mathbf{d} - \mathbf{A}\boldsymbol{\mu},$$

so that (2.8) is equivalent to solving

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{AQx} - \mathbf{b}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{\lambda^2}{2}\|\mathbf{x}\|^2_{\mathbf{Q}} \tag{2.11}$$

or

$$(\mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{AQ} + \lambda^2\mathbf{I})\mathbf{x} = \mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{b}.$$

The genHyBR method we use to solve (2.11) is built upon the generalized Golub-Kahan (genGK) bidiagonalization method. The basic idea behind the genHyBR method is first, using the genGK method, generate a basis $\mathbf{V}_k$ for the Krylov subspace

$$\mathcal{S}_k \equiv \text{Span}\{\mathbf{V}_k\} = \mathcal{K}_k(\mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{AQ}, \mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{b}) \tag{2.12}$$

$$= \mathcal{K}_k(\mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{AQ} + \lambda^2\mathbf{I}, \mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{b}) \tag{2.13}$$

where $\mathcal{K}_k(\mathbf{M}, \mathbf{g}) = \text{Span}\{\mathbf{g}, \mathbf{Mg}, \ldots, \mathbf{M}^{k-1}\mathbf{g}\}$, and second to solve (2.11) in this subspace. Note that we have used the property that Krylov subspaces are shift invariant i.e. $\mathcal{K}_k(\mathbf{M}, \mathbf{g}) = \mathcal{K}_k(\mathbf{M} + \lambda^2\mathbf{I}, \mathbf{g})$.

Generalized Golub-Kahan (genGK) bidiagonalization Method

A basis for $\mathcal{K}_k(\mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{AQ} + \lambda^2\mathbf{I}, \mathbf{A}^\top\boldsymbol{\Gamma}^{-1}_{\text{noise}}\mathbf{b})$ can be generated using the genGK bidiagonalization process[1] summarized in Algorithm 2.2.1, where at the end of k steps, we have

[1]Generalized Golub-Kahan methods were first proposed by Benbow [15] for generalized least squares problems, and used in several applications, see e.g. [7, 6, 86]. However, the specific form of the bidiagonalization was developed in [33].

the matrices

$$\mathbf{U}_{k+1} \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_{k+1}], \mathbf{V}_k \equiv [\mathbf{v}_1, \ldots, \mathbf{v}_k], \quad \text{and} \quad \mathbf{B}_k \equiv \begin{bmatrix} \alpha_1 & & \\ \beta_2 & \ddots & \\ & \ddots & \alpha_k \\ & & \beta_{k+1} \end{bmatrix} \tag{2.14}$$

that, in exact arithmetic, satisfy

$$\mathbf{A}\mathbf{Q}\mathbf{V}_k = \mathbf{U}_{k+1}\mathbf{B}_k \tag{2.15}$$

$$\mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{U}_{k+1} = \mathbf{V}_k\mathbf{B}_k^\top + \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^\top \tag{2.16}$$

and

$$\mathbf{U}_{k+1}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{U}_{k+1} = \mathbf{I}_{k+1}, \quad \mathbf{V}_k^\top \mathbf{Q}\mathbf{V}_k = \mathbf{I}_k. \tag{2.17}$$

Vector $\mathbf{e}_{k+1}$ corresponds to the $(k+1)$st standard unit vector or the last column of $\mathbf{I}_{k+1}$.

Algorithm 2.2.1 gen-GK bidiagonalization

Output: $[\mathbf{U}_k, \mathbf{V}_k, \mathbf{B}_k] = \texttt{gen-GK}(\mathbf{A}, \mathbf{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{b}, k)$

$\beta_1 = \|\mathbf{b}\|_{\mathbf{\Gamma}_{\text{noise}}^{-1}}$
$\beta_1\mathbf{u}_1 = \mathbf{b}$
$\alpha_1 = \|\mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{u}_1\|_{\mathbf{Q}}$
$\alpha_1\mathbf{v}_1 = \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{u}_1$
for $i = 1, \ldots, k$ **do**
  $\beta_{i+1} = \|\mathbf{A}\mathbf{Q}\mathbf{v}_i - \alpha_i\mathbf{u}_i\|_{\mathbf{\Gamma}_{\text{noise}}^{-1}}$
  $\beta_{i+1}\mathbf{u}_{i+1} = \mathbf{A}\mathbf{Q}\mathbf{v}_i - \alpha_i\mathbf{u}_i$
  $\alpha_{i+1} = \|\mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{u}_{i+1} - \beta_{i+1}\mathbf{v}_i\|_{\mathbf{Q}}$
  $\alpha_{i+1}\mathbf{v}_{i+1} = \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{u}_{i+1} - \beta_{i+1}\mathbf{v}_i$
end for

Finding the MAP Estimate with genGK Method

After finding a basis for $\mathcal{K}_k$, we seek an approximate solution to (2.11) of the form $\mathbf{x}_k = \mathbf{V}_k\mathbf{z}_k$, so that $\mathbf{x}_k \in \mathcal{K}_k$. We can then determine $\mathbf{z}_k$ by solving either the gen-LSQR

sub-problem

$$\min_{\mathbf{x}_k \in \mathcal{K}_k} \frac{1}{2}\|\mathbf{AQx}_k - \mathbf{b}\|^2_{\mathbf{\Gamma}^{-1}_{\text{noise}}} + \frac{\lambda^2}{2}\|\mathbf{x}_k\|^2_{\mathbf{Q}} \quad \Leftrightarrow \quad \min_{\mathbf{z}_k \in \mathbb{R}^k} \frac{1}{2}\|\mathbf{B}_k\mathbf{z}_k - \beta_1\mathbf{e}_1\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{z}_k\|_2^2, \quad (2.18)$$

or the gen-LSMR sub-problem

$$\min_{\mathbf{x}_k \in \mathcal{K}_k} \frac{1}{2}\|\mathbf{A}^\top\mathbf{\Gamma}^{-1}_{\text{noise}}(\mathbf{AQx}_k - \mathbf{b})\|^2_{\mathbf{Q}} + \frac{\lambda^2}{2}\|\mathbf{x}_k\|^2_{\mathbf{Q}} \quad \Leftrightarrow \quad \min_{\mathbf{z}_k \in \mathbb{R}^k} \frac{1}{2}\|\bar{\mathbf{B}}_k\mathbf{z}_k - \bar{\beta}_1\mathbf{e}_1\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{z}_k\|_2^2, \quad (2.19)$$

where,

$$\bar{\mathbf{B}}_k = \begin{bmatrix} \mathbf{B}_k^\top\mathbf{B}_k \\ \bar{\beta}_{k+1}\mathbf{e}_1^\top \end{bmatrix} \qquad \text{and} \qquad \bar{\beta}_k = \beta_k\alpha_k. \quad (2.20)$$

Here gen-LSQR and genLSMR generalize the standard LSQR and LSMR problems; [88, 45].

For a fixed precision parameter, λ, an approximate MAP estimate can be recovered by undoing the change of variables,

$$\mathbf{s}_k = \boldsymbol{\mu} + \mathbf{Qx}_k = \boldsymbol{\mu} + \mathbf{QV}_k\mathbf{z}_k, \quad (2.21)$$

where, now, $\mathbf{s}_k \in \boldsymbol{\mu} + \mathbf{Q}\mathcal{K}_k$. In the next section we will discuss how the precision parameter is computed for each iteration.

2.2.2 The genHyBR Method [33]

As stated previously in hybrid methods the precision parameter λ is computed on-the-fly during the iterative process. We will focus on three methods for finding λ: Generalized Cross Validation (GCV), Discrepancy Principle (DP), and Unbiased Predictive Risk Estimator (UPRE). For each method, we also provide expressions for the projected problem (using the genGK estimates) and for comparison we show the how the methods are used for the general-form Tikhonov problem (2.10). Note: For ease of notation, in this section we assume that $\boldsymbol{\mu} = \mathbf{0}$.

First, we look at the Generalized Cross Validation (GCV) [55]. For (2.10) we can find

the parameter λ_{gcv} that minimizes the GCV function

$$G(\lambda) = \frac{n\|\mathbf{A}\mathbf{s}_\lambda - \mathbf{d}\|^2_{\mathbf{\Gamma}^{-1}_{\text{noise}}}}{\left[\mathsf{trace}\left(\mathbf{I}_m - \mathbf{L}_{\mathbf{\Gamma}}\mathbf{A}\mathbf{A}^\dagger_\lambda\right)\right]^2} \tag{2.22}$$

where $\mathbf{L}^\top_{\mathbf{\Gamma}}\mathbf{L}_{\mathbf{\Gamma}} = \mathbf{\Gamma}^{-1}_{\text{noise}}$ and

$$\mathbf{A}^\dagger_\lambda = (\mathbf{A}^\top\mathbf{\Gamma}^{-1}_{\text{noise}}\mathbf{A} + \lambda^2\mathbf{Q}^{-1})^{-1}\mathbf{A}^\top\mathbf{L}^\top_{\mathbf{\Gamma}}.$$

We can instead find use the GCV function that corresponds to the projected problem (2.18), where λ_{gcv} at the kth iterate minimizes

$$G_{\text{proj}}(\lambda) = \frac{k\|(\mathbf{I} - \mathbf{B}_k\mathbf{B}^\dagger_{k,\lambda})\beta_1\mathbf{e}_1\|^2_2}{\left[\mathsf{trace}\left(\mathbf{I}_{k+1} - \mathbf{B}_k\mathbf{B}^\dagger_{k,\lambda}\right)\right]^2}, \tag{2.23}$$

where

$$\mathbf{B}^\dagger_{k,\lambda} = (\mathbf{B}^\top_k\mathbf{B}_k + \lambda^2\mathbf{I})^{-1}\mathbf{B}^\top_k.$$

A similar GCV function can be found for the projected problem in (2.19) by replacing $\mathbf{B}_k$ with $\bar{\mathbf{B}}_k$ and β_1 with $\bar{\beta}_k$. We will also note that a weighted-GCV (WGCV) approach [29, 99] has been suggested where a weighting parameter is included in the denominator of (2.23). We will let λ_{wgcv} be the minimizer of the WGCV function. The GCV method is robust, if the noise in the problem is white, but the GCV function can have a flat minimum causing an underestimate of λ_{gcv} which can lead to "ridiculous under-smoothing" of the solution [73, 59].

Another method is the Discrepancy Principle (DP), where λ_{dp} is found so that the residual norm is on the order of the noise in the data, i.e.,

$$\|\mathbf{A}\mathbf{s}_\lambda - \mathbf{d}\|^2_{\mathbf{\Gamma}^{-1}_{\text{noise}}} = \tau\delta, \tag{2.24}$$

where δ is an estimate of the squared norm of the noise and $\tau \gtrapprox 1$. Again, we can find λ_{dp} by solving at the projected problem, i.e.,

$$\|\mathbf{B}_k\mathbf{z}_{k,\lambda} - \beta_1\mathbf{e}_1\|^2_2 = \tau\delta. \tag{2.25}$$

A drawback of the DP method is that the norm of the noise may be unknown and must be estimated. Furthermore, λ_{dp} is very sensitive to δ and even with a good estimate, the solutions can be over-smoothed [73].

Lastly, we consider the Unbiased Predictive Risk Estimator (UPRE) method, where λ_{upre} is taken to minimize

$$U(\lambda) = \frac{1}{n}\|\mathbf{A}\mathbf{s}_\lambda - \mathbf{d}\|^2_{\mathbf{\Gamma}^{-1}_{\text{noise}}} + \frac{2\eta^2}{n}\mathsf{trace}\left(\mathbf{L}_{\mathbf{\Gamma}}\mathbf{A}\mathbf{A}^\dagger_\lambda\right) - \eta^2, \tag{2.26}$$

where $\eta^2 = \delta/m$, (recall m is the dimension of the data). For the projected problem we compute λ_{upre} by minimizing

$$U_{\text{proj}}(\lambda) = \frac{1}{k}\|\mathbf{B}_k\mathbf{z}_{k,\lambda} - \beta_1\mathbf{e}_1\|_2^2 + \frac{2\eta^2}{k}\mathsf{trace}\left(\mathbf{B}_k\mathbf{B}^\dagger_{k,\lambda}\right) - \eta^2. \tag{2.27}$$

In [99], it was shown that the λ_{upre} found for the projected problem is a good estimate for the full problem. As noted previously, the norm of the noise may not be known exactly.

Now that we described all of the main components of the genHyBR method in Algorithm 2.2.2 we give a sketch of the complete method. We remind the reader the benefit of using this hybrid approach is that this algorithm automatically determines the number of iterations k and the precision parameter λ. In particular, the stopping iteration is determined using a combination of approaches, including a maximum number of iterations, a GCV function defined in terms of the iteration, and tolerances on the residual. We remark that early termination of the genHyBR iterations can negatively affect the reconstruction and later approximations, but a later termination will not have a significant impact due to the inclusion of proper regularization. Thus, for subsequent UQ, a safer option is to perform a few extra iterations of the genHyBR method.

In the next section we will discuss how the genHyBR method can be used to find an approximation of the posterior covariance (2.6).

2.3 Approximation and Accuracy of Posterior Covariance Using the genHyBR Method

In this section, we will first briefly review existing methods to find the posterior covariance, then describe how the genHyBR method can be used to approximate the posterior

Algorithm 2.2.2 genHyBR

Output: $[\mathbf{s}_k, \mathbf{U}_k, \mathbf{V}_k, \mathbf{B}_k, \mathbf{QV}_k\ \lambda] = \texttt{genHyBR}(\mathbf{A}, \mathbf{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{b}, k_{max})$
  $[\mathbf{U}_1, \mathbf{V}_1, \mathbf{B}_1] = \texttt{gen-GK}(\mathbf{A}, \mathbf{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{b}, 1)$
  while not converged and $k \leq k_{max}$ **do**
    $[\mathbf{U}_k, \mathbf{V}_k, \mathbf{B}_k] = \texttt{gen-GK}(\mathbf{A}, \mathbf{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{b}, k)$
    $[\mathbf{z}_k, \lambda] = \texttt{projected}(\mathbf{B}_k, \text{method})$
    check for convergence
  end while
  $\mathbf{s}_k = \mathbf{QV}_k\mathbf{z}_k$

covariance, and then show theoretical results that demonstrate the accuracy of our approximated posterior covariance and the accuracy of the posterior distribution.

2.3.1 Review of Existing Methods

Others have worked on efficient approximations of $\mathbf{\Gamma}_{\text{post}}$ based on a low-rank perturbation to the prior covariance matrix [44, 23, 24, 110]. To explain this approach, first rewrite the posterior covariance matrix (2.5) as

$$\mathbf{\Gamma}_{\text{post}} = \lambda^{-2}\mathbf{Q}^{1/2}(\mathbf{I} + \lambda^{-2}\mathbf{Q}^{1/2}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{A}\mathbf{Q}^{1/2})^{-1}\mathbf{Q}^{1/2}.$$

One could then compute a low-rank approximation

$$\mathbf{Q}^{1/2}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{A}\mathbf{Q}^{1/2} \approx \mathbf{V}_k\mathbf{\Lambda}_k\mathbf{V}_k^\top$$

using the singular value decomposition (SVD) and use this to develop an efficient representation of $\mathbf{\Gamma}_{\text{post}}$ as

$$\mathbf{\Gamma}_{\text{post}} \approx \lambda^{-2}(\mathbf{Q} - \mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{D}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2}) \quad \text{where} \quad \mathbf{D}_k = \mathbf{\Lambda}_k(\mathbf{\Lambda}_k + \lambda^2\mathbf{I})^{-1}.$$

The computation of the SVD is expensive, so the authors in [23, 104] use a randomized approach to efficiently compute a low-rank approximation. We use the Krylov subspaces generated by the genHyBR method.

2.3.2 Approximating the Posterior

To begin, recall that

$$\mathbf{\Gamma}_{\text{post}} \equiv (\lambda^2 \mathbf{Q}^{-1} + \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{A})^{-1} = (\lambda^2 \mathbf{Q}^{-1} + \mathbf{H})^{-1},$$

where we define $\mathbf{H} \equiv \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{A}$. We will use the genHyBR method to compute a low rank approximation of $\mathbf{H}$. We define the Ritz pairs $(\vartheta, \mathbf{y})$, the approximate eigenvalues and vectors, obtained as the solution of the following eigenvalue problem,

$$(\mathbf{H}\mathbf{Q}\mathbf{V}_k\mathbf{y} - \vartheta \mathbf{V}_k\mathbf{y}) \ \perp_{\mathbf{Q}} \ \text{Span}\left\{\mathbf{V}_k\right\}.$$

Here the orthogonality condition $\perp_{\mathbf{Q}}$ is defined with respect to the weighted inner product $\langle \cdot, \cdot \rangle_{\mathbf{Q}}$. The Ritz pairs can be obtained by the solution of the eigenvalue problem

$$\mathbf{B}_k^\top \mathbf{B}_k \mathbf{y}_j = \vartheta_j \mathbf{y}_j \qquad j = 1, \ldots, k. \tag{2.28}$$

The Ritz pairs can be combined to express the eigenvalue decomposition

$$\mathbf{B}_k^\top \mathbf{B}_k = \mathbf{Y}_k \mathbf{\Theta}_k \mathbf{Y}_k^\top$$

and the accuracy of the Ritz pairs can be quantified as

$$\|\mathbf{H}\mathbf{Q}\mathbf{V}_k\mathbf{y}_j - \vartheta_j \mathbf{V}_k\mathbf{y}_j\|_{\mathbf{Q}} = \alpha_{k+1}\beta_{k+1}|\mathbf{e}_k^\top \mathbf{y}_j| \qquad j = 1, \ldots, k. \tag{2.29}$$

The best k-rank approximation of $\mathbf{H}$ over the space $\mathcal{K}_k \equiv \mathcal{K}_k(\mathbf{H}\mathbf{Q}, \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b})$ with basis $\mathbf{V}_k$ is given by $\mathbf{H} \approx \mathbf{V}_k \mathbf{T}_k \mathbf{V}_k^\top$. This comes from the fact that

$$\mathbf{T}_k \equiv \mathbf{B}_k^\top \mathbf{B}_k = \min_{\mathbf{\Delta} \in \mathbf{R}^{k \times k}} \|\mathbf{H}\mathbf{Q}\mathbf{V}_k - \mathbf{V}_k\mathbf{\Delta}\|_{\mathbf{Q}}, \tag{2.30}$$

see Appendix A.1 for a derivation, [111] has a similar result. Here we define the matrix $\|\cdot\|_{\mathbf{Q}}$ norm to be $\|\mathbf{M}\|_{\mathbf{Q}} = \max_{\|\mathbf{x}\|_2 = 1} \|\mathbf{M}\mathbf{x}\|_{\mathbf{Q}}$.

An approximation of this kind has been previously explored in [104, 44, 23, 24]; however, the error estimates developed in the above references assume that the exact eigenpairs are available. If the Ritz pairs converge to the exact eigenpairs of the matrix $\mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}^{1/2}$, then furthermore, the optimality result in [110, Theorem 2.3] applies here as

well.

We use the following low-rank approximation of $\mathbf{H}$ which is constructed using the either using the genHyBR iterates as

$$\widehat{\mathbf{H}} \equiv \mathbf{V}_k \mathbf{T}_k \mathbf{V}_k^\top. \tag{2.31}$$

Using this low-rank approximation, we can define the *approximate posterior distribution* $\widehat{\pi}_{\text{post}}$ by the measure $\mathcal{N}(\mathbf{s}_k, \widehat{\mathbf{\Gamma}}_{\text{post}})$, which is a Gaussian distribution with covariance matrix

$$\widehat{\mathbf{\Gamma}}_{\text{post}} \equiv (\lambda^2 \mathbf{Q}^{-1} + \widehat{\mathbf{H}})^{-1} \tag{2.32}$$

and mean $\mathbf{s}_k$ defined in (2.21). Using (2.32), we note that

$$\mathbf{s}_k = \boldsymbol{\mu} + \widehat{\mathbf{\Gamma}}_{\text{post}} \mathbf{A}^\top \mathbf{\Gamma}_{\text{noise}}^{-1} \mathbf{b}. \tag{2.33}$$

See Appendix A.2 for a derivation.

2.3.3 Accuracy of Posterior Covariance

In order to discuss the theoretical accuracy of the posterior covariance approximation we first derive a way to monitor the accuracy of the low-rank approximation using the information available from the gen-GK method. This result is similar to [108, Proposition 3.3].

Proposition 2.3.1. *Let* $\mathbf{H}_\mathbf{Q} = \mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}^{1/2}$ *and* $\widehat{\mathbf{H}}_\mathbf{Q} = \mathbf{Q}^{1/2}\widehat{\mathbf{H}}\mathbf{Q}^{1/2}$. *After running* k *steps of Algorithm 2.2.2, the error in the low-rank approximation* $\widehat{\mathbf{H}}$, *measured as*

$$\omega_k = \|\mathbf{H}_\mathbf{Q} - \widehat{\mathbf{H}}_\mathbf{Q}\|_F, \tag{2.34}$$

satisfies the recurrence

$$\omega_k^2 = \omega_{k+1}^2 + 2|\alpha_{k+1}\beta_{k+1}|^2 + |\alpha_{k+1}^2 + \beta_{k+2}^2|^2, \quad \text{for } k = 1, \ldots, n-2.$$

Proof. First, we recognize that $\widehat{\mathbf{H}} = \mathbf{V}_k \mathbf{T}_k \mathbf{V}_k^\top$, where $\mathbf{T}_k = \mathbf{V}_k^\top \mathbf{Q}\mathbf{H}\mathbf{Q}\mathbf{V}_k$ is a tridiagonal

matrix of the form

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1^2+\beta_2^2 & \alpha_2\beta_2 & & & \\ \alpha_2\beta_2 & \alpha_2^2+\beta_3^2 & \alpha_3\beta_3 & & \\ \ddots & \ddots & \ddots & & \\ & \alpha_{k-1}\beta_{k-1} & \alpha_{k-1}^2+\beta_k^2 & \alpha_k\beta_k & \\ & & \alpha_k\beta_k & \alpha_k^2+\beta_{k+1}^2 \end{bmatrix}.$$

For simplicity denote $\widehat{\mathbf{V}}_k = \mathbf{Q}^{1/2}\mathbf{V}_k$ and note that the columns of $\widehat{\mathbf{V}}_k$ are orthonormal with respect to the standard inner product. We write

$$\widehat{\mathbf{H}}_{\mathbf{Q}} = \mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2} = \mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{V}_k^\top\mathbf{Q}\mathbf{H}\mathbf{Q}\mathbf{V}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2} = \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}\widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top.$$

Then we write

$$\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}} = (\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\mathbf{H}_{\mathbf{Q}} + \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top).$$

The observation that $(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\mathbf{H}_{\mathbf{Q}} \perp \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)$ with respect to the trace inner product means we can apply Pythagoras' theorem to obtain

$$\omega_k^2 = \|(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\mathbf{H}_{\mathbf{Q}}\|_F^2 + \|\widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\|_F^2.$$

The second term is easy since using the gen-GK relationships, we have

$$\widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top) = \alpha_{k+1}\beta_{k+1}\widehat{\mathbf{v}}_k\widehat{\mathbf{v}}_{k+1}^\top,$$

and thus $\|\alpha_{k+1}\beta_{k+1}\widehat{\mathbf{v}}_k\widehat{\mathbf{v}}_{k+1}^\top\|_F^2 = |\alpha_{k+1}\beta_{k+1}|^2$.
For the first term, denote $\eta_k = \|(\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\mathbf{H}_{\mathbf{Q}}\|_F$, so that

$$\omega_k^2 = \eta_k^2 + |\alpha_{k+1}\beta_{k+1}|^2. \tag{2.35}$$

Then write $\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top = \mathbf{I} - \widehat{\mathbf{V}}_{k+1}\widehat{\mathbf{V}}_{k+1}^\top + \widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^\top$ and again apply Pythagoras' theorem to get

$$\eta_k^2 = \eta_{k+1}^2 + \|\widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^\top\mathbf{H}_{\mathbf{Q}}\|_F^2.$$

From the gen-GK relations, it can be verified that

$$\begin{aligned}\mathbf{H}_{\mathbf{Q}}\widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^{\top} = \alpha_{k+1}\beta_{k+1}\widehat{\mathbf{v}}_{k}\widehat{\mathbf{v}}_{k+1}^{\top} + (\alpha_{k+1}^2 + \beta_{k+2}^2)\widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^{\top} \\ + \alpha_{k+2}\beta_{k+2}\widehat{\mathbf{v}}_{k+2}\widehat{\mathbf{v}}_{k+1}^{\top}.\end{aligned} \tag{2.36}$$

Since each term is mutually orthogonal, this implies

$$\eta_k^2 = \eta_{k+1}^2 + |\alpha_{k+1}\beta_{k+1}|^2 + |\alpha_{k+1}^2 + \beta_{k+2}^2|^2 + |\alpha_{k+2}\beta_{k+2}|^2.$$

Together with (2.35), we get the desired recurrence. □

This proposition shows that, in exact arithmetic, the error in the low-rank approximation $\widehat{\mathbf{H}}$ to $\mathbf{H}$ decreases monotonically as the iterations progress. In [108], the authors provide analysis for the case when the noise and prior covariance are the identity, that for finite precision if the orthogonality is maintained during the Lanczos process then a accurate low-rank approximation can be produced. Estimates for ω_k can be obtained in terms of the singular values of $\mathbf{\Gamma}_{\text{noise}}^{-1/2}\mathbf{A}\mathbf{Q}^{1/2}$ following the approach in [108, Theorem 3.2] and [66, Theorem 2.7]. However, we do not pursue them here.

The recurrence relation in Proposition 2.3.1 can be used to derive the following error estimates for $\mathbf{\Gamma}_{\text{post}}$.

Theorem 2.3.1. *The approximate posterior covariance matrix* $\widehat{\mathbf{\Gamma}}_{\text{post}}$ *satisfies*

$$\|\mathbf{\Gamma}_{post} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F \leq \lambda^{-2} \min\left\{\omega_k\lambda^{-2}\|\mathbf{Q}\|_2, \frac{\omega_k\|\mathbf{Q}\|_F}{\lambda^2 + \omega_k}\right\}.$$

Proof. We now consider the error in the posterior covariance matrix. For the first bound, using

$$\mathbf{\Gamma}_{\text{post}} = \mathbf{Q}^{1/2}(\lambda^2\mathbf{I} + \mathbf{H}_{\mathbf{Q}})^{-1}\mathbf{Q}^{1/2},$$

we have

$$\begin{aligned}\|\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F &\leq \|\mathbf{Q}\|_2\|(\lambda^2\mathbf{I} + \mathbf{H}_{\mathbf{Q}})^{-1} - (\lambda^2\mathbf{I} + \widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\|_F &(2.37)\\ &= \lambda^{-2}\|\mathbf{Q}\|_2\|(\mathbf{I} + \lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\|_F. &(2.38)\end{aligned}$$

With $f(x) = x/(1+x)$, it is verifiable that

$$(\mathbf{I} + \lambda^{-2}\mathbf{H_Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1} = f(\lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q}) - f(\lambda^{-2}\mathbf{H_Q}).$$

The function f is operator monotone [18, Proposition V.1.6] and satisfies $f(0) = 0$. Since both $\lambda^{-2}\mathbf{H_Q}$ and $\lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q}$ are positive semi-definite, using [18, Theorem X.1.3], we obtain

$$\|(\mathbf{I} + \lambda^{-2}\mathbf{H_Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1}\|_F \leq \||\mathbf{E}|(\mathbf{I} + |\mathbf{E}|)^{-1}\|_F,$$

where we let $\mathbf{E} = \lambda^{-2}(\mathbf{H_Q} - \widehat{\mathbf{H}}_\mathbf{Q})$, and $|\mathbf{E}| = (\mathbf{E}^*\mathbf{E})^{1/2}$. Note that both $|\mathbf{E}|$ and $\mathbf{E}$ have the same singular values, so $\||\mathbf{E}|\|_F = \|\mathbf{E}\|_F$. Since $|\mathbf{E}|$ is positive semi-definite, the singular values of $(\mathbf{I} + |\mathbf{E}|)^{-1}$ are at most 1. By submultiplicativity inequality and $\||\mathbf{E}|\|_F = \|\mathbf{E}\|_F$, we have

$$\|(\lambda^2\mathbf{I} + \mathbf{H_Q})^{-1} - (\lambda^2\mathbf{I} + \widehat{\mathbf{H}}_\mathbf{Q})^{-1}\|_F \leq \|\lambda^{-2}(\mathbf{H_Q} - \widehat{\mathbf{H}}_\mathbf{Q})\|_F = \lambda^{-2}\omega_k \tag{2.39}$$

and hence the desired result:

$$\|\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F \leq \lambda^{-2}\|\mathbf{Q}\|_2\|\lambda^{-2}(\mathbf{H_Q} - \widehat{\mathbf{H}}_\mathbf{Q})\|_F = \lambda^{-4}\omega_k\|\mathbf{Q}\|_2. \tag{2.40}$$

For the second bound, we reserve the use of spectral and Frobenius norms

$$\|\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F \leq \lambda^{-2}\|\mathbf{Q}\|_F\|(\mathbf{I} + \lambda^{-2}\mathbf{H_Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1}\|_2.$$

Again, let $\mathbf{E} = \lambda^{-2}(\mathbf{H_Q} - \widehat{\mathbf{H}}_\mathbf{Q})$, and use [18, Theorem X.1.1] with $f(x) = x/(1+x)$, to obtain

$$\|(\mathbf{I} + \lambda^{-2}\mathbf{H_Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1}\|_2 \leq \frac{\|\mathbf{E}\|_2}{1 + \|\mathbf{E}\|_2}.$$

It is readily verified that if $0 \leq a \leq b$, then $a(1+a)^{-1} \leq b(1+b)^{-1}$, and so

$$\|(\mathbf{I} + \lambda^{-2}\mathbf{H_Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1}\|_2 \leq \frac{\|\mathbf{E}\|_2}{1 + \|\mathbf{E}\|_2} \leq \frac{\|\mathbf{E}\|_F}{1 + \|\mathbf{E}\|_F} = \frac{\omega_k}{\lambda^2 + \omega_k}. \tag{2.41}$$

The recognition that $\|\mathbf{E}\|_F = \lambda^{-2}\omega_k$ completes the proof. □

Theorem 2.3.1 quantifies the error in the posterior covariance matrix in the Frobenius norm. However, the authors in [110] argue that the Frobenius norm is not the appropriate

metric to measure the distance between covariance matrices. Instead, they advocate the Förstner distance since it respects the geometry of the cone of positive definite covariance metrics. We take a different approach and consider metrics between the approximate and the true posterior distributions.

2.3.4 Accuracy of Posterior Distribution

The Kullback-Leibler (KL) divergence is a measure of "distance" between two different probability distributions. The KL divergence is not a true metric on the set of probability measures, since it is not symmetric and does not satisfy the triangle inequality [112]; despite this the the KL divergence is widely used. ~~Despite these shortcomings, the KL divergence is widely used since it has many favorable properties.~~ Both the true and the approximate posterior distributions are Gaussian, so the KL divergence takes the form (using [112, Exercise 5.2]):

$$D_{\mathrm{KL}}(\widehat{\pi}_{\mathrm{post}}\|\pi_{\mathrm{post}}) = \frac{1}{2}\left[\mathsf{trace}(\boldsymbol{\Gamma}_{\mathrm{post}}^{-1}\widehat{\boldsymbol{\Gamma}}_{\mathrm{post}}) + \|\mathbf{s}_{\mathrm{post}} - \mathbf{s}_k\|^2_{\boldsymbol{\Gamma}_{\mathrm{post}}^{-1}} - n + \log\frac{\det\boldsymbol{\Gamma}_{\mathrm{post}}}{\det\widehat{\boldsymbol{\Gamma}}_{\mathrm{post}}}\right]. \quad (2.42)$$

Note that this definition is for a fixed n, but is not valid as $n \to \infty$. We first present a result that can be used to monitor the accuracy of the trace of $\mathbf{H}_{\mathbf{Q}}$.

