Probabilistic Approximations of Matrix Decompositions for Inverse Problems

Owen Dillon

Department of Mathematics
The University of Auckland

Supervisor: Jari P. Kaipio

Acknowledgements

First, thanks to my supervisor Jari Kaipio. His knowledge, guidance and support has been hugely influential since my first year of university. My academic development would be unrecognisable were it not for his input. I also acknowledge the University of Auckland for hosting my PhD. and awarding me a scholarship.

I also thank the University of Eastern Finland for supporting me on 3 seperate visits. In particular, thanks to Aku Seppanen, Ville Kolehmainen, Matti Kortelainen, Jenni Tick, Antti Voss, and Matti Niskanen, for helping to arrange my visit and partaking in productive discussions.

I would also like to thank my friends and fellow students Anton Gulley, Attique Ur-Rehman, Hwan Goh, Tim Evans, Susana Guzman, Pascal Cheon, Lia Lee, Scarlett Shi, Elías Sigüenza, Annabelle Xu, Vincent Wolfgramm-Russell, Kam Hung Yau, Nemanja Poznanović, Gareth Gordon and Abhishek Bhardwaj. All contributed valuable ideas to this thesis, and our discussions gave me confidence in the topics this thesis would cover. They have also all helped to make my time completing this thesis thoroughly enjoyable. In particular, thanks to Ruanui Nicholson for his help with Finite Element Methods and his distractions on everything else.

I would also like to thank my family for their constant support. In particular, thanks to my parents Ross and Christine Dillon for coping with me these past 25 years. I also thank my fiancée Cherry Ngan, who inspired me to start this program and is a constant source of inspiration and motivation.

Abstract

Inverse problems arise in a wide range of subjects, such as medicine , cosmology, and engineering. These problems are often high dimensional, making them difficult to analyse and solve, particularly in industrial applications where timeframes are narrow. The aim of this thesis is to provide broadly applicable methods to reduce the computational costs involved in inverse problems.

I consider solving inverse problems in two stages. The first is the *offline* phase. This is the stage of modelling and analysing the problem. This is sometimes called the "laboratory" or "research" stage. The second stage is *online*, where real data is coming in and a corresponding solution is found. Typically, the offline stage requires and can access greater computational resources than the online stage. Methods of reducing computational costs at the offline and online stage are presented in this thesis.

This thesis primarily takes the Bayesian viewpoint of inverse problems. Much of the analysis in this thesis is of linear inverse problems with Gaussian unknowns. Such problems can be expressed in terms of linear algebra, so much of this thesis is concerned with numerical linear algebra. A particular focus is approximate matrix decompositions. This thesis makes use of the Sherman-Morrison-Woodbury formula/matrix inversion lemma, Schur Complements, pseudoinverses, the eigenvalue decomposition, the singular value decomposition, the Cholesky decomposition and particularly the QR decomposition. This thesis presents a methodology of computing the QR decomposition of sample approximations to matrices, and demonstrates applications of such factorisations to inverse problems. This thesis also makes use of probabilistic algorithms for constructing approximate matrix decompositions. A probabilistic method of constructing locally accurate matrix approximations is introduced.

A particular focus of this thesis is the Bayesian approximation error framework, in which simulations are computed at the offline stage in order to reduce computational cost at the online stage. The Bayesian approximation error, sample QR factorisation, and locally accurate probabilistic approximations are combined to reduce computational costs.

The methods of this thesis are demonstrated separately, typically on 1D deconvolution. These methods are then combined and applied to the linear problems of 2D deconvolution and x-ray tomography. The methods of this thesis are also applied to the nonlinear simplified conductivity imaging problem.

List of Notation

$d \in \mathbb{R}^{n_d}$	Data
n_d	Dimensionality of data
$\mathbf{x}$	Unknown of interest
$\mathbf{e}$	Noise
$\mathbf{A}_e$	Model i.e. $d = \mathbf{A}_e(\mathbf{x}, \mathbf{e})$
$\mathbf{A}$	Forward model/forward operator i.e. $d = \mathbf{A}(\mathbf{x}) + e$ in additive noise model
$e \in \mathbb{R}^{n_d}$	Finite dimensional noise
$x \in \mathbb{R}^{n_x}$	Finite dimensional approximation of $\mathbf{x}$
n_x	Dimensionality of x
A_e	Finite dimensional model i.e. $d = A_e(x, e)$
A	Finite dimensional forward model/forward operator i.e. $d = A(x) + e$ in additive noise model. $d = Ax + e$ with $A \in \mathbb{R}^{n_\times n_x}$ in linear additive noise model
Ω	Domain e.g. $\Omega = [0, 1] \times [0, 1]$
$\partial\Omega$	Boundary of domain
$\mathbf{t}$	"Spatial" variable i.e. $\mathbf{t} \in \Omega$
n_t	Spatial dimension e.g. if $\Omega = [0, 1] \times [0, 1]$ then $n_t = 2$
$t_x \in \mathbb{R}^{n_x \times n_t}$	Finite dimensional approximation of spatial variable i.e. $\mathbf{x}(t(j, :)) = x(j, :)$
$d_t = \mathbf{A}_e(\mathbf{x}_t, \mathbf{e}_t)$	A particular realisation of the model $d = \mathbf{A}_e(\mathbf{x}, \mathbf{e})$
$\mathbf{x}_t$	Ground truth
$B \in \mathbb{R}^{n \times m}$	General $n \times m$ real valued matrix
$B(j, :) \in \mathbb{R}^{1 \times m}$	j'th row of B
$B(:, k) \in \mathbb{R}^{n \times 1}$	k'th column of B
$B(j, k) \in \mathbb{R}$	j'th row of k'th column of B
$B(j : k, :)$	Rows j to k of B
$b \in \mathbb{R}^n$	General length n vector
$I_n \in \mathbb{R}^{n \times n}$	Identity matrix
$I_{n,m} \in \mathbb{R}^{m \times m}$	$I_{n,m} = \begin{pmatrix} I_n & 0_{n,(m-n)} \\ 0_{(m-n),n} & 0_{(m-n),(m-n)} \end{pmatrix}$
$0_{n,m} \in \mathbb{R}^{n \times m}$	Zero matrix
$U \in \mathbb{R}^{n \times n}$	Unitary matrix i.e. $UU^T = U^T U = I_n$
$u_j \in \mathbb{R}^n$	The j'th column of U i.e. $u_j = U(:, j)$
Λ	A diagonal matrix of decreasing eigenvalues i.e $\Lambda(j, j) = \lambda_j$ with $\lambda_j > \lambda_k$ for $j < k$
λ_j	j'th eigenvalue

$B = U_B \Lambda_B U_B^T$	Eigendecomposition of symmetric $B \in \mathbb{R}^{n \times n}$ i.e. $Bu_j = \lambda_j u_j$
$V \in \mathbb{R}^{m \times m}$	Unitary matrix
$v_j \in \mathbb{R}^m$	The j'th column of V
$D \in \mathbb{R}^{n \times m}$	Diagonal matrix of decreasing singular values $D(j,j) = d_j$ with $d_j > d_r \geq 0$ for $j < k$
d_j	j'th singular value
κ_B	The condition number of matrix B, $\kappa_B = \frac{d_1}{d_{\min n,m}}$
$B = U_B D_B V_B^T$	Singular value decomposition of matrix $B \in \mathbb{R}^{n \times m}$
$r_{B,t}$	True rank of matrix B i.e. $d_{r_{B,t}} > 0$ and $d_{r_{B,t}+j} = 0$ for all $j > 0$
$U_r \in \mathbb{R}^{n \times r}$	First r columns of U, $U_r = U(:, 1:r)$
$U_{\text{null}} \in \mathbb{R}^{n \times (n-r)}$	Columns $r+1$ to n of U, $U_{\text{null}} = U(:, r+1:n)$
$D_r \in \mathbb{R}^{r \times r}$	Principle $r \times r$ submatrix of D, $D_r = D(1:r, 1:r) \in \mathbb{R}^{r \times r}$
$B = U_{B,r} D_{B,r} V_{B,r}^T$	Thin Singular Value Decomposition of rank r matrix B
$B^\dagger = V_{B,r_t} D_{B,r_t}^{-1} U_{B,r_t}^T \in \mathbb{R}^{m \times n}$	(Moore-Penrose) Pseudoinverse of $B \in \mathbb{R}^{n \times m}$
$B_r^\dagger = V_{B,r} D_{B,r}^{-1} U_{B,r}^T \in \mathbb{R}^{m \times n}$	r term pseudoinverse of $B \in \mathbb{R}^{n \times m}$
$x^\dagger$	(Moore-Penrose) Pseudoinverse estimate of x_t i.e. if $d_t = Ax_t + e_t$, $x^\dagger = A^\dagger d_t$
$x_r^\dagger$	r term pseudoinverse/tSVD estimate of x_t i.e. if $d_t = Ax_t + e_t$, $x^\dagger = A_r^\dagger d_t$
$\|b\|_p$	l_p norm of vector $b \in \mathbb{R}^n$ e.g. $\|b\|_2 = \left(b(1)^2 + b(2)^2 + \cdots + b(n)^2\right)^{\frac{1}{2}}$
$\|B\|_p$	induced l_p norm of matrix B i.e. $\|B\|_p = \max_{\|b\|_p = 1} \|Bb\|_p$
x_{ls}	A solution of the least squares problem $x_{\text{ls}} = \min_x \{\|Ax - y\|_2\}$
X_{ls}	The set of all x_{ls}
$B^\dagger \in \mathbb{R}^{m \times n}$	The pseudoinverse of $B \in \mathbb{R}^{n \times m}$, $B^\dagger = V_{B,r} D_{B,r}^{-1} U_{B,r}^T$
Ω_x	Domain containing unknown of interest x
$\mathbf{z}$	Auxiliary unknown, typically parameter values in region around Ω_x
Ω_z	Domain of auxiliary unknown i.e $\Omega = \Omega_x \cup \Omega_z$
$\mathbf{s}$	Unknown over Ω i.e. for $\mathbf{t} \in \Omega_x$, $\mathbf{s}(\mathbf{t}) = \mathbf{x}(\mathbf{t})$ and for $\mathbf{t} \in \Omega_z$, $\mathbf{s}(\mathbf{t}) = \mathbf{z}(\mathbf{t})$
ξ	Additional unknowns, typically extra model parameters
ω	Combined unknown i.e. $A(x, z, \xi) = A(\omega)$
F	"Physics" mapping
c	Underlying "physical" values i.e. $c = F(s)$
M	"Measurement" mapping. $y = A(x, z, \xi) = M\Big(F(x, z, \xi)\Big)$

$x \sim \mathcal{N}(\mu_x, \Gamma_x)$	Stating variable x to be normally distributed with mean $\mu_x \in \mathbb{R}^{n_x}$ and covariance $\Gamma_x \in \mathbb{R}^{n_x \times n_x}$
$\pi_x(x) = \pi(x)$	Probability density function of variable x. $\pi(x) \geq 0$ for all x and $\int_\infty \pi(x)dx = 1$
$\mathbb{E}$	Expectation operator i.e. $\mathbb{E}(x) = \mu_x$ and $\mathbb{E}((x-\mu_x)(x-\mu_x)^T) = \Gamma_x$
σ_e^2	Homogeneous noise variance $\sigma_e^2 \in \mathbb{R}$ i.e. $e \sim \mathcal{N}(\mu_e, \Gamma_e)$ with $\Gamma_e = \sigma_e^2 I_{n_d}$
x_{tik}	Standard Tikhonov regularised solution $x_{\text{tik}} = \min_x \left\{ \|y - Ax\|_2^2 + \alpha \|x\|_2^2 \right\}$
$\alpha \in \mathbb{R}$	Tikhonov regularisation parameter
$\hat{A}$	"Regularised" A. In standard Tikhonov estimation, $\hat{A} = \begin{pmatrix} A \\ \sqrt{\alpha} I_{n_x} \end{pmatrix} \in \mathbb{R}^{(n_d+n_x)\times n_x}$
$\hat{d}$	"Regularised" d. For standard Tikhonov, $\hat{d} = \begin{pmatrix} d \\ 0_{n_x,1} \end{pmatrix} \in \mathbb{R}^{(n_d+n_x)\times 1}$
$\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$	Higher resolution approximation of unknowns of interest i.e. $n_{\bar{x}} > n_x$
$\pi_x(x)$	Prior probability distribution
$\pi_{d\|x}(d\|x)$	Likelihood probability distribution
$\pi_{x\|d}(x\|d)$	Posterior probability distribution
x_{MAP}	Maximum a Posteriori estimate
$x_{\text{CM}} = \mu_{x\|d}$	Conditional mean estimate
x_j	The j'th sample of x
$\varepsilon \in \mathbb{R}^{n_d}$	Bayesian approximation error e.g. $\varepsilon = \bar{A}(\bar{\omega}) - A(x)$
$\hat{\mu}_x$	Approximation of μ_x, typically from samples
$\hat{\Gamma}$	Approximation of Γ, typically from samples
$\Gamma_{x\|d} \in \mathbb{R}^{n_x \times n_x}$	Posterior covariance
$\hat{\sigma} \in \mathbb{R}^{n_x}$	Diagonal of $\Gamma_{x\|d}$
$P_{x,\bar{x}} \in \mathbb{R}^{n_x \times n_{\bar{x}}}$	Projection matrix from $\bar{x}$ to x i.e. $x = P_{x,\bar{x}}\bar{x}$
$C_{\bar{x},\bar{s}} \in \mathbb{R}^{n_{\bar{x}} \times n_{\bar{s}}}$	"Cutting" matrix such that $\bar{x} = C_{\bar{x},\bar{s}}\bar{s}$. Note that $C_{\bar{x},\bar{s}} = P_{\bar{x},\bar{s}}$
L	A matrix such that $LL^T = \Gamma$
$\tilde{L}_x$	A matrix such that $\tilde{L}^T\tilde{L} = \Gamma^{-1}$
$\hat{X} \in \mathbb{R}^{n_x \times m}$	Sample matrix of x i.e. $\hat{X} = (x_1, x_2, \ldots, x_m)$
$X \in \mathbb{R}^{n_x \times m}$	Mean removed sample matrix i.e. $X = (x_1 - \mu_x, x_2 - \mu_x, \ldots, x_m - \mu_x)$
$\Upsilon \in \mathbb{R}^{n_d \times m}$	Mean removed sample matrix of ε
$\hat{Q}_x \in \mathbb{R}^{n_x \times n_x}$	Unitary matrix spanning columns of X
$\hat{R}_x \in \mathbb{R}^{n_x \times m}$	Upper triangular matrix such that $X = \hat{Q}_x\hat{R}_x$
$Q_x \in \mathbb{R}^{n_d \times r_x}$	Matrix with orthonormal columns approximately spanning columns of X
$R_x \in \mathbb{R}^{r_x \times m}$	Upper triangular matrix such that $X \approx Q_x R_x$
$M_x \in \mathbb{R}^{r_x \times r_x}$	$M_x = \frac{1}{m} R_x R_x^T$
$M_{\varepsilon\|x} \in \mathbb{R}^{r_\varepsilon \times r_\varepsilon}$	$M_{\varepsilon\|x} = \frac{1}{m}\left(R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x^T (R_x R_x^T)^{-1} R_x R_\varepsilon^T \right)$
$\kappa \in \mathbb{R}$	Small positive number
$A_{s,r}$	r term series expansion approximation of A e.g. $A \approx A_s = U_r D_r V_r^T$
A_c	Compressed approximation to A .g. $A \approx A_c = U_r D_r V_r^T$

$\mathcal{O}$	Order e.g. an operation requiring $2n^2 + 42n + 9,999$ flops would be said to require $\mathcal{O}(n^2)$ flops
ζ	Residual e.g. $\zeta = x^\dagger - x_t$
$\mathcal{F}(\mathbf{x})$	Fourier transform of $\mathbf{x}$
$\mathcal{F}(x)$	Discrete (fast) Fourier transform of x
J_x	The Jacobian of A at particular value x
$\mathbf{TV}(\mathbf{x})$	Total variation (TV) of $\mathbf{x}$. For differentiable $\mathbf{x}$ over Ω, $\mathbf{TV}(\mathbf{x}) = \int_\Omega \lvert\nabla\mathbf{x}(\mathbf{t})\rvert d\mathbf{t}$
$\mathrm{TV(x)}$	Finite dimensional analogue total variation
$\nabla\mathbf{x}$	Gradient of $\mathbf{x}$
$\nabla \cdot \mathbf{F}$	Divergence of continuously differentiable vector field $\mathbf{F}$
K_x	Stiffness matrix of finite element method corresponding to particular x

Contents

Chapter 1

Introduction

We can generally think of inverse problems as problems where we wish to reconstruct parameters from indirect measurements. In this thesis, I define inverse problems to be problems of parameter estimation that are *ill-posed* in the sense of Hadamard [1]. When solutions to a problem exist, are unique, and are stable, the problem is *well-posed*. When any of these conditions are not satisfied, the problem is ill-posed. In this thesis, "inverse problems" are ill-posed problems of reconstructing parameters from indirect measurements corrupted by noise. This is the typical usage of the term inverse problem in mathematical literature [2, 3, 4, 5].

Inverse problems as defined above arise naturally in a wide variety of applications, such as such as medicine [6, 7, 8, 9, 10], cosmology [11, 12, 13, 14], engineering [15, 16, 17, 18], synthetic aperture radar [19, 20], finding and evaluating mineral deposits [21, 22, 23, 24, 25] and astronomy [14, 26]. As a result, there is much interest and activity in the field of inverse problems. Inverse problems are often high dimensional, making them difficult to analyse and solve, particularly in industrial applications where timeframes are narrow [27, 28, 29, 30].

A recurring issue with inverse problems is computational cost [3, 4, 31, 32]. This is often due to the *forward model* requiring extremely high dimensional approximations to simulate the physical system to sufficient accuracy. In many applications, the computational power required makes it prohibitively difficult to analyse or implement a solver of the inverse problem. For example, in the dental tomography problem of [33, 34, 35], the dimension of the data and measurements can be $10^6 - 10^7$. Approaching this problem directly would require a $10^{12} - 10^{14}$ element linear forward model, and creating the inverse model would require $10^{18} - 10^{21}$ arithmetic operations. The reason this particular problem is currently approachable is due to the sparsity of the forward model and some useful analytic results, dramatically reducing the above generic operator (full matrix) computation costs. This thesis aims to not only develop new methods that could apply to new problems, but also show how they can be incorporated to improve existing methods that e.g. exploit sparsity.

Thesis Aims

The aim of this thesis is to provide methods by which the computational burden associated with inverse problems can be reduced. I shall consider both the *offline* (laboratory) and *online* (implementation) stages of solving inverse problems.

The "recursive QR" method of computing low rank approximations of matrices from samples is introduced in this thesis. This is of particular interest when combined with the Bayesian Approximation Error (BAE) framework of [36]. When recursive QR and BAE are used together, low rank approximations to relevant matrices with can be constructed with minimal additional time.

The results of [37] are extended in this thesis. The original work shows how randomised sampling can be used to form approximate decompositions of matrices, referred to as "global sample approximations" or "global sample compressions" in this thesis. Since publication in 2011, the original paper has accrued over 580 citations in such diverse fields as image analysis [38], neuroscience [39] and geophysics [40]. The extension in this thesis is to the problem of finding an approximation that is "accurate" in a specific region. This extension is also applicable to nonlinear operators. Such approximations are referred to as "local sample approximations" or "local sample compressions" in this thesis. This thesis demonstrates how to combine local sample compressions with recursive QR and BAE.

The ideas of BAE, recursive QR and local sample approximation are combined in a variety of ways in this thesis to make multiple methodological improvements. Application of these methods is shown with example problems. In this thesis, a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive is used. In the applications considered, matrices of interest contain $70,000,000,000$ elements and may not be sparse e.g. covariance matrices or inverses of sparse matrices. In IEEE 754 double-precision binary floating-point format, such a matrix would occupy 550 GB of RAM. Using the methods of this thesis, the offline stage takes around 1 minute and the online stage takes around 0.3 seconds.

Thesis Structure

Chapter 2 of this thesis formally reviews and describes the field of inverse problems. The chapter concludes by introducing the problem of deconvolving a signal in 1 dimension. This particular problem is returned to multiple times for the sake of demonstrating various methods.

Chapter 3 discusses some methods of computing estimates in inverse problems. These methods are called "regularisation", and allow for a "plausible" solution to be found.

Chapter 4 reviews the Bayesian framework for inverse problems, and demonstrates how regularisation methods can be interpreted within this framework. The Bayesian framework provides a natural way to both explicitly consider the assumptions made in regularisation methods and to quantify uncertainty in reconstructions.

Chapter 5 discusses *priors*. The prior is used to formally describe how "plausible" a solution is. The construction of various priors is shown, along with how these priors impact the inverse problem "solution". The computational cost of using these priors is also discussed.

Chapter 6 discusses various computational methods for inverse problems. This is the most important chapter in the thesis. Of particular interest is my discussion of the "Bayesian Approximation Error" (BAE) framework of [36] and the "global sample approximation" of [37]. The technique of "recursive QR" is presented, and application of recursive QR to BAE and global sample approximation is shown. An extension of [37] to form the "local sample approximation" is also presented.

Chapter 7 combines the concepts of earlier chapters to present a computationally efficient approach for inverse problems making use of the methods of Chapter 6.

Chapters 8, 9 and 10 each approach a seperate inverse problem with the methods of this thesis. These chapters are not in depth studies of the applications, presented instead as demonstrations of the methodological improvements proposed in Chapters 6 and 7.

Chapter 8 applies the methods of this thesis to 2D deconvolution. This is a natural problem to consider, increasing the computational complexity over the 1D deconvolution example.

Chapter 9 applies the methods of this thesis to x-ray tomography. This is also a linear problem, but with different structure and implementation issues. This demonstrates how the methods of this thesis can be adapted for a variety of problem types.

Chapter 10 applies the methods of this thesis to a simplified case of conductivity imaging. This nonlinear problem demonstrates how the presented methods can be applied to a more general case.

Chapter 11 concludes the thesis. A summary of the ideas presented in this thesis and how they can be applied together to reduce computational costs in the offline and online phases is given, along with a discussion of some potential future work.

Chapter 2

Inverse Problems

This chapter outlines the nature of inverse problems. Rigorous investigation of inverse problems is generally attributed to Tikhonov and Arsenin [41]. This chapter primarily follows the discussion of [4].

Let a physical system modelled as

$$d = \mathbf{A}_e(\mathbf{x}, \mathbf{e}) \tag{2.1}$$

where d is measured data, $\mathbf{A}_e$ is a physical model, $\mathbf{x}$ are parameters of the model and $\mathbf{e}$ are additional unknowns. Let

$$\mathbf{y} = \mathbf{A}(\mathbf{x}) \tag{2.2}$$

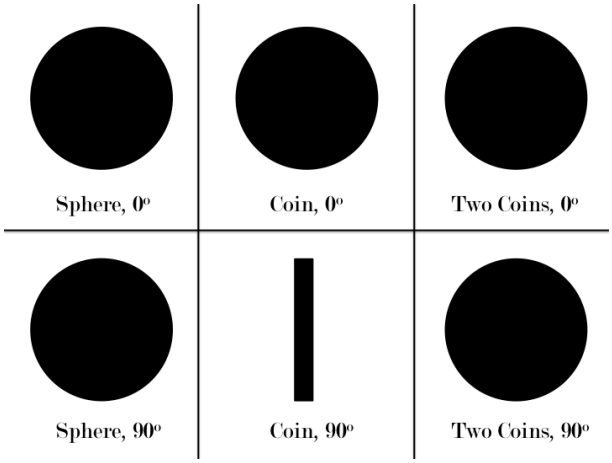
where $\mathbf{y}$ is the *model prediction*. Note the absence of e. We refer to finding $\mathbf{y}$ given $\mathbf{x}$ as the *forward problem*. We refer to $\mathbf{A}$ is the *forward operator*, or *forward mapping*. We refer to finding $\mathbf{x}$ given d as the *inverse problem*.

Consider having an object casting its shadow on a wall. The forward model is examining the object and predicting the shadow. In this case, $\mathbf{x}$ is the geometry of the object and d is the shadow. The inverse problem is estimating the geometry of the object by examining the shadow. This example illustrates the problem of *uniqueness* in inverse problems. If you see a circle on the screen, you do not know if the projectionist is holding up a ball, a disc, a rod, or any other object with a round silhouette. The "solution" to the inverse problem is not unique.

Now suppose you see the shadow once the object is rotated 90 degrees. If you still see a circle, you might conclude the object is a sphere. If you see a vertical line, you might think it is a coin. This is illustrated in 2.1, along with a third possibility - that the two circle shadows could also correspond to two interlocked coins. This additional data removes some possible solutions, but solutions are still not unique.

We also consider *stability*. In the forward problem, this would correspond to a small change in the object causing a large change in the shadow. This cannot happen for this forward model, so this forward problem is considered stable.

Figure 2.1: Shadows of 3 objects at 0 and 90 degree rotations



Consider now the stability of the inverse problem. Could a small change in the shadow d cause a large change in our reconstruction of $\mathbf{x}$? A small aberration in the shadow of a long object could require removing a large volume from the reconstruction. This inverse problem would therefore be considered somewhat unstable.

Another consideration is *existence*. For example, what if *we* do not move the object, but the object moves *on its own*, changing the shadow d? Our model $\mathbf{A}_e$ does not allow for this possibility, so we would say there is no such $\mathbf{x}$ for this data d.

Consider the problem of finding $\mathbf{x}$ in the equation $d = \mathbf{A}_e(\mathbf{x}, \mathbf{e})$ where d and $\mathbf{A}_e$ are known. When solutions to the problem exist, are unique and are stable, the problem is *well-posed* (in the sense of Hadamard) [1]. We would also refer to $\mathbf{A}_e$ as *well-conditioned*. If a problem fails to satisfy *any* of these conditions, the problem is *ill-posed*, and $\mathbf{A}_e$ as *ill-conditioned*. In this thesis, "inverse problems" are ill-posed problems of reconstructing parameters from indirect measurements corrupted by noise. This definition is consistent with most mathematical texts on inverse problems [2, 3, 4].

2.1 Linear Finite Dimensional Inverse Problems

Let a system be modelled as

$$d = \mathbf{A}_e(\mathbf{x}, e) = \mathbf{A}(\mathbf{x}) + e \tag{2.3}$$

which we call the additive noise model. Let this model be represented as

$$d = Ax + e \tag{2.4}$$

where $d \in \mathbb{R}^{n_d}$, $x \in \mathbb{R}^{n_x}$, $A \in \mathbb{R}^{n_d \times n_x}$ and $e \in \mathbb{R}^{n_d}$. This thesis refers to A and x as the *discretisation* of $\mathbf{A}$ and $\mathbf{x}$. Linear models are used in various applications, such as radio astronomy [26], computed tomography [10] and radar imaging [20]. By considering finite dimensional linear models, we can ground our discussion of inverse problems in linear algebra.

We are primarily concerned with $\mathbb{R}$ in this thesis, so we give results for real valued matrices $B \in \mathbb{R}^{n \times m}$ where n and m are arbitrary natural numbers. Many of these results extend to $\mathbb{C}$, but I only give results for $\mathbb{R}$ as this is the focus of this thesis.

Many of these results can be found in standard linear algebra textbooks such as [42]. We primarily refer to [43] and [44]. Application of these concepts to inverse problems is given in [2]. Additional notes on computation can be found in [32] and analysis in [45].

Theorem 2.1.1 (Eigendecomposition of Real Symmetric Matrices). *For all real, symmetric matrices $B \in \mathbb{R}^{n \times n}$ there exist matrices U and Λ such that*

$$B = U \Lambda U^T \tag{2.5}$$

where $U = \left(u_1, u_2, \ldots, u_n\right) \in \mathbb{R}^{n \times n}$ with $UU^T = U^T U = I_n$ and $u_j \in \mathbb{R}^n$, and $\Lambda = \operatorname{diag}\left(\lambda_1, \lambda_2, \ldots, \lambda_n\right)$ with $\lambda_j \in \mathbb{R}$ and $\lambda_j \geq \lambda_k$ for all $j \leq k$. Furthermore $Bu_j = \lambda_j u_j$.

Proof See [42, 43].

Now consider the more general case. Let B be an arbitrary matrix in $\mathbb{R}^{n \times m}$. The matrices $BB^T \in \mathbb{R}^{n \times n}$ and $B^T B \in \mathbb{R}^{m \times m}$ are symmetric. This leads us to the Singular Value Decomposition (SVD).

Theorem 2.1.2 (The Singular Value Decomposition (SVD)). *For all matrices $B \in \mathbb{R}^{n \times m}$ there exist matrices $U \in \mathbb{R}^{n \times n}$, $D \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{m \times m}$ such that*

$$B = UDV^T \tag{2.6}$$

with $BB^T = U(DD^T)U^T$ the eigendecomposition of BB^T and $B^T B = V(D^T D)V^T$ the eigendecomposition of $B^T B$. That is, $UU^T = U^T U = I_n$, $VV^T = V^T V = I_m$, and

$$D = \begin{pmatrix} D_{r_t} & 0_{r_t, m-r_t} \\ 0_{n-r_t, r_t} & 0_{n-r_t, m-r_t} \end{pmatrix} \tag{2.7}$$

where $D = \operatorname{diag}\left(d_1, d_2, \ldots, d_{r_t}\right) \in \mathbb{R}^{\mathrm{r_t} \times \mathrm{r_t}}$ and $d_j \geq d_k > 0$ for all $j \leq k$.

Proof See [42, 43].

We refer to r_t as the *rank* of B. We refer to B as a *rank r_t matrix*. Note that the right 0 blocks of D disappear when $r_t = m$, the bottom 0 blocks disappear for $r_t = n$, and $0 \leq r_t \leq \min\{n, m\}$.

We often break up the SVD as

$$B = UDV^T = U_{r_t} D_{r_t} V_{r_t}^T + U_{\text{null}} 0_{n-r_t, m-r_t} V_{\text{null}}^T \tag{2.8}$$

$$= U_{r_t} D_{r_t} V_{r_t}^T \tag{2.9}$$

where $U_{r_t} \in \mathbb{R}^{n \times r_t}$, $D_{r_t} \in \mathbb{R}^{r_t \times r_t}$, $V_{r_t} \in \mathbb{R}^{m \times r_t}$, $U_{\text{null}} \in \mathbb{R}^{n \times (n-r_t)}$, and $V_{\text{null}} \in \mathbb{R}^{m \times (m-r_t)}$. The form in Equation (2.9) is called the *thin* SVD.

We now define the *4 fundamental subspaces* of B. The columns of V_{null} make up the *null space* or *kernel* of B, so named because

$$B\left(\sum_{j=r_t+1}^{m} b_j v_j\right) = \sum_{j=r_t+1}^{m} b_j \underbrace{(Bv_j)}_{=0} = 0. \tag{2.10}$$

Similarly, the columns of U_{null} make up the null space of B^T, also known as the *left nullspace* or *cokernel* of B. The columns of U_{r_t} make up the *range*, *column space* or *image* of B. The columns of V_{r_t} form the *row space* or *coimage* of B. Note that for any $x \in \mathbb{R}^m$ we can write

$$Bx = B\sum_{j=1}^{m} b_j v_j = B\sum_{j=1}^{r_t} b_j v_j \tag{2.11}$$

$$= y = \sum_{j=1}^{r_t} c_j u_j = \sum_{j=1}^{r_t} d_j b_j u_j \tag{2.12}$$

where $b_i \in \mathbb{R}$ and $d_j b_j = c_j \in \mathbb{R}$.

We now return to the model $d = Ax + e = y + e$. The SVD $A = UDV^T$ allows us to analyse various aspects of the the forward operator:

- The rank r_t of the matrix. When $r_t < \min\{n_d, n_x\}$, we say A is *rank deficient*.
- Given a tolerance τ, which is typically a fraction of the largest singular value e.g. $\tau = \frac{d_1}{100}$ or some small multiple of machine tolerance, we take the number of diagonal elements with $||d_j|| > \tau$ as r. We call r the *effective* or *numerical* rank, depending on how we choose the tolerance. When $r < \min\{n_d, n_x\}$, we say A is *effectively* or *numerically rank deficient*.
- Decomposing $x = b_1 v_1 + b_2 v_2 + \cdots + b_m v_m$ implies $y = Ax = d_1 b_1 u_1 + d_2 b_2 u_2 + \cdots + d_m b_m u_m$. Keeping in mind that $d_1 \geq d_2 \geq \cdots \geq d_m$, we see how components of x are amplified in y.

Now consider the problem of finding $x \in \mathbb{R}^{n_x}$ in the equation $Ax = y$ given we know $A \in \mathbb{R}^{n_y \times n_x}$ and $y \in \mathbb{R}^{n_y}$. In the case that $n_x < n_y$, we say the problem is *overdetermined*. In the case that $n_x > n_y$, we say the problem is *underdetermined*.

The *condition number* of full rank A is defined as $\kappa = \frac{d_1}{d_{r_t}}$. If A is rank deficient, we define $\kappa = \infty$. When κ is low (near 1) finding x from y is a *stable* problem. When κ is large, finding x from y is an *unstable* problem. Finding x from data $d = Ax + e$ would therefore be an inverse problem by our definition.

We wish to solve $Ax = y$ for x. It could be no such x exists. We instead find the *noiseless least squares estimate*

$$x_{\text{ls,n}} = \min_x \{\|Ax - y\|_2\} \tag{2.13}$$

where $\|\bullet\|_2$ is the typical vector 2-norm. Note that x_{ls} is not necessarily unique. We refer to the set of all $x_{\text{ls,n}}$ as $X_{\text{ls,n}}$.

We now define some language for this thesis. Let

$$d = Ax_t + e \tag{2.14}$$

where x_t is the *ground truth*. The *least squares problem* is finding the (noisy) least squares estimate

$$x_{\text{ls}} = \min_x \{\|Ax - d\|_2\} \tag{2.15}$$

where we note the substitution of y from Equation (2.13) for d in Equation (2.15). We define this x_{ls} to be a *solution* of the least squares problem, and a *estimate* of the ground truth.

The *Moore-Penrose pseudoinverse* of A is formed as

$$\begin{aligned} A^\dagger &= VD^\dagger U^T && (2.16)\\ &= \begin{pmatrix} V_{r_t} & V_{\text{null}} \end{pmatrix} \begin{pmatrix} D_{r_t}^{-1} & 0_{r_t,m-r_t} \\ 0_{n-r_t,r_t} & 0_{n-r_t,m-r_t} \end{pmatrix} \begin{pmatrix} U_{r_t}^T \\ U_{\text{null}}^T \end{pmatrix} && (2.17)\\ &= V_{r_t} D_{r_t}^{-1} U_{r_t}^T && (2.18) \end{aligned}$$

where we note that $D_{r_t}^{-1} = \text{diag}(\frac{1}{\text{d}_1}, \frac{1}{\text{d}_2}, \ldots, \frac{1}{\text{d}_{\text{r}_\text{t}}})$. In this thesis, we will refer to the Moore-Penrose pseudoinverse as simply the *pseudoinverse*, and the form in Equation (2.18) as the *thin pseudoinverse*. Other types of pseudoinverse can be found in [43, 44].

Note that

$$AA^\dagger = UDV^T VD^\dagger U^T = UDD^\dagger U^T = U\hat{I}_{r_t,n_y} U^T \tag{2.19}$$

where

$$\hat{I}_{r_t,n_y} = \begin{pmatrix} I_{r_t} & 0_{r_t,n_y-r_t} \\ 0_{n_y-r_t,r_t} & 0_{n_y-r_t,n_y-r_t} \end{pmatrix} \in \mathbb{R}^{n_y \times n_y} \tag{2.20}$$

and $I_r \in \mathbb{R}^{r_t \times r_t}$ is the $(r \times r)$ identity matrix. Similarly

$$A^\dagger A = VD^\dagger U^T UDV^T = VD^\dagger DV^T = VI_{r,n_x}V^T \tag{2.21}$$

and clearly in the case that $r = n_x$, $A^\dagger A = VI_{n_x}V^T = I_{n_x}$ and in the case that $r = n_y$, $AA^\dagger = UI_{r,n_y}U^T = I_{n_y}$. We now present a key theorem.

Theorem 2.1.3 (Optimality of the Pseudoinverse).

$$x^{\dagger,n} = A^\dagger y = \min_{x \in X_{\mathrm{ls,n}}} ||x||_2 \tag{2.22}$$

Proof See [42, 43].

In other words, $x^{\dagger,n}$ is the solution of the noiseless least squares problem with minimum 2-norm. We call $x^{\dagger,n}$ the (noiseless) *pseudoinverse solution* of the (noiseless) least squares problem. We can use the SVD to characterise *all* least squares solutions.

Theorem 2.1.4 (Least Squares solutions and the SVD). *For all* $x_{\mathrm{ls,n}} \in X_{\mathrm{ls,n}}$,

$$x_{\mathrm{ls,n}} = \min_x \left\{ \|Ax - y\|_2 \right\} = A^\dagger y + \sum_{j=r_t+1}^{n_x} b_j v_j \tag{2.23}$$

where each $b_j \in \mathbb{R}$ *is arbitrary.*

Proof Proofs can be found in [42, 43, 44]. We give a proof here to emphasise the interpretation.

Given that $x^{\dagger,n} = A^\dagger y$ minimises $\|Ax - y\|_2$, i.e. $\left\|Ax^\dagger - y\right\|_2 = \alpha$, then

$$\begin{aligned}
\left\| A\left(x^\dagger + \sum_{i=r+1}^{n_x} b_i v_i \right) - y \right\|_2 &= \left\| Ax^{\dagger,n} r + \sum_{i=r+1}^{n_x} b_i \underbrace{(Av_i)}_{=0} - y \right\|_2 \\
&= \left\| Ax^{\dagger,n} \right\|_2 \\
&= \alpha
\end{aligned}$$

Given that the singular values d_j of A are positive, the above is also used in the proof of the optimality of the pseudoinverse.

The pseudoinverse solution is widely used. This would imply that solutions with smaller 2-norms are in some way preferable. This leads us into the idea of picking $x = x^\dagger + \sum_{i=r+1}^{n_x} b_i v_i$ to have particular features. This will be discussed in more depth later in this thesis in Chapters 4 and 5.

We now consider a few cases:

- $n_x \geq n_y = r_t$. This is the underdetermined full rank case. Then $A = UDV^T = UD_{n_y}V_{n_y}^T$. Therefore $A\sum_{j=n_y+1}^{n_x} b_j v_j = 0_{n_y,1}$ for all $b_j \in \mathbb{R}$, and $Ax = y$ has solutions for all y. Therefore

$$x_{\text{ls,n}} = \min_x \{\|Ax - y\|_2\} = A^\dagger y + \sum_{j=n_y+1}^{n_x} b_j v_j$$

with

$$Ax_{\text{ls,n}} = y$$

for all y. That is, exact solutions of the noiseless least squares problem exist but are not unique.

- $n_x \geq n_y > r_t$. This is the underdetermined rank deficient case. Then $A = UDV^T = U_{r_t}D_{r_t}V_{r_t}^T$. We can write $y = \sum_{j=1}^{r_t} c_j u_j + \sum_{j=r_t+1}^{n_y} c_j u_j$ and note that

$$x_{\text{ls,n}} = \min_x \{\|Ax - y\|_2\} = A^\dagger y + \sum_{j=r_t+1}^{n_x} b_j v_j$$

with

$$\|Ax_{\text{ls,n}} - y\|_2 = \left\| \sum_{j=r_t+1}^{n_y} c_j u_j \right\|_2 = \sqrt{\sum_{j=r_t+1}^{n_y} c_j^2}$$

i.e. we can find x such that Ax fits y up to the projection of y on the subspace spanned by the columns of U_r. So exact solutions of the noiseless least squares problem *may* exist but are not unique.

- $n_y \geq n_x = r_t$. This is the overdetermined full (column) rank case. Then $A = UDV^T = U_{n_x}D_{n_x}V^T$. Therefore

$$x_{\text{ls,n}} = \min_x \{\|Ax - y\|_2\} = A^\dagger y$$

where we note that $x_{\text{ls,n}}$ is unique. We further note that

$$\|Ax_{\text{ls,n}} - y\|_2 = \left\| \sum_{j=n_x+1}^{n_y} c_j u_j \right\|_2 = \sqrt{\sum_{j=n_x+1}^{n_y} c_j^2}$$

i.e. we can find x such that Ax fits y up to the projection of y on the subspace spanned by the columns of U_r. So exact solutions of the noiseless least squares problem may exist and are unique.

- $n_y \geq n_x > r_t$. This is the overdetermined rank deficient case. Then $A = UDV^T = U_{r_t} D_{r_t} V_{r_t}^T$. Therefore

$$x_{\mathrm{ls,n}} = \min_x \{\|Ax - y\|_2\} = A^\dagger y + \sum_{j=r_t+1}^{n_x} b_j v_j$$

noting that $x_{\mathrm{ls,n}}$ is not unique in this case. We further note that

$$\|Ax_{\mathrm{ls,n}} - y\|_2 = \left\| \sum_{j=n_x+1}^{n_y} c_j u_j \right\|_2 = \sqrt{\sum_{j=n_x+1}^{n_y} c_j^2}$$

similar to the $n_y \geq n_x = r_t$ case. So exact solutions of the noiseless least squares problem may exist and are not unique.

We can demonstrate concepts relating to the SVD in the simple case of fitting a polynomial to data.

2.2 Demonstration: Fitting a Polynomial

Let a real valued function $\mathbf{y}(\mathbf{t})$ be known at points $t = \big(t(1), t(2), \ldots, t(n)\big) \in \mathbb{R}^n$. Let $t(j) \neq t(k)$ for all $j \neq k$. The function values are in a vector $y = \Big(\mathbf{y}\big(t(1)\big), \mathbf{y}\big(t(2)\big), \ldots, \mathbf{y}\big(t(n)\big)\Big) \in \mathbb{R}^n$. We want to approximate $\mathbf{y}(t)$ with a polynomial $\mathbf{x}(t) = x(1)t^2 + x(2)t + x(3)$ where $x = \big(x(1), x(2), x(3)\big) \in \mathbb{R}^3$ is the vector of polynomial coefficients. We are trying to find x such that $y(j) \approx x(3)t(j)^2 + x(2)t(j) + x(1)$ i.e. we try to fit x to y.

We construct a matrix

$$A = \begin{pmatrix} 1 & t(1) & t(1)^2 \\ 1 & t(2) & t(2)^2 \\ \vdots & \vdots & \vdots \\ 1 & t(n) & t(n)^2 \end{pmatrix} \tag{2.24}$$

such that $Ax \approx y$. We shall work through a few different cases to show how the problem can be understood in terms of the SVD.

Case 1: $n = 2$. Finding x from y is underdetermined. We shall take

$$t = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, y = \begin{pmatrix} 6 \\ 15 \end{pmatrix}, \tag{2.25}$$

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix} = UDV^T \tag{2.26}$$

$$\approx \begin{pmatrix} -0.3245 & -0.9459 \\ -0.9459 & 0.3245 \end{pmatrix} \begin{pmatrix} 4.8375 & 0 & 0 \\ 0 & 0.7735 & 0 \end{pmatrix} \begin{pmatrix} -0.2626 & -0.8033 & 0.5345 \\ -0.4581 & -0.3837 & -0.8018 \\ -0.8492 & 0.4554 & 0.2673 \end{pmatrix}^T. \tag{2.27}$$

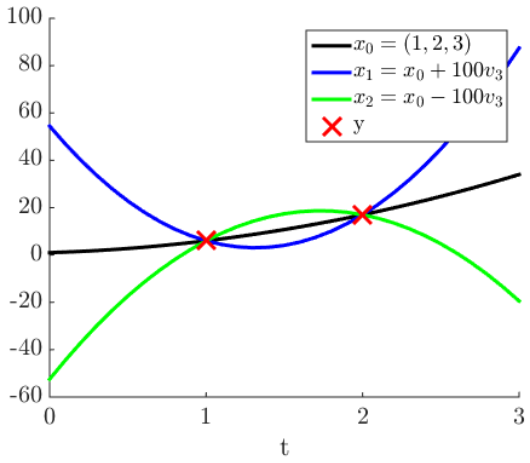


Figure 2.2: A few parabolas that fit the data

We see immediately that this is an underdetermined full rank problem. This is what we expect, as we know that there are infinitely many parabolas that pass exactly through a given 2 points. In other words, exact solutions exist but are not unique. We show this in Figure 2.2, where we plot 3 parabolas that fit the data, but with different nullspace component v_3.

We now analyse $y = Ax$. We generate multiple $x_j \in \mathbb{R}^3$ with $\|x_j\|_2 = 1$. We plot these example x_j values in Figure 2.3. We then compute $y_j = Ax_j$ for each x_j. These are plotted in Figure 2.4, along with d_1u_1 and d_2u_2. This shows how we can expect to get y's with a large component along u_1 and a smaller component in u_2. Also recall that components of x along v_3 will map to 0. It is this effect that "fills in" the ellipse in Figure 2.4.

We now analyse $x = A^\dagger y$ where $A^\dagger = VD^\dagger U^T$. We generate multiple y_j with $\|y_j\|_2 = 1$. These are plotted in Figure 2.5. We then plot $x_j = A^\dagger y_j$ in Figure 2.6 as red circles. We also

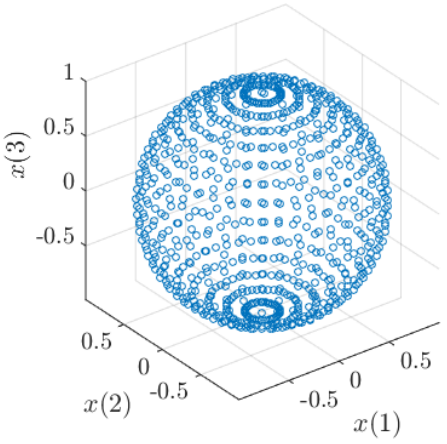


Figure 2.3: Several x_j with $\|x_j\|_2 = 1$

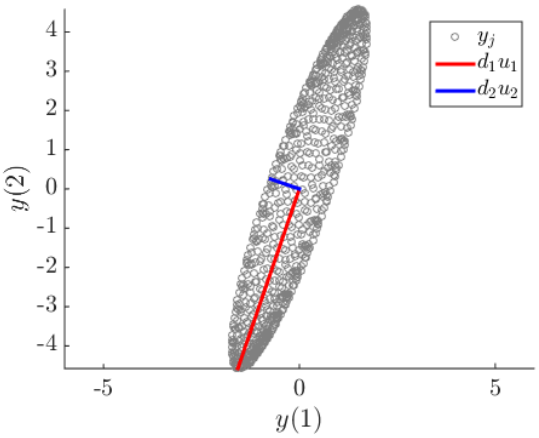


Figure 2.4: Corresponding $y_j = Ax_j$

plot $x_{j,2} = x_j + v_3$ as orange circles. Each $x_{j,2}$ is a solution to $\min_x \{\|Ax - y\|\}$, but with a nullspace v_3 component, plotted in green. We also plot $\frac{1}{d_1}v_1$ in black and $\frac{1}{d_2}v_2$ in blue.

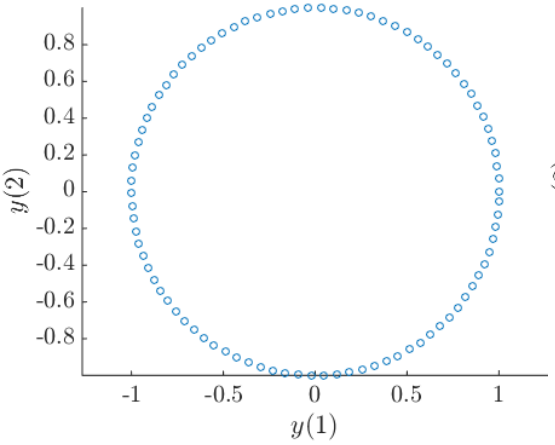


Figure 2.5: Several y_j with $\|y_j\|_2 = 1$.

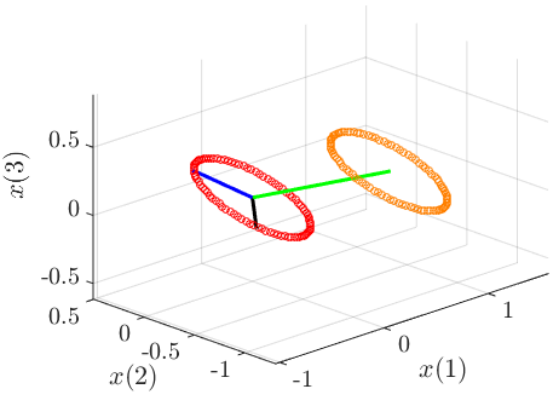


Figure 2.6: Corresponding $x_j = A^\dagger y_j$

These visualisations show how information can be nearly/completely lost, in terms of the SVD. These concepts are at the core of inverse problems, and we will return to them often.

The above example is trivial, yet easy to visualise. We can see and understand that solutions are not unique.

Consider the case $n = 3$ i.e. we know $y = \Big(\mathbf{y}\big(t(1)\big), \mathbf{y}\big(t(2)\big), \mathbf{y}\big(t(3)\big)\Big)$. There is x such that $y = Ax$. The solution x is exact and unique.

Consider the case $n > 3$. There *may* be an exact solution x to $Ax = y$, but there *probably* is no exact solution. The least squares solution however does exist and is unique. We will not go into detail for these cases as that would offer little additional insight. We shall instead discuss a simple inverse problem that raises issues of *stability*.

2.3 1D Deconvolution

Let $\mathbf{s}(\mathbf{t})$ be a real valued 1 dimensional function. We take the support of $\mathbf{s}$ to be $\Omega = [0, 20]$. Let $\Omega_x = [5, 15]$ the domain of $\mathbf{x}(\mathbf{t})$ and $\Omega_z = [0, 5) \cup (15, 20]$ the domain of $\mathbf{z}(\mathbf{t})$.

Let $\mathbf{c}(\mathbf{t})$ be the convolution of $\mathbf{s}(\mathbf{t})$ with the kernel $\boldsymbol{\xi}(\mathbf{t})$. This can be modelled as

$$\mathbf{c}(\mathbf{t}) = \mathbf{F}(\mathbf{s}, \boldsymbol{\xi}) \tag{2.28}$$

$$= \int_{-\infty}^{\infty} \boldsymbol{\xi}(\mathbf{t} - \boldsymbol{\tau})\mathbf{s}(\boldsymbol{\tau})d\boldsymbol{\tau} \tag{2.29}$$

where $\mathbf{F}$ is the continuous convolution operator.

Let $t = \big(t(1), t(2), \ldots, t(n)\big) = \big(0, \Delta_t, 2\Delta_t, \ldots, n\Delta_t = 20\big) \in \mathbb{R}^n$ be evenly spaced points in Ω. Let $t_x \in \mathbb{R}^{n_x}$ be the subset of t in Ω_x and $t_z \in \mathbb{R}^{n_z}$ be the subset of t in Ω_z. Note that $n = n_x + n_z$.

Let $t_d \in \mathbb{R}^{n_d}$ be a subset of t_x. We say that these are the *measurement points*. Note that the measurement points are in Ω_x for this problem. Let

$$\mathbf{y} = \mathbf{M}\Big(\mathbf{c}(\mathbf{t})\Big) \tag{2.30}$$

$$= \mathbf{M}\Big(\mathbf{F}(\mathbf{s}, \boldsymbol{\xi})\Big) \tag{2.31}$$

$$= \mathbf{A}(\mathbf{s}, \boldsymbol{\xi}) \tag{2.32}$$

where $\mathbf{y} \in \mathbf{R}^{n_d}$ is the forward model prediction. $\mathbf{M}$ is the continuous "measurement" operator, here simply extracting $\mathbf{c}(\mathbf{t})$ at points t_d. $\mathbf{A}$ is the continuous forward operator.

Let

$$x = \Big(\mathbf{x}\big(t_x(1)\big), \mathbf{x}\big(t_x(2)\big), \ldots, \mathbf{x}\big(t_x(n_x)\big)\Big) \in \mathbb{R}^{n_x} \tag{2.33}$$

$$z = \Big(\mathbf{z}\big(t_z(1)\big), \mathbf{z}\big(t_z(2)\big), \ldots, \mathbf{z}\big(t_z(n_z)\big)\Big) \in \mathbb{R}^{n_z} \tag{2.34}$$

$$s = (x, z) \in \mathbb{R}^n \tag{2.35}$$

$$\xi = \Big(\boldsymbol{\xi}\big(t(1)\big), \boldsymbol{\xi}\big(t(2)\big), \ldots, \boldsymbol{\xi}\big(t(n)\big)\Big) \in \mathbb{R}^n \tag{2.36}$$

be finite dimensional approximations of $\mathbf{x}$, $\mathbf{z}$, $\mathbf{s}$ and $\boldsymbol{\xi}$ respectively. We now consider the finite dimensional problem

$$\begin{aligned} d &= A(s, \xi) + e && (2.37)\\ &= MF_\xi s + e && (2.38)\\ &= Mc + e && (2.39)\\ &= y + e && (2.40) \end{aligned}$$

where $d \in \mathbb{R}^{n_d}$ is the measured data. $F \in \mathbb{R}^{n \times n}$ is the discrete convolution operator, approximating the integration in Equation (2.29) by quadrature, using the values in s and ξ. $c \in \mathbb{R}^n$ is an approximate finite dimensional approximation of the convolved signal, $c(j) \approx \mathbf{c}(t(j))$. $M \in \mathbb{R}^{n_d \times n}$ is the discrete measurement operator, extracting values of c corresponding to the measurement points t_d. $y \in \mathbb{R}^{n_d}$ is an approximate finite dimensional approximation of the convolved signal at the measurement points, $y(j) \approx \mathbf{y}(j) = \mathbf{c}(t_d(j))$.

We use a Gaussian kernel for this problem. We plot the convolution kernel in Figure 2.7. Note that the integral in Equation (2.29) extends over all of $\mathbf{R}$. The subdomain Ω_z is wider than the kernel ξ. The signal $\mathbf{z}$ in Ω_z is referred to as "padding" for the signal $\mathbf{x}$ in Ω_x. The effect of padding is to ensure that

$$\begin{aligned} \mathbf{c}(\mathbf{t}) &= \mathbf{F}(\mathbf{s}, \boldsymbol{\xi}) && (2.41)\\ &= \int_{-\infty}^{\infty} \boldsymbol{\xi}(\mathbf{t} - \boldsymbol{\tau})\mathbf{s}(\boldsymbol{\tau}) d\boldsymbol{\tau} && (2.42)\\ &\approx \int_{\Omega} \boldsymbol{\xi}(\mathbf{t} - \boldsymbol{\tau})\mathbf{s}(\boldsymbol{\tau}) d\boldsymbol{\tau} && (2.43) \end{aligned}$$

for $\mathbf{t} \in \Omega_x$. Padding is important for improving the quality of the approximation

$$x(j) \approx \mathbf{c}(t_x(j)). \tag{2.44}$$

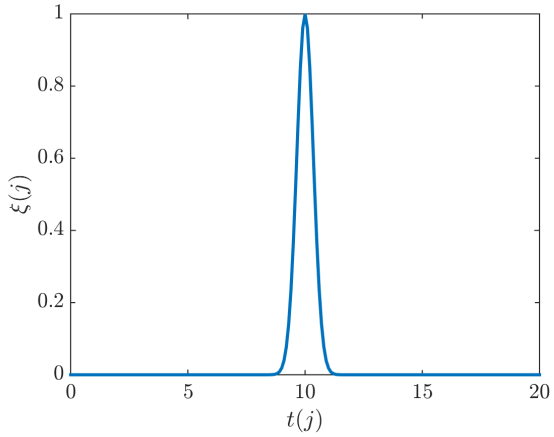


Figure 2.7: The convolution kernel $\xi(t)$

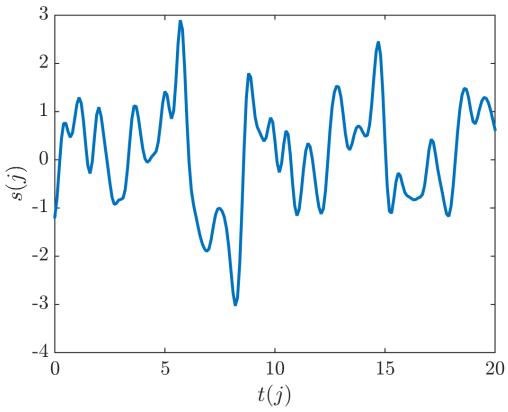


Figure 2.8: An example discretised signal s

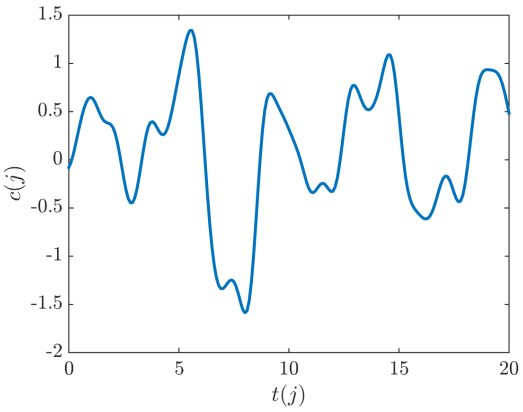


Figure 2.9: The convolution $c = Fs$

Let $\Delta_t = 0.1$. With this discretisation, $n_x = 101, n_z = 100$ and $n = n_x + n_z = 201$. We show an example discretised signal s in Figure 2.8, and the corresponding convolved signal $c = Fs$ in Figure 2.9.

We now examine the discretised convolution operator F. Note that F is symmetric, so we can compute the eigendecomposition $F = U\Lambda U^T$ rather than the more general SVD. The eigenvalues have been plotted in Figure 2.10. In this case, only approximately a quarter of the eigenvalues are appreciably large. This can be interpreted as about three quarters of the information in s being almost removed from c. We can express this formally by considering

the expansion

$$s = \sum_{j=1}^{n} b_j u_j \tag{2.45}$$

where $b_j \in \mathbb{R}$ and $u_j \in \mathbb{R}^n$ is an eigenvector of F. We can then write

$$c = Fs = U\Lambda U^T s = U\Lambda U^T \sum_{j=1}^{n} b_j u_j = \sum_{j=1}^{n} \lambda_j b_j u_j \tag{2.46}$$

$$\approx \sum_{j=1}^{50} \lambda_j b_j u_j \tag{2.47}$$

where we note that eigenvalues λ_j of F are such that $\lambda_j \approx 0$ for $j > 50$.

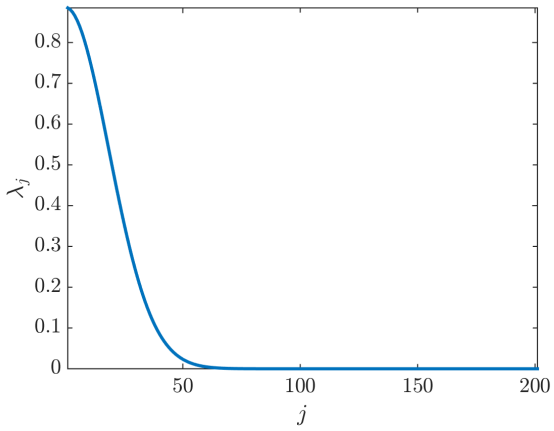


Figure 2.10: The eigenvalues of F

We now examine the eigenvectors of F. Figure 2.11 shows the 4 largest eigenvectors of F. Unsurprisingly, we see they correspond to slowly oscillating/very smooth inputs. These are the kind of signals we would expect to have the largest effect on the output of a convolution operator. Figure 2.12 shows the 4 smallest eigenvectors of F. Note that these eigenvectors oscillate rapidly. This is expected - signals that oscillate rapidly around 0 will average to 0 after convolution. It is these rapid variations that are difficult to reconstruct, as they will have little impact on the data.

This gives us insight into F, the "physics" part of A. To discuss A itself, we must include the measurement model M. We shall examine A for 2 cases of M.

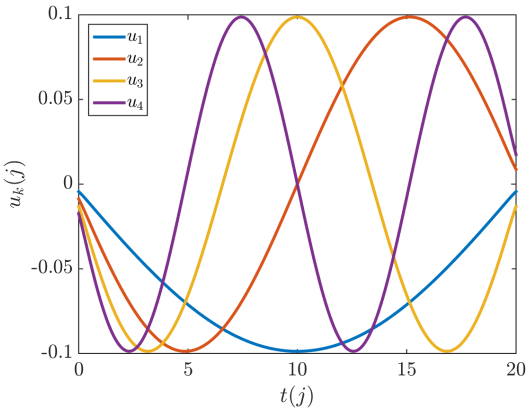


Figure 2.11: The largest eigenvectors of F.

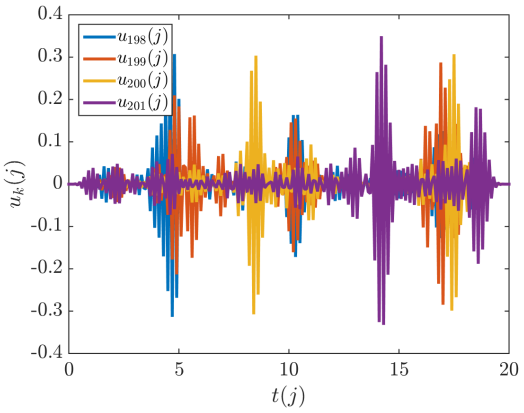


Figure 2.12: The smallest eigenvectors of F.

2.3.1 Deconvolution: Case 1

Suppose $n_d = 11$, $t_d = (5, 6, \ldots, 14, 15)$. Now our model $A = MF \in \mathbb{R}^{11\times 201}$ is simply the 11 rows of F corresponding to the 11 measurement points t_d. The "ground truth" x_t is shown in Figure 2.13. Figure 2.14 shows the full convolved signal c in Ω_x, along with the measured data d. Note that $d \neq y$ due to the noise e. In this case, $e \sim \mathcal{N}(0_{n_d}, \sigma_e^2 I_{n_d})$ with $\sigma_e^2 \approx \frac{\max\{|c|\}}{100}$ i.e. 1% additive white noise.

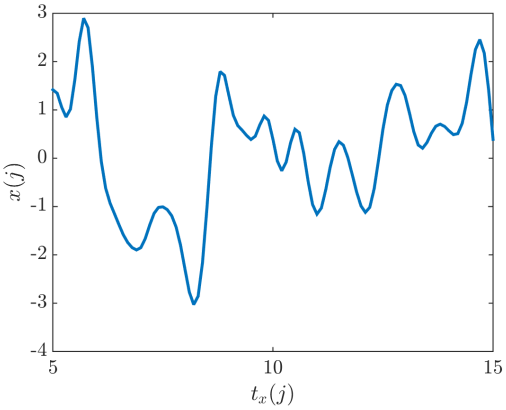


Figure 2.13: The signal x in the region Ω_x

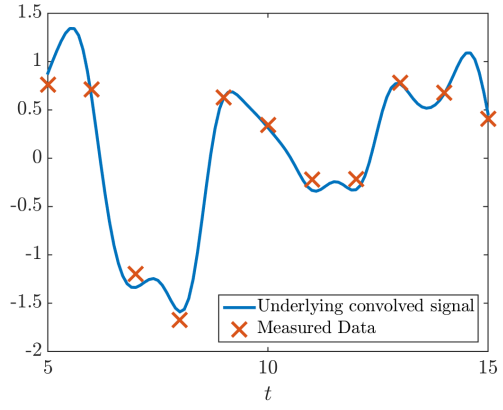


Figure 2.14: The convolution c on Ω_x and the measured data d

We now investigate the singular value decomposition $A = UDV^T$ of A. Note that we can

write $A = UDV^T = UD_rV_r^T$ where $r = 11$. We can write the least squares solutions as

$$s_{\text{ls}} = V_r D_r^{-1} U^T c + \sum_{i=12}^{201} b_i v_i \tag{2.48}$$

$$= A^\dagger c + \sum_{i=12}^{201} b_i v_i \tag{2.49}$$

for arbitrary $b_i \in \mathbb{R}$. We have much freedom in the fit. The singular values are plotted in Figure 2.15, and we see that while the inverse problem is extremely underdetermined, the low number of measurement points has truncated A to the point that the condition number $\kappa_A \approx 1.3$ is quite reasonable.

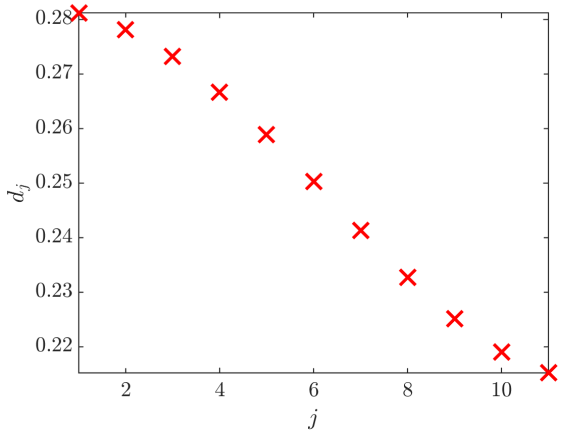


Figure 2.15: The singular values of A with $n_m = 11$

Consider the discrepancy between the pseudoinverse solutions to the noiseless and noisy least squares problems. We can consider this formally as

$$\left\| A^\dagger d - A^\dagger y \right\|_2 = \left\| A^\dagger (y + e) - A^\dagger y \right\|_2 \tag{2.50}$$

$$= \left\| A^\dagger e \right\|_2 \tag{2.51}$$

$$\leq \frac{\|e\|_2}{d_1 1} \approx 4.6 \|e\|_2 \tag{2.52}$$

i.e. the largest possible difference between the noisy and noiseless pseudoinverse least squares solutions is around 4.6 times the norm of the noise.

The pseudoinverse solution $x^\dagger = A^\dagger d$ to the noisy least squares problem $\min_x \{\|d - Ax\|\}$ is shown in Figure 2.16 alongside the ground truth x_t. We see that $x^\dagger$ is a poor estimate of x_t.

Figure 2.17 shows the data d along with$Ax^\dagger$. Note that even though $x^\dagger$ is a poor approximation to x_t, $Ax^\dagger$ is a good approximation to d.