Proposition 2.3.2. *Let $\theta_k = \mathsf{trace}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}})$. Then θ_k, satisfies the recurrence relation*

$$\theta_k = \theta_{k+1} + (\alpha_{k+1}^2 + \beta_{k+1}^2)^2 \;\; \textit{for } k = 1, \ldots, n-1.$$

Proof. The linearity and cyclic property of trace estimator implies

$$\begin{aligned}\theta_k &= \mathsf{trace}\left(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}\widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\right)\\ &= \mathsf{trace}\left((\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top)\mathbf{H}_{\mathbf{Q}}\right).\end{aligned}$$

As in the proof of Proposition 2.3.1, write $\mathbf{I} - \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top = \mathbf{I} - \widehat{\mathbf{V}}_{k+1}\widehat{\mathbf{V}}_{k+1}^\top + \widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^\top$, so that

$$\theta_k = \theta_{k+1} + \mathsf{trace}(\widehat{\mathbf{v}}_{k+1}\widehat{\mathbf{v}}_{k+1}^\top\mathbf{H}_{\mathbf{Q}}).$$

The proof is finished if we apply the trace to the right hand side of (2.36). □

Note that the Cauchy interlacing theorem [65, Theorem 4.3.17] implies that θ_k is non-negative; therefore, as with Proposition 2.3.1, this result implies that θ_k is monotonically decreasing.

Theorem 2.3.2. *At the end of $k < n$ iterations, the KL divergence from the approximate posterior to the true posterior distribution satisfies*

$$0 \leq D_{KL}(\widehat{\pi}_{post} \| \pi_{post}) \leq \frac{\lambda^{-2}}{2} \left[\theta_k + \frac{\omega_k^2}{\lambda^2 + \omega_k} \alpha_1^2 \beta_1^2 \right].$$

Proof. The lower bound follows from the property of the KL divergence and the fact that the distributions are not degenerate. The proof for the upper bound begins by providing an alternate expression for the error in the KL divergence.

$$D_{\mathrm{KL}}(\widehat{\pi}_{\mathrm{post}} \| \pi_{\mathrm{post}}) = \frac{1}{2} \left[\mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3 \right],$$

where

$$\begin{aligned}
\mathcal{E}_1 &= \mathsf{trace}(\widehat{\mathbf{\Gamma}}_{\mathrm{post}} \mathbf{\Gamma}_{\mathrm{post}}^{-1}) - n, \\
\mathcal{E}_2 &= \log\det(\mathbf{\Gamma}_{\mathrm{post}}) - \log\det(\widehat{\mathbf{\Gamma}}_{\mathrm{post}}), \\
\mathcal{E}_3 &= \| \mathbf{s}_{\mathrm{post}} - \mathbf{s}_k \|^2_{\mathbf{\Gamma}_{\mathrm{post}}^{-1}}.
\end{aligned}$$

We will tackle each term individually. For the first term $\mathcal{E}_1$, apply the second part of Appendix A.3 to obtain

$$\begin{aligned}
\mathsf{trace}(\widehat{\mathbf{\Gamma}}_{\mathrm{post}} {\mathbf{\Gamma}_{\mathrm{post}}}^{-1}) &= \mathsf{trace}\left[(\mathbf{I} + \lambda^{-2} \widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} (\mathbf{I} + \lambda^{-2} \mathbf{H}_{\mathbf{Q}}) \right] \\
&\leq n + \lambda^{-2} \mathsf{trace}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}}).
\end{aligned}$$

Therefore, $\mathcal{E}_1 \leq \lambda^{-2} \theta_k$.

The second term $\mathcal{E}_2$ simplifies since

$$\log\det(\mathbf{\Gamma}_{\mathrm{post}}) - \log\det(\widehat{\mathbf{\Gamma}}_{\mathrm{post}}) = \log\det(\mathbf{I} + \lambda^{-2} \widehat{\mathbf{H}}_{\mathbf{Q}}) - \log\det(\mathbf{I} + \lambda^{-2} \mathbf{H}_{\mathbf{Q}}).$$

Let $\mathbf{M} = \lambda^{-2}(\mathbf{H}_{\mathbf{Q}})$, then with $\mathbf{P} = \widehat{\mathbf{V}}_k \widehat{\mathbf{V}}_k^\top$ we have $\lambda^{-2} \widehat{\mathbf{H}}_{\mathbf{Q}} = \mathbf{PMP}$. When we apply the third inequality in Appendix A.3 we have that $\mathcal{E}_2 \leq 0$.

For the third term, notice that

$$\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}} = \lambda^{-2}\mathbf{Q}^{1/2}\left(\left(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}}\right)^{-1} - \left(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}}\right)^{-1}\right)\mathbf{Q}^{1/2}$$

and let $\mathbf{D} = (\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}$. Then

$$\|\mathbf{s}_{\text{post}} - \mathbf{s}_k\|^2_{\mathbf{\Gamma}_{\text{post}}{}^{-1}} = \widehat{\mathbf{b}}^\top\mathbf{D}(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})\mathbf{D}\widehat{\mathbf{b}} \leq \|\mathbf{D}\widehat{\mathbf{b}}\|_2\|(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})\mathbf{D}\widehat{\mathbf{b}}\|_2,$$

where $\widehat{\mathbf{b}} = \mathbf{Q}^{1/2}\mathbf{A}^\top\mathbf{\Gamma}^{-1}_{\text{noise}}\mathbf{b}$. The inequality is due to Cauchy-Schwartz. Using (2.41), we can bound

$$\|\mathbf{D}\widehat{\mathbf{b}}\|_2 \leq \frac{\omega_k\|\widehat{\mathbf{b}}\|_2}{\lambda^2+\omega_k}.$$

Next, with $\mathbf{E} = \lambda^{-2}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}})$, consider the simplification

$$\begin{aligned}(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})\mathbf{D} &= \mathbf{I} - (\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} \\ &= (\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} + (\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} \\ &= -\mathbf{E}(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1},\end{aligned}$$

so that $\|(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})\mathbf{D}\widehat{\mathbf{b}}\|_2 \leq \lambda^{-2}\omega_k\|\widehat{\mathbf{b}}\|_2$. Here, we have used submultiplicativity and the fact that singular values of $(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}$ are at most 1. We also have that $\|\widehat{\mathbf{b}}\|_2 = \alpha_1\beta_1$ (this comes from the fact $\widehat{\mathbf{b}} = \mathbf{Q}^{1/2}\mathbf{A}^\top\mathbf{\Gamma}^{-1}_{\text{noise}}\mathbf{b}$ and relations shown in Algorithm 2.2.1). Putting everything together, we see

$$\mathcal{E}_3 \leq \frac{\lambda^{-2}\omega_k^2\alpha_1^2\beta_1^2}{\lambda^2+\omega_k}.$$

Gathering the bounds for $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ we have the desired result. □

Both θ_k and ω_k are monotonically decreasing, implying that the KL divergence between the true and the approximate posterior is likely getting smaller as the iterations progress. This theorem can be useful in providing bounds for the error using other metrics.

For example, consider the Hellinger metric and Total Variation (TV) distance denoted by $d_H(\pi_{\text{post}}, \widehat{\pi}_{\text{post}})$ and $d_{\text{TV}}(\pi_{\text{post}}, \widehat{\pi}_{\text{post}})$ respectively. Combining Pinsker's inequality [112,

Theorem 5.4] and Kraft's inequality [112, Theorem 5.10], we have the following relationship

$$d_H^2(\pi_{\text{post}}, \widehat{\pi}_{\text{post}}) \leq d_{\text{TV}}(\pi_{\text{post}}, \widehat{\pi}_{\text{post}}) \leq \sqrt{2D_{\text{KL}}(\widehat{\pi}_{\text{post}} \| \pi_{\text{post}})}. \tag{2.43}$$

Thus, Theorem 2.3.2 can be used to find upper bounds for the Hellinger metric and the TV distance between the true and approximate posterior distributions. Furthermore, suppose $f : (\mathbb{R}^n, \|\cdot\|_{\mathbb{R}^n}) \to (\mathbb{R}^d, \|\cdot\|_{\mathbb{R}^d})$ is a function with finite second moments with respect to both distributions, then by [112, Proposition 5.12]

$$\|\mathbb{E}_{\pi_{\text{post}}}[f] - \mathbb{E}_{\widehat{\pi}_{\text{post}}}[f]\|_{\mathbb{R}^n} \leq 2\sqrt{\mathbb{E}_{\pi_{\text{post}}}[\|f\|_{\mathbb{R}^d}^2] + \mathbb{E}_{\widehat{\pi}_{\text{post}}}[\|f\|_{\mathbb{R}^d}^2]}\, d_H(\pi_{\text{post}}, \widehat{\pi}_{\text{post}}).$$

This implies that the error in the expectation of a function computed using the approximate posterior instead of the true posterior can be bounded by combining (2.43) and Theorem 2.3.2.

2.3.5 Computation of Information-Theoretic Metrics

In addition to providing a measure of distance between the true and approximate posterior distributions, the KL divergence can also be used to measure the information gain between the prior and the posterior distributions. Similar to the derivation in (2.42), since both π_{prior} and π_{post} are Gaussian, the KL divergence takes the form

$$\begin{aligned} D_{\text{KL}}(\pi_{\text{post}} \| \pi_{\text{prior}}) &= \frac{1}{2}\left[\mathsf{trace}(\lambda^2 \mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}}) + \lambda^2 \|\mathbf{s}_{\text{post}} - \boldsymbol{\mu}\|_{\mathbf{Q}^{-1}}^2 - n - \log\det(\lambda^2 \mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}})\right] \\ &= \frac{1}{2}\left[\mathsf{trace}\left((\mathbf{I} + \lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1}\right) + \lambda^2 \|\mathbf{s}_{\text{post}} - \boldsymbol{\mu}\|_{\mathbf{Q}^{-1}}^2 - n \right. \\ &\quad \left. + \ \log\det(\mathbf{I} + \lambda^{-2}\mathbf{H}_{\mathbf{Q}})\right]. \end{aligned}$$

See Appendix A.4 for details. Then, using the approximations generated by the genHyBR method, we consider the approximation

$$D_{\text{KL}} \equiv D_{\text{KL}}(\pi_{\text{post}} \| \pi_{\text{prior}}) \approx D_{\text{KL}}(\widehat{\pi}_{\text{post}} \| \pi_{\text{prior}}) \equiv \widehat{D}_{\text{KL}}.$$

Using the facts (see Appendix A.4) that

$$\begin{aligned}\log\det(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}}) &= \log\det(\mathbf{I}+\lambda^{-2}\mathbf{T}_k),\\ \mathsf{trace}(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} &= n-\mathsf{trace}(\mathbf{T}_k(\mathbf{T}_k+\lambda^2\mathbf{I})^{-1}),\\ \|\mathbf{s}_k-\boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}} &= \|\boldsymbol{\mu}+\mathbf{Q}\mathbf{V}_k\mathbf{z}_k-\boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}\\ &= \|\mathbf{Q}\mathbf{V}_k\mathbf{z}_k\|^2_{\mathbf{Q}^{-1}} = \|\mathbf{z}_k\|_2^2\end{aligned}$$

we get

$$\widehat{D}_{\mathrm{KL}} = \frac{1}{2}\left[-\mathsf{trace}(\mathbf{T}_k(\mathbf{T}_k+\lambda^2\mathbf{I})^{-1})+\lambda^2\|\mathbf{z}_k\|_2^2+\log\det(\mathbf{I}+\lambda^{-2}\mathbf{T}_k)\right].$$

Note that all of the terms only involve $k\times k$ tridiagonal matrices and, therefore, $\widehat{D}_{\mathrm{KL}}$ can be computed in $\mathcal{O}(k^3)$ operations once the gen-GK bidiagonalization has been computed.

The following result quantifies the accuracy of the estimator for the KL divergence between the posterior and the prior. Notice that the bound is similar to Theorem 2.3.2.

Theorem 2.3.3. *The error in the KL divergence, in exact arithmetic, is given by*

$$|D_{KL}-\widehat{D}_{KL}| \le \lambda^{-2}\left[\theta_k+\frac{\lambda^2\omega_k}{\lambda^2+\omega_k}\alpha_1^2\beta_1^2\right],$$

where ω_k and θ_k were defined in Proposition 2.3.1 and 2.3.2 respectively.

Proof. The error in the KL-divergence satisfies

$$|D_{\mathrm{KL}}-\widehat{D}_{\mathrm{KL}}| \le \mathcal{E}_1+\mathcal{E}_2+\mathcal{E}_3,$$

where

$$\begin{aligned}\mathcal{E}_1 &= \frac{1}{2}\left|\mathsf{trace}(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1}-\mathsf{trace}(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right|,\\ \mathcal{E}_2 &= \frac{1}{2}\left|\log\det(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})-\log\det(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})\right|,\\ \mathcal{E}_3 &= \frac{\lambda^2}{2}\left|\|\mathbf{s}_{\mathrm{post}}-\boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}-\|\mathbf{s}_k-\boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}\right|\\ &= \frac{\lambda^2}{2}\left|(\mathbf{s}_{\mathrm{post}}-\boldsymbol{\mu})^\top\mathbf{Q}^{-1}(\mathbf{s}_{\mathrm{post}}-\boldsymbol{\mu})-(\mathbf{s}_k-\boldsymbol{\mu})^\top\mathbf{Q}^{-1}(\mathbf{s}_k-\boldsymbol{\mu})\right|.\end{aligned}$$

We tackle the first two terms together. As in the proof of Theorem 2.3.2, let $\mathbf{M}=\lambda^{-2}(\mathbf{H}_{\mathbf{Q}})$,

then with $\mathbf{P} = \widehat{\mathbf{V}}_k \widehat{\mathbf{V}}_k^\top$ we have $\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}} = \mathbf{PMP}$. We apply the first and the third parts of Lemma A.3.1 to obtain

$$\mathcal{E}_1 \leq \frac{\lambda^{-2}}{2}\mathsf{trace}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}}) \qquad \mathcal{E}_2 \leq \frac{\lambda^{-2}}{2}\mathsf{trace}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}}).$$

For the third term, let $\mathbf{s}_{\text{post}} = \mathbf{s}_k + \mathbf{e}$, then

$$\mathcal{E}_3 = \frac{1}{2}\lambda^2 |(\mathbf{s}_{\text{post}} - \mathbf{s}_k)^\top \mathbf{Q}^{-1}(\mathbf{s}_{\text{post}} - \boldsymbol{\mu}) + (\mathbf{s}_k - \boldsymbol{\mu})^\top \mathbf{Q}^{-1}\mathbf{e}|.$$

Notice that $\mathbf{e} = \mathbf{s}_{\text{post}} - \mathbf{s}_k = (\boldsymbol{\Gamma}_{\text{post}} - \widehat{\boldsymbol{\Gamma}}_{\text{post}})\mathbf{A}^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{b}$. Using Algorithm 2.2.1 we let

$$\widehat{\mathbf{b}} \equiv \mathbf{Q}^{1/2}\mathbf{A}^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{b} = \alpha_1\beta_1 \mathbf{Q}^{1/2}\mathbf{v}_1,$$

and write

$$\mathbf{Q}^{-1/2}\mathbf{e} = \left((\lambda^2\mathbf{I} + \mathbf{H}_{\mathbf{Q}})^{-1} - (\lambda^2\mathbf{I} + \widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right)\widehat{\mathbf{b}}.$$

So, the submultiplicative inequality and (2.41) implies

$$\begin{aligned}\|\mathbf{Q}^{-1/2}\mathbf{e}\|_2 &\leq \lambda^{-2}\|(\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1} - (\mathbf{I} + \lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1}\|_2 \|\widehat{\mathbf{b}}\|_2 \\ &\leq \lambda^{-2}\frac{\omega_k}{\lambda^2 + \omega_k}\alpha_1\beta_1\,,\end{aligned}$$

where we have used (2.39). Next, applying the Cauchy-Schwartz inequality

$$|\mathbf{e}^\top \mathbf{Q}^{-1}(\mathbf{s}_{\text{post}} - \boldsymbol{\mu})| \leq \|\mathbf{Q}^{-1/2}\mathbf{e}\|_2 \|\mathbf{Q}^{-1/2}(\mathbf{s}_{\text{post}} - \boldsymbol{\mu})\|_2.$$

Then, rewriting $\mathbf{s}_{\text{post}} = \boldsymbol{\mu} + \boldsymbol{\Gamma}_{\text{post}}\mathbf{A}^\top \boldsymbol{\Gamma}_{\text{noise}}^{-1}\mathbf{b}$, we have

$$\|\mathbf{Q}^{-1/2}(\mathbf{s}_{\text{post}} - \boldsymbol{\mu})\|_2 = \|(\mathbf{I} + \lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1}\widehat{\mathbf{b}}\|_2 \leq \|\widehat{\mathbf{b}}\|_2 = \alpha_1\beta_1,$$

since the singular values of $(\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}$ are less than 1. The other term is bounded in the same way. So, we have

$$\mathcal{E}_3 \leq \frac{\omega_k}{\lambda^2 + \omega_k}\alpha_1^2\beta_1^2.$$

Putting everything together along with $\mathcal{E}_1 + \mathcal{E}_2 \leq \lambda^{-2}\theta_k$ gives the desired result. □

2.4 Optimality Criteria

Optimal Experimental Design (OED) [26, 8] is an area of inverse problems which seeks to find the best experimental conditions (which we call design) to collect the measures necessary for accurately and efficiently reconstruct the parameters of interest. OED involves finding a design that optimizes certain design criteria, known as OED criteria. The criteria that we consider in this chapter quantify the uncertainty associated with the parameter reconstructions. However, computing these design criteria involve the posterior covariance matrix and therefore their computation can be prohibitively expensive for many applications of interest, such as PAT. Previous work has used randomized algorithms, [61, 1], to efficiently and accurately compute the design criteria. In this section, after reviewing the various design criteria, we show how to efficiently approximate the design criteria using the genHyBR method and discuss the error in the approximation.

2.4.1 Weighted A-Optimality

The weighted A-optimal criterion is defined as

$$\phi_A \equiv \mathsf{trace}(\mathbf{W}_A \mathbf{\Gamma}_{\text{post}}),$$

where $\mathbf{W}_A$ is a positive semi-definite matrix. The weighted A-optimal criterion minimizes the average variance of the parameter estimates [27]. We can estimate the A-optimal criterion by

$$\widehat{\phi}_A = \mathsf{trace}(\mathbf{W}_A \widehat{\mathbf{\Gamma}}_{\text{post}}).$$

Computing the criterion itself involves further work, due to the large size of the problems of interest we cannot explicitly construct $\widehat{\mathbf{\Gamma}}_{\text{post}}$, much less $\mathbf{\Gamma}_{\text{post}}$. In order to compute the optimality criterion we use the Sherman-Morrison-Woodbury formula [65, Eqn. (0.7.4.1)] and write

$$\widehat{\mathbf{\Gamma}}_{\text{post}} = (\lambda^{-2}\mathbf{Q}^{-1} + \widehat{\mathbf{H}})^{-1} = (\lambda^{-2}\mathbf{Q}^{-1} + \mathbf{V}_k^\top \mathbf{T}_k \mathbf{V}_k)^{-1} = \lambda^{-2}\left(\mathbf{Q} - \mathbf{Q}\mathbf{V}_k\left(\lambda^2 \mathbf{T}_k^{-1} + \mathbf{I}\right)^{-1}\mathbf{V}_k^\top \mathbf{Q}\right).$$

By doing so we are now able to exploit are ability to do mat-vecs with $\mathbf{Q}$. We then do a Cholesky factorization, which has a cost of $\mathcal{O}(k^3)$ [95], to get $\mathbf{L}^\top\mathbf{L} = \left(\lambda^2 \mathbf{T}_k^{-1} + \mathbf{I}\right)^{-1}$. We

then have

$$\widehat{\mathbf{\Gamma}}_{\text{post}} = \lambda^{-2}\left(\mathbf{Q} - \mathbf{Q}\mathbf{V}_k\mathbf{L}^\top\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}\right).$$

Doing this we can compute $\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top$ at a cost of $\mathcal{O}\left(nk^2\right)$, since our Algorithm 2.2.2 has already computed $\mathbf{Q}\mathbf{V}_k$. Now to find the weighted A-optimality criteria, using properties of the trace we have

$$\begin{aligned}\widehat{\phi}_A &= \mathsf{trace}\left(\mathbf{Q}\mathbf{W}_A - \mathbf{Q}\mathbf{V}_k\mathbf{L}^\top\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}\mathbf{W}_A\right)\\ &= \mathsf{trace}\left(\mathbf{Q}\mathbf{W}_A\right) - \mathsf{trace}\left(\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}\mathbf{W}_A\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top\right).\end{aligned}$$

For the case of $\mathbf{W}_A = \mathbf{I}$ the computation is much simpler since

$$\widehat{\phi}_A = \sum_{j=1}^{n}\mathbf{e}_j^\top\mathbf{Q}\mathbf{e}_j - \sum_{j=1}^{k}\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}(j,:)\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top(:,j),$$

where $\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}(:,j)$ is the j-th column and $\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top(j,:)$ is the j-th row of $\mathbf{L}\mathbf{V}_k^\top\mathbf{Q}$. The computation of the second term can be done with a cost of $\mathcal{O}\left(k^2n\right)$. Depending on the size of the problem directly computing the trace of the first term may be too expensive and randomized trace estimators [118, 102] would be a better computational method.

The general case, $\mathbf{W}_A \neq \mathbf{I}$ can be handled in a similar manner, but the computation cost can be more substantial depending on the size and structure of $\mathbf{W}_A$.

Now that we can compute $\widehat{\phi}_A$, we can find a bound on the error of the true and approximated criterion. We have

$$\begin{aligned}|\phi_A - \widehat{\phi}_A| &= |\mathsf{trace}\left(\mathbf{W}_A(\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}})\right)|\\ &= \left|\mathsf{trace}\left(\lambda^{-2}\mathbf{W}_A^{1/2}\mathbf{Q}^{1/2}\left((\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\lambda^2\mathbf{I}+\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right)\mathbf{Q}^{1/2}\mathbf{W}_A^{1/2}\right)\right|\\ &= \left|\mathsf{trace}\left(\lambda^{-2}\mathbf{Q}^{1/2}\mathbf{W}_A^{1/2}\mathbf{W}_A^{1/2}\mathbf{Q}^{1/2}\left((\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\lambda^2\mathbf{I}+\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right)\right)\right|.\end{aligned}$$

Clearly $\mathbf{Q}^{1/2}\mathbf{W}_A^{1/2}\mathbf{W}_A^{1/2}\mathbf{Q}^{1/2}$ is a symmetric matrix and $(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\lambda^2\mathbf{I}+\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}$ is a symmetric positive semi-definite matrix. We are able to use the fact proved in [1, Lemma 8], that for a square matrix $\mathbf{X}$ and symmetric positive semi-definite matrix $\mathbf{Y}$

$$|\mathsf{trace}(\mathbf{X}\mathbf{Y})| \leq \|\mathbf{X}\|_2\mathsf{trace}(\mathbf{Y})$$

to write

$$|\phi_A - \widehat{\phi}_A| \leq \lambda^{-2}\|\mathbf{Q}^{1/2}\mathbf{W}_A^{1/2}\mathbf{W}_A^{1/2}\mathbf{Q}^{1/2}\|_2 \left|\mathsf{trace}\left((\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right)\right|.$$

Using sub-multiplicativity, the fact that $\|\mathbf{X}^s\mathbf{Y}^s\| \leq \|\mathbf{XY}\|^s$ for $0 \leq s \leq 1$ for positive matrices $\mathbf{X}, \mathbf{Y}$ [18, Theorem IX.2.1], and symmetry we have

$$|\phi_A - \widehat{\phi}_A| \leq \lambda^{-2}\|\mathbf{Q}\mathbf{W}_A\|_2 \left|\mathsf{trace}\left((\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1} - (\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right)\right|.$$

We note that $\widehat{\mathbf{H}}_{\mathbf{Q}} = \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top\mathbf{H}_{\mathbf{Q}}\widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top$ and we can use Lemma A.3.1 with $\mathbf{P} = \widehat{\mathbf{V}}_k\widehat{\mathbf{V}}_k^\top$ to get

$$|\phi_A - \widehat{\phi}_A| \leq \lambda^{-4}\|\mathbf{Q}\mathbf{W}_A\|_2 \left|\mathsf{trace}\left(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}}\right)\right| = \lambda^{-4}\|\mathbf{Q}\mathbf{W}_A\|_2\theta_k. \tag{2.44}$$

We expect the bound to decrease as we take more iterations since θ_k is monotonically decreasing. We will note that in practice, an upper bound for $\|\mathbf{Q}\mathbf{W}_A\|_2$ can be computed using matrix-free techniques.

2.4.2 C-Optimality

The C-optimal criterion is defined as

$$\phi_C \equiv \mathbf{c}^\top\boldsymbol{\Gamma}_{\text{post}}\mathbf{c},$$

where $\mathbf{c} \in \mathbb{R}^n$ is a user defined vector. We will note that the C-optimal criterion is just a case of the A-optimal criterion with $\mathbf{W}_A = \mathbf{c}\mathbf{c}^\top$ since

$$\phi_C = \mathsf{trace}(\phi_C) = \mathsf{trace}\left(\mathbf{c}^\top\boldsymbol{\Gamma}_{\text{post}}\mathbf{c}\right) = \mathsf{trace}\left(\mathbf{c}\mathbf{c}^\top\boldsymbol{\Gamma}_{\text{post}}\right),$$

and $\mathbf{W}_A = \mathbf{c}\mathbf{c}^\top$ is a positive semi-definite matrix. The C-optimal criterion minimizes the variance of the best linear unbiased estimate for a given linear combination of the model parameters (determined by $\mathbf{c}$) [39]. We can approximate the C-optimal criterion by

$$\widehat{\phi}_C = \mathbf{c}^\top\widehat{\boldsymbol{\Gamma}}_{\text{post}}\mathbf{c}$$

In order to compute the approximate C-optimality criterion, we follow the same

approach in Section 2.4.1 and obtain

$$\widehat{\phi}_C = \lambda^{-2} \left(\mathbf{c}^\top \mathbf{Q}\mathbf{c} - \mathbf{c}^\top \mathbf{Q}\mathbf{V}_k \mathbf{L}^\top \mathbf{L}\mathbf{V}_k^\top \mathbf{Q}\mathbf{c} \right).$$

As before, with $\mathbf{Q}\mathbf{V}_k$ pre-computed by Algorithm 2.2.2, we can compute $\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top$ at a cost of $\mathcal{O}(nk^2)$ and $\mathbf{c}^\top\mathbf{Q}\mathbf{V}_k\mathbf{L}^\top$ at a cost of $\mathcal{O}(nk)$ and the total computational cost would be $\mathcal{O}(nk^2 + n\log n)$.

Similar to the bound for the weighted A-optimal criterion, we have

$$|\phi_C - \widehat{\phi}_C| \leq \lambda^{-4} \|\mathbf{Q}^{1/2}\mathbf{c}\mathbf{c}^\top\mathbf{Q}^{1/2}\|_2 \theta_k.$$

Clearly $\mathbf{Q}^{1/2}\mathbf{c}\mathbf{c}^\top\mathbf{Q}^{1/2}$ is a rank one matrix so the bound simplifies to

$$|\phi_C - \widehat{\phi}_C| \leq \lambda^{-4} \|\mathbf{Q}^{1/2}\mathbf{c}\|_2^2 \theta_k = \lambda^{-4} |\mathbf{c}^\top\mathbf{Q}\mathbf{c}| \theta_k. \tag{2.45}$$

We expect the bound to decrease as we take more iterations since θ_k is monotonically decreasing.

2.4.3 D-Optimality

Related to the KL divergence is the D-optimal criterion for optimal experimental design, which is defined as

$$\phi_D \equiv \log\det(\mathbf{\Gamma}_{\text{post}}) - \log\det(\lambda^{-2}\mathbf{Q}) = \log\det(\mathbf{I} + \lambda^{-2}\mathbf{H}_\mathbf{Q}).$$

The D-optimal criterion can be seen as the expected KL divergence, with the expectation taken over the posterior distribution. A precise statement of this result was stated and derived in [1, Theorem 1]. Similar to the KL divergence, we can estimate the D-optimal criterion as

$$\widehat{\phi}_D = \log\det(\mathbf{I} + \lambda^{-2}\mathbf{T}_k).$$

Computing the approximate D-optimal criterion is straight forward and relatively inexpensive since we only need to compute the determinant of a $k \times k$ matrix which has a cost of $\mathcal{O}(k^3)$ [95].

A bound for the error in the D-optimal criterion is readily seen from the proof of

Theorem 2.3.3, and is given by

$$|\phi_D - \widehat{\phi}_D| \leq \lambda^{-2}\theta_k. \tag{2.46}$$

The computation of the error bound is trivial, since all the necessary quantities have been computed using the genHyBR algorithm.

2.5 Numerical Results

In this section, we use two numerical example setups to illustrate the theoretical error estimates and the performance of the algorithms presented in this chapter. The smaller "Heat" example is used to show the performance of the theoretical bounds. The larger "PAT" example is used to show the performance of the genHyBR algorithm and the associated uncertainty estimates.

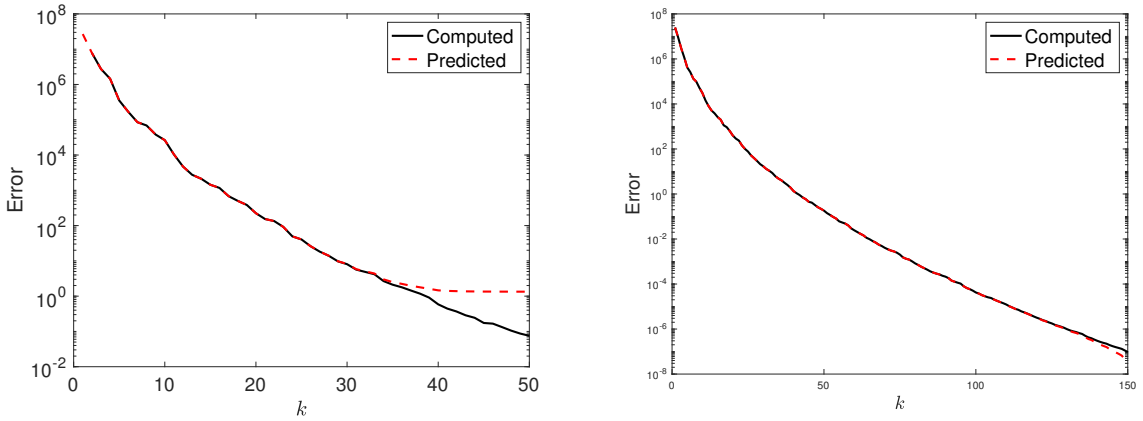
2.5.1 Heat Example

First, we investigate the accuracy of the bounds derived in Sections 2.3 and 2.4 using the `heat` example from the Regularization Toolbox [58]. Matrix $\mathbf{A}$ is 256×256, and the observations were generated as in (2.1). In the experiments, we take $\boldsymbol{\delta}$ to be 1% additive Gaussian white noise. We let $\mathbf{Q}$ be a 256×256 covariance matrix that was generated using an exponential kernel $C_\nu(x, y) = \exp(-r/\ell)$, where $\ell = 0.1$ is the correlation length. We use the genHyBR method to compute an approximate MAP estimate and simultaneously estimate a good precision parameter. Using a weighted generalized cross validation (WGCV) method, the computed precision parameter was $\lambda^2 \approx 5 \times 10^3$. The precision parameter was then fixed for the remainder of this section.

Numerical Accuracy of the Posterior Covariance

In Figure 2.2a, we track the accuracy of $\omega_k = \|\mathbf{H}_\mathbf{Q} - \widehat{\mathbf{H}}_\mathbf{Q}\|_F$ as a function of the number of iterations. It is readily seen that the error decreases considerably and monotonically with increasing k and that ω_k obtained by recursion is in close agreement with the actual error for the first 30 or so iterations. After that point, the effects of floating point errors appear to affect the accuracy of the recurrence. This is likely due to the loss of orthogonality in the Krylov basis vectors, despite using the full reorthogonalization. This is a well-known

issue in Krylov methods. However, typically the genHyBR algorithm stops much earlier (we monitor the residual to determine when to end the iterations). In this particular example, 12 iterations were sufficient.



(a) **(b)**

Figure 2.2 (a) In this plot, we provide the computed values of ω_k as a function of the iteration k. In the dotted line, we provide the values for ω_k, as computed by the recurrence relationship presented in Proposition 2.3.1. **(b)** Here, we show the computed values of θ_k as a function of the iteration k. The dotted line is θ_k computed by the recurrence relationship presented in Proposition 2.3.1.

In Figure 2.3, we provide, in the solid line, the computed errors for the posterior covariance matrix $\|\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F$, which decrease considerably with more iterations. To better illustrate the bounds in Theorem 2.3.1, we provide predicted bounds $\omega_k \lambda^{-4} \|\mathbf{Q}\|_2$ (denoted "Predicted 1") and $\frac{\omega_k \lambda^{-2} \|\mathbf{Q}\|_2}{\lambda^2 + \omega_k}$ (denoted "Predicted 2"). Though both bounds are qualitatively good, the first bound is slightly better at later iterations, the second bound is more informative at earlier iterations. This can be attributed to the difference in the behavior of ω_k in the first bound versus $\omega_k/(\lambda^2 + \omega_k)$ in the second bound. The overall bound in Theorem 2.3.1 is obtained by taking the minimum value per iteration. Since these bounds involve ω_k floating point errors effect the quality of the bound at later iterations, but we can also see that the computed error behaves well. These plots provide evidence that the low-rank approximation $\widehat{\mathbf{H}}_{\mathbf{Q}}$ constructed using available components from the gen-GK bidiagonalization are quite accurate for the practical number of iterations

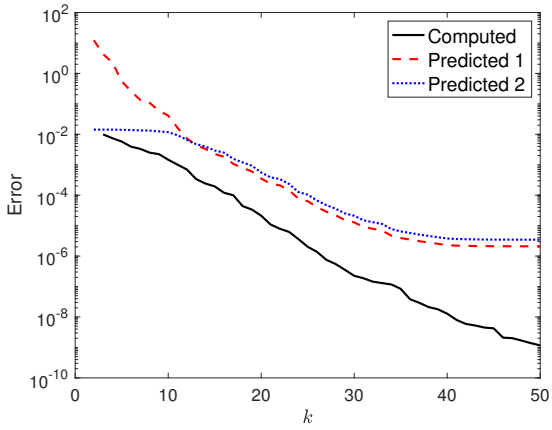


Figure 2.3 Here, we provide the errors for the posterior covariance matrix $\|\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}}\|_F$ as a function of the iteration, along with the two predicted bounds proposed in Theorem 2.3.1.

required and also that the bounds describing their behavior are informative.

Numerical Accuracy of the Posterior Distribution

For the next illustration, we use the same problem setup, but we investigate a bound for the KL divergence from the posterior to the prior distribution and the bound for the KL divergence from the approximate posterior to true posterior distribution. We first look at the recurrence relationship in Proposition 2.3.2 for $\theta_k = \mathsf{trace}(\mathbf{H}_{\mathbf{Q}} - \widehat{\mathbf{H}}_{\mathbf{Q}})$. In Figure 2.2b, we plot the error as a function of the number of iterations. As with ω_k, the error decreases significantly and monotonically as the number of iterations increases. The effects of floating point error are seen at much later iterations.

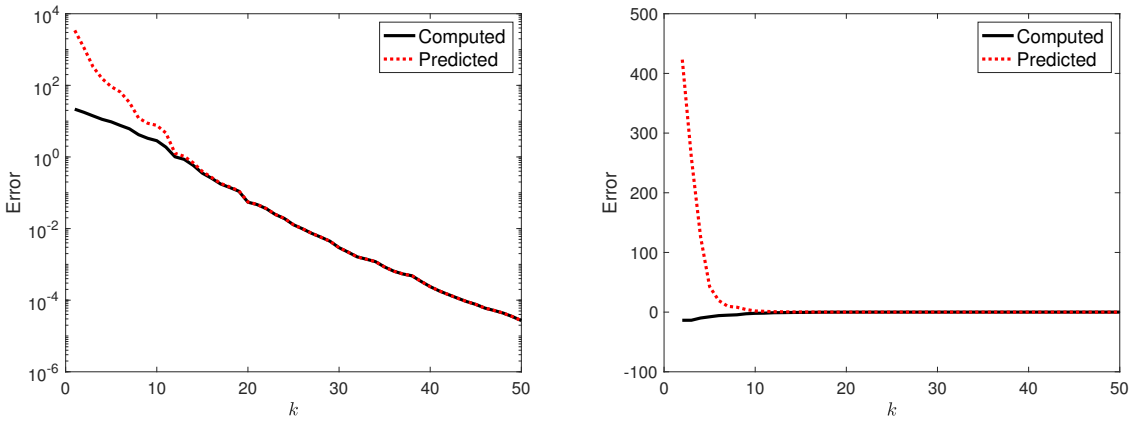
When we initially ran numerical experiments for the KL divergence in Theorem 2.3.3 we found that the bound for the quadratic term $\lambda^2\|\mathbf{s}_{\text{post}} - \boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}$ was too pessimistic, which in turn made the bound too pessimistic. For this reason, we simplified the expression for the KL divergence to

$$D_{\text{KL}} = \frac{1}{2}\left[\mathsf{trace}(\mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}}) - n - \log\det(\mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}})\right],$$

and the corresponding approximation is

$$\widehat{D}_{\text{KL}} = \frac{1}{2}\left[-\mathsf{trace}(\mathbf{T}_k(\mathbf{T}_k + \lambda^2\mathbf{I})^{-1}) + \log\det(\mathbf{I} + \lambda^{-2}\mathbf{T}_k)\right].$$

Theorem 2.3.3 then simplifies to $|D_{\mathrm{KL}} - \widehat{D}_{\mathrm{KL}}| \leq \lambda^{-2}\theta_k$. The error in the KL divergence is plotted in Figure 2.4a, along with the corresponding bound. We see that that the bound captures the behavior of the KL divergence quite well. As for the quadratic term, we found empirically that the error decreases monotonically and is comparable to the simplified expression for the KL divergence. Even the pessimistic bound of Theorem 2.3.3 suggests that the error eventually decreases to zero with enough iterations. However, a more refined analysis is needed to develop informative bounds for the quadratic term and will be considered in future work.



(a) **(b)**

Figure 2.4 (a) This figure provides the computed error in the simplified KL divergence between the approximated posterior and prior, along with the predicted bound, as a function of the iteration k. **(b)** This figure provides the computed error in the simplified KL divergence between the approximate and true posterior distribution, along with the predicted bound, as a function of the iteration k.

Similarly, for the KL divergence from the approximate to the true posterior distribution, we found the bound on the quadratic term was too pessimistic. We simplified the bound in Theorem 2.3.2 by removing the quadratic term for the same reasons as mentioned previously. In Figure 2.4b, we plotted the simplified KL divergence from the approximate to the true posterior distribution along with the bound.

Numerical Accuracy of the Optimality Criteria

Retaining the same setup as the previous experiment, we look at the accuracy of the bounds on the optimality criteria derived in Section 2.4. We start with the weighted A-optimality criteria. In Figure 2.5a, we display the bounds for the weighted A-optimality (2.44) along with the computed error. Here $\mathbf{W}_A$ was taken to be the identity matrix. In Figure 2.5b, we plot the computed and predicted difference for the C-optimality criteria (2.45, and we take $\mathbf{c}$ to be a normal random vector vector. Finally, we look at the bounds for the D-optimality criteria. In Figure 2.5c, we plot the computed and predicted difference for the D-optimality criteria (2.46). In all three cases we see that our predicted bound captures the behavior of the error.

2.5.2 PAT

In this experiment, we use the `PRspherical` test problem from the IRTools toolbox [50, 60], which models spherical means tomography such as in photo-acoustic tomography. The true image $\mathbf{s}$ and forward model matrix $\mathbf{A}$ that models spherical means tomography are provided by the toolbox. We use the default settings provided by the toolbox; see [50] for details. To simulate measurement error, we add 2% additive Gaussian noise. Our grid size is 128×128 and we define $\mathbf{Q}$ with a Matérn kernel, $\nu = 1/2$ and $\ell = 0.006$ (see Section 2.1.1 for definition). For the chosen setup, the size $\mathbf{A}$ is a sparse matrix of size 23168×16384, $\mathbf{b}$ is size 23168×1, and $\mathbf{Q}$ is a matrix, that is block-Toeplitz with Toeplitz blocks, of size 16384×16384 or $\approx 2.68 \times 10^8$ entries. For this problem it is clear that $\mathbf{Q}$ should not be stored explicitly. Instead, circulant embedding and FFT-based techniques [84] are used to efficiently perform mat-vecs.

MAP Estimate

We look at two different true images, an image we call 'smooth' provided by the IRTools toolbox and an image we will refer to as the 'blood vessel' image from the k-wave toolbox [120]. For both images we compute the MAP estimate using the genHyBR method. Since we have the true images and the realizations of the noise, we are able to calculate the optimal precision parameter λ and use it to find the solution. In Figure 2.6, we compare the true 'smooth' image with the reconstructed image found by using the optimal precision parameter and visually they appear nearly identical. We found that the

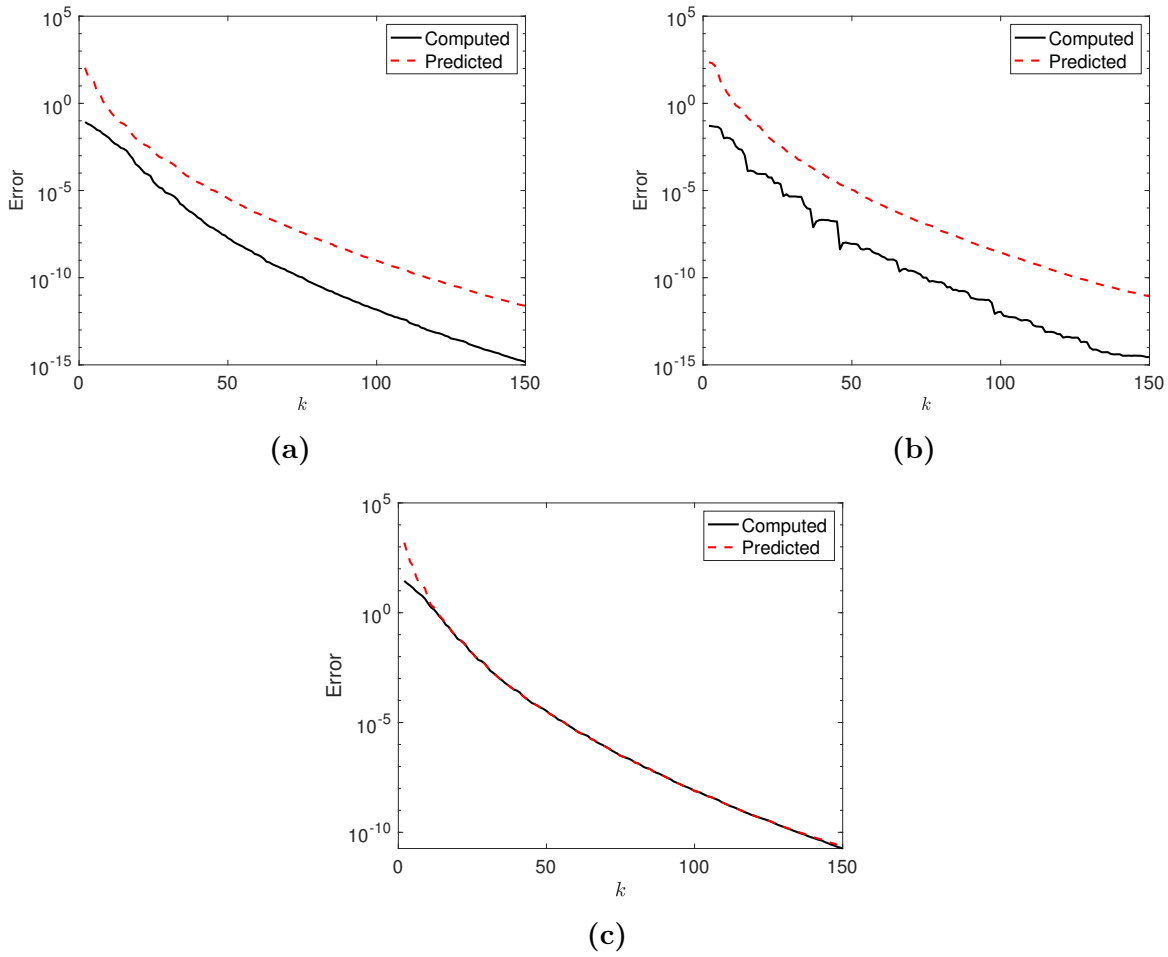


Figure 2.5 (a) The figure shows the computed and predicted difference of the true and approximated weighted A-optimality criterion, Section 2.4.1, with $\mathbf{W}_A = \mathbf{I}$ as a function of the iteration k. **(b)** Here, we show the computed and predicted difference of the true and approximated C-optimality criterion, Section 2.4.2, as a function of the iteration k. **(c)** The figure shows the computed and predicted difference of the true and approximated D-optimality criterion, Section 2.4.3, as a function of the iteration k.

optimal squared precision parameter $\lambda^2 \approx 8.31$ and that the relative error in the 2-norm is 0.62%.

For the 'blood vessel' image in Figure 2.7 we compare the true image with the image reconstructed with the genHyBR method and the optimal precision parameter. The reconstruction appears to capture the shape of the blood vessel. For this image the we found the value of the optimal squared precision parameter $\lambda^2 \approx 0.4216$ and the relative error in the 2-norm is 7.7%. The large relative error is likely because the Gaussian prior

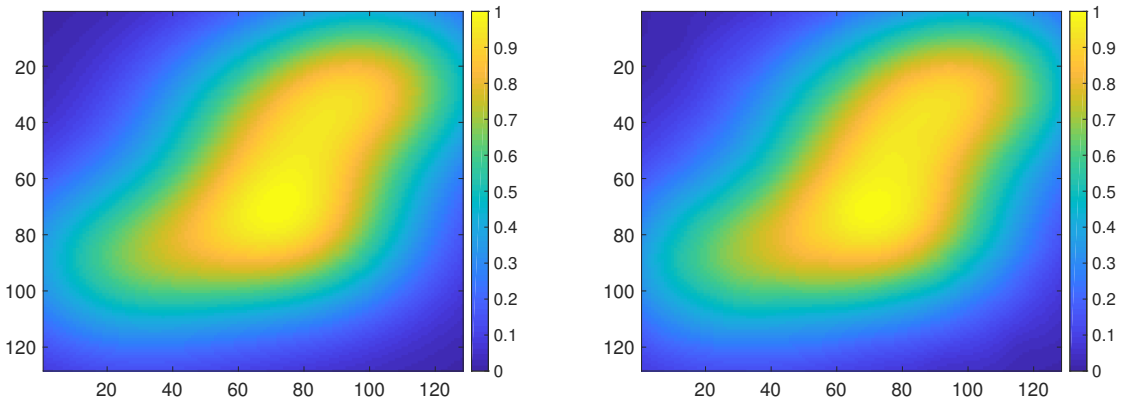


Figure 2.6 True 'smooth' image (left) and MAP estimate found with the optimal λ^2 (right).

is not able to capture edge information effectively.

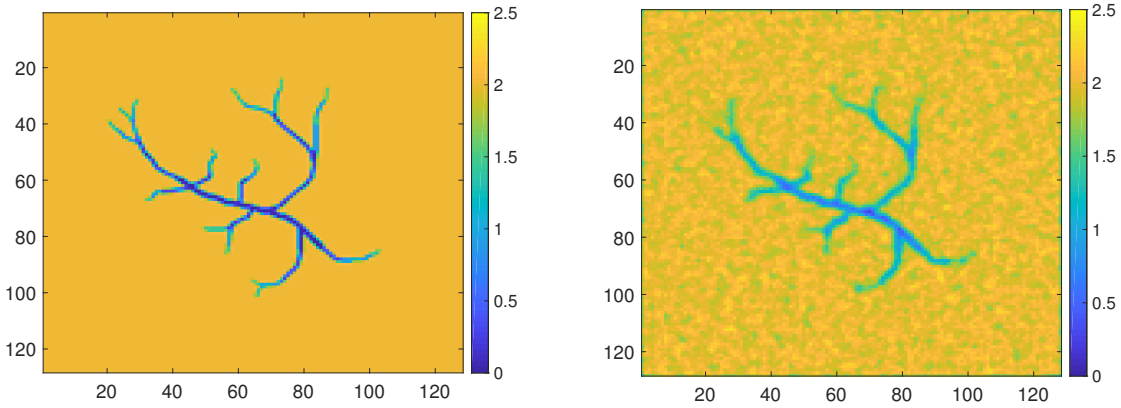


Figure 2.7 True 'blood vessel' image (left) and MAP estimate found with the optimal precision parameter (right).

Precision Parameter Selection

In the last section we used the optimal precision parameter, λ, which is not available in most real-world problems since it requires exact knowledge of the noise. In this section we look at the four different methods, described in Section 2.2.2, used to compute the precision parameter for each of our true images. In Table 2.1, we show the squared

precision parameter, λ^2, computed by each of the different methods, and in Figure 2.8 we show the reconstructions of the 'smooth' image found with those values. From the table we can see that the GCV and WGCV methods found λ^2 values closest to the optimal value and that corresponds to what is seen visually in the figure.

Table 2.1 We show the squared precision parameter value, λ^2, obtained using each of the methods and the relative error between the true and computed solutions for the 'smooth' image.

Method	λ^2	Rel. Err. (%)
Optimal	8.31	0.62
DP	2.1652e-04	4.84
GCV	24.9061	2.24
WGCV	16.9394	1.14
UPRE	0.0040	4.84

Table 2.2 has the squared precision parameter, λ^2, values computed by the different methods for the true 'blood vessel' image along with the relative error in the 2-norm for each method. Figure 2.9 shows the reconstructed images for each method. From the table we can see that the WGCV method's precision parameter was closest to the optimal. The figure shows that the λ^2 computed by the DP and UPRE methods lead to drastic under-smoothing of the reconstructed solutions, which corresponds to the larger relative error seen in the table. While the DP and UPRE methods preform badly for this choice of Matérn kernel for other choices, they achieve much better reconstructions.

Variance of Reconstructed Solution

Along with the MAP estimate, we would like to know the variance of the reconstructed parameters and give an estimate of the uncertainty. Recall that the variance is the diagonal of the covariance matrix. Due to the size of the problem we cannot form $\widehat{\mathbf{\Gamma}}_{\text{post}}$ explicitly, but we can estimate the variance using the approximations from the genHyBR method. We can compute the variance in a manner similar to the computation of the approximate C-optimality criteria discussed in Section 2.4.2. In Figure 2.10, we plot the variance for the 'smooth' and 'blood vessel' images.

Table 2.2 We show the squared precision parameter value, λ^2, obtained using each of the methods and the relative error between the true and computed solutions for the 'blood vessel' image.

Method	λ^2	Rel. Err. (%)
Optimal	0.8556	7.7
DP	3.7109e-05	59.6
GCV	1.5511	8.7
WGCV	1.0381	7.9
UPRE	4.5489e-04	59.6

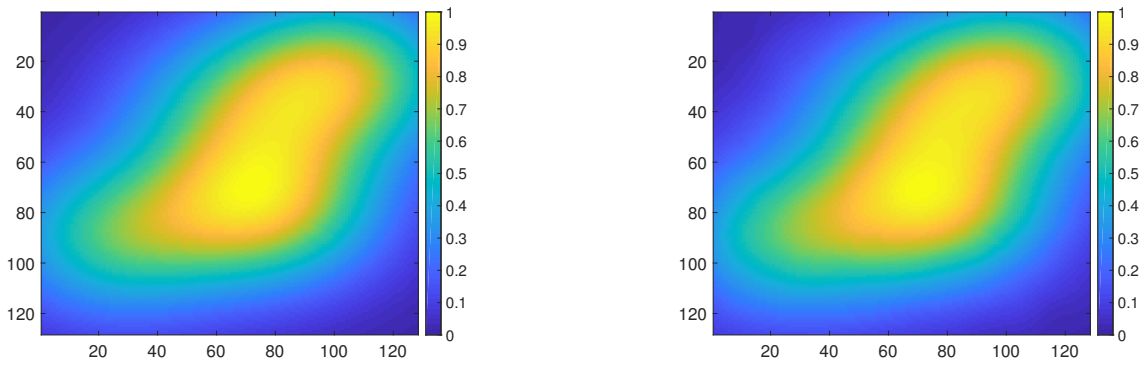
Optimality Criteria

Lastly, we look at the computation of the optimality criteria for the PAT problem. In Section 2.4 we discussed three different optimality criteria and showed how they can be computed. Unlike the "Heat" example, we cannot compute the true optimality criteria due to the large size of the problem, and here we only compute the approximations. In Figure 2.11, we plot the approximations to the A-optimal, C-optimal, and D-optimal criteria as a function of the number of iterations for the 'smooth' image. For A-optimal criterion we chose $\mathbf{W}_A = \mathbf{I}$ and for the C-optimal criterion we choose the entries of $\mathbf{c}$ to be from a random normal distribution.

2.6 Conclusion

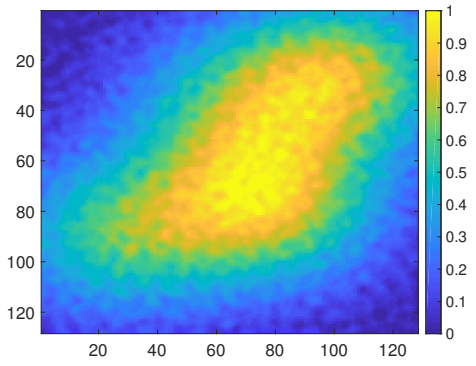
This chapter addresses the challenging problem of providing an efficient representation for the posterior covariance matrix arising in high-dimensional inverse problems. To this end, Krylov subspace methods are exploited to derive an approximation to the posterior covariance matrix as a low-rank perturbation of the prior covariance matrix. The approximation is computed using information generated from the genHyBR algorithm while computing the MAP estimate. As a result, we obtain an approximate and efficient representation for "free." Several results are presented to quantify the accuracy of this representation and of the resulting posterior distribution. We also show how to efficiently compute measures of uncertainty involving the posterior distribution.

There are several avenues for further research. The first important question is: Can we replace the bounds in the Frobenius norm by the spectral norm? The reason we employed the Frobenius norm is the recurrence relation in Proposition 2.3.1. Another issue worth

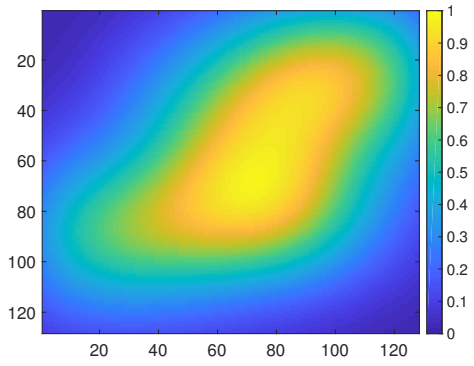


(a) True

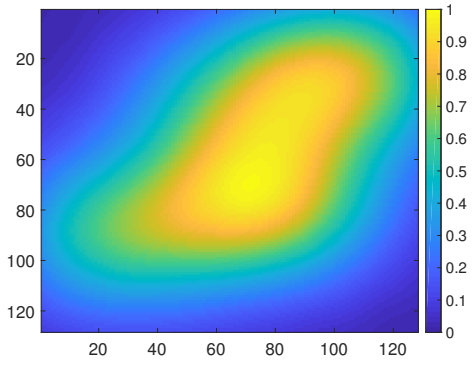
(b) Optimal



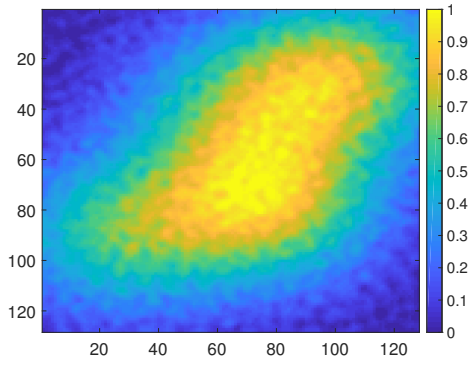
(c) DP



(d) GCV



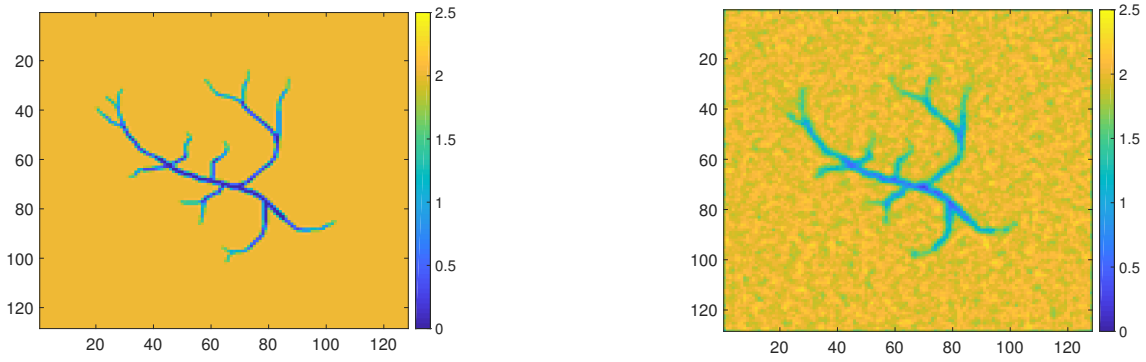
(e) WGCV



(f) UPRE

Figure 2.8 The reconstructions of the 'smooth' image for λ^2 found using the different methods.

exploring is whether we can give bounds for the error in the low-rank approximation ω_k explicitly in terms of the eigenvalues of $\mathbf{H_Q}$. This can be beneficial for deciding a priori



(a) True

(b) Optimal

(c) DP

(d) GCV

(e) WGCV

(f) UPRE

Figure 2.9 The reconstructions of the 'blood vessel' image for λ^2 found using the different methods.

the number of iterations required for an accurate low-rank approximation when the rate of decay of eigenvalues of $\mathbf{H_Q}$ is known. Finally, we are interested in exploring the use of

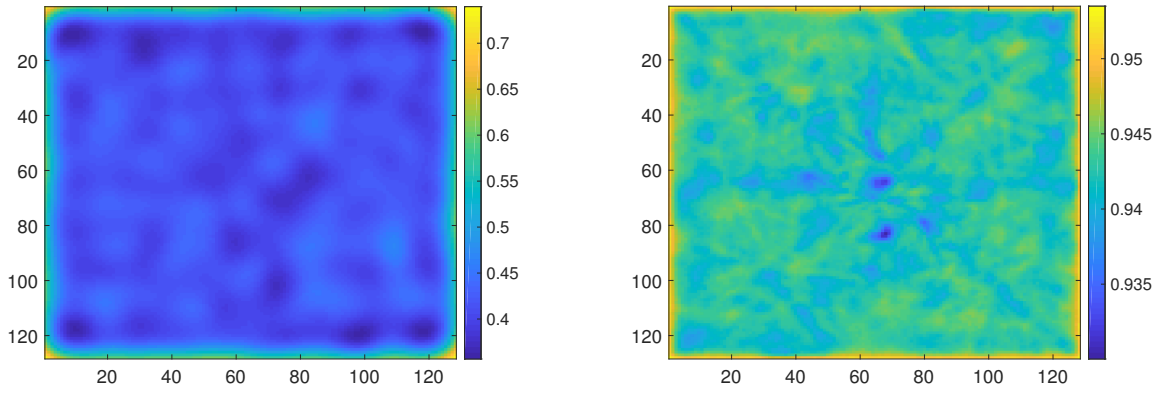
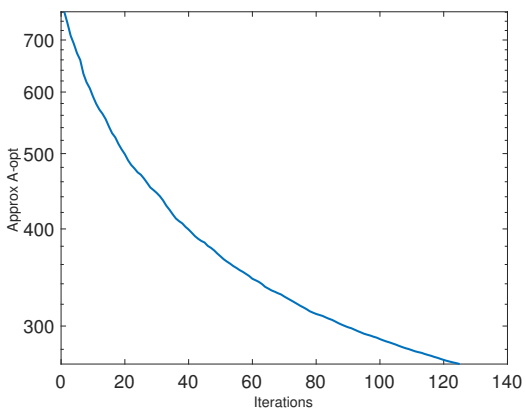
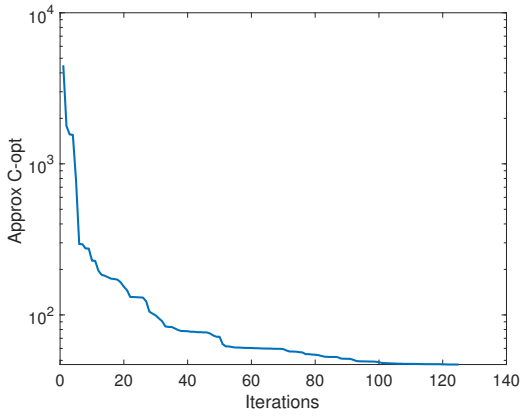


Figure 2.10 The variance field for 'smooth' (left) and 'blood vessel' (right) images.

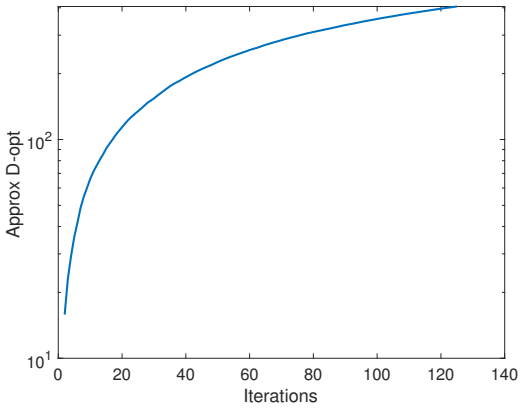
the approximate posterior distribution as a surrogate for the exact posterior distribution inside a Markov Chain Monte Carlo (MCMC) sampler. This is of particular interest for nonlinear problems where the posterior distribution is non-Gaussian. MCMC methods rely heavily on the availability of a good proposal distribution. One approach is to linearize the forward operator about the MAP estimate (the so-called Laplace's approximation) resulting in a Gaussian distribution with similar structure to π_{post}. This approximation to the true posterior distribution can be used as a proposal distribution, see for e.g. [82, 93].



(a) A-optimality



(b) C-optimality



(c) D-optimality

Figure 2.11 The approximations to various optimality criteria for the 'smooth' image as a function of the number of iterations.

CHAPTER

3

SAMPLING FROM GAUSSIAN POSTERIOR DISTRIBUTIONS

As was discussed in the previous chapter, in Bayesian inverse problems the posterior distributions are often very high-dimensional and visualizing them can be challenging. A popular method to visualize and quantify the uncertainty is to generate samples from the posterior distribution (also sometimes known as conditional realizations), which can be used for quantifying the reconstruction uncertainty. For instance, to compute the expected value of a quantity of interest $q(\cdot)$ defined as

$$\mathcal{Q} \equiv \mathbb{E}\left[q(\mathbf{s}) \mid \mathbf{d}\right] = \int_{\mathbb{R}^n} q(\mathbf{s})\pi(\mathbf{s} \mid \mathbf{d})d\mathbf{s},$$

where $\mathbf{s}$ is a random variable and $\mathbf{d}$ is data. Suppose, we have samples $\{\mathbf{s}^{(j)}\}_{j=1}^{N}$ then

$$\mathcal{Q}_N \equiv N^{-1}\sum_{j=1}^{N} q(\mathbf{s}^{(j)})$$

is the *Monte Carlo estimate* of $\mathcal{Q}$. Furthermore, the Monte Carlo estimate converges to the expected value of the quantity of interest, i.e., $\mathcal{Q}_N \to \mathcal{Q}$ as $N \to \infty$ almost surely, by the strong law of large numbers. Before describing our proposed methods to generate samples, we briefly review a few methods for sampling from high-dimensional distributions. This list is not meant to be exhaustive and we refer the reader to [113, 70, 19, 116] for a more comprehensive review.