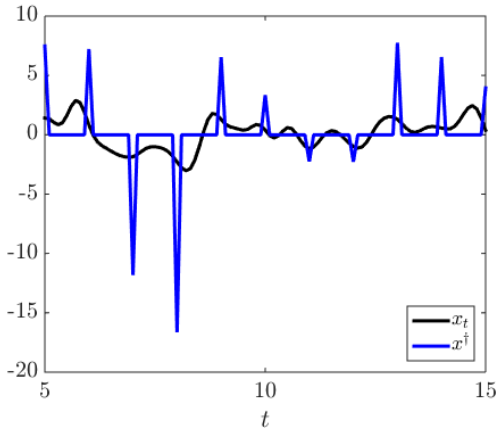


Figure 2.16: The pseudoinverse estimate $x^\dagger$ alongside ground truth x_t

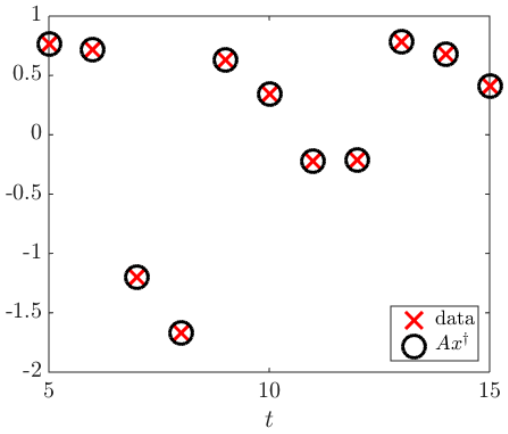


Figure 2.17: The measured data alongside model predictions $Ax^\dagger$

2.3.2 Deconvolution: Case 2

Let $n_m = 51$, $t_d = (5, 5.2, 5.4, \ldots, 14.8, 15)$. Now the model $A = MF \in \mathbb{R}^{51\times 201}$ is every second row of F. The singular values of A are plotted in Figure 2.18. Note that the spectrum of A now contains many (approximate) zeros. In this case the condition number is $\kappa_A \approx 10^6$ and the discrepancy in the noiseless and noisy least squares solutions is

$$\left\| A^\dagger d - A^\dagger y \right\|_2 \leq \frac{\|e\|_2}{d_{51}} \approx 10^7 \|e\|_2 \tag{2.53}$$

which may cause problems. So while our data has "improved" in the sense that we have more measurement points, the noise in the data can now cause large changes in the estimate $x^\dagger$.

Figure 2.19 shows the convolved signal $c = Fs$ on Ω_x along with the measured data $d = As + e = M(Fs) + e$. Note that the variance of the noise is the same as in Figure 2.14, but we have much more data available. We might expect the reconstruction of x_t to be more accurate now that additional data is available. However, our analysis of the singular values makes us predict $x^\dagger$ to be a worse estimate of x_t than in the $n_d = 11$ case.

The singular vectors of A in this case are similar to the eigenvectors of F shown in Figures 2.11 and 2.12. The singular vectors of A corresponding to large singular values oscillate slowly, while the smaller singular vectors oscillate rapidly. I intuitively expect that white noise e, which

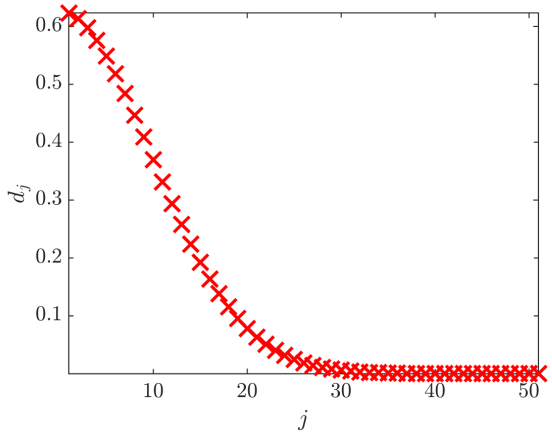


Figure 2.18: The singular values of A with $n_m = 51$

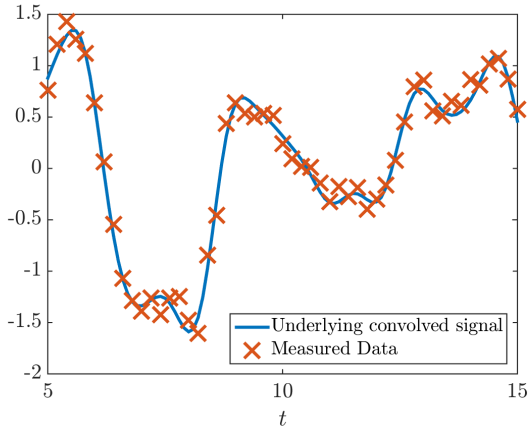


Figure 2.19: The convolved signal and the measured data

has zero correlation length and therefore oscillates rapidly, will project predominantly onto the smaller singular vectors of A. This can be expressed this mathematically as

$$e = \sum_{j=1}^{51} b_j v_j \tag{2.54}$$

$$\approx \sum_{j\approx 40}^{51} b_j v_j \tag{2.55}$$

where $b_j \in \mathbb{R}$ is some projection coefficient and $v_j = V(:, j)$ in $A = UDV^T$, the SVD of A. We found the above approximation was valid for this operator A and white noise e.

The pseudoinverse solution of the noisy least squares problem can be expanded as

$$x^\dagger = A^\dagger d = A^\dagger(y + e) = A^\dagger y + A^\dagger e \tag{2.56}$$

$$\approx A^\dagger y + \sum_{j\approx 40}^{51} \frac{1}{d_j} b_j v_j \tag{2.57}$$

where I note that $\frac{1}{\sigma_j}$ is very large and b_j is not trivially small for $j > 40$. Another concern is that estimation of the smaller singular values and corresponding singular vectors is susceptible to numerical error. These terms are amplified by taking $\frac{1}{d_j}$. These terms may also only be artefacts of the modelling of $\mathbf{A}$.

I therefore expect the pseudoinverse estimate $x^\dagger$ will likely be highly sensitive to noise and give physically meaningless estimates. It should be noted that this happens specifically when the noise e has nontrivial projection onto left singular vectors of A corresponding to near zero singular values. This is typically the case when A promotes smoothness and e has little/no cross correlation as in white noise.

As with the $n_m = 11$ case, I will calculate the minimum norm/standard least squares solution $x^\dagger \in X_{\text{ls}} = A^\dagger d$ and $Ax^\dagger = A(A^\dagger d)$. As expected, the least squares solution fits the data well, as shown in Figure 2.21. The estimate $x^\dagger$ of x_t is plotted alongside x_t in Figure 2.20. Note the scale.

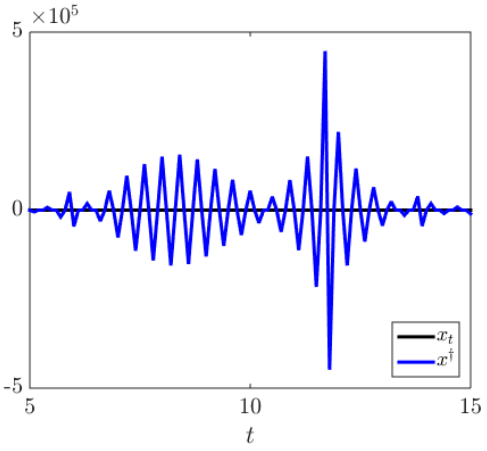


Figure 2.20: The pseudoinverse estimate $x^\dagger$ alongside the ground truth x_t

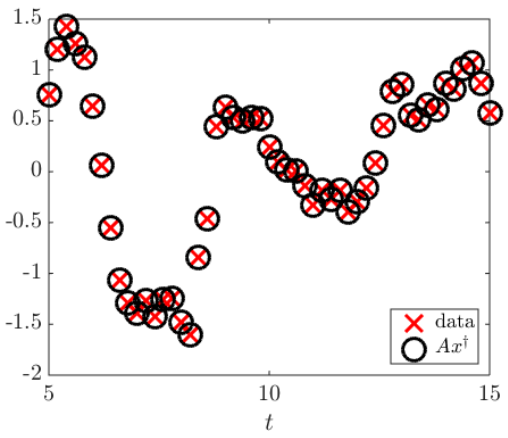


Figure 2.21: The measured data alongside the model predictions $Ax^\dagger$

2.4 Some Notes on Computational Complexity

This thesis largely investigates reducing computational complexity. We review some concepts of computational complexity now. More in depth reviews can be found in [32, 46].

A *flop* is a *floating point operation*. For example, the computations $1 + 2$, $1 - 2$, 1×2 and $\frac{1}{2}$ all require one flop.

In this section, let $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{n \times p}$ be arbitrary matrices, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}^p$ be arbitrary vectors, and $a \in \mathbb{R}$ be an arbitrary constant.

The scalar-vector product ab requires n flops. The matrix-vector product Bb requires $m(2n - 1)$ flops. Alternatively we say Bb requires $\mathcal{O}(mn)$ flops. The matrix-matrix product BC requires $mp(2n - 1)$, or $\mathcal{O}(mnp)$, flops.

Note that flop counts can be reduced when the relevant quantities have specific structure. For example, $I_n b = b$ is computed in either $2n^2 - n$, n or 0 flops. In this thesis, flop counts are for arbitrary full matrices using standard algorithms unless specifically stated otherwise.

If $n = m$ and B^{-1} exists, computing B^{-1} requires $\mathcal{O}(n^3)$ flops. Computing the SVD of B requires $\mathcal{O}(mn^2 + n^3)$ flops. More details on the derivation of these flop counts is found in [32].

The computational cost of a certain task is more than just the flop count. A flop count is a useful theoretical measure of computational complexity, but in practice the computational cost depends on the structure of the task and the computational platform. For example, a method of completing a task that uses fewer flops but much more memory may take more time to complete the task. In this thesis, the benchmark of computational complexity of a task is how long the task takes to complete on my 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive.

Chapter 3

Regularisation

Chapter 2 briefly reviewed the concept of inverse problems. Section 2.1 considered finite dimensional linear additive noise models of the form

$$d = Ax + e \tag{3.1}$$

where $d \in \mathbb{R}^{n_d}$ is the data, $A \in \mathbb{R}^{n_d \times n_x}$ is the forward operator, $x \in \mathbb{R}^{n_x}$ is the unknown of interest and e is additive noise/unknowns. Let

$$d_t = Ax_t + e_t \tag{3.2}$$

where x_t is the "ground truth", e_t the "true" error and d_t is the corresponding data. Chapter 2 discussed the pseudoinverse solution to the noisy least squares problem as an approximation of x_t from data d_t i.e.

$$x_t \approx x^\dagger = A^\dagger d_t \tag{3.3}$$

where $A^\dagger$ is the pseudoinverse of A. In Section 2.3, $x^\dagger$ was a poor approximation of x_t.

In this thesis, the field of constructing estimates x_{reg} of x_t from data d is referred to as *regularisation*. This definition is widely used in inverse problem literature [2, 3, 4, 41] although we note "regularisation" is a term used in other contexts as well.

In this thesis, regularisation methods are used with the aim of estimates x_{reg} being *representative* of the ground truth x_t. The estimate x_1 is said to be more representative than x_2 if

$$\|x_t - x_1\|_2 < \|x_t - x_2\|_2 \tag{3.4}$$

unless explicitly stated otherwise. It is often the case that x_t is a finite dimensional approximation to the infinite dimensional ground truth $\mathbf{x}_t$. Whether x_t is representative of $\mathbf{x}_t$ is problem specific e.g. is x_t of high enough resolution.

Consider the matrix $B \in \mathbb{R}^{n \times m}$. In this thesis, replacing B with $B \approx C$ with $\kappa_C < \kappa_B$ is called *regularising* B. C is called the *regularised form* of B.

This Chapter reviews several different regularisation techniques, and demonstrates how they work with the deconvolution example of Chapter 2. These techniques are provided for background, and will be referenced to throughout this thesis. Further discussion of regularisation can be found in [2, 3, 4, 32, 47].

3.1 Discretisation Methods

Some early approaches to regularisation involve reducing the number of unknowns n_x or the amount of data n_d. We will discuss methods of this type first, as a simple introduction to the field of regularisation.

Consider the model

$$\bar{d}_t = \mathbf{A}_e(\mathbf{x}_t, \mathbf{e}_t) \tag{3.5}$$

where $\bar{d} \in \mathbf{R}^{n_{\bar{d}}}$ is the data, $\mathbf{A}_e$ is an operator, $\mathbf{x}_t$ is the ground truth and $\mathbf{e}_t$ is additional unknowns and errors. Let a finite dimensional approximation be

$$\bar{d}_t = \bar{A}_e(\bar{x}_t, \bar{e}_t) \tag{3.6}$$

where $x_t \in \mathbb{R}^{n_{\bar{x}}}$ is a finite dimensional approximation of $\mathbf{x}_t$, $\bar{A}$ is a finite dimensional approximation of $\mathbf{A}_e$ and $\bar{e}_t$ is a finite dimensional approximation of $\mathbf{e}_t$. The aim is to estimate $\bar{x}_t$ from $\bar{d}_t$.

3.1.1 Regularisation by Discretisation

Suppose the operator $\bar{A}_e$ is ill-conditioned. We could instead consider the model

$$\bar{d} \approx A_{x,\bar{e}}(x, \bar{e}) \tag{3.7}$$

where $x \in \mathbb{R}^{n_x}$ with $n_x < n_{\bar{x}}$. We refer to Equation (3.7) as the *coarse* model, and Equation (3.6) as the *fine* model. The hope is that the estimate $x_{\mathrm{reg,x}} \in \mathbb{R}^{n_x}$ found from $\bar{d}_t$ using the coarse model $A_{x,\bar{e}}$ is more representative of $\mathbf{x}_t$ than the estimate $\bar{x}_{\mathrm{reg}} \in \mathbb{R}^{n_{\bar{x}}}$ found using the fine model $\bar{A}_{\bar{e}}$. Substituting $\bar{A}_{\bar{e}}$ and $\bar{x}$ for $A_{x,\bar{e}}$ and x with the aim of finding more representative estimates is called *regularisation by discretisation* or *regularisation by projection* [4].

The effectiveness of regularisation by discretisation is intuitively clear by considering the limiting case $n_x = 1$. We consider the linear additive noise case

$$\bar{d} \approx A_{x,\bar{e}}(x, \bar{e}) \tag{3.8}$$

$$= A_x x + \bar{e} \tag{3.9}$$

where A_x is now a $n_d \times 1$ "matrix", with $\kappa_A = \frac{d_1}{d_r} = \frac{||A||_2}{||A||_2} = 1$. Let the estimate $x_{\text{reg,x}}$ be the pseudoinverse estimate $x^\dagger$. This is found as

$$x_{\text{reg,x}} = x^\dagger \tag{3.10}$$

$$= A_x^\dagger d = \frac{1}{\|A_x\|^2} A_x^T d \tag{3.11}$$

and is hopefully representative of $\mathbf{x}_t$.

Recall the deconvolution example of Section 2.3. Choosing low values of n_x causes pseudoinverse estimates $x^\dagger$ to stay approximately near 0. Such estimates are nearer x_t than, for example, the $n_d = 51$, $n_x = 101$ case presented.

Regularisation by discretisation has clear limitations. A finite dimensional approximation x may be too coarse a approximation of $\mathbf{x}$ to be useful i.e. even knowing x_t exactly would not be useful. There is also no formal guarantee that the estimate is representative i.e. no guarantee that $x_{\text{reg,x}} \approx x_t = \mathbf{P}_{x,\mathbf{x}}\mathbf{x}_t$ where $\mathbf{P}_{x,\mathbf{x}}$ projects $\mathbf{x}_t$ to x_t.

3.1.2 Regularisation by Measurement Truncation

Suppose the operator $\bar{A}_e$ is ill-conditioned. We could instead consider

$$d \approx A_{d,e}(\bar{x}, e) \tag{3.12}$$

where $d \in \mathbb{R}^{n_d}$ with $n_d < n_{\bar{d}}$ is the truncated data and e is the truncated error and additional unknowns. The hope is that the estimate $\bar{x}_{\text{reg,d}} \in \mathbb{R}^{n_{\bar{x}}}$ found from d using the coarser model $A_{d,e}$ is more representative of $\mathbf{x}_t$ than the estimate $\bar{x}_{\text{reg}} \in \mathbb{R}^{n_{\bar{x}}}$ found using the model $\bar{A}_{\bar{e}}$.

Substituting $\bar{d} \in \mathbb{R}^{n_{\bar{d}}}$ and $\bar{A}_{\bar{e}}$ for $d \in \mathbb{R}^{n_d}$ and $A_{d,e}$ corresponds to either altering the experimental design, or simply culling measurements after the fact. In this thesis, we call substituting $\bar{d}$ for d with the aim of finding a more representative estimate *regularisation by measurement truncation* to emphasise the similarity with regularisation by discretisation. Regularisation by measurement truncation is sometimes called *decimation* or *downsampling*. We use the name regularisation by measurement truncation to highlight the similarity to regularisation by discretisation.

An example of regularisation by measurement truncation is the deconvolution example in Chapter 2. The pseudoinverse estimate $x_{\bar{d}}^\dagger$ with $n_{\bar{d}} = 51$ is shown in Figure 2.20. The pseudoinverse estimate $x_d^\dagger$ with $n_d = 11$ is shown in Figure 2.16. The $x_d^\dagger$ estimate is clearly more representative of x_t.

We can explain how regularising a matrix A by measurement truncation or discretisation works by considering the SVD $A = UDV^T$. Measurement truncation effectively reduces the dimension of U, while regularisation by discretisation reduces the dimension of V. Both subsequently shrink the dimension of the rectangular matrix D, reducing the number of singular

values. As seen in Chapter 2, this *may* improve the condition number of A which may in turn result in more representative estimates.

Regularisation by discretisation and regularisation by measurement truncation both attempt to substitute $\bar{A}_{\bar{e}}$ with better conditioned operators $A_{x,\bar{e}}$ and $A_{d,e}$ respectively. In the linear case, this corresponds to reducing the number of rows or columns in $\bar{A}_{\bar{e}}$ respectively. Another approach is to remove *components* of the operator e.g. remove singular vectors of linear $\bar{A}_{\bar{e}}$. This is the fundamental idea behind regularisation by truncated series expansion.

3.2 Truncated Series Expansion

A wide variety of series expansions exist. We will primarily consider expansions of matrices, which is covered in [42] and in greater depth in [32, 45]. An example of a series expansion of a nonlinear operator is the Taylor series [48]. When the expansion being considered is of a stochastic process, a thorough discussion can be found in [49]. Expansions used in nonlinear inverse problems often make use of the celebrated Karhunen-Loève theorem [50, 51]. Particularly notable is the Proper Orthogonal Decomposition (POD) [52, 53, 54]. Note that the POD of a matrix is exactly the SVD. A particularly concise analysis is given in [52].

Consider the linear finite dimensional additive noise model

$$d = Ax + e = y + e \tag{3.13}$$

and recall that we can expand the forward operator A as

$$A = UDV^T = U_{r_t} D_{r_t} V_{r_t}^T \tag{3.14}$$

where r_t is the rank of A. The pseudoinverse estimate can be formed as

$$\begin{aligned} x^\dagger &= V_{r_t} D_{r_t}^{-1} U_{r_t}^T d && (3.15)\\ &= A^\dagger d && (3.16)\\ &= A^\dagger y + A^\dagger e. && (3.17) \end{aligned}$$

Note that $\left\|A^\dagger\right\|_2 = \frac{1}{d_{r_t}}$, where d_{r_t} is potentially very small. The $A^\dagger e$ term is therefore potentially large. Computation of smaller singular values is also susceptible to numerical error [32], leading to large error when constructing $A^\dagger$. A potential means of avoiding these issues is to instead consider the approximation

$$A \approx A_r = U_r D_r V_r^T \tag{3.18}$$

where $r < r_t$. We call A_r the *truncated SVD* (tSVD) of A. The smallest singular value is now d_r. Typically, r is chosen either to control the condition number $\frac{d_1}{d_r}$ or to ensure $d_r > \epsilon$, where

ϵ is some numerically motivated small number e.g. a multiple of machine epsilon. Sometimes r is chosen as the computational resources available only allow for r singular vectors to be computed.

It is worth noting that two subtly different approaches exist in this category. We can either create a truncated expansion of A, or of x. If we expand A, the solution x we give will necessarily be in terms of this expansion. On the other hand, an expansion of x means A will only interact with these components, so we have implicitly imposed a truncated expansion on (the right hand side of) A.

The tSVD is a particular example of a truncated series expansion. When the model is a matrix, the SVD is a natural expansion to use, as it is well understood and has a wealth of computational methods for its computation [32]. Furthermore,

$$A_r = \min_{B:\text{rank}(B)=r} \|A - B\|_2 \tag{3.19}$$

that is, the tSVD is the best (in matrix 2-norm) rank r approximation to A [42]. Other approximations to $A \in \mathbb{R}^{n\times m}$ can be written in the form

$$A \approx A_{s,r} = \sum_{j=1}^{r} a_j(\mathbf{b}_j\mathbf{c}_j^T) \tag{3.20}$$

where $a_j \in \mathbb{R}$, $\mathbf{b}_j \in \mathbb{R}^{n_d}$ and $\mathbf{c}_j \in \mathbb{R}^{n_x}$. We make use of bold face for vectors in this section to avoid confusion with scalars. Note that $A_{s,r}$ is still an r term expansion of A, with the tSVD expansion A_r being the (matrix 2-norm) "best" expansion. Another expansion could be the $\mathbf{b}_j$'s and $\mathbf{c}_j$'s are discretised sines and cosines. While this may not be the "best" r term expansion of A, only the scalar a_j's need to be computed. For example, suppose the matrix A is a 2D image. The JPEG compression of A uses discrete cosines for the $\mathbf{b}_j$'s and $\mathbf{c}_j$'s [55], meaning only the scalar a_j's need to be computed and stored in memory in order to reconstruct the image.

Now suppose A is some image processing technique, and $\mathbf{x}$ is a (vectorised) image. We might only want $\mathbf{x}$ to be used as a JPEG of certain quality, so we might impose that $\mathbf{x}$ be expressed in the JPEG (discrete sines and cosines) basis. That is, $\mathbf{x}$ is of the form

$$\mathbf{x} = \sum_{j=1}^{r} a_j\mathbf{b}_j \tag{3.21}$$

$$= B\mathbf{a} \tag{3.22}$$

where $a_j \in \mathbb{R}$, $\mathbf{a} = (a_1, a_2, \ldots, a_r) \in \mathbb{R}^r$, $\mathbf{b}_j \in \mathbb{R}^{n_x}$ and $B = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r) \in \mathbb{R}^{n_x\times r}$. Rather than considering the n_x dimensional problem of estimating $\mathbf{x}_t \in \mathbb{R}^{n_x}$, we consider the r dimensional problem of finding $\mathbf{a}_t \in \mathbb{R}^r$ such that $\mathbf{x}_t = B\mathbf{a}_t$. This is an example of the subtle difference between expanding A and expanding $\mathbf{x}$.

The approach of compressing $\mathbf{x}$ and/or $\mathbf{d}$ with an expansion not derived from A is similar to regularisation by discretisation and regularisation by measurement truncation. For example in regularisation by discretisation, the $n_{\bar{x}}$ dimensional problem of estimating $\bar{\mathbf{x}}_t$ is replaced with the n_x problem of estimating $\mathbf{x}_t$. By expressing $\mathbf{x}$ in a r term expansion, the n_x dimensional problem of estimating $\mathbf{x}_t \in \mathbb{R}^{n_x}$ is replaced with the r dimensional problem of finding $\mathbf{a}_t \in \mathbb{R}^r$. Note that regularisation by discretisation and truncated series expansion both reduce the "resolution" of the estimates, even though the number of elements in $\mathbf{x}$ remains the same when taking the truncated series expansion. This view of "resolution" will be used throughout this thesis.

It is worth noting that some methods apply a truncated series expansion implicitly. For example, some methods may convert data to the *frequency domain*, and apply a *filter*, [47, 56]. Converting data to the frequency domain is the same as expressing

$$\mathbf{d} = \sum_{j=1}^{n_d} \bar{a}_j \mathbf{b}_j \tag{3.23}$$

$$= B\bar{\mathbf{a}} \tag{3.24}$$

where $\bar{a}_j \in \mathbb{R}$, $\bar{\mathbf{a}} = (\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{n_x}) \in \mathbb{R}^{n_d}$ is the *frequency domain data* and $\mathbf{b}_j \in \mathbb{R}^{n_d}$ is a discrete sine/cosine such that $BB^T = B^TB = I_{n_d}$ where $B = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$. Let

$$\mathbf{a} = F(\bar{\mathbf{a}}) \tag{3.25}$$

where $\mathbf{a}$ is the *filtered* frequency domain data, and F is the filter.

Consider the *idealised low-pass* filter, also known as the *sinc* or *square* filter. For this filter, $F = I_{r,n_d}$ i.e.

$$\mathbf{a} = F(\bar{\mathbf{a}}) \tag{3.26}$$

$$= I_{r,n_d}\bar{\mathbf{a}} \tag{3.27}$$

where $r \in \mathbb{N}$ is the *cutoff wavenumber*. Further analysis is then done on $\mathbf{a}$. This is the same as expressing data $\mathbf{d}$ as a r term series expansion of sines and cosines. We will discuss more complex filters in Chapters 5 and 6.

Let us return to the $n_d = 51$ deconvolution example of Section 2.3.2. Take the tSVD expansion

$$A = UDV^T \tag{3.28}$$

$$\approx U_r D_r V_r^T = A_r \tag{3.29}$$

to form the r term tSVD pseudoinverse estimate

$$x_r^\dagger = A_r^\dagger d = V_r D_r^{-1} U_r^T d \tag{3.30}$$

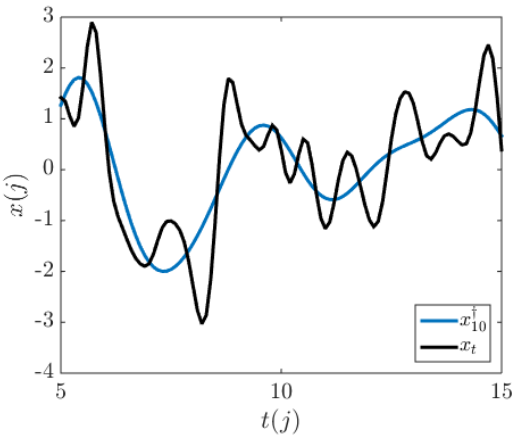


Figure 3.1: 10 term tSVD estimate $x_{10}^\dagger = A_{10}^\dagger d_t$ alongside x_t

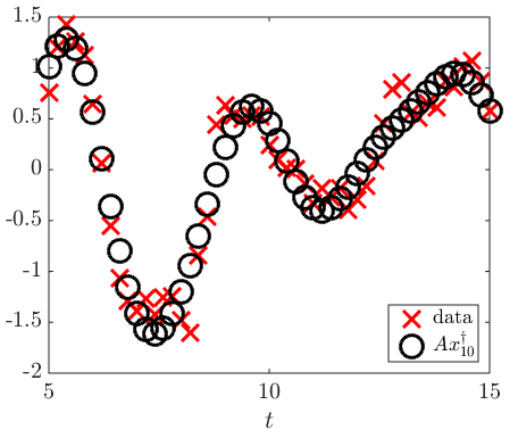


Figure 3.2: 10 term tSVD estimate model predictions $Ax_{10}^\dagger = AA_{10}^\dagger d_t$ alongside d_t

where we will consider $r = 10, 20$ and 30.

The estimate $x_{10}^\dagger$ found using a 10 term truncated SVD expansion of A is shown in Figure 3.1 alongside the ground truth x_t. Note the estimate is "overly smooth". This makes sense when we consider that the largest singular vectors of A were found in Chapter 2 to be smooth. The estimate $x_{10}^\dagger$ is a combination of "smooth" terms and is therefore smooth.

The quality of the fit to the data d of the 10 term tSVD estimate is shown in Figure 3.2. Note the visible disagreement between the data d and the predictions $Ax_{10}^\dagger = AA_r^\dagger d$.

Compare the 10 term tSVD estimate in Figure 3.1 and the fit to the data in Figure 3.2 to the (51 term) pseudoinverse estimate in Figure 2.20 and the fit to the data in Figure 2.21. Note that a better fit to the data d is not necessarily a better approximation to the ground truth x_t.

The estimate $x_{20}^\dagger$ found using a 20 term tSVD expansion of A is shown in Figure 3.3. Note that this better captures the "less smooth" components of the x_t. This is because $x_{20}^\dagger$ is constructed with some "less smooth" components that $x_{10}^\dagger$ lacked. We also plot the predictions $Ax_{20}^\dagger = AA_{20}^\dagger d$ with the data d in Figure 3.4. Note that $Ax_{20}^\dagger$ is closer to d than $Ax_{10}^\dagger$.

The estimate $x_{tsvd,30}$ found using a 30 term tSVD expansion of A is shown in Figure 3.5. Note that this solution is now "overly nonsmooth". This can be understood by expanding

$$x_{30}^\dagger = A_{30}^\dagger d \tag{3.31}$$

$$= A_{20}^\dagger d + \sum_{j=21}^{30} \frac{1}{d_j} v_j (u_j^T y) + \sum_{j=21}^{30} \frac{1}{d_j} v_j (u_j^T e) \tag{3.32}$$

and noting that y is smooth while u_j and e are nonsmooth for $j > 20$. Therefore $u_j^T y$ will

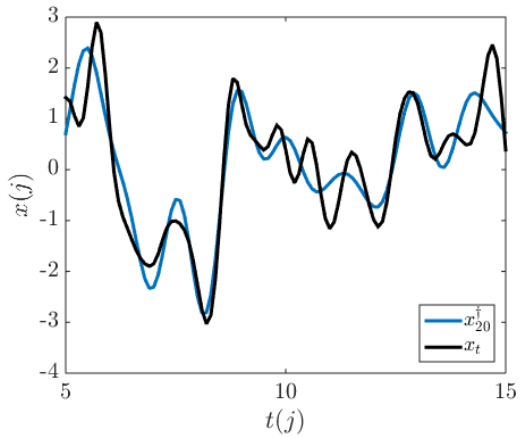


Figure 3.3: 20 term tSVD estimate $x_{20}^\dagger = A_{20}^\dagger d_t$ alongside x_t

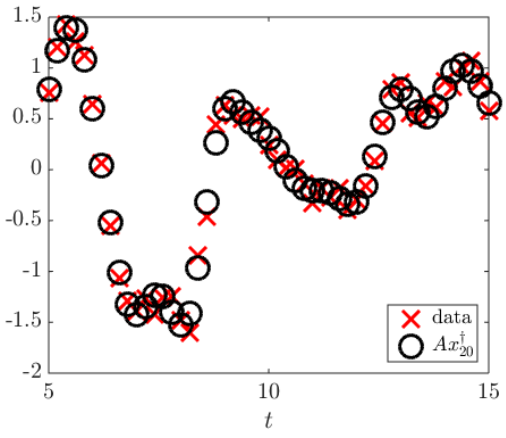


Figure 3.4: 20 term tSVD estimate model predictions $Ax_{20}^\dagger = AA_{20}^\dagger d_t$ alongside d_t

likely be small while $u_j^T e$ will likely be large, and then amplified by $\frac{1}{d_j}$ which is also large for $j > 20$.

The predictions $Ax_{30}^\dagger = AA_{30}^\dagger d$ are plotted in Figure 3.6 alongside the data d. Note that the fit to the data is closer than in the 10 term and 20 term cases shown in figures 3.2 and 3.4 respectively. This is because we are converging to the (r=51) pseudoinverse estimate $A^\dagger d$ shown in Figure 2.20 which fits the data as shown in Figure 2.21.

The above discussion has focussed on taking the truncated expansion induced by the model A as found by the SVD. However, as discussed earlier, we can also create a truncated expansion by directly imposing conditions on x and/or d, for example that they only consist of a few discrete sines and cosines.

Let

$$P_{f,r} = \begin{pmatrix} \mathbf{1} & cos\left(\frac{\pi(t-5)}{10}\right) & sin\left(\frac{\pi(t-5)}{10}\right) & cos\left(\frac{2\pi(t-5)}{10}\right) & sin\left(\frac{2\pi(t-5)}{10}\right) & \dots \end{pmatrix} \in \mathbb{R}^{n_x \times r} \tag{3.33}$$

which is used to compute $a_{f,r} = P_{f,r}x$, the r term sine and cosine approximation of x. We can return the basis of x as $x \approx x_{f,r} = P_{f,r}^T a_{f,r}$.

Let $r = 20$. The "best" approximation of x_t that can found using this reduced basis is $x_{f,20,t} = P_{f,20}^T a_{f,20,t} = P_{f,20}^T P_{f,20} x_t$. This is shown in Figure 3.7. Note that by expressing x in the reduced basis, the nonsmooth terms that seemed to lead to unreliable estimates are removed. The effective resolution has also been reduced.

The inverse problem is to estimate $a_{f,20,t}$ from data d. This effectively substitutes $A \in \mathbb{R}^{n_d \times n_x}$ with $\tilde{A}_{f,20} = AP_{f,20} \in \mathbb{R}^{n_d \times 20}$, reinforcing the similarity with regularisation by

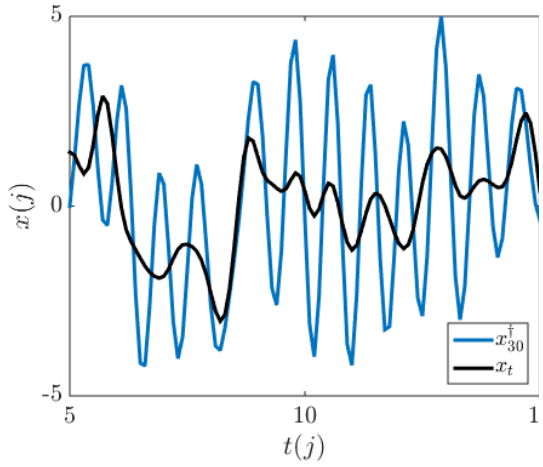


Figure 3.5: 30 term tSVD estimate $x_{30}^{\dagger} = A_{30}^{\dagger} d_t$ alongside x_t

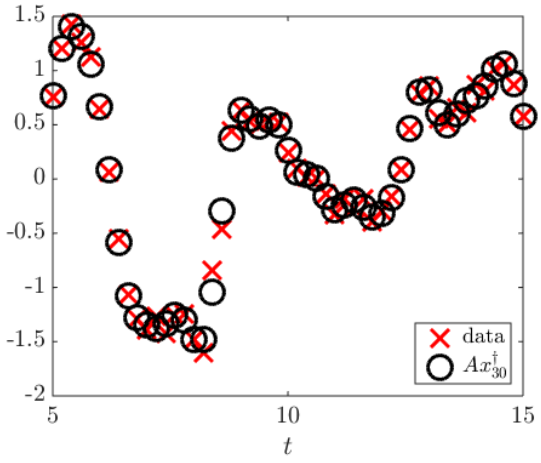


Figure 3.6: 30 term tSVD estimate model predictions $Ax_{30}^{\dagger} = AA_{30}^{\dagger} d_t$ alongside d_t

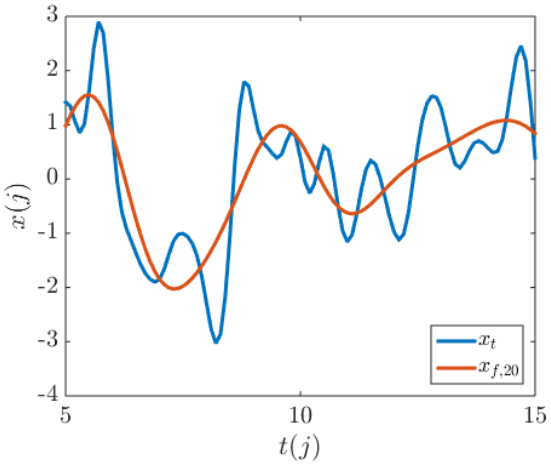


Figure 3.7: The 20 term sine/cosine expansion $x_{f,20}$ of x_t alongside x_t

discretisation. We can perform an analogue to regularisation by measurement truncation by imposing similar conditions on d.

Figure 3.8 shows the pseudoinverse estimate found by taking a 20 term sine and cosine expansion of x, found as $x_{f,20}^{\dagger} = \tilde{A}_{f,20}^{\dagger} d = (AP_{f,20})^{\dagger} d$. The estimate performs fairly well and is worst near the boundary. This is because we have assumed x to be zero outside the domain when this expansion was constructed. This results in $x_{f,20}^{\dagger}$ "overcompensating" near

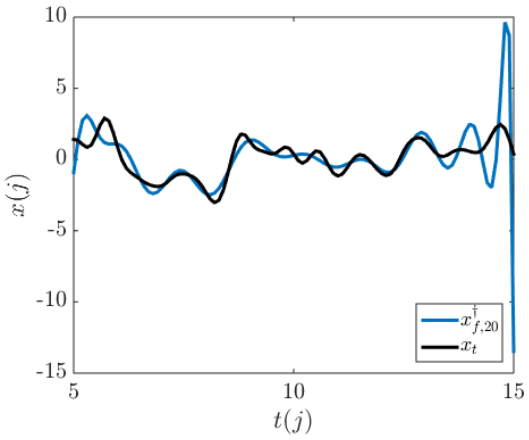


Figure 3.8: The 20 term sine/cosine basis pseudoinverse estimate $x^\dagger_{f,20} = \tilde{A}^\dagger_{f,20} d_t$ alongside x_t

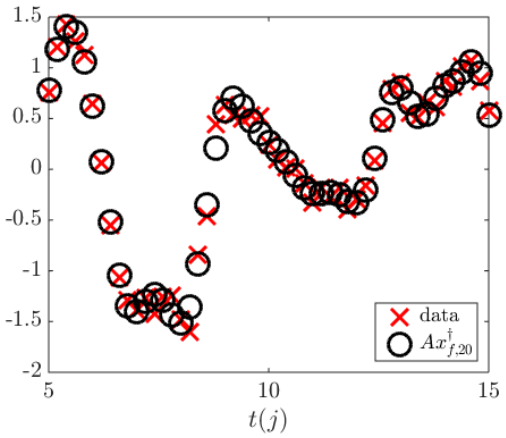


Figure 3.9: The $x^\dagger_{f,20}$ estimate predictions $Ax^\dagger_{f,20}$ alongside the data d_t

the boundary in order to have $d \approx Ax^\dagger_{f,20}$. We wanted to illustrate this is a potential pitfall of the method. This effect can be mitigated by adjusting the basis. Different types of expansions are discussed in [49].

3.3 Truncated Iterative Methods

Recall that least squares estimates $x_{\text{ls}} \in X_{\text{ls}}$ of the ground truth x_t satisfy

$$x_{\text{ls}} = \min_x \{\|A(x) - d\|_2\} \tag{3.34}$$

where A is the forward operator and d is the data. Suppose we find the minimum with an iterative method i.e. construct a sequence x_j such that

$$\|A(x_j) - d\|_2 \leq \|A(x_k) - d\|_2 \tag{3.35}$$

for $j > k$. However, it may be found that iterate r, that is x_r, is a better estimate of x_t i.e.

$$\|x_r - x_t\|_2 \leq \|x_j - x_t\|_2 \tag{3.36}$$

for $j \neq r$. Truncating the iterative method at the r'th step in order to obtain better estimates of x_t is called *regularisation by truncated iterations*.

Consider the deconvolution problem of Section 2.3.2. The minimisation $\min_x ||A(x) - d||_2$ is performed by Landweber iterations [57] as

$$x_{j+1} = x_j + \beta(A^T d - A^T A x_j) \tag{3.37}$$

where $\beta \in \mathbb{R}$ is sometimes called the *stepsize*. It should be noted that the Landweber method for ill-posed $A(x) = d$ semi-converges, approaching a regularised solution in early iterates before becoming unstable [58, 59, 60].

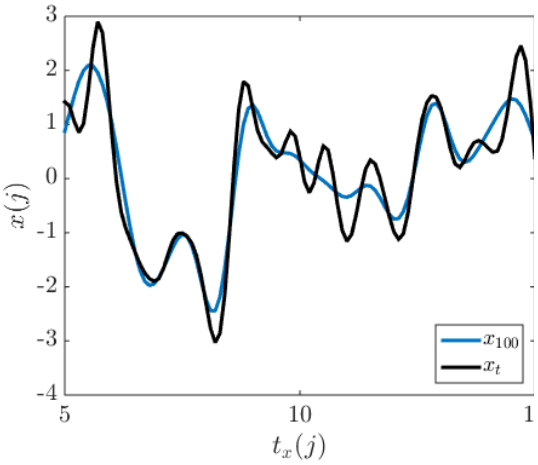


Figure 3.10: The 100'th iterate x_5

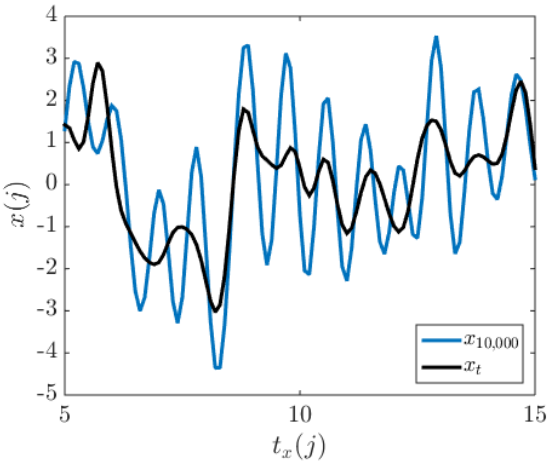


Figure 3.11: The 10,000'th iterate x_{20}

For this problem a relatively small stepsize and relatively large number of iterates are used to demonstrate the method, whereas typically a stopping criteria and adaptive stepsize would be used [60]. The 100'th iterate x_{100} is shown in Figure 3.10 alongside the ground truth x_t. The 10,000'th iterate $x_{10,000}$ is shown in Figure 3.11 alongside the ground truth x_t. In this case, the truncated iterate estimate is closer to the ground truth.

3.4 Tikhonov Regularisation

The final regularisation method we will discuss is Tikhonov Regularisation [41, 61]. Our analysis largely follows [4]. Recall that the least squares problem, finding $x_{\text{ls}} = \min_x\{\|A(x) - d\|_2^2\}$ is potentially neither unique or stable. Consider instead the *Tikhonov regularised problem*, finding

$$x_{\text{tik}} = \min_x\{||A(x) - d||_2^2 + \alpha||x||_2^2\} \tag{3.38}$$

where x_{tik} is the *Tikhonov estimate* of x_t, $\alpha > 0$ is the *regularisation parameter*, and $\|x\|_2^2$ is the *regularisation functional*. The addition of the regularisation functional will work to counteract the overly large solutions we found by least squares, for example in Figure 2.20 from the deconvolution problem of Chapter 2.

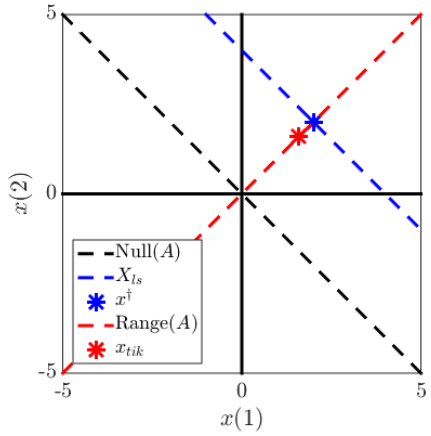


Figure 3.12: Comparing Tikhonov to Least Squares in $\mathbb{R}^2$

Consider estimating x_t in the equation $Ax_t = y$ with $n_x = n_y = 2$, and A has rank 1. The elements of the nullspace of A form a line in $\mathbb{R}^2$, and the elements x_{ls} of $X_{\text{ls}} = \min_x\{\|Ax - y\|_2\}$ form a parallel line. The element of X_{ls} nearest the origin is $x^\dagger = A^\dagger y$. The elements of the range of A form a line in $\mathbb{R}^2$ through the origin and $x^\dagger$. The Tikhonov solution $x_{\text{tik}} = \min_x\{\|Ax - y\|_2^2 + \alpha \|x\|_2^2\}$ lies somewhere between the origin and $x^\dagger$, depending on the value of α. This is shown graphically in Figure 3.12.

The Tikhonov estimate for linear inverse problems can be rewritten as

$$x_{\text{tik}} = \min_x \left\{ \left\| \hat{A}x - \hat{d} \right\|_2^2 \right\} \tag{3.39}$$

$$= \hat{A}^\dagger \hat{d} \tag{3.40}$$

where $\hat{A} = \begin{pmatrix} A \\ \sqrt{\alpha} I_{n_x} \end{pmatrix} \in \mathbb{R}^{(n_d+n_x)\times n_x}$ and $\hat{d} = \begin{pmatrix} d \\ 0_{n_x,1} \end{pmatrix} \in \mathbb{R}^{n_d+n_x}$. Note that $\hat{A}$ is full rank, so the Tikhonov solution is unique. This reformulation allows us to use the same numerical methods to find x_{tik} as x_{ls}.

Let us now consider Tikhonov regularisation in terms of the SVD. We can rewrite the Tikhonov estimate similarly to the normal equations of least squares as

$$(A^T A + \alpha I) x_{\text{tik},\alpha} = A^T d \tag{3.41}$$

$$x_{\text{tik},\alpha} = (A^T A + \alpha I)^{-1} A^T d \tag{3.42}$$

where the additional subscript in $x_{\text{tik},\alpha}$ is to highlight that this is a particular solution obtained for a specific α. Note that the matrix $A^T A + \alpha I$ will have eigenvalues $\lambda_k \geq \alpha > 0$ for all k, implying that $\left\|(A^T A + \alpha I)^{-1}\right\|_2 \leq \frac{1}{\alpha}$, controlling the norm of the estimates x_{tik}.

Let $A = UDV^T$ be the SVD of A. This can be expressed as $A = \sum_{j=1}^{r} d_j u_j v_j^T$. The Tikhonov estimate can be written as

$$x_{\text{tik},\alpha} = \sum_{j=1}^{r} \frac{d_j}{d_j^2 + \alpha}(d^T u_j)v_j + \sum_{j=r+1}^{\min\{n_x,n_d\}} \frac{1}{\alpha}(d^T u_j)v_j \tag{3.43}$$

where we note that $(d^T u_k) \in \mathbb{R}$.

We now define the *filter factors* $q_j \in \mathbb{R}$ of different regularisation methods. We express estimates x_{reg} of x_t as

$$x_{\text{reg}} = \sum_{j=1}^{\min\{n_x,n_y\}} q_j(d^T u_j)v_j \tag{3.44}$$

to highlight the differences in the estimation schemes. The pseudoinverse estimate $x^\dagger = A^\dagger d$ has $q_j = \frac{1}{d_j}$ for $j \leq r_t$ and $q_j = 0$ for $j > r_t$. The r term tSVD estimate $x_r^\dagger$ has $q_j = \frac{1}{d_j}$ for $j \leq r$ and $q_j = 0$ for $j > r$. The Tikhonov estimate x_{tik} has $q_j = \frac{d_j}{d_j^2+\alpha^2}$ for $j \leq r_t$ and $q_j = \frac{1}{\alpha}$ for $j > r_t$. Note that x_{tik} is composed of $\min\{n_x, n_d\}$ components of the SVD of A, while $x^\dagger$ and $x_r^\dagger$ only make use of the first $r_t < \min\{n_x, n_d\}$ and $r \leq r_t \leq \min\{n_x, n_d\}$ components respectively.

There remains the question as to how the regularisation parameter α should be chosen. We present 2 commonly used methods.

3.4.1 Morozov Discrepancy Principle

Let

$$d = Ax_t + e_t \tag{3.45}$$

where $d \in \mathbb{R}^{n_d}$ is measured data, $A \in \mathbb{R}^{n_d \times n_x}$ is the linear finite dimensional forward operator, x_t is the ground truth and e_t is the "true" noise. The Tikhonov estimate to x_t is

$$x_{\text{tik},\alpha} = \min_x \{\|Ax - d\|_2^2 + \alpha \|x\|_2^2\}. \tag{3.46}$$

for a particular choice of $\alpha \in \mathbb{R}$.

The core theory behind the Morozov discrepancy principle is that the solution should not have residual $\zeta(\alpha) = \|Ax_{\text{tik},\alpha} - d\|_2$ less that the noise level $\sigma = ||e_t||_2$. Note that this method requires knowledge of σ, and that $\sigma \neq \sigma_e$, the standard deviation of white noise e. σ can be estimated from e.g. measurements on known $x_t = 0_{n_x,1}$.

A notable drawback of estimating α by the Morozov discrepancy principle is that multiple inversions need to be performed, each with different values of α, in order to evaluate $\zeta(\alpha) =$

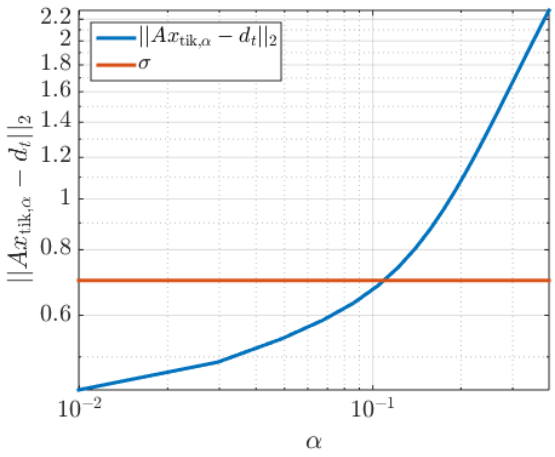


Figure 3.13: Evaluating the residual $\zeta(\alpha) = \|Ax_{\text{tik},\alpha} - d\|_2$ for a range of α's against $\sigma = \|e_t\|_2$.

$\|Ax_{\text{tik},\alpha} - d\|_2$. This is a problem in situations where inversions are required rapidly. Typically a value of α is obtained for a few test cases, and then fixed for subsequent inversions.

I performed Tikhonov regularisation on the deconvolution example of Section 2.3.2 using the Morozov discrepancy principle to pick α. I first evaluate $x_{tik,\alpha} = \min_x\{\|Ax - d\|_2^2 + \alpha \|x\|_2^2\}$ for a range of α's, and then choose α such that $\zeta(\alpha) = \|Ax_{\text{tik},\alpha} - d\|_2 \approx \sigma$. This is shown graphically in Figure 3.13. By Morozov discrepancy, $\alpha \approx 0.1$ for this problem.

The Tikhonov estimate found with the Morozov discrepancy principle $x_{\text{tik},0.1}$ and the ground truth x_t are shown in Figure 3.14. The estimate fits quite well, if slightly "overly smooth". Given the nature of the forward operator and the data, it would be optimisitc to expect a better estimate. The predictions $Ax_{\text{tik},0.1}$ are plotted along with the data in Figure 3.15. Compare the difference between $Ax_{\text{tik},0.1}$ and d with the difference between y and d in Figure 2.19. The size of the deviations appear comparable, reflecting that the Tikhonov solution with Morozov fits data d up to σ, and the data is within around σ of y.

Note that the estimate $x_{\text{tik},0.1}$ seems to stay closer to 0 than x_t. This is a result of $\|x\|_2$ being penalised in the construction of $x_{\text{tik},0.1}$. This effect could be mitigated by using a different penalty term. We discuss a different interpretation of penalty terms in Chapter 4 and present some different types of penalty term in Chapter 5.

3.4.2 L-Curve Criterion

Suppose we compute $\|Ax_{\text{tik},\alpha} - d\|_2$ and $\|x_{\text{tik},\alpha}\|_2$ for a range of α's. If we were to plot $\log(\|Ax_{\text{tik},\alpha} - d\|_2)$ against $\log(\|x_{\text{tik},\alpha}\|_2)$ for a range of α's, we typically see an "L" shaped

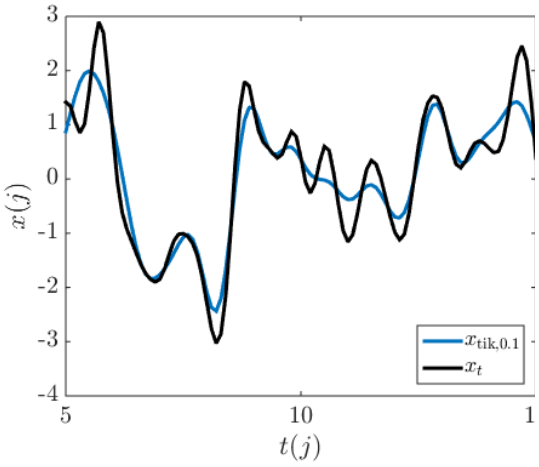


Figure 3.14: The Tikhonov estimate $x_{\text{tik},0.1}$ found using Morozov discrepancy principle alongside x_t

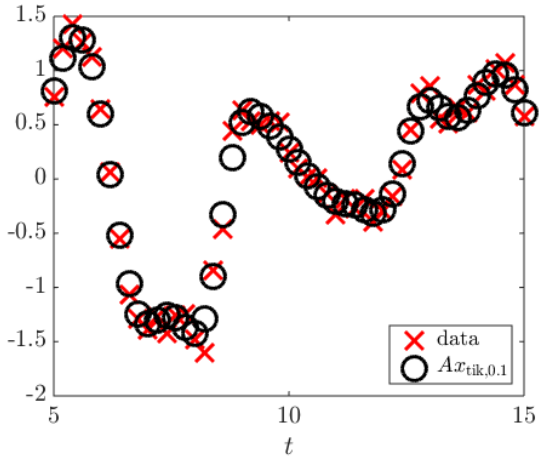


Figure 3.15: The Tikhonov Morozov estimate model predictions $Ax_{\text{tik},0.1}$ alongside data d_t

curve. The *L-Curve Criterion* is to take α slightly right of the bend in the "L" [2, 4]. Note that this reflects a balance between the norm of the solution and the residual, the two terms of the regularisation functional.

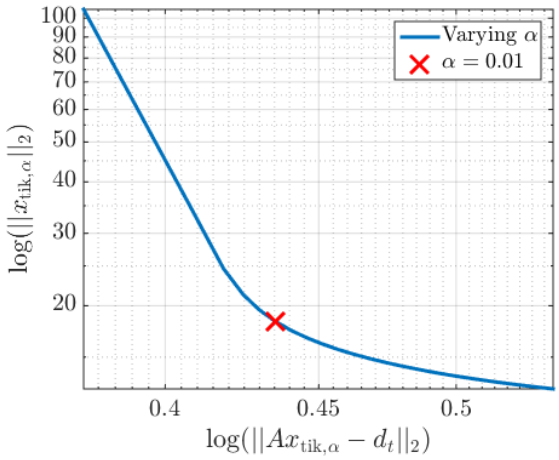


Figure 3.16: Evaluating the residual and norm of $x_{\text{tik},\alpha}$ for a range of α's to find the "L".

We compute an L-Curve plot for the deconvolution example of Section 2.3.2. The plot is

shown in Figure 3.16. The value of α corresponding to the red x on the curve was found to be around $\alpha \approx 0.01$. Note that this is much smaller than the value found by the Morozov criterion, $\alpha_{\text{morozov}} \approx 0.1$.

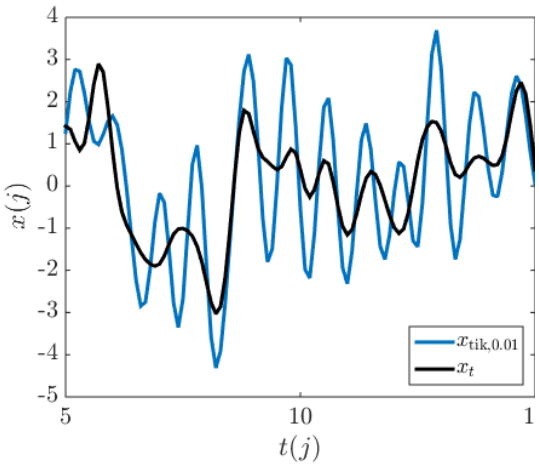


Figure 3.17: The Tikhonov estimate $x_{\text{tik},0.01}$ found using L-curve criterion alongside x_t

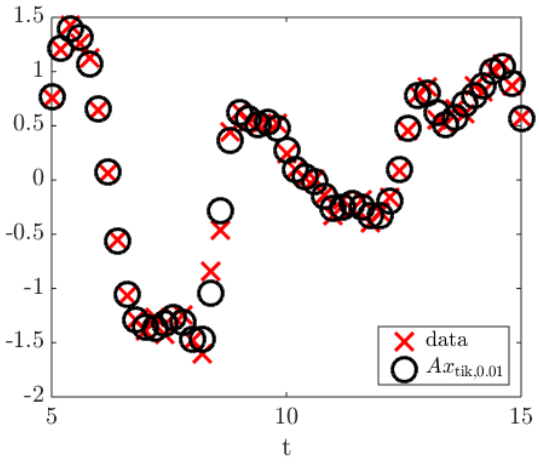


Figure 3.18: The Tikhonov L-curve estimate predictions $Ax_{\text{tik},0.01}$ alongside data d_t

The Tikhonov estimate found using the L-curve criterion $x_{\text{tik},0.01}$ is shown in Figure 3.17. Note that the nonsmooth terms in $x_{\text{tik},0.01}$ are overly large. These terms were lacking in the Morozov estimate $x_{\text{tik},0.1}$ shown in Figure 3.14, although the Morozov estimate appears better overall. The L-curve estimate appears more like the pseudoinverse estimate $x^\dagger$, unsurprising as in the limiting $\alpha = 0$ case $x_{\text{tik},0} = x^\dagger$. The model predictions $Ax_{\text{tik},0.01}$ are plotted alongside the data in Figure 3.18.

Compare the Morozov estimate $x_{\text{tik},0.1}$ of Figure 3.14, the L-curve estimate $x_{\text{tik},0.01}$ of Figure 3.17, and the minimum norm least squares/pseudoinverse estimate $x^\dagger$ of Figure 2.20, and the corresponding predictions shown in Figures 3.15 3.18 and 2.21. Observe that the respective estimates of x_t are worse, while the predictions fit the data better.

The estimate obtained using Morozov's discrepancy principle gives a larger value of α and an apparently better estimate than the L-curve criterion in this particular problem. It should be stressed that neither of these results are guaranteed. There is however an argument to be made as to why the Morozov solution should be expected to be "better". Recall that the Morozov principle incorporates the additional information $\sigma = ||e||_2$. The Morozov estimate makes use of additional information, so could be expected to perform better, although this is not guaranteed.

3.4.3 Generalised Tikhonov Regularisation

Recall that we stated the Tikhonov regularised solution as

$$x_{\text{tik}} = \min_x \{ \|A(x) - d\|_2^2 + \alpha \|x\|_2^2 \} \tag{3.47}$$

where α is the regularisation parameter and $\|x\|_2^2$ is the regularisation functional. A natural extension is the generalised Tikhonov solution, formed as

$$x_{\text{gtik}} = \min_x \left\{ B_1\big(A(x) - d\big) + \alpha B_2\big(x\big) \right\} \tag{3.48}$$

where B_1 and B_2 are functionals. The choices of each depends on both the forward problem A and the features we predict x_t will possess. We reinterpret this formulation in Chapter 4, and then give some examples of B_1 and B_2 in Chapter 5.

This concludes the discussion of regularisation in this thesis. The next Chapter reformulates inverse problems in terms of Bayesian inference. The Bayesian framework is then used to re-examine regularisation methods. These discussions will be referred to throughout the rest of this thesis, particularly in Chapter 6 where concepts of regularisation are utilised for the development of new methods.

Chapter 4

Bayesian Inversion

Another approach to inverse problems is to view them as Bayesian inference. Rather than saying data d implies a particular solution x, we instead consider the probability distribution of x given the data d. The "solution" in this framework is the *posterior distribution* $\pi(x|d)$.

We will largely follow the derivation and analysis of [4]. Derivations can also be found in [62, 63] for alternative viewpoints.

The Bayesian framework allows for intuitive analysis of regularisation methods. Explicit understanding of the assumptions in regularisation will be used throughout this thesis for the development and justification of new methods. Of particular note is the reinterpretation of the penalty functionals in generalised Tikhonov inversion as related to *likelihood* and *prior* probability distributions. This interpretation allows reasonable estimates of the penalty terms to be constructed without using computationally expensive techniques such as the Morozov discrepancy principle and the L-curve criterion. Another advantage of the Bayesian interpretation is it gives a natural way to quantify uncertainty in the inversion.

We now outline the derivation of the posterior distribution.

4.1 Construction of the Bayesian Solution

We wish to find the probability distribution $\pi_{x|d}(x|d)$, the *conditional distribution* of the unknowns of interest x given the measured data d. In the Bayesian setting, this distribution, known as the *posterior*, is considered the "solution" of our inverse problem. In this thesis $\pi(x|d)$ can be assumed to be the posterior distribution unless explicitly stated.

Consider the additive noise model

$$d = A(x) + e \tag{4.1}$$

with d the measured data, x the unknown of interest, A the forward model and e some additive noise. We define the *prior* probability distribution $\pi_x(x)$ on x. We also define a distribution

$\pi_e(e)$ on e. In this thesis, $\pi_x(x) = \pi(x)$ and $\pi_e(e) = \pi(e)$ unless explicitly stated. The prior probability distribution will often be referred to simply as the prior.

Consider the *joint probability distribution* of x and d

$$\pi_{x,d}(x, d) \tag{4.2}$$

where the *marginal probability distribution* of x given d defined as

$$\pi_x(x) = \int_{\mathbb{R}^{n_d}} \pi_{x,d}(x, d)d(d) \tag{4.3}$$

is the prior. Apologies for the awkward integration with respect to $d(d)$.

Suppose x is known, with $x = x_t$. The conditional distribution of d given $x = x_t$ is

$$\pi_{d|x}(d|x_t) = \frac{\pi_{x,d}(x_t, d)}{\pi_x(x_t)} \tag{4.4}$$

for $\pi_x(x_t) \neq 0$. The distribution $\pi_{d|x}(d|x)$ is called the *likelihood*.

Suppose d is known, with $d = d_t$. The conditional distribution of x given $d = d_t$, the posterior, is

$$\pi_{x|d}(x|d_t) = \frac{\pi_{x,d_t}(x, d_t)}{\pi_d(d_t)} \tag{4.5}$$

for $\pi_d(d_t) = \int_{\mathbb{R}^{n_x}} \pi_{x,d}(x, d_t)dx \neq 0$.

Consider the joint distribution

$$\pi_{d,x,e}(d, x, e) = \pi_{d,e|x}(d, e|x)\pi_x(x) = \pi_{d|x,e}(d|x, e)\pi_{e|x}(e|x)\pi_x(x) \tag{4.6}$$

where the equivalences can be derived in a similar fashion to the above. Note that

$$\pi_{d|x,e}(d|x, e) = \delta(d - A(x) - e) \tag{4.7}$$

as $d = A(x) + e$. We can then marginalise $\pi_{d,e|x}(d, e|x)$ over e as

$$\begin{aligned} \pi_{d|x}(d|x) &= \int_{\mathbb{R}^{n_d}} \pi_{d,e|x}(d, e|x)de \\ &= \int_{\mathbb{R}^{n_d}} \pi_{d|x,e}(d|x, e)\pi_{e|x}(e|x)de \\ &= \int_{\mathbb{R}^{n_d}} \delta(d - A(x) - e)\pi_{e|x}(e|x)de \\ &= \pi_{e|x}(d - A(x)|x) \end{aligned} \tag{4.8}$$

where we use the identity of 4.6. Equation (4.8) expresses the likelihood in terms of the conditional distribution of the noise.

Bayes theorem [64] states that

$$\pi_{x|d}(x|d_t) = \frac{\pi_{d|x}(d_t|x)\pi_x(x)}{\pi_d(d_t)} \tag{4.9}$$

however we observe that $\pi_d(d_t)$ is a constant and use the result of 4.8 to write

$$\pi_{x|d}(x|d_t) \propto \pi_{e|x}(d_t - A(x)|x)\pi_x(x) \tag{4.10}$$

so the posterior is proportional to the product of the prior and the likelihood.

If we assume $\pi(x, e) = \pi(x)\pi(e)$ i.e. x and e are mutually independent, then $\pi_{e|x} = \pi_e$ which simplifies the posterior to

$$\pi_{x|d}(x|d_t) \propto \pi_e(d_t - A(x))\pi_x(x) \tag{4.11}$$

i.e. the likelihood is in terms of the noise distribution only.

Let x and e be normally distributed. We denote this as $\pi_x(x) \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $\pi_e(e) \sim \mathcal{N}(\mu_e, \Gamma_e)$ with μ the mean and Γ the covariance. The normal distribution, also known as the Gaussian distribution, can be written as

$$\mathcal{N}(\mu_x, \Gamma_x) = \left((2\pi)^{n_x}|\Gamma_x|\right)^{\frac{-1}{2}} \exp\left(\frac{1}{2}(x-\mu_x)^T\Gamma_x^{-1}(x-\mu_x)\right) \tag{4.12}$$

$$= \left((2\pi)^{n_x}|\Gamma_x|\right)^{\frac{-1}{2}} \exp\left(\frac{1}{2}\left\|\tilde{L}_x(x-\mu_x)\right\|_2^2\right) \tag{4.13}$$

where $|\Gamma_x|$ is the trace of Γ_x and $\tilde{L}_x^T\tilde{L}_x = \Gamma_x^{-1}$. Note that this assumes $|\Gamma_x| \neq 0$ and Γ_x^{-1} exists.

The posterior can be expressed as

$$\pi_{x|d}(x|d) \propto \pi_e(d - A(x))\pi_x(x) \tag{4.14}$$

$$= \exp\left(-\frac{1}{2}\left(||\tilde{L}_e(d - A(x) - \mu_e)||_2^2 + ||\tilde{L}_x(x-\mu_x)||_2^2\right)\right) \tag{4.15}$$

$$= \left(-\frac{1}{2}\left(V(x)\right)\right) \tag{4.16}$$

with $\tilde{L}_e^T\tilde{L}_e = \Gamma_e^{-1}$, $\tilde{L}_x^T\tilde{L}_x = \Gamma_x^{-1}$ and $V(x) = ||\tilde{L}_e(d - A(x) - \mu_e)||_2^2 + ||\tilde{L}_x(x-\mu_x)||_2^2$, sometimes referred to as the *potential*. Note that this assumes Γ_x and Γ_e are invertible.

The above is an expression for the entire distribution $\pi_{x|d}(x|d)$. This can be used to define specific *point estimates* of x_t given $d_t = Ax_t + e_t$.The conditional mean (CM) is a common point estimate, defined as

$$x_{\text{CM}} = \mu_{x|d_t} = \mathbb{E}(x|d = d_t) = \int_{\mathbb{R}^{n_x}} x\pi_{x|d}(x|d_t)dx \tag{4.17}$$

where $\mathbb{E}$ is the expectation operator. Another common point estimate is the *Maximum a Posteriori* (MAP) estimate, defined as

$$x_{\text{MAP}} = \max_x \{\pi_{x|d}(x|d_t)\} \tag{4.18}$$

i.e. the x at the "most likely"/"highest point" of the posterior. For normally distributed x and e the MAP can be expressed as

$$x_{\text{MAP}} = \min_x \{V(x)\} = \min_x \left\{ ||\tilde{L}_e(d_t - A(x) - \mu_e)||_2^2 + ||\tilde{L}_x(x - \mu_x)||_2^2 \right\} \tag{4.19}$$

which shares obvious similarities with the Tikhonov estimate. In fact, the Tikhonov estimate

$$x_{\text{tik}} = \min_x \left\{ ||d_t - A(x)||_2^2 + \alpha||x||_2^2 \right\} \tag{4.20}$$

is equivalent to the MAP with $\mu_x = 0_{n_x,1}$, $\mu_e = 0_{n_d,1}$, $\Gamma_e = \sigma_e^2 I_{n_d}$ and $\Gamma_x = \sigma_x^2 I_{n_x}$ where $\sigma_e, \sigma_x \in \mathbb{R}$. Therefore, $\alpha = \frac{\sigma_e^2}{\sigma_x^2}$. This observation allows us to arrive at an intuitively understandable value for α, as we often have crude estimates of the variances σ_e^2 and σ_x^2 before attempting reconstructions. This concept will be elaborated on further when discussing prior models in Chapter 5.