Existing Methods

For general Bayesian inverse problems, Markov Chain Monte Carlo (MCMC) methods are popular and well-known ways to generate samples from a posterior distribution. MCMC methods construct Markov chains whose stationary or target distribution is the posterior distribution. At each iteration in a typical MCMC algorithm, a proposed state is generated from a proposal distribution. The posterior p.d.f is evaluated at the proposed state and is subjected to an accept/reject step to decide whether or not to move from the current to the proposed state. After a suitable number of iterates are discarded as part of burn-in process, the iterates from the chain can be considered to be samples from the target distribution. However, a downside of using MCMC methods is that they typically require many iterations for convergence and are computationally expensive for high-dimensional problems. For more details on MCMC the interested reader can see [79, 20].

A related class of Monte Carlo methods are Importance Sampling methods, which encompass, but are not limited to, mixture, multiple, and adaptive importance sampling [22]. Adaptive importance sampling (AIS) methods, which can include mixture and multiple methods, have become increasingly popular. AIS methods work by drawing samples from a proposal distribution, weighing the samples by looking at the mismatch between the proposal and the target (posterior) distributions, and adapting the proposal distribution based on the weights and samples [22]. Like MCMC methods, AIS methods also need many samples to converge to the posterior distribution.

In the case where the Bayesian inverse problem has Gaussian priors (but potentially non-Gaussian likelihoods) more specific methods are available. The Randomize-Then-Optimize (RTO) method [13, 12] solves an optimization problem using perturbed cost functions to generate proposals; these proposals can then be used in the context of an MCMC or importance sampling framework to obtain samples from the posterior distribution. In [131], the authors extended the RTO method to a specific non-Gaussian

prior. Randomized MAP (rMAP) method is an alternative technique which is identical to the RTO for the case that the forward problem is linear and the prior is Gaussian, and for the non-linear case they use different cost functions [129].

In this chapter, we consider Krylov based sampling methods for the linear Bayesian inverse problem with Gaussian likelihoods and priors described in Section 2.1. We continue to consider problems where computation of the square root and inverse of the prior covariance matrix are not feasible, but matrix-vector products (mat-vecs) involving $\mathbf{Q}$ can be done efficiently. The idea of using Krylov subspace methods for sampling from Gaussian random processes seems to have originated from [106]. Variants of this idea have also been proposed in [89, 28] and have found applications in Bayesian inverse problems in [54, 109]. The use of a low-rank surrogate of $\mathbf{H}_\mathbf{Q}$ (see Section 2.3.3) has also been explored in [23, 24] and is similar to Method 1 (c.f., Section 3.2) that we propose. However, none of the aforementioned methods can handle the case where the inverse prior covariance, $\mathbf{Q}^{-1}$, or $\mathbf{Q}^{-1/2}$ are not available.

Overview of Main Contributions

We use preconditioned Krylov subspace methods to generate samples from the posterior distribution that corresponds to the linear Bayesian inverse problem described in Section 2.1. We develop two different algorithms for generating samples from the posterior distribution using preconditioned Lanczos methods.

- The first proposed algorithm computes a low-rank approximation of $\mathbf{H}$, the data misfit Hessian see Section 2.3.2, using the genHyBR approach and then uses this low-rank approximation to generate samples from the *approximate* posterior distribution.
- The second proposed algorithm generates approximate samples from the *exact* posterior distribution.

A direct application of existing approaches (reviewed in Section 3.1.1) to the posterior covariance matrix is prohibitively expensive since they involve repeated applications of $\mathbf{Q}^{-1}$. To avoid this, we present several reformulations. The first approach we describe computes a low-rank approximation of $\mathbf{H}$ using the genHyBR approach and then uses this low-rank approximation to generate samples from the *approximate* posterior distribution. Any low-rank approximation can be used, provided it is sufficiently accurate. On the other hand, the second approach generates approximate samples from the *exact* posterior

distribution. Both methods use preconditioners, albeit in different ways. Before we describe the proposed algorithms, we will review Lanczos method, which is based on Krylov subspaces.

3.1 Background

Let $\bar{\boldsymbol{\nu}} \in \mathbb{R}^n$ and let $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$ be any symmetric positive definite matrix. Suppose the goal is to obtain samples from the Gaussian distribution $\mathcal{N}(\bar{\boldsymbol{\nu}}, \boldsymbol{\Gamma})$. Throughout this chapter, let $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. If we have a factorization of the form $\boldsymbol{\Gamma} = \mathbf{S}_{\boldsymbol{\Gamma}} \mathbf{S}_{\boldsymbol{\Gamma}}^\top$, then

$$\boldsymbol{\nu} = \bar{\boldsymbol{\nu}} + \mathbf{S}_{\boldsymbol{\Gamma}} \boldsymbol{\epsilon}$$

is a sample from $\mathcal{N}(\bar{\boldsymbol{\nu}}, \boldsymbol{\Gamma})$, where it can be readily shown that $\mathbb{E}[\boldsymbol{\nu}] = \bar{\boldsymbol{\nu}}$ and

$$\mathrm{Cov}(\boldsymbol{\nu}) = \mathbb{E}[(\boldsymbol{\nu} - \bar{\boldsymbol{\nu}})(\boldsymbol{\nu} - \bar{\boldsymbol{\nu}})^\top] = \mathbb{E}[\mathbf{S}_{\boldsymbol{\Gamma}} \boldsymbol{\epsilon} \boldsymbol{\epsilon} \mathbf{S}_{\boldsymbol{\Gamma}}^\top] = \boldsymbol{\Gamma}.$$

Note that any matrix $\mathbf{S}_{\boldsymbol{\Gamma}}$ that satisfies $\mathbf{S}_{\boldsymbol{\Gamma}} \mathbf{S}_{\boldsymbol{\Gamma}}^\top = \boldsymbol{\Gamma}$ can be used to generate samples. We show how Krylov subspace solvers, in particular preconditioned versions, can be used to efficiently generate approximate samples from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}^{-1})$. These approaches will be extended for sampling from the posterior distribution in Sections 3.2 and 3.3.

3.1.1 Sampling From a Gaussian Distribution Using Lanczos Process

Given $\boldsymbol{\Gamma}$ and starting guess $\boldsymbol{\epsilon}$, after K steps of the symmetric Lanczos process, we have matrix $\mathbf{W}_K = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{n \times K}$ that contains orthonormal columns and tridiagonal matrix

$$\mathbf{C}_K = \begin{bmatrix} \gamma_1 & \delta_2 & & & \\ \delta_2 & \gamma_2 & \delta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{K-1} & \gamma_{K-1} & \delta_K \\ & & & \delta_K & \gamma_K \end{bmatrix} \in \mathbb{R}^{K \times K} \tag{3.1}$$

such that in exact arithmetic we have the following relation,

$$\boldsymbol{\Gamma}\mathbf{W}_K = \mathbf{W}_K\mathbf{C}_K + \delta_{K+1}\mathbf{w}_{K+1}\mathbf{e}_K^\top \implies \mathbf{W}_K^\top\boldsymbol{\Gamma}\mathbf{W}_K = \mathbf{C}_K. \tag{3.2}$$

The Lanczos process is summarized in Algorithm 3.1.1.

Algorithm 3.1.1 Lanczos tridiagonalization

Output: $[\mathbf{W}_K, \mathbf{C}_K] = \texttt{Lanczos}(\boldsymbol{\Gamma}, \boldsymbol{\epsilon}, K)$

1: $\delta_0 = 1, \mathbf{w}_0 = \mathbf{0}, \delta_1 = \|\boldsymbol{\epsilon}\|_2, \mathbf{w}_1 = \boldsymbol{\epsilon}/\delta_1$
2: **for** $i = 1, \ldots, K$ **do**
3: $\quad\gamma_i = \mathbf{w}_i^\top\boldsymbol{\Gamma}\mathbf{w}_i$,
4: $\quad\mathbf{r} = \boldsymbol{\Gamma}\mathbf{w}_i - \gamma_i\mathbf{w}_i - \delta_{i-1}\mathbf{w}_{i-1}$
5: $\quad\mathbf{w}_{i+1} = \mathbf{r}/\delta_i$, where $\delta_i = \|\mathbf{r}\|_2$
6: **end for**

At the end of K iterations, the Lanczos process generates an orthonormal basis to the Krylov subspace

$$\mathcal{K}_K(\boldsymbol{\Gamma}, \boldsymbol{\epsilon}) = \text{Span}\{\boldsymbol{\epsilon}, \boldsymbol{\Gamma}\boldsymbol{\epsilon}, \ldots, \boldsymbol{\Gamma}^{K-1}\boldsymbol{\epsilon}\}.$$

The Lanczos iterates can be used to approximate $f(\boldsymbol{\Gamma})\boldsymbol{\epsilon}$, where f is a matrix function of interest. For sampling, $f(\cdot)$ can either be the square root or the inverse square root. Our derivation follows that in [28, Section 2.1]. Specifically, the approximation $\boldsymbol{\xi}_K^*$

$$\boldsymbol{\xi}_K^* = \mathbf{W}_K\mathbf{W}_K^\top f(\boldsymbol{\Gamma})\boldsymbol{\epsilon}.$$

is the optimal approximation in $\mathcal{K}_K$ (optimal in the sense that the 2-norm error is minimized). From Algorithm 3.1.1, we can rewrite the optimal approximation as

$$\boldsymbol{\xi}_K^* = \mathbf{W}_K\mathbf{W}_K^\top f(\boldsymbol{\Gamma})\mathbf{W}_K\delta_1\mathbf{e}_1.$$

Following [28], if we approximate $\mathbf{W}_K^\top f(\boldsymbol{\Gamma})\mathbf{W}_K \approx f(\mathbf{W}_K^\top\boldsymbol{\Gamma}\mathbf{W}_K)$ then using (3.2), we have

$$\tilde{\boldsymbol{\xi}}_K^* = \mathbf{W}_K f(\mathbf{C}_K)\delta_1\mathbf{e}_1.$$

To sample from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ we use $f(\boldsymbol{\Gamma}) = \boldsymbol{\Gamma}^{1/2}$ and to sample from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}^{-1})$, we

use $f(\mathbf{\Gamma}) = \mathbf{\Gamma}^{-1/2}$. The corresponding approximate samples from these distributions are generated as

$$\boldsymbol{\xi}_K = \mathbf{W}_K \mathbf{C}_K^{1/2} \delta_1 \mathbf{e}_1 \quad \text{and} \quad \boldsymbol{\zeta}_K = \mathbf{W}_K \mathbf{C}_K^{-1/2} \delta_1 \mathbf{e}_1, \tag{3.3}$$

respectively. We remark that the iterates generated by the Lanczos process and the matrix $\mathbf{C}_K$, depend on the realization $\boldsymbol{\epsilon}$; in other words, the Lanczos process has to be generated from scratch for each random sample. Although this can become costly if many samples are desired, the process is embarrassingly parallel across the samples. Furthermore, various methods (e.g., recycling techniques [90, 91]) can be exploited for handling multiple right hand sides efficiently; however we do not pursue them here.

Convergence of the Lanczos Process

In this section, we will let $\boldsymbol{\epsilon}$ be fixed. The approximation of the samples improves as k increases, and we expect typical convergence behavior for the Lanczos process whereby convergence to extremal (i.e., largest and smallest) eigenvalues will be fast. The following result [109, Theorem 3.3] sheds light onto the convergence of Krylov subspace methods for sampling. The error in the sample $\boldsymbol{\zeta}_K$ is given by

$$\|\mathbf{\Gamma}^{-1/2}\boldsymbol{\epsilon} - \boldsymbol{\zeta}_K\|_2 \leq \sqrt{\lambda_{\min}(\mathbf{\Gamma})} \|\mathbf{r}_K\|_2,$$

where $\lambda_{\min}(\mathbf{\Gamma})$ is the smallest eigenvalue of $\mathbf{\Gamma}$ and $\mathbf{r}_K = \boldsymbol{\epsilon} - \mathbf{\Gamma}\mathbf{x}_K$ is the residual vector at the k-th iteration of the conjugate gradient method and $\mathbf{x}_k = \mathbf{W}_K \mathbf{C}_K^{-1} \delta_1 \mathbf{e}_1$. The residual vector $\|\mathbf{r}_K\|_2$ can be bounded using standard techniques in Krylov subspace methods [100]. To use this as a stopping criterion, we note that $\|\mathbf{r}_K\|_2 = \delta_1 |\mathbf{e}_K^\top \mathbf{C}_K^{-1} \mathbf{e}_1|$ and by the Cauchy interlacing theorem $\lambda_{\min}(\mathbf{\Gamma}) \leq \lambda_{\min}(\mathbf{C}_K)$ [18, Corollary, III.1.3]. Combining the two bounds we have

$$\|\mathbf{\Gamma}^{-1/2}\boldsymbol{\epsilon} - \boldsymbol{\zeta}_K\|_2 \leq \sqrt{\lambda_{\min}(\mathbf{C}_K)} \delta_1 |\mathbf{e}_K^\top \mathbf{C}_K^{-1} \mathbf{e}_1|.$$

However, in numerical experiments we found that the bound was too pessimistic. Instead we adopted the approach in [28], we define the relative error norm as

$$e_K = \frac{\|\boldsymbol{\zeta}_K - \mathbf{\Gamma}^{-1/2}\boldsymbol{\epsilon}\|_2}{\|\mathbf{\Gamma}^{-1/2}\boldsymbol{\epsilon}\|_2}.$$

In practice, this quantity cannot be computed, but it can be estimated using successive iterates as

$$\tilde{e}_K = \frac{\|\boldsymbol{\zeta}_K - \boldsymbol{\zeta}_{k+1}\|_2}{\|\boldsymbol{\zeta}_{k+1}\|_2}.$$

When convergence is fast, we found this bound to be more representative of the true error in numerical experiments. The downside is that computing this is expensive since it costs $\mathcal{O}(nk^2)$ flops. However, this cost can be avoided by first writing

$$\boldsymbol{\zeta}_K = \mathbf{W}_K \widehat{\boldsymbol{\zeta}}_K, \quad \text{where} \quad \widehat{\boldsymbol{\zeta}}_K = \delta_1 \mathbf{C}_K^{-1/2} \mathbf{e}_1.$$

Since the columns of $\mathbf{W}_K$ are orthonormal, then

$$\tilde{e}_K = \frac{\|\widehat{\boldsymbol{\zeta}}'_K - \widehat{\boldsymbol{\zeta}}_{K+1}\|_2}{\|\widehat{\boldsymbol{\zeta}}_{K+1}\|_2} \qquad \boldsymbol{\zeta}'_K \equiv \begin{bmatrix} \widehat{\boldsymbol{\zeta}}_k \\ 0 \end{bmatrix}. \tag{3.4}$$

Therefore, $\tilde{e}_K$ can be computed in $\mathcal{O}(K^3)$ operations rather than $\mathcal{O}(nK^2)$ operations. A similar approach can be used to monitor the convergence of $\boldsymbol{\xi}_K$ to $\boldsymbol{\Gamma}^{1/2}\boldsymbol{\epsilon}$.

3.1.2 Preconditioned Lanczos Solvers

It is well known that an appropriate preconditioner can significantly accelerate convergence of Krylov subspace methods for solving linear systems. Assume that we have a preconditioner $\mathbf{G}$ which satisfies $\boldsymbol{\Gamma}^{-1} \approx \mathbf{G}^\top \mathbf{G}$. Then, the same preconditioner can be used to accelerate the convergence of Krylov subspace methods for generating samples, as we now show. Let

$$\mathbf{S}_{\boldsymbol{\Gamma}} = \mathbf{G}^{-1}(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top)^{1/2} \quad \text{and} \quad \mathbf{S}_{\boldsymbol{\Gamma}^{-1}} = \mathbf{G}^\top(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top)^{-1/2},$$

then it is easy to see that

$$\boldsymbol{\Gamma} = \mathbf{G}^{-1}(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top)\mathbf{G}^{-\top} = \mathbf{G}^{-1}(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top)^{1/2}(\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top)^{1/2}\mathbf{G}^{-\top} = \mathbf{S}_{\boldsymbol{\Gamma}}\mathbf{S}_{\boldsymbol{\Gamma}}^\top \tag{3.5}$$

and similarly $\boldsymbol{\Gamma}^{-1} = \mathbf{S}_{\boldsymbol{\Gamma}^{-1}}\mathbf{S}_{\boldsymbol{\Gamma}^{-1}}^\top$. The Lanczos process is then applied to $\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top$ and approximate samples from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$ and $\mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}^{-1})$ can be obtained by computing

$$\boldsymbol{\xi}_K = \mathbf{G}^{-1}\mathbf{W}_K\mathbf{C}_K^{1/2}\delta_1\mathbf{e}_1 \qquad \boldsymbol{\zeta}_K = \mathbf{G}^\top\mathbf{W}_K\mathbf{C}_K^{-1/2}\delta_1\mathbf{e}_1. \tag{3.6}$$

Algorithm 3.1.2 summarizes the preconditioned Lanczos sampling process. In practice, the vectors $\mathbf{w}_i$ tend to lose orthogonality in floating point arithmetic [28] and, therefore we reorthogonalize the vectors to alleviate potential loss of orthogonality.

Algorithm 3.1.2 Preconditioned Lanczos sampling

Output: $[\boldsymbol{\xi}, \boldsymbol{\zeta}] = \texttt{LanczosSampling}(\boldsymbol{\Gamma}, \mathbf{G}\ \boldsymbol{\epsilon})$

1: $\delta_0 = 1, \mathbf{w}_0 = \mathbf{0}, \delta_1 = \|\boldsymbol{\epsilon}\|_2, \mathbf{w}_1 = \boldsymbol{\epsilon}/\delta_1$
2: Set $i = 0$
3: **while** not converged **do**
4: Set $i \leftarrow i + 1$
5: $\gamma_i = \mathbf{w}_i^\top \mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top \mathbf{w}_i$,
6: $\mathbf{r} = \mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top \mathbf{w}_i - \gamma_i \mathbf{w}_i - \delta_{i-1}\mathbf{w}_{i-1}$
7: $\mathbf{w}_{i+1} = \mathbf{r}/\delta_i$, where $\delta_i = \|\mathbf{r}\|_2$
8: **end while**
9: Let the number of iterations be K. Set $\widehat{\boldsymbol{\xi}}_K = \mathbf{C}_K^{1/2}\delta_1 \mathbf{e}_1$, $\widehat{\boldsymbol{\zeta}}_K = \mathbf{C}_K^{-1/2}\delta_1 \mathbf{e}_1$ (see (3.1))
10: Set $\boldsymbol{\xi} = \mathbf{G}^{-1}\mathbf{W}_K\widehat{\boldsymbol{\xi}}_K$, $\boldsymbol{\zeta} = \mathbf{G}^\top \mathbf{W}_K \widehat{\boldsymbol{\zeta}}_K$.

If $\mathbf{G}$ is a good preconditioner, in the sense that $\boldsymbol{\Gamma}^{-1} \approx \mathbf{G}^\top\mathbf{G}$ (alternatively, $\mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^\top \approx \mathbf{I}$), then the Krylov subspace method is expected to converge rapidly. The choice of preconditioner depends on the specific problem. In [17], the authors develop a method for computing a sparse incomplete factorization of symmetric positive definite (SPD) matrices and use that factorization as the approximate inverse (AINV) preconditioner. In [16, 71] a stable method to compute the AINV preconditioner was developed independently. In [74] factorized sparse approximate inverse (FSAI) preconditioners were proposed for non-symmetric positive definite matrices. In [28] they discuss how stable AINV and FSAI preconditioners can be used with preconditioned Krylov subspace methods to sample from Gaussian distributions.

In this work, we use preconditioners of the form $\mathbf{G} = (-\boldsymbol{\Delta})^\gamma$ for parameters $\gamma \geq 0$, where $\boldsymbol{\Delta}$ is the Laplacian operator discretized using the finite-differences operator. This choice of preconditioners is inspired by [80], and exploits the fact that integral operators based on the Matérn kernels (see Section 2.1.1) have inverses which are fractional differential operators. Now that we have reviewed preconditioned Lanczos methods we will describe our proposed algorithms.

3.2 Method 1: Sampling Using Approximate Posterior Covariance

Consider generating samples from π_{post}, where $\mathbf{\Gamma}_{\text{post}} = \lambda^{-2}(\mathbf{Q}^{-1} + \lambda^2\mathbf{H})^{-1}$ is the posterior covariance matrix. Given a preconditioner $\mathbf{G}$, which we assume to be invertible, we can write

$$\mathbf{Q}^{-1} = \mathbf{G}^\top(\mathbf{G}\mathbf{Q}\mathbf{G}^\top)^{-1}\mathbf{G}.$$

Then consider the factorization $\mathbf{Q}^{-1} = \mathbf{L}\mathbf{L}^\top$ where

$$\mathbf{L} \equiv \mathbf{G}^\top(\mathbf{G}\mathbf{Q}\mathbf{G}^\top)^{-1/2}. \tag{3.7}$$

An important point to note is that while writing such a factorization, we do not propose to compute it explicitly. Instead, we access it in a matrix-free fashion using techniques from Algorithm 3.2.2.

Plugging the formula for $\mathbf{L}$ into the expression for the posterior covariance, we obtain

$$\mathbf{\Gamma}_{\text{post}} = \lambda^{-2}(\mathbf{L}\mathbf{L}^\top + \mathbf{H})^{-1} = \lambda^{-2}\mathbf{L}^{-\top}(\mathbf{I} + \mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-1})^{-\top}\mathbf{L}^{-\top}.$$

The low-rank approximation of $\mathbf{H}$ in (2.31) can be used to derive an approximate factorization of the posterior covariance matrix

$$\widehat{\mathbf{\Gamma}}_{\text{post}} = \widehat{\mathbf{S}}_{\mathbf{\Gamma}}\widehat{\mathbf{S}}_{\mathbf{\Gamma}}^\top \quad \text{where} \quad \widehat{\mathbf{S}}_{\mathbf{\Gamma}} \equiv \lambda^{-1}\mathbf{L}^{-\top}(\mathbf{I} + \mathbf{L}^{-1}\widehat{\mathbf{H}}\mathbf{L}^{-\top})^{-1/2}. \tag{3.8}$$

To efficiently compute mat-vecs with $\widehat{\mathbf{S}}_{\mathbf{\Gamma}}$, there are two stages: a precomputation stage and the sampling stage. In the precomputation stage, we compute the low-rank representation

$$\lambda^{-2}\mathbf{L}^{-1}\widehat{\mathbf{H}}\mathbf{L}^{-\top} = \lambda^{-2}\mathbf{L}^{-1}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top\mathbf{L}^{-\top} = \mathbf{Z}_k\mathbf{\Theta}_k\mathbf{Z}_k^\top,$$

where $\mathbf{Z}_k$ has orthonormal columns and $\mathbf{\Theta}_k$ is a diagonal matrix with non-negative entries. Computing the low-rank representation is accomplished using Algorithm 3.2.1 with $\mathbf{Y}_k = \mathbf{L}^{-1}\mathbf{V}_k\mathbf{M}_k$ where $\mathbf{M}_k$ is the lower Cholesky factorization of $\lambda^{-2}\mathbf{T}_k = \lambda^{-2}\mathbf{B}_k^\top\mathbf{B}_k$.

Now, we have

$$\widehat{\mathbf{S}}_{\mathbf{\Gamma}} \equiv \lambda^{-1}\mathbf{L}^{-\top}(\mathbf{I} + \mathbf{L}^{-1}\widehat{\mathbf{H}}\mathbf{L}^{-\top})^{-1/2} = \lambda^{-1}\mathbf{L}^{-\top}(\mathbf{I} - \mathbf{Z}_k\mathbf{D}_k\mathbf{Z}_k^\top).$$

Algorithm 3.2.1 Low-rank representation $\mathbf{Z}\boldsymbol{\Theta}\mathbf{Z}^\top = \mathbf{Y}\mathbf{Y}^\top$

Output: $[\mathbf{Z}, \boldsymbol{\Theta}] = \texttt{Lowrank}(\mathbf{Y})$ for an arbitrary $\mathbf{Y} \in \mathbb{R}^{n\times k}$ with $k \leq n$

1: Compute thin-QR factorization $\mathbf{QR} = \mathbf{Y}$
2: Compute eigenvalue decomposition $\mathbf{RR}^\top = \mathbf{U}\boldsymbol{\Theta}\mathbf{U}^\top$
3: Compute $\mathbf{Z} = \mathbf{QU}$

In this step, we have used a variation of the Woodbury identity [65, Equation (0.7.4.1)]

$$(\mathbf{I} + \mathbf{Z}_k\boldsymbol{\Theta}_k\mathbf{Z}_k^\top)^{-1/2} = \mathbf{I} - \mathbf{Z}_k\mathbf{D}_k\mathbf{Z}_k^\top \qquad \mathbf{D}_k = \mathbf{I}_k \pm (\mathbf{I}_k + \boldsymbol{\Theta}_k)^{-1/2}.$$

In the sampling stage, given $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we can compute a sample from $\widehat{\pi}_{\text{post}}$ as

$$\boldsymbol{\xi} = \mathbf{s}_k + \lambda^{-1}\mathbf{L}^{-\top}(\boldsymbol{\epsilon} - \mathbf{I} - \mathbf{Z}_k\mathbf{D}_k\mathbf{Z}_k^\top\boldsymbol{\epsilon}).$$

In summary, the procedure for computing samples $\boldsymbol{\xi}^{(j)} \sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Gamma}}_{\text{post}})$ is provided in 3.2.2. Computing mat-vecs with $\mathbf{L}$ (including its inverse and transpose) is done using the preconditioned Lanczos method described in 3.1.2. Note that

$$\mathbf{L}^{-1} = (\mathbf{GQG}^\top)^{1/2}\mathbf{G}^{-\top} \qquad \mathbf{L}^{-\top} = \mathbf{G}^{-1}(\mathbf{GQG}^\top)^{1/2}.$$

The accuracy of the generated samples is discussed in Section 3.4.

3.3 Method 2: Sampling Using the Full Posterior Covariance

The second approach we describe generates approximate samples from the exact posterior distribution. First, we rewrite the posterior covariance matrix as

$$\boldsymbol{\Gamma}_{\text{post}} = (\lambda^2\mathbf{Q}^{-1} + \mathbf{H})^{-1} = \mathbf{QF}^{-1}\mathbf{Q} \qquad \mathbf{F} \equiv \lambda^2\mathbf{Q} + \mathbf{QHQ}.$$

We define

$$\mathbf{S}_\mathbf{F} \equiv \mathbf{QF}^{-1/2}$$

such that $\boldsymbol{\Gamma}_{\text{post}} = \mathbf{S}_\mathbf{F}\mathbf{S}_\mathbf{F}^\top$. In this method, computing a factorization of $\boldsymbol{\Gamma}_{\text{post}}$ requires computing square roots with $\mathbf{F}$. Assume that we have a preconditioner $\mathbf{G}$ satisfying

Algorithm 3.2.2 Method 1: Generate N samples from $\widehat{\pi}_{\text{post}}$

Output: $[\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(N)}] = \texttt{Method1}(\mathbf{A}, \boldsymbol{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{G}, \mathbf{b}, N)$

Use genHyBR to get k, $\mathbf{s}_k, \lambda, \mathbf{V}_k, \mathbf{B}_k$ (see 2.2)
Compute Cholesky factorization $\mathbf{M}_k\mathbf{M}_k^\top = \lambda^{-2}\mathbf{B}_k^\top\mathbf{B}_k$
(**Stage 1: Precomputation stage**)
for $j = 1, \ldots, k$ **do**
  $\mathbf{z}^{(j)} = \mathbf{G}^{-\top}\mathbf{V}_k\mathbf{M}_k(:, j)$
  $[\boldsymbol{\xi}^{(j)}, \sim] = \texttt{LanczosSampling}(\mathbf{GQG}^\top, \mathbf{I}, \mathbf{z}^{(j)})$
  $\mathbf{Y}_k(:, j) = \boldsymbol{\xi}^{(j)}$ (Computes $\mathbf{Y}_k(:, j) = \mathbf{L}^{-1}\mathbf{V}_k\mathbf{M}_k(:, j)$)
end for
Compute $[\mathbf{Z}_k, \boldsymbol{\Theta}_k] = \texttt{Lowrank}(\mathbf{Y}_k)$
Compute $\mathbf{D}_k = \mathbf{I}_k \pm (\mathbf{I}_k + \boldsymbol{\Theta}_k)^{-1/2}$
(**Stage 2: Sampling stage**)
for $j = 1, \ldots, N$ **do**
  Draw sample $\boldsymbol{\epsilon}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Compute $\mathbf{z}^{(j)} = \boldsymbol{\epsilon}^{(j)} - \mathbf{Z}_k\mathbf{D}_k\mathbf{Z}_k^\top\boldsymbol{\epsilon}^{(j)}$
  $[\boldsymbol{\xi}^{(j)}, \sim] = \texttt{LanczosSampling}(\mathbf{Q}, \mathbf{G}, \mathbf{z}^{(j)})$ (Computes $\boldsymbol{\xi}^{(j)} = \lambda^{-1}\mathbf{L}^{-\top}\boldsymbol{\epsilon}^{(j)}$)
  Compute $\boldsymbol{\xi}^{(j)} \leftarrow \mathbf{s}_k + \boldsymbol{\xi}^{(j)}$
end for

$\mathbf{GG}^\top \approx \mathbf{F}^{-1}$. Armed with this preconditioner, we have the following factorization

$$\boldsymbol{\Gamma}_{\text{post}} = \mathbf{S}_\mathbf{F}\mathbf{S}_\mathbf{F}^\top \qquad \mathbf{S}_\mathbf{F} \equiv \mathbf{QG}^\top(\mathbf{GFG}^\top)^{-1/2}.$$

Note that this factorization of $\boldsymbol{\Gamma}_{\text{post}}$ is exact even though $\mathbf{GG}^\top \approx \mathbf{F}^{-1}$; see (3.5). The application of the matrix $\mathbf{G}^\top(\mathbf{GFG}^\top)^{-1/2}$ to a randomly drawn vector can be accomplished by the Lanczos approach described in Section 3.1.1.

Algorithm 3.3.1 Method 2: Sampling from π_{post}

Output: $[\boldsymbol{\xi}] = \texttt{Method2}(\mathbf{A}, \boldsymbol{\Gamma}_{\text{noise}}, \mathbf{Q}, \mathbf{G}, \mathbf{s}_{\text{post}})$

Draw sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Compute $\mathbf{F} = \lambda^2\mathbf{Q} + \mathbf{QHQ}$
Compute $[\mathbf{T},\mathbf{W}] = \texttt{Lanczos}(\mathbf{GFG}^\top, \boldsymbol{\epsilon})$, see Alg. 3.1.1
Compute $\mathbf{z} = \mathbf{G}^\top\mathbf{WT}^{-1/2}\delta_1\mathbf{e}_1$
Compute $\boldsymbol{\xi} = \mathbf{s}_{\text{post}} + \mathbf{Qz}$

As currently described, computing approximate samples from $\boldsymbol{\Gamma}_{\text{post}}$ requires computing $\mathbf{s}_{\text{post}}$ and applying the matrices $\mathbf{A}$, $\mathbf{A}^\top$ once, and $\mathbf{Q}$ twice. However, this may be compu-

tationally expensive for several problems of interest. Here we use $\mathbf{s}_k$ as an approximation to $\mathbf{s}_{\text{post}}$. A variant of this method, not considered in this work, follows by replacing the data-misfit part of the Hessian $\mathbf{H}$ by its low-rank approximation $\widehat{\mathbf{H}}$, defined in (2.31). Define

$$\widehat{\mathbf{F}} \equiv \lambda^2 \mathbf{Q} + \mathbf{Q}\widehat{\mathbf{H}}\mathbf{Q}.$$

Therefore, we compute the following factorization of the approximate posterior covariance

$$\widehat{\mathbf{\Gamma}}_{\text{post}} = \widehat{\mathbf{S}}_{\mathbf{F}}\widehat{\mathbf{S}}_{\mathbf{F}}^{\top} \qquad \widehat{\mathbf{S}}_{\mathbf{F}} \equiv \mathbf{Q}\mathbf{G}^{\top}(\mathbf{G}\widehat{\mathbf{F}}\mathbf{G}^{\top})^{-1/2}.$$

3.4 Comparing the Methods

We now compare the two proposed methods for generating approximate samples from the posterior. The first approach only uses the forward operator $\mathbf{A}$ in the precomputation phase to generate the low-rank approximation and subsequently uses the low-rank approximation as a surrogate. This can be computationally advantageous if the forward operator is very expensive to apply or if many samples are desired. On the other hand, if a greater degree of accuracy is important or only a few samples are needed, then the second approach is recommended since it targets the full posterior distribution. We summarize the computational costs of both methods in Table 3.1

Table 3.1 A summary of the main computational costs for Method 1 and Method 2. The number of genHyBR iterations is denoted k, and the number of iterations in the preconditioned Lanczos sampling algorithm is denoted K. The columns labeled $\mathbf{A}$ and $\mathbf{A}^{\top}$ contain the number of mat-vecs with the forward and the adjoint operator respectively; $\mathbf{Q}$ denotes the number of mat-vecs with $\mathbf{Q}$, $\mathbf{G}/\mathbf{G}^{\top}$ denotes the number of mat-vecs with the preconditioner, and $\mathbf{G}^{-1}/\mathbf{G}^{-\top}$ denotes the number of solves involving the preconditioner.

	Component	$\mathbf{A}$	$\mathbf{A}^{\top}$	$\mathbf{Q}$	$\mathbf{G}/\mathbf{G}^{\top}$	$\mathbf{G}^{-1}/\mathbf{G}^{-\top}$
Method 1	genHyBR	k	k	$2k$	$-$	$-$
	Precomputation	$-$	$-$	kK	$2kK$	K
	Sampling	$-$	$-$	$K+1$	$2K$	1
Method 2	Sampling	K	K	$2K+1$	$2K+1$	$-$

In Method 1, we generate samples from the approximate posterior distribution; the

following result quantifies the error in the samples. Define $\mathbf{S} = \mathbf{Q}^{1/2}(\lambda^2\mathbf{I} + \mathbf{H}_\mathbf{Q})^{-1/2}$ such that $\mathbf{\Gamma}_{\text{post}} = \mathbf{S}\mathbf{S}^\top$ and let $\boldsymbol{\epsilon}$ be a random draw from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, then

$$\mathbf{s} = \mathbf{s}_{\text{post}} + \mathbf{S}\boldsymbol{\epsilon} \quad \text{and} \quad \widehat{\mathbf{s}} = \mathbf{s}_k + \widehat{\mathbf{S}}\boldsymbol{\epsilon}$$

are samples from π_{post} and $\widehat{\pi}_{\text{post}}$ respectively, where $\widehat{\mathbf{S}}$ is defined in (3.8).

Theorem 3.4.1. *Let $\widehat{\mathbf{\Gamma}}_{\text{post}}$ be the approximate posterior covariance matrix generated by running k steps of the genHyBR algorithm and let $\boldsymbol{\epsilon}$ be a fixed sample. The error in the sample $\widehat{\mathbf{s}}$ satisfies*

$$\|\mathbf{s} - \widehat{\mathbf{s}}\|_{\lambda^2\mathbf{Q}^{-1}} \leq \lambda^{-1}\left(\frac{\omega_k\alpha_1\beta_1}{\lambda^2 + \omega_k} + \sqrt{\frac{\lambda^2\omega_k}{\lambda^2 + \omega_k}}\|\boldsymbol{\epsilon}\|_2\right).$$

Proof. By the triangle inequality, we have

$$\|\mathbf{s} - \widehat{\mathbf{s}}\|_{\lambda^2\mathbf{Q}^{-1}} \leq \|\mathbf{s}_{\text{post}} - \mathbf{s}_k\|_{\lambda^2\mathbf{Q}^{-1}} + \|\mathbf{S}\boldsymbol{\epsilon} - \widehat{\mathbf{S}}\boldsymbol{\epsilon}\|_{\lambda^2\mathbf{Q}^{-1}}.$$

Similar to previous proofs in Chapter 2, we use $\mathbf{s}_{\text{post}} - \mathbf{s}_k = (\mathbf{\Gamma}_{\text{post}} - \widehat{\mathbf{\Gamma}}_{\text{post}})\mathbf{A}^\top{\mathbf{\Gamma}_{\text{noise}}}^{-1}\mathbf{b} = \lambda^{-2}\mathbf{Q}^{1/2}\mathbf{D}\widehat{\mathbf{b}}$, where $\mathbf{D} = (\mathbf{I} + \lambda^{-2}\mathbf{H}_\mathbf{Q})^{-1} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1}$ to get

$$\|\mathbf{s}_{\text{post}} - \mathbf{s}_k\|^2_{\lambda^2\mathbf{Q}^{-1}} = \widehat{\mathbf{b}}^\top\mathbf{D}\mathbf{Q}^{1/2}\lambda^{-2}(\lambda^2\mathbf{Q}^{-1})\lambda^{-2}\mathbf{Q}^{1/2}\mathbf{D}\widehat{\mathbf{b}} = \lambda^{-2}\|\mathbf{D}\widehat{\mathbf{b}}\|_2^2.$$

Thus,

$$\|\mathbf{s}_{\text{post}} - \mathbf{s}_k\|_{\lambda^2\mathbf{Q}^{-1}} = \lambda^{-1}\|\mathbf{D}\widehat{\mathbf{b}}\|_2 \leq \lambda^{-1}\frac{\omega_k\alpha_1\beta_1}{\lambda^2 + \omega_k},$$

For the second term, we write

$$\lambda\mathbf{Q}^{-1/2}(\mathbf{S} - \widehat{\mathbf{S}})\boldsymbol{\epsilon} = \left[(\mathbf{I} + \lambda^{-2}\mathbf{H}_\mathbf{Q})^{-1/2} - (\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1/2}\right]\boldsymbol{\epsilon}.$$

Then,

$$\|\mathbf{S}\boldsymbol{\epsilon} - \widehat{\mathbf{S}}\boldsymbol{\epsilon}\|_{\lambda^2\mathbf{Q}^{-1}} = \|\lambda\mathbf{Q}^{-1/2}(\mathbf{S} - \widehat{\mathbf{S}})\boldsymbol{\epsilon}\|_2 \leq \|(\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1/2} - (\mathbf{I} + \lambda^{-2}\mathbf{H}_\mathbf{Q})^{-1/2}\|_2\|\boldsymbol{\epsilon}\|_2.$$

When we apply [18, Theorem X.1.1 and (X.2)], we have

$$\|(\mathbf{I} + \lambda^{-2}\widehat{\mathbf{H}}_\mathbf{Q})^{-1/2} - (\mathbf{I} + \lambda^{-2}\mathbf{H}_\mathbf{Q})^{-1/2}\|_2 \leq \|\mathbf{D}\|_2^{1/2}. \tag{3.9}$$

From (2.41), $\|\mathbf{D}\|_2 \leq \omega_k/(\lambda^2 + \omega_k)$. Plugging this into (3.9) gives the desired result. □

Theorem 3.4.1 states that if ω_k is sufficiently small, then the accuracy of the samples is high. The samples, thus generated, can then be used *as is* in applications. Otherwise they can be used as candidate draws from a proposal distribution $\widehat{\pi}_{\text{post}}$. To generate samples from the full posterior distribution, the approximated distribution can be used inside an independence sampler, similar to the approach in [21].

3.5 Numerical Results

In this section, we begin by discussing the preconditioners we choose and demonstrate their effectiveness. We then show the efficiency of the preconditioned sampling methods described in Sections 3.2 and 3.3 for generating samples from the approximate posterior and posterior distributions.

3.5.1 Choice of Preconditioners

In our first example, we explain the choice of preconditioners and show the performance of these preconditioners. We use preconditioners of the form $\mathbf{G} = (-\boldsymbol{\Delta})^{\gamma}$ for parameters $\gamma \geq 1$, where $\boldsymbol{\Delta}$ is the Laplacian operator discretized using the finite-differences operator, see Section 3.1.2.

In this experiment, we pick three different covariance matrices corresponding to Matérn parameters $\nu = 1/2$, $3/2$, and $5/2$; recall that this parameter controls the mean-squared differentiability of the underlying process. For a precise definition of the Matérn covariance function, see Section 2.1.1. We now briefly discuss the choice of the exponent γ. We choose $\gamma = 1/2$, 1, and 2 corresponding to $\nu = 1/2$, $3/2$, and $5/2$ respectively. The domain is taken to be $[0, 1]^2$, and we choose a 300×300 grid of evenly spaced points; thus, $\mathbf{Q}$ is a $90,000 \times 90,000$ matrix that is block-Toeplitz with Toeplitz blocks. Constructing such a matrix is never done explicitly; instead, circulant embedding and FFT-based techniques are used to efficiently perform mat-vecs. The correlation length ℓ is taken to be 0.25. In Figure 3.1, we provide the relative differences (computed as $\tilde{\mathrm{e}}_k$ from (3.4)) per iteration of the preconditioned and unpreconditioned Lanczos approach for sampling from $\mathcal{N}(\mathbf{0}, \mathbf{Q})$. We use a fixed sample $\boldsymbol{\epsilon}$ for all the values of $\hat{\mathrm{I}}_{\mathrm{j}}$ that we tested. It is readily seen that for $\nu = 1/2$ and $3/2$, including the preconditioner can dramatically speed up the convergence.

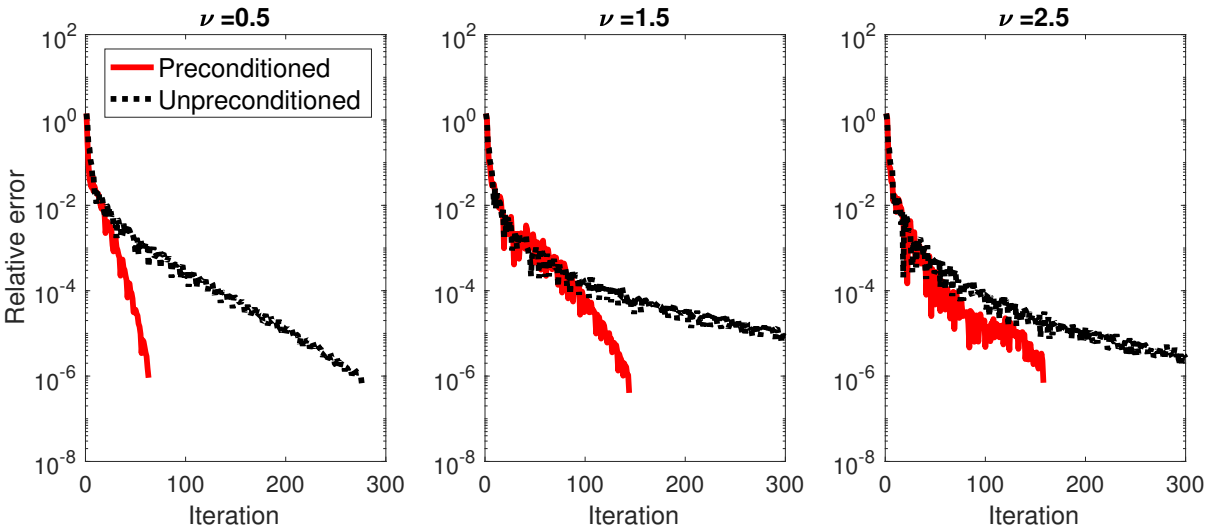


Figure 3.1 The relative differences $\tilde{e}_k$ using the Lanczos based sampling approach described in Section 3.1.1 applied to the prior covariance matrix $\mathbf{Q}$. The relative error plotted here is $\tilde{e}_k$ computed as (3.4). Preconditioners are based on fractional powers of the Laplacian $(-\boldsymbol{\Delta})^{\gamma}$. The plots correspond to various choices of ν in the Matérn covariance kernel and γ in the preconditioner. (left) $\nu = 1/2$ and $\gamma = 1/2$, (middle) $\nu = 3/2$ and $\gamma = 1$, and (right) $\nu = 5/2$ and $\gamma = 2$.

Some improvement is seen for the case of $\nu = 5/2$, but the unpreconditioned solver does not even converge within the maximum allotted number of iterations, which was set to 300. Also, observe that the number of iterations that it takes to converge increases with increasing parameter ν; this is because the systems become more and more ill-conditioned, with increasing ν, for a fixed grid size. In summary, we see that fractional powers of the Laplacian operator can be good preconditioners to use within the Lanczos methods described in Section 3.1.1 for sampling, provided ν is small. Next, we investigate the use of these preconditioners for sampling from the posterior and approximate posterior distributions.

3.5.2 Sampling from the Approximate Posterior Distribution

In this experiment, we choose two different test problems from the IRTools toolbox [49, 60]. Specifically, we choose the `PRspherical` which models spherical means tomography such as in photo-acoustic tomography (this is the same operator used in the Section 2.5), and `PRtomo` which models parallel X-ray tomography. For both applications, the true image $\mathbf{s}$ and forward model matrix $\mathbf{A}$ are provided. We use the default settings provided

by the toolbox; see [49] for details. To simulate measurement error, we add 2% additive Gaussian noise.

For the `PRspherical` problem, we first compute the MAP estimate for a grid size of 128×128 and for $\mathbf{Q}$ representing a Matérn kernel with $\nu = 1/2$ and $\ell = 0.25$. The reconstruction was computed using genHyBR and is provided in the left panel of Figure 3.2. The relative reconstruction error in the 2-norm was 0.0168, and the regularization parameter determined using WGCV was $\lambda^2 \approx 19.48$. The regularization parameter was fixed for the remainder of this experiment. In Figure 3.2, we also show a random draw from the prior distribution $\mathcal{N}(\mathbf{0}, \lambda^{-2}\mathbf{Q})$ in the middle panel and a random draw from the posterior distribution computed using Method 1 in Section 3.3 in the right panel. The same random vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ was used for both draws.

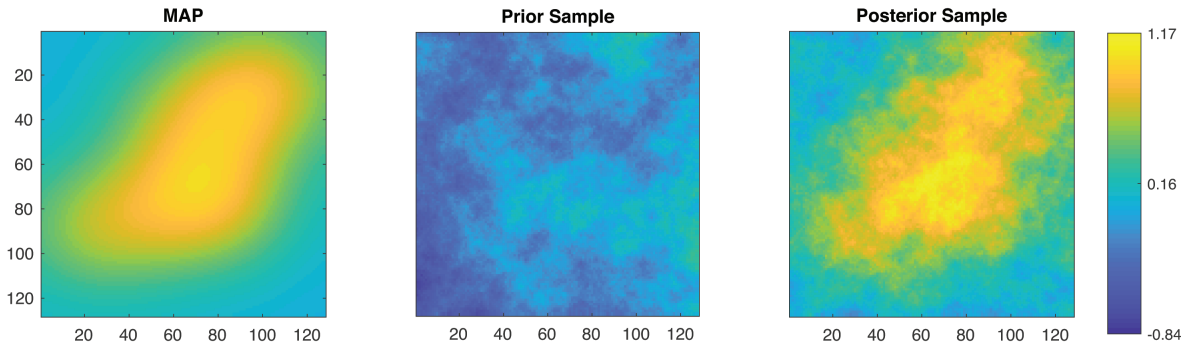


Figure 3.2 For the `PRspherical` problem, we provide the computed MAP estimate (left), a random draw from the prior distribution (middle), and a random draw from the posterior distribution computed using Method 1 (right).

In the next experiment, we investigate the performance of Method 1 described in Algorithm 3.2.2 for generating samples from the approximate posterior distribution $\widehat{\pi}_{\text{post}}$. The problem setup is the same as the previous experiment. We first use the genHyBR method to obtain the MAP estimate, the regularization parameter λ^2, and the low-rank approximation $\widehat{\mathbf{H}}_{\mathbf{Q}}$. In the third column of Table 3.2, we report the number of genHyBR iterations; see [33, 29] for details on stopping criteria. Then, we use Algorithm 3.2.2 with and without a preconditioner, described in Section 3.5.1, to generate samples. Notice that step 6 of Algorithm 3.2.2 requires the application of $\mathbf{L}^{-1}$ to the low-rank approximation;

this is accomplished by using the approach described in Section 3.1.1. The number of Lanczos iterations required for Step 6 is reported in the 'Precomputation' columns of Table 3.2. Then, for each sample, step 14 of Algorithm 3.2.2 requires the application of $\mathbf{L}^{-\top}$, which is also done using a Lanczos iterative process; the number of iterations for this step, averaged over 10 samples, is listed in the 'Sampling' columns of Table 3.2. We also provide the time in seconds it takes for Stage 1 and 10 runs of Stage 2 of Algorithm 3.2.2 with and without a preconditioner in the 'Time' columns of Table 3.2.

Table 3.2 We compare the performance of Method 1 described in Algorithm 3.2.2 with and without a preconditioner for the `PRspherical` and `PRtomo` applications. The first two columns contain the number of unknowns and the number of measurements. k is the number of genHyBR iterations required to compute the MAP estimate. The number of Lanczos iterations required to apply $\mathbf{L}^{-\top}$ (Step 6 of Algorithm 3.2.2) is reported under 'Precomp.', and the average number of iterations (averaged over 10 runs) to apply $\mathbf{L}^{-1}$ (Step 14 of Algorithm 3.2.2) is provided under 'Sampling'. The time in seconds it takes for to generate 10 samples is provided under 'Time'.

`PRspherical` application

# unknowns	# measurements	k	Preconditioner			No Preconditioner		
			Precomp.	Sampling	Time	Precomp.	Sampling	Time
16×16	368	52	761	14.7	0.49	1461	29.7	0.58
32×32	$1,440$	32	653	21.1	1.13	1481	51.0	2.50
64×64	$5,824$	27	745	30.1	3.58	2024	92.6	17.42
128×128	$23,168$	36	1378	42.5	37.82	4397	156.0	221.66
256×256	$92,672$	63	3321	60.8	417.83	12990	267.8	3246.38

`PRtomo` application

# unknowns	# measurements	k	Preconditioner			No Preconditioner		
			Precomp.	Sampling	Time	Precomp.	Sampling	Time
16×16	$4,140$	93	1335	14.0	0.56	2607	29.0	1.21
32×32	$8,100$	40	815	21.0	0.91	1872	51.4	3.41
64×64	$16,380$	32	890	29.5	3.51	2412	93.0	13.94
128×128	$32,580$	37	1432	42.1	27.34	4547	155.7	123.90
256×256	$65,160$	51	2645	61.0	201.31	10327	266.1	2294.45

We make a few remarks about the results. First, the precomputation step to generate the low-rank approximation requires a considerable number of mat-vecs involving $\mathbf{Q}$ but far fewer involving $\mathbf{A}$. Next, the number of iterations required for generating the samples is, on average, smaller than those reported in Table 3.3 for a comparable problem size. The reason for this is that the preconditioner is designed for $\mathbf{Q}$ rather than for $\mathbf{F}$ (as required by Method 2). Finally, the use of a preconditioner is effective in reducing the number of iterations and time it takes to generate samples.

3.5.3 Sampling from the Posterior Distribution

To begin, we repeat the first experiment of Section 3.5.2 with all the same parameters and instead use Method 2 to sample from the posterior distribution. In Figure 3.3, we show the computed MAP estimate (left), a random draw from the prior distribution (middle), and a random draw from the posterior distribution computed using Method 2 (right).

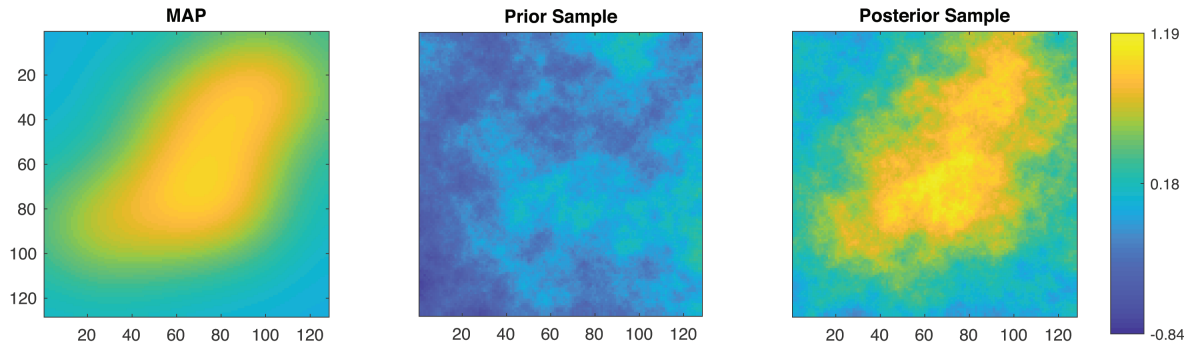


Figure 3.3 For the `PRspherical` problem, we provide the computed MAP estimate (left), a random draw from the prior distribution (middle), and a random draw from the posterior distribution computed using Method 2 (right).

For the next experiment we demonstrate the performance of the preconditioned Lanczos solver proposed in Section 3.3 on both applications. We vary the grid sizes from 16×16 to 256×256, and fix all other parameters ($\nu = 1/2$, 2% additive Gaussian noise) except the regularization parameter, which was determined separately for each problem using WGCV. The choice of preconditioners was described in Section 3.5.1. In Table 3.3

we report the number of iterations for the Lanczos solver to converge (i.e., achieving a residual tolerance of 10^{-6}) with and without a preconditioner.

Several observations can be made. First, for both applications the number of iterations required to achieve a desired tolerance increases with increasing problem size. This is to be expected since the number of measurements is also increasing with increasing problem size, and the iterative solver has to work harder to process the additional "information content." Second, in both applications, the use of a preconditioner cuts the number of iterations roughly in half. Since each iteration involves one forward and adjoint mat-vec involving $\mathbf{A}$, each iteration can be quite expensive; the use of a preconditioner is beneficial in this case. Finally, another important observation is that although the preconditioners proposed in Section 3.5.1 were designed for the prior covariance matrix $\mathbf{Q}$, here they were used for the matrix $\mathbf{F}$ instead; nevertheless, the results in Table 3.3 demonstrate that the preconditioners were similarly effective.

Table 3.3 For various examples from the `PRspherical` and `PRtomo` applications, we compare the performance of Method 2 described in Algorithm 3.3.1. We report the number of unknown parameters and measurements for each problem. Then, we provide the number of iterations (averaged over 10 different runs) required for convergence and the time it takes to generate 10 samples in the preconditioned and unpreconditioned cases.

`PRspherical` application

# unknowns	# measurements	Preconditioner		No preconditioner	
		Sampling	Time	Sampling	Time
16×16	368	20.0	0.12	40.5	0.27
32×32	$1,440$	28.0	0.65	67.5	1.70
64×64	$5,824$	38.9	2.49	118.0	12.11
128×128	$23,168$	53.7	21.31	206.3	135.60
256×256	$92,672$	72.7	153.78	359.2	1569.49

`PRtomo` application

# unknowns	# measurements	Preconditioner		No preconditioner	
		Sampling	Time	Sampling	Time
16×16	$4,140$	21.8	0.19	38.8	0.33
32×32	$8,100$	27.1	0.57	66.4	1.96
64×64	$16,380$	36.4	2.72	116.4	10.30
128×128	$32,580$	52.0	15.90	205.3	85.61
256×256	$65,160$	71.8	85.49	354.7	1146.69

3.6 Conclusion

In this chapter we present two variants that utilize a preconditioned Lanczos solver to efficiently generate samples from the posterior distribution. The first approach generates samples from an approximate posterior distribution, whereas the second approach generates samples from the exact posterior distribution. The approximate samples can be used *as is* or as candidate draws from a proposal distribution that closely approximates the exact posterior distribution.

For further research we could investigate the use of preconditioned Lanczos solvers in conjunction with the RTO method [13]. In doing so we could avoid sampling from the posterior (or approximated posterior) altogether and instead obtain a posterior sample by sampling only from the prior and noise distributions.

CHAPTER

4

EFFICIENT NEWTON-BASED APPROACHES TO SOLVE DETERMINISTIC QUANTITATIVE PHOTO-ACOUSTIC TOMOGRAPHY

We now turn attention to Quantitative Photo-acoustic Tomography, which can be represented as a nonlinear deterministic inverse problem. While the specific problem we focus on in this chapter is the Quantitative Photo-acoustic (QPAT) we will note that the techniques used here are applicable to a wide variety of other applications. In Chapter 1, we described the physical process that occurs during the QPAT process and how it relates to PAT, so we will omit discussion of it here.

In the past decade, the body of work on solving the QPAT inverse problem has expanded rapidly. Earlier works used fixed point iteration methods [36] to solve the inverse problem. This type of solution method was soon replaced by methods that incorporated

Jacobian and gradient information. Initially Jacobian based methods [48, 34, 114] were investigated. The major drawback of these methods were the computational cost and storage associated with the Jacobian. This lead groups to pursue gradient based solution methods [35, 10, 47, 81, 46]. The authors of [34, 114] explored Gauss Newton methods. Most authors used quasi-Newton methods [48, 35, 10, 47, 81, 46]. In a recent paper [67] the authors use inexact-Newton-CG methods on a linearized residual function.

Overview of Main Contributions

The main contributions of this chapter are as follows:

- We define a Newton-based approach for solving the QPAT problem. The resulting Newton step is computed inexactly using Krylov subspace methods.
- We investigate different regularization types and preconditioners for efficiently computing the Newton step.
- We demonstrate that the additional effort required to derive and implement the Newton approach is justified by the reduction in iterations and computational time.

In Section 4.1 we describe the forward and inverse problems. Then, in Section 4.2 we discuss the optimize-then-discretize approach taken with adjoint based Lagrange multipliers. In Sections 4.2.2 and 4.3.2 we describe and motivate the choice of regularization terms and propose a preconditioner to accelerate the computation of the Newton search direction.

4.1 Background

In this section we describe the equations for the forward and inverse QPAT problems. We refer the reader to Chapter 1 for background on the physical process that these partial differential equations (PDEs) are describing.

4.1.1 Forward Problem

As mentioned in the Chapter 1 the forward model for the QPAT can be modeled by the Diffusion Approximation (DA) of the Radiative Transfer Equation (RTE) [107]. In [107],

the authors discuss how the DA can be made depending on the type of source used. For a collimated source the forward problem can be described as

$$\begin{aligned} -\nabla \cdot (D\nabla u) + \mu u &= f, \quad x \in \Omega, \\ u + AD\nabla u \cdot \mathbf{n} &= 0, \quad x \in \partial\Omega, \end{aligned} \tag{4.1}$$

where

- Ω is the domain, either 2D or 3D,
- $\partial\Omega$ is the boundary of the domain,
- u is the photon density,
- f is a point source,
- $\mathbf{n}$ is the outward pointing unit vector,
- A is the interface constant, assumed to be known experimentally,
- $D > 0$ is the diffusion coefficient,
- $\mu > 0$ is the absorption coefficient.

We note that the coefficients D, mu, the source f and the photon density u are spatially dependent, but for ease of notation we do not write the dependence explicitly.

The diffusion coefficient, D is related to the reduced scattering coefficient, μ_s' by

$$D = \frac{1}{3\mu_s'},$$

where μ_s' us assumed to be a constant. The point source term takes the from

$$f = I\delta(x - \bar{x}),$$

where δ is the Dirac-delta function, $\bar{x}$ is the source location, and I is the photon density of the NIR light source. The point source is located inside the domain at a distance of $1/\mu_s'$ from the boundary [107]. It should be noted that reconstructions are not as accurate close (less than the mean free path) to the collimated source [107]. An alternative approach known as the diffuse source approach uses the point sources on the boundary as opposed to the interior of the domain, but we do not follow this approach here. In this chapter,

we will only consider the collimated source case. For ease of notation, in the subsequent sections we will drop the explicit spatial dependence.

4.1.2 Inverse Problem

We focus on reconstructing the absorption coefficient, μ, by solving an inverse problem and assume all other parameters are known. We make the assumptions that the diffusion coefficient is known and that our data comes from PAT image reconstruction process. Before we describe the inverse problem, we first make a change of variables in order to ensure that our reconstruction remains positive. We now define the absorption coefficient as

$$\mu = e^m, \tag{4.2}$$

and our goal is to find m. For the inverse problem we look at minimizing the data misfit between our reconstructed solution and the data. Here our data is the absorbed energy, ψ_i^{obs}, for each source $i = 1, \ldots, n_s$, where n_s is the total number of sources. It has been proven that, with some assumptions on the continuity of μ and D, at least two "well-chosen" sources are needed in order to allow unique and stable reconstructions [9, 122]. We make no assumptions on the continuity of the absorption parameter and in order to ensure a unique and stable reconstruction we incorporate a regularization term, which additionally allows us to incorporate some prior knowledge of the parameter into the inverse problem. This leads us to our cost function

$$J(m) = \min_m \phi(m) + R(m), \tag{4.3}$$

where $\phi(m)$ is the data-misfit term with form

$$\phi(m) = \sum_{i=1}^{n_s} \left[\int_\Omega (e^m u_i(m) - \psi_i^{obs})^2 \, dx \right],$$

and $R(m)$ is the regularization term and optimization problem is constrained by the PDEs for $i = 1, ..., n_s$

$$\begin{aligned} -\nabla \cdot (D\nabla u_i) + e^m u_i &= f_i, \quad x \in \Omega \\ u_i + AD\nabla u_i \cdot \mathbf{n} &= 0, \quad x \in \partial\Omega. \end{aligned}$$

The reconstruction of the log-absorption coefficient m is a nonlinear inverse problem which we tackle using Newton-based approaches.

4.2 Newton-Based Approaches

In order to solve the inverse problem, we take an optimize-the-discretize approach. We will first derive the first order optimality conditions and second order derivative information using variational derivatives in infinite dimensional space. We then discretize the problem and describe the numerical algorithm we use to solve the inverse problem and reconstruct the parameter of interest.

4.2.1 Adjoint-based Gradient and Hessian Computation

We will use the method of Lagrange multipliers to find the first and second order optimality conditions. For the rest of the chapter we will use the notation $\{u_i\}$ to mean $u_1, ..., u_{n_s}$ and similarly for $\{p_i\}$. We begin by taking the cost function (4.3) and the PDE (4.1), in weak form, and constructing the Lagrangian,

$$\begin{aligned}\mathcal{L}(\{u_i\},\{p_i\},m) = \sum_{i=1}^{n_s} \Bigg[&\int_\Omega (e^m u_i - \psi_i^{obs})^2 \, dx \, dx + \int_\Omega D\nabla p_i \cdot \nabla u_i \, dx \\ &+ \int_\Omega e^m p_i u_i \, dx - \int_\Omega p_i f_i \, dx + \int_{\partial\Omega} \frac{1}{A} p_i u_i \, ds \Bigg] + \int_\Omega R(m) dx,\end{aligned}$$

where $p_i \in H^1(\Omega)$ for $i = 1, ..., n_s$, are the Lagrange multipliers. Recall that $H^1(\Omega)$ is the space of square integrable functions with square integrable (weak) partial derivatives.

First Variations

We calculate the first variations of the Lagrangian for each parameter to find the first order optimality conditions: the state, adjoint, and gradient equations. For example the first variation of the Lagrangian, $\mathcal{L}(\{u_i\},\{p_i\},m)$ with respect to the parameter of interest m in the direction $\tilde{m}$, written as $\mathcal{L}_m(\{u_i\},\{p_i\},m)[\tilde{m}]$, will give us the gradient equation. In the Lagrangian approach we treat all of the parameters as independent and to construct the gradient, we must first find the state variables, u_i's and Lagrange multipliers, p_i's.