This equivalence between the MAP and Tikhonov estimates is sometimes used as an argument for using the MAP over the CM as Tikhonov estimates are already widely used [2, 3, 41]. In the case of linear problems with normally distributed x and e, the MAP and CM coincide. An example of the MAP and CM not coinciding is shown in Figure 4.1. This is an example of a case where the MAP might be a more meaningful point estimate than the CM.

The MAP estimate for normally distributed x and e is from data $d_t = Ax_t + e_t$ is

$$\begin{aligned} x_{\text{MAP}} &= \min_x \{||\tilde{L}_e(d_t - Ax - \mu_e)||_2^2 + ||\tilde{L}_x(x - \mu_x)||_2^2\} && (4.21)\\ &= \min_x \left\{ \left\| \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix} x - \begin{pmatrix} \tilde{L}_e(d_t - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} \right\|_2^2 \right\} && (4.22)\\ &= \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_e(d_t - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} && (4.23)\\ &= \hat{A}^\dagger \hat{d}_t && (4.24) \end{aligned}$$

similarly to the Tikhonov estimate. Note that Null($\hat{A}$) = Null(A) $\cap$ Null($\tilde{L}_x$), so if Γ_x is full rank, the problem $\hat{d} = \hat{A}x$ is exactly and uniquely solvable for x.

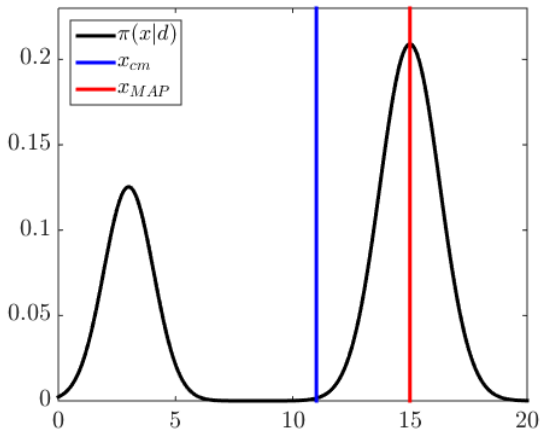


Figure 4.1: Comparing the Conditional Mean with the Maximum a Posteriori point approximations of a probability distribution

4.1.1 A More Generalised Posterior

Recall that, assuming additive noise, the posterior can be written as

$$\pi_{x|d}(x|d_t) \propto \pi_{e|x}(d_t - A(x)|x)\pi_x(x). \tag{4.25}$$

The earlier derivation assumed e and x to be mutually independent and normally distributed. Instead, let x and e be mutually dependant but $\pi_{e|x}$ is normal i.e.

$$\pi_{e|x}(d_t - A(x)|x) \sim \mathcal{N}(\mu_{e|x}, \Gamma_{e|x}) \tag{4.26}$$

with $\mu_{e|x}$ the conditional mean of e given x, and $\Gamma_{e|x}$ the conditional covariance of e given x.

Let $\pi(x)$ belong to the family of *exponential priors*, defined as

$$\pi(x) \propto \exp\left(-\frac{1}{2}G(x)\right) \tag{4.27}$$

of which normal distributions are a member.

The posterior distribution in the above more general case is

$$\pi_{x|d}(x|d) \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_{e|x}\big(d - A(x) - \mu_{e|x}\big)\right\|_2^2 + G(x)\right)\right) \tag{4.28}$$

with potential

$$V(x) = \left\|\tilde{L}_{e|x}\big(d - A(x) - \mu_{e|x}\big)\right\|_2^2 + G(x). \tag{4.29}$$

This more general form of the posterior will be used repeatedly in this thesis. Note again the similarities between the MAP estimate $x_{\text{MAP}} = \min\{V(x)\}$ and the generalised Tikhonov estimate.

4.2 Regularisation Methods in the Bayesian Framework

Often a hard line is drawn between regularisation and the Bayesian approach to inverse problems. However, many regularisation methods can be interpreted in the Bayesian setting. These considerations will be useful later as we develop hybrid approaches in Chapters 6 and 7.

4.2.1 Least Squares

Consider the finite dimensional additive noise model

$$d = Ax + e \tag{4.30}$$

with a specific realisation

$$d_t = Ax_t + e_t \tag{4.31}$$

where we wish to estimate x_t from d_t.

Consider the least squares estimate

$$x_{\text{ls}} = \min_x \left\{ \|Ax - d_t\|_2^2 \right\} \tag{4.32}$$

which we note is not necessarily unique. The MAP estimate assuming mutually independent normally distributed x and e is

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_e\big(d - Ax - \mu_e\big) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\}. \tag{4.33}$$

Note that the MAP and LS estimates are equivalent if $\tilde{L}_e \propto I_{n_d}$, $\mu_e = 0_{n_d,1}$ and $\tilde{L}_x = 0_{n_x,n_x}$. In other words, the least squares estimate assumes e to be zero mean white noise. The least squares estimate also assumes $\Gamma_x^{-1} = 0_{n_x,n_x}$ i.e. no prior information about x. Another interpretation is that least squares is a *maximum likelihood* estimate, as we are only finding the maximum of the likelihood $\pi_{d|x}(d_t|x) = \pi_e(d_t - Ax)$. Both interpretations give insight into when standard least squares may be valid.

Recall that the least squares estimate is not necessarily unique. Consider the minimum norm least squares/pseudoinverse estimate $x^\dagger = A^\dagger d_t$ found using the pseudoinverse. Recall that we expand A using the singular value decomposition as $A = \sum_{j=1}^{r_t} d_j u_j v_j^T$ where r_t is the rank of A. We construct the pseudoinverse as $A^\dagger = \sum_{j=1}^{r_t} \frac{1}{d_j} v_j u_j^T$. For further analysis,

we will also consider the matrix $V_{\text{null}} = (v_{r_t+1}, v_{r_t+2}, \ldots, v_{n_x})$, the matrix of right singular vectors spanning the nullspace of A. We use this to construct $P_{\text{null}} = V_{\text{null}} V_{\text{null}}^T$, the projection matrix to the nullspace of A.

The minimum norm least squares estimate can be written as

$$x^\dagger = A^\dagger d_t \tag{4.34}$$

$$= \min_{x_{\text{ls}}} \left\{ \|x_{\text{ls}}\|_2^2 \right\} \text{ where } x_{\text{ls}} = \min_x \left\{ \|Ax - d_t\|_2^2 \right\} \tag{4.35}$$

$$= \min_x \left\{ \|Ax - d_t\|_2^2 \right\} \text{ such that } P_{\text{null}} x = 0_{n_x,1} \tag{4.36}$$

Now consider the MAP estimate

$$x_{\text{MAP}} = \min_x \left\{ \|Ax - d_t\|_2^2 + \|\alpha P_{\text{null}} x\|_2^2 \right\} \tag{4.37}$$

i.e. we have chosen $\Gamma_x^{-1} = \alpha^2 P_{\text{null}}^T P_{\text{null}}$. Note that

$$\lim_{\alpha \to \infty} x_{MAP} = \lim_{\alpha \to \infty} \min_x \left\{ \|Ax - d_t\|_2^2 + \|\alpha P_{\text{null}} x\|_2^2 \right\} \tag{4.38}$$

$$= \min_x \{ \|Ax - d_t\| \} \text{ such that } P_{\text{null}} x = 0_{n_x,1} \tag{4.39}$$

$$= x^\dagger \tag{4.40}$$

that is, we can express the pseudoinverse estimate as a limiting case of the MAP in the Bayesian framework. This is an analytically useful trick that will be used throughout this review of regularisation methods in the Bayesian framework, allowing a more explicit formulation of the assumptions being made. Here we see that the pseudoinverse estimate assumes that x has exactly zero component in the nullspace of A.

4.2.2 Regularisation by Discretisation

Recall that a method of regularising an equation of the form $d = \bar{A}\bar{x} + e$ is to simply reduce the dimension of $\bar{x}$ i.e. we go from $\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$ to $x = P_{x,\bar{x}} \bar{x} \in \mathbb{R}^{n_x}$ with $n_x < n_{\bar{x}}$ where $P_{x,\bar{x}}$ projects from $\bar{x}$ to x.

Consider the "fine discretisation" least squares problem

$$\bar{x}_{\text{ls}} = \min_{\bar{x}} \left\{ \left\| \bar{A}\bar{x} - d_t \right\|_2^2 \right\} \tag{4.41}$$

and compare with the MAP estimate

$$\bar{x}_{\text{MAP}} = \min_{\bar{x}} \left\{ \left\| \bar{A}\bar{x} - d_t \right\|_2^2 + \|\alpha(\bar{x} - P_{x,\bar{x}} \bar{x})\|_2^2 \right\} \tag{4.42}$$

$$= \min_{\bar{x}} \left\{ \left\| \bar{A}\bar{x} - d_t \right\|_2^2 + \|\alpha(I - P_{x,\bar{x}}) \bar{x}\|_2^2 \right\} \tag{4.43}$$

where our prior on $\bar{x}$ is normal with $\Gamma_{\bar{x}}^{-1} = \alpha^2(I - P_{x,\bar{x}})^T(I - P_{x,\bar{x}})$. Consider the limit

$$\lim_{\alpha\to\infty} \min_{\bar{x}} \left\{ \left\| \bar{A}\bar{x} - d_t \right\|_2^2 + \|\alpha P_{x,\bar{x}}\bar{x}\|_2^2 \right\} = \min_{\bar{x}} \left\{ \|Ax - d_t\|_2^2 \right\} \tag{4.44}$$

i.e. regularisation by discretisation is equivalent to using a prior/penalty term that assigns zero probability to solutions not in the basis of x. This interpretation is important to keep in mind when considering how to discretise a problem.

4.2.3 Regularisation by Measurement Truncation

If the original model was $\bar{d} = Ax + \bar{e}$, then the truncated measurement model can be written as $d = P_{d,\bar{d}}\bar{d} = P_{d,\bar{d}}(Ax + \bar{e})$. The problem is now to find

$$\min_{x} \left\{ \left\| P_{d,\bar{d}}(Ax - \bar{d}) \right\|_2^2 \right\} \tag{4.45}$$

so in the Bayesian setting we keep $\Gamma_x^{-1} = 0_{n_x,n_x}$ as in least squares, but now we also have $\Gamma_e^{-1} = P_{d,\bar{d}}^T P_{d,\bar{d}}$. This is equivalent to saying that for the elements of $\bar{d}$ projected to d, the measurement noise, and in turn the measurements themselves due to the marginalisation, are perfectly correlated.

4.2.4 Truncated Series Expansion

As mentioned in Section 3.2 a wide variety of expansions exist. We will only discuss the tSVD for the linear finite dimensional additive noise model $d = Ax + e$. We replace $A = UDV^T$ with $A_r = U_r D_r V_r^T$, the optimal (w.r.t. the 2-norm) rank r approximation to A. The true rank of A is r_t, and $r \leq r_t$. The tSVD estimate is

$$x_{\text{tSVD}} = \min_{x} \{\|A_r x - d_t\|\} \tag{4.46}$$

$$= A_r^\dagger d_t = V_r D_r^{-1} U_r^T d \tag{4.47}$$

i.e. the pseudoinverse estimate but using A_r instead of A. The rest of the analysis is similar to the analysis of the pseudoinverse estimate in Section 4.2.1.

4.2.5 Truncated Iterative Methods

Suppose we regularise our estimates by only taking r steps of an iterative method. Let X_r be the set of all x that can be reached by step r of the method of choice. For example, suppose the method is Landweber iterations and $r = 1$. For this case,

$$X_1 = \left\{ x : x = x_0 + \beta(A^T d - A^T A x_0) \right\} \tag{4.48}$$

where x_0 is some initial estimate e.g. $x_0 = \mu_x$. If β is small, X_1 is a small subset of $\mathbb{R}^{n_x}$.

Let $P_{X_r,x}$ be the projection matrix from x to X_r. Consider the MAP estimate

$$x_{\text{MAP}} = \min_x \left\{ \|Ax - d_t\|_2^2 + \|\alpha P_{X_r,x} x\|_2^2 \right\} \tag{4.49}$$

i.e. we have chosen $\Gamma_x^{-1} = \alpha^2 P_{X_r,x}^T P_{X_r,x}$. The MAP estimate above will be equivalent to the truncated iteration estimate in the limit $\alpha \to \infty$.

Restricting the solution space to X_r has parallels to Krylov subspace methods [32]. There are also parallels to the sample based methods of [37]. The underlying idea is that repeated evaluations of matrix vector products reveals the dominant components (e.g. SVD) of a matrix. This concept will be considered further in Chapter 6.

4.2.6 Tikhonov Regularisation

We have already discussed the similarity between the Bayesian MAP estimate

$$x_{\text{MAP}} = \min_x \{ \left\| \tilde{L}_e(d_t - A(x) - \mu_e) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \} \tag{4.50}$$

and the standard Tikhonov estimate

$$x_{\text{tik}} = \min_x \{ \|d_t - A(x)\|_2^2 + \alpha \|x\|_2^2 \} \tag{4.51}$$

which is equivalent to the Bayesian MAP with $\mu_x = 0_{n_x,1}$, $\mu_e = 0_{n_d,1}$, $\Gamma_e = \sigma_e^2 I$ and $\Gamma_x = \sigma_x^2 I$ corresponding to $\alpha = \frac{\sigma_e^2}{\sigma_x^2}$. The Bayesian interpretation allows us to choose a value of α without resorting to computationally expensive methods such as the Morozov discrepancy principle and the L-curve criterion. Estimates of σ_e and σ_x can often be reached through considerations of the problem at hand.

Consider the 1D deconvolution example of Section 2.3.2. We know the standard deviation of the noise to be $\sigma_e = 0.1$. We also now the ground truth in this example. We estimate $\sigma_x \approx 1.3$. We would then say the "Bayesian" value of α would be around $\frac{0.1^2}{1.3^2} \approx 0.006$. The estimate found with this value of α is shown in Figure 4.2. This choice of α is too small, resulting in an estimate with overly large nonsmooth terms, similar to the L-curve fit of $\alpha = 0.01$. This choice of $\alpha = 0.0006$ would however be a good starting point if we were then to implement Morozov discrepancy principle or L-curve criterion, potentially reducing the computational cost.

The Bayesian estimate of $\alpha = 0.0006$ results in a worse estimate of x_t in this problem than the Morozov ($\alpha = 0.1$) and L-curve ($\alpha = 0.01$) estimates. We can explain this result from the Bayesian perspective. These estimates assume x is normally distributed with $\Gamma_x = \sigma_x^2 I$ i.e. we assumed that x has no correlation structure. When we look at x however this assumption is clearly false. The smooth variation indicates a distinct correlation structure. The estimation

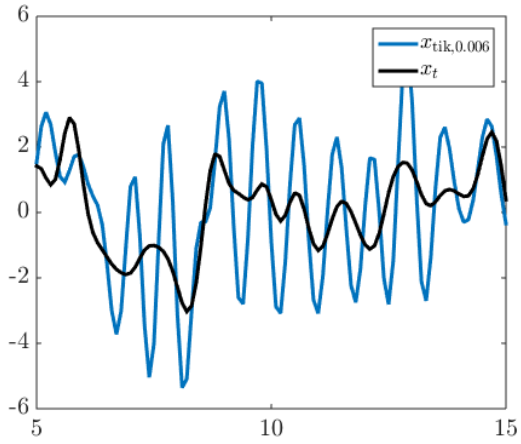


Figure 4.2: Estimate $x_{\text{tik},0.0006}$ where $\alpha = 0.0006$ comes from Bayesian considerations, alonside ground truth x_t

is bad because our prior model is bad. The Morozov and L-curve estimates perform better, but that is because these estimates better account for the flawed prior model.

This gives some practical insight into applying Bayesian methods in practice. Ideas such as Morozov discrepancy principle and L-curve criterion could be used to correct inaccurate priors. This is a concept that will be explored in Section 5.1.2.

4.2.7 Exponential Priors and Generalised Tikhonov Regularisation

Recall that $\pi_{x|d}(x|d_t) \propto \pi_{d|x}(d_t|x)\pi_x(x) = \pi_{e|x}(d_t - A(x,e)|x)\pi_x(x)$. In the case that x and e are mutually independent, we can write $\pi_{x|d}(x|d_t) \propto \pi_e(d_t - A(x,e))\pi(x)$. A useful structure to impose is that $\pi(e)$ and $\pi(x)$ belong to the family of exponential priors i.e. $\pi(e) \propto \exp(G_e(e))$ and $\pi(x) \propto \exp(G_x(x))$. Having probability densities in this form allows us to write

$$\pi_{x|d}(x|d_t) \propto \exp\left(G_e\left(d_t - A(x,e)\right) + G_x(x)\right) \tag{4.52}$$

which allows us to write the MAP as

$$x_{\text{MAP}} = \min_x \left\{G_e\left(d_t - A(x,e)\right) + G_x(x)\right\} \tag{4.53}$$

as in generalised Tikhonov regularisation. In the case that $\pi(e) \sim \mathcal{N}(\mu_e, \Gamma_e = (\tilde{L}_e^T \tilde{L}_e)^{-1})$, $G_e(e) = \left\|\tilde{L}_e(e - \mu_e)\right\|_2^2$.

Performing Bayesian inversions successfully is clearly reliant on not just our model A of the physical system, but also the prior model on x. We will consider some useful prior models in Chapter 5. Before that we provide another benefit of the Bayesian framework, namely the natural extension to uncertainty quantification.

4.3 Uncertainty Quantification

We have stated that the Bayesian "solution" to an inverse problem is the posterior density $\pi_{x|d}(x|d)$. The MAP was then presented as an estimate of x_t. Various regularisation methods were then reinterpreted in terms of the posterior and the MAP.

The point estimate however is not representative of the "entire solution" $\pi(x|d)$. We can provide a more meaningful representation by including some kind of spread estimate. That is, we want to *quantify our uncertainty* of x. Quantifying uncertainty is called *uncertainty quantification*, and is discussed in [49]. While point estimates are useful, a potentailly more useful statement might be "with probability p, x is in this region" [65].

We denote the region x lies in with probability p given data $d = d_t$ as $\Omega_{p,x|d}$. We can define the region as the solution to

$$p = \int_{\Omega_{p,x|d}} \pi(x|d)dx \tag{4.54}$$

with the condition that $\pi(x|d) > p^*$ for all $x \in \Omega_{p,x|d}$ to make this region unique. Finding $\Omega_{p,x|d}$ is potentially computationally demanding. Suppose we evaluated the integral using m point quadrature in each dimension of x. Then the computation would require m^{n_x} evaluations of $\pi(x|d)$ which at the very least would require m^{n_x} evaluations of A. This integration would then need to be performed multiple times over different regions in an attempt to find $\Omega_{p,x|d}$. For particular distributions, more computationally feasible methods exist. This section will outline some commonly used simplified approaches to uncertainty quantification.

4.3.1 Linear Model Normal Unknowns

Consider the posterior $\pi_{x|d}(x|d_t)$ in the case that $d_t = Ax_t + e_t$ with $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $e \sim \mathcal{N}(\mu_e, \Gamma_e)$. In this case, the posterior is

$$\pi_{x|d}(x|d_t) \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_e(d_t - Ax - \mu_e)\right\|_2^2 + \left\|\tilde{L}_x(x - \mu_x)\right\|_2^2\right)\right) \tag{4.55}$$

which is a normal distribution. More specifically, it is a normal distribution with MAP

$$x_{\text{MAP}} = \begin{pmatrix}\tilde{L}_e A \\ \tilde{L}_x\end{pmatrix}^{\dagger}\begin{pmatrix}\tilde{L}_e(d - \mu_e) \\ \tilde{L}_x\mu_x\end{pmatrix} \tag{4.56}$$

which we know to be equal to the CM. In the case that x and e are mutually independent

$$x_{\mathrm{CM}} = \mu_{x|d=d_t} = \mu_x + \Gamma_{xd}\Gamma_d^{-1}(d_t - \mu_d) \tag{4.57}$$

$$= \mu_x + \Gamma_x A^T \left(A\Gamma_x A^T + \Gamma_e\right)^{-1} (d_t - A\mu_x - \mu_e) \tag{4.58}$$

and conditional covariance

$$\Gamma_{x|d} = \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} \tag{4.59}$$

$$= \Gamma_x - \Gamma_x A^T \left(A\Gamma_x A^T + \Gamma_e\right)^{-1} A\Gamma_x \tag{4.60}$$

$$= \left(\Gamma_x^{-1} + A^T\Gamma_e^{-1}A\right)^{-1} \tag{4.61}$$

where the equivalence between 4.60 and 4.61 comes from an application of the matrix inversion lemma/Sherman-Morrison-Woodbury formula. Derivation of these results are readily available in the literature, for example [3, 4, 65, 66].

Note that in this case the posterior distribution can be characterised completely with linear algebra. The rigorous uncertainty quantification would be to now find $\Omega_{p,x|d}$, the set of all x such that the posterior is always above some value and that the posterior integrates over to give p. The construction of this "trust region" can be found in [67] as

$$\left\|\tilde{L}_{x|d}(x - \mu_{x|d})\right\|_2^2 \leq \chi^2_{n_x}(p) \tag{4.62}$$

where $\tilde{L}_{x|d}^T\tilde{L}_{x|d} = \Gamma_{x|d}^{-1}$ and $\chi^2_{n_x}(p)$ is the probability p chi squared quantile function for n_x degrees of freedom.

The above region is difficult to interpret and display. In this thesis, the typical quantification of uncertainty is the *posterior error interval*. We define

$$\hat{\sigma}_{x|d} = \mathrm{diag}(\Gamma_{x|d}) \tag{4.63}$$

and define the $k\hat{\sigma}$ posterior error interval to be $x_{MAP} \pm k\hat{\sigma}_{x|d}$. In the case that $\pi_{x|d}$ is a normal distribution and $\Gamma_{x|d}$ is a diagonal matrix, approximately 66% of x_t will be in the $k = 2$ interval, and 99% in the $k = 3$ interval. While this uncertainty estimate is crude, it is simple to compute and interpret.

4.3.2 Nonlinear Model Normal Unknowns

Consider the posterior $\pi_{x|d}(x|d_t)$ in the case that $d_t = A(x_t) + e_t$ with $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $e \sim \mathcal{N}(\mu_e, \Gamma_e)$. Note that $A(x)$ is not a linear mapping of x. In this case, the posterior is

$$\pi(x|d) \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_e(d_t - A(x) - \mu_e)\right\|_2^2 + \left\|\tilde{L}_x(x - \mu_x)\right\|_2^2\right)\right) \tag{4.64}$$

and the MAP and CM can not be computed just by linear algebra, in contrast to the linear case. The MAP is expressed as the solution to the optimisation problem

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - A(x) - \mu_e) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{4.65}$$

$$= \min_x \left\{ \begin{pmatrix} \tilde{L}_e A(x) \\ \tilde{L}_x x \end{pmatrix} - \begin{pmatrix} \tilde{L}_e(d - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} \right\} \tag{4.66}$$

$$= \min_x \left\{ \hat{A}(x) - \hat{d} \right\} \tag{4.67}$$

where we redefine $\hat{A}$ and $\hat{d}$ for the nonlinear case in this section. The optimisation can be solved by e.g. a *line search* method such as the *Gauss-Newton algorithm*. Line search algorithms are iterative methods of the form

$$x_{j+1} = x_j + \beta_j \vec{x}_j \tag{4.68}$$

where $\beta_j \in \mathbb{R}$ is the *step length* and $\vec{x}_j \in \mathbb{R}^{n_d}$ is the *search direction*. In the Gauss-Newton algorithm, the search direction is

$$\vec{x}_j = J_{x_j}^{\dagger} \big(\hat{A}(x_j) - \hat{d} \big) \tag{4.69}$$

where $J_{x_j} \in \mathbb{R}^{n_d \times n_x}$ is the *Jacobian matrix* of $\hat{A}$ at x_j defined as

$$J(i,k) = \frac{\partial d(i)}{\partial x_j(k)} \tag{4.70}$$

where i and k are indices. A more in depth discussion of line search algorithms can be found in e.g. [68].

Gaussian distributions can be characterised by just a point (MAP, CM) and a matrix (covariance). The posterior distribution for nonlinear problems is not necessarily gaussian, and typically more than just a vector and a matrix is required to characterise the posterior distribution. We can however compute a *local Gaussian approximation* to the posterior, known as the *Laplace approximation*. Figure 4.1 is an example where a Gaussian approximation to the true distribution might be accurate near the MAP.

The Jacobian can be used to form an affine approximation to A around x_{MAP} as

$$A(x) \approx B_{\text{MAP}}(x) = x_{MAP} + J_{\text{MAP}}(x - x_{\text{MAP}}) \tag{4.71}$$

where $B_{\text{MAP}}(x)$ is an affine approximation to A and J_{MAP} is the Jacobian of A at x_{MAP}. $B_{\text{MAP}}(x)$ is also the *first order Taylor expansion* of A at x_{MAP}, as discussed in [48, 68].

The Laplace approximation [69, 70, 71] to the posterior covariance is

$$\hat{\Gamma}_{x|d} = \Gamma_x - \Gamma_x J^T \left(J\Gamma_x J^T + \Gamma_e \right)^{-1} J\Gamma_x \tag{4.72}$$

$$= \left(\Gamma_x^{-1} + J^T \Gamma_e^{-1} J \right)^{-1} \tag{4.73}$$

where $\Gamma_{x|d} \approx \hat{\Gamma}_{x|d}$ is found by replacing $A(x)$ with $B(x)$. The Laplace approximation is used in a range of texts, for example [4, 49].

From this, we can perform approximate uncertainty quantification similar to the linear case. Note that both Gauss-Newton line search and the Laplace approximation make use of the Jacobian.

Chapter 5

Prior Models

Loosely speaking, the prior encodes information about x before any data $d = d_t$ is collected. Suppose we have two realisations of x, x_{likely} and x_{unlikely}. A prior should be such that $\pi_x(x_{\text{unlikely}}) < \pi_x(x_{\text{likely}})$. Typically x_t is predicted to have particular features e.g. be "smooth". A prior is then chosen such that draws of the prior also have these features.

Consider a prior $\pi_x(x)$ on $x \in \mathbb{R}^{n_x}$. Let $0 < \kappa \ll \max_x\{\pi_x(x)\}$ be a small number. In this thesis, if $\pi_x(x) > \kappa$ in a relatively small region of $\mathbb{R}^{n_x}$, we say the prior is *strict* or *narrow*. $\pi_x(x) > \kappa$ in a relatively large region of $\mathbb{R}^{n_x}$, we say the prior is *loose* or *wide*. The strictest prior is that x is fixed. This can be thought of as $\Gamma_x = 0_{n_x,n_x}$. The loosest prior is that all $x \in \mathbb{R}^{n_x}$ are equally likely. This can be though of as $\Gamma_x^{-1} = 0_{n_x,n_x}$. Recall that the MAP estimate with $\Gamma_x^{-1} = 0_{n_x,n_x}$ was found to be equivalent to the least squares estimate in Section 4.2.1.

In this thesis, a *draw* or *sample* $x_j \in \mathbb{R}^{n_x}$ of random variable x is such that $\pi_x(x_j) > 0$. Suppose $x \sim \mathcal{U}(0,1)$ i.e. $n_x = 1$, $\pi_x(x) = 1$ for $0 < x < 1$ and $\pi_x(x) = 0$ elsewhere. Then $x_j = 0.1234$ is a sample of x, and a draw from $\pi_x(x)$. Analysing draws is a useful way of interpreting the features of the prior.

In this thesis, we will typically consider discretised forms of Gaussian processes. This allows for intuitive construction of finite dimensional covariance matrices and natural computational implementation, although the subtle difference in interpretation can lead to useful analytic results. More detail can be found in e.g. [4, 73, 92]. This chapter reviews several types of prior with useful properties.

5.1 Gaussian Priors

A widely used type of prior is *Gaussian* or *normal*. A review of Gaussian priors can be found in [4]. Recall that a normally distributed variable x is denoted as $x \sim \pi_x(x) = \mathcal{N}(\mu_x, \Gamma_x)$ where $\mu_x \in \mathbb{R}^{n_x}$ is the mean of x and $\Gamma_x \in \mathbb{R}^{n_x \times n_x}$ is the covariance of x. The normal distribution

is defined to be

$$\mathcal{N}(\mu_x, \Gamma_x) = \left((2\pi)^{n_x}|\Gamma_x|\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu_x)^T\Gamma_x^{-1}(x-\mu_x)\right) \tag{5.1}$$

$$\propto \exp\left(-\frac{1}{2}\left\|\tilde{L}_x(x-\mu_x)\right\|_2^2\right) \tag{5.2}$$

where $\tilde{L}_x^T\tilde{L}_x = \Gamma_x^{-1}$. Note the use of Γ_x^{-1}.

Let

$$d_t = Ax_t + e_t \tag{5.3}$$

be a realisation of $d = Ax + e$ where $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $e \sim \mathcal{N}(\mu_e, \Gamma_e)$. The MAP estimate is

$$x_{\text{MAP}} = \min_x \left\{ \left\|\tilde{L}_e\left(d_t - Ax - \mu_e\right)\right\|_2^2 + \left\|\tilde{L}_x\left(x-\mu_x\right)\right\|_2^2 \right\} \tag{5.4}$$

$$= \min_x \left\{ \left\| \begin{pmatrix} \tilde{L}_eA \\ \tilde{L}_x \end{pmatrix} x - \begin{pmatrix} \tilde{L}_e(d_t-\mu_e) \\ \tilde{L}_x\mu_x \end{pmatrix} \right\|_2^2 \right\} \tag{5.5}$$

$$= \begin{pmatrix} \tilde{L}_eA \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_e(d_t-\mu_e) \\ \tilde{L}_x\mu_x \end{pmatrix} \tag{5.6}$$

i.e. the MAP estimate can be found as the solution to a linear algebra problem. This demonstrates the computational attractiveness of Gaussian models.

Let $w_j \in \mathbb{R}^{n_x}$ be a draw of white noise $w \sim \pi_w(w) = \mathcal{N}(0_{n_x,1}, I_{n_x})$. Let L_x be such that $L_xL_x^T = \Gamma_x$. Then

$$x_j = L_xw_j + \mu_x \tag{5.7}$$

is a draw of x from $\pi(x) = \mathcal{N}(\mu_x, \Gamma_x)$. Note that these samples have the correct mean as

$$\mathbb{E}(x_j) = \mathbb{E}(\mu_x + L_xw_j) = \mu_x + L_x\mathbb{E}(w_j) = \mu_x \tag{5.8}$$

and the covariance is

$$\text{Covariance}(x_j) = \mathbb{E}\left((x_j-\mu_{x_j})(x_j-\mu_{x_j})^T\right) = \mathbb{E}(L_xw_jw_j^TL_x^T) \tag{5.9}$$

$$= L_x\mathbb{E}(w_jw_j^T)L_x = L_xI_{n_d}L_x^T = \Gamma_x \tag{5.10}$$

so Equation (5.7) is valid. Note that Equation (5.7) states drawing from a Gaussian distribution as an affine function of white noise. In other words, if a Gaussian prior is desired with certain features, it may be more intuitive to consider the affine transformation of white noise that would produce such features. This concept is considered further in Section 5.1.3

5.1.1 Gaussian Smoothness Priors

Consider the function $\mathbf{x}(\mathbf{t})$ over $\mathbf{t} \in \Omega \subset \mathbb{R}^{n_t}$. Let $t \in \mathbb{R}^{n_x \times n_t}$ such that $t(j,:) \in \Omega$. Let $x(j) = \mathbf{x}\big(t(j,:)\big)$. We might assume the difference $|x(j) - x(k)|$ to depend on the distance $\|t_j - t_k\|_2^2$. In this case, we might use a so-called (Gaussian) "smoothness" prior. In a Gaussian smoothness prior, the covariance matrix Γ_x is such that elements of x are correlated based on proximity.

In this thesis, we define a smoothness prior covariance as

$$\Gamma(j,k) = \left(\frac{h}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l^2}\right) \tag{5.11}$$

where h is called the *height* and l is called the *correlation length*. We expect a draw x_j plotted over t to have "bumps" of height approximately h and length approximately l. Equation (5.11) is a *covariance function* for Gaussian smoothness priors. More specifically, Equation (5.11) is a modification of the "exponential squared" covariance function [73]. The covariance function of Equation (5.11) is *stationary* as it only depends on $t_j - t_k$ i.e. it is translation invariant. The covariance function of Equation (5.11) is also *isotropic* as it only depends on $\|t_j - t_k\|_2$ i.e. it is radially invariant. Examples of *nonstationary* and *anisotropic* covariance functions can be found in Section 5.1.4.

We now construct an example smoothness prior. Let $\mathbf{t} \in \mathbb{R}$ be the spatial variable over domain $\Omega = [0, 20]$ and $\mathbf{x}(\mathbf{t})$ be a function on Ω. Let $t \in \mathbb{R}^{n_x}$ be a finite dimensional approximation of $\mathbf{t}$ on Ω with $t(1) = 0$, $t(n_x) = 20$ and $t(j+1) - t(j) = \Delta_t$. Let $x \in \mathbb{R}^{n_x}$ be a finite dimensional approximation of $\mathbf{x}$ with $\mathbf{x}(t(j)) = x(j)$. Let $n_x = 201$ i.e. $\Delta_t = 0.1$. This is the same discretisation used for the deconvolution problem first considered in Section 2.3.

Let $h = 3$ and $l = 0.3$. The first row of Γ_x is plotted in Figure 5.1 and the entire matrix is visualised in Figure 5.2. Note that the matrix is effectively sparse, with only the 10 or so values near the diagonal being relatively large.

Draws x_j of x can be computed as $x_j = L_x w_j$ where $L_x L_x^T = \Gamma_x$. We shall compute L_x as the *Cholesky decomposition* of Γ_x, where L_x is the *Cholesky factor*. Discussion of the Cholesky decomposition and how it is computed can be found in [32]. Note that the Cholesky decomposition gives *a* matrix such that $L_x L_x^T = \Gamma_x$, but other matrices $\hat{L}_x$ such that $\hat{L}_x \hat{L}_x^T = \Gamma_x$ do exist [45]. The Cholesky decomposition is used here as it can be computed efficiently. Note that the Cholesky factor only exists for positive definite matrices. The Cholesky factor L_x is visualised in Figure 5.3. Note that L_x is upper triangular and effectively sparse.

A draw w_j of white noise $w \sim \mathcal{N}(0_{n_x,1}, I_{n_x})$ is shown in Figure 5.4. The draw w_j was computed using the random number generator in MATLAB as $w_j = \text{randn}(n_x, 1)$. The corresponding draw $x_j = L_x w_j$ of $x \sim \mathcal{N}(0_{n_x,1}, \Gamma_x)$ is shown in Figure 5.5. Comparing Figures 5.4 and 5.5 illustrates why this is called a "smoothness" prior. Note that x_j has "bumps" of width around 0.6 i.e. $2l$ and height around ± 3 i.e. h. This x_j is in fact the unconvolved signal s_t from Section 2.3.

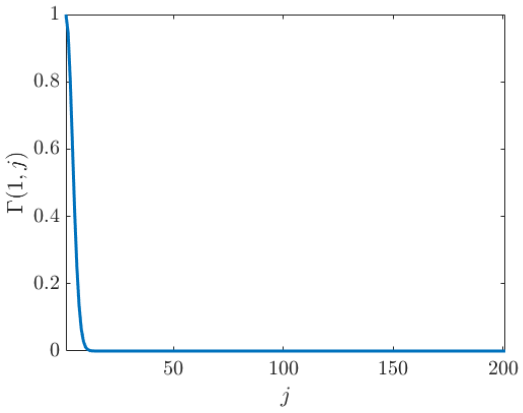


Figure 5.1: First row of Γ_x

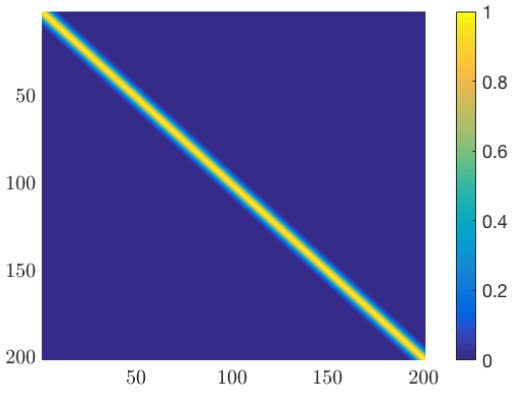


Figure 5.2: Visualisation of Γ_x

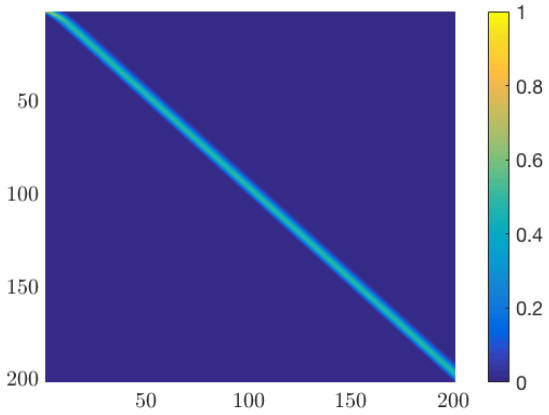


Figure 5.3: Visualisation of L_x

Consider again the $n_d = 51$ deconvolution problem of Section 2.3.2. In this section, we treat $x = s$ i.e. x contains values corresponding to $\mathbf{t}$ over the entire domain $\Omega = [0, 20]$, but we only show the estimates on $\Omega_x = [5, 15]$.

This convolution is modelled as $d = Ax + e$. The MAP estimate can be expressed as

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_e \left(d_t - Ax - \mu_e \right) \right\|_2^2 + \left\| \tilde{L}_x \left(x - \mu_x \right) \right\|_2^2 \right\} \tag{5.12}$$

$$= \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_e (d - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{5.13}$$

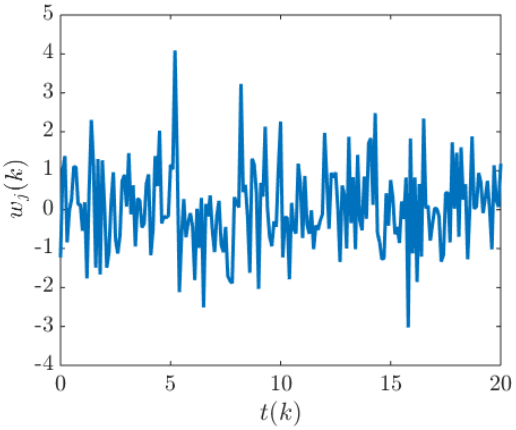


Figure 5.4: White noise w_j

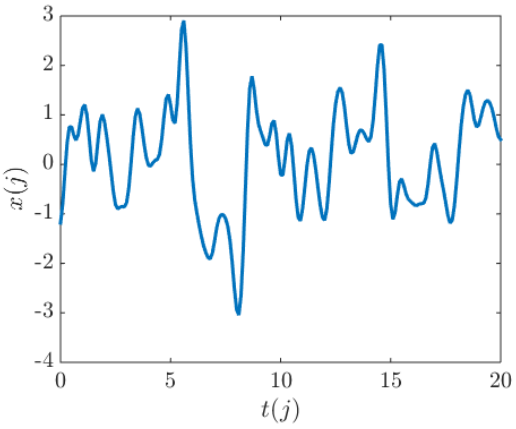


Figure 5.5: Draw of $\pi_x(x)$ computed as $x_j = Lw_j$

where $\tilde{L}_e^T \tilde{L}_e = \Gamma_e^{-1}$ and $\tilde{L}_x^T \tilde{L}_x = \Gamma_x^{-1}$. The covariance Γ_x was constructed as in Equation (5.11) with $h = 3$ and $l = 0.3$ i.e. $\pi_x(x)$ is a Gaussian smoothness prior.

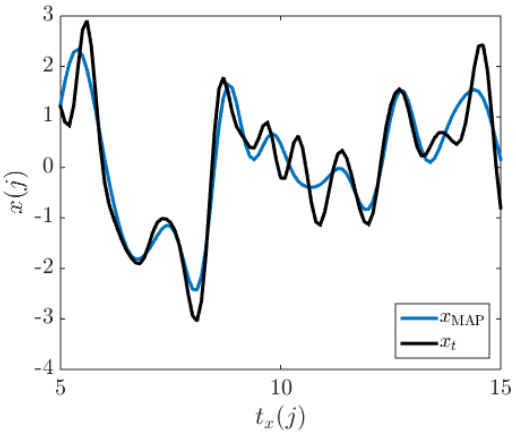


Figure 5.6: Ground truth and MAP estimate

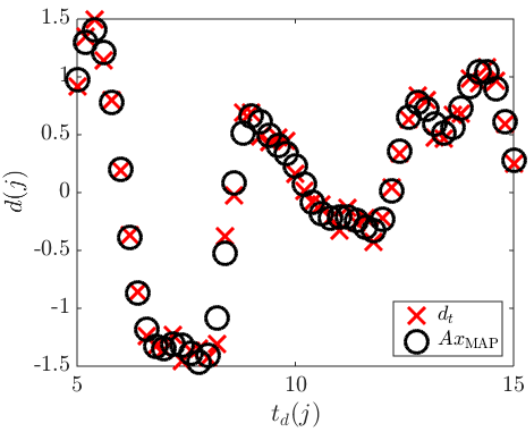


Figure 5.7: Model predictions Ax_{MAP} and data d_t

The MAP estimate x_{MAP} and ground truth x_t are plotted in Figure 5.6. Note that the MAP seems to be slightly "over smooth", similar to the Morozov estimate of Section 3.4.1. The MAP estimate is the product of 1 inversion, whereas the Morozov estimate required multiple inversions be computed in order to estimate the regularisation parameter α. The predictions

Ax_{MAP} are plotted alongside the data d_t in Figure 5.7.

The Bayesian framework can be used to quantify uncertainty in the estimate. This model $d = Ax + e$ has linear forward map A and normally distributed x and e. The posterior is therefore a normal distribution with covariance

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A^T \Gamma_e^{-1} A)^{-1}. \tag{5.14}$$

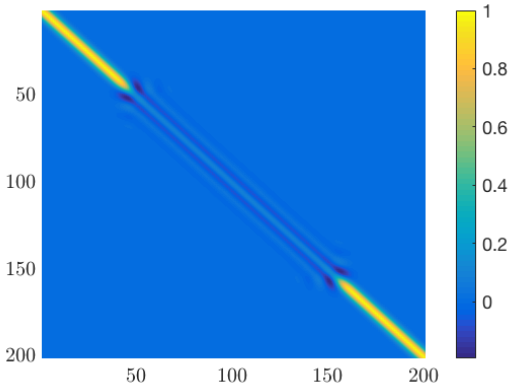


Figure 5.8: Posterior covariance $\Gamma_{x|d}$

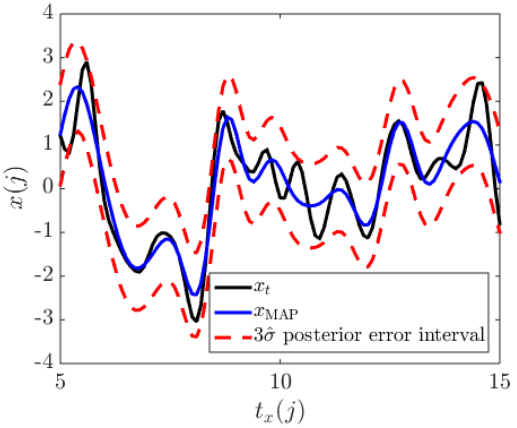


Figure 5.9: Ground truth, MAP estimate and posterior error interval

The posterior covariance matrix is visualised in Figure 5.8. Note that the central block $\Gamma_{x|d}(50:150, 50:150)$ corresponds to Ω_x. Recall that $d(j) = c\big(t_d(j)\big)$ and $t_d(j) \in \Omega_x$ for this problem. The rest of $\Gamma_{x|d}$ appears similar to Γ_x from Figure 5.2.

Let $\text{diag}(\Gamma_{x|d}) = \hat{\sigma} \in \mathbb{R}^{n_x}$. This is used to construct the approximate posterior error interval $x_{\text{MAP}} \pm 3\hat{\sigma}$ as described in Section 4.3. The posterior error interval is plotted in Figure 5.9, and we do see x_t is mostly contained. In this case, the ground truth x_t is a sample of the prior used in the MAP estimation, which is a slightly idealised situation.

5.1.2 Prior Tuning and Hyperpriors

The exponential squared smoothness prior covariance function of Equation (5.11) is defined by 2 parameters, height h and correlation length l. In the above example, $h_t = 3$ and $l_t = 0.03$, the true values of h and t, were known. Adjusting the parameters in a prior to get the most representative estimates of x_t is called *tuning* the prior. The parameters of the prior can be treated as random variables. The probability distribution of the parameters in the prior is known as the *hyperprior*.

Recall that the standard Tikhonov estimate is

$$x_{\text{tik}} = \min_x \left\{ \|d_t - Ax\|_2^2 + \alpha \|x\|_2^2 \right\} \tag{5.15}$$

which is the same as the map estimate if $\Gamma_e \propto I$ and $\Gamma_x \propto I$. For a smoothness prior, $\Gamma_x \approx \left(\frac{h}{3}\right)^2 I$ when $l < \Delta_t$. The Tikhonov estimates of Sections 3.4.1 and 3.4.2 therefore correspond to estimating l too small.

Let us consider 4 cases. Each will have h and l too small or too large. In this section, $l = 0$ will be taken to mean $l \ll \Delta_t$, noting that the covariance function of Equation (5.11) is not defined for $l = 0$. Case 1 has $h = 0.1$ and $l = 0$, and the corresponding estimate $x_{\text{MAP},h=0.1,l=0}$ of x_t is shown in Figure 5.10. The MAP estimate stays near 0, and the posterior error intervals are far too narrow. This is a consequence of the $h = 0.1$, $l = 0$ smoothness prior being too strict.

Case 2 has $h = 10$ and $l = 0$, and the corresponding estimate $x_{\text{MAP},h=10,l=0}$ of x_t is shown in Figure 5.11. The MAP estimate largely follows the ground truth, although nonsmooth terms are slightly overamplified in the MAP. This case loosely corresponds to the standard Tikhonov solution with α slightly too small. Compare this with the Morozov estimate of Section 3.4.1 in which α was slightly too large. While the MAP estimate with $h = 10$ and $l = 0$ is fairly representative, the posterior error intervals are far too wide. This is a result of the $h = 10$, $l = 0$ smoothness prior being too loose. While it could be argued that overly wide posterior error intervals is preferred to overly narrow posterior error intervals, these intervals are practically worthless.

Case 3 has $h = 0.1$ and $l = 3$, and the corresponding estimate $x_{\text{MAP},h=0.1,l=3}$ of x_t is shown in Figure 5.12. The MAP estimate stays near 0, and the posterior error intervals are far too narrow.

Case 4 has $h = 10$ and $l = 3$, and the corresponding estimate $x_{\text{MAP},h=10,l=3}$ of x_t is shown in Figure 5.13. The MAP estimate appears overly smooth, and the posterior error intervals are far too narrow.

Cases 3 and 4 have $l = 3$, and the MAP estimates are far too smooth. This is a consequence of the $l = 3$ smoothness prior promoting very smooth $x \in \mathbb{R}^{n_x}$. The corresponding posterior error intervals are also far too narrow. This is likely because the posterior error intervals are computed with $\hat{\sigma} = \text{diag}(\Gamma_{x|d})$, a diagonal approximation to the posterior. We would expect the prior and covariance to have large off diagonal components for $l = 3$, so the diagonal approximation is probably not accurate.

The prior covariance Γ_x for case 2, $h = 10$ and $l = 0$, is visualised in Figure 5.14. The corresponding posterior $\Gamma_{x|d}$ is shown in Figure 5.16. Note that these matrices are approximately diagonal.

The prior covariance Γ_x for case 4, $h = 10$ and $l = 3$, is visualised in Figure 5.15. The corresponding posterior $\Gamma_{x|d}$ is shown in Figure 5.17. Note that $\Gamma_{x|d}$ corresponding to the

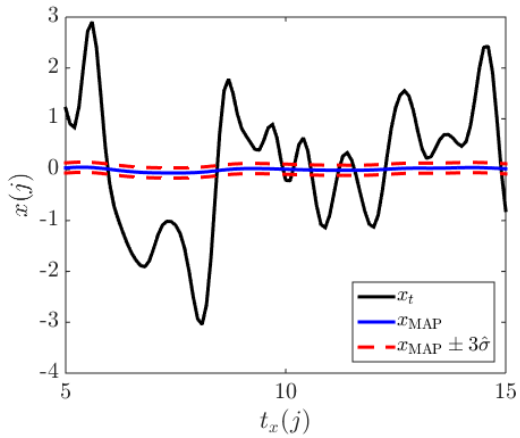


Figure 5.10: $h = 0.1, l = 0$

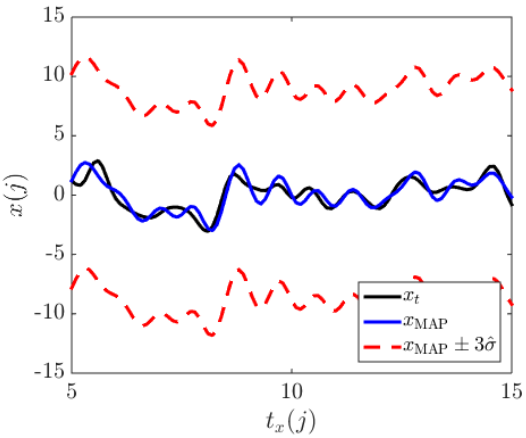


Figure 5.11: $h = 10, l = 0$

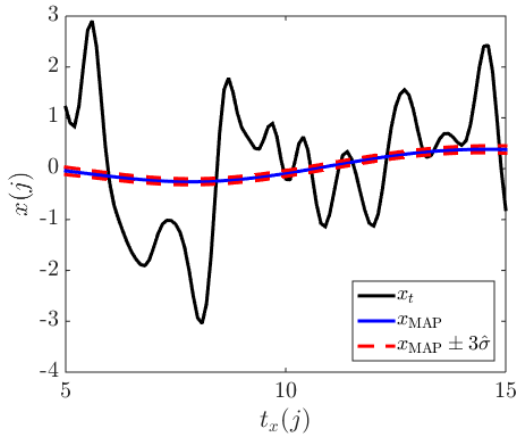


Figure 5.12: $h = 0.1, l = 3$

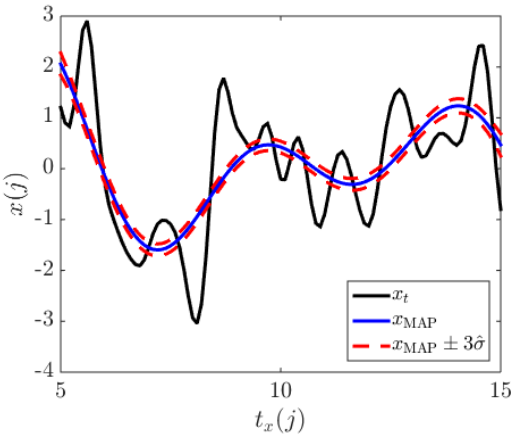


Figure 5.13: $h = 10, l = 3$

region of interest, $\Gamma_{x|d}(50:150, 50:150)$, is not diagonal at all. This explains why the $3\hat{\sigma}$ posterior error interval in Figure 5.13 was far too narrow.

Consider a prior controlled by parameters $p \in \mathbb{R}^{n_p}$. We denote this as $\pi_x^p(x)$. For example, the above discussion was of a Gaussian smoothness prior with $p = (h, l) \in \mathbb{R}^2$. In this thesis, prior tuning refers to the process of picking p such that $\pi_x^p(x_{\text{likely}}) > \pi_x^p(x_{\text{unlikely}})$.

The *hyperprior* is the probability distribution $\pi_p(p)$ on p. Let $\pi_p(p) \propto \exp(G_p(p))$ and $\omega = (x, p) \in \mathbb{R}^{n_x+n_p}$. Let $d = A(x)+e$ with $\pi_x(x) \propto \exp(G_x^p(x, p))$ and $\pi(e) \propto \exp(G_e(e))$. Note that the function G_x^p is a function of p.

In order to estimate x_t from $d_t = A(x_t) + e_t$, we form the MAP estimate of $\omega_t = (x_t, p_t)$

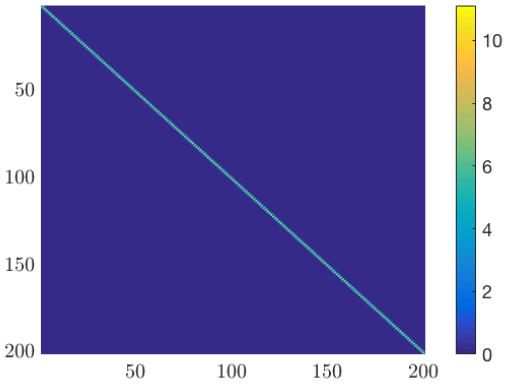


Figure 5.14: Γ_x, $h = 10$, $l = 0$

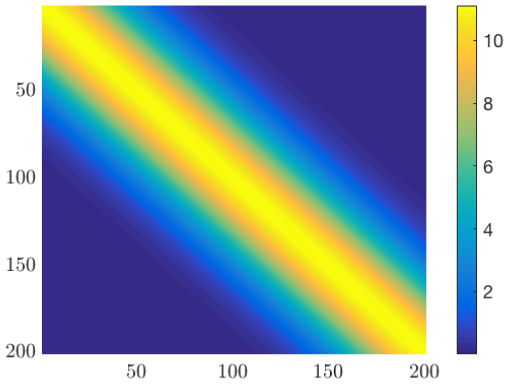


Figure 5.15: Γ_x, $h = 10$, $l = 3$

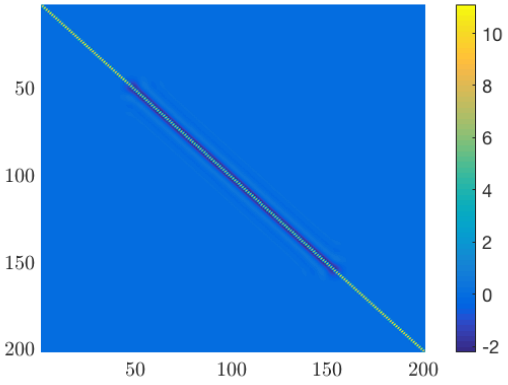


Figure 5.16: $\Gamma_{x|d}$, $h = 10$, $l = 0$

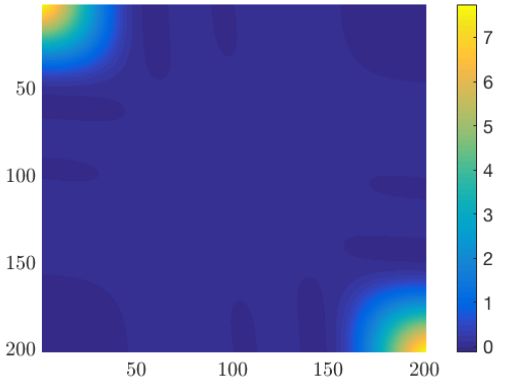


Figure 5.17: $\Gamma_{x|d}$, $h = 10$, $l = 3$

as

$$\omega_{\text{MAP}} = \begin{pmatrix} x_{\text{MAP}} \\ p_{\text{MAP}} \end{pmatrix} = \min_{\omega} \left\{ G_e\big(d_t - A(x)\big) + G_x^p(x,p) + G_p(p) \right\} \tag{5.16}$$

and then extract x_{MAP}. Note that the optimisation must be performed with respect to x and p. This is potentially a large increase in computational complexity over just optimising with respect to x. Suppose the optimisation is performed with an iterative method, and $\pi_x(x)$ is a Gaussian smoothness prior. At each iteration, a new p is used. This means computing a new Γ_x, and then finding a new $\tilde{L}_x$ such that $\tilde{L}_x^T \tilde{L}_x = \Gamma_x^{-1}$. So even though only 2 additional numbers $p = (h, l)$ are being estimated, an additional $\mathcal{O}(n_x^3)$ flops are used each iteration. A potential solution to this computational issue is discussed in Section 6.3.

The remainder of this section will consider adapting concepts from the Morozov discrepancy principle and the L-curve criterion to computing the MAP estimate with a Gaussian smoothness prior. Consider the generalised Tikhonov problem

$$x_{\text{tik}} = \min_x \left\{ G_e(d - A(x, e)) + \alpha G_x(x) \right\}. \tag{5.17}$$

Applying the Morozov discrepancy principle or the L-curve criterion to estimate α in the generalised Tikhonov problem has been considered in e.g. [74, 75, 76, 77]. In the Tikhonov problem, $p = \alpha$ and $G_{x,p}(x) = pG_x(x)$. I consider the case that $G_{x,p}(x) \neq pG_x(x)$. I suspect the Morozov discrepancy principle and L-curve criterion have been applied to this case in practice, although I cannot find explicit discussion in the literature.

Consider the deconvolution problem of Section 2.3.2. We will compute a MAP estimate of x_t using a smoothness prior, as in Section 5.1.1. Let $h_t = 3$ be known. We will estimate $l_t = 0.3$ first by the Morozov discrepancy principle, then by the L-curve criterion.

Let x_l to be the solution to

$$x_l = \min_x \left\{ \left\| \tilde{L}_e(Ax - d_t) \right\|_2^2 + \left\| \tilde{L}_{x,l}(x) \right\|_2^2 \right\} \tag{5.18}$$

where $\tilde{L}_{x,l}^T \tilde{L}_{x,l} = \Gamma_{x,l}^{-1}$. The covariance $\Gamma_{x,l}$ is constructed from the covariance function of Equation (5.11) with $h = 3$ and correlation length l. The "true" covariance in this problem is $\Gamma_{x,0.3}$. The Morozov discrepancy principle for standard Tikhonov regularisation estimates α_t with α_M as

$$\alpha_M = \min_\alpha \left\{ \|Ax_{\text{tik},\alpha} - d_t\|_2 - \sigma \right\} \tag{5.19}$$

where $x_{\text{tik},\alpha}$ is the Tikhonov estimate found with such α and $\sigma = \mathbb{E}(\|e\|_2)$. I generalise the Morozov approach by estimating l_t with l_M as

$$l_M = \min_l \left\{ \|Ax_l - d_t\|_2 - \sigma \right\}. \tag{5.20}$$

Several estimates x_l are computed with different l values. Corresponding $\|Ax_l - d_t\|_2$ and $\|e\|_2$ is plotted in Figure 5.18. Observe that $l_M \approx 0.6$.

We now compute the MAP estimate and posterior covariance for $l_M = 0.6$. The MAP with $3\hat{\sigma}$ posterior error intervals is shown in Figure 5.19. The MAP is too smooth, as expected when using $l_M \approx 2l_t$. The posterior error interval is slightly too narrow. The posterior covariance is visualised in Figure 5.20. Note that the diagonal approximation around $\Gamma_{x|d}(50:150, 50:150)$ is not valid, resulting in the overly narrow posterior error intervals of Figure 5.19.

I now adapt the L-curve criterion to the problem of estimating l_t. I compute the log-log plot of $\|x_l\|_2$ against $\|Ax_l - d\|_2$ and look for a "kink". The log-log plot is shown in Figure 5.21. This is an unusual "L-curve" plot, but recall that the L-curve criterion is typically used

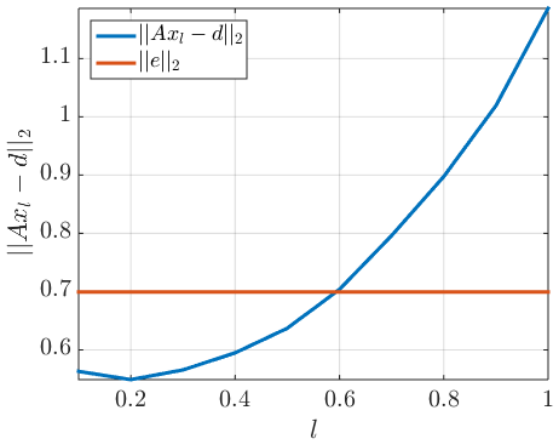


Figure 5.18: Finding l such that $\|Ax_l - d_t\|_2 \approx \sigma$

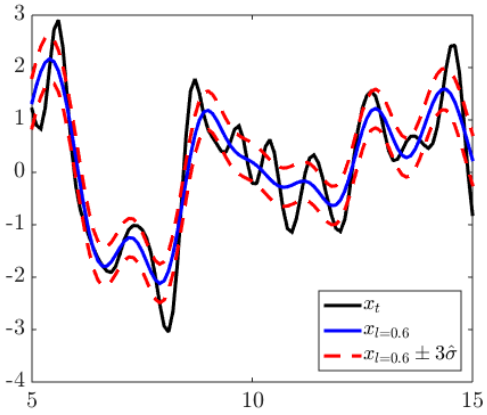


Figure 5.19: MAP estimate and posterior error interval with $l_M = 0.6$ alongside ground truth

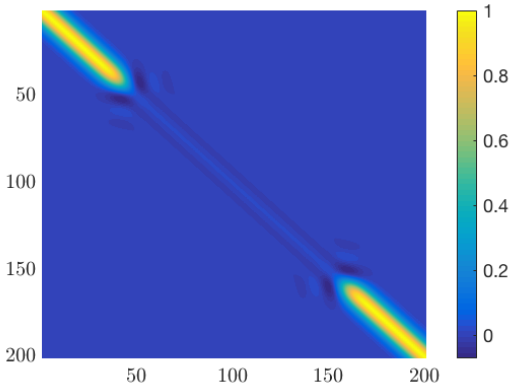


Figure 5.20: $\Gamma_{x|d}$ with $l_M = 0.6$

to control the norm of the estimates [78], while in this case we are attempting to adapt the L-curve criterion to control the correlation length/smoothness of the estimate. While there is no L shaped curve, a clear kink is still visible at $l = 0.2$. I take $l_L = 0.2$ to be the adapted L-curve estimate of $l_t = 0.3$.

The MAP and $3\hat{\sigma}$ posterior error interval for $l_L = 0.2$ is plotted in Figure 5.22. The estimates compare well when compared to the "best case" reconstruction of Section 5.1.1 which used the true $l_t = 0.3$. The posterior covariance $\Gamma_{x|d}$ for $l_L = 0.2$ is visualised in Figure 5.23.

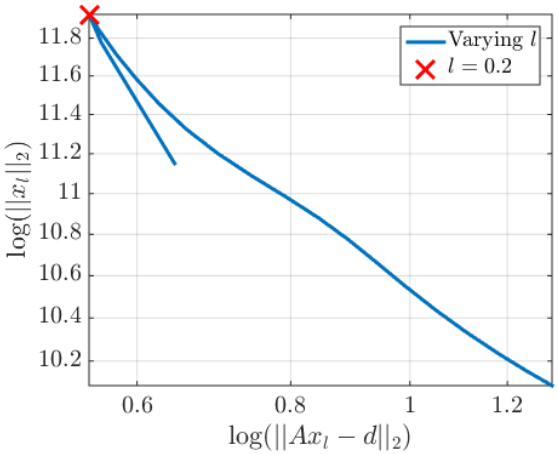


Figure 5.21: The "L-curve" for l

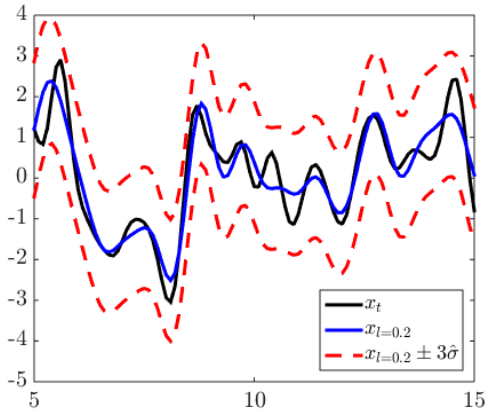


Figure 5.22: MAP estimate and posterior error estimate with $l_L = 0.2$ alongside ground truth

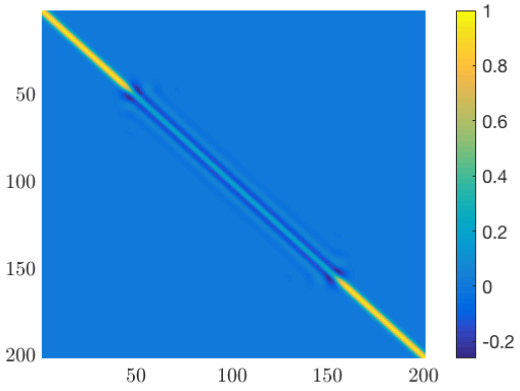


Figure 5.23: $\Gamma_{x|d}$ with $l_L = 0.2$

The adapted versions of the Morozov discrepancy principle and L-curve criterion I defined performed fairly well for this problem. This problem involved evaluating the single parameter $l \in \mathbb{R}$ i.e. $n_p = 1$. Generalisations of the Morozov discrepancy principle and L-curve criterion for $n_p > 1$ can be found in the literature e.g. [77].

5.1.3 Direct Filter Construction

Recall that for Gaussian priors, draws can be computed as $x_j = L_x w_j + \mu_x$ where w_j is a draw of $w \sim \mathcal{N}(0_{n_x,1}, I_{n_x})$ and $L_x L_x^T = \Gamma_x$. The matrix L_x is sometimes called the *information filter*. Similarly, a matrix $\tilde{L}_x$ such that $w_j = \tilde{L}_x(x_j - \mu_x)$ is the *innovation filter* or *whitening filter* [4, 79]. Note that $\tilde{L}_x^T \tilde{L}_x = \Gamma_x^{-1}$.

It may be that a filter is constructed without first constructing the covariance. In this thesis, constructing a filter before the covariance is called *direct filter construction*.