We find the state variables, u_i's by solving the state equation

$$\mathcal{L}_{p_i}(\{u_i\},\{p_i\},m)[\tilde{p}] = \int_\Omega D\nabla\tilde{p}\cdot\nabla u_i\ dx + \int_\Omega e^m\tilde{p}u_i\ dx \\ - \int_\Omega \tilde{p}f\ dx + \int_{\partial\Omega}\frac{1}{A}\tilde{p}u_i\ ds = 0,\ \ \forall\tilde{p}\in H^1(\Omega),$$

for each $i = 1, ..., n_s$ and the adjoint variables, p_i's by solving the adjoint equation

$$\mathcal{L}_{u_i}(\{u_i\},\{p_i\},m)[\tilde{u}] = \int_\Omega 2e^m\tilde{u}(e^m u_i - \psi_i^{obs})\ dx + \int_\Omega D\nabla p_i\cdot\nabla\tilde{u}\ dx \\ + \int_\Omega e^m p_i\tilde{u}\ dx + \int_{\partial\Omega}\frac{1}{A}p_i\tilde{u}\ ds = 0,\ \ \forall\tilde{u}\in H^1(\Omega),$$

for each $i = 1, ..., n_s$. The derivations for the state and adjoint equations are in Appendix B.1.1 and B.1.2. We can then use the state and adjoint variables to construct the gradient of the cost function (4.3)

$$g(m)[\tilde{m}] = \sum_{i=1}^{n_s}\left[\int_\Omega 2\tilde{m}e^m u_i(e^m u_i - \psi_i^{obs})\ dx + \int_\Omega \tilde{m}e^m p_i u_i\ dx\right] \tag{4.4}$$

$$+ \int_\Omega R'(m,\tilde{m})\ dx, \quad \forall\tilde{m}\in H^1(\Omega), \tag{4.5}$$

where $R'(m,\tilde{m})$ is the first variation of the regularization term w.r.t the parameter m. The derivation can be found in Appendix B.1.3.

Second Variations for One Source

The second variations of the Lagrangian are used to compute the action of the Hessian in a direction $\hat{m}$. For simplicity, we choose to illustrate the second variations for only one source and in Appendix B.1.4 we derive the second variations for multiple sources. In order to calculate the second variations, we introduce auxiliary variables $\hat{u}$ and $\hat{p}$, which are the incremental state and incremental adjoint variables and are solutions of the incremental state and incremental adjoint equations, respectively. In Appendix B.1.4, we derive all of the second variations (here the arguments are dropped for compactness) that

are then used to form the Newton system ~~a system of equations~~

$$\begin{aligned}
\mathcal{L}_{uu}(\tilde{u},\hat{u}) + \mathcal{L}_{um}(\tilde{u},\hat{m}) \quad + \mathcal{L}_{up}(\tilde{u},\hat{p}) \quad &= 0,\ \forall \tilde{u} \qquad \text{(Incremental Adjoint Eqn.)} \\
\mathcal{L}_{mu}(\tilde{m},\hat{u}) + \mathcal{L}_{mm}(\tilde{m},\hat{m}) + \mathcal{L}_{mp}(\tilde{m},\hat{p}) &= -g(m)[\tilde{m}],\ \forall \tilde{m} \qquad \text{(Hessian Apply)} \\
\mathcal{L}_{pu}(\tilde{p},\hat{u}) + \mathcal{L}_{pm}(\tilde{p},\hat{m}) \quad &= 0,\ \forall \tilde{p} \qquad \text{(Incremental State Eqn.)}
\end{aligned} \tag{4.6}$$

that can be used to describe the application (or action) of the Hessian in a given direction $\hat{m}, \hat{u}, \hat{p}$. Before we describe the discretization of the optimality conditions, we will describe $R(m)$, the two choices of regularization terms used in this chapter.

4.2.2 Regularization Type

We look at two different types of regularization terms, $R(m)$. The first type H^1-Tikhonov regularization [37], we will refer to as 'H1', combines minimizing the parameter of interest and minimizing the gradient of the the parameter:

$$R_{\text{H1}}(m) = \frac{\lambda}{2} \int_\Omega |m|^2 + |\nabla m|^2 \, dx.$$

We experimented with solving the inverse problem with only one of the two terms in the regularization term but found the combination to be better. The first and second variations of this regularization term are simply

$$R'_{\text{H1}}(m,\tilde{m}) = \lambda \int_\Omega m\tilde{m} + \nabla m \cdot \nabla \tilde{m} \, dx,$$

and

$$R''_{\text{H1}}(m,\tilde{m},\hat{m}) = \lambda \int_\Omega \hat{m}\tilde{m} + \nabla \hat{m} \cdot \nabla \tilde{m} \, dx.$$

This choice of regularization type is easy to implement, but a drawback is that it promotes smoothing of reconstructed parameter which, in many applications (e.g. QPAT) is not realistic.

The other regularization type we consider is a modified form of total variation (TV) regularization. TV regularization preserves edges and discontinuities in the reconstructed parameter but is highly nonlinear and non-differentiable when $\nabla m = 0$. The modified TV

regularization we use, which we will call 'TV' for the rest of the chapter, is defined as

$$R_{\mathrm{TV}}(m) = \frac{\lambda}{2}\int_{\Omega}(\nabla m \cdot \nabla m + \epsilon)^{1/2}\, dx,$$

where $\epsilon > 0$, is needed to unsure existence of the 1st and 2nd variations. The first variation is

$$R'_{\mathrm{TV}}(m, \tilde{m}) = \lambda \int_{\Omega} \frac{1}{(\nabla m \cdot \nabla m + \epsilon)^{1/2}} \nabla \tilde{m} \cdot \nabla m \, dx.$$

We will note that the second variation is

$$R''_{\mathrm{TV}}(m, \tilde{m}, \hat{m}) = \lambda \int_{\Omega} \frac{1}{(\nabla m \cdot \nabla m + \epsilon)^{1/2}} \left[\left(I - \frac{\nabla m \otimes \nabla m}{\nabla m \cdot \nabla m + \epsilon} \right) \nabla \tilde{m} \right] \cdot \nabla \hat{m} \, dx$$

The term $\left(I - \frac{\nabla m \otimes \nabla m}{\nabla m \cdot \nabla m + \varepsilon}\right)$ is highly nonlinear and will greatly affect the convergence of the Newton method we use to solve the inverse problem [125]. We make the choice to replace the second variation with

$$R''_{\mathrm{TV}}(m, \tilde{m}, \hat{m}) = \lambda \int_{\Omega} \frac{1}{(\nabla m \cdot \nabla m + \epsilon)^{1/2}} \nabla \tilde{m} \cdot \nabla \hat{m} \, dx.$$

This limits the Newton method to only first order convergence, but makes the method more robust [125].

Each of the regularization types contain a regularization parameter, λ, that balances how much influence the regularization term has on the cost function. For example, if $\lambda = 0$ we would strictly be minimizing the misfit between the data and reconstruction.

Choosing the Regularization Parameter

In Chapter 2.2.2, we discussed different methods that can be used to choose λ, the regularization (precision) parameter for a linear problem. Some of these techniques can be adapted for the nonlinear inverse problem setting. For QPAT we used the L-curve criterion to choose a regularization parameter. The L-curve criterion is used to find a balance between the misfit and regularization terms. This is done by plotting, on a log-log scale, the norm of the misfit versus the norm of the regularization term for various values of λ and the value that corresponds to the point of greatest curve is taken to be the regularization parameter. The reader is referred to [57] for more details. The method has no guarantee it will give a good regularization parameter but is still considered a good

heuristic [59].

4.2.3 Discretization

Now that we have described the optimality conditions, we can now consider our discretized problem. We discretize (discrete values are represented by bold font) using finite elements and represent our parameter of interest as a vector $\mathbf{m}$. The forward problem, which is the constraint of the optimization problem, can be written as

$$\mathbf{A}(\mathbf{m})\mathbf{u}_i(\mathbf{m}) = \mathbf{f}_i, \quad \text{for } i = 1, ..., n_s$$

where $\mathbf{A}(\mathbf{m})$ is the discretized forward problem for the given $\mathbf{m}$ and $\mathbf{f}_i$ is a discretized source. We can then write the data misfit term as

$$\phi(\mathbf{m}) = \sum_{i=1}^{n_s} \|\mathbf{E}(\mathbf{m})\mathbf{A}(\mathbf{m})^{-1}\mathbf{f}_i - \mathbf{d}_i\|^2_{\mathbf{M}},$$

where $\mathbf{E}(\mathbf{m})$ is a composition of the discretized representation of exponentiation of the parameter interest with the projection operator needed to correspond to the data space, $\mathbf{d}_i$ represents the discretized data, and $\mathbf{M}$ is a mass matrix. The form of the discretization of the adjoint and gradient equations is omitted since it will not provide meaningful insights to our problem.

We can discretize the Newton system(4.6) for only one source to get the following discretized system

$$\begin{bmatrix} \mathbf{L_{uu}} & \mathbf{L_{um}} & \mathbf{L_{up}} \\ \mathbf{L_{mu}} & \mathbf{R} + \mathbf{L_{mm}} & \mathbf{L_{mp}} \\ \mathbf{L_{pu}} & \mathbf{L_{pm}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{\mathbf{m}} \\ \hat{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} 0 \\ -\mathbf{g} \\ 0 \end{bmatrix}, \tag{4.7}$$

where the the discretized components of the matrix correspond to the system in the previous equations (4.6). In Appendix B.1.4 we describe the application of the Hessian for multiple sources. For reasons to be addressed later in the chapter we write the discretization of second variation w.r.t. m, $\mathcal{L}_{mm}(\tilde{m}, \hat{m})$ as $\mathbf{R} + \mathbf{L_{mm}}$, where $\mathbf{R}$ corresponds second variation of the regularization term and $\mathbf{L_{mm}}$ corresponds to the remaining terms. We reduce the system by eliminating the incremental state and adjoint variables, $\hat{\mathbf{u}}$ and

$\hat{\mathbf{p}}$, by using

$$\begin{aligned}\hat{\mathbf{u}} &= \mathbf{L}_{\mathbf{pu}}^{-1}\mathbf{L}_{\mathbf{pm}}\hat{\mathbf{m}} \\ \hat{\mathbf{p}} &= -\mathbf{L}_{\mathbf{up}}^{-1}(\mathbf{L}_{\mathbf{uu}}\hat{\mathbf{u}} + \mathbf{L}_{\mathbf{um}}\hat{\mathbf{m}}) \\ &= \mathbf{L}_{\mathbf{up}}^{-1}(\mathbf{L}_{\mathbf{uu}}\mathbf{L}_{\mathbf{pu}}^{-1}\mathbf{L}_{\mathbf{pm}} - \mathbf{L}_{\mathbf{um}})\hat{\mathbf{m}},\end{aligned}$$

to get the reduced linear system

$$\mathbf{H}\hat{\mathbf{m}} = -\mathbf{g},$$

where $\mathbf{g}$ is the discretized form of (4.4). The application of the Hessian to the direction $\hat{\mathbf{m}}$ can then be written as

$$\mathbf{H}\hat{\mathbf{m}} = \underbrace{(\mathbf{R} + \mathbf{L}_{\mathbf{mm}})}_{\text{Hessian of reg}}\hat{\mathbf{m}} + \underbrace{(\mathbf{L}_{\mathbf{mp}}\mathbf{L}_{\mathbf{up}}^{-1}(\mathbf{L}_{\mathbf{uu}}\mathbf{L}_{\mathbf{pu}}^{-1}\mathbf{L}_{\mathbf{pm}} - \mathbf{L}_{\mathbf{um}}) - \mathbf{L}_{\mathbf{mu}}\mathbf{L}_{\mathbf{pu}}^{-1}\mathbf{L}_{\mathbf{pm}}}_{\text{Hessian of data misfit}}\hat{\mathbf{m}}. \tag{4.8}$$

4.3 Inexact Newton-CG Method

Now that we have a gradient and a way to describe the application of the Hessian to a vector, we are able to use second order iterative methods, in particular Newton-CG (conjugate gradient) methods. Newton-CG methods are iterative methods used for optimization of nonlinear problems. They exhibit second order convergence and can be augmented with line search methods in order to obtain global convergence [83]. The core of Newton's method is that given a cost function $J(\mathbf{m})$ to minimize and an initial parameter guess $\mathbf{m}_0$ the solution can be found iteratively by

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\hat{\mathbf{m}}_k, \text{ where } \mathbf{H}_k\hat{\mathbf{m}}_k = -\mathbf{g}_k,$$

and for the k-th iteration $\mathbf{g}_k$ is the gradient of J, $\mathbf{H}_k$ is the Hessian of J, α_k is the step length, and $\hat{\mathbf{m}}_k$ is the step direction. We do not solve $\mathbf{H}_k\hat{\mathbf{m}}_k = -\mathbf{g}_k$ exactly, instead we solve

$$\mathbf{H}_k\hat{\mathbf{m}}_k + \mathbf{g}_k = \mathbf{r}_k,$$

where $\mathbf{r}_k$ is the residual. This is done because exact computation if $\hat{\mathbf{m}}_k$ can be expensive and we only need to find a descent direction that is an approximation of the actual descent direction. To solve this system, we use the conjugate gradient (CG) algorithm and we

refer the reader to [83] for more details.

A issue that can happen is that $\mathbf{H}_k$ may not be positive definite which would make the resulting $\hat{\mathbf{m}}_k$ not a descent direction. Typically, this occurs when the initial guess is "far" from the minimum. In order to account for this, we can use a Gauss-Newton approximation of the Hessian for a few of the initial iterations. In the Gauss-Newton approximation we drop the second derivative components $\mathbf{L_{mm}}$, $\mathbf{L_{um}}$, $\mathbf{L_{mu}}$ from (4.8) to get the Gauss-Newton Hessian apply

$$\mathbf{H}_k^{\mathrm{GN}}\hat{\mathbf{m}}_k = \mathbf{R}\hat{\mathbf{m}}_k + \underbrace{(\mathbf{L_{mp}}\mathbf{L_{up}}^{-1}\mathbf{L_{uu}}\mathbf{L_{pu}}^{-1}\mathbf{L_{pm}})}_{\text{Hessian of Misfit}}\hat{\mathbf{m}}_k \tag{4.9}$$

The Gauss-Newton Hessian, assuming $\mathbf{R}$ is positive definite, is always positive semi-definite which guarantees that $\hat{\mathbf{m}}_\mathbf{k}$ will be a descent direction. Additionally, we check the negative curvature condition to ensure $\hat{\mathbf{m}}_k$ is a descent direction. If negative curvature is detected on the first iteration we set $\hat{\mathbf{m}}_k = \mathbf{g}_k$, if negative curvature is found at later iterations we terminate the CG iterations and use the last iterate as the step direction [83].

Another factor we consider is choosing an appropriate step length α_k by using a backtracking linesearch method. A poorly chosen α_k may result in completely overstepping the minimum or not obtaining a sufficient decrease resulting in slow convergence. We use the Armijo condition [83]

$$J(\mathbf{m}_k + \alpha_k\hat{\mathbf{m}}_k) \leq J(\mathbf{m}_k) + c\alpha_k\mathbf{g}_k^\top\hat{\mathbf{m}}_k, \tag{4.10}$$

where $0 < c < 1$, to find a step length that will result in sufficient decrease. We will note that the Armijo condition is one of the Wolfe conditions [83]. For this problem we chose not to use the Wolfe or strong Wolfe conditions because that would potentially involve multiple additional computations of the gradient. In Table 4.1 we provide the number of PDE solves needed for one iteration of the inexact-Newton-CG algorithm.

4.3.1 Stopping Criteria

We discuss the stopping criteria for the Newton and Conjugate gradient iterations.

Table 4.1 In this table we provide the number of forward, adjoint, incremental state, and incremental adjoint PDE solves needed for one iteration of the inexact-Newton-CG algorithm.

	Forward	Adjoint	Incremental State	Incremental Adjoint
Initialization	n_s	0	0	0
Gradient	n_s	n_s	0	0
Hessian	0	0	n_s	n_s
Linesearch	n_s	0	0	0
Total	$3n_s$	n_s	n_s	n_s

Newton/GN

We explore two different stopping criteria for the outer Newton/GN iterations. The first stopping criteria, which we will refer to as 'gradient', will stop the iterations when the norm of the gradient is less than or equal to some tolerance times the norm of the initial gradient:

$$\|\mathbf{g}_k\|_{\mathbf{M}} \leq \varepsilon \|\mathbf{g}_0\|_{\mathbf{M}},$$

where $\|\cdot\|_{\mathbf{M}}$ is the 2-norm weighted by the mass matrix $\mathbf{M}$.

CG

For the CG iterations we also use two different stopping criteria. The first is simply uses the relative residual

$$\|\mathbf{H}_k\mathbf{m}_k + \mathbf{g}_k\|_2 \leq c_{rtol}\|\mathbf{r}_k\|_2.$$

The other stopping criteria use, is the 'inexact' stopping criteria, which terminates the iterations when

$$\|\mathbf{H}_k\mathbf{m}_k + \mathbf{g}_k\|_2 \leq \min\left(0.5, \frac{\|\mathbf{g}_k\|_2}{\|\mathbf{g}_0\|_2}\right).$$

4.3.2 Preconditioner

Each CG iteration requires one incremental forward and incremental adjoint solve per source and can be computationally expensive. To mitigate this, we explore the use of a preconditioner in order to reduce number of CG iterations. It is well known that the number of CG iterations depend on the condition number and the clustering of eigenvalues; the more cluster the faster the convergence. Following the authors of [127], we choose the preconditioner to be the discretized regularization term $\mathbf{R}$. If we assume that $\mathbf{R}$ is

invertible then we can factor $\mathbf{R^{-1}} = \mathbf{L}\mathbf{L}^\top$. Using this factorization, we can consider the Gauss-Newton Hessian and rewrite it as

$$\mathbf{H}_{\text{GN}} = \mathbf{H}_{\text{misfit}} + \mathbf{R} = \mathbf{L}^{-\top}(\mathbf{L}^\top \mathbf{H}_{\text{misfit}} \mathbf{L} + \mathbf{I})\mathbf{L}^{-1}, \tag{4.11}$$

where $\mathbf{H}$ is the Hessian of the misfit from (4.9) If we chose $\mathbf{R}$ to be our preconditioner for our reduced linear system and use (4.11) we have that

$$\mathbf{R}^{-1}\mathbf{H}\hat{\mathbf{m}} = -\mathbf{R}^{-1}\mathbf{g} \implies (\mathbf{L}^\top \mathbf{H}_{\text{misfit}} \mathbf{L} + \mathbf{I})\mathbf{L}^{-1}\hat{\mathbf{m}} = -\mathbf{L}^\top \mathbf{g}.$$

This means that the CG iterations will depend on the spectrum of $(\mathbf{L}^\top \mathbf{H}_{\text{misfit}} \mathbf{L} + \mathbf{I})$ and since $\mathbf{H}_{\text{misfit}}$ is often a low rank operator we would expect more clustering of the eigenvalues. The condition number of $(\mathbf{L}\mathbf{H}_{\text{misfit}}\mathbf{L}^\top + \mathbf{I}) = (\widetilde{\mathbf{H}} + \mathbf{I})$ is

$$\kappa(\widetilde{\mathbf{H}} + \mathbf{I}) = \frac{\lambda_1(\widetilde{\mathbf{H}}) + 1}{\lambda_N(\widetilde{\mathbf{H}}) + 1}.$$

This is well-conditioned if $\widetilde{\mathbf{H}}$ is a "small" perturbation of the identity matrix.

If $\mathbf{R}$ is too expensive to factorize or invert we consider an approximate factorization $\mathbf{R}^{-1} \approx \widetilde{\mathbf{R}}^{-1} = \widetilde{\mathbf{L}}^T\widetilde{\mathbf{L}}$, then by Weyl's Theorem [65, Theorem 4.3.1] we have

$$\kappa\big(\underbrace{\widetilde{\mathbf{L}}\mathbf{H}_{\text{misfit}}\widetilde{\mathbf{L}}^\top}_{\mathbf{H}'_{\text{misfit}}} + \underbrace{\widetilde{\mathbf{L}}\mathbf{R}\widetilde{\mathbf{L}}^\top}_{\mathbf{R}'}\big) \leq \frac{\lambda_1(\mathbf{H}'_{\text{misfit}}) + \overbrace{\lambda_1(\mathbf{R}')}^{\approx 1}}{\lambda_N(\mathbf{H}'_{\text{misfit}}) + \underbrace{\lambda_N(\mathbf{R}')}_{\approx 1}}.$$

This is also well-conditioned if $\mathbf{H}'_{\text{misfit}}$ is a "small" perturbation of the identity matrix.

If $\mathbf{R}$ is not invertible a mass matrix can be added, so the preconditioner can be defined as

$$\widehat{\mathbf{R}} = \mathbf{R} + \gamma\mathbf{M},$$

where $\gamma > 0$ is small and $\mathbf{M}$ is a mass matrix in the parameter space. Figure 4.1 shows the normalized spectrum of the Gauss-Newton Hessian matrix and we see clustering of the eigenvalues when the preconditioner is used. One of the properties of CG methods is that when there is clustering of the eigenvalues the convergence rate is faster. In Section 4.4.1, we look at the effect of this preconditioner.

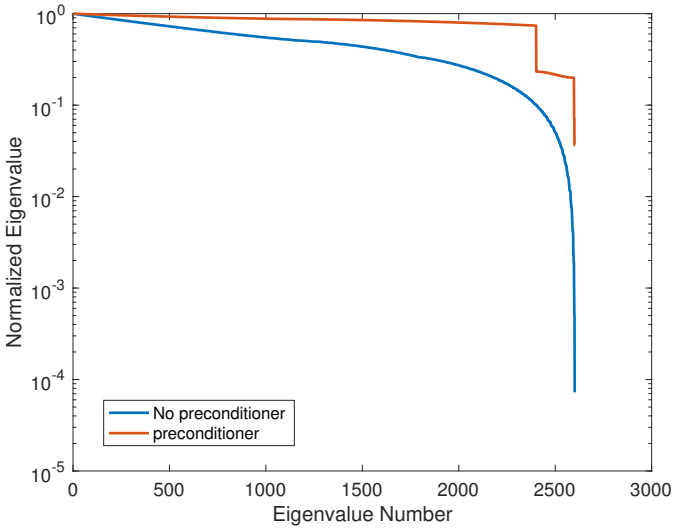


Figure 4.1 Normalized spectrum of the Gauss-Newton Hessian with and without preconditioning.

4.4 Numerical Results

In order to solve the inverse problem we rely on a software called FEniCS [41, 2] to implement the PDEs. FEniCS allows users to input the variational form of the PDE they are trying to solve and specify a mesh they wish to solve the PDE on, then it is able to discretize and solve the PDE using finite element methods. The code used to solve the inverse problem is adapted from hIPPYlib [127, 123, 124, 126] and uses PETSc [11] for linear algebra operations and solvers. All of the experiments were run on a cluster powered by the Slurm batch engine running on Ubuntu 16.04. Nodes on the cluster are powered by two Intel(R) Xeon(R) CPU E5-2690 8-core CPUs @ 2.90GHz with 128GB of DDR3 RAM, only 1 core was used in all of the experiments.

For all of the experiments, to avoid an inverse crime, we first generate synthetic data obtained by solving the forward problem with the true parameter on a finer mesh of size 401×401. We then project the resulting field on to a coarser grid of size 101×101 and add 2% Gaussian noise to simulate measurement error. The mesh elements are from the Lagrange family of degree 1. We assumed all the other parameters described in Section 4.1.1 were constant with the following values taken from [107]: interface constant, $A = 2.74$, the diffusion coefficient, $D = 1/6.0075$, and $\mu_s' = 2.0025$. In all the experiments the domain is $[0, 1] \times [0, 1]$ and the initial parameter guess was a constant value of $\ln 0.02 \approx -3.9$.

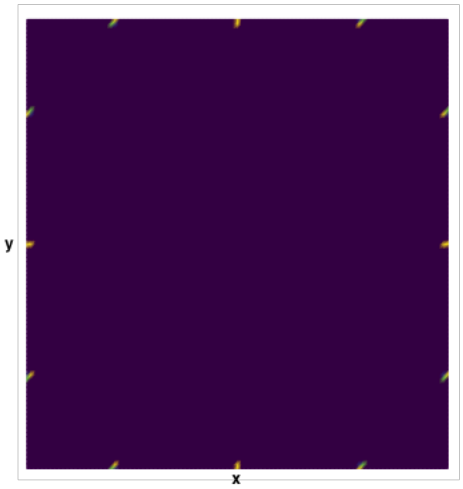


Figure 4.2 Plot of the location of the sources which are indicated in yellow.

4.4.1 Method Type

For our first experiment, we compared different optimization methods that can be used to solve the QPAT inverse problem. We used the 'H1' regularization type described in Section 4.2.2 and we considered a Gauss-Newton method with and without preconditioning, a L-BFGS method, and an inexact Newton method. For this experiment there are 12 sources, 3 on each side located an equal distance apart, see Figure 4.2.

In Table 4.2 we report the number of iterations, relative error of the computed solution compared with the true solution, and the time it took for the method to converge. The Gauss-Newton methods used performed the worse in terms of number of iterations and time. We were able to demonstrate that the preconditioner described in Section 4.3.2 was effective in reducing the number of CG iterations and in fact reduces the iterations more than tenfold. The L-BFGS method used for this experiment was the default method provided in Python's `scipy.optimize` package [68, 128]. While all of the methods achieved the same order of relative error it is clear that the Newton method without preconditioning performed the best in terms of computational time. The Gauss-Newton approach has a lower cost per iteration compared to the Newton approach; however, the Newton approach takes fewer iterations and hence converges faster. This justifies the additional effort in deriving and implementing the Newton approach. In Figure 4.3, we show the reconstructed solutions for the methods in Table 4.2. They are visually identical and capture the features of the true image well. We can see that due to the choice of regularization type the edges

of the reconstructed features are not sharp.

In Table 4.3 we show the effects of our preconditioner for the Newton method for the relative residual CG stopping criteria. Here it is clear that our preconditioner increases the number of CG iterations and thus the computational cost, the opposite of our goal. This is due to the fact that the Newton Hessian has additional terms that are not captured by the preconditioner we use. This clearly dictates the need to design a preconditioner based on the Newton Hessian in the future.

Table 4.2 Here we compare methods for the 'H1' regularization type using gradient and inexact stopping criteria. We see that using a preconditioner drastically decreases the number of CG iterations for the Gauss-Newton method.

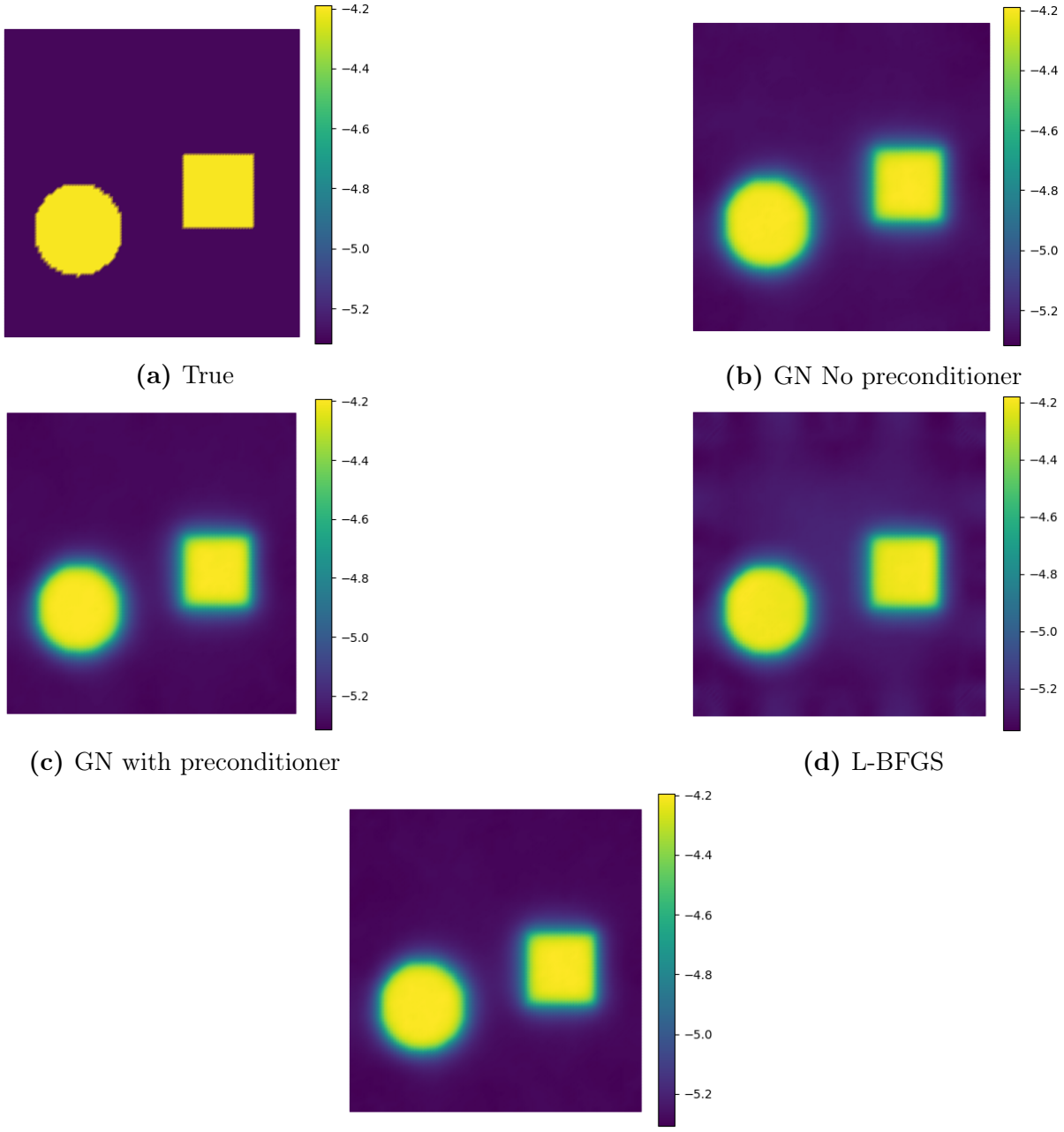
Meth.	Precond.	Outer Iter.	CG Iter.	Rel. Err.	Time(s)
L-BFGS	N/A	8	N/A	1.746E-02	856.60
GN	No	57	5166	2.472E-02	1770.13
GN	Yes	64	369	2.449E-02	1503.17
Newt	No	41	113	2.404E-02	628.60

Table 4.3 Here we compare the Newton method for 'H1' regularization type using gradient and relative residual stopping criteria. We can see that the preconditioner from Section 4.3.2 actually increases the number of CG iterations.