Suppose we wished to have a prior on x such that x is smooth. We could construct Γ_x using a smoothness covariance function as in section 5.1.1. However, we could instead reuse our convolution operator F from Section 2.3, which we know smooths data i.e. let $L_x = F$ and $\Gamma_x = FF^T$. A draw $x_j = Fw_j$ is shown in Figure 5.24. Observe that the draw is smooth.

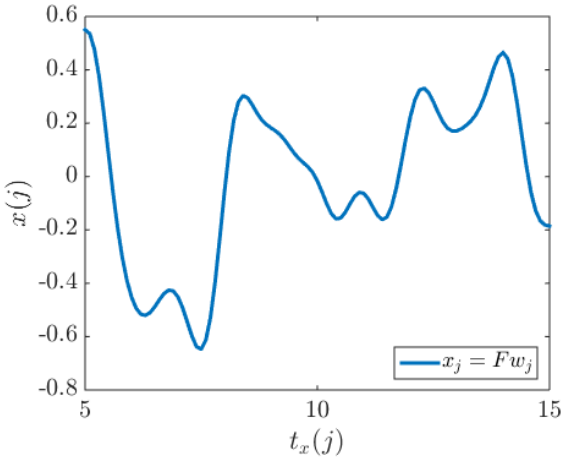


Figure 5.24: Draw from the prior, where L_x was constructed directly as a convolution operator

When constructing a filter directly, it can be easy to accidentally impose additional structure on x. For example, by taking $L_x = F$ where F is the convolution matrix of Section 2.3, draws will be near 0 near the computational domain edges $\mathbf{t} = 0$ and $\mathbf{t} = 20$. We avoided this problem by restricting to the region of interest $\mathbf{t} \in [5, 15]$.

Another method of implementing a smoothing filter would be to use a (fast) Fourier transform method for convolution. More information on the fast Fourier transform (FFT) can be found in [32]. Draws can be computed as

$$x_j = L_x w_j = \mathcal{F}^{-1}\Big(k_f \odot \mathcal{F}(w)\Big) \tag{5.21}$$

where $k_f = \mathcal{F}(k)$ is the Fourier transform of the convolution kernel and $\odot$ is the *Hadamard*

product. More details on this result can be found in [80, 81]. The Hadamard product of $B_1 \in \mathbb{R}^{n\times m}$ and $B_2 \in \mathbb{R}^{n\times m}$ is $B_3 = B_1 \odot B_2$ where $B_3(j,k) = B_1(j,k) \times B_2(j,k)$.

The FFT filter requires $k_f \in \mathbb{R}^{n_x}$ be stored in memory, and draws can be evaluated in $\mathcal{O}(n_x \log(n_x))$ flops [32]. Compare this to storing $L_x \in \mathbb{R}^{n_x \times n_x}$ in memory, and computing draws in $\mathcal{O}(n_x^2)$ flops. Recall that the convolution matrix F of Section 2.3 assumed the signal was 0 outside the domain. The Fourier transform formulation assumes the signal is *circulant* i.e. the signal represents 1 cycle of a periodic function [81].

The FFT formulation also gives a simple way of constructing an innovation filter. Consider

$$w_j = \tilde{L}_x x_j = \mathcal{F}^{-1}\left(\tilde{k}_f \odot \mathcal{F}(x)\right) \tag{5.22}$$

where

$$\tilde{k}_f(j) = \frac{1}{k_f(j)} \tag{5.23}$$

where we call k_f the *Fourier filter* and $\tilde{k}_f$ the *inverse Fourier filter*.

In the case that $k_f(j) = 0$, $\tilde{k}_f$ is undefined. We can instead construct a "regularised form"

$$\tilde{k}_{f,\kappa}(j) = \frac{k_f(j)}{k_f(j)^2 + \kappa^2} \tag{5.24}$$

where $\kappa \in \mathbb{R}$ is some positive number. Recall the similarity to the filter factors of standard Tikhonov regularisation discussed in Section 3.4. The above is a form of the *Wiener filter* [79].

Alternatively, we can form the *average inverse Fourier filter* from samples. We compute m samples w_j and x_j using the information filter, and inverse Fourier filter samples as

$$\tilde{k}_{f,j} = \frac{\mathcal{F}(w_j)}{\mathcal{F}(x_j)}. \tag{5.25}$$

or $\tilde{k}_{f,j} = 0$ when $\mathcal{F}(x_j) = 0$. We then compute

$$\tilde{k}_f = \frac{1}{m}\sum_{j=1}^{m} \tilde{k}_{f,j} \tag{5.26}$$

which we call the *sample average inverse Fourier filter*. This technique can also be used when we only have samples x_j and w_j without additional knowledge of the underlying prior. The concept of a Fourier filter constructed from samples is considered further in Section 6.8, and is applied in Chapters 8 and 9.

5.1.4 Structured Gaussian Priors

Suppose we have *structural information* about x. An example of structural information is knowing x to have larger values in a particular region. Such information can easily be incorporated into μ_x.

Consider the case that Ω_x can be divided into 2 subdomains $\Omega_{x,1}$ and $\Omega_{x,2}$ where $\Omega_x = \Omega_{x,1} \cup \Omega_{x,2}$ and $\Omega_{x,1} \cap \Omega_{x,2} = \emptyset$. Suppose x_{Ω_1}, the values of x in $\Omega_{x,1}$ have little relationship with x_{Ω_2}, the values of x in $\Omega_{x,2}$. We can express this with a Gaussian prior by assigning little cross correlation to x_{Ω_1} and x_{Ω_2}. This could be the case where x represents density, and Ω_1 and Ω_2 are the locations of 2 different materials.

Let $\Omega_x = [5, 15]$, and $x \sim \mathcal{N}(0_{n_x,1}, \Gamma_x)$. Let $\Omega_{x,1} = [5, 10]$ and $\Omega_{x,2} = [10, 15]$. Let

$$\Gamma_x(j,k) = \begin{cases} \left(\frac{h_1}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l_1^2}\right) & t_j, t_k \in [5, 10] \\ \left(\frac{h_2}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l_2^2}\right) & t_j, t_k \in (10, 15] \\ \left(\frac{h_3}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l_3^2}\right) & \text{otherwise} \end{cases} \tag{5.27}$$

i.e. different smoothness priors in and between Ω_{x_1} and Ω_{x_2}. If there is no correlation between Ω_{x_1} and Ω_{x_2},

$$\Gamma_x(j,k) = \begin{cases} \left(\frac{h_1}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l_1^2}\right) & t_j, t_k \in [5, 10] \\ \left(\frac{h_2}{3}\right)^2 \exp\left(\frac{-\|t_j - t_k\|_2^2}{2l_2^2}\right) & t_j, t_k \in (10, 15] \\ 0 & \text{otherwise} \end{cases} \tag{5.28}$$

which we call the $h_3 = l_3 = 0$ case, even though the general form in Equation (5.27) is not defined for $h_3 = l_3 = 0$.

Let $h_1 = 3$, $l_1 = 0.3$, $h_2 = 5$, $l_2 = 2$ and $h_3 = l_3 = 0$ i.e. the no cross correlation case. The covariance Γ_x is visualised in Figure 5.25. Note the correlation structure around element 101 i.e. $t(101) = 10$. This is an example of a nonstationary prior. A draw x_j using this prior is shown in Figure 5.26. Note the jump at $t = 10$ as we move into the new region, and the greater smoothness in the $(10, 15]$ region.

An important note to make with priors structured as above is that subdomain locations need to be known exactly when Γ_x is constructed. If the correlation structure of x changed at $t = 13$ instead of $t = 10$, draws in the $[10, 13]$ region would be far too smooth. Parameterised priors $\pi_x^p(x)$ were discussed in Section 5.1.2. The prior used above can be thought of as $p = (h_1, h_2, h_3, l_1, l_2, l_3, t^*)$ where $t^* = 10$ is the division between the regions. The methods of Section 5.1.2 could be applied to estimating t^*.

We now consider isotropic priors. Specifically, we will construct an isotropic Gaussian smoothness type prior in 2D. Let $\Omega = [0, 10] \times [5, 15]$ where $\mathbf{t}^{(1)} \in [0, 10]$ is the horizontal

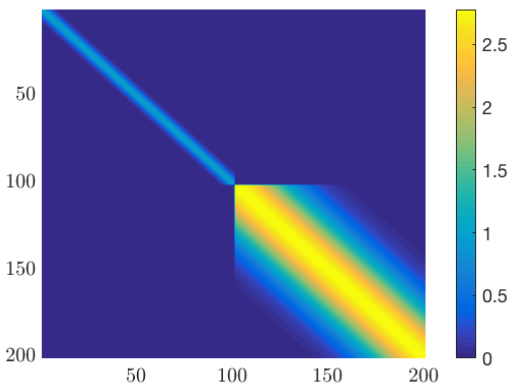


Figure 5.25: Structured prior covariance Γ_x

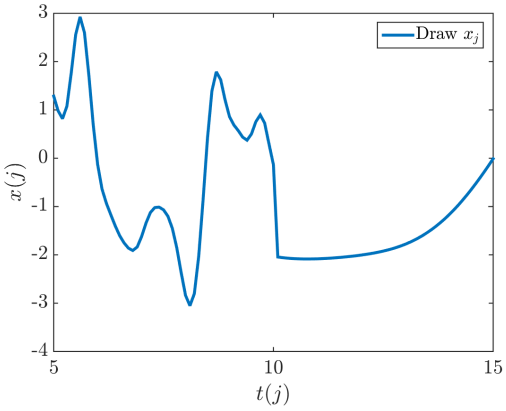


Figure 5.26: Draw x_j from structured prior

axis and $\mathbf{t}^{(2)} \in [5, 15]$ is the vertical axis. Let $\Delta_t = 0.2$. Let $t^{(1)} = (0, \Delta_t, 2\Delta_t, \ldots, 10) \in \mathbb{R}^{51}$ and $t^{(2)} = (5, 5 + \Delta_t, 5 + 2\Delta_t, \ldots, 15) \in \mathbb{R}^{51}$ be finite dimensional approximations of $\mathbf{t}^{(1)}$ and $\mathbf{t}^{(1)}$. Let $\mathbf{x}(\mathbf{t})$ be a function for $\mathbf{t} \in \Omega$. Let

$$\begin{aligned} x = \Big(& \mathbf{x}\big(t^{(1)}(1), t^{(2)}(1)\big), \mathbf{x}\big(t^{(1)}(1), t^{(2)}(2)\big), \ldots \\ & \ldots, \mathbf{x}\big(t^{(1)}(1), t^{(2)}(51)\big), \mathbf{x}\big(t^{(1)}(2), t^{(2)}(1)\big), \ldots \\ & \ldots, \mathbf{x}\big(t^{(1)}(51), t^{(2)}(51)\big) \Big) \in \mathbb{R}^{2601} \end{aligned} \tag{5.29}$$

be the *vectorised finite dimensional approximation* of $\mathbf{x}$.

Let us compute an isotropic smoothness prior on x with $h = 3$ and $l = 1$. The corresponding prior covariance Γ_x is visualised in Figure 5.27. The submatrix $\Gamma_x(1 : 510, 1 : 510)$ is represented in Figure 5.28. Note the 51×51 blocks of Γ_x that each look like 1D smoothness covariances e.g. in Figure 5.15. This is a consequence of the vectorisation of x.

We now compute a draw x_j from the above 2D isotropic smoothness prior. A side on view of x_j on Ω is shown in Figure 5.29 and a top down view in 5.30. Note the "bumps" with approximate height 3 and radius 1, corresponding to $h = 3$ and $l = 1$.

Rather than roughly symmetric "bumps", we may want "bands" in x. Consider the covariance function

$$\Gamma_x(j, k) = \left(\frac{h}{3}\right)^2 \exp\left(-\frac{1}{2}\left(\frac{\big(t^{(1)}(j) - (t^{(1)}(j)\big)^2}{l_1^2} + \frac{\big(t^{(2)}(j) - (t^{(2)}(j)\big)^2}{l_2^2}\right)\right) \tag{5.30}$$

i.e. the standard smoothness covariance function but with different correlation lengths for each

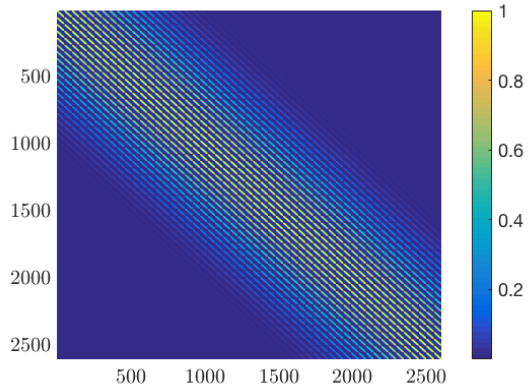


Figure 5.27: Isotropic smoothness Γ_x

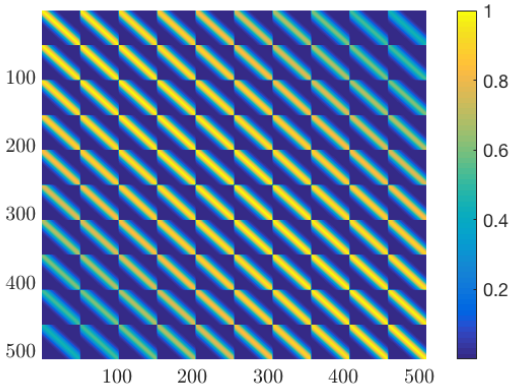


Figure 5.28: $\Gamma_x(1:510, 1:510)$

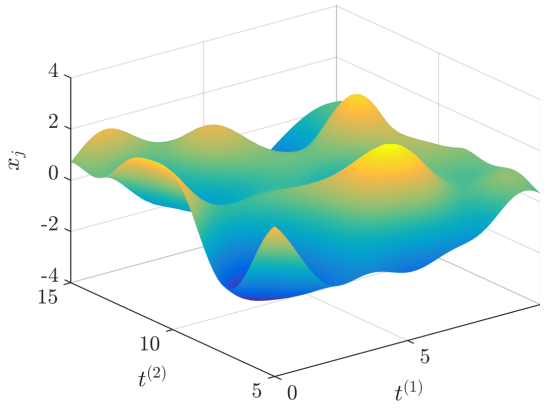


Figure 5.29: Draw from 2D smoothness prior

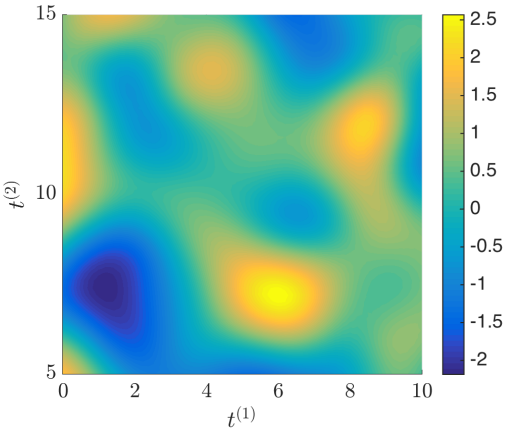


Figure 5.30: Top down view

axis. We rewrite Equation (5.30) as

$$\Gamma(i,j) = \left(\frac{h}{3}\right)^2 \exp\left(-\frac{1}{2}\left\|W(t_i - t_j)\right\|_2^2\right) \tag{5.31}$$

where

$$W = \begin{pmatrix} \frac{1}{l_1} & 0 \\ 0 & \frac{1}{l_2} \end{pmatrix}. \tag{5.32}$$

Now suppose we wish to rotate the bands by angle θ. This can be achieved by using a

rotation matrix

$$R_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{5.33}$$

to form the covariance matrix

$$\Gamma(j,k) = \frac{h^2}{3} \exp\left(-\frac{1}{2} \left\| W_\theta (t_i - t_j) \right\|_2^2 \right) \tag{5.34}$$

where $W_\theta = W R_\theta$.

We now construct the rotated anisotropic smoothness prior with $l_1 = 2$, $l_2 = 0.2$ and $\theta = \frac{\pi}{4}$ by Equation (5.34). The corresponding covariance Γ_x is visualised in Figure 5.31 and $\Gamma_x(1:510, 1:510)$ is represented in Figure 5.32. A draw x_j of this prior is shown side on in Figure 5.33 and top down in fig 5.34. Observe that x_j has "bands" with approximate length $2l_1 = 4$ and width $2l_2 = 0.4$ rotated by $\theta = \frac{\pi}{4}$. Note that this prior has parameters $p = (h, l_1, l_2, \theta)$, but the locations of the bands are not explicitly parameterised.

5.2 L-1 Priors

Gaussian priors are of the form

$$\pi(x) \propto \exp(-\frac{1}{2} \left\| \tilde{L}_x (x - \mu_x) \right\|_2^2) \tag{5.35}$$

whereas *L-1 priors* are of the form

$$\pi(x) \propto \exp(-\frac{1}{2} \left\| \tilde{L}_x (x - \mu_x) \right\|_1) \tag{5.36}$$

where we note the change to the *L-1 norm*

$$\|x\|_1 = |x(1)| + |x(2)| + \cdots + |x(n_x)|. \tag{5.37}$$

L-1 priors are also known as *Laplace priors* [65]. The term L-1 prior is used often in inverse problem literature such as [3, 4].

To draw samples from a distribution $\pi(x)$, first compute the *cumulative distribution* $F_\pi(x) = \int_{-\infty}^{x} \pi_x(s) ds$, a nondecreasing function from 0 to 1. A sample x_j is computed with *inverse probability transform sampling* as $x_j = F_\pi^{-1}(u_j)$ where u_j is a sample of $u \sim \mathcal{U}(0,1)$. More discussion on inverse transform sampling can be found in [82].

Let $\pi(x) = \exp(-\alpha \|x\|_1)$ with $x > 0$. This is the *positive constrained L-1 prior*, and has inverse cumulative distribution $F_\pi^{-1}(u) = -\frac{1}{\alpha} log(1-u)$. A draw on a 1-D domain with $\alpha = 1$, $n_x = 200$ is shown in Figure 5.35. A draw on a 2-D domain with $\alpha = 1$, $n_x = 1,000$ is shown in Figure 5.36.

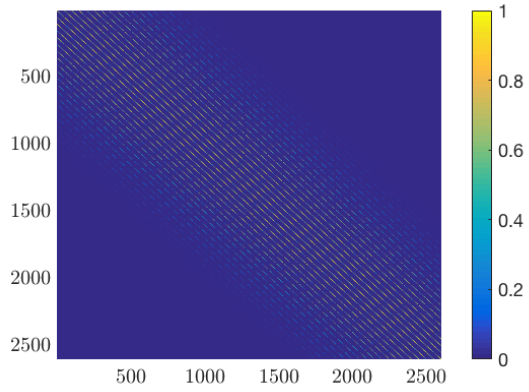


Figure 5.31: Anisotropic Γ_x

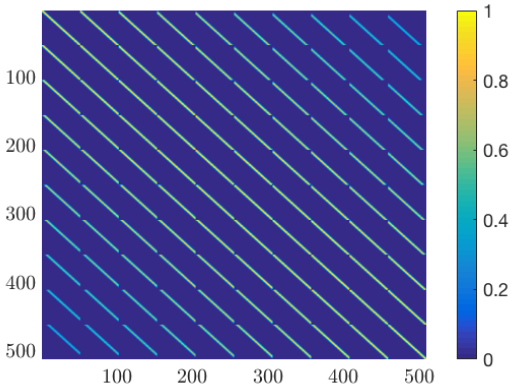


Figure 5.32: $\Gamma_x(1:510, 1:510)$

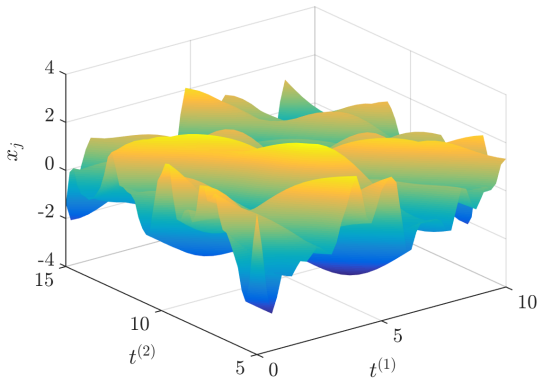


Figure 5.33: Draw x_j, side view

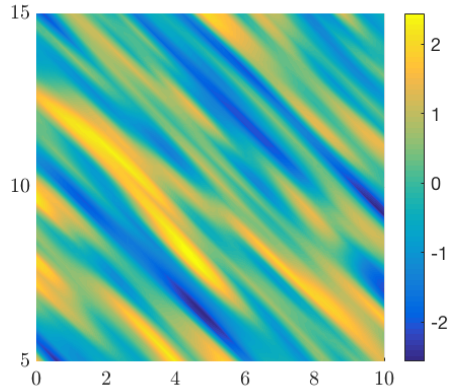


Figure 5.34: Draw x_j, overhead view

Consider the generalised L-1 prior $\pi(x) \propto \exp(-\frac{1}{2}\left\|\tilde{L}_x(x-\mu_x)\right\|_1)$. Note that $\tilde{L}_x^T\tilde{L}_x \neq \Gamma_x^{-1}$. Let $d = A(x) + e$ where $e \sim \mathcal{N}(\mu_e, \Gamma_e)$. The posterior distribution is

$$\pi_{x|d}(x|d_t) \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_e(d_t - A(x) - \mu_e)\right\|_2^2 + \left\|\tilde{L}_x(x-\mu_x)\right\|_1\right)\right) \tag{5.38}$$

with MAP

$$x_{\text{MAP}} = \min_x \left\{\left\|\tilde{L}_e(d_t - A(x) - \mu_e)\right\|_2^2 + \left\|\tilde{L}_x(x-\mu_x)\right\|_1\right\} \tag{5.39}$$

which is not equal to the CM. The MAP estimate with L-1 prior and Gaussian noise is also known as the LASSO estimate in statistics [83]. Even if $A(x) = Ax$, the posterior distribution

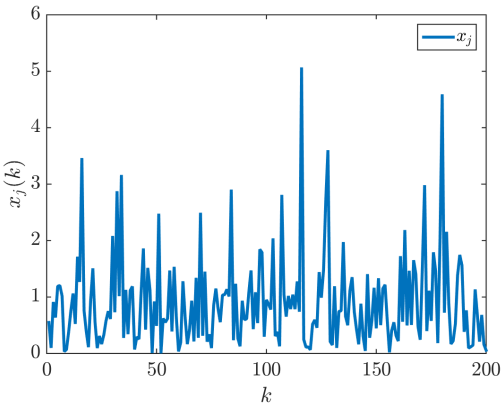


Figure 5.35: 1-D L-1 prior draw

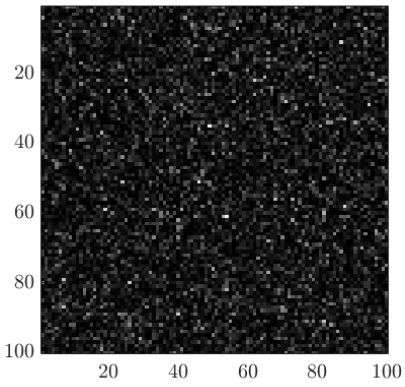


Figure 5.36: 2-D L-1 prior draw

is not Gaussian when an L-1 prior is used, so the MAP cannot be found as the solution to a linear algebra problem. Computationally efficient methods of finding the MAP by convex optimisation can be found in [84, 85].

L-1 priors are sometimes used to *promote sparsity* i.e. the estimate x_{MAP} formed as in Equation (5.40) with $\mu_x = 0_{n_x,1}$ is expected to have a few relatively large elements, but most elements near 0. Examples of the sparsity promotion in practice can be found in e.g. [86, 87, 88]. This is reflected in the L-1 draws of Figures 5.35 and 5.36. A visual explanation of how the L-1 norm promotes sparsity better than the L-2 norm is given in Figure 5.37. The element of a set e.g. X_{ls} with smallest L-1 norm is likely sparse. Note that the Lasso estimate is not in the set X_{ls}, with Figure 5.37 demonstrating a simplified case. The case described in Figure 5.37 corresponds to

$$\lim_{\alpha \downarrow 0} x_{\mathrm{MAP}} = \lim_{\alpha \downarrow 0} \min_{x} \left\{ \|d_t - A(x)\|_2^2 + \alpha \|x\|_1 \right\} \tag{5.40}$$

similar to the pseudoinverse $x^\dagger = A^\dagger d$ estimate.

5.3 Total Variation

Another useful type of exponential prior is the *total variation* (TV) prior. A thorough review of total variation in the context of image analysis can be found on [89]. Let $\mathbf{x}(\mathbf{t})$ with $\mathbf{t} \in \mathbb{R}$ be a differentiable function on Ω. The total variation of $\mathbf{x}$ on Ω is

$$\mathbf{TV}(\mathbf{x}) = \int_{\Omega} |\mathbf{x}'(\mathbf{t})| d\mathbf{t} \tag{5.41}$$

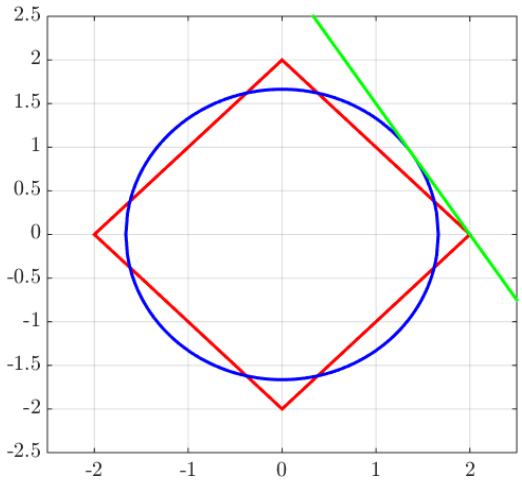


Figure 5.37: The green line is a set of points X. The red diamond is the set of points Y_1 where for all $y_1 \in Y_1$, $\|y_1\|_1 = \alpha_1$. The blue circle is the set of points Y_2 where for all $y_2 \in Y_2$, $\|y_2\|_2 = \alpha_2$. The point $x_1 \in X$ at the intersection of the green line and red diamond is the element of X with minimum L-1 norm. The point $x_2 \in X$ at the intersection of the green line and blue circle is the element of X with minimum L-2 norm.

and the total variation prior on $\mathbf{x}$ is

$$\pi_{\mathbf{x}}(\mathbf{x}) \propto \exp\big(\alpha \mathbf{TV}(\mathbf{x})\big) \tag{5.42}$$

with $\alpha \in \mathbb{R}$.

Let x and t be finite dimensional approximations to $\mathbf{x}$ and $\mathbf{t}$ i.e. $x(j) = \mathbf{x}(t(j))$. Let t be equispaced i.e. $t(j+1) - t(j) = \Delta_t$. A finite dimensional approximation to the TV operator is

$$\mathrm{TV}(\mathrm{x}) = \|T_+ x\|_1 + \|T_- x\|_1 \tag{5.43}$$

where

$$T_+ = \frac{\Delta_t}{2} \begin{pmatrix} 1 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n_x \times n_x} \tag{5.44}$$

and $T_- = T_+^T$. Note that we are taking a central difference estimate of the total variation, whereas a purely forward or backwards difference approach could be implemented as an L-1 prior. Note that only in the forward/backwards, single dimensional case can the TV prior be implemented as a purely L-1 type prior, as other cases require a summation of L-1 terms.

Let x be a 1-D signal. We observe data d as $d = x + e$ where $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I)$. Let a particular realisation be $d_t = x_t + e_t$. Estimating x_t from d_t is called denoising. The MAP estimate of x_t using a TV prior is

$$x_{\text{MAP},\alpha} = \min_x \left\{ \left\| \frac{1}{\sigma_e}(d - x) \right\|_2^2 + \alpha \text{TV}(x) \right\} \tag{5.45}$$

where α is some positive number. Computing the MAP estimate as in Equation (5.45) is also known as *total variation denoising*.

We solve the nonlinear optimisation problem of Equation (5.45) with a gradient based method. This requires derivatives of the TV functional. Note that the TV functional in Equation (5.52) makes use of the absolute value function $f(\tau) = |\tau|$ where $\tau \in \mathbb{R}$ in this section. The absolute value function is not differentiable at $\tau = 0$. In this thesis, the approximation

$$|\tau| \approx g_\rho(\tau) = \frac{1}{\rho}\log\big(\cosh(\rho\tau)\big) \tag{5.46}$$

is used. Increasing $\rho \in \mathbb{R}$ improves the quality of the approximation. The derivative g'_ρ of g_ρ is

$$\frac{d}{d\tau}\left(\frac{1}{\rho}\log\big(\cosh(\rho\tau)\big)\right) = g'_\rho(\tau) = \tanh(\rho\tau). \tag{5.47}$$

Shown in Figure 5.38 is $f(\tau) = |\tau|$ alongside g_ρ with $\rho = 10$ and $\rho = 200$. Corresponding derivatives are shown in Figure 5.39 although we define $f'(0) = 0$. Consider finding the minimum of f with a derivative based method. Based on Figure 5.39, a line search might use small ρ initially, to avoid "jumping over" the minimum.

I compute the MAP of Equation (5.45) by *gradient descent*. A review of gradient descent can be found in [46]. The gradient ϕ_k of the functional in Equation (5.45) at x_k, the k'th iterate of x, making the approximation of Equation (5.46) is

$$\phi_k(j) = \frac{\partial}{\partial x_k(j)}\left(\left\|\frac{1}{\sigma_e}(d_t - x_k)\right\|_2^2 + \alpha \text{TV}(\text{x}_\text{k})\right) \tag{5.48}$$

$$\approx \frac{2}{\sigma_e^2}\Big(x_k(j) - d_t(j)\Big) + \frac{\alpha}{\Delta_t}\left(\tanh\Big(\rho\big(x_k(j) - x_k(j-1)\big)\Big) + \tanh\Big(\rho\big(x_k(j) - x_k(j+1)\big)\Big)\right) \tag{5.49}$$

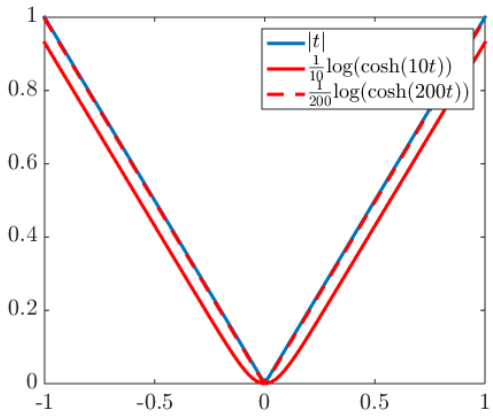


Figure 5.38: Absolute value function $f(\tau) = |\tau|$ alongside approximations $g_\rho(\tau)$ with $\rho = 10$ and $\rho = 200$

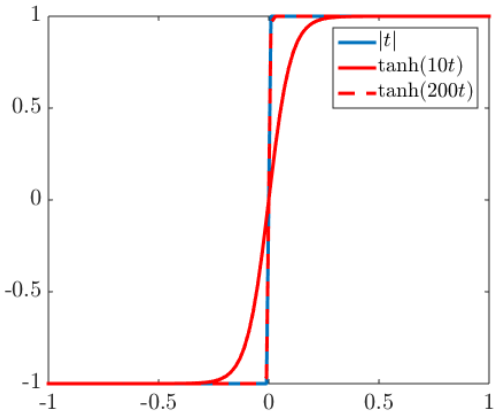


Figure 5.39: $f'(\tau)$ with $f'(0) = 0$ alongside $g'_\rho(\tau)$ with $\rho = 10$ and $\rho = 200$

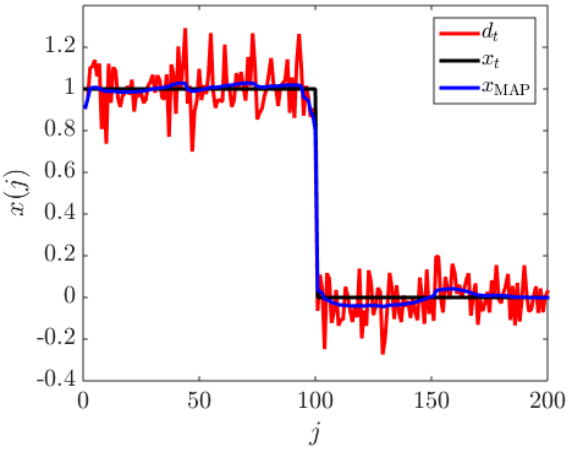


Figure 5.40: True, noisy and TV MAP signals

which is used to iteratively approximate the MAP as

$$x_{k+1} = x_k + \beta\phi_k \tag{5.50}$$

where β is the step length. At each iterate, a larger value of ρ is used.

The particular realisation $d_t = x_t + e_t$ is plotted in Figure 5.40. The MAP computed with steepest descent as described above is also shown. Note that the TV MAP estimate is generally

"flat", and captures the sharp drop in x_t at $t = 100$.

Let us consider use of TV regularisation on 2-D images. Let $d = x + e$ where x is a (vectorised) 2-D image, d is a noisy image and $e \sim \mathcal{N}(0_{n_d,1}, \Gamma_e)$ is additive noise. The MAP estimate for a particular realisation $d_t = x_t + e_t$ is

$$x_{\mathrm{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - x) \right\|_2^2 + \alpha TV(x) \right\} \tag{5.51}$$

as in the 1-D case, but with the 2-D total variation functional

$$\mathrm{TV(x)} = \|T_{\mathrm{left}} x\|_1 + \|T_{\mathrm{right}} x\|_1 + \|T_{\mathrm{up}} x\|_1 + \|T_{\mathrm{down}} x\|_1 \tag{5.52}$$

where T_{left} takes the difference to the left pixel of $x(j)$, T_{up} takes the difference to the pixel above, and so on.

The ground truth x_t is shown in Figure 5.41. This is a subsection of the common "Barbara" test image. The noiseless $e_t = 0$ TV MAP "estimate" with large $\alpha = \alpha_1$

$$x_{\mathrm{MAP,e_t=0},\alpha_1} = \min_x \left\{ \left\| \tilde{L}_e(d_t - x) \right\|_2^2 + \alpha_1 TV(x) \right\} \tag{5.53}$$

is shown in Figure 5.43. Note the "cartoonish" quality of the image, which now consists of a few distinct patches of colour.

Now consider estimating x_t from $d_t = x_t + e_t$ where $e_t \neq 0_{n_d,1}$ is a realisation of $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$. The data d_t is shown in Figure 5.42. The noiseless TV MAP "estimate" with $\alpha = \alpha_2 \leq \alpha_1$ is shown in Figure 5.44. Note that this is less "cartoonish" than the noiseless TV MAP estimate with α_1 of Figure 5.43. The noisy TV MAP estimate with $\alpha = \alpha_2$ is shown in Figure 5.46. Notice how much of the noisy "fuzz" has been removed, and the image only looks slightly "cartoonish". The noisy TV MAP estimate with $\alpha = \alpha_1$ is shown in Figure 5.45. The noisy "fuzz" is gone, but the image is very "cartoonish". Note that the noiseless and noisy MAP with $\alpha = \alpha_1$ in Figures 5.43 and 5.45 are very similar.

Another interesting image processing application is reconstructing an image when many pixels have been masked. From an original image x_t, "masked" data is $d_t = Mx_t$ where the "mask" M sets various elements to 0. In this case, we will be masking half of the pixels. The MAP estimate

$$x_{\mathrm{MAP}} = \min_x \left\{ \|M(d_t - x)\|_2^2 + \alpha \mathrm{TV}(x) \right\} \tag{5.54}$$

is the "reconstructed" image. The masked image d_t is shown in Figure 5.47. The reconstruction is shown in Figure 5.48. In this case the mask M is known exactly and there is no noise. In practical applications e.g. removing scratches from images, there will be noise and M has to be estimated.

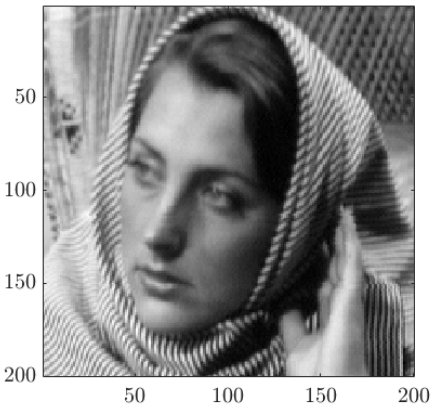


Figure 5.41: x_t, a subsection of "Barbara"

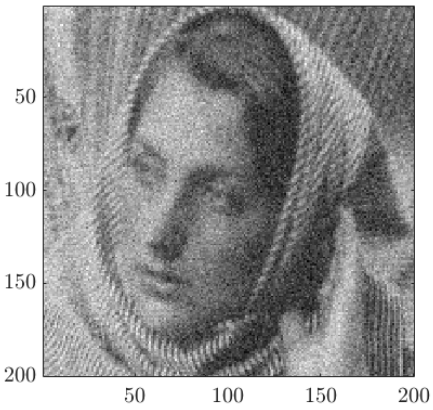


Figure 5.42: Noisy data d_t

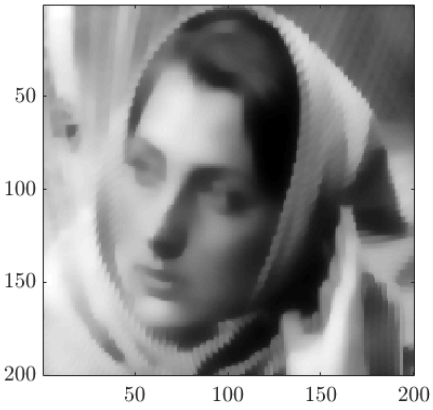


Figure 5.43: Noiseless TV MAP "estimate" $x_{\mathrm{MAP},\mathrm{e_t}=0,\alpha_1}$

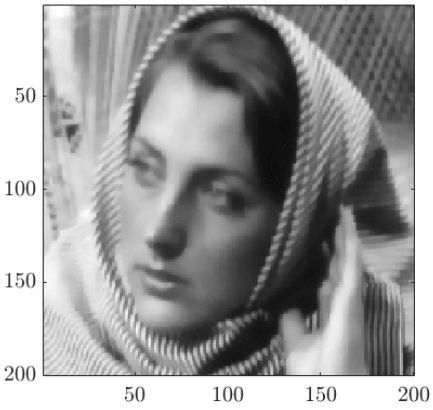


Figure 5.44: Noiseless TV MAP "estimate" $x_{\mathrm{MAP},\mathrm{e_t}=0,\alpha_2}$

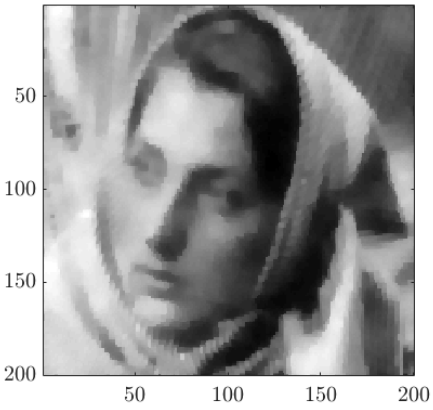


Figure 5.45: $\alpha = \alpha_1$ TV MAP estimate $x_{\mathrm{MAP},\alpha_1}$

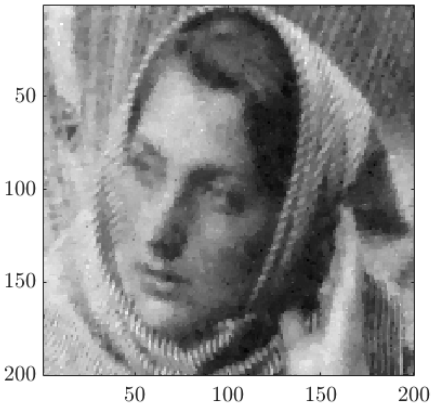


Figure 5.46: $\alpha = \alpha_2$ TV MAP "estimate" $x_{\mathrm{MAP},\alpha_2}$

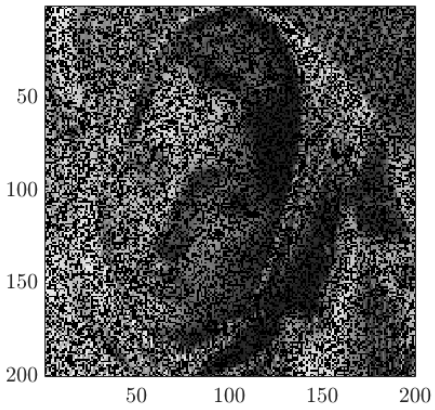


Figure 5.47: Masked "Barbara"

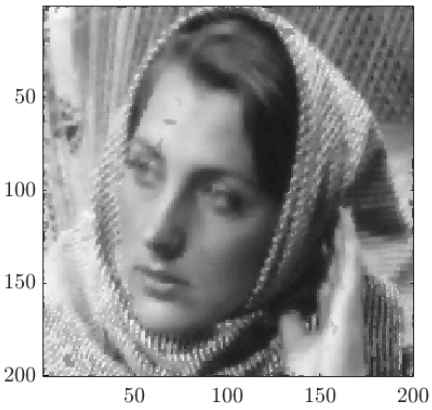


Figure 5.48: TV Reconstruction

It is important to note that the total variation prior is not *discretisation invariant* [90]. Changing the dimension of the unknown affects the correlation structure of the solution. Discretisation invariance is discussed further in Section 6.3.2. More details on discretisation invariance of priors can be found in [91]. There are methods of making the TV prior discretisation invariant, in particular, the typical TV functional can be synthesised with a Gaussian type prior as in [92].

5.4 Differential Priors

Another useful family of priors to consider are "differential" priors, in which the prior on x is in terms of a differential equation. For example, the TV prior on $\mathbf{x}$ as defined in equation 5.41 as a differential equation and is therefore a differential type prior. The covariance functions of Gaussian priors are considered in terms of differential equations in [73] and [93]. A general review of differential priors can be found in [94]. Explicit use of differential priors is beyond the scope of this thesis, although results of analysis of priors as differential equations will be used, particularly in Section 6.5.

Chapter 6

Computational Methods

Let $d = A_e(x, e)$ be a model, and $d_t = A_e(x_t, e_t)$ be a particular realisation. Estimating x_t from d_t and quantifying the uncertainty in the estimate is often computationally intensive [4, 36]. This chapter presents methods of reducing this computational cost.

The computational cost is considered at the offline and online stage. The online stage is observing d_t and computing an estimate of x_t, possibly with uncertainty quantification. The offline stage is everything leading up to the online stage. The offline stage will typically involve running simulations, tuning parameters, comparing models and so on.

Each phase has different computational considerations. Consider [95] in which EIT measurements are taken of a pipe to reconstruct the internal flow. During the offline phase, as much computational power as the research institute can provide is available, and available time for computations is on the order of funding rounds. In the online phase, on the factory floor, reconstructions need to compute within milliseconds on cheap commercial hardware [96, 97].

The *recursive QR* method is proposed in Section 6.4.2. Recursive QR is a method of computing estimates of statistical quantities from samples, parallel with samples being computed. This method is developed for scenarios in which samples are computed relatively slowly.

The concept of *local sample compression* is proposed in Section 6.6, extending the work of [37]. The local sample compression method finds a low rank approximation to an operator in a particular region. In the context of inverse problems, local sample compression allows us to form a low rank approximation to the forward operator A in the effective support of the prior $\pi_x(x)$.

The methods of this chapter are adapted and updated with the aim of being applied together to inverse problems. How to apply the methods to a general inverse problem is explained in Chapter 7. The methods are then applied together in Chapters 8, 9 and 10.

6.1 Simple Example of Precomputations

Computations at the offline phase can be used to reduce the computations required at the online phase. This section presents a trivial example to demonstrate the concept.

Consider the finite dimensional linear A Gaussian x, e and additive noise model $d = Ax+e$ where $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $e \sim \mathcal{N}(0_{n_d,1}, \Gamma_e)$. Let $d_t = Ax_t + e_t$ be a particular realisation. The MAP estimate of x_t from d_t is

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - Ax) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{6.1}$$

$$= \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_e d_t \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.2}$$

$$= \begin{pmatrix} B_1 & B_2 \end{pmatrix} \begin{pmatrix} \tilde{L}_e d_t \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.3}$$

$$= B_1 \tilde{L}_e d + B_2 \tilde{L}_x \mu_x \tag{6.4}$$

$$= \hat{B} d + b \tag{6.5}$$

where $B = \begin{pmatrix} B_1 & B_2 \end{pmatrix} = \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix}^\dagger$, $\hat{B} = B_1 \tilde{L}_e$ and $b = B_2 \tilde{L}_x \mu_x$. Note that $\hat{B} \in \mathbb{R}^{n_x \times n_d}$ and $b \in \mathbb{R}^{n_x}$ can be computed offline. The online computation of x_{MAP} is computing the product $\hat{B}d$ and adding b. Recall that the posterior covariance is

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A^T \Gamma_e^{-1} A)^{-1} \tag{6.6}$$

which can also be computed at the offline phase. If $\hat{\sigma} = \text{diag}(\Gamma_{x|d})$ is used to compute posterior error intervals as e.g. $x_{\text{MAP}} \pm 3\hat{\sigma}$, then the online uncertainty quantification computation is adding/subtracting $3\hat{\sigma}$ from the MAP.

6.2 Improper Priors and Reduced Bases

Let $x \sim \mathcal{N}(\mu_x, \Gamma_x)$. This can be expressed as

$$\pi(x) = \exp(-\frac{1}{2} \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2) \tag{6.7}$$

where $\tilde{L}_x^T \tilde{L}_x = \Gamma_x^{-1}$. Note that Γ_x^{-1} exists if and only if Γ_x is full rank. A prior with rank deficient Γ_x is a type of *improper prior*. A proper prior is a prior such that

$$\int_{\mathbb{R}^{n_x}} \pi_x(x) dx = 1 \tag{6.8}$$

and a prior that does not satisfy the above is an improper prior. More details on proper and improper priors can be found in [98].

Let

$$\Gamma_x = U_x \Lambda_x U_x^T \tag{6.9}$$

$$= \begin{pmatrix} U_{x,r_{x_t}} & U_{x,\text{null}} \end{pmatrix} \begin{pmatrix} \Lambda_{x,r_{x_t}} & 0_{r_{x_t},n_x-r_{x_t}} \\ 0_{n_x-r_{x_t},r_{x_t}} & 0_{n_x-r_{x_t},n_x-r_{x_t}} \end{pmatrix} \begin{pmatrix} U_{x,r_{x_t}}^T \\ U_{x,\text{null}}^T \end{pmatrix} \tag{6.10}$$

$$= U_{x,r_{x_t}} \Lambda_{x,r_x} U_{x,r_{x_t}}^T \tag{6.11}$$

be the eigendecomposition of rank r_{x_t} prior covariance matrix Γ_x. For the rest of this discussion, the effective rank r_x is considered rather than the true rank r_{x_t}.

Consider a pseudoinverse type approach for Γ_x rank deficient. An innovation filter $\tilde{L}_{x,r_x} \in \mathbb{R}^{r_x \times n_x}$ can be formed as

$$\tilde{L}_{x,r_x} = \Lambda_{x,r_x}^{-\frac{1}{2}} U_{x,r_x}^T \tag{6.12}$$

as $\tilde{L}_{x,r_x}^T \tilde{L}_{x,r_x} = \Gamma_x^\dagger$. A prior might be attempted of the form

$$\pi_x(x) = b\text{exp}(-\frac{1}{2} \left\| \tilde{L}_{x,r_x}(x - \mu_x) \right\|_2^2). \tag{6.13}$$

where $b \in \mathbb{R}$ is some constant. However, there is no b such that the $\int_{\mathbb{R}^{n_x}} \pi_x(x)dx = 1$. Note that

$$\mu_x = \max_x \{\pi_x(x)\} \tag{6.14}$$

and

$$\pi_x(\mu_x) = b. \tag{6.15}$$

Let $x_{\text{null}} = \sum_{j=r_x+1}^{n_x} b_j u_j$ be an element of the nullspace of Γ_x, where the $b_j \in \mathbb{R}$ are arbitrary. Note that $\tilde{L}_{x,r_x}(x - \mu_x + x_{\text{null}}) = \tilde{L}_{x,r_x}(x - \mu_x)$ i.e. x and $x + x_{\text{null}}$ are "equally likely" with this prior. However, x with a x_{null} component would be expected to have 0 probability. The pseudoinverse of the covariance should therefore not be used for constructing the innovation filter, as such a filter assigns no penalty to unwanted x_{null} components.

Suppose Γ_x has rank $r_x < n_x$. Such a covariance could be replaced by

$$\Gamma_x \approx \Gamma_{x,+\kappa} = \Gamma_x + \kappa I_{n_x} \tag{6.16}$$

where $\kappa \in \mathbb{R}$ is a small positive number. Note that $\Gamma_{x,+\kappa}$ is a positive definite matrix, so Γ_x^{-1} exists and filters can be computed.

Another method of ensuring a positive definite covariance is to introduce a *penalty subspace*. The penalty subspace approximation is

$$\Gamma_x \approx \Gamma_{x,\kappa^2} = \Gamma_x + \kappa^2 U_{x,\text{null}} U_{x,\text{null}}^T \tag{6.17}$$

$$= \Gamma_x + \kappa^2 (I_{n_x} - U_{x,r_x} U_{x,r_x}^T) \tag{6.18}$$

$$= U_{x,r_x} \Lambda_{x,r_x} U_{x,r_x}^T + \kappa^2 U_{x,\text{null}} U_{x,\text{null}}^T \tag{6.19}$$

$$= \begin{pmatrix} U_{x,r_x} & U_{x,\text{null}} \end{pmatrix} \begin{pmatrix} \Lambda_{x,r_x} & 0_{r_x,n_x-r_x} \\ 0_{n_x-r_x,r_x} & \kappa^2 I_{n_x-r_x} \end{pmatrix} \begin{pmatrix} U_{x,r_x}^T \\ U_{x,\text{null}}^T \end{pmatrix} \tag{6.20}$$

where $\kappa \in \mathbb{R}$ is a small positive number. Note that

$$L_{x,\kappa} = \begin{pmatrix} U_{x,r_x} & U_{x,\text{null}} \end{pmatrix} \begin{pmatrix} \Lambda_{x,r_x}^{\frac{1}{2}} & 0_{r_x,n_x-r_x} \\ 0_{n_x-r_x,r_x} & \kappa I_{n_x-r_x} \end{pmatrix} \tag{6.21}$$

$$\Gamma_{x,\kappa^2}^{-1} = \begin{pmatrix} U_{x,r_x} & U_{x,\text{null}} \end{pmatrix} \begin{pmatrix} \Lambda_{x,r_x}^{-1} & 0_{r_x,n_x-r_x} \\ 0_{n_x-r_x,r_x} & \frac{1}{\kappa^2} I_{n_x-r_x} \end{pmatrix} \begin{pmatrix} U_{x,r_x}^T \\ U_{x,\text{null}}^T \end{pmatrix} \tag{6.22}$$

$$\tilde{L}_{x,\kappa} = \begin{pmatrix} \Lambda_{x,r_x}^{\frac{-1}{2}} & 0_{r_x,n_x-r_x} \\ 0_{n_x-r_x,r_x} & \frac{1}{\kappa} I_{n_x-r_x} \end{pmatrix} \begin{pmatrix} U_{x,r_x}^T \\ U_{x,\text{null}}^T \end{pmatrix}. \tag{6.23}$$

Another method of ensuring a positive definite covariance is to explicitly remove the nullspace of Γ_x. Let

$$a = U_{x,r_x}^T x \tag{6.24}$$

with $a \in \mathbb{R}^{r_x}$ be a new approximation of the unknowns. Note that

$$\Gamma_a = \Lambda_{x,r_x} \tag{6.25}$$

is full rank. This reduces the n_x dimensional problem of estimating x to the r_x dimensional problem of estimating a. Let $\hat{a}$ be an estimate of ground truth a_t. Then the estimate $\hat{x} = U_{x,r_x}\hat{a}$ of x_t has explicitly no nullspace component, in contrast with solutions found using the pseudoinverse filter $\tilde{L}_{x,r_x} = \Lambda_{x,r_x}^{-\frac{1}{2}} U_{x,r_x}^T$.

6.2.1 Basis Reduction Example

The use of a reduced basis in inverse problems is widespread. For example, the tSVD solution restricts estimates to the subspace spanned by the r largest singular vectors i.e. $x_r^\dagger = \sum_{j=1}^r a_j v_j$ where $a_j \in \mathbb{R}$ and $v_j \in \mathbb{R}^{n_x}$ is the j'th column of V_r in $A \approx A_r = U_r D_r V_r^T$, the rank r thin SVD of A. The proper orthogonal decomposition (POD) is another example of a reduced basis used in inverse problems, that can be extended to nonlinear operators [52, 53, 54].

The likelihood-informed subspace (LIS) [99] is an example of a potentially low dimensional subspace inverse problem estimates can be restricted to. Restricting A to a subspace results in a reduced order model $\hat{A}_r$, that is potentially faster to evaluate. Using reduced order models can be particularly useful in reducing the computational cost of MCMC [100]. Computation of reduced order models and bases is considered further in Section 6.6.

A common and interesting case is that the prior covariance is almost rank deficient. For example in a smoothness prior, increasing the correlation length increases the number of near-zero eigenvalues. Recall the 1D deconvolution problem with a Gaussian smoothness prior from Section 5.1.1. Consider the smoothness prior covariance matrices $\Gamma_{x,l}$. The eigenvalues of $\Gamma_{x,l}$ with $l = 0.1$, $l = l_t = 0.3$ and $l = 1$ are shown in Figure 6.1. Note that larger correlation lengths l correspond to a larger number of relatively small eigenvalues.

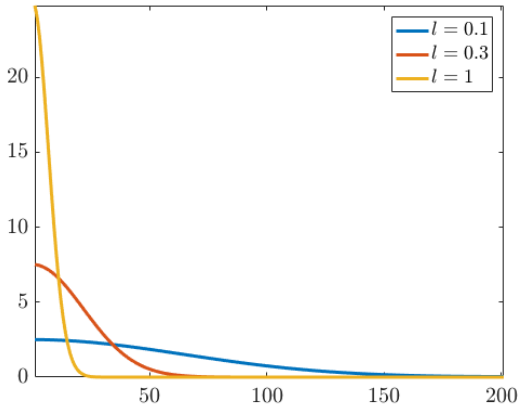


Figure 6.1: Eigenvalues of smoothness priors with different correlation lengths

Consider the 1D deconvolution problem of Section 2.3.2. Let Γ_x be a smoothness prior with $h = h_t = 3$ and $l = l_t = 0.3$. Note from Figure 6.1 that

$$\Gamma_x = U_x \Lambda_x U_x^T \approx U_{x,100} \Lambda_{x,100} U_{x,100}^T \tag{6.26}$$

i.e. $r_x = 100$. In this section, we estimate

$$a_t = U_{x,100}^T x_t \tag{6.27}$$

from data d_t where $d_t = Ax_t + e_t$. The MAP estimate of a_t is

$$a_{\text{MAP}} = \min_a \left\{ \left\| \tilde{L}_e(d_t - AU_{x,100}a) \right\|_2^2 + \left\| \Lambda_{x,100}^{-\frac{1}{2}} a \right\|_2^2 \right\} \tag{6.28}$$

$$= \begin{pmatrix} \tilde{L}_e A U_{x,100} \\ \Lambda_{x,100}^{-\frac{1}{2}} \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_e d_t \\ 0 \end{pmatrix} \tag{6.29}$$

$$= \hat{A}_a \hat{d}_a \tag{6.30}$$

where $\hat{A}_a \in \mathbb{R}^{(n_d+100)\times 100}$ and $\hat{d}_a \in \mathbb{R}^{n_d+100}$. This can be used to compute the estimate $x_{\text{MAP,a}} = U_{x,100} a_{\text{MAP}}$ of x_t. Uncertainty can be quantified by forming the posterior on a as

$$\Gamma_{a|d} = (\Gamma_a^{-1} + (AU_{x,100})^T \Gamma_e^{-1} AU_{x,100})^{-1} \tag{6.31}$$

which can be converted to a posterior on x as

$$\Gamma_{x_a|d} = U_{x,100} \Gamma_{a|d} U_{x,100}^T. \tag{6.32}$$

The estimate $x_{\text{MAP,a}}$ is shown in Figure 6.2 along with posterior error intervals taken from $\Gamma_{x_a|d}$. The estimates found without reducing the basis are shown in Figure 6.3.

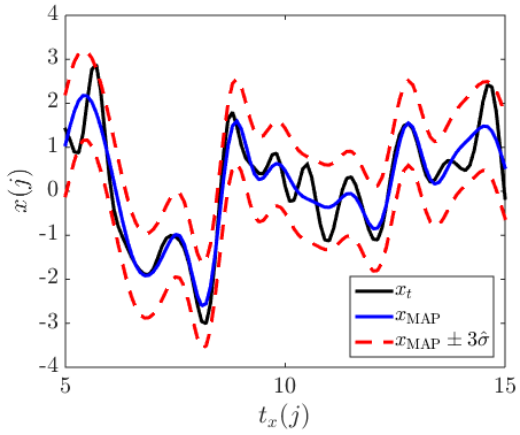


Figure 6.2: Reduced basis $r_x = 100$ estimates $x_{\text{MAP,a}}$, $\hat{\sigma}_{x,a} = \text{diag}(\Gamma_{x_a|d})$

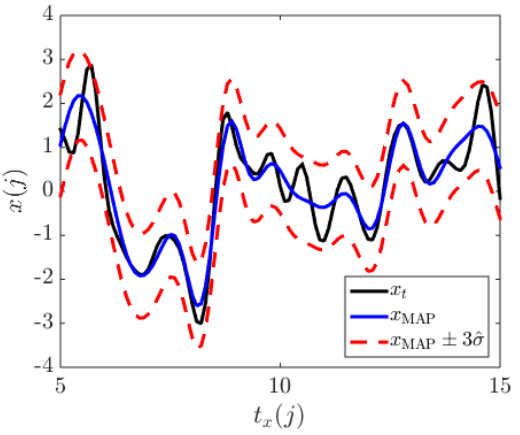


Figure 6.3: Complete basis $n_x = 201$ estimates x_{MAP}, $\hat{\sigma}_x = \text{diag}(\Gamma_{x|d})$

The above example is a linear problem with Gaussian error and prior models. Section 6.1 showed that computing the MAP at the online stage can be reduced to

$$x_{\text{MAP}} = b + \hat{B}d \tag{6.33}$$

where $\hat{B} \in \mathbb{R}^{n_x \times n_d}$ and $b \in \mathbb{R}^{n_x}$. Similarly, the reduced basis approach can be reduced to

$$x_{\mathrm{MAP,a}} = b + U_{x,r_x}\left(\hat{B}_a d\right) \tag{6.34}$$

where $U_{x,r_x} \in \mathbb{R}^{n_x \times r_x}$ and $\hat{B}_a \in \mathbb{R}^{r_x \times n_d}$. Computing x_{MAP} requires $2n_x n_d$ flops, while computing $x_{\mathrm{MAP,a}}$ requires $2r_x(n_d + n_x - 1)$ flops. Therefore, reducing the basis is a potential saving in computations and memory for such a case. Note that the reduced basis approach also presents potential computational cost reductions for other cases e.g. nonlinear A, although the cost reduction is more difficult to quantify [52, 99, 100].

6.3 The Bayesian Approximation Error

This section reviews the Bayesian Approximation Error (BAE) approach. The underlying idea of BAE is to simulate data in the offline phase with an accurate but computationally expensive model, and use these simulations to correct a less accurate but computationally efficient model that will be used in the online phase. A formal outline can be found in [36]. This method has been used in a variety of applications such as electrical impedance tomography [27], aquifer estimation from seismic data [22], fluorescence diffuse optical tomography [101], dental x-ray CT [33], full-wave ultrasound imaging [102], fluorescence molecular tomography [103] and quantitative photoacoustic tomography [104].

Let the "accurate, expensive" model be

$$d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e \tag{6.35}$$
$$= \bar{A}(\bar{\omega}) + e \tag{6.36}$$
$$= \bar{y} + e \tag{6.37}$$

where $\bar{A}$ is the forward model, $\bar{x}$ is the unknown of interest, $\bar{z}$ is an additional unknown, $\bar{\xi}$ is a model parameter, $\bar{y}$ is the forward model prediction and e is additive error. $\bar{\omega}$ represents $\bar{x}$, $\bar{z}$ and $\bar{\xi}$. Note that the above is an additive error model. BAE can be adapted to more general models, as described in [36].

Consider the deconvolution example from Chapter 2. In this problem, $\bar{z}$ is the signal outside the region of interest $[5, 15]$. The convolution kernel is defined by $\bar{\xi}$. Typically, we would need to also reconstruct $\bar{z}$ and $\bar{\xi}$ in order to find $\bar{x}$ from d.

Let the "approximate, cheap" model be

$$d \approx A(x, z, \xi) + e \tag{6.38}$$
$$= y + e \tag{6.39}$$

where A is the approximate forward model, x, z and ξ are approximations of $\bar{x}$, $\bar{z}$ and $\bar{\xi}$ and y is the approximate forward model prediction. Typically $x \in \mathbb{R}^{n_x}$ and $\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$ with

$n_x < n_{\bar{x}}$ and $x = P\bar{x}$ where P is a projection matrix. Often x will be the "sufficient resolution" approximation of the unknowns of interest. It is often also the case that z and ξ are fixed as e.g. $z = \mu_{\bar{z}}$ and $\xi = \mu_{\bar{\xi}}$. In this thesis we write $A(x, z, \xi) = A(x)$ if z and ξ are fixed.

The model can be expressed as

$$d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e \tag{6.40}$$
$$= A(x) + \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) - A(x) + e \tag{6.41}$$
$$= A(x) + \varepsilon + e \tag{6.42}$$

where $\varepsilon = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) - A(x) = \bar{y} - y$ is called the *approximation error*. Note that the above equations are exact. The Bayesian approximation error approach is to build a *statistical model* of ε during the offline phase, and use the statistical model to correct predictions from A during the online phase.

In this thesis, BAE involves treating $\pi(x)$ and $\pi(\varepsilon)$ as normally distributed dependant variables. Treating x and ε as mutually independent leads instead to the *enhanced error model* [105].

The model of ε is computed from sample statistics. First compute m samples ε_j of ε with $1 \leq j \leq m$ as

$$\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - A(x_j) \tag{6.43}$$

where $\bar{x}_j, \bar{z}_j, \bar{\xi}_j$ and x_j are also draws. Note that $x_j = P_{x,\bar{x}}\bar{x}_j$. Theses samples are used to construct sample matrices $\hat{X} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^{n_x \times m}$ and $\hat{\Upsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m) \in \mathbb{R}^{n_d \times m}$ where each column is a sample. The sample means are computed as

$$\hat{\mu}_x = \frac{1}{m}\sum_{j=1}^{m} x_j \tag{6.44}$$
$$\hat{\mu}_\varepsilon = \frac{1}{m}\sum_{j=1}^{m} \varepsilon_j \tag{6.45}$$

and sample covariances as

$$\hat{\Gamma}_x = \frac{1}{m-1}\hat{X}\hat{X}^T - \frac{m}{m-1}\hat{\mu}_x\hat{\mu}_x^T \tag{6.46}$$
$$\hat{\Gamma}_\varepsilon = \frac{1}{m-1}\hat{\Upsilon}\hat{\Upsilon}^T - \frac{m}{m-1}\hat{\mu}_\varepsilon\hat{\mu}_\varepsilon^T \tag{6.47}$$
$$\hat{\Gamma}_{\varepsilon x} = \frac{1}{m-1}\hat{\Upsilon}\hat{X}^T - \frac{m}{m-1}\hat{\mu}_\varepsilon\hat{\mu}_x^T \tag{6.48}$$

which we refer to as the *unbiased least squares* estimates of the *normal sample statistics* μ and Γ [106]. The conditional mean and covariance of ε given x are

$$\hat{\mu}_{\varepsilon|x} = \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) \tag{6.49}$$

$$\hat{\Gamma}_{\varepsilon|x} = \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{\varepsilon x}^T \tag{6.50}$$

where we note that the use of sample covariance estimates ensure $\hat{\Gamma}_{\varepsilon|x}$ will be positive (semi)definite.

BAE effectively adds an affine correction term to the model A. Consider the predictions of d given x. In the Bayesian framework, we could use $\mathbb{E}(d|x)$. Without BAE,

$$\mathbb{E}(d|x) \approx \mathbb{E}(A(x) + e|x) = A(x) + \mu_e \tag{6.51}$$

whereas with BAE we can incorporate knowledge of ε to form

$$\mathbb{E}(d|x) \approx \mathbb{E}(A(x) + \varepsilon + e|x) \tag{6.52}$$

$$= A(x) + \hat{\mu}_{\varepsilon|x} + \mu_e \tag{6.53}$$

$$= A(x) + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) + \hat{\mu}_\varepsilon + \mu_e \tag{6.54}$$

hence the interpretation as an affine correction.

Define the *total error* ν as

$$\nu = \varepsilon + e \tag{6.55}$$

with

$$\hat{\mu}_{\nu|x} = \mu_e + \hat{\mu}_{\varepsilon|x} = \mu_e + \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) \tag{6.56}$$

$$\hat{\Gamma}_{\nu|x} = \Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{\varepsilon x}^T \tag{6.57}$$

where we note that $\left\|\hat{\Gamma}_{\nu|x}\right\|_2 > \left\|\hat{\Gamma}_e\right\|_2$ as $\hat{\Gamma}_{\varepsilon|x}$ is positive semidefinite i.e the approximation error framework increases the error covariance to account for the differences between the accurate and the cheap models. The addition of approximation errors is particularly important when approximation errors *dominate* measurement errors i.e. $\left\|\hat{\Gamma}_{\varepsilon|x}\right\|_2 > \left\|\hat{\Gamma}_e\right\|_2$ [91].

The derivation of the posterior distribution with approximation errors is similar to the derivation in Section 4.1, resulting in

$$\pi_{x|d}(x|d) \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_{\nu|x}\big(d - A(x) - \mu_{\nu|x}\big)\right\|_2^2 + G_x(x)\right)\right) \tag{6.58}$$

where $\pi_x(x) \propto \exp\big(-\frac{1}{2}G_x(x)\big)$ and $\tilde{L}_{\nu|x}^T\tilde{L}_{\nu|x} = \hat{\Gamma}_{\nu|x}^{-1}$. Note that $\pi_x(x)$ is approximated as normal in the construction of $\hat{\mu}_{\varepsilon|x}$, but $\pi_x(x)$ appears exactly in the posterior. In the case that $\pi_x(x)$ is normal, the posterior is

$$\pi(x|d)_{x|d} \propto \exp\left(-\frac{1}{2}\left(\left\|\tilde{L}_{\nu|x}\big(d - A(x) - \mu_{\nu|x}\big)\right\|_2^2 + \left\|\tilde{L}_x(x-\mu_x)\right\|_2^2\right)\right) \tag{6.59}$$

where $\tilde{L}_x^T\tilde{L}_x = \Gamma_x^{-1}$.