Meth.	Precond.	Outer Iter.	CG Iter.	Rel. Err.	Time(s)
Newt	Yes	37	9612	2.411E-02	1568.83
Newt	No	37	1319	2.409E-02	694.66

4.4.2 Regularization Type

In this experiment, we compare the two choices of regularization types described in Section 4.2.2. We use the Newton method with the gradient stopping criteria for the Newton iterations and the inexact stopping criteria for the CG iterations and report the results of the iteration count, relative error and time in Table 4.4. We can see that even with using the approximation of the TV Hessian the method converges in a comparable number of



(a) True

(b) GN No preconditioner

(c) GN with preconditioner

(d) L-BFGS

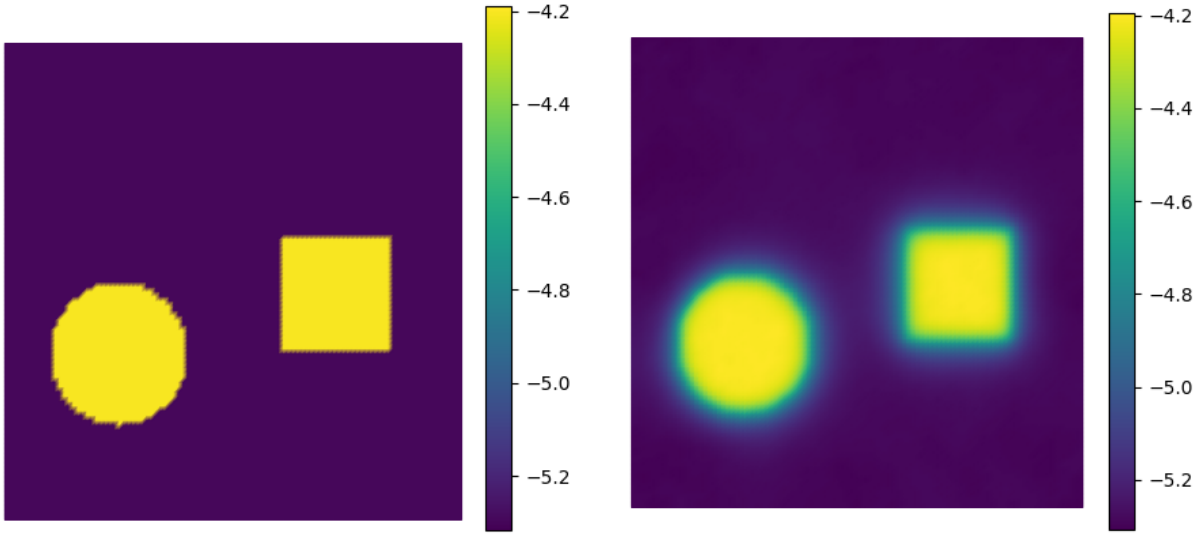
(e) Newton

Figure 4.3 The reconstructions of the QPAT image found using the different methods, the relative error for the reconstructions are provided in Table 4.2.

iterations and computational time. The relative error for both regularization types are similar, but in Figure 4.4 it is easily seen that the TV regularization preserves the edges of parameter better.

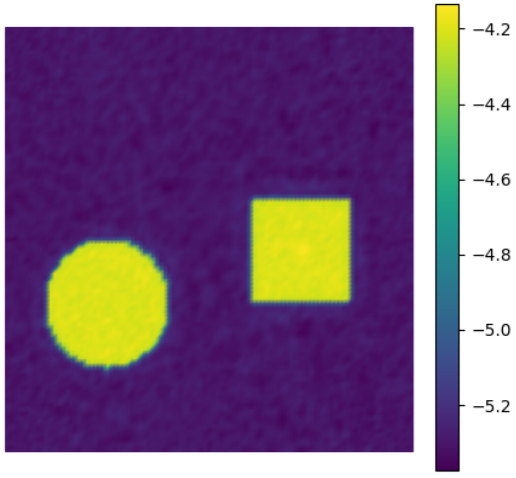
Table 4.4 Here we compare the 'TV' and 'H1' regularization types for the Newton method with the gradient stopping criteria for the Newton iterations and the inexact stopping criteria for the CG iterations.

Meth.	Reg.	Newt Iter.	CG Iter.	Rel. Err.	Time(s)
Newt	TV	10	237	8.034E-03	171.37
Newt	H1	41	113	2.404E-02	628.60



(a) True image

(b) 'H1'



(c) 'TV'

Figure 4.4 Reconstructions using different regularization types; the relative error is provided in Table 4.4.

4.4.3 Increasing the Number of Sources

For this experiment, we look at how increasing the number of sources effects the number of iterations and time it takes the method to converge. In Table 4.5, we recorded the results of increasing the number of sources for the Newton method using the 'H1' regularization type with the gradient and relative residual stopping criteria. The table shows that increasing the number of sources corresponds to an increase on the number of iterations and an increase in the time it takes the method to find a solution. The relative errors of the solutions are all the same order. This suggests that increasing the number of sources increases the cost but does not improve the reconstruction accuracy.

Table 4.5 Here we can see the effects of increasing the number of sources. We use the 'H1' Newton method without preconditioning for the gradient and relative residual stopping criteria

# of sources	Newt Iter.	CG Iter.	Rel. Err.	Time(s)
12	41	113	2.404E-02	628.60
20	54	1304	2.299E-02	1551.24
40	95	2297	2.263E-02	5466.65
60	229	4824	2.236E-02	22744.03
84	187	3778	2.226E-02	22597.89

4.5 Conclusion

In this chapter, we presented an efficient algorithm for the deterministic solution to the QPAT inverse problem. We derived the first and second order optimality conditions for the problem and demonstrated how the discretized conditions can be used in a Newton algorithm. We compared the LBFGS, Gauss-Newton, and Newton approach and demonstrated that our method was capable of achieving the same order of accuracy in a faster amount of time. Additionally, we compared two different regularization types and showed that they had comparable results.

For future work there is a need to derive better preconditioners for the Newton approach. We demonstrated that using additional sources does not significantly increase

the accuracy of the reconstruction but increases the computational cost. This suggests that incorporating randomization, in a manner similar to the authors of [56], could reduce the cost associated with multiple sources. Another future direction is to adopt a Bayesian approach for QPAT and efficiently solve for the MAP estimate. We could then linearize the QPAT problem and use the prior preconditioned Hessian to sample from the posterior distribution using methods described in Chapter 3.

CHAPTER

5

CONCLUSIONS

In this thesis, we developed efficient methods for solving inverse problems and quantifying the uncertainty in large-scale inverse problems arising in imaging applications. The algorithms that we developed were validated on synthetic problems from a model application, Photo-acoustic tomography (PAT), which is an image reconstruction inverse problem with great potential in biomedical imaging.

In the first part of the thesis, Chapters 2 and 3, we focused on uncertainty quantification for linear inverse problems with Gaussian posterior distributions. This is a good model for the first part of the PAT reconstruction process. In Chapter 2, we developed error bounds for monitoring the accuracy of the approximate posterior covariance matrix based on the generalized hybrid Golub-Kahan (genHyBR) [33] iterates. We derived and showed how to efficiently compute bounds for the Kullback-Leibler divergence between the posterior and the approximate posterior, and between the posterior and prior distributions. Additionally, we developed and demonstrated error bounds for the accuracy of approximated optimality criteria. We demonstrated effectiveness of the bounds on a smaller toy problem and we used of the genHyBR method to solve the linear Bayesian inverse problem associated

with PAT. In Chapter 3, we used preconditioned Krylov subspace methods to generate samples from the posterior distribution of linear Bayesian inverse problems with Gaussian posteriors. We developed two sampling algorithms based on preconditioned Lanczos methods to generate samples from the posterior distribution. The first algorithm uses a low-rank approximation of the posterior covariance constructed by the genHyBR algorithm to generate samples from the approximate posterior distribution. Additionally, we derived error bounds for the accuracy of a sample generated from this approximate posterior covariance. The second algorithm generates approximate samples from the posterior distribution. For both preconditioned sampling methods we demonstrate the efficiency of the algorithm on the PAT model problem. In the second part of the thesis, Chapter 4, we developed Gauss-Newton and Newton solver for the deterministic non-linear Quantitative photo-acoustic tomography (QPAT) inverse problem which is the second part of the PAT reconstruction process. We studied certain aspects of solver: the the regularization type, choice of preconditioner, choice of stopping criteria, and behavior of the solver as the number of sources increased. We demonstrated the performance of our solvers on a synthetic model problem in QPAT.

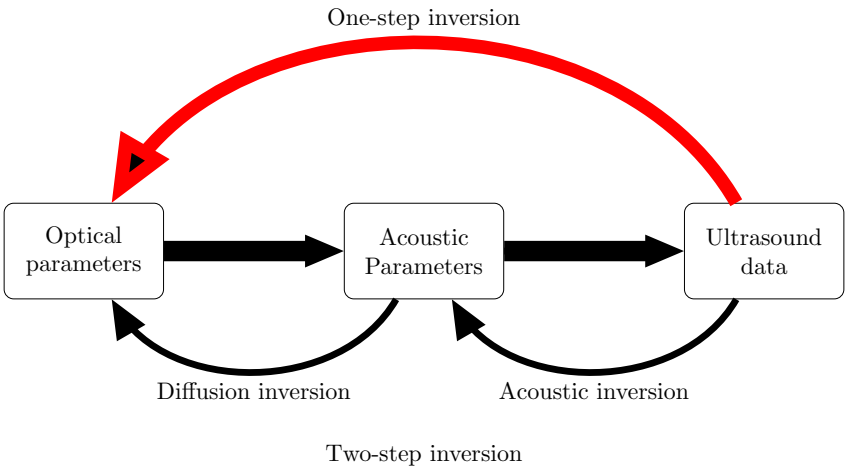


Figure 5.1 A flowchart that compares the one-step and two-step PAT inverse problem.

Future work can take several different directions. While this thesis adopted a two-step approach, an alternative approach is to consider a one-step method for solving the PAT

inverse problem. That is, the goal is to directly estimate the optical parameters from the measured data. While the one-step approach has many potential benefits from a modeling standpoint, it is computationally challenging since we now have to solve a coupled inverse problem. Figure 5.1 is a schematic that explains the idea one-step and two-step inversion process entail. There have been some advances in developing a practical one-step method [40, 122, 94, 67]. One possibility is to extend the Newton-based solvers in Chapter 4 to the one-step case. Another possibility is extending the UQ methods developed in Chapters 2 and 3 by using a Laplace's approximation to the non-Gaussian posterior distribution.

BIBLIOGRAPHY

[1] A. Alexanderian and A. K. Saibaba. "Efficient D-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems". In: *SIAM Journal on Scientific Computing* 40.5 (2018).

[2] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. "The FEniCS Project version 1.5". In: *Archive of Numerical Software* 3.100 (2015).

[3] S. Ambikasaran, J. Li, P. Kitanidis, and E. Darve. "Large-scale stochastic linear inversion using Hierarchical matrices". In: *Computational Geosciences* (2012).

[4] M. A. Anastasio, Jin Zhang, Xiaochuan Pan, Yu Zou, Geng Ku, and L. V. Wang. "Half-time image reconstruction in thermoacoustic tomography". In: *IEEE Transactions on Medical Imaging* 24.2 (2005), pp. 199–210.

[5] M. A. Anastasio, J. Zhang, D. Modgil, and P. J. L. Rivière. "Application of inverse source concepts to photoacoustic tomography". In: *Inverse Problems* 23.6 (2007), S21–S35.

[6] M. Arioli. "Generalized Golub-Kahan bidiagonalization and stopping criteria". In: *SIAM Journal on Matrix Analysis and Applications* 34.2 (2013), pp. 571–592.

[7] M. Arioli and D. Orban. "Iterative methods for symmetric quasi-definite linear systems - Part I: Theory". In: *Cahier du GERAD G-2013-32, GERAD, Montréal, QC, Canada* (2013).

[8] A. Atkinson, A. Donev, and R. Tobias. *Optimum Experimental Designs, with SAS*. 1st ed. Oxford University Press USA - OSO, 2007.

[9] G. Bal and K. Ren. "Multi-source quantitative photoacoustic tomography in a diffusive regime". In: *Inverse Problems* 27.7 (2011).

[10] G. Bal and K. Ren. "On multi-spectral quantitative photoacoustic tomography in diffusive regime". In: *Inverse Problems* 28.2 (2012).

[11] S. Balay et al. *PETSc Web page*. `https://www.mcs.anl.gov/petsc`. 2019.

[12] J. M. Bardsley, A. Seppänen, A. Solonen, H. Haario, and J. Kaipio. "Randomize-then-optimize for sampling and uncertainty quantification in electrical impedance tomography". In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 1136–1158.

[13] J. M. Bardsley, A. Solonen, H. Haario, and M. Laine. "Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems". In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1895–A1910.

[14] F. Bazán and L. Borges. "GKB-FP: an algorithm for large-scale discrete ill-posed problems". In: *BIT Numerical Mathematics* 50.3 (2010), pp. 481–507.

[15] S. J. Benbow. "Solving generalized least-squares problems with LSQR". In: *SIAM Journal on Matrix Analysis and Applications* 21.1 (1999), pp. 166–177.

[16] M. Benzi, J. K. Cullum, and M. Tůma. "Robust Approximate Inverse Preconditioning for the Conjugate Gradient Method". In: *SIAM Journal on Scientific Computing* 22.4 (2000), pp. 1317–1332.

[17] M. Benzi, C. D. Meyer, and M. Tůma. "A Sparse Approximate Inverse Preconditioner for the Conjugate Gradient Method". In: *SIAM Journal on Scientific Computing* 17.5 (1996), pp. 1135–1149.

[18] R. Bhatia. *Matrix Analysis*. Vol. 169. Springer Science & Business Media, 2013.

[19] L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, K. Wilcox, and Y. Marzouk. *Large-scale inverse problems and quantification of uncertainty*. Vol. 712. John Wiley & Sons, 2011.

[20] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. 1st ed. Boca Raton, FL, USA: Chapman, Hall/CRC Taylor, and Francis Group, 2011.

[21] D. A. Brown, A. Saibaba, and S. Vallélian. "Low rank independence samplers in Bayesian inverse problems". In: *arXiv preprint arXiv:1609.07180* (2018).

[22] M. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Míguez, and P. M. Djurić. "Adaptive Importance Sampling: The past, the present, and the future". In: *IEEE Signal Processing Magazine* 34.4 (2017).

[23] T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler, and L. C. Wilcox. "Extreme-scale UQ for Bayesian inverse problems governed by PDEs". In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press. 2012, p. 3.

[24] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler. "A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion". In: *SIAM Journal on Scientific Computing* 35.6 (2013), A2494–A2523.

[25] D. Calvetti and E. Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Vol. 2. New York, NY: Springer, 2007.

[26] K. Chaloner and I. Verdinelli. "Bayesian experimental design: A review". In: *Statistical Science* 10.3 (1995), pp. 273–304.

[27] H. Chernoff. "Locally optimal designs for estimating parameters". In: *The Annals of Mathematical Statistics* 24 (1953), pp. 586–602.

[28] E. Chow and Y. Saad. "Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions". In: *SIAM Journal on Scientific Computing* 36.2 (2014), A588–A608.

[29] J. Chung, J. G. Nagy, and D. P. O'Leary. "A weighted GCV method for Lanczos hybrid regularization". In: *Electronic Transactions on Numerical Analysis* 28 (2008), pp. 149–167.

[30] J. Chung and L. Nguyen. "Motion Estimation and Correction in Photoacoustic Tomographic Reconstruction". In: *SIAM Journal on Imaging Sciences* 10.1 (2017), pp. 216–242. eprint: `https://doi.org/10.1137/16M1082901`.

[31] J. Chung and K. Palmer. "A hybrid LSMR algorithm for large-scale Tikhonov regularization". In: *SIAM Journal on Scientific Computing* 37.5 (2015), S562–S580.

[32] J. Chung and A. Saibaba. "Generalized hybrid iterative methods for large scale Bayesian inverse problems". In: *SIAM Journal on Scientific Computing* 39.5 (2016), S24–S26.

[33] J. Chung and A. K. Saibaba. "Generalized hybrid iterative methods for large-scale Bayesian inverse problems". In: *SIAM Journal on Scientific Computing* 39.5 (2017), S24–S46.

[34] B. Cox, T. Tarvainen, and S. Arridge. "Multiple Illumination Quantitative Photoacoustic Tomography using Transport and Diffusion Models". In: *American Mathematical Society Contemporary Mathematics Series* 559 (2011).

[35] B. T. Cox, S. R. Arridge, and P. C. Beard. "Gradient-based quantitative photoacoustic image reconstruction for molecular imaging". In: *Photons Plus Ultrasound: Imaging and Sensing 2007: The Eighth Conference on Biomedical Thermoacoustics, Optoacoustics, and Acousto-optics*. SPIE BiOS. 2007, 64371T.

[36] B. T. Cox, S. R. Arridge, K. P. Köstli, and P. C. Beard. "Two-dimensional quantitative photoacoustic image reconstruction of absorption distributions in scattering media by use of a simple iterative method". In: *Applied Optics* 45 (8 2006).

[37] B. Crestel. "Advanced techniques for multi-source, multi-parameter, and multi-physics inverse problems". PhD thesis. University of Texas at Austin, 2017.

[38] A. Danielli, K. Maslov, A. Garcia-Uribe, A. M. Winkler, C. Li, L. Wang, Y. Chen, G. W. D. II, and L. V. Wang. "Label-free photoacoustic nanoscopy". In: *Journal of Biomedical Optics* 19.8 (2014).

[39] H. Dette and T. Holland-Letz. "A Geometric Characterization of c-Optimal Designs for Heteroscedastic Regression". In: *The Annals of Statistics* 37.6B (2009), pp. 4088–4103.

[40] T. Ding, K. Ren, and S. Vallélian. "A one-step reconstruction algorithm for quantitative photoacoustic imaging". In: *Inverse Problems* 31.9 (2015).

[41] FEniCS Project. *FEniCS Project 2017.1*. 2017. URL: `https://fenicsproject.org`.

[42] D. Finch, M. Haltmeier, and Rakesh. "Inversion of Spherical Means and the Wave Equation in Even Dimensions". In: *SIAM Journal on Applied Mathematics* 68.2 (2007), pp. 392–412. eprint: `https://doi.org/10.1137/070682137`.

[43] D. Finch and S. K. Patch. "Determining a Function from Its Mean Values Over a Family of Spheres". In: *SIAM Journal on Mathematical Analysis* 35.5 (2004), pp. 1213–1240. eprint: `https://doi.org/10.1137/S0036141002417814`.

[44] H. Flath, L. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. "Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations". In: *SIAM Journal on Scientific Computing* 33.1 (2011), pp. 407–432.

[45] D. C. Fong and M. A. Saunders. "LSMR: An iterative algorithm for sparse least-squares problems". In: *SIAM J. Scientific Computing* 33.5 (2011), pp. 2950–2971.

[46] H. Gao, J. Feng, and L. Song. "Limited-view multi-source quantitative photoacoustic tomography". In: *Inverse Problems* 31.6 (2015).

[47] H. Gao, S. Osher, and H. Zhao. "Quantitative Photoacoustic Tomography". In: *Mathematical Modeling in Biomedical Imaging II. Lecture Notes in Mathematics*. Ed. by H. Ammari, J. Garnier, H. Kang, and K. Solna. Berlin, Heidelberg: Springer, 2012, pp. 131–158.

[48] H. Gao, H. Zhao, and S. Osher. "Bregman methods in quantitative photoacoustic tomography". In: *UCLA CAM Report* 10-42 (2010).

[49] S. Gazzola, P. C. Hansen, and J. G. Nagy. "IR Tools: A MATLAB package of iterative regularization methods and large-scale test problems". In: *arXiv preprint arXiv:1712.05602* (2017).

[50] S. Gazzola, P. C. Hansen, and J. G. Nagy. "IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems". In: *Numerical Algorithms* (2018), pp. 1–39.

[51] S. Gazzola and J. G. Nagy. "Generalized Arnoldi-Tikhonov method for sparse reconstruction". In: *SIAM Journal on Scientific Computing* 36.2 (2014), B225–B247.

[52] S. Gazzola, P. Novati, and M. R. Russo. "On Krylov projection methods and Tikhonov regularization". In: *Electron. Trans. Numer. Anal* 44 (2015), pp. 83–123.

[53] M. G. Genton. "Classes of kernels for machine learning: A statistics perspective". In: *Journal of Machine Learning Research* 2 (2001), pp. 299–312.

[54] C. Gilavert, S. Moussaoui, and J. Idier. "Efficient Gaussian sampling for solving large-scale inverse problems using MCMC". In: *IEEE Transactions on Signal Processing* 63.1 (2015), pp. 70–80.

[55] G. H. Golub, M. Heath, and G. Wahba. "Generalized Cross-Validation as a method for choosing a good ridge parameter". In: *Technometrics* 21.2 (1979), pp. 215–223.

[56] E. Haber, M. Chung, and F. Herrmann. "An effective method for parameter estimation with PDE constraints with multiple right-hand sides". In: *SIAM Journal on Optimization* 22 (3 2012).

[57] P. C. Hansen. "The L-Curve and its Use in the Numerical Treatment of Inverse Problems". In: *in Computational Inverse Problems in Electrocardiology, ed. P. Johnston, Advances in Computational Bioengineering*. WIT Press, 2000, pp. 119–142.

[58] P. C. Hansen. "Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems". In: *Numerical algorithms* 6.1 (1994), pp. 1–35.

[59] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.

[60] P. C. Hansen and J. S. Jørgensen. "AIR Tools II: algebraic iterative reconstruction methods, improved implementation". In: *Numerical Algorithms* (2017), pp. 1–31.

[61] E. Herman, A. Alexanderian, and A. K. Saibaba. "Randomization and reweighted ℓ1-minimization for A-optimal design of linear inverse problems". In: *In Revision - SIAM Journal of Scientific Computing* (2019). Preprint arXiv:1906.03791.

[62] I. Hnětynková, M. Plešinger, and Z. Strakoš. "The regularizing effect of the Golub–Kahan iterative bidiagonalization and revealing the noise level in the data". In: *BIT Numerical Mathematics* 49.4 (2009), pp. 669–696.

[63] M. E. Hochstenbach and L. Reichel. "An iterative method for Tikhonov regularization with a general linear regularization operator". In: *J. Integral Equations Appl* 22 (2010), pp. 463–480.

[64] M. E. Hochstenbach, L. Reichel, and X. Yu. "A Golub–Kahan-type reduction method for matrix pairs". In: *Journal of Scientific Computing* 65.2 (2015), pp. 767–789.

[65] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[66] Y. Huang and Z. Jia. "Some results on the regularization of LSQR for large-scale discrete ill-posed problems". In: *Science China Mathematics* 60.4 (2017), pp. 701–718.

[67] A. Javaherian and S. Holman. "Direct quantitative photoacoustic tomography for realistic acoustic media". In: *Inverse Problems* 35.8 (2019).

[68] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–.

[69] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. 1st ed. New York, NY: Springer, 2005.

[70] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Vol. 160. Springer Science & Business Media, 2006.

[71] S. Kharchenko, L. Y. Kolotilina, A. Nikishin, and A. Yeremin. "A robust AINV-type method for constructing sparse approximate inverse preconditioners in factored form". In: *Numerical Linear Algebra with Applications* 8.3 (2001), pp. 165–179.

[72] M. E. Kilmer, P. C. Hansen, and M. I. Español. "A projection-based approach to general-form Tikhonov regularization". In: *SIAM Journal on Scientific Computing* 29.1 (2007), pp. 315–330.

[73] M. E. Kilmer and D. P. O'Leary. "Choosing regularization parameters in iterative methods for ill-posed problems". In: *SIAM Journal on Matrix Analysis and Applications* 22 (2001), pp. 1204–1221.

[74] L. Y. Kolotilina and A. Y. Yeremin. "Factorized Sparse Approximate Inverse Preconditionings I. Theory". In: *SIAM Journal on Matrix Analysis and Applications* 14.1 (1993), pp. 45–58.

[75] P. Kuchment and L. Kunyansky. "Mathematics of photoacoustic and thermoacoustic tomography". English (US). In: *Handbook of Mathematical Methods in Imaging: Volume 1, Second Edition*. Springer New York, 2015, pp. 1117–1167.

[76] L. Kunyansky. "Inversion of the spherical means transform in corner-like domains by reduction to the classical Radon transform". In: *Inverse Problems* 31.9 (2015).

[77] L. A. Kunyansky. "A series solution and a fast algorithm for the inversion of the spherical mean Radon transform". In: *Inverse Problems* 23.6 (2007), S11–S20.

[78] L. A. Kunyansky. "Explicit inversion formulae for the spherical mean Radon transform". In: *Inverse Problems* 23.1 (2007), pp. 373–383.

[79] F. Liang, C. Liu, and R. J. Carroll. *Advanced Markov Chain Monte Carlo Methods: Learning from Past Samples*. 1st ed. Chichester, West Sussex, UK: John Wiley and Sons Ltd, 2010.

[80] F. Lindgren, H. Rue, and J. Lindström. "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.4 (2011), pp. 423–498.

[81] A. V. Mamonov and K. Ren. "Quantitative photoacoustic imaging in radiative transport regime". In: *Preprint arXiv:1207.4664* (2012).

[82] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas. "A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion". In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1460–A1487.

[83] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd ed. New York, NY: Springer, 2006.

[84] W. Nowak, S. Tenkleve, and O. Cirpka. "Efficient computation of linearized cross-covariance and
auto-covariance matrices of interdependent quantities". In: *Mathematical Geology* 35.1 (2003), pp. 53–66.

[85] D. P. O'Leary and J. A. Simmons. "A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems". In: *SIAM Journal on Scientific and Statistical Computing* 714.2 (1981), pp. 474–489.

[86] D. Orban and M. Arioli. *Iterative Solution of Symmetric Quasi-Definite Linear Systems*. SIAM, 2017.

[87] D. V. Ouellette. "Schur complements and statistics". In: *Linear Algebra and its Applications* 36 (1981), pp. 187 –295.

[88] C. C. Paige and M. A. Saunders. "LSQR: An algorithm for sparse linear equations and sparse least squares". In: *ACM Transactions on Mathematical Software (TOMS)* 8.1 (1982), pp. 43–71.

[89] A. Parker and C. Fox. "Sampling Gaussian distributions in Krylov spaces with conjugate gradients". In: *SIAM Journal on Scientific Computing* 34.3 (2012), B312–B334.

[90] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. *Recycling Krylov Subspaces for Sequences of Linear Systems*. Tech. rep. Sandia National Laboratories, Mar. 2004.

[91] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. "Recycling Krylov Subspaces for Sequences of Linear Systems". In: *SIAM Journal on Scientific Computing* 28 (5 2006).

[92] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Vol. 7. SIAM, 1980.

[93] N. Petra, J. Martin, G. Stadler, and O. Ghattas. "A computational framework for infinite-dimensional Bayesian inverse problems, part II: stochastic Newton MCMC with application to ice sheet flow inverse problems". In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1525–A1555.

[94] A. Pulkkinen, B. T. Cox, S. R. Arridge, H. Goh, J. P. Kaipio, and T. Tarvainen. "Direct Estimation of Optical Parameters From Photoacoustic Time Series in Quantitative Photoacoustic Tomography". In: *IEEE Transactions on Medical Imaging* 35.11 (2016).

[95] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer-Verlag New York, 2000.

[96] C. E. Rasmussen and C. Williams. "Gaussian processes for machine learning". In: *the MIT Press* 2.3 (2006), p. 4.

[97] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. 1st ed. The MIT Press: Springer, 2006.

[98] L. Reichel, F. Sgallari, and Q. Ye. "Tikhonov regularization based on generalized Krylov subspace methods". In: *Applied Numerical Mathematics* 62.9 (2012), pp. 1215–1228.

[99] R. A. Renaut, S. Vatankhah, and V. E. Ardestani. "Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems". In: *SIAM Journal on Scientific Computing* 39.2 (2017), B221–B243.

[100] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Second. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003, pp. xviii+528.

[101] A. Saibaba and P. Kitanidis. "Efficient methods for large-scale linear inversion using a geostatistical approach". In: *Water Resources Research* 48.5 (2012), W05522.

[102] A. K. Saibaba, A. Alexanderian, and I. C. Ipsen. "Randomized matrix-free trace and log-determinant estimators". In: *Numerische Mathematik* 137 (2 2017), pp. 353–395.

[103] A. K. Saibaba, J. Chung, and K. Petroske. "Uncertainty quantification in large Bayesian linear inverse problems using Krylov subspace methods". In: *In Revision: Numerical Linear Algebra with Applications* (2019). Pre-print arXiv:1808.09066v2.

[104] A. K. Saibaba and P. K. Kitanidis. "Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems". In: *Advances in Water Resources* 82.0 (2015), pp. 124 –138.

[105] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. 2nd ed. New York, NY: Springer, 2018.

[106] M. K. Schneider and A. S. Willsky. "A Krylov subspace method for covariance approximation and simulation of random processes and fields". In: *Multidimensional Systems and Signal Processing* 14.4 (2003), pp. 295–318.

[107] M. Schweiger, S. Arridge, M. Hiraoka, and D. Delpy. "The Finite Element Method for the propagation of light in scattering media: Boundary and source conditions". In: *Medical Physics* 22 (11 1995).

[108] H. D. Simon and H. Zha. "Low-rank matrix approximation using the Lanczos bidiagonalization process with applications". In: *SIAM Journal on Scientific Computing* 21.6 (2000), pp. 2257–2274.

[109] D. P. Simpson. "Krylov subspace methods for approximating functions of symmetric positive definite matrices with applications to applied statistics and anomalous diffusion". PhD thesis. Queensland University of Technology, 2008.

[110] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk. "Optimal low-rank approximations of Bayesian linear inverse problems". In: *SIAM Journal on Scientific Computing* 37.6 (2015), A2451–A2487. eprint: `http://dx.doi.org/10.1137/140977308`.

[111] G. Stewart. "Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix". In: *Numerische Mathematik* 83 (1999).

[112] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Vol. 63. Springer, 2015.

[113] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2005.

[114] T. Tarvainen, B. T. Cox, J. P. Kaipio, and S. R. Arridge. "Reconstructing absorption and scattering distributions in quantitative photoacoustic tomography". In: *Inverse Problems* 28 (2012).

[115] L. Tenorio. *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems*. SIAM, 2017.

[116] L. Tenorio. *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems*. SIAM, 2017.

[117] L. Tenorio, F. Andersson, M. D. Hoop, and P. Ma. "Data analysis tools for uncertainty quantification of inverse problems". In: *Inverse Problems* 27.4 (2011), p. 045001.

[118] S. Toledo and H. Avron. "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix". In: *Journal of the ACM* 58.8 (2 2011).

[119] TomoWave Laboratories, Inc. *Imaging*. 2020. URL: `http://www.tomowave.com/imaging.html` (visited on 02/12/2020).

[120] B. E. Treeby and B. T. Cox. "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields". In: *Journal of Biomedical Optics* 15 (2 2010).

[121] G. J. Tserevelakis, I. Vrouvaki, P. Siozos, K. Melessanaki, K. Hatzigiannakis, C. Fotakis, and G. Zacharakis. "Photoacoustic imaging reveals hidden underdrawings in paintings". In: *Scientific Reports* 7.1 (2017).

[122] S. Vallélian. "Quantitative PAT with unknown ultrasound speed: uncertainty characterization and reconstruction methods". PhD thesis. The University of Texas at Austin, May 2015.

[123] U. Villa, N. Petra, and O. Ghattas. "hIPPYlib: an Extensible Software Framework for Large-scale Deterministic and Bayesian Inverse Problems". In: (2016).

[124] U. Villa, N. Petra, and O. Ghattas. "hIPPYlib: An Extensible Software Framework for Large-scale Deterministic and Bayesian Inverse Problems". In: *Journal of Open Source Software* 3.30 (2018).

[125] U. Villa. *Image Denoising: Tikhonov and Total Variation Regularization*. 2018. URL: `http://g2s3.com/labs/notebooks/ImageDenoising.html` (visited on 01/16/2020).

[126] U. Villa, N. Petra, and O. Ghattas. "hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs; Part I: Deterministic Inversion and Linearized Bayesian Inference". In: *arXiv e-prints* (2019). arXiv: `1909.03948`.

[127] U. Villa, N. Petra, and O. Ghattas. *hIPPYlib: An extensible software framework for large-scale deterministic and linearized Bayesian inversion*. 2020. URL: `http://hippylib.github.io`.

[128] P. Virtanen et al. "SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python". In: *arXiv e-prints*, arXiv:1907.10121 (July 2019), arXiv:1907.10121. arXiv: `1907.10121 [cs.MS]`.

[129] K. Wang, T. Bui-Thanh, and O. Ghattas. "A randomized maximum a posteriori method for posterior sampling of high dimensional nonlinear Bayesian inverse problems". In: *SIAM Journal on Scientific Computing* 40.1 (2018), A142–A171.

[130] L. V. Wang and J. Yao. "A practical guide to photoacoustic tomography in the life sciences". In: *Nature Methods* 13 (2016), 627?638.