Consider the model

$$d = Ax + e \tag{6.60}$$

where x and e are normally distributed independent variables. Consider a particular realisation $d_t = Ax_t + e_t$. The MAP estimate of x_t given d_t without BAE is

$$\begin{aligned} x_{\mathrm{MAP},\varepsilon=0} &= \min_x\{||\tilde{L}_e(d_t - Ax - \mu_e)||_2^2 + ||\tilde{L}_x(x-\mu_x)||_2^2\} && (6.61)\\ &= \min_x\left\{\left\|\begin{pmatrix}\tilde{L}_eA\\ \tilde{L}_x\end{pmatrix}x - \begin{pmatrix}\tilde{L}_e(d_t-\mu_e)\\ \tilde{L}_x\mu_x\end{pmatrix}\right\|_2^2\right\} && (6.62)\\ &= \begin{pmatrix}\tilde{L}_eA\\ \tilde{L}_x\end{pmatrix}^\dagger\begin{pmatrix}\tilde{L}_e(d_t-\mu_e)\\ \tilde{L}_x\mu_x\end{pmatrix} && (6.63)\\ &= \hat{A}_{\varepsilon=0}^\dagger\hat{d}_{t,\varepsilon=0} && (6.64)\end{aligned}$$

where the $\varepsilon = 0$ subscript is used to clarify that this estimate does not make use of BAE. Compare this with MAP estimate found with BAE

$$\begin{aligned} x_{\mathrm{MAP}} &= \min_x\left\{\left\|\tilde{L}_{\nu|x}\big(d_t - Ax - \mu_{\nu|x}\big)\right\|_2^2 + \left\|\tilde{L}_x(x-\mu_x)\right\|_2^2\right\} && (6.65)\\ &= \min_x\left\{\left\|\tilde{L}_{\nu|x}\big(d_t - Ax - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x-\hat{\mu_x}) - \mu_e - \hat{\mu}_\varepsilon\big)\right\|_2^2 + \left\|\tilde{L}_x(x-\mu_x)\right\|_2^2\right\} && (6.66)\\ &= \begin{pmatrix}\tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})\\ \tilde{L}_x\end{pmatrix}^\dagger\begin{pmatrix}\tilde{L}_{\nu|x}(d - \mu_e - \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\mu}_x)\\ \tilde{L}_x\mu_x\end{pmatrix}. && (6.67)\end{aligned}$$

Note that A has been replaced with $A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}$, a hybrid physical and statistical model, and that $d - \mu_e$ is replaced with $d - \mu_e - \hat{\mu}_\varepsilon$, and the substitution from weighing the data by $\tilde{L}_e$ to $\tilde{L}_{\nu|x}$ where $||\hat{\Gamma}_{\nu|x}||_2 > ||\hat{\Gamma}_e||_2$, in turn increasing the weight given to the prior when computing the MAP estimate with BAE.

Now consider the posterior covariance. The derivation can be found in [36] and is similar to the $d = Ax + e$ case of Section 4.1, yielding

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A^T \Gamma_{\nu|x} A)^{-1} \tag{6.68}$$

where A is replaced by Jacobian J if the operator is nonlinear and we make the Laplace approximation.

6.3.1 Inverse Crimes and Synthetic Demonstrations

This section reviews a potential error made when considering inverse problems, as described in [5, 91]. When performing simulations to check the validity of a method, an *inverse crime* may be committed. An example of an inverse crime is using the same model to simulate data *and* estimate the ground truth. The simulated estimates are then more accurate than for real data.

A common way to avoid committing an inverse crime is to use an additional model for simulating data for inversions. BAE would then require use of 3 models. The "accurate" model $\bar{A}$, the approximate model A, and the "simulation" model $\bar{\bar{A}}$. Typically this simply means using 3 discretisations, with $\bar{\bar{A}}$ using the finest, $\bar{A}$ the second finest, and A the coarsest. The models $\bar{A}$ and A are used to build a model of ε.

Let $d_s = \bar{\bar{A}}(\bar{\bar{x}}_s, \bar{\bar{z}}_s, \bar{\bar{\xi}}_s) + e_s$ be simulated data. The coarse model A and the statistical model of ε are then used to estimate $x_s = P_{x,\bar{\bar{x}}} \bar{\bar{x}}_s$. The data d_s is used when demonstrating the validity of the estimation of x_s.

How $\bar{\bar{x}}_s$ is chosen should also be considered. Taking $\bar{\bar{x}}_s$ as a draw of the prior is a low level inverse crime. The demonstration case $\bar{\bar{x}}_s$ should instead be drawn from a different (but reasonable) prior or, ideally, come from a lab experiment. For example in Section 5.3, $\bar{\bar{x}}_s$ was Barbara, not a draw of the TV prior.

Ideally, we would not even use $\bar{\bar{A}}$, instead using real world measurements d_r. Note however that this would also require real world knowledge of corresponding x_r to check that the reconstruction was correct. For the sake of the BAE framework, it would be even better still to not need to use $\bar{A}$, instead using a library of lab results to compute the difference between $A(x)$ and reality. This would require a vast amount of noiseless lab measurements $\bar{y}_r$ and corresponding x_r, which is not generally forthcoming.

6.3.2 Discretisation Invariance

Another important consideration is *discretisation invariance* [5]. This was mentioned briefly in Section 5.3. This section reviews prior discretisation invariance in greater detail, and is similar to [91].

Consider a smoothness prior for $\bar{s} = \begin{pmatrix} \bar{x} \\ \bar{z} \end{pmatrix}$ with covariance $\Gamma_{\bar{s}}$. Let $L_{\bar{s}}$ be such that $L_{\bar{s}} L_{\bar{s}}^T =$

$\Gamma_{\bar{s}}$. Draws are computed as

$$\bar{s}_j = L_{\bar{s}} w_j \tag{6.69}$$

where $w_j \in \mathbb{R}^{n_{\bar{s}}}$ is a draw of $w \in \mathcal{N}(0_{n_{\bar{s}},1}, I_{n_{\bar{s}}})$. Draws of $\bar{x}$ are

$$\bar{x}_j = P_{\bar{x},\bar{s}} \bar{s}_j = P_{\bar{x},\bar{s}} L_{\bar{s}} w_j \tag{6.70}$$

where $P_{\bar{x},\bar{s}}$ is a matrix that cuts out the $\bar{z}$ component. The corresponding covariance matrix for $\bar{x}$ is therefore

$$\begin{aligned} \Gamma_{\bar{x}} &= P_{\bar{x},\bar{s}} L_{\bar{s}} (P_{\bar{x},\bar{s}} L_{\bar{s}})^T & (6.71) \\ &= P_{\bar{x},\bar{s}} \Gamma_{\bar{s}} P_{\bar{x},\bar{s}}^T. & (6.72) \end{aligned}$$

Now consider the prior on x. Samples x_j of x are computed as

$$x_j = P_{x,\bar{x}} \bar{x}_j = P_{x,\bar{x}} P_{\bar{x},\bar{s}} L_{\bar{s}} w_j = P_{x,\bar{s}} L_{\bar{s}} w_j \tag{6.73}$$

where $P_{x,\bar{x}}$ is the projection matrix from $\bar{x}$ to x. The covariance of x is therefore

$$\begin{aligned} \Gamma_x &= P_{x,\bar{s}} L_{\bar{s}} (P_{x,\bar{s}} L_{\bar{s}})^T & (6.74) \\ &= P_{x,\bar{s}} \Gamma_{\bar{s}} P_{x,\bar{s}}^T. & (6.75) \end{aligned}$$

Note that the above construction considers samples first, and then constructs the implied covariances. The alternative approach would be to compute all of the covariance matrices "directly" e.g. constructing covariances from the corresponding covariance function. In the case of the smoothness prior, this means computing $\Gamma_{\bar{z}}$, $\Gamma_{\bar{s}}$ and Γ_x with the same h and l. The problem with this approach is that the priors will not be *consistent*. Let $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ where $\Gamma_x = P_{x,\bar{s}} \Gamma_{\bar{s}} P_{x,\bar{s}}^T$ and $\tilde{x} \sim \mathcal{N}(\mu_x, \Gamma_{\tilde{x}})$ where $\Gamma_{\tilde{x}}$ is constructed from the covariance function. The correlation structures are fundamentally different i.e. $\Gamma_{\tilde{x}} \neq \Gamma_x$. Furthermore, the draws $\tilde{x}_j = L_{\tilde{x}} P C w_j \neq P \bar{x}_j$ which would make the approximation errors $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - A(\tilde{x}_j)$ meaningless.

6.3.3 BAE Deconvolution Example

Let us apply BAE to the 1D deconvolution example of Section 2.3.2. Let $\mathbf{s}(\mathbf{t})$ be a signal on $\Omega = [0, 20]$. Let $\mathbf{s}(\mathbf{t}) = \mathbf{x}(\mathbf{t})$ for $\mathbf{t} \in \Omega_x = [5, 15]$ and $\mathbf{s}(\mathbf{t}) = \mathbf{z}(\mathbf{t})$ for $\mathbf{t} \in \Omega_z = [0, 5) \cup (15, 20]$.

We will discretise our domain in 2 ways. Let

$$\Delta_{\bar{t}} = 0.1 \tag{6.76}$$
$$\bar{t} = (0, \Delta_{\bar{t}}, 2\Delta_{\bar{t}}, \ldots, 20) \in \mathbb{R}^{201} \tag{6.77}$$
$$\bar{s}(j) = \mathbf{s}(\bar{t}(j)) \tag{6.78}$$
$$\bar{t}_x = (5, 5 + \Delta_{\bar{t}}, 5 + 2\Delta_{\bar{t}}, \ldots, 15) \in \mathbb{R}^{101} \tag{6.79}$$
$$\bar{x}(j) = \mathbf{x}(\bar{t}_x(j)) = \mathbf{s}(\bar{t}_x(j)) \tag{6.80}$$
$$\Delta_t = 0.2 \tag{6.81}$$
$$t = (5, 5 + \Delta_t, 5 + 2\Delta_t, \ldots, 15) \in \mathbb{R}^{51} \tag{6.82}$$
$$x(j) = \mathbf{x}(t(j)) = \mathbf{s}(t(j)) \tag{6.83}$$

i.e. $\bar{s} \in \mathbb{R}^{201}$, $\bar{x} \in \mathbb{R}^{101}$, $\bar{z} \in \mathbb{R}^{100}$, $x \in \mathbb{R}^{51}$. In the approximate model, implicitly $z = \mu_{\bar{z}} = 0_{n_{\bar{z}},1}$ i.e. A has no padding. Let $x = P_{x,\bar{x}}\bar{x}$ where

$$P_{x,\bar{x}} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 1 & 1 \end{pmatrix} \in \mathbb{R}^{51\times 101} \tag{6.84}$$

and similarly $P_{x,s} = \begin{pmatrix} 0_{51,50} & P_{x,\bar{x}} & 0_{51,50} \end{pmatrix} \in \mathbb{R}^{51\times 201}$.

Let us consider discretisation variance. Let $\Gamma_{\bar{s}}$ be constructed from the exponential squared smoothness covariance function of Section 5.1.1 with height h_t and correlation length l_t. This covariance is used to construct $L_{\bar{s}}$, allowing draws to be computed as

$$\bar{x}_j = P_{\bar{x},\bar{s}} L_{\bar{s}} w_j \tag{6.85}$$

where w_j is a white noise draw and $P_{\bar{x},\bar{s}}$ removes the $\bar{z}$ part of $\bar{s}$. The discretisation invariant prior covariance on x is therefore

$$\Gamma_x = P_{x,\bar{s}} \Gamma_{\bar{s}} P_{x,\bar{s}}^T \tag{6.86}$$

and the filter is $L_x = P_{x,\bar{s}} L_{\bar{s}}$. Draws of x are computed as

$$x_j = L_x w_j. \tag{6.87}$$

Let $\Gamma_{\tilde{x}}$ be constructed using the same covariance function, h_t, and l_t. This is then used to construct the filter $L_{\tilde{x}}$. Draws are computed as

$$\tilde{x}_j = L_{\tilde{x}} \tilde{w}_j = L_{\tilde{x}} P_{x,\bar{s}} w_j \tag{6.88}$$

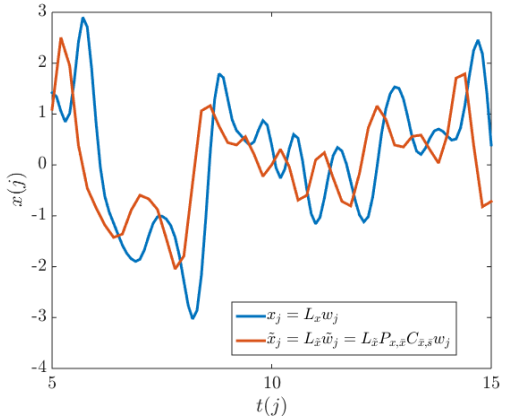


Figure 6.4: Draws, same white noise w_j

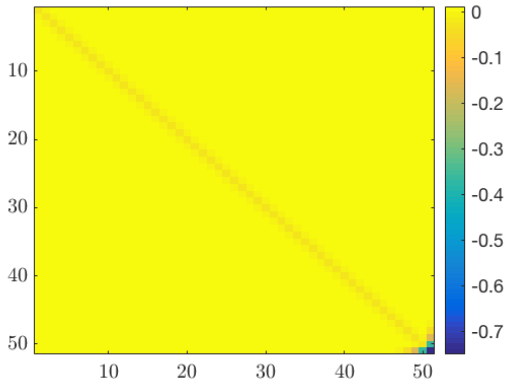


Figure 6.5: $\Gamma_x - \Gamma_{\tilde{x}}$

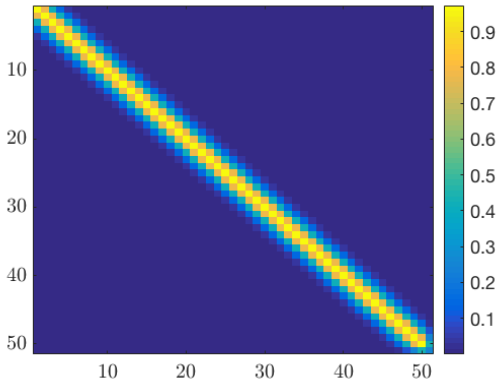


Figure 6.6: $\Gamma_x = PC\Gamma_{\tilde{s}}C^T P^T$

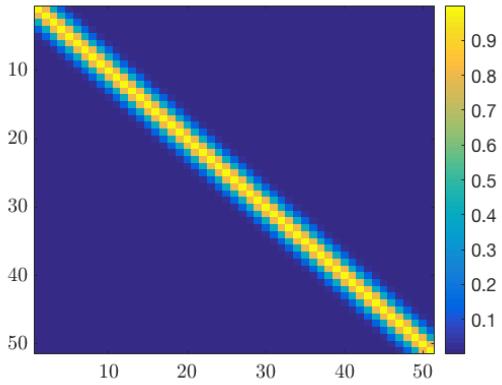


Figure 6.7: $\Gamma_{\tilde{x}}$ from covariance function

noting that w_j needs to be cut and downsampled. Example draws with the same w_j are shown in Figure 6.4. The discretisation invariant prior covariance Γ_x is visualised in Figure 6.6, and $\Gamma_{\tilde{x}}$ is visualised in Figure 6.7. The difference of the two matrices $\Gamma_x - \Gamma_{\tilde{x}}$ is visualised in Figure 6.5.

Let

$$\mathbf{k}(\mathbf{t}, \xi) = \exp\left(-\frac{\mathbf{t}^2}{\xi^2} \right) \tag{6.89}$$

be the convolution kernel with $\xi \sim \mathbb{U}(0.3, 0.7)$ the *kernel width*, recalling that $\mathbf{t}$ represents the spatial variable on a continuous domain. Let $\bar{k}(j) = \mathbf{k}(\bar{t}(j), \xi)$ be used in the construction of $\bar{A}$ and $k(j) = \mathbf{k}(t(j), \mu_\xi)$ be used in the construction of A. Kernels of different widths are

shown in Figure 6.8. Let $\bar{F}_\xi \in \mathbb{R}^{201\times 201}$ be the convolution matrix using the kernel of width ξ. Let $\bar{F}_\xi \bar{s} = c_\xi \in \mathbb{R}^{201}$ be the convolved signal. A signal $\bar{s}$ is plotted alongside 3 convolved signals c_ξ computed with the 3 kernels of Figure 6.8.

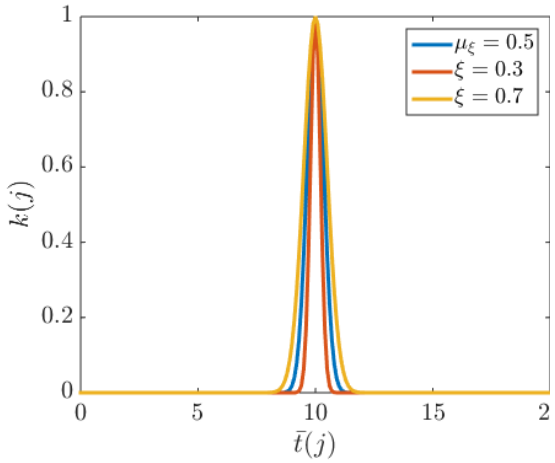


Figure 6.8: Different kernels $k(t)$

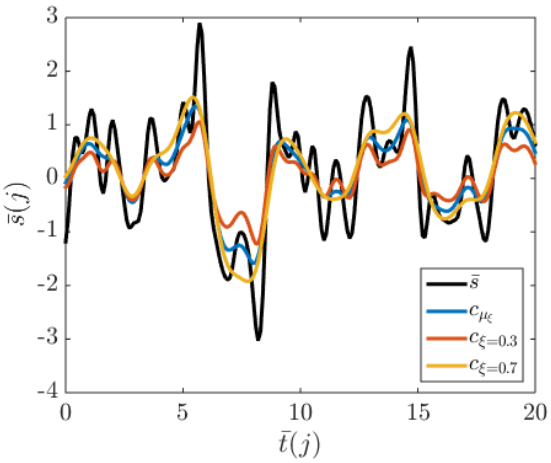


Figure 6.9: Signal, and convolution with different kernel widths

The model $\bar{A} \in \mathbb{R}^{51\times 201}$ is constructed as

$$\bar{A} = M_{d,\bar{x}} P_{\bar{x},\bar{s}} \bar{F}_\xi \tag{6.90}$$

where $C_{\bar{x},\bar{s}} \in \mathbb{R}^{201\times 101}$ extracts the part of the signal s in Ω_x i.e. $\bar{x} = P_{\bar{x},\bar{s}}\bar{s}$ and $M_{d,\bar{x}} \in \mathbb{R}^{51\times 101}$ maps to the measurement points. The model $A \in \mathbb{R}^{51\times 101}$ is constructed as

$$A = 2M_{d,\bar{x}} P_{\bar{x},\bar{s}} \bar{F}_{\mu_\xi} P_{\bar{x},\bar{s}}^T P_{x,\bar{x}}^T \tag{6.91}$$

where we note that $2P_{\bar{x},\bar{s}}^T P_{x,\bar{x}}^T$ effectively upsamples x to $\bar{x}$ and pads with $\bar{z} = 0_{n_{\bar{z}},1}$ and $\bar{F}_{\mu_\xi}$ is reused. Constructing A in this way can simplify the BAE process when $\bar{F}$ comes from black box software or is otherwise difficult to adapt to the discretisation we wish to use.

The model predictions $\bar{y}_j = \bar{A}(\bar{x}_j, \bar{z}_j, \xi_j)$ and $y_j = Ax_j$ are shown in Figure 6.10. Note that $\bar{A}$ uses $\xi_j = 0.4$ and uses information about $\bar{z}$, while the "approximate" model fixes $\xi = 0.5$ and $z = 0$.

We now have everything required to begin the offline simulations of ε. We draw $m = 1,000$ samples $\bar{s}_j$ and ξ_j from which we compute x_j and $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \xi_j) - Ax_j$. We construct $\hat{X} = (x_1, x_2, \ldots, x_m)$ and $\hat{\Upsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m)$. These data matrices are used to construct the sample means and covariance. The marginal approximation error covariance $\hat{\Gamma}_\varepsilon$ is visualised in Figure 6.11 and the conditional covariance $\hat{\Gamma}_{\varepsilon|x}$ is visualised in Figure 6.12. Note the relative

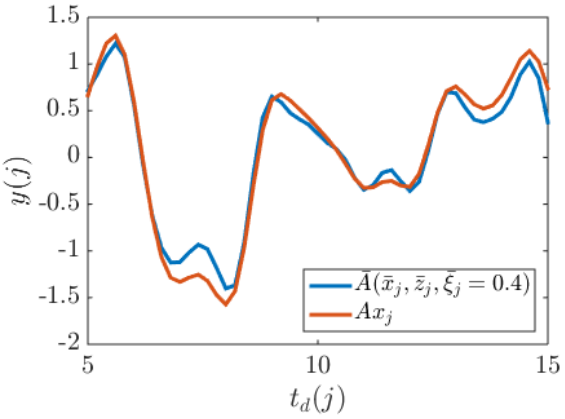


Figure 6.10: Forward mode predictions of $\bar{A}$ and A

"size" of each from the colorbar scales. Observe that the approximation error variance is largest near $t = 5$ and $t = 15$, a consequence of not padding the domain with signal outside of $[5, 15]$.

The eigenvalues of Γ_ε are plotted in Figure 6.13. Observe that the approximation error covariance is effectively low rank. Recall that BAE effectively replaces A with $A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}$. The matrix $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}$ is visualised in Figure 6.14. Note that most of the "correction" seems to be at entries corresponding to the endpoints $t = 5$ and $t = 15$.

Let $d_t = \bar{\bar{A}}(\bar{\bar{x}}_t, \bar{\bar{z}}_t, \xi_t) + e_t$ be data simulated with the extra fine forward model $\bar{\bar{A}}$. This model is computed similarly to $\bar{A}$, but with $n_{\bar{\bar{x}}} > n_{\bar{x}}$ and $n_{\bar{\bar{s}}} > n_{\bar{s}}$ i.e. at an even finer discretisation. Let $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$ where $\sigma_e = 0.001$. In this case, $\|\Gamma_\varepsilon\|_2 \gg \|\Gamma_e\|_2$. The MAP found without BAE is

$$x_{\text{MAP},\varepsilon=0} = \min_x \left\{ \left\| \tilde{L}_e(d_t - Ax) \right\|_2^2 + \left\| \tilde{L}_x(x) \right\|_2^2 \right\} \tag{6.92}$$

$$= \begin{pmatrix} \tilde{L}_e A \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_e d_t \\ 0 \end{pmatrix} \tag{6.93}$$

with posterior covariance

$$\Gamma_{x|d,\varepsilon=0} = (\Gamma_x^{-1} + A^T \Gamma_e A)^{-1} \tag{6.94}$$

where we note $\mu_e = 0_{n_d,1}$, $\mu_x = 0_{n_x,1}$ in our model. The MAP found with BAE is

$$x_{\text{MAP},\varepsilon} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d_t - Ax - \mu_{\epsilon|x}) \right\|_2^2 + \left\| \tilde{L}_x(x) \right\|_2^2 \right\} \tag{6.95}$$

$$= \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d_t - \hat{\mu}_\varepsilon) \\ 0_{n_x,1} \end{pmatrix} \tag{6.96}$$

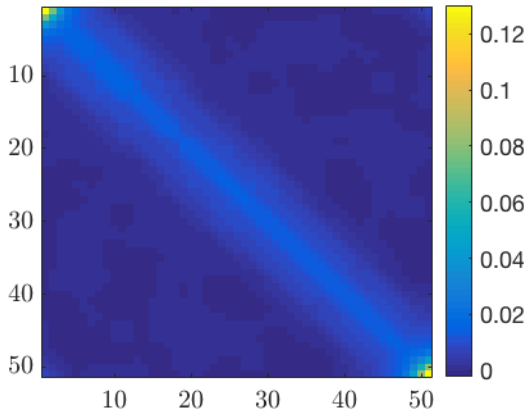


Figure 6.11: $\hat{\Gamma}_{\varepsilon}$

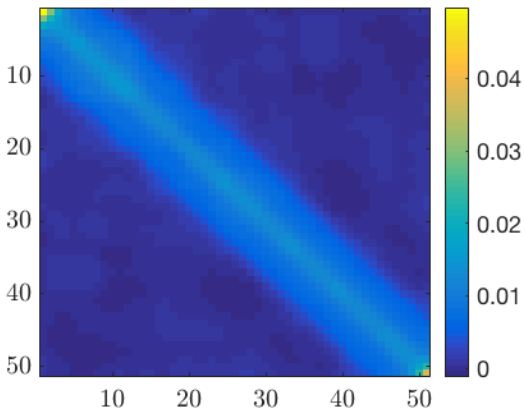


Figure 6.12: $\hat{\Gamma}_{\varepsilon|x}$

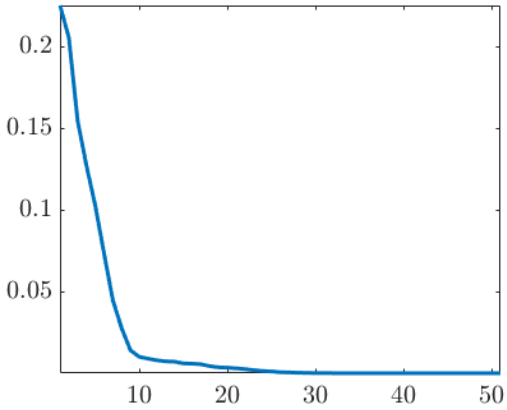


Figure 6.13: Eigenvalues of $\hat{\Gamma}_{\varepsilon}$

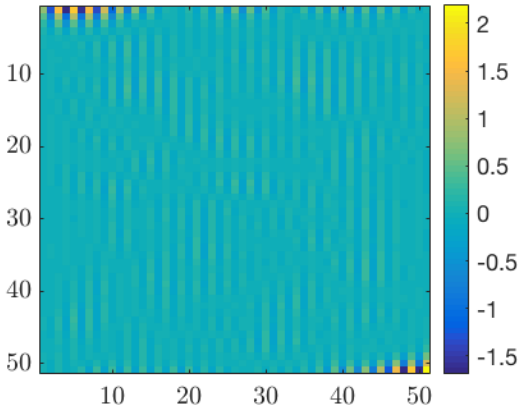


Figure 6.14: $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_{x}^{-1}$

with posterior covariance

$$\Gamma_{x|d,\varepsilon} = (\Gamma_x^{-1} + A^T\Gamma_{\nu|x}A)^{-1}. \tag{6.97}$$

We will take $3\hat{\sigma}$ posterior error intervals from the posterior covariance diagonal.

The MAP estimate and posterior error intervals found without BAE are shown in Figure 6.15. Note that $x_{\mathrm{MAP},\varepsilon=0}$ seems to "overestimate" the solution, similar to the Tikhonov estimates with too small a penalty term as in Sections 3.4.2 and 4.2.6. Note also that the posterior error intervals are far too narrow. The "overestimating" MAP and undersized posterior error intervals are a result of underestimating the error, using only e instead of $\nu = e + \varepsilon$.

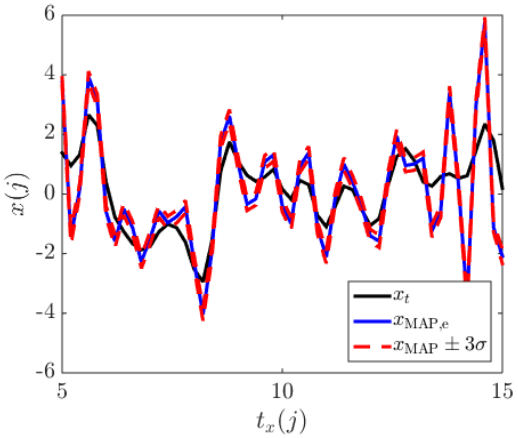


Figure 6.15: MAP and posterior error intervals from standard error model alongside ground truth

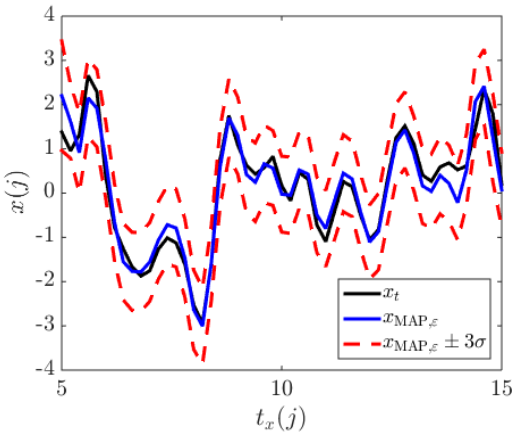


Figure 6.16: MAP and posterior error intervals from BAE model alongside ground truth

The MAP estimate and posterior error intervals found with BAE are shown in Figure 6.16. The MAP and posterior error intervals appear representative. Note that the MAP is furthest from the ground truth, and the posterior error intervals are widest, around $t = 5$ and $t = 15$.

6.4 Computational Simplifications for BAE

The Bayesian Approximation Error framework involves a large amount of computations in the offline stage. This section discusses the computational cost of BAE and how it can be reduced.

Recall that the MAP solution found with BAE is

$$x_{\text{MAP},\varepsilon} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d_t - Ax - \hat{\mu}_{\varepsilon|x}) \right\|_2^2 + \left\| \tilde{L}_x(x) \right\|_2^2 \right\} \tag{6.98}$$

$$= \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_{\nu|x}(d_t - \hat{\mu}_\varepsilon) \\ 0_{n_x,1} \end{pmatrix} \tag{6.99}$$

where

$$\tilde{L}_{\nu|x}^T \tilde{L}_{\nu|x} = \Gamma_{\nu|x}^{-1} \tag{6.100}$$

$$= (\Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon})^{-1} \tag{6.101}$$

which requires we compute

$$\hat{\mu}_{\hat{x}} = \frac{1}{m}\sum_{j=1}^{m} x_j \qquad \hat{\Gamma}_{\hat{x}} = \frac{1}{m-1}\hat{X}\hat{X}^T - \frac{m}{m-1}\hat{\mu}_{\hat{x}}\hat{\mu}_{\hat{x}}^T \tag{6.102}$$

$$\hat{\mu}_{\hat{\varepsilon}} = \frac{1}{m}\sum_{j=1}^{m} \varepsilon_j \qquad \hat{\Gamma}_{\hat{\varepsilon}} = \frac{1}{m-1}\hat{\Upsilon}\hat{\Upsilon}^T - \frac{m}{m-1}\hat{\mu}_{\hat{\varepsilon}}\hat{\mu}_{\hat{\varepsilon}}^T \tag{6.103}$$

$$\hat{\Gamma}_{\varepsilon x} = \frac{1}{m-1}\hat{\Upsilon}\hat{X}^T - \frac{m}{m-1}\hat{\mu}_{\hat{\varepsilon}}\hat{\mu}_{\hat{x}}^T \tag{6.104}$$

from samples. If we were to approach this "directly", we would compute the matrix multiplications for the sample statistics, then invert $\hat{\Gamma}_{\hat{x}}$, then compute the other matrix multiplies and additions to get $\Gamma_{\nu|x}$, then invert this matrix, then compute the factor $\tilde{L}_{\nu|x}$.

6.4.1 QR factorisation for BAE

Each sample ε_j requires an evaluation of $\bar{A}$ and A. This means that samples of ε "arrive" relatively slowly, in the sense that a large number of computations are performed per sample. The method proposed in this section was developed to compute the statistical quantities of interest e.g. $\hat{\Gamma}_{\varepsilon|x}$ in parallel with samples being generated.

It is worth noting that the methodology below was designed for BAE, but could be applied to computing conditional statistics generally. In particular, this technique is useful when high dimensional samples are arriving relatively slowly, allowing covariances and conditional covariances to be computed very quickly as soon as sampling is completed, or even while the sampling is being performed. It is particularly useful when the sample statistical quantities of interest are of low effective rank.

The method of this section makes use of the QR decomposition. More information on the QR factorisation can be found in [32]. Note that typically slight variations such as pivoted QR are used - we use the standard form here for clarity. Consider the sample matrix

$$\hat{X} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^{n_x \times m} \tag{6.105}$$

with QR factorisation

$$\hat{X} = \hat{Q}_{\hat{x}}\hat{R}_{\hat{x}} \tag{6.106}$$

where $\hat{Q}_{\hat{x}} \in \mathbb{R}^{n_x \times n_x}$ with $\hat{Q}_{\hat{x}}\hat{Q}_{\hat{x}}^T = \hat{Q}_{\hat{x}}^T\hat{Q}_{\hat{x}} = I_{n_x}$ and $\hat{R}_{\hat{x}} \in \mathbb{R}^{n_x \times m}$ is upper triangular. In the case that $\hat{X}$ has rank $r_x < n_x$, only the top r_x rows of R_x have nonzero elements. The "thin" QR factorisation is

$$\hat{X} = \hat{Q}_{\hat{x}}\hat{R}_{\hat{x}} = \begin{pmatrix} Q_{\hat{x}} & Q_{\hat{x},\text{null}} \end{pmatrix} \begin{pmatrix} R_{\hat{x}} \\ 0_{n_x - r_x, m} \end{pmatrix} \tag{6.107}$$

$$= Q_{\hat{x}}R_{\hat{x}} \tag{6.108}$$

where $Q_{\hat{x}} \in \mathbb{R}^{n_x \times r_x}$ has orthonormal columns and $R_{\hat{x}} \in \mathbb{R}^{r_x \times m}$ is upper triangular. Note that the information of $\hat{X} \in \mathbb{R}^{n_x \times m}$ is encoded in $Q_{\hat{x}} \in \mathbb{R}^{n_x \times r_x}$ and $R_{\hat{x}} \in \mathbb{R}^{r_x \times m}$, potentially saving memory.

A useful aspect of the QR decomposition is that it can be computed as columns of the matrix to be factorised arrive. Consider again $\hat{X}$. When x_1 is computed, we can compute

$$R_{\hat{x}}(1,1) = \|x_1\|_2 \tag{6.109}$$

$$Q_{\hat{x}}(:,1) = q_1 = \frac{x_1}{R_{\hat{x}}(1,1)} \tag{6.110}$$

the first entry of $R_{\hat{x}}$ and the first first column of $Q_{\hat{x}}$. When x_2 arrives, we can compute

$$R_{\hat{x}}(1,2) = q_1^T x_2 \tag{6.111}$$

$$R_{\hat{x}}(2,2) = \|x_2 - R_{\hat{x}}(1,2)q_1\|_2 \tag{6.112}$$

$$Q_{\hat{x}}(:,2) = q_2 = \frac{x_2 - R_{\hat{x}}(1,2)q_1}{\|x_2 - R_{\hat{x}}(1,2)q_1\|_2} \tag{6.113}$$

and so on for x_3, x_4 up to x_m. That is, perform Gram-Schmidt orthogonalisation on the samples as they arrive. I use Gram-Schmidt here as it is simple and familiar, however other orthogonalisation methods such as those in [32] can be used. This is not necessarily the most efficient way to factorise sample matrices, however the advantage is that the factorisation can be computed in parallel with samples being computed.

I include a tolerance τ in the QR factorisation. This both reduces the numerical loss of orthogonality in Gram-Schmidt orthogonalisation and allows for reducing the rank of the approximations through increasing the size of the tolerance. Suppose we already have $Q_{\hat{x},r_j} \in \mathbb{R}^{n_x \times r_j}$ and $R_{\hat{x},r_j} \in \mathbb{R}^{r_j \times j}$, the QR factorisation of the data matrix $\hat{X}_j \in \mathbb{R}^{n_x \times j}$ corresponding to the first j samples. Note that $r_j \leq j$ is the rank of $\hat{X}_j$. Now suppose sample x_{j+1} arrives. We compute

$$R_{\hat{x},r_j}(1:r_j, j+1) = Q_{\hat{x},r_j}^T x_{j+1} \tag{6.114}$$

$$\zeta = \left\| x_j - Q_{\hat{x},r_j} R_{\hat{x},r_j}(1:r_j, j) \right\|_2 \tag{6.115}$$

where ζ is the *residual*. If $\zeta > \tau$, form the rank updated factorisation $\hat{X}_{j+1} = Q_{\hat{x},r_{j+1}} R_{\hat{x},r_{j+1}}$ where

$$Q_{\hat{x},r_{j+1}} = \begin{pmatrix} Q_{\hat{x},r_j} & q_{r_{j+1}} \end{pmatrix} \in \mathbb{R}^{n_x \times r_{j+1}} \tag{6.116}$$

$$R_{\hat{x},r_{j+1}} = \begin{pmatrix} R_{\hat{x},r_j}(1:r_j, 1:j) & R_{\hat{x},r_j}(1:r_j, j+1) \\ 0_{1,j} & \zeta \end{pmatrix} \in \mathbb{R}^{r_{j+1} \times (j+1)} \tag{6.117}$$

where $r_{j+1} = r_j + 1$ and

$$q_{r_{j+1}} = \frac{x_{j+1} - Q_{\hat{x},r_j} R_{\hat{x},r_j}(1:r_j, j+1)}{\zeta} \in \mathbb{R}^{n_x}. \tag{6.118}$$

Note that when $\tau > 0$ is used while constructing the QR decomposition, $\hat{X} \approx Q_{\hat{x}} R_{\hat{x}}$. We state $\hat{X} = Q_x R_x$ for notational simplicity.

Computing a QR factorisation with a tolerance is not unusual. However, in this thesis a relatively large tolerance e.g. $\tau = 0.02 R_{\hat{x}}(1,1)$ is typically used. This is done to reduce r_x, saving memory and reducing computations.

Rather than x, we consider the mean removed variable $\chi = x - \mu_x$. This is stored in the *mean removed data matrix*

$$X = (\chi_1, \chi_2, \ldots, \chi_m) = (x_1 - \mu_x, x_2 - \mu_x, \ldots, x_m - \mu_x) \in \mathbb{R}^{n_x \times m} \tag{6.119}$$

which is used to construct the covariance estimate

$$\hat{\Gamma}_x = \frac{1}{m} X X^T \approx \hat{\Gamma}_{\hat{x}} = \frac{1}{m-1} \hat{X} \hat{X}^T - \frac{m}{m-1} \hat{\mu}_x \hat{\mu}_x^T \tag{6.120}$$

i.e. we approximate the sample mean. Using the mean removed variable simplifies the algebra, however $\hat{\Gamma}_x$ is a *biased estimate* of Γ_x. Also, the QR factorisation $\hat{X} = Q_{\hat{x}} R_{\hat{x}}$ tends to have the top few rows of $R_{\hat{x}}$ much larger than lower rows. This can lead to roundoff errors, especially when computing e.g. $R_{\hat{x}} R_{\hat{x}}^T$. In the case that the mean is not known exactly, an estimate based on a smaller number of samples can be used instead.

Alternatively, construct

$$\hat{\mu}_{r_x} = \frac{1}{m} \sum_{j=1}^{m} R_x(:, j) \tag{6.121}$$

and form

$$\Gamma_x \approx \frac{1}{m-1} Q_x \left(R_x R_x^T - m \hat{\mu}_{r_x} \hat{\mu}_{r_x}^T \right) Q_x^T \tag{6.122}$$

i.e. a rank 1 downdate of $R_x R_x^T$. The above is an unbiased positive semidefinite estimate of the covariance. Note that the above still uses the mean removed variable to avoid numerical errors. For the problems considered in this thesis, $\hat{\mu}_{r_x}$ was found to be negligible, so the downdate was not applied.

The sample covariance of (mean removed) x is

$$\Gamma_x \approx \hat{\Gamma}_x = \frac{1}{m} X X^T = \frac{1}{m} Q_x R_x R_x^T Q_x^T \tag{6.123}$$

in terms of the QR decomposition of the (mean removed) data matrix. Note that the above QR form of the covariance can be used as the first step in computing the SVD of $\hat{\Gamma}_x$ [32]. Similarly for $\hat{\Gamma}_\varepsilon = \frac{1}{m} Q_{\varepsilon,k_\varepsilon} (R_{\varepsilon,k_\varepsilon} R_{\varepsilon,k_\varepsilon}^T) Q_{\varepsilon,k_\varepsilon}^T$ and $\hat{\Gamma}_{\varepsilon,x} = \frac{1}{m} Q_{\varepsilon,k_\varepsilon} (R_{\varepsilon,k_\varepsilon} R_{x,k_x}^T) Q_{x,k_x}^T$. We can compute these QR based factorisations as samples arrive, but once sampling is complete, these results can also be used to help compute the more efficient SVDs if desired.

The QR decomposition of the sample covariance matrix can be used to construct a reduced basis as in Section 6.2. Let $Q_x^T x = a_x \in \mathbb{R}^{r_x}$ be the variable of interest in the basis spanned by the vectors $q_{x,j} \in \mathbb{R}^{n_x}$, the columns of Q_x. The covariance of a_x is

$$\Gamma_{a_x} = Q_x^T \Gamma_x Q_x \tag{6.124}$$
$$\approx Q_x^T \hat{\Gamma}_x Q_x \tag{6.125}$$
$$= Q_x^T (\frac{1}{m} Q_x R_x R_x^T Q_x^T) Q_x \tag{6.126}$$
$$= \frac{1}{m} R_x R_x^T = \hat{\Gamma}_{a_x} \tag{6.127}$$

noting that $\hat{\Gamma}_{a_x} \in \mathbb{R}^{r_x \times r_x}$ is full rank.

Now consider ε. Let

$$\Upsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m) = Q_\varepsilon R_\varepsilon \tag{6.128}$$

be the thin QR decomposition of the mean removed sample matrix. Note $Q_\varepsilon \in \mathbb{R}^{n_d \times r_\epsilon}$ where r_ε is the (effective) rank of Υ. The sample covariance estimate of ε is

$$\Gamma_\varepsilon \approx \hat{\Gamma}_\varepsilon = \frac{1}{m} \Upsilon \Upsilon^T = \frac{1}{m} Q_{\varepsilon,k_\varepsilon} R_{\varepsilon,k_\varepsilon} R_{\varepsilon,k_\varepsilon}^T Q_{\varepsilon,k_\varepsilon}^T \tag{6.129}$$

where we can instead consider $Q_\varepsilon^T \varepsilon = a_\varepsilon \in \mathbb{R}^{r_\varepsilon}$ similarly to $Q_x^T x = a_x$.

The sample estimate to the conditional covariance of ε given x is

$$\Gamma_{\varepsilon|x} \approx \hat{\Gamma}_{\varepsilon|x} = \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} \hat{\Gamma}_{x\varepsilon} \tag{6.130}$$

however $\hat{\Gamma}_x^{-1}$ only exists for $r_x = n_x$. Instead consider the conditional covariance of the reduced basis variables

$$\Gamma_{a_\varepsilon|a_x} \approx \hat{\Gamma}_{a_\varepsilon|a_x} = \hat{\Gamma}_{a_\varepsilon} - \hat{\Gamma}_{a_\varepsilon a_x} \hat{\Gamma}_{a_x}^{-1} \hat{\Gamma}_{a_x a_\varepsilon} \tag{6.131}$$
$$= \frac{1}{m} \left(R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x^T (R_x R_x^T)^{-1} R_x R_\varepsilon^T \right) \tag{6.132}$$

where we note that only the products $R_x R_x^T$, $R_\varepsilon R_\varepsilon^T$ and $R_x R_\varepsilon^T$ need to be stored.

Recall that the posterior with BAE is of the form

$$\pi(x|d) \propto \exp\left(\left\| \tilde{L}_{\nu|x} (d - A(x) - \hat{\mu}_{\nu|x}) \right\|_2^2 + G_x(x) \right) \tag{6.133}$$

where $G_x(x)$ comes from the prior, and $\tilde{L}_{\nu|x}^T \tilde{L}_{\nu|x} = \hat{\Gamma}_{\nu|x}^{-1} = (\Gamma_e + \hat{\Gamma}_{\varepsilon|x})^{-1}$. We also form $\hat{\mu}_{\nu|x} = \mu_e + \mu_\varepsilon + \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} (x - \mu_x)$.

Consider $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}$. Construct $\hat{\Gamma}_x$ as

$$m\hat{\Gamma}_x = \begin{pmatrix} Q_x & Q_{x_n} \end{pmatrix} \begin{pmatrix} R_x R_x^T & 0_{r_x, n_x - r_x} \\ 0_{n_x - r_x, r_x} & \kappa_x^2 I_{n_x - r_x} \end{pmatrix} \begin{pmatrix} Q_x & Q_{x_n} \end{pmatrix} \tag{6.134}$$

where r_x is the (effective) rank of sample matrix X, $Q_x \in \mathbb{R}^{n_x \times k_x}$ and $R_x \in \mathbb{R}^{k_x \times m}$ are found by recursive QR. $Q_{x_n} \in \mathbb{R}^{n_x \times (n_x - r_x)}$ has columns spanning the rest of $\mathbb{R}^{n_x}$ i.e. the orthogonal complement of Q_x. $\kappa_x \in \mathbb{R}$ is some small positive number. The above construction of $\hat{\Gamma}_x$ is analogous to ensuring regularity with a penalty subspace as described in Section 6.2.

The above form of $\hat{\Gamma}_x$ implies

$$\begin{aligned}
\frac{1}{m}\hat{\Gamma}_x^{-1} &= \begin{pmatrix} Q_x & Q_{x_n} \end{pmatrix} \begin{pmatrix} R_x R_x^T & 0_{r_x, n_x - r_x} \\ 0_{r_x, n_x - r_x} & \kappa_x^2 I_{n_x - r_x} \end{pmatrix}^{-1} \begin{pmatrix} Q_x^T \\ Q_{x_n}^T \end{pmatrix} && (6.135) \\
&= \begin{pmatrix} Q_x & Q_{x_n} \end{pmatrix} \begin{pmatrix} (R_x R_x^T)^{-1} & 0_{r_x, n_x - r_x} \\ 0_{n_x - r_x, r_x} & \frac{1}{\kappa_x^2} I_{n_x - r_x} \end{pmatrix} \begin{pmatrix} Q_x^T \\ Q_{x_n}^T \end{pmatrix} && (6.136) \\
&= Q_x (R_x R_x^T)^{-1} Q_x^T + \frac{1}{\kappa_x^2} Q_{x_n} Q_{x_n}^T && (6.137) \\
&= Q_x (R_x R_x^T)^{-1} Q_x^T + \frac{1}{\kappa_x^2} (I_{n_x} - Q_x Q_x^T) && (6.138) \\
&= Q_x \left((R_x R_x^T)^{-1} - \frac{1}{\kappa_x^2} I_{r_x} \right) Q_x^T + \frac{1}{\kappa_x^2} I_{n_x} && (6.139)
\end{aligned}$$

where we have made use of the fact that $\begin{pmatrix} Q_x & Q_{x_n} \end{pmatrix}^{-1} = \begin{pmatrix} Q_x^T \\ Q_{x_n}^T \end{pmatrix}$ and $Q_{x_n} Q_{x_n}^T = (I_{n_x} - Q_x Q_x^T)$. Note that the only inversion that needs to be explicitly calculated is of symmetric $R_x R_x^T \in \mathbb{R}^{r_x \times r_x}$.

The matrix $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}$ can be expressed as

$$\begin{aligned}
\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} &= Q_\varepsilon R_\varepsilon R_x^T Q_x^T \left(Q_x (R_x R_x^T)^{-1} Q_x^T + \frac{1}{\kappa_x^2} (I_{n_x} - Q_x Q_x^T) \right) && (6.140) \\
&= Q_\varepsilon R_\varepsilon R_x^T Q_x^T Q_x (R_x R_x^T)^{-1} Q_x^T + \frac{1}{\kappa_x^2} Q_\varepsilon R_\varepsilon R_x^T Q_x^T (I_{n_x} - Q_x Q_x^T) && (6.141) \\
&= Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x^T && (6.142)
\end{aligned}$$

noting that $Q_x^T (I_{n_x} - Q_x Q_x^T) = 0_{r_x, n_x}$. Note that the nullspace component from $\hat{\Gamma}_x^{-1}$ cancels out. The above states the linear correction $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} \in \mathbb{R}^{n_d \times n_x}$ used in BAE as the product of $Q_\varepsilon \in \mathbb{R}^{n_d \times r_x}$, $R_\varepsilon R_x^T (R_x R_x^T)^{-1} \in \mathbb{R}^{r_\varepsilon \times r_x}$ and $Q_x \in \mathbb{R}^{n_x \times r_x}$, potentially saving memory.

Now consider the matrix $\hat{\Gamma}_{\varepsilon|x}$. This can be expressed as

$$m\hat{\Gamma}_{\varepsilon|x} = m\left(\hat{\Gamma}_{\varepsilon} - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_{x}^{-1}\hat{\Gamma}_{x\varepsilon}\right) \tag{6.143}$$

$$= Q_{\varepsilon}R_{\varepsilon}R_{\varepsilon}^{T}Q_{\varepsilon}^{T} - Q_{\varepsilon}R_{\varepsilon}R_{x}^{T}(R_{x}R_{x}^{T})^{-1}Q_{x}^{T}Q_{x}R_{x}R_{\varepsilon}Q_{\varepsilon}^{T} \tag{6.144}$$

$$= Q_{\varepsilon}R_{\varepsilon}R_{\varepsilon}^{T}Q_{\varepsilon}^{T} - Q_{\varepsilon}R_{\varepsilon}R_{x}^{T}(R_{x}R_{x}^{T})^{-1}R_{x}R_{\varepsilon}Q_{\varepsilon}^{T} \tag{6.145}$$

$$= Q_{\varepsilon}\left(R_{\varepsilon}R_{\varepsilon}^{T} - R_{\varepsilon}R_{x}^{T}(R_{x}R_{x}^{T})^{-1}R_{x}R_{\varepsilon}\right)Q_{\varepsilon}^{T} \tag{6.146}$$

where we have made use of our earlier result for $\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_{x}^{-1}$. The covariance $\hat{\Gamma}_{\nu|x}$ can be expressed as

$$\hat{\Gamma}_{\nu|x} = \Gamma_{e} + \hat{\Gamma}_{\varepsilon|x} \tag{6.147}$$

$$= \Gamma_{e} + \frac{1}{m}Q_{\varepsilon}\left(R_{\varepsilon}R_{\varepsilon}^{T} - R_{\varepsilon}R_{x}^{T}(R_{x}R_{x}^{T})^{-1}R_{x}R_{\varepsilon}\right)Q_{\varepsilon}^{T} \tag{6.148}$$

$$= \Gamma_{e} + Q_{\varepsilon}M_{\varepsilon|x}Q_{\varepsilon}^{T} \tag{6.149}$$

where $M_{\varepsilon|x} \in \mathbb{R}^{r_{\varepsilon}\times r_{\varepsilon}}$. Note that $\Gamma_{\nu|x}$ is a rank r_{ε} addition to Γ_{e}.

Let L_{e} be the Cholesky factorisation of Γ_{e} such that $L_{e}L_{e}^{T} = \Gamma_{e}$. Computing $L_{\nu|x}$ becomes a problem of computing a rank r_{ε} update to L_{e}. Methods of doing this are presented in e.g [32]. First, compute $L_{M} \in \mathbb{R}^{r_{\varepsilon}\times r_{\varepsilon}}$ such that

$$L_{M}L_{M}^{T} = M = \frac{1}{m}\left(R_{\varepsilon}R_{\varepsilon}^{T} - R_{\varepsilon}R_{x}^{T}(R_{x}R_{x}^{T})^{-1}R_{x}R_{\varepsilon}\right) \tag{6.150}$$

by e.g. Cholesky factorisation.

Let

$$L_{\nu|x,n_{d}+r_{\varepsilon}} = \begin{pmatrix} L_{e} & Q_{\varepsilon}L_{M}\end{pmatrix} \in \mathbb{R}^{n_{d}\times n_{d}+r_{\varepsilon}} \tag{6.151}$$

and note that

$$L_{\nu|x,n_{d}+r_{\varepsilon}}L_{\nu|x,n_{d}+r_{\varepsilon}}^{T} = \begin{pmatrix} L_{e} & Q_{\varepsilon}L_{M}\end{pmatrix}\begin{pmatrix} L_{e}^{T} \\ L_{M}^{T}Q_{\varepsilon}^{T}\end{pmatrix} \tag{6.152}$$

$$= L_{e}L_{e}^{T} + Q_{\varepsilon}L_{M}L_{M}^{T}Q_{\varepsilon}^{T} \tag{6.153}$$

$$= \Gamma_{e} + Q_{\varepsilon}M_{\varepsilon|x}Q_{\varepsilon}^{T} \tag{6.154}$$

$$= \hat{\Gamma}_{\nu|x} \tag{6.155}$$

i.e. $L_{\nu|x,n_{d}+r_{\varepsilon}}$ is a factor of $\Gamma_{\nu|x}$.

We can instead form $L_{\nu|x} \in \mathbb{R}^{n_d \times n_d}$ such that $L_{\nu|x} L_{\nu|x}^T = \Gamma_{\nu|x}$ as the Cholesky factor. $L_{\nu|x}$ can be found as the solution to

$$Q \begin{pmatrix} L_e^T \\ L_M^T Q_\varepsilon^T \end{pmatrix} = \begin{pmatrix} L_{\nu|x} \\ 0_{r_\varepsilon \times n_d} \end{pmatrix} \tag{6.156}$$

where orthonormal Q is computed by e.g. Givens rotations, forming

$$Q = Q_{n_d+1,1} Q_{n_d+1,2} \dots Q_{n_d+1,n_d} Q_{n_d+2,1} Q_{n_d+2,2} \dots Q_{n_d+k_\varepsilon,n_d} \tag{6.157}$$

where $Q_{j,k}$ zeros the j'th row and k'th column of $\begin{pmatrix} L_e^T \\ L_M^T Q_\varepsilon^T \end{pmatrix}$. This is the *rank r_ε Cholesky update algorithm* as described in [32]. The matrix $\tilde{L}_{\nu|x}$ such that

$$\tilde{L}_{\nu|x}^T \tilde{L}_{\nu|x} = \hat{\Gamma}_{\nu|x}^{-1} \tag{6.158}$$

appears in the BAE posterior. If we computed $L_{\nu|x}$ as a low rank Cholesky update to L_e as discussed above, we could form $\tilde{L}_{\nu|x} = L_{\nu|x}^{-1}$ as

$$\begin{aligned} (L_{\nu|x}^{-1})^T L_{\nu|x}^{-1} \hat{\Gamma}_{\nu|x} &= (L_{\nu|x}^{-1})^T L_{\nu|x}^{-1} L_{\nu|x} L_{\nu|x}^T & (6.159) \\ &= (L_{\nu|x}^{-1})^T I L_{\nu|x}^T = (L_{\nu|x} L_{\nu|x}^{-1})^T = I & (6.160) \end{aligned}$$

as required. Note that $L_{\nu|x}$ as formed above is triangular, so $L_{\nu|x}^{-1}$ can be computed efficiently by e.g. back substitution.

There is an alternative means of computing a matrix $\tilde{L}_{\nu|x}$ such that $\tilde{L}_{\nu|x}^T \tilde{L}_{\nu|x} = \hat{\Gamma}_{\nu|x}^{-1}$. First note that

$$\begin{aligned} \hat{\Gamma}_{\nu|x}^{-1} &= \left(\Gamma_e + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \right)^{-1} & (6.161) \\ &= \Gamma_e^{-1} - \Gamma_e^{-1} Q_\varepsilon \left(M_{\varepsilon|x}^{-1} + Q_\varepsilon^T \Gamma_e^{-1} Q_\varepsilon \right)^{-1} Q_\varepsilon^T \Gamma_e^{-1} & (6.162) \end{aligned}$$

by the matrix inversion lemma. First compute $L_{\hat{M}} \in \mathbb{R}^{k_\varepsilon \times k_\varepsilon}$, the Cholesky factor such that $L_{\hat{M}} L_{\hat{M}}^T = M_{\varepsilon|x}^{-1} + Q_\varepsilon^T \Gamma_e^{-1} Q_\varepsilon$. The inversion $L_{\hat{M}}^{-1}$ can be computed by back substitution. A matrix $\tilde{L}_{\nu|x}$ can then be formed as

$$\tilde{L}_{\nu|x,i} = \begin{pmatrix} \tilde{L}_e^T \\ i (L_{\hat{M}}^{-1})^T Q_\varepsilon^T \Gamma_e^{-1} \end{pmatrix} \tag{6.163}$$

as

$$\tilde{L}^T_{\nu|x,i}\tilde{L}_{\nu|x,i} = \begin{pmatrix}\tilde{L}_e & i\Gamma_e^{-1}Q_\varepsilon L_{\hat{M}}^{-1}\end{pmatrix}\begin{pmatrix}\tilde{L}_e^T \\ i(L_{\hat{M}}^{-1})^T Q_\varepsilon^T \Gamma_e^{-1}\end{pmatrix} \tag{6.164}$$

$$= \tilde{L}_e\tilde{L}_e^T + i^2\Gamma_e^{-1}Q_\varepsilon L_{\hat{M}}^{-1}(L_{\hat{M}}^{-1})^T Q_\varepsilon^T\Gamma_e^{-1} \tag{6.165}$$

$$= \Gamma_e^{-1} - \Gamma_e^{-1}Q_\varepsilon\left(M_{\varepsilon|x}^{-1} + Q_\varepsilon^T\Gamma_e^{-1}Q_\varepsilon\right)^{-1}Q_\varepsilon^T\Gamma_e^{-1} \tag{6.166}$$

as required. Note that $\tilde{L}_{\nu|x,i} \in \mathbb{C}^{(n_d+r_\varepsilon)\times n_d}$. I have not found the above form in the literature, but I expect it is used as an intermediate step in proofs. While algebraically useful, I do not recommend using the above complex number form. Instead,I recommend the Cholesky factor $\tilde{L}_{\nu|x} \in \mathbb{R}^{n_d\times n_d}$ such that $\tilde{L}^T_{\nu|x}\tilde{L}_{\nu|x} = \hat{\Gamma}^{-1}_{\nu|x}$ computed as a *rank r_ε Cholesky downdate*. The Cholesky downdate algorithm is described in [32], and is similar to the update in Equation (6.156).

6.4.2 Recursive QR for BAE

When performing the sampling stage of BAE, it may be useful to know e.g. $\hat{\Gamma}_\varepsilon$ while samples are being computed. This can be useful for deciding when "enough" samples have been taken. It is also possible to have quantities of interest e.g. $\Gamma_{\nu|x}$ computed in parallel with sampling being performed, saving overall computing time.

Suppose we have a set of m_1 samples, and we have computed entities of interest e.g. $\hat{\Gamma}_{(\varepsilon|x)_1}$, the conditional covariance of ε given x based on these first m_1 samples. We then generate an additional m_2 samples, and wish to update these statistical quantities accordingly, to reflect all $m = m_1 + m_2$ samples.

Consider

$$\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} = Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x \tag{6.167}$$

the linear part of the affine BAE correction. The factors Q_ε and Q_x can be updated by concatenating new basis vectors q from the additional m_2 samples. The product $R_\varepsilon R_x^T$ can be updated as

$$R_\varepsilon R_x^T = \begin{pmatrix}R_{\varepsilon,m_1}R_{\varepsilon,m_2}\end{pmatrix}\begin{pmatrix}R_{x,m_1}^T \\ R_{x,m_2}\end{pmatrix} \tag{6.168}$$

$$= R_{\varepsilon,m_1}R_{x,m_1}^T + R_{\varepsilon,m_2}R_{x,m_2}^T \tag{6.169}$$

where the m_1 and m_2 subscripts denote that these came from the m_1 or m_2 samples respectively. In the case that the effective ranks r_x or r_ε are updated i.e. the corresponding Q gained

r_{m_2} columns, we form

$$R_\varepsilon R_x^T = \begin{pmatrix} R_{\varepsilon,m_1} R_{x,m_1}^T & 0_{r_{\varepsilon,m_1}, r_{x,m_2}} \\ 0_{r_{\varepsilon,m_2}, r_{x,m_1}} & 0_{r_{\varepsilon,m_2}, r_{x,m_2}} \end{pmatrix} + R_{\varepsilon,m_2} R_{x,m_2}^T \tag{6.170}$$

to keep the dimensions consistent.

The matrix $(R_x R_x^T)^{-1}$ can be expressed as

$$(R_x R_x^T)^{-1} = \left(\begin{pmatrix} R_{x,m_1} R_{x,m_2} \end{pmatrix} \begin{pmatrix} R_{x,m_1}^T \\ R_{x,m_2} \end{pmatrix} \right)^{-1} \tag{6.171}$$

$$= \left(\begin{pmatrix} R_{x,m_1} R_{x,m_1}^T & 0_{r_{x,m_1}, r_{x,m_2}} \\ 0_{r_{x,m_2}, r_{x,m_1}} & 0_{r_{x,m_2}, r_{x,m_2}} \end{pmatrix} + R_{\varepsilon,m_2} R_{x,m_2}^T \right)^{-1} \tag{6.172}$$

$$= \left(M_{x,m_1} + R_{x,m_2} R_{x,m_2}^T \right)^{-1} \tag{6.173}$$

where we note that

$$M_{x,m_1}^\dagger = \begin{pmatrix} (R_{x,m_1} R_{x,m_1}^T)^{-1} & 0_{r_{x,m_1} \times r_{x,m_2}} \\ 0_{r_{x,m_2} \times r_{x,m_1}} & 0_{r_{x,m_2} \times r_{x,m_2}} \end{pmatrix}. \tag{6.174}$$

The matrix (pseudo)inversion lemma [107] is applied to form

$$(R_x R_x^T)^{-1} = \left(M_{x,m_1} + R_{x,m_2} R_{x,m_2}^T \right)^{-1} \tag{6.175}$$

$$= M_{x,m_1}^\dagger - M_{x,m_1}^\dagger R_{x,m_2} (R_{x,m_2}^T R_{x,m_2} + I_{m_2})^{-1} R_{x,m_2}^T M_{x,m_1}^\dagger \tag{6.176}$$

where we note that $(R_{x,m_1} R_{x,m_1}^T)^{-1}$ is reused, and the only inverse that needs to be calculated is $(R_{x,m_2}^T R_{x,m_2} + I_{m_2})^{-1} \in \mathbb{R}^{m_2 \times m_2}$. The inversion for the $m_2 = 1$ case i.e. continually updating, is particularly straightforward. By constantly updating i.e. having small m_2, the BAE correction $\hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1}$ is computed almost immediately after sampling concludes.

Other sample enntities of interest e.g.

$$\hat{\Gamma}_{\varepsilon|x} = Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \tag{6.177}$$

can be formed similarly e.g. making rank m_2 updates of

$$M_{\varepsilon|x} = \frac{1}{m} \left(R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x^T (R_x R_x^T)^{-1} R_x R_\varepsilon^T \right) \tag{6.178}$$

similar to updating $(R_x R_x^T)^{-1}$ and $R_\varepsilon R_x^T$ as shown above. This allows the quantities of interest in BAE to be computed in a compressed form almost immediately after sampling concludes.

The proposed combination of computing QR factors of sample matrices as samples are generated with a tolerance to reduce rank, and using the QR factors to compute sample statistics of interest e.g. $\hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} = Q_\varepsilon R_\varepsilon R_x (R_x R_x^T)^{-1} Q_x^T$, and computing these entities iteratively is referred to in this thesis as *recursive QR*. Recursive QR is used in Chapters 8, 9 and 10.

6.4.3 Example: 1D Deconvolution

Consider the 1D deconvolution problem approached with BAE from Section 6.3.3. In this section recursive QR is incorporated into the solution scheme.

Samples $\bar{x}_j$, $\bar{z}_j$ and $\bar{\xi}_j$ are drawn as in Section 6.3.3. These samples are used to compute $x_j = P\bar{x}_j$ and $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - Ax_j$. As each sample arrives, Q_x, Q_ε, $R_\varepsilon R_\varepsilon^T$, $R_x R_x^T$ and $R_\varepsilon R_x^T$ are computed as described in Section 6.4.2. I used tolerances $\tau_x = 0.01$ on the X samples for QR decomposition and $\tau_\varepsilon = 0.01$ on the ε samples. This gave $n_x = r_x = 51$ and $51 = n_\varepsilon > r_\varepsilon = 40$. The effective ranks are high relative to the problem dimension, however the problem dimensions are relatively low due to BAE.

For this example, I let $m = 200$ samples be drawn before computing

$$(R_x R_x^T)^{-1} \tag{6.179}$$

$$R_\varepsilon R_x^T (R_x R_x^T)^{-1} \tag{6.180}$$

$$\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} = Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x^T \tag{6.181}$$

$$M_{\varepsilon|x} = \frac{1}{m}\left(R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x^T (R_x R_x^T)^{-1}(R_\varepsilon R_x^T)^T\right) \tag{6.182}$$

$$L_M = \text{Chol}(M) \tag{6.183}$$

i.e. I do not recursively update these quantities. I then compute

$$\begin{pmatrix} L_{\nu|x} \\ 0_{r_\varepsilon, n_d} \end{pmatrix} = Q \begin{pmatrix} L_e^T \\ L_m^T Q_\varepsilon^T \end{pmatrix} \tag{6.184}$$

$$\tilde{L}_{\nu|x} = L_{\nu|x}^{-1} \in \mathbb{R}^{n_d \times n_d} \tag{6.185}$$

where Q is a product of Givens rotations. Note that the inversion can be performed by back substitution.

The model in this problem is of the form $d = Ax + \varepsilon + e$ where x and e are normally distributed independent variables, and ε is approximated as conditionally normal on x. Consider a particular realisation $d_t = Ax_t + \varepsilon_t + e_t$. The MAP estimate of x_t given d_t is

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}\left(d_t - Ax - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}x - \hat{\mu}_\varepsilon - \mu_e\right)\right\|_2^2 + \left\|\tilde{L}_x x\right\|_2^2 \right\} \tag{6.186}$$

$$= \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d_t - \mu_e - \hat{\mu}_\varepsilon) \\ 0_{n_x,1} \end{pmatrix} \tag{6.187}$$

and the posterior covariance is

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A^T \Gamma_{\nu|x}^{-1} A)^{-1} \tag{6.188}$$

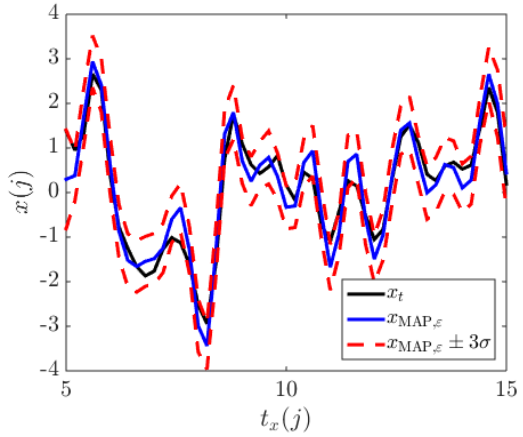


Figure 6.17: MAP estimate with posterior error estimates found with BAE and recursive QR alongside the ground truth

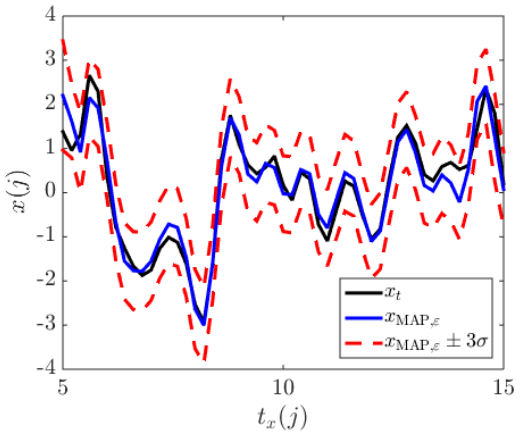


Figure 6.18: MAP estimate with posterior error estimates found with standard BAE alongside the ground truth

where we note that Γ_x^{-1} and $\tilde{L}_x$ came from the prior, not samples.

The estimates found with BAE and recursive QR are shown in Figure 6.17. The BAE estimates found without recursive QR are shown in Figure 6.18. The MAP estimates are quite similar, although the recursive QR solution is slightly more "jagged". The posterior error estimate with recursive QR is slightly narrower, but still representative. The recursive QR posterior error estimates appear narrower than they really are due to the "jagged" MAP estimate.

I recomputed the estimates with $\tau_x = 1$ and $\tau_\varepsilon = 0.4$. This results in $k_x = 36$ and $k_\varepsilon = 22$. The associated estimates are shown in Figure 6.19. Note that the solution seems overly nonsmooth, and the posterior error intervals are too narrow to represent the ground truth. While the tolerances seem to have been too large in this example, the estimates in Figure 6.19 are still more representative than the estimates seen in Figure 6.15, which ignore approximation error.

6.5 Analytical Decompositions of Priors

This section reviews some useful analytical results for commonly used covariance functions. In [93], the eigenvalues and eigenvectors of a Gaussian smoothness prior covariance in 1D are expressed analytically. This allows useful aspects of the covariance e.g. the rank, L_x and $\tilde{L}_x$ to be "computed" efficiently.

Similarly, analytical eigendecompostions exist for a variety of covariance functions in hypercubes, see [73]. It is worth noting that these are often derived for continuous forms of the

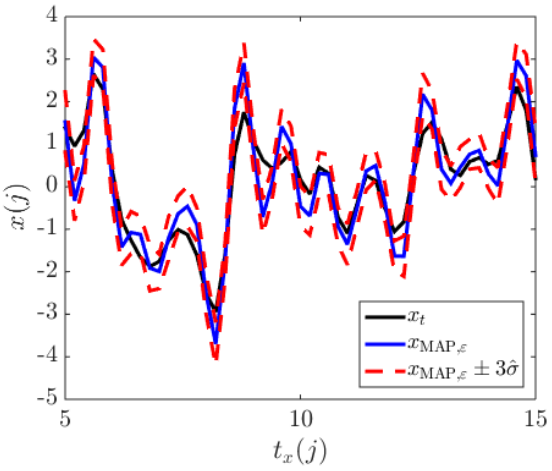


Figure 6.19: MAP estimate with posterior error estimates found with BAE and recursive QR with $\tau_x = 1$ and $\tau_\varepsilon = 0.4$ alongside the ground truth

domain, using a PDE formulation of the prior. Such formulations could potentially be used to phrase the eigenproblem in a way that is solved by e.g. FEM rather than constructing the covariance and using linear algebra techniques. This concept is discussed further in [94].

I stress that the analytical results mentioned above only exist for very narrowly defined problems. Having the analytical eigendecomposition for a homogeneous smoothness prior on a rectangle can be useful at early stages of analysis. If we were instead to use a structured prior or a nonrectangular domain, analytical results are not available.

6.6 Randomised Compression

The matrices considered in inverse problems can be prohibitively large, motivating the use of compressed approximations e.g. thin SVD. A framework that has seen recent popularity is *randomised compression*. This section primarily considers the recent landmark paper [37].

Consider $A \in \mathbb{R}^{n_d \times n_x}$. If n_d and/or n_x are extremely large, computations involving A become prohibitively computationally expensive. For example, A may be so large that it does not fit in fast (e.g. RAM) memory. A decomposed form of A e.g. the thin SVD $A = U_{r_t} D_{r_t} V_{r_t}^T$ may be more useful. For example, it may be that the thin SVD fits in fast memory while A does not. It is also trivial to compute the pseudoinverse of A once the thin SVD is known. Approximate decompositions e.g. the tSVD $A \approx A_r = U_r D_r V_r^T$ may similarly be useful.

The use of factorisations for linear algebraic tasks is widespread [108, 109]. The idea of using a randomised method for factorising a large matrix has existed for some time, for example

randomly selecting a few rows and/or columns of a matrix and treating this as representative [110, 111, 112]. More recent methods consider the product of A with a random *test matrix* W. This concept is explored thoroughly in [37]. In this thesis, compression based on analysis of AW is called *randomised compression*. Randomised compression has similarities to subspace iteration with a randomised initial subspace, a common method in linear algebra reviewed in [32]. Key papers in the development of randomised compression are [112, 113, 114].

Consider computing the r term thin SVD of $A \in \mathbb{R}^{n_d \times n_x}$. A method proposed in [37] is:

1. Form white noise test matrix $W \in \mathbb{R}^{n_x \times 2r}$ i.e. $W(j,k) \sim \mathcal{N}(0,1)$ for all j,k.
2. Form $Y = (AA^T)^q AW$ with $q = 1, 2$ by multiplying alternately with A and A^T e.g. for $q = 1$, $Y = A\Big(A^T(AW)\Big)$. Note that $Y \in \mathbb{R}^{n_d \times 2r}$.
3. Construct orthonormal Q that spans columns of Y e.g. $Y = QR$. Note $Q \in \mathbb{R}^{n_d \times \hat{r}}$ where $\hat{r} \leq 2r$ is the effective rank of Y.
4. Form $B = Q^T A$. Note that $B \in \mathbb{R}^{\hat{r} \times n_x}$.
5. Compute the r term thin SVD $B = \tilde{U}_r D_r V_r^T$ where $r \leq \hat{r}$.
6. Set $U_r = Q\tilde{U}_r$

and thus $A \approx U_r D_r V_r^T$. The above is known as the *randomised SVD algorithm*. Methods of computing other factorisations are also discussed in [37], where steps 1-3 are largely the same. Methodological refinements are also provided in [37] to e.g. reduce roundoff errors. An initial methodological improvement proposed in this thesis is that $Y = QR$ can be computed by recursive QR as described in Section 6.4.1.

The above methodology has some distinct advantages. Namely, the algorithm is simple to parallelise, and the matrix A only needs to be streamed through fast memory $2(q+1)$ times. Each computation of a matrix-matrix product is a called a *pass* of A. For example, computing the SVD as described in the above steps with $q = 0$ is a 2 pass algorithm.

6.6.1 Localised Compression

Consider $A \in \mathbb{R}^{n_d \times n_x}$. Let $A_c = U_r D_r V_r^T \approx A$ be the low rank SVD approximation to A. A typical quantification of the quality of a matrix approximation would be

$$\|A - A_c\|_2 = \max_{x:\|x\|_2=1} \|Ax - A_c x\|_2 \tag{6.189}$$

i.e. the difference in matrix 2-norm/the largest singular value of the difference. Such a measure of approximation quality is natural if the aim is to have $Ax \approx A_c x$ for all $x \in \mathbb{R}^{n_x}$. In this

section, I consider forming A_c such that $Aa \approx A_c x$ when x is known to more likely be in a subset of $\mathbb{R}^{n_x}$.

Suppose we wish to approximate the matrix

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{6.190}$$

that is, find some A_c such that $Ax \approx A_c x$. However, we may know that x is always of the form $x = (x(1), 0)^T$. In this case, the approximation A_c only needs to be valid in the 1 dimensional subspace x exists in. In this case, the product Ax is always of the form

$$Ax = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x(1) \\ 0 \end{pmatrix} = \begin{pmatrix} x(1) \\ 3x(1) \end{pmatrix} \tag{6.191}$$

so a valid approximation is

$$A_c = \begin{pmatrix} 1 & 0 \\ 3 & 0 \end{pmatrix} \tag{6.192}$$

which I term the *local approximation* of A. The accuracy of the approximation, quantified as $||Ax - A_c x||_2$, is shown in Figure 6.20. Note that $||Ax - A_c x||_2 = 0$ along $x(2) = 0$.

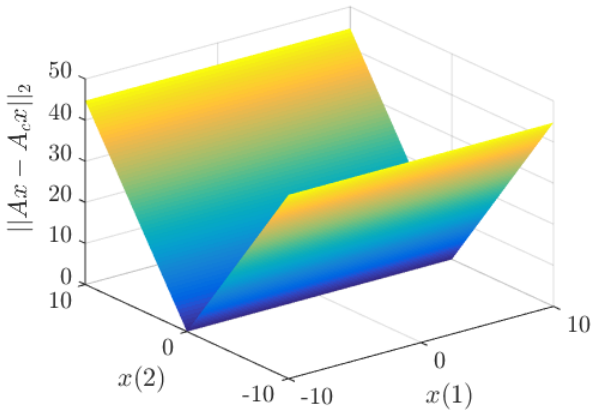


Figure 6.20: Comparing A and local approximation A_c

Consider the randomised compression of [37]. Instead of a white noise matrix

$$W = (w_1, w_2, \ldots, w_{2r}) \in \mathbb{R}^{n_x \times 2r} \tag{6.193}$$

where w_j is a sample of $w \sim \mathcal{N}(0_{n_x}, I_{n_x})$, I propose using a sample matrix

$$X = (x_1, x_2, \ldots, x_{2r}) \in \mathbb{R}^{n_x \times 2r} \tag{6.194}$$

where x_j is a sample of $x \sim \pi_x(x)$. I refer to approximations $A_{c,w}$ found by considering the action of A on W as *global sample approximations*, as $A_{c,w}$ would be expected to be "equally accurate" for all $x \in \mathbb{R}^{n_x}$ (relative to the norm of x). I refer to approximations $A_{c,x}$ found by considering the action of A on X as *local sample approximations*, as $A_{c,x}$ would be expected to be accurate for $x \sim \pi_x(x)$.

An example of a local approximation is the conditional mean. The conditional mean $\mu_{y|x}$ approximates $y = A(x)$ as $y \approx \mu_{y|x}$ i.e.

$$A(x) \approx \mu_y + \Gamma_{yx}\Gamma_x^{-1}(x - \mu_x) \tag{6.195}$$

where the approximation is exact for normally distributed x with defined Γ_x^{-1} and linear A. Note that the CM is an affine approximation. The sample conditional mean

$$\mu_{y|x} \approx \hat{\mu}_{y|x} = \hat{\mu}_y + \hat{\Gamma}_{yx}\hat{\Gamma}_x^{-1}(x - \hat{\mu}_x) \tag{6.196}$$

is a *local sample approximation* to $A(x)$. Note that the sample conditional mean is constructed just from samples of x and y, and can be constructed for nonlinear A and non-Gaussian x.

Recall that the methods in [37] used a white noise matrix W for sampling. That is, we treat $x \sim \mathcal{N}(0_{n_x,1}, I_{n_x})$. The conditional mean in this case is

$$\begin{aligned}
\mu_{y|x} &= \mu_y + \Gamma_{yx}\Gamma_x^{-1}(x - \mu_x) && (6.197)\\
&= A\mu_x + \Gamma_{yx}I_{n_x}^{-1}(x - \mu_x) && (6.198)\\
&= \Gamma_{yx}x && (6.199)
\end{aligned}$$

as $\mu_x = 0_{n_x,1}$ and $I_{n_x}^{-1} = I_{n_x}$. The cross covariance Γ_{yx} is

$$\begin{aligned}
\Gamma_{yx} &= \mathbb{E}\bigg((y - \mu_y)(x - \mu_x)^T\bigg) && (6.200)\\
&= \mathbb{E}(Axx^T) = A\mathbb{E}(xx^T) && (6.201)\\
&= A && (6.202)
\end{aligned}$$

i.e. the conditional mean approximation to a linear operator, assuming a white noise prior, is identical to the linear operator.

Consider the 1D convolution problem of Section 2.3.2, modelled as $d = Ax + e$. Consider the sample conditional mean used as a global sample approximation. First compute $Y_w = AW$ where $W(:, j) = w_j$, a sample of $w \sim \mathcal{N}(0_{n_x,1}, I_{n_x})$. This is used to compute

$$\begin{aligned}
A_{c,w} &= \hat{\Gamma}_{y_w w}\hat{\Gamma}_w^{-1} && (6.203)\\
&= Q_{y_w}R_{y_w}R_w(R_wR_w^T)^{-1}Q_w^T && (6.204)
\end{aligned}$$

where QR factorisation of sample matrices is performed as described in Section 6.4.1. Note that $A_{c,w}x_t = \hat{\mu}_{y|w=x_t}$, the "zero mean" conditional mean. In this thesis, factorised forms of sample approximations are referred to as *sample compressions*.

Let $x \sim \pi_x(x) = \mathcal{N}(0_{n_x,1}, \Gamma_x)$ where $\pi_x(x)$ is a smoothness prior (specifically the smoothness prior used in Section 5.1.1). I compute sample matrix $X = L_x W$ where $L_x L_x^T = \Gamma_x$. I also compute $Y_x = AX = AL_x W$. I factorise $Y_x = Q_{y_x} R_{y_x}$ and $X = Q_x R_x$ as described in Section 6.4.1. I then form the local sample approximation/compression

$$A_{c,x} = Q_{y_x} R_{y_x} R_x (R_x R_x^T)^{-1} Q_x^T \tag{6.205}$$

the equivalent approximation to $A_{c,w}$ but constructed from X and Y_x rather than W and Y_w.

I construct the approximations $A_{c,w}$ as in Equation (6.203) and $A_{c,x}$ as in Equation (6.205) to the convolution operator $A \in \mathbb{R}^{51\times 201}$ of Section 2.3.2. I form approximations from 20, 50, 100 and 200 samples. I compare the "accuracy" by plotting Ax_t, $A_{c,w}x_t$ and $A_{c,x}x_t$ alongside each other, where x_t is a draw of x. These are shown in Figures 6.21 through 6.24. I also plot Aw_t, $A_{c,w}w_t$ and $A_{c,x}w_t$ where w_t is a draw of $w \sim \mathcal{N}(0_{n_x,1}, I_{n_x})$. These are shown in Figures 6.25 through 6.28. Note that for the same number of samples, $\|A_{c,x}x_t - Ax_t\| < \|A_{c,w}x_t - Ax_t\|$ and $\|A_{c,x}w_t - Aw_t\| < \|A_{c,w}w_t - Aw_t\|$ i.e. for this problem, the proposed method of local sample approximations outperforms the global sample approximation.

The approximations of Equation (6.203) and Equation (6.205) are computed with a single pass of A. I now compute approximations with the $2(q+1)$ pass randomised SVD algorithm of [37]. I take $q = 1$. Let $A_{c,w}$ be computed from $Y_w = A(A^T(AW))$ and $A_{c,x}$ be computed from $Y_x = A(A^T(AX))$. I consider $r = 15$ and $r = 25$ i.e. $W, X \in \mathbb{R}^{201\times 30}$ and $W, X \in \mathbb{R}^{201\times 50}$. The products Aw_t, $A_{c,x}w_t$, $A_{c,w}w_t$, Ax_t, $A_{c,x}x_t$ and $A_{c,w}x_t$ for $r = 15$ and $r = 25$ are shown in Figures 6.29 to 6.32. Note that the local sample approximation $A_{c,x}$ appears to be a better approximation to A. For this problem, the proposed extension to [37] of local sample approximations found by using samples x_j of $x \sim \pi_x(x)$ better approximates both Ax and Aw.

6.6.2 Sample Approximations and Matrix Free Methods

Sample based approximation methods are useful for creating an approximation to an operator $\mathbf{A}$ when only the action of $\mathbf{A}$ i.e. $y = \mathbf{A}(x)$ can be accessed. Suppose we wanted the matrix approximation A of the linear operator $\mathbf{A}$ where $\mathbf{A}(x)$ is computed using matrix free methods or black box software. What is sometimes done in practice for linear $\mathbf{A}$ is to compute A "row by row", forming $A(i,:) = \mathbf{A}(\mathbf{e}_i)$ where $\mathbf{e}_i \in \mathbb{R}^{n_x}$ is the i'th identity vector. This construction involves n_x evaluations of $\mathbf{A}$. It may also be that a compressed approximation $A_c \approx A$ is desired.

Rather than "row by row", I propose constructing a local sample approximation A_c to $\mathbf{A}$. This involves computing $\mathbf{A}(x_j)$ where $x_j \in \mathbb{R}^{n_x}$ is a draw of $x \sim \pi_x(x)$, as opposed to the "row by row" method, which involves computing $\mathbf{A}(\mathbf{e}_i)$.

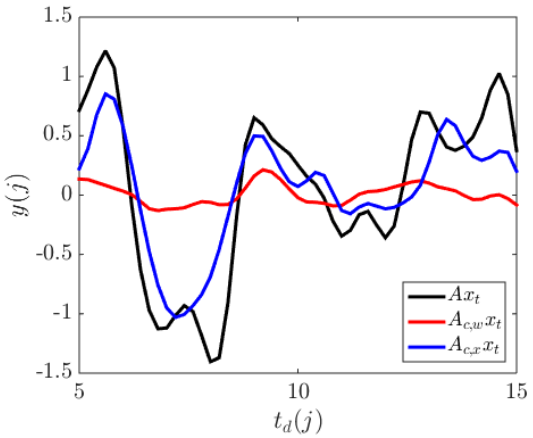


Figure 6.21: Sample approximations from 20 samples

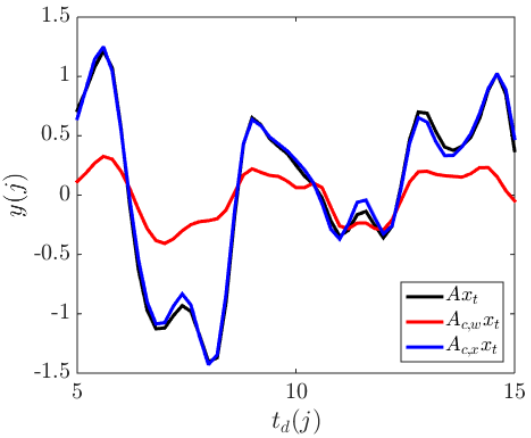


Figure 6.22: Sample approximations from 50 samples

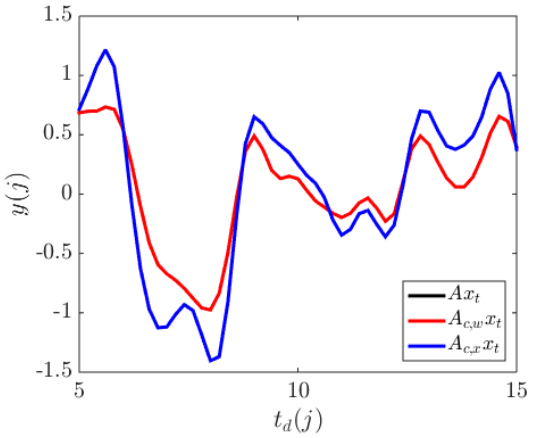


Figure 6.23: Sample approximations from 100 samples

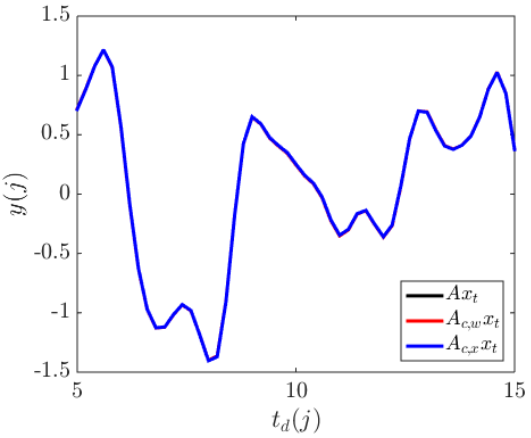


Figure 6.24: Sample approximations from 200 samples

The methods of [37] are multipass methods, requiring the ability to compute $\mathbf{A}^T x$ and $Q^T \mathbf{A}$. In the case of matrix free methods, we may not be able to compute these quantities. We shall instead discuss the single pass "conditional mean" approximation i.e. $\mathbf{A}(x) \approx A_c(x) = \hat{\mu}_{y|x}$.

Let $Y = (y_1, y_2, \ldots, y_m)$ and $X = (x_1, x_2, \ldots, x_m)$ be m sample data matrices with $y_j = \mathbf{A}(x_j)$ and $Q_y R_y$ and $Q_x R_x$ are mean removed QR factorisations. A local sample

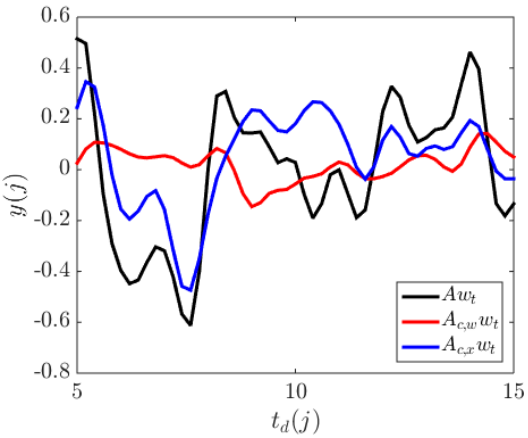


Figure 6.25: Sample approximations from 20 samples

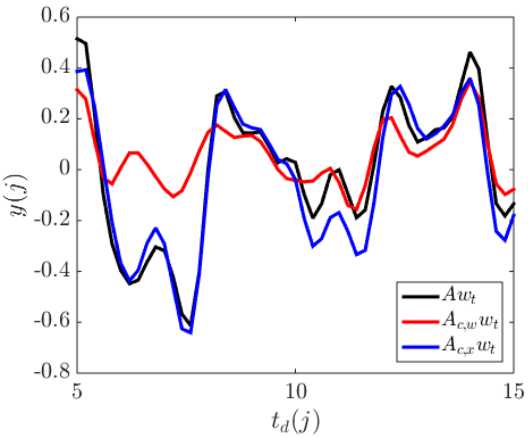


Figure 6.26: Sample approximations from 50 samples

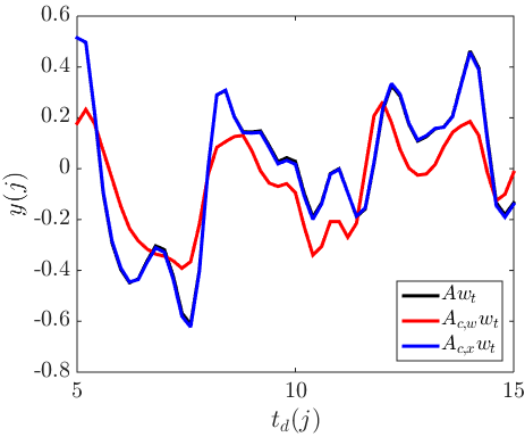


Figure 6.27: Sample approximations from 100 samples

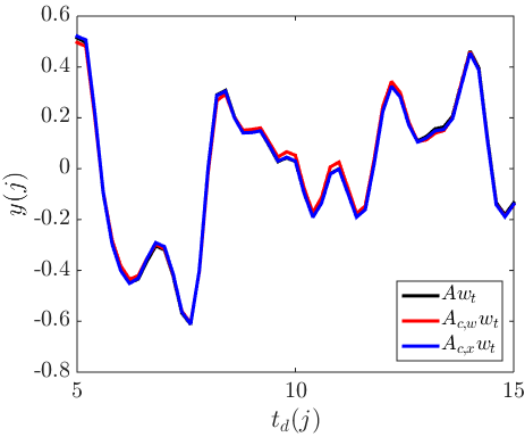


Figure 6.28: Sample approximations from 200 samples

approximation A_c can be formed as

$$A_c(x) = \hat{\mu}_{y|x} = \hat{\mu}_y + \hat{\Gamma}_{yx}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) \tag{6.206}$$

$$= \hat{\mu}_y + Q_y R_y R_x^T (R_x R_x)^{-1} Q_x^T (x - \hat{\mu_x}) \tag{6.207}$$

which is the affine single pass local sample approximation to $A(x)$. Recall that Q_x, Q_y, $R_y R_x^T$, $R_x R_x^T$ and $(R_x R_x)^{-1}$ can all be computed as samples $y_j = \mathbf{A}(x_j)$ are being generated, as described in Section 6.4.2. This allows A_c to be formed almost immediately after the final

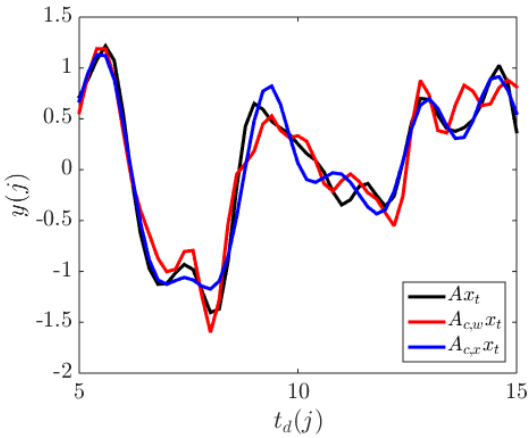


Figure 6.29: Sample SVD approximation, $r = 15$, 30 samples, 4 passes

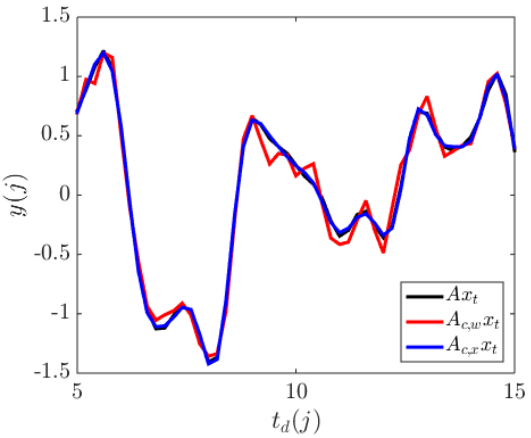


Figure 6.30: Sample SVD approximation, $r = 25$, 50 samples, 4 passes

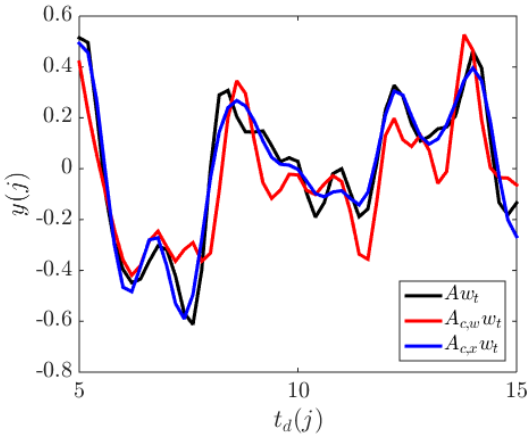


Figure 6.31: Sample SVD approximation, $r = 15$, 30 samples, 4 passes

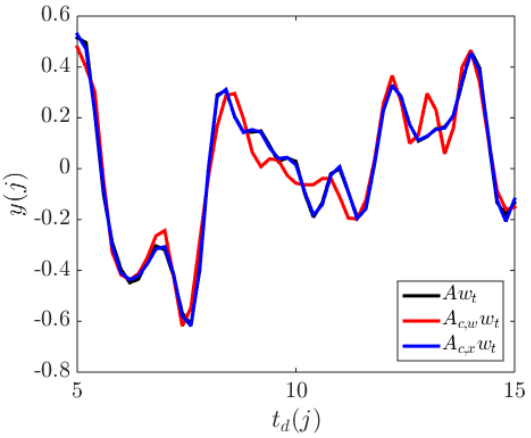


Figure 6.32: Sample SVD approximation, $r = 25$, 50 samples, 4 passes

sample y_m is computed. Note that the A_c as formulated in Equation (6.206) is identical to using the BAE framework with $\bar{A}(\bar{x}, \bar{z}, \bar{\xi}) = \bar{A}(x)$ and having "approximate" model $A(x) = 0_{n_d,1}$.

Consider the accuracy of this approximation. When using recursive QR with a tolerance τ, we know that $||\mathbf{A}(x_j) - A_c x_j||_2^2 < \tau$ by construction. This is an important consideration when choosing τ. When evaluating the quality of the approximation as $||\mathbf{A}(x_j) - A_c x_j||_2$ with $j > m$ i.e. prior draws not used in the sample approximation. The aim is to choose m so large that the prior is "exhausted" in the sense that $||\mathbf{A}(x_j) - A_c x_j||_2^2 < \tau$ for any x_j drawn from

$\pi_x(x)$.

Note that A_c as formulated in Equation (6.206) can be formed even for nonlinear $\mathbf{A}$. It could be that $\mathbf{A}$ is "sufficiently linear" for x "sufficiently likely" i.e. $\|\mathbf{A}(x) - A_c x\|_2$ is acceptably small when $\pi_x(x)$ is acceptably large. This is a problem specific judgement call, and depends on how strict $\pi_x(x)$ is, and how linearly $\mathbf{A}$ behaves in the support of $\pi_x(x)$. The nonlinear problem of Chapter 10 is approximated effectively with a local sample approximation similar to that of Equation (6.206).

Let the local sample approximation around $x = x_0$ be expressed as

$$A(x) \approx A(x_0) + A_c(x - x_0). \tag{6.208}$$

Compare this with the 1 term Taylor expansion about x_0,

$$A(x) \approx A(x_0) + J_{x_0}(x - x_0). \tag{6.209}$$

It then follows that the Jacobian can be approximated with a local sample approximation i.e. take $J_{x_0} \approx A_c$. This is potentially useful when samples $y_j = A(x_0 + x_j)$ can be computed rapidly, but explicitly forming J is slow. Approximating the Jacobian with a sample approximation is conceptually similar to the *ensemble smoother* [115, 116]. The ensemble smoother is a sample approximation of the forward operator Jacobian. The local sample approximation is simple to interpret and allows for improvements such as incorporating the adjoint and making additional model passes. I also incorporate recursive QR to the local sample approximation to arrive at a low rank approximation.

Consider the nonlinear operator $y = A(x) = \exp(-\frac{x}{2})$ with $n_x = n_y = 1$. Let $x \sim \mathcal{N}(\mu_x, \sigma_x^2) = \mathcal{N}(3, 1)$. The Jacobian based approximation is $A(x) \approx A_J(x) = A(\mu_x) + J_{\mu_x}(x - \mu_x)$. While this makes sensible use of our knowledge of μ_x, it ignores any additional knowledge of the prior. The single pass sampling approach would be to draw several x_j from $\pi(x)$, compute $y_j = A(x_j)$ and form the approximation $A(x) \approx A_c(x) = \hat{\mu}_y + \hat{\Gamma}_{yx}\hat{\Gamma}_x^{-1}(x - \hat{\mu}_x)$. These approximations are shown in Figure 6.33 alongside the "true" $y = A(x)$. The errors in the approximations are shown in Figure 6.34.

6.7 Combining Local Approximatons and BAE

Let $d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e$, and suppose samples $\bar{x}_j$, $\bar{z}_j$, $\bar{\xi}_j$ and x_j can be drawn, where x is a sufficient resolution approximation of the unknowns of interest. Let $d_t = \bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t) + e_t$ be a particular realisation. We wish to estimate $x_t = P_{x,\bar{x}}\bar{x}_t$ from d_t. A potentially computationally efficient method would be to construct a locally accurate model $A_c^\dagger \in \mathbb{R}^{n_x \times n_d}$ such that the estimate of x_t can be computed as $x_t \approx x_{\text{lc}} = A_c^\dagger d$. In this section, a method of constructing $A_c^\dagger$ using concepts from BAE and sample approximations is proposed. This method can be applied to nonlinear $\bar{A}$ and non-Gaussian priors.

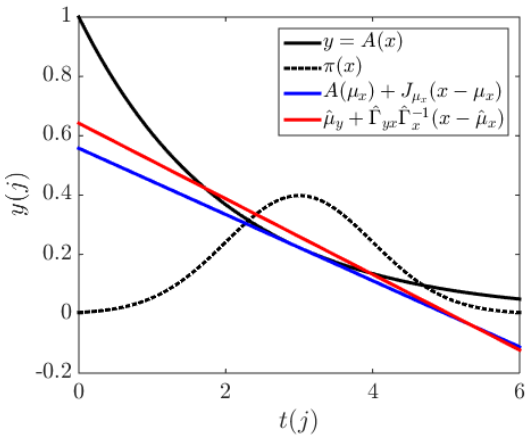


Figure 6.33: Some linear approximations

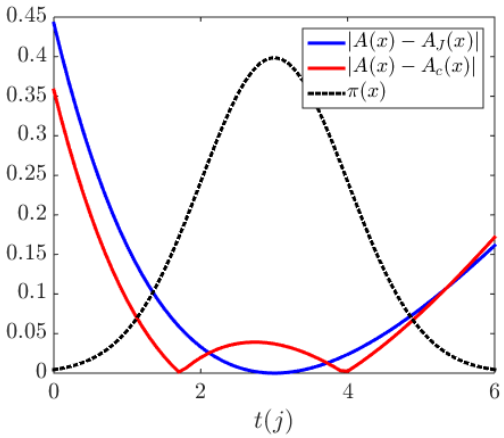


Figure 6.34: Errors in approximations

The conceptually simplest method is to construct $A_c^\dagger(d_t) = \hat{\mu}_{x|d=d_t}$ entirely from samples $d_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) + e_j$ and $x_j = P_{x,\bar{x}}\bar{x}_j$. That is, we compute $Q_d R_d$ and $Q_x R_x$ from mean removed samples d_j and x_j, then form the estimate

$$x_t \approx \hat{\mu}_{x|d=d_t} = A_c^\dagger(d_t) = \hat{\mu}_x + Q_x R_x R_d^T (R_d R_d^T)^{-1}(d - \hat{\mu}_d). \tag{6.210}$$

Note that this estimator is constructed with samples from the accurate model $\bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j)$. Such samples are computed in BAE, so the above estimator can be constructed while attempting BAE with minimal additional computations. It may also be worthwhile to construct other approximations e.g $A_{\bar{c}}(\bar{x})$, a higher dimensional affine approximation to $\bar{A}$. This allows us to assess the importance of e.g. nonlinearity, additional unknowns, etc. If we compute the linear inverse models $A_c^\dagger$, we can also see "cheaply" how inversions may behave.

Incorporating BAE into sample approximation effectively replaces the forward mapping $A_c(x)$ with

$$A_c(x) + \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}). \tag{6.211}$$

Recall that samples $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - A_c(x_j)$ are used in constructing $\hat{\mu}_{\varepsilon|x}$ and samples $y_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) = \varepsilon_j + A_c(x_j)$ are used in constructing A_c. The forward model approximated with a single pass approximation A_c and BAE can be expressed as

$$\begin{aligned} y = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) &\approx A_c(x) + \hat{\mu}_{\varepsilon|x} && (6.212)\\ &= \hat{\mu}_y + \hat{\Gamma}_{yx}\hat{\Gamma}_x^{-1}(x - \hat{\mu}_x) + \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) && (6.213)\\ &= \hat{\mu}_y + (\hat{\Gamma}_{yx} + \hat{\Gamma}_{\varepsilon x})\hat{\Gamma}_x^{-1}(x - \hat{\mu}_x) && (6.214) \end{aligned}$$

where the same samples are used for computing A_c and $\hat{\mu}_{\varepsilon|x}$. A consequence of sharing samples is that $\hat{\mu}_\varepsilon = 0_{n_d,1}$ and $\hat{\Gamma}_{\varepsilon x} = 0_{n_d,n_x}$. In other words, the forward models $y = A_c(x)$, $y = A_c(x) + \hat{\mu}_{\varepsilon|x}$ and $y = \hat{\mu}_{\varepsilon|x}$ i.e. BAE with the $A = 0_{n_d,n_x}$ model, will all be the same when computed from the same samples.

Consider the model $d = A_c(x) + e$ where A_c is a model approximation such as described in Equation (6.205) and x and e are normally distributed independent variables. Let a realisation be $d_t = A_c(x_t) + e_t$. The MAP estimate $x_{c,\text{MAP},\varepsilon=0}$ of x_t from d_t can be computed as

$$x_{c,\text{MAP},\varepsilon=0} = \begin{pmatrix} \tilde{L}_e A_c \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_e(d - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.215}$$

i.e. the approximation can be naturally substituted in like any other model A. Consider the BAE model $d = A_c(x) + \hat{\mu}_{\varepsilon|x} + e$. The MAP estimate $x_{c,\text{MAP}}$ of x_t given d_t with BAE is

$$x_{c,\text{MAP}} = \begin{pmatrix} \tilde{L}_{\nu|x}(A_c + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d - \mu_e - \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.216}$$

recalling that the quantities $\tilde{L}_{\nu|x}$ and $\tilde{L}_{\nu|x}$ are found by sampling. So while combining BAE and local sample approximations may not change the forward predictions, the MAP estimate will be changed. The uncertainty estimates are also changed when incorporating BAE.

I combine local sample approximation and BAE for the 1D deconvolution problem of Section 2.3.2. This analysis is similar to the example of Section 6.3.3 but with a higher noise variance σ_e^2 and incorporating a local sample approximation model. Recall that $\bar{x}$ is a fine discretisation on $[5, 15]$ while $\bar{z}$ is a fine discretisation on $[0, 5)$ and $(15, 20]$. Let $\bar{s} = (\bar{x}^T, \bar{z}^T)^T$. The kernel $\bar{\xi}$ is the true convolution kernel. Let $e \sim \mathcal{N}(\mu_e = 0_{n_d,1}, \Gamma_e = \sigma_e^2 I_{n_d})$ with $\sigma_e = 0.1$, approximately 10% noise level. Let $\pi_{\bar{s}}(\bar{s})$ be a Gaussian smoothness prior. I estimate x_t, a coarse discretisation on $[5, 15]$, from data d_t. Shown in Figure 6.35 is a draw $\bar{s}_t$ from the prior on $\bar{s}$, as well as a particular evaluation $\bar{c}_t = \bar{F}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t)$ where $\bar{F}$ is the "physics" convolution model. The data $d_t = \bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t) + e_t$ is shown in Figure 6.36. The covariance Γ_e is visualised in Figure 6.37 and the covariance $\Gamma_{\nu|x} = \Gamma_e + \Gamma_{\varepsilon|x}$ is visualised in Figure 6.38. Note the colourbars. Note the variance of $\Gamma_{\nu|x}$ is greater for nodes corresponding to around $t = 5$ and $t = 15$.

The accurate forward predictions $\bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t)$, the local sample approximation forward predictions $A_c(x_t)$ and the BAE corrected forward predictions $A_c(x_t) + \hat{\mu}_{\varepsilon|x=x_t}$ are shown in Figure 6.39. Note that the local sample predictions and the BAE corrected predictions are the same i.e. $\hat{\mu}_{\varepsilon|x} = 0_{n_d,1}$ as predicted when A_c is a single pass approximation and $\hat{\mu}_{\varepsilon|x}$ is constructed from the same samples.

The MAP estimate with $3\hat{\sigma}$ posterior error intervals found with BAE and a local sample

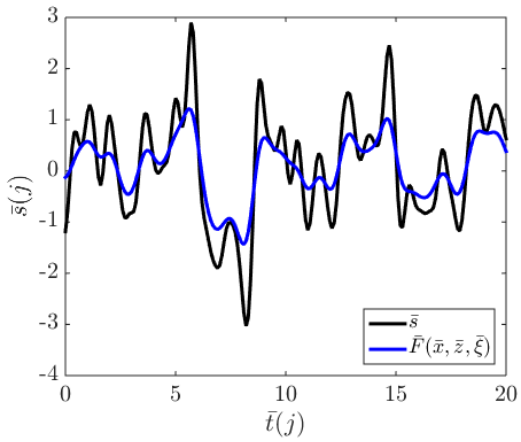


Figure 6.35: Unconvolved $\bar{s}_t$ and convolved $\bar{c}_t = \bar{F}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t)$

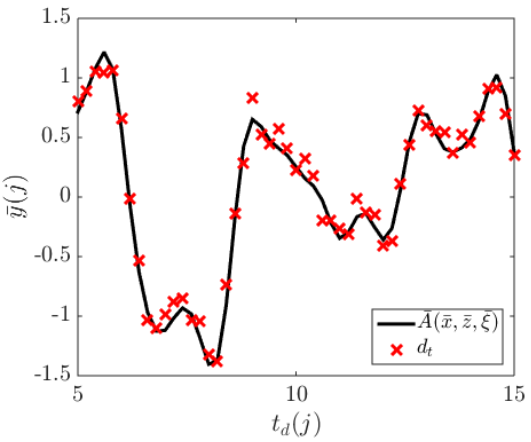


Figure 6.36: Model predictions and simulated data

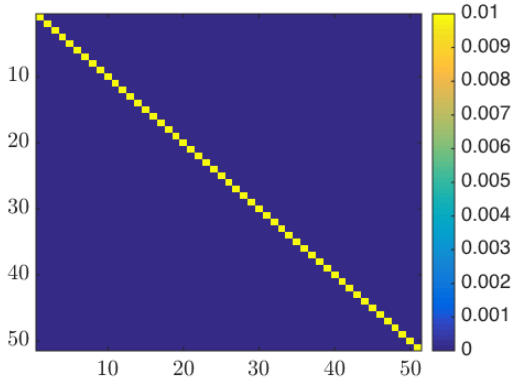


Figure 6.37: Noise covariance Γ_e

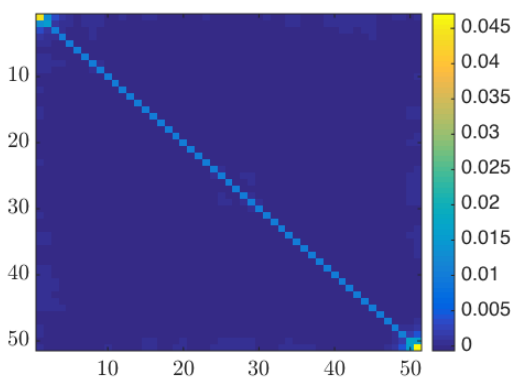


Figure 6.38: Conditional combined error covariance $\Gamma_{\nu|x}$

approximate model are shown in Figure 6.40. These were computed as

$$x_{\text{MAP}} = \begin{pmatrix} \tilde{L}_{\nu|x} A_c \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_{\nu|x} d \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.217}$$

and $\hat{\sigma}^2$ is the diagonal of

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A_c^T \Gamma_{\nu|x}^{-1} A_c)^{-1} \tag{6.218}$$

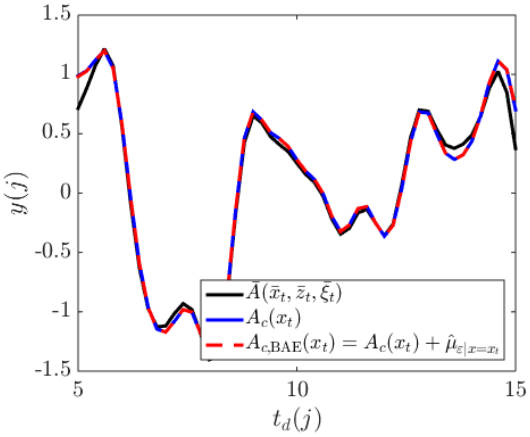


Figure 6.39: Accurate, approximate, and BAE corrected approximate forward model predictions

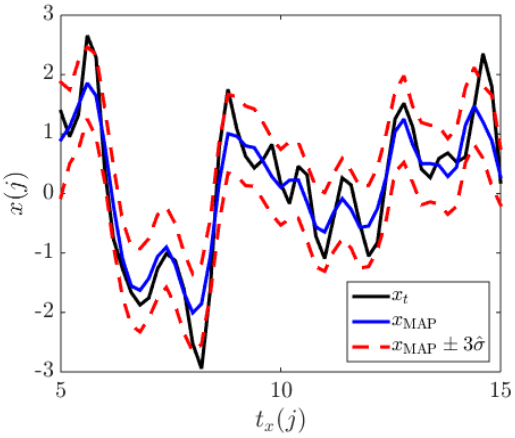


Figure 6.40: Estimate with posterior error intervals using BAE and local sample approximate model

as in Section 6.3.3 but using local sample approximation A_c rather than A.

The sample conditional mean $\hat{\mu}_{x|d=d_t}$ can also be used as an estimate of x_t given data d_t. The sample conditional mean can be expressed as a single pass local sample approximation of the mapping from d to x. Samples are computed as

$$d_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) + e_j \tag{6.219}$$

where $\bar{x}_j$, $\bar{z}_j$, $\bar{\xi}_j$ and e_j are draws. The estimate is

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}(d_t - \hat{\mu}_d) \tag{6.220}$$

$$= \hat{\mu}_x + Q_x R_x R_d^T (R_d R_d^T)^{-1} Q_d (d_t - \hat{\mu}_d) \tag{6.221}$$

making use of the QR factorisation as described in Section 6.4.1. Note that the only computationally costly aspect of constructing this estimate is computing the samples. The QR form is compressed, giving computationally efficient estimates at the online phase. The posterior covariance is

$$\Gamma_{x|d} = (\Gamma_x^{-1} + \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{dx})^{-1} \tag{6.222}$$

$$\approx \hat{\Gamma}_x - \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{dx} \tag{6.223}$$

$$= Q_x(R_x R_x^T - R_x R_d^T (R_d R_d^T)^{-1} R_d R_x^T) Q_x^T \tag{6.224}$$

where the QR form is compressed and can be available immediately as samples arrive by using recursive QR as described in section 6.4.2. The QR form can be used to form $\hat{\sigma}$ "element

by element", to avoid explicitly forming $\Gamma_{x|d} \in \mathbb{R}^{n_x \times n_x}$, which is potentially too large to fit explicitly in memory.

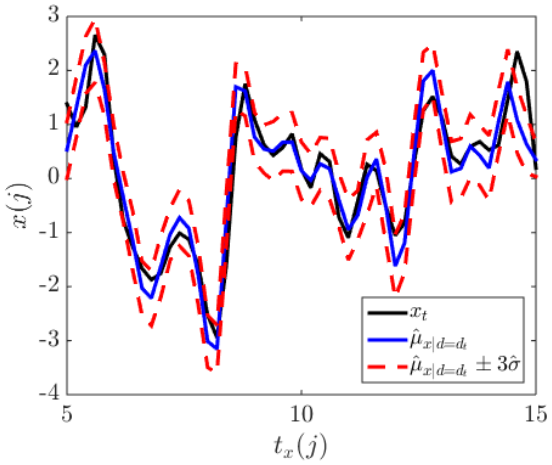


Figure 6.41: "Direct" single pass construction of $A_c^\dagger$ reconstructing x.

I apply the single pass local sample approximation approach as described above to the 1D deconvolution problem of Section 2.3.2, using the "fine" and "coarse" approximations of Section 6.3.3. The results are shown in Figure 6.41.

6.8 BAE and Sampling Compression for Other Regularisation Schemes

There may be a "natural" basis for unknowns x e.g. pixels in an image. It is natural to impose a prior e.g. smoothness in this basis based on correlation in that basis. However, there may be a more computationally efficient basis for a given problem.

This section considers the 1D deconvolution problem of Section 2.3.2, but expressed in the Fourier basis. I show that that a Gaussian smoothness prior, BAE and local sample approximations can still be applied.

Recall that the deconvolution problem can be stated as

$$d = A(x, z, \xi) + e = M\left(\int_{\infty}^{\infty} \xi(t-\tau)s(\tau)d\tau\right) + e \tag{6.225}$$

$$= M(B(s(t))) + e = M(c(t)) + e \tag{6.226}$$

where M is the mapping to measurement points, ξ is the kernel, $s(t) = \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}$ is the unconvolved signal, B is the convolution operator, and $c(t)$ is the convolved signal. The convolution

theorem [80, 81] states that the convolution operator can be expressed as

$$B(s(t)) = \int_{\infty}^{\infty} \xi(t-\tau)s(\tau)d\tau \tag{6.227}$$

$$= \mathcal{F}^{-1}\Big(\mathcal{F}(\xi(t)) \odot \mathcal{F}(s(t))\Big) \tag{6.228}$$

$$= \mathcal{F}^{-1}\Big(g(t) \odot \mathcal{F}(s(t))\Big) \tag{6.229}$$

$$= C(s(t)) \tag{6.230}$$

where $\mathcal{F}(s(t))$ is the Fourier transform of $s(t)$, $g(t)$ is the Fourier transform of $\xi(t)$ and $\odot$ denotes elementwise multiplication. We refer to B as the integral form of the convolution operator, and C as the Fourier form. An advantage of the Fourier form is that evaluating (discrete) Fourier transforms, and their inverse, can be performed rapidly. More specifically the fast Fourier transform (FFT) algorithm allows the discrete Fourier transform of a signal in $\mathbb{R}^n$ can be computed in $\mathcal{O}\Big(n\log(n)\Big)$ flops, as can the inverse discrete Fourier transform [32, 80, 81]. Note that the Fourier form assumes circularity, while the integral form assumes a zero signal outside Ω.

A means of solving the deconvolution problem using Fourier transforms is to model the system as

$$d = e + \mathcal{F}^{-1}\Big(\mathcal{F}(\xi) \odot \mathcal{F}(x)\Big) \tag{6.231}$$

noting that ξ and x are on the same discretisation as d i.e. $n_x = n_d$, hence the lack of measurement mapping M. The *Fourier inverse* estimate x_f^{-1} of x_t given data d_t is

$$x_f^{-1} = \mathcal{F}^{-1}\Big(\mathcal{F}(d_t) \oslash \mathcal{F}(\xi)\Big) \tag{6.232}$$

where $\oslash$ denotes elementwise division. Equivalently,

$$x_f^{-1} = \mathcal{F}^{-1}\Big(\mathcal{F}(d_t) \odot \tilde{g}_{\xi,0}\Big) \tag{6.233}$$

where $\tilde{g}_{\xi,0}(j) = \frac{1}{\big(\mathcal{F}(\xi)\big)(j)}$. In the case that $\big(\mathcal{F}(\xi)\big)(j) = 0$ for some j, the *Fourier pseudoinverse* estimate $x_f^{\dagger}$ of x_t is computed as

$$x_f^{\dagger} = \mathcal{F}^{-1}\Big(\mathcal{F}(d_t) \odot \tilde{g}_{\xi}\Big) \tag{6.234}$$

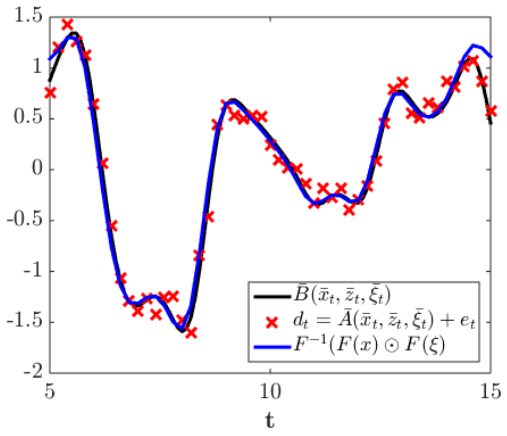


Figure 6.42: Model predictions and simulated data

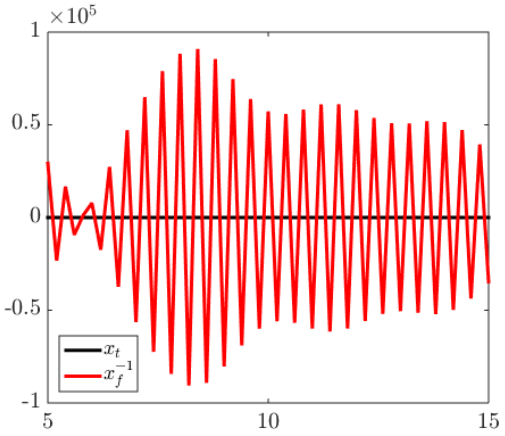


Figure 6.43: Unregularised Fourier solution

where $\tilde{g}_\xi(j) = \frac{1}{\big(\mathcal{F}(\xi)\big)(j)}$ for j such that $\big(F_{n_d}(\xi)\big)(j) \neq 0$, and $\tilde{g}_\xi(j) = 0$ for j such that $\big(\mathcal{F}(\xi)\big)(j) = 0$.

Let $\bar{A}(\bar{x}, \bar{z}, \bar{\xi})$ be the "accurate" model of Section 6.3.3. Let

$$A(x) = \mathcal{F}^{-1}\Big(\mathcal{F}(\xi) \odot \mathcal{F}(x)\Big) = \mathcal{F}^{-1}\Big(g \odot \mathcal{F}(x)\Big) \tag{6.235}$$

be the approximate Fourier model. Note that the accurate model is integral form, has padding, is on a finer discretisation, and uses the exact kernel. The forward predictions of the accurate and approximate models are shown on Figure 6.42 alongside simulated data d_t. The ground truth $x_t \in \mathbb{R}^{n_x}$ with $n_x = n_d = 51$ is plotted alongside the Fourier pseudoinverse estimate $x_f^\dagger = \mathcal{F}^{-1}\Big(\tilde{g}_\xi \odot \mathcal{F}(d_t)\Big)$ in Figure 6.43. The Fourier pseudoinverse seems to perform about as badly as the regular pseudoinverse estimate.

Let $x_{f,g}$ be an estimate of x_t computed in the Fourier domain as

$$x_{f,g} = \mathcal{F}^{-1}\Big(\mathcal{F}(d_t) \odot g\Big) \tag{6.236}$$

where $g \in \mathbb{R}^{n_x}$ is the *filter*. Estimates $x_{f,g}$ can be computed in $\mathcal{O}(nlog(n))$ flops using fast Fourier transforms. Compare $x_{f,g}$ with the MAP estimate x_{MAP} with Gaussian x and e com-

puted with the integral model B as

$$x_{\text{MAP}} = \begin{pmatrix} \tilde{L}_e B \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_e(d_t - \mu_e) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{6.237}$$

$$= \hat{A}^{\dagger} \hat{d}_t. \tag{6.238}$$

Note that computing x_{MAP} requires $\mathcal{O}(n_x^2(n_d + n_x))$ flops at the offline phase to compute $\hat{A}^{\dagger}$ with standard algorithms, and $\mathcal{O}(n_x(n_d + n_x))$ flops at the online phase to estimate x_t from d_t. Estimates with the form of $x_{f,g}$ are comparatively computationally cheap.

A common construction of filters g is $g_f(j) = \frac{f(j)}{(F\xi)(j)}$ where f is the *filter function*. A common example is the "box" filter, shown in Figure 6.44. We refer to this filter function f as $f = b$. The effect of this filter is to remove high frequencies from the solution. The motivation for the box filter is that the ground truth x_t has relatively large low frequency components relative to e_t. Applying the box filter is analogous to truncated series expansion, and to the MAP solution where we implement an infinite prior penalty on high frequency terms. The solution found with the box filter is shown in 6.45. Other filter functions can be used e.g. a "triangle" (or Bartlett window) rather than a box, but the idea remains the same [56, 79, 81].

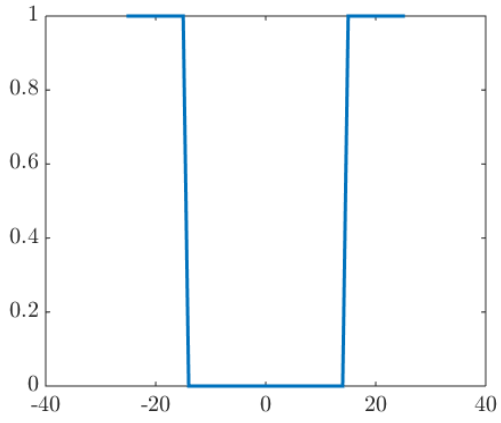


Figure 6.44: Box filter function b

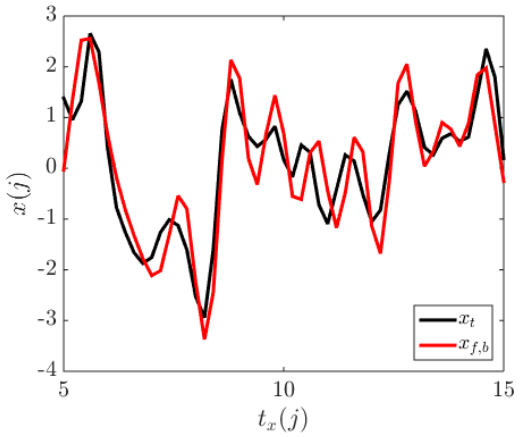


Figure 6.45: x_t and box filter Fourier estimate $x_{f,b}$

Recall that the standard Tikhonov solution for a problem modelled as

$$d = Ax + e = UDV^T x + e \tag{6.239}$$

can be found as

$$x_{\text{tik}} = V D_{\text{tik}}^{\dagger} U^T d \tag{6.240}$$

where $D_{\text{tik}}(j,j)^\dagger = \frac{d_j}{d_j^2+\alpha^2}$ where the penalty parameter α can be estimated by e.g. Morozov discrepancy principle or L-Curve criterion. The Fourier inverse estimate can be stated as

$$x_f = FG^\dagger F^* \tag{6.241}$$

where G is a diagonal matrix with $G(j,j) = \big(\mathcal{F}(\xi)\big)(j)$. We can take inspiration from Tikhonov regularisation and form diagonal $G^\dagger_{\text{tik}}(j) = \frac{G(j,j)}{G(j,j)^2+\alpha^2}$. This leads to the estimate

$$x_{f,\text{tik}} = FG^\dagger_{\text{tik}} F^* d_t \tag{6.242}$$

of x_t which can be computed in $\mathcal{O}\big(n\log(n)\big)$ flops. An example solution is shown in Figure 6.46. This is an example of incorporating a prior into a solution in an unfamiliar basis, in this case the standard Tikhonov white noise prior, but in Fourier space. The Fourier Tikhonov estimate $x_{f,\text{tik}}$ is closely related to the *Wiener Filter*. More information can be found in [56, 79, 117].

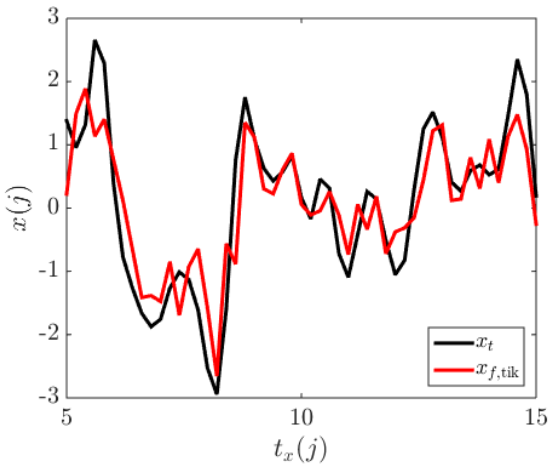


Figure 6.46: Fourier based, Tikhonov inspired estimate

I propose a method of constructing a filter that incorporates BAE and local sample compression. Let the accurate model be $d = \bar{A}(\bar{x},\bar{z},\bar{\xi}) + e$ as in Section 6.3.3. The aim is to construct a filter g such that estimates

$$x_{f,g} = \mathcal{F}^{-1}\Big(g \odot \mathcal{F}(d_t)\Big) \tag{6.243}$$

are close to the ground truth x_t. I compute draws $d_j = \bar{A}(\bar{x}_j,\bar{z}_j,\bar{\xi}_j) + e_j$ and $x_j = P_{x,\bar{x}}\bar{x}_j$. I

also compute "draws" g_j as

$$g_j(k) = \begin{cases} 0 & \text{if } \big(\mathcal{F}(d_j)\big)(k) = 0 \\ \frac{\big(\mathcal{F}(x_j)\big)(k)}{\big(\mathcal{F}(d_j)\big)(k)} & \text{otherwise} \end{cases} \tag{6.244}$$

i.e. $g_j = \min_g \left\{ x_j - \mathcal{F}^{-1}\big(g \odot \mathcal{F}(d_j)\big) \right\}$. I then compute the *sample average filter*

$$g_\pi = \frac{1}{m} \sum_{j=1}^{m} g_j \tag{6.245}$$

of m samples. Note that samples are computed with the "accurate" model as in BAE, and the filter is computed to be "accurate" for draws from the prior as in local sample compression.

I also approximate the combined error $\nu = e + \varepsilon$ from the above samples for the sake of uncertainty quantification. I approximate ε, and therefore ν, as independent of x for the sake of computational efficiency. Samples of the combined error ν_j are found as

$$\nu_j = x_j - \mathcal{F}^{-1}\big(g_\pi \odot \mathcal{F}(d_j)\big) \tag{6.246}$$

noting that the m samples used in constructing g_π need to already be computed to evaluate ν_j. The samples d_j can be reused to reduce computational cost, although this would be something of an inverse crime given that these samples are used in the construction of g_π. I form the sample covariance $\hat{\Gamma}_\nu$ and take $\hat{\sigma}^2 = \text{diag}(\hat{\Gamma}_\nu)$. The estimates x_{f,g_π} with $3\hat{\sigma}$ posterior error intervals found by the above method are shown in Figure 6.47. While the MAP estimate is overly smooth, the MAP with posterior error estimates do reasonably represent the ground truth.

6.9 Rapid Uncertainty Quantification

Recall that the posterior covariance of a linear problem with Gaussian prior and additive error with BAE is given as

$$\Gamma_{x|d} = (\Gamma_x^{-1} + A^T \Gamma_{\nu|x}^{-1} A)^{-1} \tag{6.247}$$

where $\Gamma_{\nu|x} = \Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} \hat{\Gamma}_{x\varepsilon}$. We may want to form the trust region as in [67], use an approximate posterior error interval from the diagonal, or quantify the "size" of the covariance with some matrix norm e.g. matrix 2 norm or Frobenius norm.

Recall that the conditional covariance can also be written as

$$\Gamma_{x|d} = \Gamma_x - \Gamma_{xd} \Gamma_d^{-1} \Gamma_{dx} \tag{6.248}$$

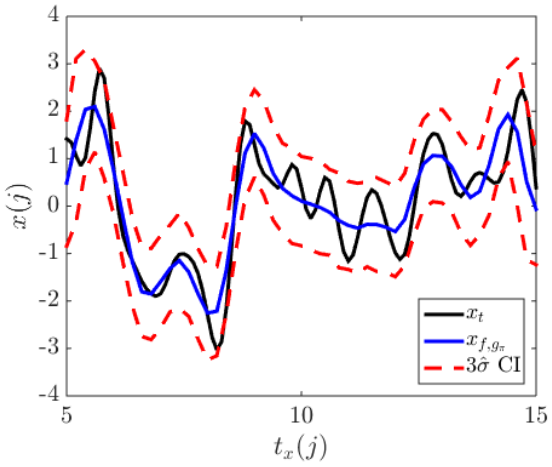


Figure 6.47: Fourier based solution x_{f,g_π} incorporating BAE and local sample approximation

which we can approximate by the sample covariance

$$\Gamma_{x|d} \approx \hat{\Gamma}_{x|d} = \hat{\Gamma}_x - \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{dx} \tag{6.249}$$

i.e. an approximate posterior covariance can be constructed just from samples of x and d. This approximation can also be made for nonlinear models with non-Gaussian unknowns, however such an approximation may not be representative of the actual uncertainty in the estimates.

As discussed in Section 6.4.2, QR factors of the sample matrices $X = (x_1, x_2, \ldots, x_m) = Q_x R_x$ and $D = (d_1, d_2, \ldots, d_m) = Q_d R_d$ can be formed while samples x_j and d_j are being computed. Note that a tolerance is used when forming $Q_x \in \mathbb{R}^{n_x \times r_x}$ and $Q_d \in \mathbb{R}^{n_d \times r_d}$ where r_x and r_d are effective ranks. The sample posterior covariance can be expressed as

$$\begin{aligned}
m\hat{\Gamma}_{x|d} &= m\left(\hat{\Gamma}_x - \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{dx}\right) && (6.250)\\
&= Q_x R_x R_x^T Q_x^T + Q_x R_x R_d^T Q_d^T (Q_d R_d R_d^T Q_d^T)^{-1} Q_d R_d R_x^T Q_x^T && (6.251)\\
&= Q_x R_x R_x^T Q_x^T + Q_x R_x R_d^T (R_d R_d^T)^{-1} R_d R_x^T Q_x^T && (6.252)\\
&= Q_x \left(R_x R_x^T + R_x R_d^T (R_d R_d^T)^{-1} R_d R_x^T \right) Q_x^T && (6.253)\\
&= Q_x M_{x|d} Q_x^T. && (6.254)
\end{aligned}$$

Note that only symmetric $(R_d R_d^T) \in \mathbb{R}^{r_d \times r_d}$ needs to be inverted. The inverted matrix $(R_d R_d^T)^{-1}$ can be computed iteratively as samples arrive by low rank updates, as described in Section 6.4.2. Note also that only $Q_x \in \mathbb{R}^{n_x \times r_x}$, $Q_d \in \mathbb{R}^{n_d \times r_d}$ and symmetric $M_{x|d} \in \mathbb{R}^{r_x \times r_x}$

need to be stored in memory for the above representation of the posterior covariance. The posterior error intervals in Figure 6.41 were taken from the diagonal of a posterior covariance formed as in Equation (6.254).