[131] Z. Wang, J. M. Bardsley, A. Solonen, T. Cui, and Y. M. Marzouk. "Bayesian Inverse Problems with l_1 priors: A Randomize-Then-Optimize Approach". In: *SIAM Journal on Scientific Computing* 39.5 (2017), S140–S166.

[132] M. Xu and L. V. Wang. "Universal back-projection algorithm for photoacoustic computed tomography". In: *Phys. Rev. E* 71 (1 2005), p. 016706.

[133] M. Xu and L. V. Wang. "Photoacoustic imaging in biomedicine". In: *Review of Scientific Instruments* 77 (4 2006).

[134] Yuan Xu, Dazi Feng, and L. V. Wang. "Exact frequency-domain reconstruction for thermoacoustic tomography. I. Planar geometry". In: *IEEE Transactions on Medical Imaging* 21.7 (2002), pp. 823–828.

[135] G. Zangerl, M. Haltmeier, L. V. Nguyen, and R. Nuster. “Full Field Inversion in Photoacoustic Tomography with Variable Sound Speed”. In: *Applied Sciences* 9 (8 2019).

[136] H. F. Zhang, K. Maslov, G. Stoica, and L. V. Wang. “Functional photoacoustic microscopy for high-resolution and noninvasive in vivo imaging”. In: *Journal of Biomedical Optics* 24 (7 2006).

[137] Y. Zhou, J. Yao, and L. V. Wang. “Tutorial on photoacoustic tomography”. In: *Journal of Biomedical Optics* 21.6 (2016).

APPENDICES

APPENDIX

A

RESULTS USED IN CHAPTER 2

A.1 Best Approximation of H, (2.30)

We will need to use the following

Fact A.1.1. *That for a given orthonormal* $\widetilde{\mathbf{Q}} \in \mathbf{R}^{m\times n}$ *and* $\widetilde{\mathbf{H}} = \widetilde{\mathbf{Q}}^\top \widetilde{\mathbf{A}}\widetilde{\mathbf{Q}}$*,*

$$\|R(\widetilde{\mathbf{H}})\|_2 \leq \|R(\mathbf{C})\|_2 \quad \forall\, \mathbf{C} \in \mathbf{R}^{m\times m}, \quad \textit{where} \quad R(\mathbf{C}) = \widetilde{\mathbf{A}}\widetilde{\mathbf{Q}} - \widetilde{\mathbf{Q}}\mathbf{C}.$$

The proof can be found in [92, Theorem 11.4.2].

Theorem A.1.1. *Given* $\mathbf{V}_k$ *a basis for the subspace* $\mathcal{S}_k \equiv \mathcal{K}_k(\mathbf{HQ}, \mathbf{A}^\top \mathbf{\Gamma}_{noise}^{-1}\mathbf{b})$ *we have that*

$$\mathbf{T}_k \equiv \mathbf{B}_k^\top \mathbf{B}_k = \min_{\mathbf{\Delta}\in\mathbf{R}^{k\times k}} \|\mathbf{HQV}_k - \mathbf{V}_k\mathbf{\Delta}\|_{\mathbf{Q}}.$$

Proof. First we note

$$\|\mathbf{M}\|_{\mathbf{Q}} = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{M}\mathbf{x}\|_{\mathbf{Q}} = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{Q}^{1/2}\mathbf{M}\mathbf{x}\|_2 = \|\mathbf{Q}^{1/2}\mathbf{M}\|_2$$

.

Using this we can then write

$$\|\mathbf{H}\mathbf{Q}\mathbf{V}_k - \mathbf{V}_k\boldsymbol{\Delta}\|_{\mathbf{Q}} = \|\underbrace{(\mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}^{1/2})}_{\widetilde{\mathbf{A}}}\underbrace{(\mathbf{Q}^{1/2}\mathbf{V}_k)}_{\widetilde{\mathbf{Q}}} - \underbrace{(\mathbf{Q}^{1/2}\mathbf{V}_k)}_{\widetilde{\mathbf{Q}}}\boldsymbol{\Delta}\|_2.$$

Written in this way it is clear that this mat Fact A.1.1 . From (2.17) it is clear that $\mathbf{Q}^{1/2}\mathbf{V}_k$ is orthonormal and we can write

$$\begin{aligned}
\widetilde{\mathbf{H}} &= \widetilde{\mathbf{Q}}^\top\widetilde{\mathbf{A}}\widetilde{\mathbf{Q}} \\
&= \mathbf{V}_k^\top\mathbf{Q}^{1/2}\mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}^{1/2}\mathbf{Q}^{1/2}\mathbf{V}_k \\
&= \mathbf{V}_k^\top\mathbf{Q}\mathbf{H}\mathbf{Q}\mathbf{V}_k \\
&= \mathbf{T}_k
\end{aligned}$$

We will use the fact A.1.1 From (2.17) it is clear that $\mathbf{Q}^{1/2}\mathbf{V}_k$ is orthonormal and we have that

$$\begin{aligned}
&\|R(\widetilde{\mathbf{H}})\|_2 \leq \|R(\mathbf{C})\|_2 \quad \forall\, \mathbf{C} \\
\implies &\|\mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}\mathbf{V}_k - \mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{T}_k\|_2 \leq \|(\mathbf{Q}^{1/2}\mathbf{H}\mathbf{Q}\mathbf{V}_k) - \mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{C}\|_2 \quad \forall\, \mathbf{C} \\
&\|\mathbf{H}\mathbf{Q}\mathbf{V}_k - \mathbf{V}_k\mathbf{T}_k\|_{\mathbf{Q}} \leq \|\mathbf{H}\mathbf{Q}\mathbf{V}_k - \mathbf{V}_k\mathbf{C}\|_{\mathbf{Q}} \quad \forall\, \mathbf{C}.
\end{aligned}$$

□

A.2 Derivation of $\mathbf{s}_k$ using $\widehat{\mathbf{\Gamma}}_{\text{post}}$, (2.33)

First, we plug in $\widehat{\mathbf{\Gamma}}_{\text{post}} = (\lambda^2\mathbf{Q}^{-1} + \widehat{\mathbf{H}})^{-1}$ and rearrange to get

$$\begin{aligned}\widehat{\mathbf{\Gamma}}_{\text{post}}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b} &= (\lambda^2\mathbf{Q}^{-1} + \mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top)^{-1}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b} \\ &= (\lambda^2\mathbf{I} + \mathbf{Q}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top)^{-1}\mathbf{Q}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b}\,.\end{aligned}$$

Then, using the gen-GK relationships, we note that

$$\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b} = \mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{U}_{k+1}\beta_1\mathbf{e}_1 = \mathbf{V}_k\mathbf{B}_k^\top\beta_1\mathbf{e}_1\,.$$

Furthermore, using the Sherman-Morrison-Woodbury formula [65, Eqn. (0.7.4.1)], we have

$$(\lambda^2\mathbf{I} + \mathbf{Q}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top)^{-1} = \lambda^{-2}\mathbf{I} - \lambda^{-4}\mathbf{Q}\mathbf{V}_k(\mathbf{T}_k^{-1} + \lambda^{-2}\mathbf{I})^{-1}\mathbf{V}_k^\top\,.$$

Thus, we get

$$\begin{aligned}\widehat{\mathbf{\Gamma}}_{\text{post}}\mathbf{A}^\top\mathbf{\Gamma}_{\text{noise}}^{-1}\mathbf{b} &= \left(\lambda^{-2}\mathbf{I} - \lambda^{-4}\mathbf{Q}\mathbf{V}_k\left(\mathbf{T}_k^{-1} + \lambda^{-2}\mathbf{I}\right)^{-1}\mathbf{V}_k^\top\right)\mathbf{Q}\mathbf{V}_k\mathbf{B}_k^\top\beta_1\mathbf{e}_1 \\ &= \mathbf{Q}\mathbf{V}_k\left(\lambda^{-2}\mathbf{I} - \lambda^{-4}\left(\mathbf{T}_k^{-1} + \lambda^{-2}\mathbf{I}\right)^{-1}\right)\mathbf{B}_k^\top\beta_1\mathbf{e}_1 \qquad (*) \\ &= \mathbf{Q}\mathbf{V}_k(\mathbf{T}_k + \lambda^2\mathbf{I})^{-1}\mathbf{B}_k^\top\beta_1\mathbf{e}_1 \\ &= \mathbf{Q}\mathbf{V}_k\mathbf{z}_k,\end{aligned}$$

where in $(*)$ we use the fact that $(\mathbf{T}_k^{-1} + \lambda^{-2}\mathbf{I})^{-1} = \lambda^2\mathbf{I} - \lambda^4(\mathbf{T}_k + \lambda^2\mathbf{I})^{-1}$. Since $\mathbf{T}_k = \mathbf{B}_k^\top\mathbf{B}_k$, then from (2.18) we have the last equality.

A.3 Lemma of independent interest used in Theorems 2.3.2, 2.3.3 and Section 2.4.1

Lemma A.3.1. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be symmetric positive semidefinite and let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be an orthogonal projection matrix. Then*

$$\begin{aligned} |\mathsf{trace}(\mathbf{I}+\mathbf{A})^{-1} - \mathsf{trace}(\mathbf{I}+\mathbf{PAP})^{-1}| &\leq \mathsf{trace}(\mathbf{A}-\mathbf{PAP}), \\ \mathsf{trace}\left[(\mathbf{I}+\mathbf{A})(\mathbf{I}+\mathbf{PAP})^{-1}\right] &\leq n + \mathsf{trace}(\mathbf{A}-\mathbf{PAP}) \\ 0 \leq \log\det(\mathbf{I}+\mathbf{A}) - \log\det(\mathbf{I}+\mathbf{PAP}) &\leq \mathsf{trace}(\mathbf{A}-\mathbf{PAP}). \end{aligned}$$

Proof. Let $\{\lambda_i\}_{i=1}^n$ and $\{\mu_i\}_{i=1}^n$ denote the eigenvalues of $\mathbf{A}$ and $\mathbf{PAP}$. Since both matrices are positive semidefinite, their eigenvalues are non-negative. Since $\mathbf{P}$ is a projection matrix, its singular values are at most 1. The multiplicative singular value inequalities [18, Problem III.6.2] say $\sigma_i(\mathbf{PA}^{1/2}) \leq \sigma_i(\mathbf{A}^{1/2})$, so $\lambda_i \geq \mu_i$ for $i = 1, \ldots, n$, and therefore, $\mathsf{trace}(\mathbf{A}) \geq \mathsf{trace}(\mathbf{PAP})$. Then for the first inequality

$$\begin{aligned} |\mathsf{trace}(\mathbf{I}+\mathbf{PAP})^{-1} - \mathsf{trace}(\mathbf{I}+\mathbf{A})^{-1}| &= \left|\sum_{i=1}^n \frac{\lambda_i - \mu_i}{(1+\mu_i)(1+\lambda_i)}\right| \\ &\leq \left|\sum_{i=1}^n (\lambda_i - \mu_i)\right| = |\mathsf{trace}(\mathbf{A}-\mathbf{PAP})|. \end{aligned}$$

The inequalities follow since λ_i, μ_i are non-negative. The absolute value disappears since $\mathsf{trace}(\mathbf{A}) \geq \mathsf{trace}(\mathbf{PAP})$.

For the second inequality, write

$$(\mathbf{I}+\mathbf{A})(\mathbf{I}+\mathbf{PAP})^{-1} = \mathbf{A}(\mathbf{I}+\mathbf{PAP})^{-1} - \mathbf{PAP}(\mathbf{I}+\mathbf{PAP})^{-1} + \mathbf{I}.$$

Both $\mathbf{A}$ and $(\mathbf{I}+\mathbf{PAP})^{-1}$ are positive semidefinite (the second matrix is definite), so the trace of their product is nonnegative [65, Exercise 7.2.26]. Then a straightforward application

of the von Neumann trace theorem [65, Theorem 7.4.1.1] leads to

$$\mathsf{trace}(\mathbf{A}(\mathbf{I}+\mathbf{PAP})^{-1}) \leq \sum_{i=1}^{n} \frac{\lambda_i}{1+\mu_i}.$$

By utilizing its eigendecomposition, we see that $\mathsf{trace}[\mathbf{PAP}(\mathbf{I}+\mathbf{PAP})^{-1}] = \sum_{i=1}^{n} \frac{\mu_i}{1+\mu_i}$. Putting it together, we get

$$\begin{aligned}\mathsf{trace}[(\mathbf{I}+\mathbf{A})(\mathbf{I}+\mathbf{PAP})^{-1}] &\leq n + \sum_{i=1}^{n}\left(\frac{\lambda_i}{1+\mu_i} - \frac{\mu_i}{1+\mu_i}\right) \\ &\leq n + \sum_{i=1}^{n} \frac{\lambda_i - \mu_i}{1+\mu_i} \;\leq\; n + \sum_{i=1}^{n}(\lambda_i - \mu_i).\end{aligned}$$

Connecting the sum of the eigenvalues with the trace delivers the desired result.

For the third inequality, use Sylvester's determinant identity [87, Corollary 2.11] to write

$$\log\det(\mathbf{I}+\mathbf{PAP}) = \log\det(\mathbf{I}+\mathbf{A}^{1/2}\mathbf{PA}^{1/2}).$$

Denote $\mathbf{B} = \mathbf{A}^{1/2}\mathbf{PA}^{1/2}$ and introduce the notation of Loewner partial ordering [65, Section 7.7]. Let $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n\times n}$ be symmetric. Then, $\mathbf{M} \preceq \mathbf{N}$ means $\mathbf{N} - \mathbf{M}$ is positive semidefinite. Since $\mathbf{P} \preceq \mathbf{I}$, it follows that $\mathbf{B} \preceq \mathbf{A}$ [65, Theorem 7.7.2]. Then apply [1, Lemma 9], to obtain

$$0 \leq \log\det(\mathbf{I}+\mathbf{A}) - \log\det(\mathbf{I}+\mathbf{B}) \leq \log\det(\mathbf{I}+\mathbf{A}-\mathbf{B}).$$

Finally since $\log(1+x) \leq x$ for $x \geq 0$, $\log\det(\mathbf{I}+\mathbf{A}-\mathbf{B}) \leq \mathsf{trace}(\mathbf{A}-\mathbf{B})$. The proof is completed by observing that $\mathsf{trace}(\mathbf{B}) = \mathsf{trace}(\mathbf{PAP})$ by the cyclic property of trace. □

A.4 Facts used in Section 2.3.5

Using cyclic properties of the trace we have

$$\begin{aligned}\mathsf{trace}(\lambda^2\mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}}) &= \mathsf{trace}(\lambda^2\mathbf{Q}^{-1/2}\mathbf{Q}^{-1/2}(\lambda^2\mathbf{Q}^{-1}+\mathbf{H})^{-1}) \\ &= \mathsf{trace}(\lambda^2\mathbf{Q}^{-1/2}(\lambda^2\mathbf{Q}^{-1}+\mathbf{H})^{-1}\mathbf{Q}^{-1/2}) \\ &= \mathsf{trace}\left((\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}})^{-1}\right).\end{aligned}$$

In this case we use log properties and the property of determinants that for square matrices of the same size $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ to show

$$\begin{aligned}\log\det(\lambda^2\mathbf{Q}^{-1}\mathbf{\Gamma}_{\text{post}}) &= \log\det(\lambda^2\mathbf{Q}^{-1/2}\mathbf{Q}^{-1/2}(\lambda^2\mathbf{Q}^{-1}+\mathbf{H})^{-1}) \\ &= \log\det(\lambda^2\mathbf{Q}^{-1/2}(\lambda^2\mathbf{Q}^{-1}+\mathbf{H})^{-1}\mathbf{Q}^{-1/2}) \\ &= -\log\det\left(\mathbf{I}+\lambda^{-2}\mathbf{H}_{\mathbf{Q}}\right).\end{aligned}$$

Using the Sherman-Morrison-Woodbury formula [65, Eqn. (0.7.4.1)] we show that

$$\begin{aligned}\mathsf{trace}\left((\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}})^{-1}\right) &= \mathsf{trace}\left((\mathbf{I}+\lambda^{-2}\mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2})^{-1}\right) \\ &= \mathsf{trace}\left(\mathbf{I}-\mathbf{Q}^{1/2}\mathbf{V}_k(\lambda^2\mathbf{I}+\mathbf{T}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2}\mathbf{Q}^{1/2}\mathbf{V}_k)^{-1}\mathbf{T}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2}\right) \\ &= n-\mathsf{trace}\left(\mathbf{T}_k(\mathbf{T}_k+\lambda^2\mathbf{I})^{-1}\right).\end{aligned}$$

The last line comes from cyclic properties of trace and (2.17).

Using Sylvester's determinant identity [87, Corollary 2.11] and (2.17) it is apparent that

$$\begin{aligned}\log\det(\mathbf{I}+\lambda^{-2}\widehat{\mathbf{H}}_{\mathbf{Q}}) &= \log\det(\mathbf{I}+\lambda^{-2}\mathbf{Q}^{1/2}\mathbf{V}_k\mathbf{T}_k\mathbf{V}_k^\top\mathbf{Q}^{1/2}) \\ &= \log\det(\mathbf{I}+\lambda^{-2}\mathbf{T}_k).\end{aligned}$$

APPENDIX

B

RESULTS USED IN CHAPTER 4

B.1 Derivation of Lagrangian and First and Second Variations

We begin with the PDE for collimated sources

$$\begin{aligned} -\nabla \cdot (D\nabla u_i) + e^m u_i &= f_i, \quad x \in \Omega, \\ u_i + AD\nabla u_i \cdot n &= 0, \quad x \in \partial\Omega, \end{aligned} \tag{B.1}$$

and cost function

$$J(m) = \sum_{i=1}^{n_s} \left[\int_\Omega (e^m u_i - \psi_i^{obs})^2 \, dx \right] + \frac{\lambda}{2} \int_\Omega R(m) \, dx, \tag{B.2}$$

which we use to form the Lagrangian

$$\mathcal{L}(\{u_i\},\{p_i\},m) = \sum_{i=1}^{n_s}\Bigg[\int_\Omega (e^m u_i - \psi_i^{obs})^2\,dx - \int_\Omega p_i \nabla\cdot(D\nabla u_i)\,dx + \int_\Omega e^m p_i u_i\,dx$$
$$- \int_\Omega p_i f_i\,dx + \int_{\partial\Omega}\frac{1}{A}p_i u_i\,ds + \int_{\partial\Omega} p_i D\nabla u_i\cdot n\,ds\Bigg]$$
$$+ \frac{\lambda}{2}\int_\Omega R(m)\,dx.$$

We use Green's identity

$$\int_\Omega k(x)\nabla u\cdot\nabla v\,dx = -\int_\Omega v\nabla\cdot(k(x)\nabla u)\,dx + \int_{\partial\Omega} vk(x)\nabla u\cdot n\,ds,$$

to rewrite the Lagrangian as

$$\mathcal{L}(\{u_i\},\{p_i\},m) = \sum_{i=1}^{n_s}\Bigg[\int_\Omega (e^m u_i - \psi_i^{obs})^2\,dx + \int_\Omega D\nabla p_i\cdot\nabla u_i\,dx + \int_\Omega e^m p_i u_i\,dx$$
$$- \int_\Omega p_i f_i\,dx + \int_{\partial\Omega}\frac{1}{A}p_i u_i\,ds\Bigg] + \frac{\lambda}{2}\int_\Omega R(m)\,dx.$$

We remind the reader that the first variation of a function is calculated as

$$\mathcal{F}_x(x,y)[\tilde{x}] = \frac{d}{d\varepsilon}\mathcal{F}(x+\varepsilon\tilde{x},y)\bigg|_{\varepsilon=0} \qquad \forall\tilde{x},$$

and the second variation is

$$\mathcal{F}_{xy}(x,y)[\tilde{x},\hat{y}] = \frac{d}{d\varepsilon}\mathcal{F}_x(x,y+\varepsilon\hat{y})\bigg|_{\varepsilon=0} \qquad \forall\tilde{x}.$$

We will define the notation $R'(m)[\tilde{m}]$ and $R''(m)[\tilde{m},\hat{m}]$ to be the first and second variations, respectively, of the regularization term, $R(m)$.

We can then state that the first order optimality conditions are

$$\begin{aligned}
\mathcal{L}_{p_j}\left(u_j, p_j, m\right)[\tilde{p}] &= 0, \quad \forall \tilde{p} \in H^1, \quad \text{for } j = 1, ..., n_s \\
\mathcal{L}_{u_j}\left(u_j, p_j, m\right)[\tilde{u}] &= 0, \quad \forall \tilde{u} \in H^1, \quad \text{for } j = 1, ..., n_s \\
\mathcal{L}_m\left(\{u_i\}, \{p_i\}, m\right)[\tilde{m}] &= g(m)[\tilde{m}], \quad \forall \tilde{m} \in H^1
\end{aligned}$$

B.1.1 Derivation of the Weak Form of the State Equation

Since we are taking a reduced order approach, we only need to look at one source (or j-th term) at a time. We begin by evaluating

$$\begin{aligned}
\mathcal{L}(u_j, (p_j + \varepsilon\tilde{p}), m) = &\int_\Omega (e^m u_j - \psi_i^{obs})^2 \, dx - \int_\Omega D\nabla(p_j + \varepsilon\tilde{p}) \cdot \nabla u_j \, dx \\
&+ \int_\Omega e^m (p_j + \varepsilon\tilde{p}) u_j \, dx - \int_\Omega (p_j + \varepsilon\tilde{p}) f \, dx \\
&+ \int_{\partial\Omega} \frac{1}{A}(p_j + \varepsilon\tilde{p}) u_j \, ds + \frac{\lambda}{2}\int_\Omega R(m) \, dx.
\end{aligned}$$

We then take the derivative with respect to ε, then set $\varepsilon = 0$, yielding

$$\mathcal{L}_{p_j}(u_j, p_j, m)[\tilde{p}] = \int_\Omega D\nabla\tilde{p} \cdot \nabla u_j \, dx + \int_\Omega e^m \tilde{p} u_j \, dx - \int_\Omega \tilde{p} f \, dx + \int_{\partial\Omega} \frac{1}{A}\tilde{p} u_j \, ds = 0, \;\; \forall \tilde{p} \in H^1,$$

for $j = 1, ..., n_s$.

B.1.2 Derivation of the Weak Form of the Adjoint Equation

Following the previous section, we begin by evaluating

$$\begin{aligned}
\mathcal{L}((u_j + \varepsilon\tilde{u}), p_j, m) = &\int_\Omega (e^m (u_j + \varepsilon\tilde{u}) - \psi_j^{obs})^2 \, dx + \int_\Omega D\nabla p_j \cdot \nabla(u_j + \varepsilon\tilde{u}) \, dx \\
&+ \int_\Omega e^m p_j (u_j + \varepsilon\tilde{u}) \, dx - \int_\Omega p_j f \, dx \\
&+ \int_{\partial\Omega} \frac{1}{A} p_j (u_j + \varepsilon\tilde{u}) \, ds + \frac{\lambda}{2}\int_\Omega R(m) \, dx
\end{aligned}$$

We then take the derivative with respect to ε, then set $\varepsilon = 0$, yielding

$$\begin{aligned}\mathcal{L}_{u_j}(u_j, p_j, m)[\tilde{u}] = &\int_\Omega 2e^m\tilde{u}(e^m u_j - \psi_j^{obs})\,dx + \int_\Omega D\nabla p_j \cdot \nabla\tilde{u}\,dx \\ &+ \int_\Omega e^m p_j\tilde{u}\,dx + \int_{\partial\Omega}\frac{1}{A}p_j\tilde{u}\,ds = 0 \;\; \forall\tilde{u} \in H^1,\end{aligned}$$

for $j = 1, ..., n_s$.

B.1.3 Derivation of the Weak Form of the Gradient Equation

We begin by evaluating

$$\begin{aligned}\mathcal{L}(\{u_i\}, \{p_i\}, (m + \varepsilon\tilde{m})) = &\sum_{i=1}^{n_s}\Bigg[\int_\Omega (e^{m+\varepsilon\tilde{m}}u_i - \psi_i^{obs})^2\,dx + \int_\Omega D\nabla p_i \cdot \nabla u_i)\,dx \\ &+ \int_\Omega e^{m+\varepsilon\tilde{m}}p_i u_i\,dx - \int_\Omega p_i f\,dx \\ &+ \int_{\partial\Omega}\frac{1}{A}p_i u_i\,ds\Bigg] + \frac{\lambda}{2}\int_\Omega R(m + \varepsilon\tilde{m})\,dx.\end{aligned}$$

We then take the derivative with respect to ε, then set $\varepsilon = 0$, yielding

$$\begin{aligned}\mathcal{L}_m(\{u_i\}, \{p_i\}, m)[\tilde{m}] = &\sum_{i=1}^{n_s}\Bigg[\int_\Omega 2\tilde{m}e^m u_i(e^m u_i - \psi_i^{obs})\,dx + \int_\Omega \tilde{m}e^m p_i u_i\,dx\Bigg] \\ &+ \frac{\lambda}{2}\int_\Omega R'(m)[\tilde{m}]\,dx = g(m)[\tilde{m}] \;\; \forall\tilde{m} \in H^1.\end{aligned}$$

B.1.4 Derivation of the Incremental Equations and the Application of the Hessian

To derive the application of the Hessian we need to take the second variations of the Lagrangian with respect to each parameter. As before we only need to look at the j-th term for $j = 1, ...n_s$.

Using the previous results, we get

$$\begin{aligned}
\mathcal{L}_{p_j u_j}(u_j, p_j, m)[\tilde{p}, \hat{u}] &= \int_\Omega D\nabla\tilde{p} \cdot \nabla\hat{u}_j \, dx + \int_\Omega e^m \tilde{p}\hat{u}_j \, dx + \int_{\partial\Omega} \frac{1}{A}\tilde{p}\hat{u}_j \, ds \\
\mathcal{L}_{p_j m}(u_j, p_j, m)[\tilde{p}, \hat{m}] &= \int_\Omega \hat{m}e^m \tilde{p}u_j \, dx \\
\mathcal{L}_{p_j p_j}(u_j, p_j, m)[\tilde{p}, \hat{p}] &= 0 \\
\mathcal{L}_{u_j u_j}(u_j, p_j, m)[\tilde{u}, \hat{u}] &= \int_\Omega 2e^{2m}\tilde{u}\hat{u}_j \, dx \\
\mathcal{L}_{u_j p_j}(u_j, p_j, m)[\tilde{u}, \hat{p}] &= \int_\Omega D\nabla\hat{p}_j \cdot \nabla\tilde{u} \, dx + \int_\Omega e^m \hat{p}_j\tilde{u} \, dx \int_{\partial\Omega} \frac{1}{A}\hat{p}_j\tilde{u} \, ds \\
\mathcal{L}_{u_j m}(u_j, p_j, m)[\tilde{u}, \hat{m}] &= \int_\Omega 4\tilde{u}u_j\hat{m}e^{2m} \, dx - \int_\Omega 2\hat{m}\tilde{u}e^m\psi_j^{obs} \, dx + \int_\Omega p_j\hat{m}\tilde{u}e^m \, dx
\end{aligned}$$

Using the second variations we have that the incremental state equation is

$$\mathcal{L}_{p_j u_j}(u_j, p_j, m)[\tilde{p}, \hat{u}] + \mathcal{L}_{p_j m}(u_j, p_j, m)[\tilde{p}, \hat{m}] + \mathcal{L}_{p_j p_j}(u_j, p_j, m)[\tilde{p}, \hat{p}] = 0, \;\; \forall\tilde{p} \in H^1,$$

which becomes

$$\int_\Omega D\nabla\tilde{p} \cdot \nabla\hat{u}_j \, dx + \int_\Omega e^m \tilde{p}\hat{u}_j \, dx + \int_{\partial\Omega} \frac{1}{A}\tilde{p}\hat{u}_j \, ds \int_\Omega \hat{m}e^m \tilde{p}u_j \, dx = 0, \forall\tilde{p} \in H^1. \tag{B.3}$$

For all $j = 1, ..., n_s$ we can solve the incremental state equations to find the incremental state variables, $\hat{u}_j$'s.

Similarly, we have that the incremental adjoint equation is

$$\mathcal{L}_{u_j u_j}(u_j, p_j, m)[\tilde{u}, \hat{u}] + \mathcal{L}_{u_j m}(u_j, p_j, m)[\tilde{u}, \hat{m}] + \mathcal{L}_{u_j p_j})(u_j, p_j, m)[\tilde{u}, \hat{p}] = 0, \;\; \forall\tilde{u} \in H^1,$$

which becomes

$$\begin{aligned}
&\int_\Omega 2e^{2m}\tilde{u}\hat{u}_j \, dx + \int_\Omega D\nabla\hat{p}_j \cdot \nabla\tilde{u} \, dx + \int_\Omega e^m \hat{p}_j\tilde{u} \, dx + \int_{\partial\Omega} \frac{1}{A}\hat{p}_j\tilde{u} \, ds \\
&+ \int_\Omega 4\tilde{u}u_j\hat{m}e^{2m} \, dx - \int_\Omega 2\hat{m}\tilde{u}e^m\psi_j^{obs} \, dx + \int_\Omega p_j\hat{m}\tilde{u}e^m \, dx = 0, \forall\tilde{u} \in H^1.
\end{aligned} \tag{B.4}$$

For all $j = 1, ..., n_s$ we can solve the incremental adjoint equations to find the incremental

adjoint variables, $\hat{p}_j$'s.

Once we have all of the incremental state and adjoint variables we can use them in the application of the Hessian in the direction $\hat{m}$. We have that

$$\mathcal{L}_{mu_j}(\{u_j\},\{p_j\},m)[\tilde{m},\hat{u}] = \int_\Omega 4\hat{u}_j u_j \tilde{m} e^{2m}\,dx - \int_\Omega 2\tilde{m}\hat{u}_j e^m \psi_j^{obs}\,dx + \int_\Omega p_j \tilde{m}\hat{u}_j e^m$$
$$\mathcal{L}_{mp_j}(\{u_j\},\{p_j\},m)[\tilde{m},\hat{p}] = \int_\Omega \tilde{m} e^m \hat{p}_j u_j\,dx,$$

for each $j = 1, ..., n_s$ and

$$\mathcal{L}_{mm}(\{u_i\},\{p_i\},m)[\tilde{m},\hat{m}] = \sum_{i=1}^{n_s}\left[\int_\Omega 2\tilde{m}\hat{m}e^m u_i(2e^m u_i - \psi_i^{obs})\,dx + \int_\Omega \tilde{m}\hat{m}e^m p_i u_i\,dx\right] + \frac{\lambda}{2}\int_\Omega R''(m)[\tilde{m},\hat{m}]\,dx.$$

Putting everything together we can write that the application of the Hessian in a direction $\hat{m}$, $H(m)[\tilde{m},\hat{m}]$ is

$$\sum_{i=1}^{n_s}\mathcal{L}_{mu_i}(\{u_i\},\{p_i\},m)[\tilde{m},\hat{u}] + \mathcal{L}_{mm}(\{u_i\},\{p_i\},m)[\tilde{m},\hat{m}] + \sum_{i=1}^{n_s}\mathcal{L}_{mp_i}(\{u_i\},\{p_i\},m)[\tilde{m},\hat{p}] = H(m)[\tilde{m},\hat{m}],\ \forall \tilde{m} \in H^1$$

which becomes

$$\sum_{i=1}^{n_s}\left[\int_\Omega 4u_i\hat{u}_i e^{2m}\tilde{m}\,dx - \int_\Omega 2\tilde{m}e^m\hat{u}_i\psi_i^{obs}\,dx + \int_\Omega p_i\tilde{m}e^m\hat{u}_i\,dx + \int_\Omega \tilde{m}\hat{p}_i u e^m\,dx + \int_\Omega 2\tilde{m}\hat{m}e^m u_i(2e^m u_i - \psi_i^{obs})\,dx + \int_\Omega p_i\tilde{m}\hat{m}e^m u_i\,dx\right] + \frac{\lambda}{2}\int_\Omega R''(m)[\tilde{m},\hat{m}]\,dx = H(m)[\tilde{m},\hat{m}]. \tag{B.5}$$