The posterior covariance can be computed with fewer sample approximations with a minimal increase in computational cost. Suppose we have a problem modelled as $d = Ax + \hat{\mu}_{\varepsilon|x} + e$ with Gaussian x and e, and sample estimate $\hat{\mu}_{\varepsilon|x}$ as in BAE. The posterior covariance can be expressed as

$$\begin{aligned}
\Gamma_{x|d} &= \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} && (6.255)\\
&= \Gamma_x - \Gamma_x A^T (A\Gamma_x A^T + \Gamma_{\nu|x})^{-1} A\Gamma_x && (6.256)\\
&= \Gamma_x - \Gamma_x A^T (A\Gamma_x A^T + \Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon})^{-1} A\Gamma_x && (6.257)
\end{aligned}$$

as described in Section 6.3. As described in Section 6.4.1, the conditional covariance $\hat{\Gamma}_{\varepsilon|x}$ can be expressed as

$$\begin{aligned}
\hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon} &= m\left(Q_\varepsilon R_\varepsilon R_\varepsilon^T Q_\varepsilon^T + Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} R_x R_\varepsilon^T Q_\varepsilon^T\right) && (6.258)\\
&= mQ_\varepsilon (R_\varepsilon R_\varepsilon^T + R_\varepsilon R_x^T (R_x R_x^T)^{-1} R_x R_\varepsilon^T) Q_\varepsilon^T && (6.259)\\
&= Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T && (6.260)
\end{aligned}$$

using the sample QR decompositions. Recall that samples of ε are computed as $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - Ax_j$. This involves computing samples $y_j = Ax_j$. The recursive QR factorisation of the samples y_j can be formed similarly. This allows for the approximation

$$A\Gamma_x A^T = \Gamma_y \approx \hat{\Gamma}_y = Q_y R_y R_y^T Q_y^T \tag{6.261}$$

therefore

$$\begin{aligned}
(A\Gamma_x A^T + \Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon})^{-1} &\approx \left(\Gamma_e + Q_y R_y R_y^T Q_y^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T\right)^{-1} && (6.262)\\
&= \left(\Gamma_e + \begin{pmatrix} Q_y & Q_\varepsilon \end{pmatrix} \begin{pmatrix} R_y R_y^T & 0_{r_y, r_\varepsilon} \\ 0_{r_\varepsilon, r_y} & M_{\varepsilon|x} \end{pmatrix} \begin{pmatrix} Q_y^T \\ Q_\varepsilon^T \end{pmatrix}\right)^{-1} && (6.263)\\
&= \left(\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T\right)^{-1} && (6.264)\\
&= \Gamma_e^{-1} - \Gamma_e^{-1} Q_{y,\varepsilon}\left(M_{y,\varepsilon|x}^{-1} + Q_{y,\varepsilon}^T \Gamma_e^{-1} Q_{y,\varepsilon}\right)^{-1} Q_{y,\varepsilon}^T \Gamma_e^{-1} && (6.265)
\end{aligned}$$

where

$$M^{-1}_{y,\varepsilon|x} = \begin{pmatrix} (R_y R_y^T)^{-1} & 0_{r_y,r_\varepsilon} \\ 0_{r_\varepsilon,r_y} & M^{-1}_{\varepsilon|x} \end{pmatrix} \tag{6.266}$$

can be computed recursively as samples arrive, as described in Section 6.4.2. The posterior covariance can be approximated as

$$\begin{aligned}
\Gamma_{x|d} &= \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} && (6.267)\\
&= \Gamma_x - \Gamma_x A^T (A\Gamma_x A^T + \Gamma_e + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon})^{-1} A\Gamma_x && (6.268)\\
&\approx \Gamma_x - \Gamma_x A^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{y,\varepsilon}\big(M^{-1}_{y,\varepsilon|x} + Q^T_{y,\varepsilon}\Gamma_e^{-1}Q_{y,\varepsilon}\big)^{-1} Q^T_{y,\varepsilon}\Gamma_e^{-1}\right) A\Gamma_x && (6.269)\\
&\approx \Gamma_x - Q_x R_x R_y^T Q_y^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{y,\varepsilon}\big(M^{-1}_{y,\varepsilon|x} + Q^T_{y,\varepsilon}\Gamma_e^{-1}Q_{y,\varepsilon}\big)^{-1} Q^T_{y,\varepsilon}\Gamma_e^{-1}\right) Q_y R_y R_x^T Q_x^T && (6.270)\\
&\approx Q_x \left(R_x R_x^T - R_x R_y^T Q_y^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{y,\varepsilon}\big(M^{-1}_{y,\varepsilon|x} + Q^T_{y,\varepsilon}\Gamma_e^{-1}Q_{y,\varepsilon}\big)^{-1} Q^T_{y,\varepsilon}\Gamma_e^{-1}\right) Q_y R_y R_x^T \right) Q_x^T && (6.271)
\end{aligned}$$

where each approximation in the above equations incorporates more sample approximations. Note that the noise covariance is not approximate, and the quantities computed from samples are "noiseless". Note the only inversions that need to be computed in the above equations are of

$$\begin{aligned}
\Gamma_e &\in \mathbb{R}^{n_d \times n_d} \\
R_y R_y^T &\in \mathbb{R}^{r_y \times r_y} \\
M_{\varepsilon|x} &\in \mathbb{R}^{r_\varepsilon \times r_\varepsilon} \text{ and} \\
\big(M^{-1}_{y,\varepsilon|x} + Q^T_{y,\varepsilon}\Gamma_e^{-1}Q_{y,\varepsilon}\big) &\in \mathbb{R}^{(r_y+r_\varepsilon)\times(r_y+r_\varepsilon)}
\end{aligned}$$

which potentially represents a large reduction in computational cost, particularly when Γ_e is simple to invert e.g. diagonal. The above construction can be computed recursively as samples arrive as described in Section 6.4.2, although the algebra is tedious. In my experience, computing $\big(M^{-1}_{y,\varepsilon|x} + Q^T_{y,\varepsilon}\Gamma_e^{-1}Q_{y,\varepsilon}\big)^{-1}$ recursively and $\Gamma_{x|d}$ once sampling concludes is sufficient.

The above formulation can also be used when A is nonlinear. The crudest approach is to compute a posterior covariance exactly as in Equations 6.267 - 6.271 i.e. quantify the uncertainty in estimates of x_t as equal regardless of x_t. A more representative approach would be to adapt the Laplace approximation, which approximates the model A as linear around the estimate x_{MAP} of x_t. Recall the local sample approximation as an approximation of the Jacobian

as discussed in Section 6.6.2. This effectively amounts to approximating J_{MAP} with a sample approximation, where samples are $y_j = A(x_{\text{MAP}} + \Delta_{x,j})$ where $\Delta_{x,j}$ is a small perturbation. If samples y_j are computed in this way, the "Laplace type" posterior covariance is as in Equations 6.267 - 6.271.

Samples can also be used to more directly estimate the error in estimates. Compute an additional $\hat{m}$ samples $d_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) + e_j$ and $x_j = P_{x,\bar{x}}\bar{x}_j$. Using the estimation scheme constructed for the online stage, compute estimates $\hat{x}_j$ of x_j from d_j, and the residual $\zeta_j = \hat{x}_j - x_j$. Note that if BAE is incorporated correctly, $\mu_\zeta \approx 0_{n_x,1}$. The posterior covariance can be approximated as

$$\Gamma_{x|d} \approx \hat{\Gamma}_\zeta \tag{6.272}$$

or

$$\hat{\sigma}_2 \approx \frac{1}{\hat{m}-1}\sum_{j=1}^{\hat{m}}(\zeta_j - \hat{\mu}_\zeta) \odot (\zeta_j - \hat{\mu}_\zeta) \tag{6.273}$$

implicitly making a diagonal approximation. In this thesis both estimating the posterior covariance or posterior variance from MAP estimate residuals is referred to as the *posterior residual approach*. More detail can be found in e.g. [118, 119]. This approach is used in Chapters 8 and 9, and was found to be effective when sample entities such as those used in constructing the sample conditional mean converge slowly.

Presented in [120] are methods of computing $v^T f(G)u$ efficiently, where v and u are vectors, G is a matrix, and f is some function of this matrix. Of particular interest is the computation of $e_j^T G^{-1} e_j$ i.e. the j'th diagonal element of G^{-1} This can be used in our case to compute

$$\hat{\sigma}^2(j) = \Gamma_{x|d}(j,j) = e_j^T(\Gamma_x^{-1} + A^T\Gamma_{\nu|x}A)^{-1}e_j. \tag{6.274}$$

This could be used to find all diagonal elements, or just a few. This is particularly useful when uncertainty estimates are only wanted in a particular region. I only mention the existence of these methods in this thesis. An alternative approach to estimating just a few diagonal elements of $\Gamma_{x|d}$ incorporating Landweber iterations is presented in Section 8.4.

Being able to quantify uncertainty quickly is particularly useful for Design of Experiments [121]. Consider a set of forward models A_{E_j} corresponding to different measurement setups E_j. Choosing which experiment E_j that gives the most accurate estimates of x_t can be thought of as choosing E_j such that $\Gamma_{x|d}$ is "smallest". This may require computing multiple posterior covariances per E_j in order to make a comparison. For such problems, the ability to compute posterior covariances relatively quickly with low rank approximations as in Equations 6.267 - 6.271 can significantly reduce computational cost at the offline phase.

Chapter 7

Combining and Implementing Methods for Inverse Problems

This chapter outlines a methodology for approaching inverse problems that makes use of the methods of Chapter 6.

Let

$$d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e \tag{7.1}$$
$$= \bar{A}(\bar{\omega}) + e \tag{7.2}$$

where d is measured data, $\bar{A}$ is an accurate model, $\bar{x}$ is an approximation of the unknown of interest, $\bar{z}$ is an auxiliary unknown (typically the region around $\bar{x}$), $\bar{\xi}$ is an additional parameter in the model and $\bar{\omega}$ is the combination of $\bar{x}$, $\bar{z}$ and $\bar{\xi}$. Let a particular realisation be

$$d_t = \bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t) + e_t \tag{7.3}$$
$$= \bar{A}(\bar{\omega}_t) + e_t \tag{7.4}$$

and we wish to estimate $x_t = P_{x,\bar{x}}\bar{x}_t$, the "sufficient resolution" ground truth. This thesis considers the case that the model of Equation (7.1) is sufficiently accurate but computationally expensive to work with. A typical case would be that the MAP estimate

$$\bar{\omega}_{\mathrm{MAP}} = \min_{\bar{\omega}} \left\{ \left\| \tilde{L}_e(\bar{A}(\bar{\omega}) - d_t) \right\|_2^2 + \bar{G}(\bar{\omega}) \right\} \tag{7.5}$$

cannot be computed by standard methods with the computational resources available at the online stage.

7.1 Reducing Computational Complexity with Sample Approximations

Suppose $\bar{A}$ is a linear model i.e. $d = \bar{A}\begin{pmatrix}\bar{x}\\ \bar{z}\\ \bar{\xi}\end{pmatrix} + e = \bar{A}\bar{\omega} + e$. Let $\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$, $\bar{z} \in \mathbb{R}^{n_{\bar{z}}}$, $\bar{\xi} \in \mathbb{R}^{n_{\bar{\xi}}}$ and $\bar{\omega} \in \mathbb{R}^{n_{\bar{\omega}}}$ where $n_{\bar{\omega}} = n_{\bar{x}} + n_{\bar{z}} + n_{\bar{\xi}}$. One way of estimating $\bar{\omega}_t$ from data $d_t = \bar{A}\bar{\omega}_t + e_t$ would be truncated series expansion as in Section 3.2. Let $\bar{A} = U_{\bar{A},r} D_{\bar{A},r} V_{\bar{A},r}^T$ be the r term thin SVD of $\bar{A}$. The r term tSVD estimate of $\bar{\omega}_t$ is

$$\bar{\omega}_r^\dagger = V_{\bar{A}} D_{\bar{A}}^{-1} U_{\bar{A}}^T d_t. \tag{7.6}$$

This estimate costs $\mathcal{O}\big(r(n_d + n_{\bar{\omega}})\big)$ flops and requires $\mathcal{O}\big(r(n_d + n_{\bar{\omega}})\big)$ numbers in storage at the online phase. With the tSVD computed, a standard Tikhonov type estimate can be found by adjusting the filter factors as described in Section 3.4, although in a reduced basis as described in Section 6.2. While such estimates may be of satisfactory accuracy and computational cost at the online stage, the offline computational cost of computing the tSVD may be prohibitive.

The tSVD could be computed by global randomised compression as described in [37] and reviewed in Section 6.6. The randomised r term tSVD of $\bar{A}$ can be computed with $2(q+1)$ passes over $\bar{A}$. Suppose draws $\bar{\omega}_j$ of $\bar{\omega}$ are available/can be computed. In this case, the local sample compression can be computed as proposed in Section 6.6.1.

Consider the single pass $(q = 0)$ sample approximations of Section 6.6.1. While the tSVD estimate is of the form

$$\bar{\omega}_r^\dagger = V_{\bar{A}} D_{\bar{A}}^{-1} U_{\bar{A}}^T d_t \tag{7.7}$$

the sample conditional mean estimate can be computed from m samples as

$$\hat{\mu}_{\bar{\omega}|d=d_t} = \hat{\mu}_{\bar{\omega}} + \hat{\Gamma}_{\bar{\omega}d}\hat{\Gamma}_d^{-1}(d_t - \hat{\mu}_d) \tag{7.8}$$

$$\approx \hat{\mu}_{\bar{\omega}} + Q_{\bar{\omega}} R_{\bar{\omega}} R_d (R_d R_d^T)^{-1} Q_d (d_t - \hat{\mu}_d) \tag{7.9}$$

where $Q_{\bar{\omega}} \in \mathbb{R}^{n_{\bar{\omega}} \times r_{\bar{\omega}}}$, $Q_d \in \mathbb{R}^{n_d \times r_d}$ and $R_{\bar{\omega}} R_d (R_d R_d^T)^{-1} \in \mathbb{R}^{r_{\bar{\omega}} \times r_d}$ can be computed from samples $\bar{\omega}_j$ and d_j as the samples are computed. Note that such an estimate can be computed in $\mathcal{O}(r_d n_d + r_{\bar{\omega}} n_{\bar{\omega}} + r_{\bar{\omega}} r_d)$ flops and requires $\mathcal{O}(r_d n_d + r_{\bar{\omega}} n_{\bar{\omega}} + r_{\bar{\omega}} r_d)$ numbers in storage at

the online phase. Alternatively, noiseless samples $\bar{y}_j = \bar{A}\bar{\omega}_j$ can be used to compute

$$\hat{\mu}_{\bar{\omega}|d=d_t} = \hat{\mu}_{\bar{\omega}} + \hat{\Gamma}_{\bar{\omega}d}\hat{\Gamma}_d^{-1}(d_t - \hat{\mu}_d) \tag{7.10}$$

$$= \hat{\mu}_{\bar{\omega}} + \hat{\Gamma}_{\bar{\omega}d}(\Gamma_e + \hat{\Gamma}_{\bar{y}})^{-1}(d_t - \hat{\mu}_d) \tag{7.11}$$

$$= \hat{\mu}_{\bar{\omega}} + \frac{1}{m}Q_{\bar{\omega}}R_{\bar{\omega}}R_{\bar{y}}^T Q_{\bar{y}}^T(\Gamma_e + \frac{1}{m}Q_{\bar{y}}R_{\bar{y}}R_{\bar{y}}^T Q_{\bar{y}}^T)^{-1}(d_t - \hat{\mu}_d) \tag{7.12}$$

$$= \hat{\mu}_{\bar{\omega}} + Q_{\bar{\omega}}M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T(\Gamma_e + Q_{\bar{y}}M_{\bar{y}}Q_{\bar{y}}^T)^{-1}(d_t - \hat{\mu}_d) \tag{7.13}$$

$$= \hat{\mu}_{\bar{\omega}} + Q_{\bar{\omega}}M_{\bar{\omega}\bar{y}}Q_y^T\left(\Gamma_e^{-1} + \Gamma_e^{-1}Q_{\bar{y}}\big(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}}\big)^{-1}Q_y^T\Gamma_e^{-1}\right)(d_t - \hat{\mu}_d) \tag{7.14}$$

by application of the matrix inversion lemma. The above form is particularly useful when Γ_e^{-1} is simple to compute e.g. $\Gamma_e = \sigma_e^2 I_{n_d}$. In that case, estimates can be computed as

$$\hat{\mu}_{\bar{\omega}|d=d_t} = \hat{\mu}_{\bar{\omega}} + Q_{\bar{\omega}}M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T\left(\frac{1}{\sigma_e^2}I_{n_d} + \frac{1}{\sigma_e^4}Q_{\bar{y}}\big(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2}I_{r_{\bar{y}}}\big)^{-1}Q_{\bar{y}}^T\right)(d_t - \hat{\mu}_d) \tag{7.15}$$

$$= \hat{\mu}_{\bar{\omega}} + Q_{\bar{\omega}}M_{\bar{\omega}\bar{y}}\left(\frac{1}{\sigma_e^2}I_{r_{\bar{y}}} + \frac{1}{\sigma_e^4}\big(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2}I_{r_{\bar{y}}}\big)^{-1}\right)Q_{\bar{y}}^T(d_t - \hat{\mu}_d). \tag{7.16}$$

Note that such an estimate can be computed in $\mathcal{O}\Big(r_{\bar{y}}(r_{\bar{y}}+n_d+n_{\bar{\omega}})+r_{\bar{\omega}}n_{\bar{\omega}}\Big)$ flops and requires $\mathcal{O}\Big(r_{\bar{y}}(r_{\bar{y}} + n_d + n_{\bar{\omega}}) + r_{\bar{\omega}}n_{\bar{\omega}}\Big)$ numbers in storage at the online phase. Note that the above estimate can be constructed even for nonlinear $\bar{A}$ and non normally distributed unknowns.

The posterior covariance can also be approximated with samples, as described in Section 6.9. The posterior covariance can be constructed from samples of $\bar{\omega}_j$ and d_j as

$$m\hat{\Gamma}_{\bar{\omega}|d} = mm\left(\hat{\Gamma}_{\bar{\omega}} - \hat{\Gamma}_{\bar{\omega}d}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{d\bar{\omega}}\right) \tag{7.17}$$

$$= Q_{\bar{\omega}}R_{\bar{\omega}}R_{\bar{\omega}}^T Q_{\bar{\omega}}R_{\bar{\omega}}^T - Q_{\bar{\omega}}R_{\bar{\omega}}R_d^T(R_dR_d^T)^{-1}R_dR_{\bar{\omega}}Q_{\bar{\omega}}^T. \tag{7.18}$$

Alternatively, samples of $\bar{y}_j = \bar{A}\bar{\omega}_j$ and knowledge of $e \sim \mathcal{N}(\mu_e, \Gamma_e)$ and $\bar{\omega} \sim \mathcal{N}(\mu_{\bar{\omega}}, \Gamma_{\bar{\omega}})$

can be used to form

$$\Gamma_{\bar{\omega}|d} = \Gamma_{\bar{\omega}} - \Gamma_{\bar{\omega}d}\Gamma_d^{-1}\Gamma_{d\bar{\omega}} \tag{7.19}$$

$$= \Gamma_{\bar{\omega}} - \Gamma_{\bar{\omega}}\bar{A}^T(\Gamma_e + \bar{A}\Gamma_{\bar{\omega}}\bar{A}^T)^{-1}\bar{A}\Gamma_{\bar{\omega}} \tag{7.20}$$

$$\approx \Gamma_{\bar{\omega}} - \Gamma_{\bar{\omega}}\bar{A}^T(\Gamma_e + \hat{\Gamma}_{\bar{y}})^{-1}\bar{A}\Gamma_{\bar{\omega}} \tag{7.21}$$

$$= \Gamma_{\bar{\omega}} - \Gamma_{\bar{\omega}}\bar{A}^T(\Gamma_e + Q_{\bar{y}}M_{\bar{y}}Q_{\bar{y}})^{-1}\bar{A}\Gamma_{\bar{\omega}} \tag{7.22}$$

$$= \Gamma_{\bar{\omega}} - \Gamma_{\bar{\omega}}\bar{A}^T\left(\Gamma_e^{-1} - \Gamma_e^{-1}Q_{\bar{y}}(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}})^{-1}Q_{\bar{y}}^T\Gamma_e^{-1}\right)\bar{A}\Gamma_{\bar{\omega}} \tag{7.23}$$

$$\approx \Gamma_{\bar{\omega}} - Q_{\bar{\omega}}M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T\left(\Gamma_e^{-1} - \Gamma_e^{-1}Q_{\bar{y}}(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}})^{-1}Q_{\bar{y}}^T\Gamma_e^{-1}\right)Q_{\bar{y}}M_{\bar{\omega}\bar{y}}^TQ_{\bar{\omega}}^T \tag{7.24}$$

$$\approx Q_{\bar{\omega}}\left(M_{\bar{\omega}} - M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T\left(\Gamma_e^{-1} - \Gamma_e^{-1}Q_{\bar{y}}(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}})^{-1}Q_{\bar{y}}^T\Gamma_e^{-1}\right)Q_{\bar{y}}M_{\bar{\omega}\bar{y}}^T\right)Q_{\bar{\omega}}^T \tag{7.25}$$

where each approximation replaces additional quantities with sample approximations. Note that the final form only requires Γ_e be known, with all other quantities being derived from samples. In the case that $\Gamma_e = \sigma_e^2 I_{n_d}$,

$$\Gamma_{\bar{\omega}|d} \approx Q_{\bar{\omega}}\left(M_{\bar{\omega}} - M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T\left(\Gamma_e^{-1} - \Gamma_e^{-1}Q_{\bar{y}}(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}})^{-1}Q_{\bar{y}}^T\Gamma_e^{-1}\right)Q_{\bar{y}}M_{\bar{\omega}\bar{y}}^T\right)Q_{\bar{\omega}}^T \tag{7.26}$$

$$= Q_{\bar{\omega}}\left(M_{\bar{\omega}} - M_{\bar{\omega}\bar{y}}Q_{\bar{y}}^T\left(\frac{1}{\sigma_e^2}I_{n_d} - \frac{1}{\sigma_e^4}Q_{\bar{y}}(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2}I_{r_{\bar{y}}})^{-1}Q_{\bar{y}}^T\right)Q_{\bar{y}}M_{\bar{\omega}\bar{y}}^T\right)Q_{\bar{\omega}}^T \tag{7.27}$$

$$= Q_{\bar{\omega}}\left(M_{\bar{\omega}} - M_{\bar{\omega}\bar{y}}\left(\frac{1}{\sigma_e^2}I_{r_{\bar{y}}} - \frac{1}{\sigma_e^4}(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2}I_{r_{\bar{y}}})^{-1}\right)M_{\bar{\omega}\bar{y}}^T\right)Q_{\bar{\omega}}^T \tag{7.28}$$

which requires $\mathcal{O}(n_{\bar{\omega}}r_{\bar{\omega}} + r_{\bar{\omega}}^2 + r_{\bar{\omega}}r_{\bar{y}} + r_{\bar{y}}^2)$ numbers be stored in memory. Note that only symmetric $M_{\bar{y}} \in \mathbb{R}^{r_{\bar{y}}\times r_{\bar{y}}}$ and $(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2}I_{r_{\bar{y}}}) \in \mathbb{R}^{r_{\bar{y}}\times r_{\bar{y}}}$ need to be inverted.

In the case that $n_{\bar{\omega}}$ is prohibitively large, the above analysis can be repeated but substituting e.g. $\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$ or $x \in \mathbb{R}^{n_x}$. For example, x_t could be estimated as

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + Q_xM_{x\bar{y}}Q_y^T\left(\Gamma_e^{-1} + \Gamma_e^{-1}Q_{\bar{y}}\big(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T\Gamma_e^{-1}Q_{\bar{y}}\big)^{-1}Q_{\bar{y}}^T\Gamma_e^{-1}\right)(d_t - \hat{\mu}_d) \tag{7.29}$$

with approximate posterior covariance

$$\hat{\Gamma}_{x|d} = Q_x \left(M_x - M_{x\bar{y}} Q_{\bar{y}}^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{\bar{y}} (M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}})^{-1} Q_{\bar{y}}^T \Gamma_e^{-1} \right) Q_{\bar{y}} M_{x\bar{y}}^T \right) Q_x^T \tag{7.30}$$

where Q_x, R_x $M_x = \frac{1}{m} R_x R_x^T$, $M_{x\bar{y}} = \frac{1}{m} R_x R_{\bar{y}}^T$ are formed from samples $x_j = P_{x,\bar{\omega}} \bar{\omega}_j$. However, these estimates are likely less accurate than estimates constructed using samples of $\bar{\omega}$. Given that the quantities in this section can be computed in parallel with samples being generated as described in Section 6.4.2, there is minimal reduction in computational cost at the offline phase of constructing e.g. $\hat{\mu}_{x|d}$ rather than $\hat{\mu}_{\bar{\omega}|d}$. There is a slight reduction however in computational cost and storage requirements at the online phase when using $\hat{\mu}_{x|d}$ rather than $\hat{\mu}_{\bar{\omega}|d}$ due to the reduction in dimension from $n_{\bar{\omega}}$ to n_x.

The recommendation of this thesis is to attempt forming local sample estimates such as $\hat{\mu}_{x|d}$ in Equation (7.29) and $\hat{\Gamma}_{\bar{\omega}|d}$ as in Equation (7.30). The samples used to compute $\hat{\mu}_{x|d}$ and $\Gamma_{\bar{\omega}|d}$ are also used in BAE, so the additional offline cost of computing $\hat{\mu}_{x|d}$ and $\Gamma_{\bar{\omega}|d}$ is minimal.

Computing $\hat{\mu}_{x|d}$ requires $\mathcal{O}\left(r_{\bar{y}}(r_{\bar{y}} + n_d + n_x) + r_x n_x \right)$ flops and $\mathcal{O}\left(r_{\bar{y}}(r_{\bar{y}} + n_d + n_x) + r_x n_x \right)$ numbers in storage at the online phase. Providing posterior error intervals requires an additional $2n_x$ flops and n_x numbers stored in memory at the online stage. This is potentially a relatively low online computational cost, so worth investigating. Such an approach is attempted in Chapters 8, 9 and 10. In chapters 8 and 9, r_x was found to be too high to justify implementing the method over other approaches. For the nonlinear problem of Chapter 10, the estimates $\hat{\mu}_{x|d}$ as in Equation (7.29) and $\hat{\Gamma}_{x|d}$ as in Equation (7.30) performed well. The online computational time is reduced from 11.2 seconds with standard nonlinear estimation methods to 0.0004 seconds when computing estimates as above. The estimates $\hat{\mu}_{x|d}$ as in Equation (7.29) and $\hat{\Gamma}_{x|d}$ as in Equation (7.30) were also more representative of the ground truth than the nonlinear estimates.

7.2 Initial Sampling

The above section discusses how samples can be used efficiently to compute e.g. approximate covariances. This section considers the efficient computation of samples.

Let $\bar{s} = \begin{pmatrix} \bar{x} \\ \bar{z} \end{pmatrix} \sim \mathcal{N}(\Gamma_{\bar{s}}, \mu_{\bar{s}})$. Samples can be computed as $\bar{s}_j = \mu_{\bar{s}} + L_{\bar{s}} w_j$ where w_j is a draw of $w \sim \mathcal{N}(0_{n_{\bar{s}},1}, I_{n_{\bar{s}}})$ and $L_{\bar{s}} L_{\bar{s}}^T = \Gamma_{\bar{s}}$. Consistent prior covariances can be constructed with projection matrices e.g. $\Gamma_x = P_{x,\bar{s}} \Gamma_{\bar{s}} P_{x,\bar{s}}^T$.

Let $\bar{\Gamma}_{\bar{s}}$ be a covariance matrix constructed with a covariance function e.g. smoothness as in Section 5.1.1. The covariance matrix can be streamed through memory and the eigendecomposition constructed as a global sample approximation as described in [37] and reviewed in Section 6.6. The intermediate step of computing the QR factorisation can be accomplished with recursive QR as described in Section 6.4.1. In the case that the covariance is well understood e.g. isotropic unstructured smoothness prior on a rectangular domain, we may have the eigendecomposition of $\bar{\Gamma}_{\bar{s}}$ analytically [73, 93] as reviewed in Section 6.5. For all these cases, a filter of the form $L_{\bar{s}} = U_{\bar{s}}\Lambda_{\bar{s}}^{\frac{1}{2}}$ can be constructed.

If it is found that $\bar{\Gamma}_{\bar{s}}$ is low effective rank, a reduced basis or penalty subspace may be used, as discussed in Section 6.2. In this section, $\Gamma_{\bar{s}}$ refers to the reduced dimension or penalty subspace approximation to $\bar{\Gamma}_{\bar{s}}$.

Suppose the prior on $\bar{s}$ was formed by direct filter construction as described in Section 5.1.3. Let $\bar{L}_{\bar{s}}$ be the filter constructed directly. In the case that $\bar{L}_{\bar{s}}$ is difficult to work with e.g. a full matrix, a sample factorisation can also be formed. For example, samples $\bar{s}_j = \bar{L}_{\bar{s}} w_j$ could be computed, and recursive QR performed on such samples as in Section 6.4.2. This can be used to construct the low rank approximation

$$\bar{L}_{\bar{s}} \approx L_{\bar{s}} = Q_{\bar{s}} R_{\bar{s}} R_w (R_w R_w^T)^{-1} Q_w^T. \tag{7.31}$$

Subsequent analysis of the problem can be done in terms of

$$\Gamma_{\bar{s}} = L_{\bar{s}} L_{\bar{s}}^T = \frac{1}{m} Q_{\bar{s}} R_{\bar{s}} R_{\bar{s}}^T Q_{\bar{s}}^T, \tag{7.32}$$

a low rank approximation of the "directly constructed" prior. This approximation can be used in a reduced basis approach, or a penalty subspace could be introduced to form

$$\Gamma_{\bar{s}} = \frac{1}{m} Q_{\bar{s}} R_{\bar{s}} R_{\bar{s}}^T Q_{\bar{s}}^T + \kappa^2 (I - Q_{\bar{s}} Q_{\bar{s}}^T) \tag{7.33}$$

where $\kappa \in \mathbb{R}$ is some small positive number.

7.3 BAE and Local Compression

Suppose samples $\bar{x}_j$, $\bar{z}_j$, $\bar{\xi}_j$ and x_j can be generated. Let $d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e$. Let a particular realisation be $d_t = \bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t) + e_t$, and we wish to estimate $x_t = P_{x,\bar{x}} \bar{x}_t$ from d_t.

Estimates of x_t could be formed with a sample conditional mean type approach as in Section 7.1. In this section I consider incorporating the BAE model

$$\begin{aligned} d &= \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e && (7.34)\\ &= A(x) + \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) - A(x) + e && (7.35)\\ &= A(x) + \varepsilon + e && (7.36)\\ &\approx A(x) + \hat{\mu}_{\varepsilon|x} + e && (7.37) \end{aligned}$$

where $\hat{\mu}_{\varepsilon|x}$ is the sample conditional mean of ε. Samples are computed as $\varepsilon_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) - A(x_j)$. Compare this to the entirely sample based approach discussed in Section7.1, which approximates the forward model as

$$d \approx \hat{\mu}_{d|x} = \hat{\mu}_d + \hat{\Gamma}_{d,x}\hat{\Gamma}_x^{-1}(x - \hat{\mu_x}) \tag{7.38}$$

for a single pass.

I expect the BAE approach will produce better estimates than purely sample based approaches in the case that $\bar{A}$ and A are nonlinear. In the case that A is linear, BAE may still represent a significant improvement. It is possible that $\hat{\mu}_{\varepsilon|x}$ can be estimated effectively with a fewer number of samples than $\hat{\mu}_{d|x}$. For example, the case $\bar{A} = A$. Another possibility is that the BAE forward model $A(x) + \hat{\mu}_{\varepsilon|x}$ can be evaluated more easily than the local sample approximation $\hat{\mu}_{d|x}$. For example, the entity $Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x^T$ in the BAE forward model as derived in Section 6.4.1 may be low rank, while the entity $Q_d R_d R_x (R_x R_x^T)^{-1} Q_x^T$ in the local sample approximation $\hat{\mu}_{d|x}$ may be high rank.

I compute $\hat{\mu}_{\varepsilon|x}$ using recursive QR as

$$\hat{\mu}_{\varepsilon|x} = \hat{\mu}_\varepsilon + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}(x - \hat{\mu}_x) \tag{7.39}$$

$$= \hat{\mu}_\varepsilon + Q_\varepsilon R_\varepsilon R_x^T (R_x^T R_x^T)^{-1} Q_x^T (x - \hat{\mu}_x) \tag{7.40}$$

as described in Section 6.4.1.

Recall the BAE MAP estimate

$$x_{MAP} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d - A(x) - \hat{\mu}_{\nu|x}) \right\|_2^2 + G_x(x) \right\} \tag{7.41}$$

where $\tilde{L}_{\nu|x}^T \tilde{L}_{\nu|x} = (\hat{\Gamma}_{\nu|x})^{-1} = (\Gamma_e + \Gamma_{\varepsilon|x})^{-1}$ and G_x is the exponential prior filter on x e.g. $G = \text{TV}(x)$.

Consider $\Gamma_{\nu|x}$. This can be formed as

$$\Gamma_{\nu|x} = \Gamma_e + \hat{\Gamma}_{\varepsilon|x} \tag{7.42}$$

$$= \Gamma_e + \frac{1}{m} Q_\varepsilon (R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x (R_x R_x^T)^{-1} R_x R_\varepsilon) Q_\varepsilon^T \tag{7.43}$$

$$= \Gamma_e + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \tag{7.44}$$

and apply the matrix inversion lemma to find

$$\Gamma_{\nu|x}^{-1} = \left(\Gamma_e + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \right)^{-1} \tag{7.45}$$

$$= \Gamma_e^{-1} - \Gamma_e^{-1} Q_\varepsilon (M_{\varepsilon|x} - Q_\varepsilon \Gamma_e^{-1} Q_\varepsilon^T)^{-1} Q_\varepsilon^T \Gamma_e^{-1} \tag{7.46}$$

a result which can be reused when we form the posterior covariance $\Gamma_{x|d} \approx \Gamma_x - \Gamma_x J^T(\Gamma_{\nu|x} + J\Gamma_x J^T)^{-1} J\Gamma_x$. We can also compute $\tilde{L}_{\nu|x}$ efficiently as an update to $\tilde{L}_e$, as described in Section 6.4.1.

Consider the BAE posterior covariance. Let J_{MAP} be the linearisation of A at x_{MAP}. The Laplace approximation to the posterior covariance is

$$\Gamma_{x|d} = \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} \tag{7.47}$$

$$\approx \Gamma_x - \Gamma_x J_{\text{MAP}}^T(\Gamma_e + J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T + \hat{\Gamma}_\varepsilon - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\hat{\Gamma}_{x\varepsilon})^{-1} J_{\text{MAP}}\Gamma_x \tag{7.48}$$

$$= \Gamma_x - \Gamma_x J_{\text{MAP}}^T\left(\Gamma_e + J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T + \frac{1}{m}Q_\varepsilon(R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x(R_x R_x^T)^{-1}R_x R_\varepsilon^T)Q_\varepsilon^T\right)^{-1} J_{\text{MAP}}\Gamma_x \tag{7.49}$$

$$= \Gamma_x - \Gamma_x J_{\text{MAP}}^T\left(\Gamma_e + J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T\right)^{-1} J_{\text{MAP}}\Gamma_x. \tag{7.50}$$

There are advantages to forming a low rank approximation to Γ_x e.g. $\Gamma_x \approx L_x L_x^T$ with

$$L_x = (l_1, l_2, \ldots, l_{r_x}) \in \mathbb{R}^{n_x \times r_x} \tag{7.51}$$

as then

$$J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T \approx \left(J_{\text{MAP}}L_x\right)\left(J_{\text{MAP}}L_x\right)^T \tag{7.52}$$

with

$$\left(J_{\text{MAP}}L_x\right)(:, j) = J_{\text{MAP}}l_j \approx \frac{1}{\kappa}\left(A(x_{MAP} + \kappa l_j) - A(x_{MAP})\right) \tag{7.53}$$

where $\kappa \in \mathbb{R}$ is a small positive number. Note that this construction allows $J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T$ to be constructed with r_x additional evaluations of A rather than explicitly forming J_{MAP}. Similarly $J_{\text{MAP}}\Gamma_x = (J_{\text{MAP}}L_x)L_x^T$ can be formed. Alternatively, compute the approximation

$$J_{\text{MAP}}\Gamma_x \approx \hat{\Gamma}_{yx} = \frac{1}{m}Q_y R_y R_x^T Q_x^T \tag{7.54}$$

$$J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T \approx \hat{\Gamma}_y = \frac{1}{m}Q_y R_y R_y^T Q_y^T \tag{7.55}$$

where Q_y, Q_x, R_y and R_x are computed from samples x_j from the prior and $y_j = A(x_{\text{MAP}} + \kappa x_j)$. Note that such local sample approximations can be constructed with non normal unknowns and nonlinear A.

Let $BB^T \approx J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T$ with $B \in \mathbb{R}^{n_d \times r_y}$ be the low rank approximation found either by a ow rank approximation to Γ_x or sampling. Note that

$$\left(\Gamma_e + J_{\text{MAP}}\Gamma_x J_{\text{MAP}}^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T\right)^{-1} \approx \left(\Gamma_e + BB^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T\right)^{-1} \tag{7.56}$$

can be computed as a low rank update of Γ_e^{-1} by the matrix inversion lemma i.e.

$$\left(\Gamma_e + BB^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T\right)^{-1} = \left(\Gamma_e + \begin{pmatrix} B & Q_\varepsilon \end{pmatrix} \begin{pmatrix} I_{r_y} & 0_{r_y,r_\varepsilon} \\ 0_{r_\varepsilon,r_y} & M_{\varepsilon|x} \end{pmatrix} \begin{pmatrix} B^T \\ Q_\varepsilon^T \end{pmatrix}\right)^{-1} \tag{7.57}$$

$$= \Gamma_e^{-1} - \Gamma_e^{-1} \begin{pmatrix} B & Q_\varepsilon \end{pmatrix} \left(\begin{pmatrix} I_{r_y} & 0_{r_y,r_\varepsilon} \\ 0_{r_\varepsilon,r_y} & M_{\varepsilon|x}^{-1} \end{pmatrix} + \begin{pmatrix} B & Q_\varepsilon \end{pmatrix} \Gamma_e^{-1} \begin{pmatrix} B^T \\ Q_\varepsilon^T \end{pmatrix} \right)^{-1} \begin{pmatrix} B^T \\ Q_\varepsilon^T \end{pmatrix} \Gamma_e^{-1} \tag{7.58}$$

which can then be reincorporated into $\Gamma_{x|d}$.

7.4 Point estimates for Linear Gaussian Inverse Problems

Consider the linear A BAE forward model

$$d = Ax + \hat{\mu}_{\nu|x} + e \tag{7.59}$$

where $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ and $e \sim \mathcal{N}(0_{n_d,1}, \Gamma_e)$. The MAP estimate is

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d - Ax - \hat{\mu}_{\nu|x}) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{7.60}$$

$$= \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d - Ax - \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x(x - \mu_x) - \hat{\mu}_\varepsilon) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{7.61}$$

$$= \min_x \left\{ \left\| \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x) \\ \tilde{L}_x \end{pmatrix} x - \begin{pmatrix} \tilde{L}_{\nu|x}(d + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x\mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x\mu_x \end{pmatrix} \right\|_2^2 \right\} \tag{7.62}$$

$$= \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x\mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x\mu_x \end{pmatrix}. \tag{7.63}$$

Note that for this case, $x_{MAP} = \mu_{x|d}$. When using BAE, the conditional mean is

$$\mu_{x|d} = \mu_x + \Gamma_{xd}\Gamma_d^{-1}(d - \mu_d) \tag{7.64}$$

$$= \mu_x + \Gamma_x A^T \left(\Gamma_e + A\Gamma_x A^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \right)^{-1} (d - A\mu_x - \hat{\mu}_\varepsilon) \tag{7.65}$$

$$\approx \mu_x + \frac{1}{m} Q_x R_x R_y^T Q_y^T \left(\Gamma_e + \frac{1}{m} Q_y R_y R_y^T Q_y^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T \right)^{-1} (d - A\mu_x - \hat{\mu}_\varepsilon) \tag{7.66}$$

$$= \mu_x + Q_x M_{xy} Q_y^T \left(\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y\varepsilon}^T \right)^{-1} (d - A\mu_x - \hat{\mu}_\varepsilon) \tag{7.67}$$

$$= \mu_x + Q_x M_{xy} Q_y^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{y,\varepsilon} (M_{y,\varepsilon|x}^{-1} + Q_{y\varepsilon}^T \Gamma_e^{-1} Q_{y\varepsilon})^{-1} Q_{y\varepsilon}^T \Gamma_e^{-1} \right) (d - A\mu_x - \hat{\mu}_\varepsilon) \tag{7.68}$$

incorporating samples $y_j = Ax_j$ similar to Section 7.1. Note that

$$M_{y,\varepsilon|x}^{-1} = \begin{pmatrix} M_y & 0_{r_y, r_\varepsilon} \\ 0_{r_\varepsilon, r_y} & M_{\varepsilon|x} \end{pmatrix}^{-1} \tag{7.69}$$

$$= \begin{pmatrix} M_y^{-1} & 0_{r_y, r_\varepsilon} \\ 0_{r_\varepsilon, r_y} & M_{\varepsilon|x}^{-1} \end{pmatrix} \tag{7.70}$$

can be computed recursively as samples are computed, as described in Section 6.4.2.

The conditional mean formulation above makes use of sample approximations to Γ_x and $A\Gamma_x$. It could be that the sample approximations do not converge fast enough for the computational savings to be worthwhile. Note that the MAP estimate

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_{\nu|x}(d - Ax - \hat{\mu}_{\nu|x}) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{7.71}$$

$$= \begin{pmatrix} \tilde{L}_{\nu|x} \left(A + Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x \right) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d + Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x \mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{7.72}$$

$$= \hat{A}^\dagger \hat{d} \tag{7.73}$$

makes use of different sample quantities. It may be that the MAP estimate found for a certain number of samples is more representative of the ground truth than the conditional mean. This is the case for the problems of Chapters 8 and 9.

Note that x_{MAP} can be found as the solution to

$$\hat{A} x_{\text{MAP}} = \hat{d} \tag{7.74}$$

by an iterative solver e.g. Landweber iterations. This means that $\hat{A}^\dagger$ neither needs to be computed or stored. Computing $\hat{A}^\dagger$ is a potentially large computational cost at the offline phase. Using $\hat{A}^\dagger$ may also be more difficult at the online phase e.g. $\hat{A}$ might be sparse but $\hat{A}^\dagger$ is full.

The Landweber sequence x_k of approximations to x_{MAP} is

$$x_{k+1} = x_k + \beta(\hat{A}^T\hat{d} - \hat{A}^T\hat{A}x_k) \tag{7.75}$$

where $\beta \in \mathbb{R}$ is the step length. A typical initial guess would be $x_0 = \mu_x$. Given that the conditional mean estimate can be found with minimal additional computations by recursive QR, I propose taking $x_0 = \hat{\mu}_{x|d}$ computed as in Equation (7.29), potentially reducing the required number of iterates. The $\hat{A}^T\hat{d}$ term can be expanded as

$$\hat{A}^T\hat{d} = \left(\left(A^T + Q_x(R_xR_x^T)^{-1}R_xR_\varepsilon^TQ_\varepsilon^T\right)\tilde{L}_{\nu|x}^T \quad \tilde{L}_x^T\right)\begin{pmatrix}\tilde{L}_{\nu|x}(d + Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x\mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x\mu_x\end{pmatrix} \tag{7.76}$$

$$= \left(A^T + Q_x(R_xR_x^T)^{-1}R_xR_\varepsilon^TQ_\varepsilon^T\right)\Gamma_{\nu|x}^{-1}(d + Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x\mu_x - \hat{\mu}_\varepsilon) + \Gamma_x^{-1}\mu_x \tag{7.77}$$

and

$$\Gamma_{\nu|x}^{-1} = (\Gamma_e + Q_\varepsilon M_{\varepsilon|x}Q_\varepsilon^T)^{-1} \tag{7.78}$$

$$= \Gamma_e^{-1} - \Gamma_e^{-1}Q_\varepsilon(M_{\varepsilon|x}^{-1} + Q_\varepsilon^T\Gamma_e^{-1}Q_\varepsilon)^{-1}Q_\varepsilon^T\Gamma_e^{-1} \tag{7.79}$$

by the matrix inversion lemma. The $\hat{A}^T\hat{A}$ term can be expanded as

$$\hat{A}^T\hat{A} = \left(\left(A^T + Q_x(R_xR_x^T)^{-1}R_xR_\varepsilon^TQ_\varepsilon^T\right)\tilde{L}_{\nu|x}^T \quad \tilde{L}_x^T\right)\begin{pmatrix}\tilde{L}_{\nu|x}\left(A + Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x\right) \\ \tilde{L}_x\end{pmatrix} \tag{7.80}$$

$$= \left(A^T + Q_x(R_xR_x^T)^{-1}R_xR_\varepsilon^TQ_\varepsilon^T\right)\Gamma_{\nu|x}^{-1}\left(A + Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x\right) + \Gamma_x^{-1}. \tag{7.81}$$

Note that the innovation filters $\tilde{L}_{\nu|x}$ and $\tilde{L}_x$ do not appear in the Landweber iteration, instead only the unfactorised forms $\Gamma_{\nu|x}^{-1}$ and Γ_x^{-1} are used. This observation removes some computational cost at the offline phase. Care should be taken to compute $\hat{A}^T\hat{d}$ and $\hat{A}^T\hat{A}x$ from right to left (matrix-vector products only) rather than try and explicitly form the matrices. This is useful when products Av can be computed rapidly (e.g. when A is sparse).

7.5 Discussion

Various combinations of the methods of this thesis have been proposed in this chapter. Let the model be

$$d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e \tag{7.82}$$

and we wish to estimate x_t from d_t where $d_t = \bar{A}(\bar{x}_t, \bar{z}_t, \bar{\xi}_t) + e_t$. Let $e \sim \mathcal{N}(0_{n_d,1}, \Gamma_e)$.

I recommend first forming the sample estimate

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + Q_x M_{x\bar{y}} Q_y^T \left(\Gamma_e^{-1} + \Gamma_e^{-1} Q_{\bar{y}} \big(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}} \big)^{-1} Q_{\bar{y}}^T \Gamma_e^{-1} \right) (d_t - \hat{\mu}_d) \tag{7.83}$$

with approximate posterior covariance

$$\hat{\Gamma}_{x|d} = Q_x \left(M_x - M_{x\bar{y}} Q_{\bar{y}}^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{\bar{y}} (M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}})^{-1} Q_{\bar{y}}^T \Gamma_e^{-1} \right) Q_{\bar{y}} M_{x\bar{y}}^T \right) Q_x^T \tag{7.84}$$

making use of recursive QR as samples are generated. This method was attempted for the problems in Chapters 8, 9 and 10. The method was found to work well for the nonlinear problem of Chapter 10.

The above estimate may perform poorly when e.g. r_x or $r_{\bar{y}}$ is large, requiring many samples be computed at the offline stage and using large matrices at the online stage. This is the situation for the problems in Chapters 8 and 9. Instead, the MAP estimate with linear approximate model and BAE $d \approx Ax + \hat{\mu}_{\varepsilon|x} + e$ is found by Landweber iterations. That is, the sequence approximating the MAP is formed as

$$x_{k+1} = x_k + \beta(\hat{A}^T \hat{d} - \hat{A}^T \hat{A} x_k) \tag{7.85}$$

where

$$\hat{A}^T \hat{d} = \left(A^T + Q_x (R_x R_x^T)^{-1} R_x R_\varepsilon^T Q_\varepsilon^T \right) \Gamma_{\nu|x}^{-1} (d + Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x \mu_x - \hat{\mu}_\varepsilon) + \Gamma_x^{-1} \mu_x \tag{7.86}$$

and

$$\Gamma_{\nu|x}^{-1} = \Gamma_e^{-1} - \Gamma_e^{-1} Q_\varepsilon (M_{\varepsilon|x}^{-1} + Q_\varepsilon^T \Gamma_e^{-1} Q_\varepsilon)^{-1} Q_\varepsilon^T \Gamma_e^{-1} \tag{7.87}$$

and

$$\hat{A}^T \hat{A} = \left(A^T + Q_x (R_x R_x^T)^{-1} R_x R_\varepsilon^T Q_\varepsilon^T \right) \Gamma_{\nu|x}^{-1} \left(A + Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x \right) + \Gamma_x^{-1}. \tag{7.88}$$

The estimates found in this way were found to be of sufficient accuracy and computational cost at the online stage. The entities constructed from samples were found to be effectively low rank, and could be estimated to sufficient accuracy from a relatively small number of offline samples. Posterior error intervals can be computed by posterior residuals from an additional $\hat{m}$ samples. The samples $d_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) + e_j$ and $x_j = P_{x,\bar{x}}\bar{x}_j$ are used to compute estimates $x_{\mathrm{MAP,j}}$ of x_j from d_j, and the residual $\zeta_j = x_{\mathrm{MAP,j}} - x_j$. The posterior error interval is formed as

$$\hat{\sigma}_2 \approx \frac{1}{\hat{m}-1}\sum_{j=1}^{\hat{m}}(\zeta_j - \hat{\mu}_\zeta) \odot (\zeta_j - \hat{\mu}_\zeta) \tag{7.89}$$

implicitly making a diagonal approximation. This approach is used in Chapters 8 and 9, and was found to be effective when sample entities such as those used in constructing the sample conditional mean converge slowly.

Chapter 8

2D Deconvolution

This chapter will consider deconvolution in 2D. The purpose of this chapter is not to give an in depth study of 2D deconvolution, but rather to demonstrate the validity of approximations and effectiveness of methods proposed in Chapters 6 and 7.

Let the full domain be $\Omega = [1, 512] \times [1, 512]$ which we discretise into $\bar{t}_1 = (1 : 1 : 512) \in \mathbb{R}^{512}$ and $\bar{t}_2 = (1 : 1 : 512) \in \mathbb{R}^{512}$ i.e. a 512×512 pixel image. The region of interest is the upper right hand quadrant Ω_x with $\bar{t}_{\bar{x},1} = (1 : 1 : 256) \in \mathbb{R}^{256}$ and $\bar{t}_{\bar{x},1} = (257 : 1 : 512) \in \mathbb{R}^{256}$. Define $n_{\bar{s}} = 512$ and $N_{\bar{s}} = n_{\bar{s}}^2 = 512^2 = 262,144$. Similarly, $n_{\bar{x}} = 256$ and $N_{\bar{x}} = n_{\bar{x}}^2 = 256^2 = 65,536$. The rest of the pixels in the image are $\bar{z}$. All pixel values are stored in $\bar{s} \in \mathbb{R}^{262,144}$, pixel values in the region of interest are stored in $\bar{x} \in \mathbb{R}^{65,536}$, and pixel values outside the region of interest are store in $\bar{z} \in \mathbb{R}^{196,608}$.

The convolution kernel is

$$\mathbf{k}(\mathbf{t}, \bar{\xi}) = \frac{1}{2\pi\bar{\xi}^2} \exp\left(\frac{-\|t_0 - \mathbf{t}\|_2^2}{2\bar{\xi}^2}\right) \tag{8.1}$$

where $t_0 = (256.5, 256.5)^T$ is the centre of the image and $\bar{\xi} \in \mathbb{R}^1$ controls the width of this Gaussian kernel. Let $\bar{\xi} \sim \mathcal{U}(2, 3)$.

The data is in Ω_x but downsampled from 256×256 pixels to 128×128 pixels i.e. $n_d = 128$, $N_d = 16,384$. This discretisation is the same as for $x \in \mathbb{R}^{16,384}$, the region of interest at an acceptable resolution i.e. $n_x = 128$ and $N_x = 16,384$.

I define the "accurate" model

$$d = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) + e \tag{8.2}$$

and the "simple" model

$$d \approx A(x) + e. \tag{8.3}$$

Note that the simple model is at lower resolution, ignores three quarters of the image, and uses a fixed kernel width $\mu_\xi = 2.5$. Both models compute the convolution with fast Fourier transforms. The models can be expanded as

$$\bar{c} = \bar{F}(\bar{x}, \bar{z}, \bar{\xi}) = \mathcal{F}^{-1}\left(\mathcal{F}(\bar{s}) \odot \mathcal{F}(\bar{k})\right) \tag{8.4}$$

$$d = \bar{y} + e = \bar{M}\bar{c} + e \tag{8.5}$$

$$d \approx y + e = A(x) + e = F(x) + e = \mathcal{F}^{-1}\left(\mathcal{F}(x) \odot \mathcal{F}(k)\right) \tag{8.6}$$

where $\bar{c}$ is the convolved signal, $\bar{F}$ is the accurate physics model, $\bar{k}(j) = \mathbf{k}(\bar{t}(j), \bar{\xi})$, $\bar{y}$ is the accurate noiseless model prediction, $\bar{M}$ extracts the component of $\bar{c}$ in Ω_x and downsamples to the discretisation of d, y is the simple noiseless model prediction, F is the simple physics model and $k(j) = \mathbf{k}(t_x(j), \mu_\xi)$. Note that $y = A(x) = F(x)$ while $\bar{y} = \bar{A}(\bar{x}, \bar{z}, \bar{\xi}) = \bar{M}\bar{F}(\bar{x}, \bar{z}, \bar{\xi})$.

Shown in Figure 8.1 is the original image. Shown in Figure 8.2 is the image in the region of interest. Shown in Figure 8.3 is the image in the region of interest at the "acceptable" resolution of x.

Shown in Figure 8.4 is the image convolved with $\bar{\xi} = 2$. Figure 8.5 shows the convolved image in the region of interest. Figure 8.6 shows the measured data $d = \bar{A}(\bar{x}, \bar{z}, 2) + e$ where $\sigma_e = 0.02$. This translates to around 5% noise. Figures 8.7 through 8.9 use kernel $\bar{\xi} = 3$.

Let $\bar{s} \sim \mathcal{N}(0_{N_{\bar{s}}}, \Gamma_{\bar{s}})$ be a smoothness prior. Note that $\Gamma_{\bar{s}} \in \mathbb{R}^{N_{\bar{s}} \times N_{\bar{s}}}$ i.e. a $262,144 \times 262,144$ matrix. Rather than try and form the covariance with a covariance function, I construct the information filter directly as

$$\bar{s}_j = L_{\bar{s}}\bar{w}_j = h\bar{F}(\bar{w}_j, l) \tag{8.7}$$

$$= h\mathcal{F}^{-1}\left(\mathcal{F}(\bar{w}_j) \odot \mathcal{F}(k_l)\right) \tag{8.8}$$

$$= \mathcal{F}^{-1}\left(\mathcal{F}(\bar{w}_j) \odot g_{\bar{s}}\right) \tag{8.9}$$

where $k_l(j) = \mathbf{k}(t(j), \bar{\xi} = l)$. This construction only requires $g_{\bar{s}} \in \mathbb{R}^{N_{\bar{s}}}$ be stored, and draws can be computed in $\mathcal{O}\left(N_{\bar{s}}\log(N_{\bar{s}})\right)$ flops. More details on this approach can be found in Section 5.1.3. I choose $h = 2$ and $l = 1.5$, as the corresponding draws have "bumps" with approximate amplitude and width of the features I expect to recover.

A sample conditional mean type approach could be implemented, as described in Section 7.1. For example, the estimate

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\Gamma_e^{-1} + \Gamma_e^{-1} Q_{\bar{y}} \left(M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}}\right)^{-1} Q_y^T \Gamma_e^{-1}\right)(d_t - \hat{\mu}_d) \tag{8.10}$$

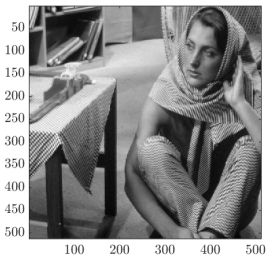


Figure 8.1: "Full" ground truth $\bar{s}_t$ on Ω

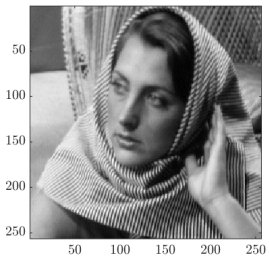


Figure 8.2: High resolution ground truth $\bar{x}_t$

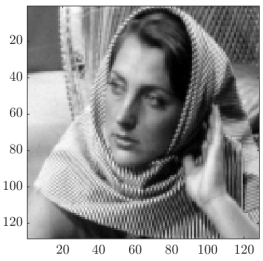


Figure 8.3: Acceptable resolution ground truth x_t

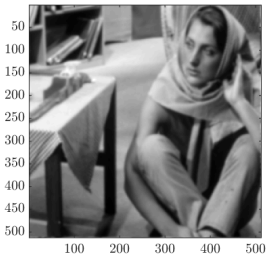


Figure 8.4: Narrow kernel convolution $\bar{F}(\bar{x}, \bar{z}, 2)$

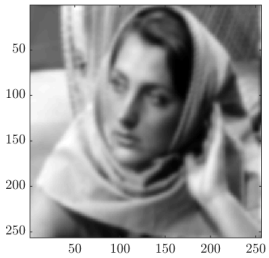


Figure 8.5: $y_{\xi=2} = \bar{A}(\bar{x}, \bar{z}, 2)$

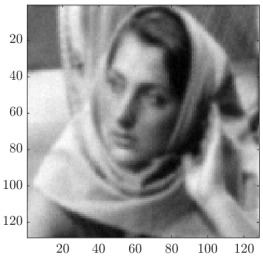


Figure 8.6: $d_{\xi=2} = y_{\xi=2} + e_t$

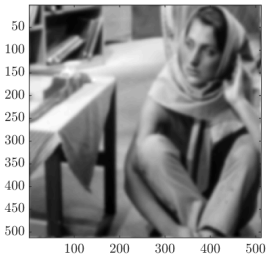


Figure 8.7: Wide kernel convolution $\bar{F}(\bar{x}, \bar{z}, 3)$

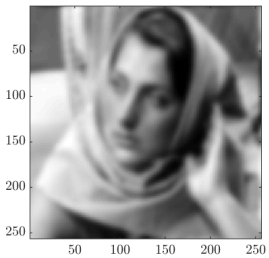


Figure 8.8: $y_{\xi=3} = \bar{A}(\bar{x}, \bar{z}, 3)$

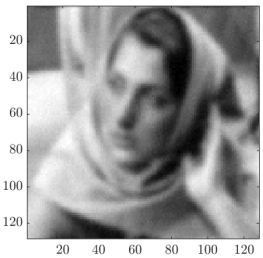


Figure 8.9: $d_{\xi=3} = y_{\xi=3} + e_t$

could be constructed with samples x_j and $\bar{y}_j$. However, it was found that $r_{\bar{y}} \approx n_{\bar{y}}$ and $r_x \approx n_x$ for this problem. This means a large number of samples need to be computed in order for the sample estimates to converge, and the resulting matrices are still relatively large. I therefore do not implement such an estimator for this problem.

8.1 Filter Construction with BAE and Local Sample Approximation

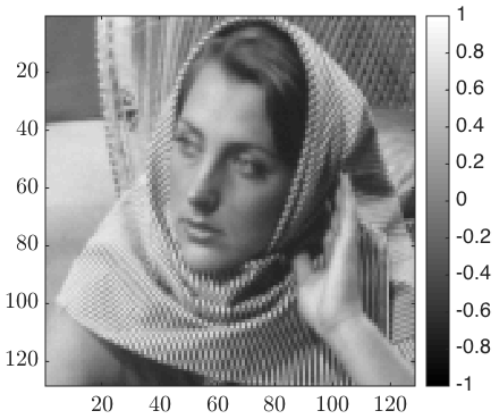


Figure 8.10: Ground truth x_t

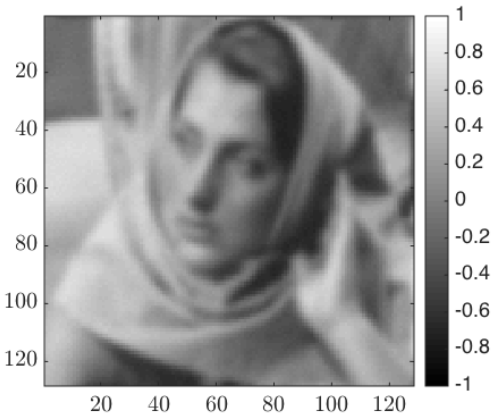


Figure 8.11: Data d_t

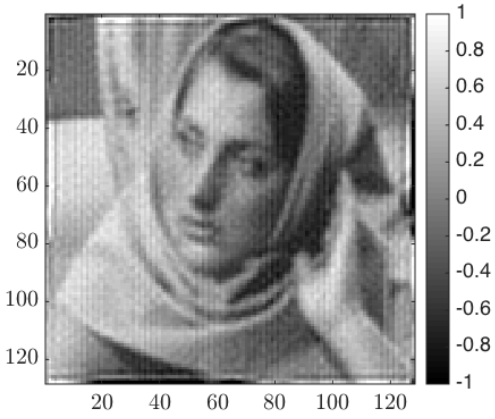


Figure 8.12: Estimate x_g of x_t

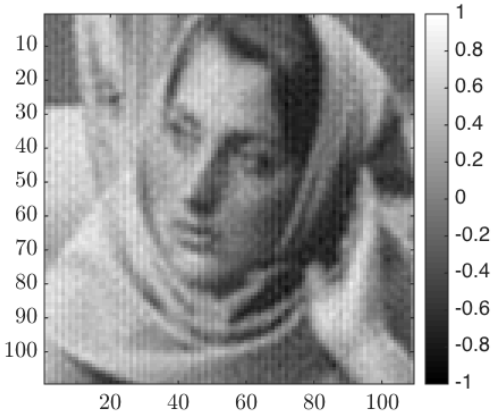


Figure 8.13: x_g, edges removed

I implement an estimation scheme incorporating concepts of BAE and local sample approximation. The methodology is similar to that described in Section 6.7. I draw m samples $\bar{s}_j$, $\bar{\xi}_j$ and e_j, from which $x_j = M\bar{s}_j$ and $d_j = \bar{A}(\bar{s}_j, \bar{\xi}_j) + e_j$ are computed. I compute

corresponding filters

$$g_j(k) = \frac{\Big(\mathcal{F}(x_j)\Big)(k)}{\Big(\mathcal{F}(d_j)\Big)(k)} \tag{8.11}$$

where $\mathcal{F}$ is the Fourier transform on the same scale as x. I form

$$g_\pi = \frac{1}{m}\sum g_j \tag{8.12}$$

the average filter. Note that each sample is computed with $\mathcal{O}(N_{\bar{s}}\log(N_{\bar{s}}))$ flops.

I compute estimates x_g of x_t as

$$x_g = \mathcal{F}^{-1}(g_\pi \odot \mathcal{F}(d)) \tag{8.13}$$

noting that such estimates require just $g_\pi \in \mathbb{R}^{N_x}$ in memory and $\mathcal{O}\Big(N_x\log(N_x)\Big)$ flops at the online stage.

I compute solutions with $m = 1,000$ samples. The ground truth x_t is shown in Figure 8.10. The measured data d_t computed with extra fine model $\bar{\bar{A}}(\bar{x},\bar{z},\bar{\xi})$ is shown in Figure 8.11. The estimate x_g of x_t from d_t is shown in Figure 8.12. Note that the reconstruction is poor near the edges of the image. The estimate with the edges removed is shown in Figure 8.13.

8.2 Iterative BAE MAP solution

The sample CM estimate was found to involve high rank sample entities that did not converge sufficiently quickly. This is a linear A Gaussian x problem, so the CM and MAP are the same. However, the MAP is computed differently, and with some different sample entities. In this section, I present a MAP estimate incorporating BAE and local sample approximations.

The BAE MAP estimate can be stated as

$$x_{\text{MAP}} = \begin{pmatrix}\tilde{L}_{\nu|x}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})\\ \tilde{L}_x\end{pmatrix}^\dagger \begin{pmatrix}\tilde{L}_{\nu|x}(d+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x-\hat{\mu}_\varepsilon)\\ \tilde{L}_x\mu_x\end{pmatrix} \tag{8.14}$$

$$= \hat{A}^\dagger\hat{d} \tag{8.15}$$

where $\tilde{L}_{\nu|x}^T\tilde{L}_{\nu|x} = \Gamma_{\nu|x}^{-1} = (\Gamma_e+\hat{\Gamma}_{\varepsilon|x})^{-1}$. Note that the MAP estimate makes use of the sample entities

$$\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} = Q_\varepsilon R_\varepsilon R_x^T(R_xR_x^T)^{-1}Q_x^T \tag{8.16}$$

and

$$\hat{\Gamma}_{\varepsilon|x} = \frac{1}{m} Q_\varepsilon (R_\varepsilon R_\varepsilon^T - R_\varepsilon R_x^T (R_x R_x^T) R_x R_\varepsilon^T) Q_\varepsilon^T. \tag{8.17}$$

While the quantities relating to x e.g. $(R_x R_x^T)$ may not have converged, it may be that the MAP estimate is still representative.

It is worth mentioning the enhanced error model, which treats ε as independent of x. In the enhanced error model, $\Gamma_{\nu|x} = \Gamma_\nu = \Gamma_e + \frac{1}{m} Q_\varepsilon R_\varepsilon R_\varepsilon^T Q_\varepsilon^T$ and $\hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} = 0_{n_d, n_x}$. The enhanced error model may be a good approach when Γ_x is of high effective rank, as the sample quantities used may converge in fewer samples.

Recall that $\Gamma_{\nu|x} = \Gamma_e + \hat{\Gamma}_{\varepsilon|x} = \Gamma_e + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T$. The inverse is

$$\begin{aligned} \Gamma_{\nu|x}^{-1} &= (\Gamma_e + \hat{\Gamma}_{\varepsilon|x})^{-1} && (8.18)\\ &= (\Gamma_e + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T)^{-1} && (8.19)\\ &= \Gamma_e^{-1} - \Gamma_e^{-1} Q_\varepsilon (M_{\varepsilon|x}^{-1} + Q_\varepsilon^T \Gamma_e^{-1} Q_\varepsilon)^{-1} Q_\varepsilon^T \Gamma_e^{-1} && (8.20) \end{aligned}$$

by the matrix inversion lemma. In this problem $\Gamma_e = \sigma_e^2 I_{N_x}$ so

$$\begin{aligned} \Gamma_{\nu|x}^{-1} &= \Gamma_e^{-1} - \Gamma_e^{-1} Q_\varepsilon (M_{\varepsilon|x}^{-1} + Q_\varepsilon^T \Gamma_e^{-1} Q_\varepsilon)^{-1} Q_\varepsilon^T \Gamma_e^{-1} && (8.21)\\ &= \frac{1}{\sigma_e^2} I_{N_x} - \frac{1}{\sigma_e^4} Q_\varepsilon (M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_\varepsilon})^{-1} Q_\varepsilon^T && (8.22) \end{aligned}$$

and recall that $M_{\varepsilon|x}^{-1}$ can be inverted in parallel with sample being computed as described in Section 6.4.2.

I construct a sequence x_k approximating x_{MAP} by Landweber iterations i.e.

$$x_{k+1} = x_k + \beta(\hat{A}^T \hat{d} - \hat{A}^T \hat{A} x_k) \tag{8.23}$$

where $\beta \in \mathbb{R}$ is the step length. I incorporate the earlier estimate x_g of x_t from Section 8.1 by setting $x_0 = x_g$. Note that

$$\begin{aligned} \hat{A}^T \hat{d} &= (A + \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1})^T \Gamma_{\nu|x}^{-1} (d + \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1} \mu_x - \mu_d - \hat{\mu}_\varepsilon) + \Gamma_x^{-1} \mu_x && (8.24)\\ &= (A + \hat{\Gamma}_{\varepsilon x} \hat{\Gamma}_x^{-1})^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_\varepsilon (M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_\varepsilon})^{-1} Q_\varepsilon^T \right) (d + \mu_\omega) + \Gamma_x^{-1} \mu_x && (8.25) \end{aligned}$$

where

$$\mu_\omega = Q_\varepsilon R_\varepsilon R_x^T (R_x R_x^T)^{-1} Q_x^T \mu_x - \mu_d - \hat{\mu}_\varepsilon \tag{8.26}$$

and

$$\hat{A}^T\hat{A} = \begin{pmatrix}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\tilde{L}_{\nu|x}^T & \tilde{L}_x^T\end{pmatrix}\begin{pmatrix}\tilde{L}_{\nu|x}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})\\ \tilde{L}_x\end{pmatrix} \tag{8.27}$$

$$= (A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\Gamma_{\nu|x}^{-1}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) + \Gamma_x^{-1} \tag{8.28}$$

$$= (A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\left(\frac{1}{\sigma_e^2}I_{n_d} - \frac{1}{\sigma_e^4}Q_\varepsilon(M_{\varepsilon|x}^{-1}+\frac{1}{\sigma_e^2}I_{r_\varepsilon})^{-1}Q_\varepsilon^T\right)(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) + \Gamma_x^{-1} \tag{8.29}$$

noting that $A = A^T$. Note that $\hat{A}^T\hat{d}$ and $\hat{A}^T\hat{A}x$ can be evaluated efficiently by taking care with the order of operations. Specifically, $\hat{A}^T\hat{A}x$ can be computed in $\mathcal{O}\Big(N_x\big(r_x + r_\varepsilon + \log(N_x)\big)\Big)$ flops. Compare this to using $\hat{A}^\dagger$ explicitly, which would require $\mathcal{O}(N_x^3)$ to find $\hat{A}^\dagger$, and then $\mathcal{O}(N_x^2)$ flops to compute $\hat{A}^\dagger\hat{d}_t$ at the online stage.

Landweber iterates converge when $0 < \beta < \frac{2}{\hat{\lambda}}$ where $\hat{\lambda} = \text{norm}(\hat{A}^T\hat{A})$ is the largest eigenvalue of $(\hat{A}^T\hat{A})$ i.e. $\hat{\lambda} = \max_{u:\|u\|=1}\{\left\|\hat{A}^T\hat{A}u\right\|\}$. I computed several $b_j = \frac{\left\|\hat{A}^T\hat{A}w_j\right\|}{\|w_j\|}$ where w_j is a white noise sample. I found that $\max_j\{b_j\} \approx 10^5$, so took $\beta = 10^{-6}$. This is the norm estimation method of [122], an early example of a probabilistic linear algebra algorithm. I computed 100 Landweber iterates, however the iterates were found to change little after iterate 8. For online estimation of the MAP, I chose to take 10 iterates.

The BAE sample entities and BAE filter are computed from only 200 draws. The MAP is estimated in 10 Landweber iterations. The ground truth x_t is shown in Figure 8.14. The noisy data d_t is shown in Figure 8.15. The MAP estimate is shown in Figure 8.16. Note that the MAP estimate performs poorly near the edges, but better than x_g. Shown in Figure 8.17 is the MAP with the edges removed. Computed on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive, the offline phase took 35 seconds, and the online took 0.63 seconds.

The estimation scheme of this section integrates matrix free methods, BAE, recursive QR, sample filter construction, local sample approximations and penalty subspaces. The offline phase required $\mathcal{O}\Big(mN_x(r_x+r_\varepsilon+\log(N_x)) + mN_{\bar{s}}\log(N_{\bar{s}})\Big)$ flops with $m = 200$ a sufficient number of samples. It also required storage of $\mathcal{O}(N_{\bar{s}} + (r_x+r_\varepsilon)N_x)$ numbers in memory. The online estimation involved $\mathcal{O}(N_x(r_x + r_\varepsilon + \log(N_x))$ flops. Contrast this to a straightforward computation of $\hat{\tilde{A}}^\dagger$, which would require $\mathcal{O}(N_{\bar{s}}^3)$ flops and numbers in memory at the offline phase to form and store inversions at the full resolution over the entire domain, and then $\mathcal{O}(N_{\bar{s}}^2)$ flops to compute an inversion at the online phase.

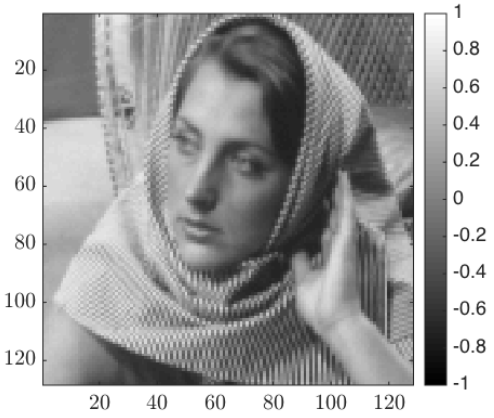


Figure 8.14: True x_t

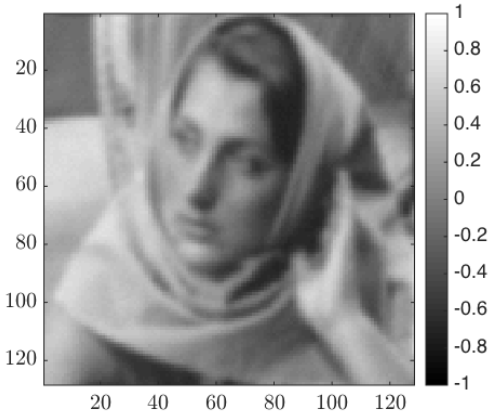


Figure 8.15: Data d

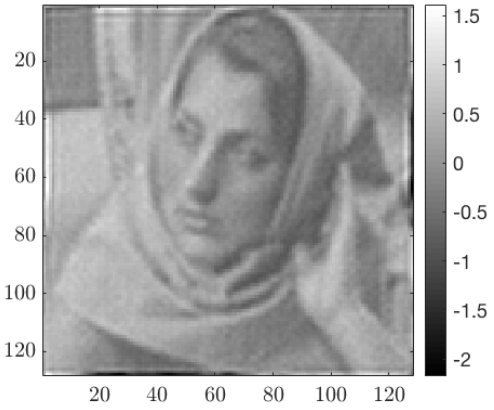


Figure 8.16: x_{MAP}

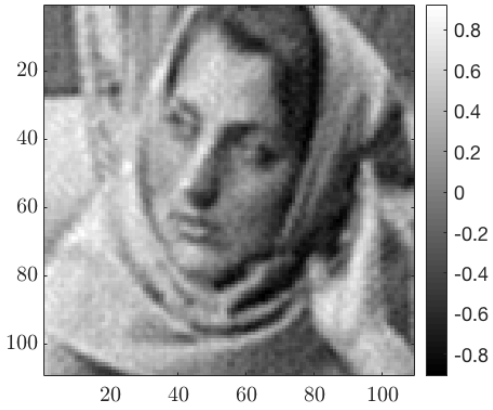


Figure 8.17: x_{MAP}, padding removed

8.3 Uncertainty Quantification

Let us now consider the posterior covariance

$$\Gamma_{x|d} = \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} \tag{8.30}$$

$$= \Gamma_x - \Gamma_x A^T(\Gamma_e + A\Gamma_x A^T + \hat{\Gamma}_{\varepsilon|x})^{-1}A\Gamma_x \tag{8.31}$$

$$= \left(\Gamma_x^{-1} + A^T(\Gamma_e + \hat{\Gamma}_{\varepsilon|x})^{-1}A\right)^{-1}. \tag{8.32}$$

Note that in this problem the matrix A is not explicitly constructed. Each of the above forms of the posterior covariance involves prohibitively large inversions. This section instead considers various methods of computing an approximate posterior covariance.

8.3.1 Sample Posterior Covariance

We can approximate the posterior covariance from samples using recursive QR as

$$\hat{\Gamma}_{x|d} = \hat{\Gamma}_x - \hat{\Gamma}_{xd}(\Gamma_e + \hat{\Gamma}_y + \hat{\Gamma}_{\varepsilon|x})^{-1}\hat{\Gamma}_{dx} \tag{8.33}$$

$$= Q_x M_x Q_x^T - Q_x M_{xy} Q_y^T (\Gamma_e + Q_y M_y Q_y^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T)^{-1} Q_y M_{xy}^T Q_y^T \tag{8.34}$$

$$= Q_x \Big(M_x - M_{xy} Q_y^T (\Gamma_e + Q_y M_y Q_y^T + Q_\varepsilon M_{\varepsilon|x} Q_\varepsilon^T)^{-1} Q_y M_{xy}^T \Big) Q_x^T \tag{8.35}$$

$$= Q_x \Big(M_x - M_{xy} Q_y^T (\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} Q_y M_{xy}^T \Big) Q_x^T \tag{8.36}$$

as described in Section 7.1. In this case, $\Gamma_e = \sigma_e^2 I_{n_d}$ so

$$\hat{\Gamma}_d^{-1} = (\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} \tag{8.37}$$

$$= (\sigma_e^2 I_{n_d} + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} \tag{8.38}$$

$$= \frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{y,\varepsilon} (M_{y,\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_y + r_\varepsilon})^{-1} Q_{y,\varepsilon}^T \tag{8.39}$$

$$= \frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{y,\varepsilon} \begin{pmatrix} (M_y^{-1} + \frac{1}{\sigma_e^2} I_{r_y})^{-1} & 0_{r_y, r_\varepsilon} \\ 0_{r_\varepsilon, r_y} & (M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_\varepsilon})^{-1} \end{pmatrix}^{-1} Q_{y,\varepsilon}^T \tag{8.40}$$

$$= \frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_y (M_y^{-1} + \frac{1}{\sigma_e^2} I_{r_y})^{-1} Q_y^T - \frac{1}{\sigma_e^4} Q_\varepsilon (M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_\varepsilon})^{-1} Q_\varepsilon^T. \tag{8.41}$$

This can be substituted into $\hat{\Gamma}_{x|d}$ to form

$$\hat{\Gamma}_{x|d} = Q_x \Big(M_x - M_{xy} Q_y^T (\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} Q_y M_{xy}^T \Big) Q_x^T \tag{8.42}$$

$$= Q_x \Big(M_x - \frac{1}{\sigma_e^2} M_{xy} \big(I_{r_y} + \frac{1}{\sigma_e^4} (M_y^{-1} + \frac{1}{\sigma_e^2} I_{r_y})^{-1} + \frac{1}{\sigma_e^4} Q_y^T Q_\varepsilon (M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2} I_{r_\varepsilon})^{-1} Q_\varepsilon^T Q_y \big) M_{xy}^T \Big) Q_x^T \tag{8.43}$$

$$= Q_x M_{e,x|d} Q_x^T \tag{8.44}$$

noting the orthonormality of the columns of Q_y. Note that $Q_x \in \mathbb{R}^{N_x \times r_x}$ and $M_{e,x|d} \in \mathbb{R}^{r_x \times r_x}$. The diagonal $\hat{\sigma}^2$ of $\Gamma_{x|d}$ can be constructed by first forming

$$Q_x M_{e,x|d} = B_{x|d} \in \mathbb{R}^{N_x \times r_x} \tag{8.45}$$

and then

$$\hat{\Gamma}_{x|d}(j,j) = B_{x|d}(j,:)Q_x(:,j)^T = \hat{\sigma}^2(j) \tag{8.46}$$

which avoids ever forming an $N_x \times N_x$ matrix in memory.

I constructed the above using $m = 200$ samples. On my laptop, the offline computations took 34.6 seconds, finding the MAP with Landweber iterations took 0.639 seconds and computing the posterior covariance diagonal $\hat{\sigma}^2$ in 0.638 seconds. This demonstrates the computational efficiency of this form for the posterior covariance. Unfortunately, the above approximate covariance diagonal does not seem to be representative. $\hat{\sigma}^2$ is plotted over Ω_x in Figure 8.18. The "noisy" appearance of $\hat{\sigma}^2$ indicates that the posterior covariance estimate has not converged i.e. $m = 200$ is too small. While $m = 200$ seems to have been enough to compute the MAP point estimate, it is not enough to estimate $\hat{\Gamma}_{x|d}$. The BAE MAP estimate along the 20'th column of pixels is shown in Figure 8.19 along with $3\hat{\sigma}$ posterior error intervals. These intervals are far too wide, likely a result of sample estimates involving x_j not having converged/$m = 200$ being too small. Recall that the conditional mean estimate was also found to involve entities of too high effective rank to be used effectively.

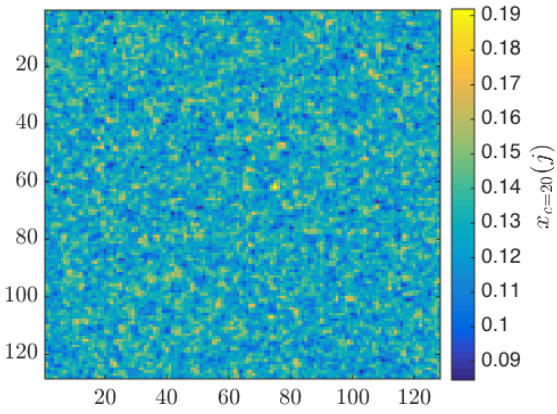


Figure 8.18: Posterior covariance diagonal $\hat{\sigma}^2$

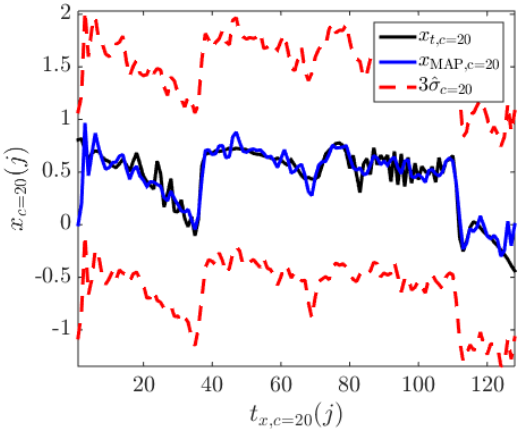


Figure 8.19: Estimates on pixel column 20

8.3.2 Sample Posterior Covariance by Diagonal Approximations

In this section, sample covariances are approximated as diagonal. This approach is widely used, with examples and justifications provided in [65, 106]. The justification in this case is threefold. Firstly, the ultimate confidence intervals used are assuming a diagonal posterior covariance. Secondly, the correlation length used in $\pi(x)$ means Γ_x is effectively diagonal. Thirdly and more generally, the sample estimate of e.g. $\hat{\Gamma}_x \in \mathbb{R}^{n_x \times n_x}$ requires N_x^2 numbers

be estimated from samples whereas by approximating Γ_x as diagonal i.e. $\hat{\Gamma}(j,j) = \hat{\sigma}_x(j)$ and 0 otherwise, only the N_x elements of $\hat{\sigma}_x$ need to be estimated from samples. This may allow estimates of $\hat{\sigma}_x$ to converge in fewer samples than estimates of a full $\hat{\Gamma}_x$, which is particularly important when taking inverses.

By taking diagonal approximations, $\hat{\sigma}$ is formed as

$$\begin{aligned}\hat{\sigma}^2 &= \text{diag}(\Gamma_{x|d}) && (8.47)\\ &= \text{diag}(\Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx}) && (8.48)\\ &\approx \hat{\sigma}_x^2 - \hat{\sigma}_x \odot \hat{\sigma}_y \odot (\hat{\sigma}_e^2 + \hat{\sigma}_y^2 + \hat{\sigma}_\varepsilon^2)^{-1} \odot \hat{\sigma}_y \odot \hat{\sigma}_x && (8.49)\end{aligned}$$

where

$$\hat{\sigma}_x^2 = \frac{1}{m}\sum_{j=1}^{m}(x_j - \hat{\mu}_x) \odot (x_j - \hat{\mu}_x) \tag{8.50}$$

and similarly for e.g. $\hat{\sigma}_y$. The computational cost of approximating the covariances like this is trivial. I computed $m = 200$ samples, and forming $\hat{\sigma}$ took 0.017 seconds. This estimate of $\hat{\sigma}^2$ is plotted in Figure 8.20. Note that $\hat{\sigma}$ is much higher near the edges of the domain, as expected as the approximation truncates data from outside Ω_x. The diagonal approximation $\hat{\sigma}$ in Figure 8.20 appears less noisy than that of Figure 8.18, potentially indicating that this approximation has converged. The MAP estimate with $3\hat{\sigma}$ posterior error intervals is shown in Figure 8.21. This posterior error interval appears closer to representative, although still perhaps wider than necessary.

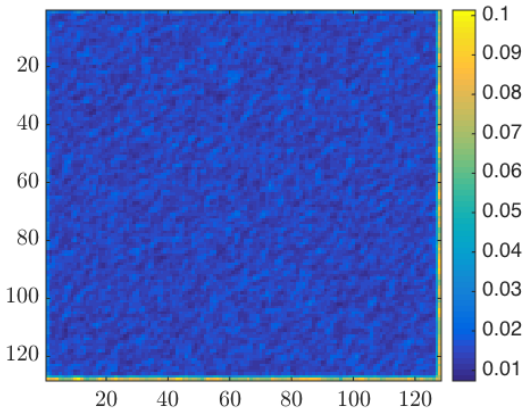


Figure 8.20: $\hat{\sigma}^2$, diagonal sample covariances

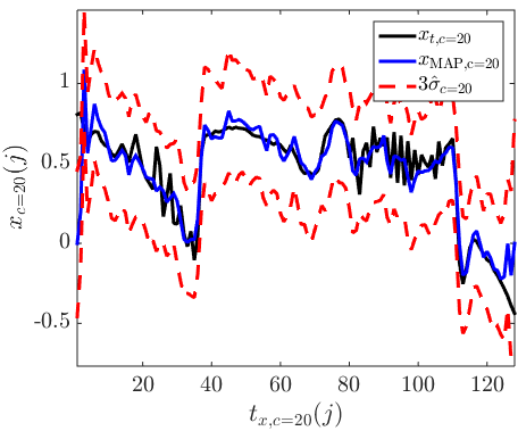


Figure 8.21: Estimates on pixel column 20

8.3.3 Posterior Residual Estimation of Posterior Covariance

In this section, I make use of the speed at which estimates x_{MAP} can be computed to implement a posterior residual approach. Recall that estimation of x_{MAP} makes use of m samples. With the estimation scheme constructed, I compute an additional $\hat{m}$ samples $d_j = \bar{A}(\bar{x}_j, \bar{z}_j, \bar{\xi}_j) + e_j$ and $x_j = P_{x,\bar{x}}\bar{x}_j$. From this, I compute the estimate $x_{\text{MAP,j}}$ of x_j from d_j, and the residual $\zeta_j = x_{\text{MAP,j}} - x_j$. Note that if BAE is incorporated correctly, $\mu_\zeta \approx 0_{n_x,1}$. The posterior error interval is approximated as

$$\hat{\sigma}_2 \approx \frac{1}{\hat{m}-1}\sum_{j=1}^{\hat{m}}(\zeta_j - \hat{\mu}_\zeta) \odot (\zeta_j - \hat{\mu}_\zeta) \tag{8.51}$$

implicitly making a diagonal approximation.

I computed $m = 200$ samples for BAE in 35 seconds. I then estimated the posterior variance with $\hat{m} = 50$ simulated reconstructions in 37 seconds. The mean posterior residual $\hat{\mu}_\zeta$ is shown in Figure 8.22. The posterior residual mean seems noisy and has relatively large amplitude, suggesting that either m or $\hat{m}$ need to be larger. Recall that $m = 200$ gave satisfactory point estimates, and $\hat{m} = 50$ has already doubled the computational cost at the offline phase. I continue with $m = 200$, $\hat{m} = 50$, not wanting to increase the computational cost further. Shown in Figure 8.23 is the posterior residual variance $\hat{\sigma}^2_\zeta$, which I use as an approximation to the posterior variance. Observe that the residual variance is larger near the edges of Ω_x. The MAP estimate along the 20'th column of pixels is shown in Figure 8.24 with $3\hat{\sigma} = 3\hat{\sigma}_\zeta$ posterior error intervals. These intervals appear too wide, and the estimate performs poorly near the boundary. Shown in Figure 8.25 is the MAP estimate with pixels near the boundary removed, and a $1\hat{\sigma}$ posterior error interval. The $1\hat{\sigma}$ posterior error interval seems to contain around 99% of the ground truth, which would be expected for a $3\hat{\sigma}$ posterior error interval.

I increase the offline computational cost by taking $m = 1,000$ samples for the BAE stage, and $\hat{m} = 200$ samples for the posterior residual uncertainty quantification. The BAE stage took 514 seconds, and the posterior residual stage took 626 seconds. Online estimation now takes 2.83 seconds as r_x and r_ε increase from 200 to 1,000. r_x and r_ε could be lowered by increasing the recursive QR tolerances, or by computing further decompositions e.g. $U\Lambda U^T = R_x R_x^T$ and truncating.

The MAP estimate is shown in Figure 8.26. The MAP estimate with the edges removed is shown in Figure 8.27. Note that the MAP estimate with $m = 1,000$ does not appear much better than the estimate with $m = 200$. This is expected, as the quantities constructed from samples in my estimation scheme were found to have sufficiently converged at $m = 200$. The residual mean is shown in Figure 8.28. Recall that this quantity is expected to converge to 0, and note the colour scale. The residual variance on Ω_x is shown in Figure 8.29. Note that the variance appears "smooth", and is largest near the boundary. This is what I predicted the posterior variance should look like. The MAP estimate along the 20'th column of pixels with

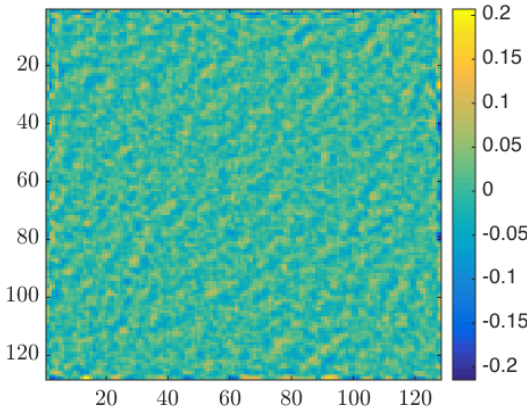


Figure 8.22: Residual mean $\hat{\mu}_\zeta$

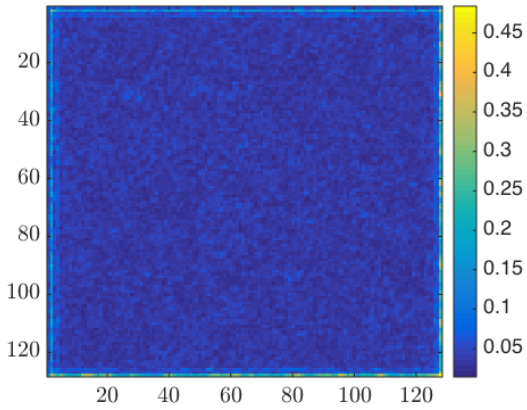


Figure 8.23: Residual variance $\hat{\sigma}^2$

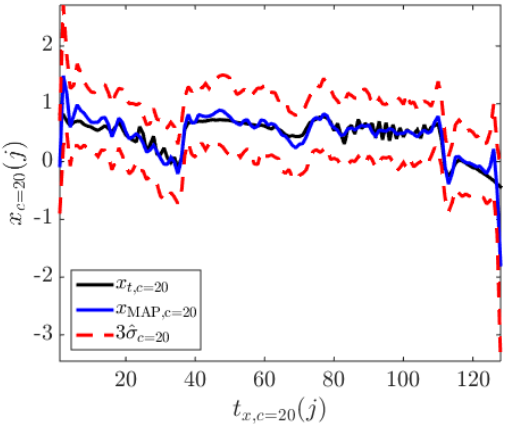


Figure 8.24: Reconstruction with $3\hat{\sigma}$ CI

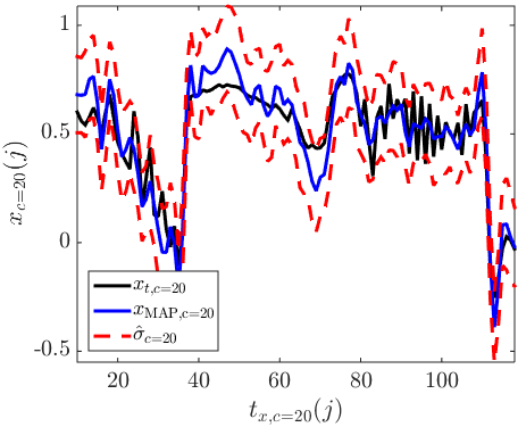


Figure 8.25: Zoomed reconstruction, $1\hat{\sigma}$ CI

$3\hat{\sigma}$ posterior error intervals is shown in Figure 8.30. The MAP estimate along pixel column 20 with the first and last 10 pixels removed and $1\hat{\sigma}$ posterior error intervals is shown in Figure 8.31. Note that the estimates with $m = 200$, $\hat{m} = 50$ and $m = 1,000$, $\hat{m} = 200$ appear similar. This suggests that $m = 200$, $\hat{m} = 50$ is sufficient. A more accurate MAP estimate would require a more fundamental change to the estimation scheme e.g. implement a TV prior. It could also be that posterior error intervals constructed as $\hat{\sigma}$ are not representative of the true uncertainty, likely due to the correlation structure of the problem.

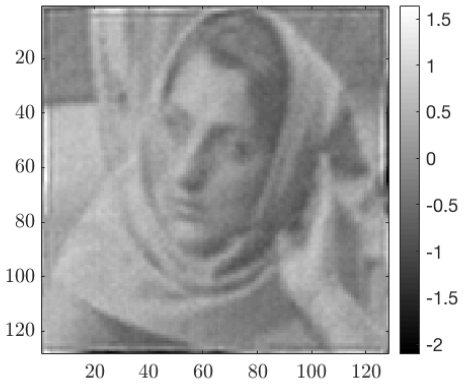


Figure 8.26: x_{MAP} from 1,000 BAE samples

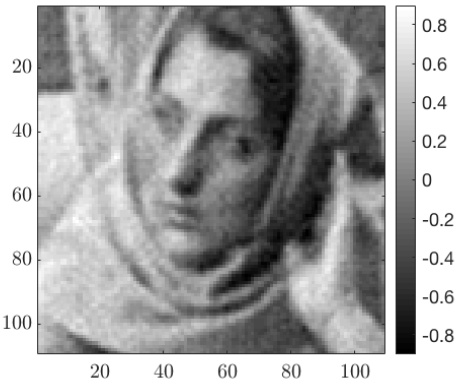


Figure 8.27: x_{MAP}, padding removed

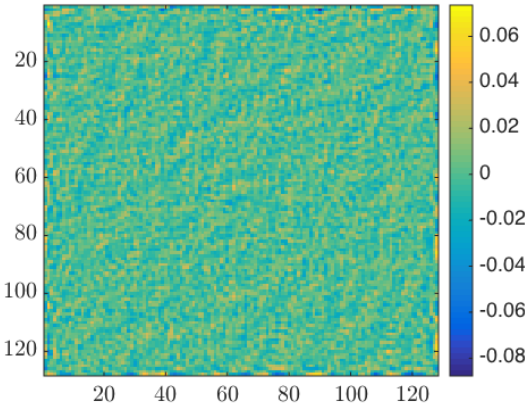


Figure 8.28: Residual mean $\hat{\mu}_\zeta$

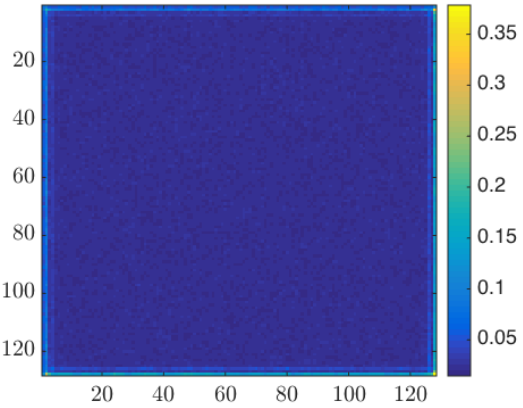


Figure 8.29: Residual variance $\hat{\sigma}^2$

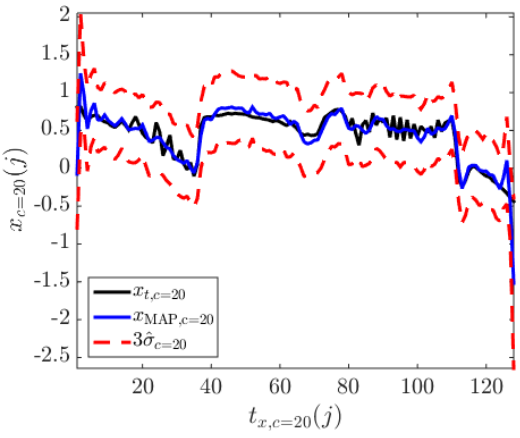


Figure 8.30: Reconstruction with $3\hat{\sigma}$ CI

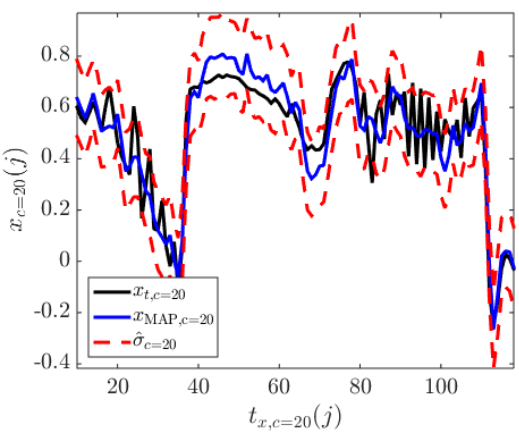


Figure 8.31: Zoomed reconstruction, $1\hat{\sigma}$ CI

8.4 Local Posterior Covariance Formation

Suppose only a few elements of $\Gamma_{x|d}$ are desired. For example, $\Gamma_{x|d}(\mathcal{I}_{c=20}, \mathcal{I}_{c=20}) = \hat{\sigma}_{c=20}$, the elements of $\hat{\sigma}^2$ corresponding to the 20'th column of pixels. I refer to estimating only $\Gamma_{x|d}(\mathcal{I}_{c=20}, \mathcal{I}_{c=20})$ as *local posterior covariance estimation*.

Recall that the posterior covariance is

$$\begin{aligned}
\Gamma_{x|d} &= \Gamma_x - \Gamma_{xd}\Gamma_d^{-1}\Gamma_{dx} && (8.52)\\
&= \Gamma_x - \Gamma_x A^T(\Gamma_e + A\Gamma_x A^T + \hat{\Gamma}_{\varepsilon|x})^{-1}A\Gamma_x && (8.53)\\
&= \left(\Gamma_x^{-1} + A^T(\Gamma_e + \hat{\Gamma}_{\varepsilon|x})^{-1}A\right)^{-1} && (8.54)\\
&= \left(\Gamma_x^{-1} + A^T(\frac{1}{\sigma_e^2}I_{N_x} + Q_\varepsilon M_{\varepsilon|x}Q_\varepsilon^T)^{-1}A\right)^{-1} && (8.55)
\end{aligned}$$

and the prior covariance Γ_x seems to be of high rank. The products $\Gamma_x x$, Ax and $\Gamma_d x$ can be computed rapidly. The diagonal elements of $\Gamma_{x|d}$ can be expressed as

$$\begin{aligned}
\Gamma_{x|d}(j,j) &= e_j^T\Gamma_{x|d}e_j && (8.56)\\
&= e_j^T\Gamma_x e_j - e_j^T\Gamma_x A^T\Gamma_d^{-1}A\Gamma_x e_j && (8.57)\\
&= v_{x,j}^T v_{x,j} - v_{Ax,j}^T\Gamma_d^{-1}v_{Ax,j} && (8.58)
\end{aligned}$$

where $v_{x,j} = L_x(e_j)$ and $v_{Ax,j} = AL_xL_xe_j$ are vectors that can be computed rapidly with Fourier transforms. The main hurdle is computing the product $v_{Ax,j}^T\Gamma_d^{-1}v_{Ax,j}$. Such problems are considered in [120], specifically in Section 7.2. The approach outlined in [120] is complex and difficult to implement. I instead propose estimating $v_{Ax,j}^T\Gamma_d^{-1}v_{Ax,j}$ by Landweber iterations. The aim is to evaluate

$$\begin{aligned}
v^T\Gamma_d^{-1}v &= v^T(\Gamma_e + A\Gamma_x A^T + Q_\varepsilon M_{\varepsilon|x}Q_\varepsilon^T)^{-1}v && (8.59)\\
&= v^T(L_eL_e^T + AL_xL_x^TA^T + Q_\varepsilon L_M Q_\varepsilon^T)^{-1}v && (8.60)\\
&= v^T(\tilde{L}_d^T\tilde{L}_d)v && (8.61)\\
&= (\tilde{L}_d v)^T(\tilde{L}_d v) && (8.62)\\
&= u^Tu && (8.63)
\end{aligned}$$

So we wish to find u such that $L_d u = v$ where $L_dL_d^T = \Gamma_d$. We can form L_d as

$$L_d = \begin{pmatrix} L_e & AL_x & Q_\varepsilon L_M \end{pmatrix} \tag{8.64}$$

and we note the only sizeable computation in the above is computing the Cholesky factor $L_M \in \mathbb{R}^{r_\varepsilon \times r_\varepsilon}$ of $M_{\varepsilon|x} \in \mathbb{R}^{r_\varepsilon \times r_\varepsilon}$. Recall that r_ε is likely small relative to n_d. The problem is

now finding u such that $L_d u = v$. We can find this by Landweber iterations as

$$u_{k+1} = u_k + \beta(L_d^T v - L_d^T L_d u_k). \tag{8.65}$$

While more elegant methods are presented in [120], the above Landweber approach is simple to implement, parallelisable, requires minimal RAM, is computationally tractable and simple to understand.

For each entry $\hat{\sigma}(j)$, the above scheme computes $v_{x,j} = L_x e_j$, $v_{xx,j} = L_x v_{x,j}$ and $v_{Ax,j} = A v_{xx,j}$. This takes $\mathcal{O}\Big(3N_x \log(N_x)\Big)$ flops. If we estimate $u_j = \tilde{L}_d v_{Ax,j}$ with p Landweber iterations (typically small e.g. $p = 5$), this involves $2p+1$ evaluations of $L_d = \begin{pmatrix} L_e & AL_x & Q_\varepsilon L_M \end{pmatrix}$, adding $\mathcal{O}\Big((2p+1)(N_x + 3N_x \log(N_x) + N_x r_\varepsilon^2)\Big)$ flops. Thus evaluating a single diagonal element involves $\mathcal{O}\Big(2(p+1)N_x(3\log(N_x) + r_\varepsilon^2)\Big)$ flops.

Computing the entirety of $\hat{\sigma} \in \mathbb{R}^{N_x}$ as above would require repeating the above steps N_x times. This represents a large (albeit tractable and parallelisable) computational cost. The above approach is useful when uncertainty quantification is only desired for a small subregion, in this example only along the 20'th column of pixels.

Entire columns of $\Gamma_{x|d}$ can be computed similarly to the above as $\Gamma_{x|d}(:, j) = \Gamma_{x|d} e_j$. However, we wish to avoid forming the entire matrix $\Gamma_{x|d} \in \mathbb{R}^{16,384 \times 16,384}$ in memory, hence the emphasis on just computing diagonal elements. The flop count for evaluating the entire column is of the same order as for evaluating just a diagonal element. In this example, the computational cost of estimating an element $\hat{\sigma}(j)$ and a posterior residual $\zeta_j = x_{\text{MAP,j}} - x_j$ is comparable.

I evaluate $\hat{\sigma}_{c=20}$, the posterior variances along the 20'th column of pixels in the image with the above Landweber type approach. The 200 BAE samples computed in 34 seconds. Computing $\hat{\sigma}_{c=20}$ takes another 17 seconds. Computing the MAP estimate over Ω_x took 0.63 seconds. The MAP estimate $x_{\text{MAP,c=20}}$ along the 20'th pixel column with $3\hat{\sigma}_{c=20}$ posterior error intervals is shown in Figure 8.32. The estimate with padding pixels removed and only $1\hat{\sigma}$ posterior error intervals is shown in Figure 8.33. Note that the posterior error intervals appear too wide, consistent with earlier estimates in this chapter. The local posterior covariance posterior error intervals of Figure 8.33 are wider than the posterior variance estimates of Figure 8.25 of Section 8.3.3, and compute in about half the time. The posterior residual approach however produces uncertainty estimates over all $16,384$ pixels, whereas the local posterior covariance approach only estimates uncertainty of the 128 pixels of interest, the 20'th column.

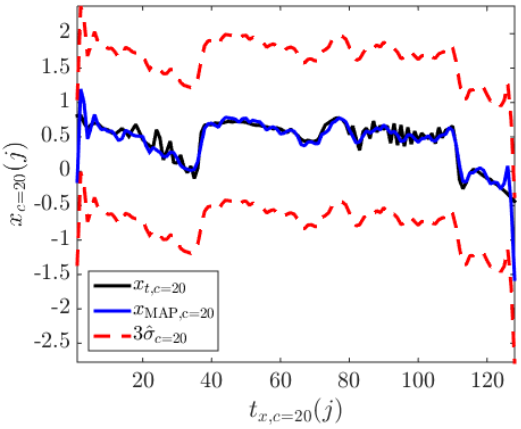


Figure 8.32: MAP estimate with $3\hat{\sigma}$ CI

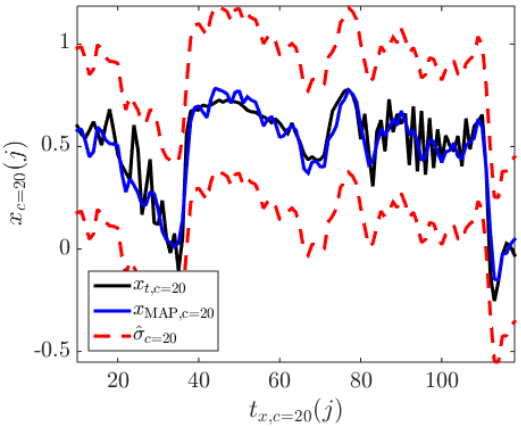


Figure 8.33: Padding removed, $1\hat{\sigma}$ CI

8.5 Discussion

For this problem , the effective rank of the prior covariance Γ_x was found to be too high for approximations such as

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + Q_x M_{xy} Q_y^T \left(\Gamma_e^{-1} + \Gamma_e^{-1} Q_y \left(M_y^{-1} + Q_y^T \Gamma_e^{-1} Q_y \right)^{-1} Q_y^T \Gamma_e^{-1} \right) (d_t - \hat{\mu}_d) \tag{8.66}$$

with approximate posterior covariance

$$\hat{\Gamma}_{x|d} = Q_x \left(M_x - M_{xy} Q_y^T (\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} Q_y M_{xy}^T \right) Q_x^T \tag{8.67}$$

to be computed reliably from a reasonable number of samples. In contrast, the Fourier space BAE type local sample estimate

$$x_g = \mathcal{F}^{-1}(g_\pi \odot \mathcal{F}(d)) \tag{8.68}$$

was found to give reasonable results from just $m = 200$ samples. This estimate was then used as the first step x_0 of the Landweber iteration

$$x_{k+1} = x_k + \beta(\hat{A}^T \hat{d} - \hat{A}^T \hat{A} x_k) \tag{8.69}$$

used to compute the MAP estimate

$$x_{\mathrm{MAP}} = \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{8.70}$$

$$= \hat{A}^\dagger \hat{d} \tag{8.71}$$

which gave fairly good estimates of x_t with reasonable computational cost.

Computations were performed on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive. The MAP estimate could be found as $x_{\text{MAP}} = \hat{A}^{\dagger}\hat{d}$. Note however that $\hat{A} \in \mathbb{R}^{(n_d+N_x)\times N_x}$. Treated as a full matrix, $\hat{A}$ has 536,870,912 entries. In IEEE 754 double-precision binary floating-point format, such a matrix would occupy 4 GB of memory. I attempted to compute the MAP as $x_{\text{MAP}} = \hat{A}\backslash\hat{d}$ in MATLAB, however the laptop shut down after several hours.

Several approximation posterior error intervals were tried. The most representative intervals were found by the posterior residual approach of Section 8.3.3. An additional $\hat{m} = 50$ samples $\zeta_j = x_{\text{MAP,j}} - x_j$ were computed, and the diagonal posterior residual approximation

$$\hat{\sigma}_2 \approx \frac{1}{m}\sum_{j=1}^{m}(\zeta_j - \hat{\mu}_\zeta) \odot (\zeta_j - \hat{\mu}_\zeta) \tag{8.72}$$

was used. While all the posterior error intervals attempted were found to be too wide, this is likely due to the strong correlation in the posterior not being explicitly accounted for in the posterior error estimates by e.g. computing trust regions from the entire covariance, not just the diagonal. Notably, the posterior residual estimates found with $\hat{m} = 200$ converged to a similarly wide posterior error estimate.

By the methods of this chapter, the MAP estimate took 35 seconds of offline computation and 0.63 seconds of online computation. The posterior residual construction of the posterior error estimates took an additional 37 seconds of offline computation. The MAP estimate (with padding removed) and posterior error estimate $\hat{\sigma}^2$ over Ω_x can be seen in Figures 8.17 and 8.23 respectively. The MAP estimate with $1\hat{\sigma}$ confidence interval along the 20'th column of pixels with padding removed is shown in Figure 8.31. The problem of this chapter demonstrates the effectiveness of combining recursive QR (to construct low rank sample entities), local sample approximation (the Fourier space BAE type local sample estimate) and BAE.

Chapter 9

X-Ray Tomography

This chapter applies methods of this thesis to the problem of x-ray tomography, also known as computed tomography (CT). The analysis of CT in this thesis is brief, as the problem is only being used to demonstrate the methods of Chapters 6 and 7. This chapter follows the conventions of [10, 123]. More information on CT can be found in [124], while [4, 5] consider CT more explicitly in terms of inverse problems. This chapter is primarily concerned with the application of the methods of this thesis to *local tomography*. The sections prior to 9.3 are primarily provided for context on the structure of the problem, so the analysis is light.

In x-ray tomography, an x-ray source and detector sit opposite each other in a gantry, with an object in the middle. The source emits a burst of x-rays, which pass through the object, are partially absorbed, and then arrive at the detector. The detector records a standard x-ray "shadow" image. Each shadow image is called an exposure. In x-ray tomography, exposures are made at various positions around the object as the gantry is moved, and the exposures are combined to produce a 3D estimate of the internal density distribution of the object. The density of a material is related to the wavelength of the x-rays used and the *absorption coefficient*. In this thesis, we consider estimation of the absorption coefficient distribution assuming x-rays of a homogeneous wavelength.

This chapter considers 2D x-ray tomography. That is, the object is represented in 2D and each exposure produces a 1D "image". The 2D x-ray tomography problem is equivalent to evaluating a single independent "slice" of the 3D tomography problem [5, 10, 33]. The 2D estimates can be combined into a 3D estimate [33].

This chapter considers *sparse full angle* CT. Only 18 exposures are used, hence *sparse*, and the gantry is rotated 180°, hence *full angle* [123]. Section 9.3 considers the *sparse full angle local tomography* problem in which only a subsection of the object is reconstructed. Traditional CT algorithms such as Radon transforms with filtered back projection are reviewed in [124]. However, these traditional methods are known to produce unsatisfactory estimates when applied to sparse CT [10, 125, 123, 126, 127]. The dense full angle CT problem is mildly

ill-posed, and the sparse full angle CT problem is more ill-posed [5, 33, 125].

Let Ω be a domain, and $\mathbf{t} \in \Omega$. Consider an x-ray beam, hereon called a *pencil*. Let L be the path of the pencil, going from the x-ray source to a pixel of the detector. Let $\mathbf{I}(\mathbf{t})$ be the intensity of the pencil at $\mathbf{t} \in \Omega$. Let $\mathbf{s}(\mathbf{t}) \in \mathbb{R}$ be the absorption coefficient at $\mathbf{t} \in \Omega$. Over a small distance $d\mathbf{t}$ the intensity changes as

$$\frac{d\mathbf{I}}{\mathbf{I}} = -\mathbf{s}(\mathbf{t})d\mathbf{t}. \tag{9.1}$$

Let $\mathbf{I}_0$ be the intensity of the pencil when emitted and I_1 the intensity of the pencil when detected. Therefore

$$\int_L \mathbf{s}(\mathbf{t})d\mathbf{t} = -\int_L \frac{\mathbf{I}'(\mathbf{t})}{\mathbf{I}(\mathbf{t})}d\mathbf{t} = \log(\mathbf{I}_0) - \log(\mathbf{I}_1) \tag{9.2}$$

i.e. the measured data are line integrals of $\mathbf{s}(\mathbf{t})$ along pencil paths L. Let $\mathbf{y}(j) = \log(\mathbf{I}_{0,j}) - \log(\mathbf{I}_{1,j})$ be the predicted measured value corresponding to the j'th pencil. Let $s \in \mathbb{R}^{N_s}$ be a finite dimensional approximation of $\mathbf{s}$ e.g. a pixelated image. Let

$$y = As \tag{9.3}$$

where $A \in \mathbb{R}^{n_d \times N_s}$ is a finite dimensional forward operator and $y(j) \approx \mathbf{y}(j)$. The j'th row of A is an approximation to $\int_{L_j} \mathbf{s}(\mathbf{t})d\mathbf{t}$, the integral of $\mathbf{s}$ along the line L_j, the path of the j'th pencil. Let

$$d = As + e \tag{9.4}$$

where $d \in \mathbb{R}^{n_d}$ is the data and $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$ is the measurement noise i.e. this is a linear A additive Gaussian noise problem.

An example beam intensity is shown in Figure 9.1. Observe how the intensity decays faster where the absorption coefficient is higher. Let each exposure consist of n_p pencils. Note that the number of pencils is the number of pixels on the x-ray detector. Let n_a be the number of exposures. The data dimension is therefore $n_d = n_a n_p$.

9.1 CT with Coarse Discretisation

This section considers a coarse discretisation finite dimensional approximation to the sparse full angle CT problem. The purpose of this section is to demonstrate some of the structure of the CT problem. By using a coarse discretisation, the structure of the CT problem can be explored without also employing the methods of this thesis.

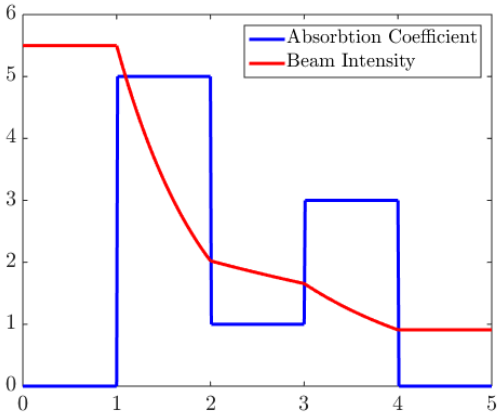


Figure 9.1: Beam intensity decay

Let $\Omega = [1, 64] \times [1, 64]$. Let

$$t = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ \\ \vdots & \vdots \\ 64 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 64 & 64 \end{pmatrix} \in \mathbb{R}^{4,096 \times 2} \tag{9.5}$$

be the spatial discretisation i.e. $\Delta_t = 1$. Let $x(j) = \mathbf{s}(t(j, :))$ be the finite dimensional approximation to the absorption coefficient distribution in Ω. Let $n_x = 64$ and $N_x = 4,096$. The ground truth x_t is shown in Figure 9.2. The $n_p = 64$ pencils of a single exposure are shown in Figure 9.4. The $n_a = 18$ different exposures are shown in Figure 9.5, where each "cone" corresponds to an exposure. This results in $n_d = n_a n_p = 1,152$. The forward model is

$$d = Ax + e = y + e \tag{9.6}$$

where $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$ and A is explicitly constructed. Note that the forward operator A is a matrix in $\mathbb{R}^{n_d \times N_x} = \mathbb{R}^{1,152 \times 4,096}$ i.e. this is an underdetermined problem. The noiseless *sinogram* corresponding to x_t is shown in Figure 9.3. Each column of the sinogram is a 1D "image" corresponding to an exposure. The leftmost column of Figure 9.3 corresponds to the exposure shown in Figure 9.4.

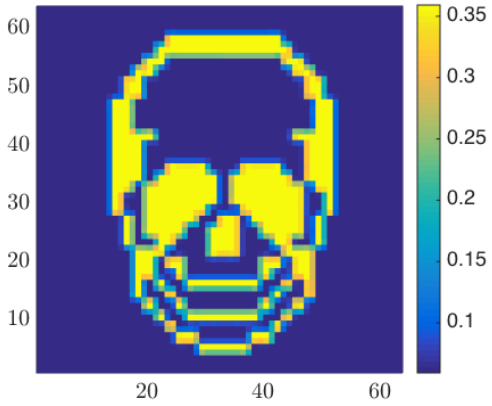


Figure 9.2: Ground truth x_t

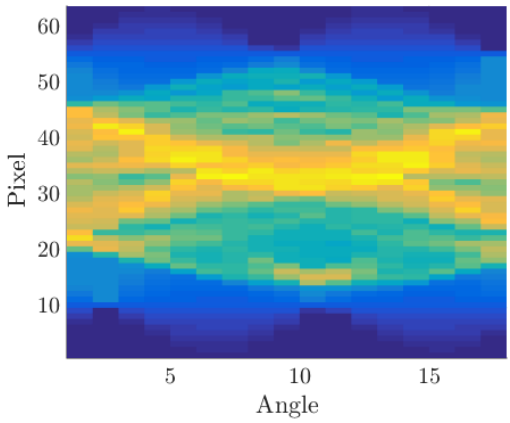


Figure 9.3: Sinogram of y_t

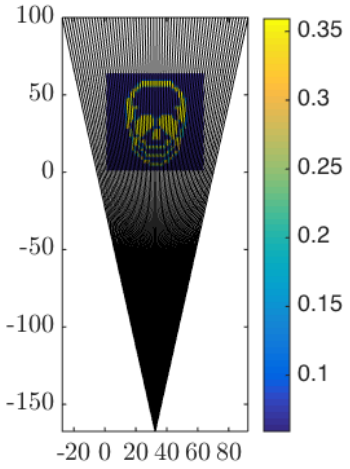


Figure 9.4: X-ray pencils

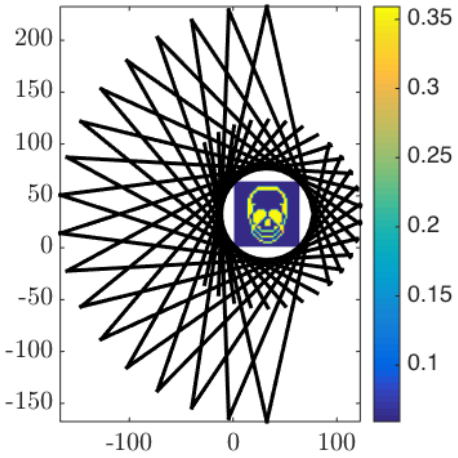


Figure 9.5: Exposure angles

Each row of A corresponds to a pencil. Each pencil passes through approximately $3n_x$ of N_x pixels in the object, so $A(j,:) \in \mathbb{R}^{1\times N_x}$ has approximately $3n_x$ nonzero entries i.e. A is sparse. The first 64 rows of A are shown in Figure 9.6, corresponding to the first exposure as shown in Figure 9.4.

Because the dimensions of the problem in this section are relatively small, I can approach the inverse problem in a straightforward manner. A Gaussian smoothness prior $x \sim \mathcal{N}(0_{n_x,1}, \Gamma_x)$ is constructed with the smoothness covariance function of Section 5.1.1. The factor L_x is computed by the Cholesky algorithm. $\tilde{L}_x = L_x^{-1}$ is computed by back substitu-

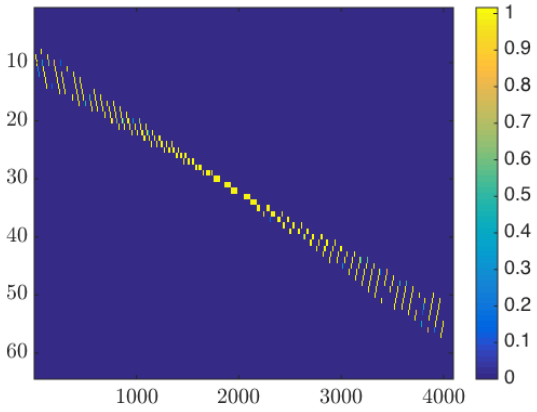


Figure 9.6: Matrix $A(1:64,:)$ for a single exposure with 64 pencils of a 64×64 pixel object

tion. This entire process took 7 minutes on a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB hard drive. The MAP estimate

$$x_{\text{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - Ax) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{9.7}$$

$$= \begin{pmatrix} \frac{1}{\sigma_e} A \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \frac{1}{\sigma_e} d_t \\ 0_{n_x,1} \end{pmatrix} \tag{9.8}$$

$$= \hat{A}^{\dagger} \hat{d}_t \tag{9.9}$$

was found as $\hat{A} \setminus \hat{d}$ in MATLAB, where $\hat{A}$ and $\hat{d}$ were formed explicitly. Computing $\hat{A} \setminus \hat{d}$ took 112 seconds.

The ground truth x_t of Figure 9.2 is reproduced in Figure 9.7. The forward evaluation $y = Ax_t$ is visualised as a sinogram in Figure 9.8. The data $d_t = Ax_t + e_t$ is shown in Figure 9.10, where $\sigma_e \approx 0.01\text{max}(y)$. The corresponding MAP estimate is shown in Figure 9.9. Note that an inverse crime is committed when the MAP estimate of this section is computed.

9.2 A Practical CT Implementation

This section will consider a higher dimensional case of the sparse full angle CT problem. The straightforward approach of Section 9.1 is not practical for this case with the computational resources available. This section makes use of the proposed methods of Chapters 6 and 7.

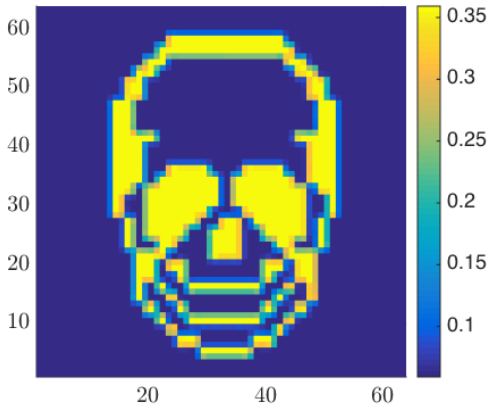


Figure 9.7: Object x_t

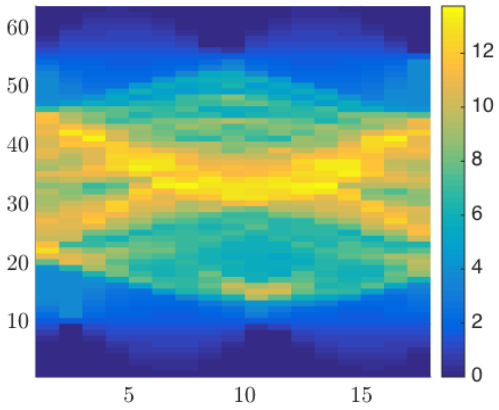


Figure 9.8: Noiseless sinogram of $y_t = Ax_t$

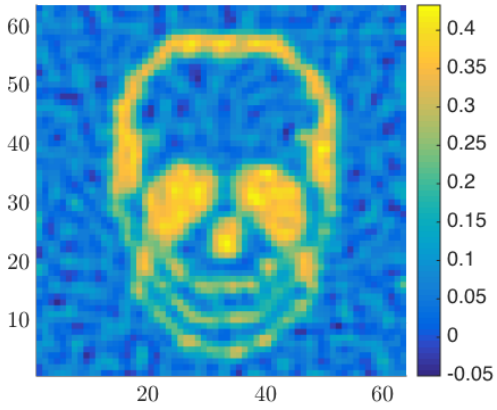


Figure 9.9: MAP estimate

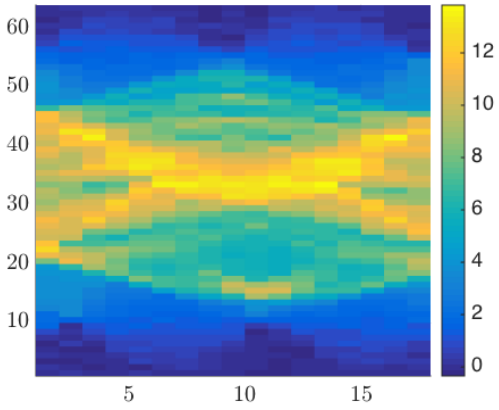


Figure 9.10: Measured sinogram $d_t = Ax_t + e_t$

Let $\Omega = [1, 512] \times [1, 512]$. Let

$$\bar{t} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ \vdots & \vdots \\ 512 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 512 & 512 \end{pmatrix} \in \mathbb{R}^{262,144 \times 2} \tag{9.10}$$

be a spatial discretisation i.e. $\Delta_{\bar{t}} = 1$. Let $\bar{s}(j) = \mathbf{s}(\bar{t}(j,:))$ be the finite dimensional approximation to the absorption coefficient distribution in Ω. Let $n_{\bar{s}} = 512$ and $N_{\bar{s}} = 262,144$. Let the region of interest be the front and central segment $\Omega_x = [1, 256] \times [129, 384]$ i.e.

$$\bar{t}_x = \begin{pmatrix} 1 & 129 \\ 2 & 129 \\ 3 & 129 \\ \vdots & \vdots \\ 256 & 129 \\ 1 & 130 \\ \vdots & \vdots \\ 256 & 384 \end{pmatrix} \in \mathbb{R}^{65,536 \times 2} \tag{9.11}$$

is a spatial discretisation of Ω_x. Let $\bar{x}(j) = \mathbf{s}(\bar{t}_x(j,:))$. Let $n_{\bar{x}} = 256$ and $N_{\bar{x}} = 65,536$. Let $\bar{t}_z = \bar{t} \setminus \bar{t}_x$ and $\bar{z}(j) = \mathbf{s}(\bar{t}_z(j,:))$.

Let

$$t_x = \begin{pmatrix} 1 & 129 \\ 3 & 129 \\ 5 & 129 \\ \vdots & \vdots \\ 256 & 129 \\ 1 & 131 \\ \vdots & \vdots \\ 256 & 384 \end{pmatrix} \in \mathbb{R}^{16,384 \times 2} \tag{9.12}$$

be the "sufficient resolution" discretisation of Ω_x i.e. $\Delta_t = 2$. Let $x(j) \approx \mathbf{s}(t_x(j,:))$. Let $n_x = 128$ and $N_x = 16,384$. Let

$$P_{x,\bar{x}} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 0 & 0 & 0_{1,n_{\bar{x}}} & 1 & 1 & 0 & 0 & 0_{1,N_{\bar{x}}-n_{\bar{x}}-8} \\ 0 & 0 & 1 & 1 & 0_{1,n_{\bar{x}}} & 0 & 0 & 1 & 1 & 0_{1,N_{\bar{x}}-n_{\bar{x}}-8} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \in \mathbb{R}^{N_x \times N_{\bar{x}}} \tag{9.13}$$

be such that $x = P_{x,\bar{x}}\bar{x}$, $P_{\bar{x},\bar{s}} \in \mathbb{R}^{N_{\bar{x}} \times N_{\bar{s}}}$ be such that $\bar{x} = P_{\bar{x},\bar{s}}\bar{s}$ and $P_{x,\bar{s}} = P_{x,\bar{x}}P_{\bar{x},\bar{s}} \in \mathbb{R}^{N_x \times N_{\bar{s}}}$. Let $P_{\bar{x},x} = P^{\dagger}_{x,\bar{x}} = 4P^T_{x,\bar{x}}$. Let $P_{\bar{s},x} = P^{\dagger}_{x,\bar{s}} = 4P^T_{\bar{x},\bar{s}}P^T_{x,\bar{x}}$.

Let there be $n_p = 500$ pencils at each exposure. Let $n_a = 18$ exposures be taken, at the same angles as in the earlier section. The accurate forward model is

$$d = \bar{y} + e = \bar{A}\bar{s} + e = \bar{A}(\bar{x}, \bar{z}) + e \tag{9.14}$$

where $\bar{A} \in \mathbb{R}^{9{,}000 \times 262{,}144}$. Let the simple model be

$$d = y + \varepsilon + e \approx y + e = Ax + e = \bar{A}P_{\bar{s},x}x + e \tag{9.15}$$

which allows us to construct $A \in \mathbb{R}^{9{,}000 \times 16{,}384}$ without recomputing the pencil paths.

Let

$$\bar{\bar{t}} = \begin{pmatrix} 1 & 1 \\ 1.5 & 1 \\ 2 & 1 \\ \vdots & \vdots \\ 512 & 1 \\ 1 & 1.5 \\ \vdots & \vdots \\ 512 & 512 \end{pmatrix} \in \mathbb{R}^{1{,}048{,}576 \times 2} \tag{9.16}$$

be a spatial discretisation i.e. $\Delta_{\bar{\bar{t}}} = 0.5$. Let $\bar{\bar{s}}(j) = \mathbf{s}(\bar{\bar{t}}(j,:))$. Let

$$d = \bar{\bar{y}} + e = \bar{\bar{A}}\bar{\bar{s}} + e \tag{9.17}$$

be the very accurate model used for simulating data. The test image $\bar{s}_t$ used in this problem is a 512×512 image, so $\bar{\bar{s}}_t \in \mathbb{R}^{1{,}048{,}576}$ was constructed by upsampling and convolving with a narrow kernel.

This chapter uses a real world x-ray CT reconstruction of a human head as the ground truth $\bar{s}_t$, shown in Figure 9.11. This image was taken from wikimedia commons, uploaded by Mikael Häggström of the Department of Radiology at Uppsala University Hospital under an open license. The exposure "cones" are shown in Figure 9.12. The ground truth in the region of interest $\bar{x}_t$ is shown in Figure 9.13. The ground truth in the region of interest at the desired discretisation is shown in Figure 9.14.

9.2.1 Full Domain Inversion

In this subsection, the MAP estimate of $\bar{s}_t$ is computed with the accurate model $\bar{A}$. The model $\bar{A}$ is also used when computing d_t, so an inverse crime is committed during the estimation. The purpose of this subsection is to demonstrate the highest reasonable quality for MAP estimates in this problem. This problem is sufficiently high dimensional that some of the methods proposed in this thesis need to be used for estimation to be computationally reasonable. Note that BAE is not used in this subsection. The analysis in this Subsection is comparatively brief, as the aim is primarily to provide a point of comparison for the estimates in Section 9.3.

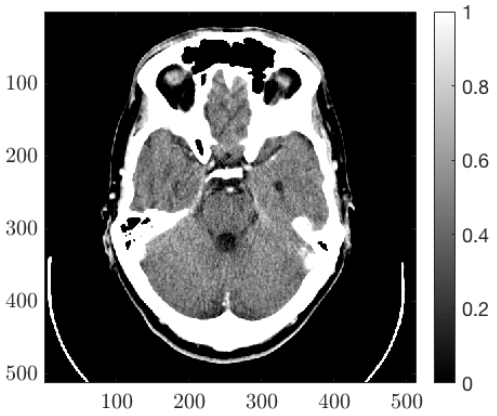


Figure 9.11: Ground truth $\bar{s}_t$ on Ω

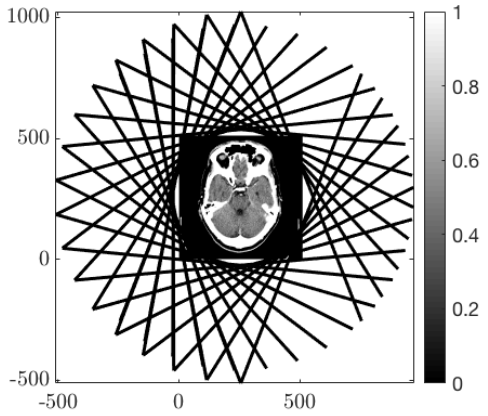


Figure 9.12: Exposures being taken

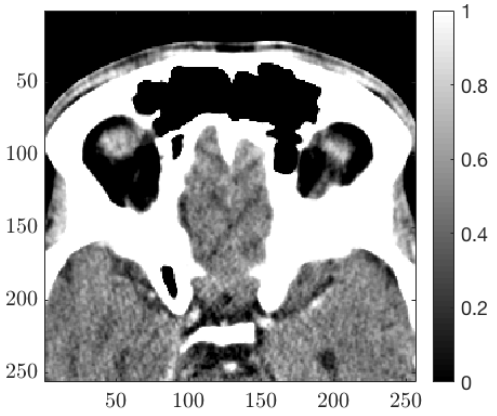


Figure 9.13: Ground truth $\bar{x}_t$ on Ω_x

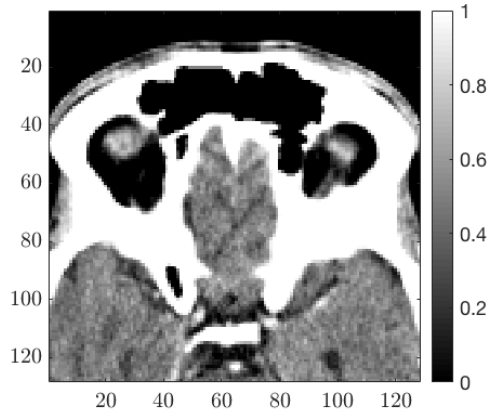


Figure 9.14: Ground truth x_t on Ω at coarse discretisation

Let $\bar{s}$ be normally distributed with a smoothness prior. The filter $L_{\bar{s}}$ is constructed directly with Fourier transforms as in Chapter 8. Draws are computed as

$$\bar{s}_j = \mu_{\bar{s}} + L_{\bar{s}}(w_j) \tag{9.18}$$

$$= \mu_{\bar{s}} + \mathcal{F}^{-1}\left(k_{f,\bar{s}} \odot \mathcal{F}(w_j)\right) \tag{9.19}$$

where $\mathcal{F}(k_{\bar{s}}) = k_{f,\bar{s}} \in \mathbb{R}^{N_{\bar{s}}}$ is the Fourier transform of a Gaussian kernel. A kernel of width 2 and amplitude 0.7 was found to produce draws with features on the scale desired. The innova-

tion filter is formed as

$$w_j = \tilde{L}_{\bar{s}}(\bar{s}_j - \mu_{\bar{s}}) = \mathcal{F}^{-1}\left(\tilde{k}_{f,\bar{s}} \odot \mathcal{F}(\bar{s}_j - \mu_{\bar{s}})\right) \tag{9.20}$$

where $\tilde{k}_{f,\bar{s}}(j) = \frac{1}{k_{f,\bar{s}}(j)}$. Note that $k_{f,\bar{s}}$ can be computed in $\mathcal{O}\left(N_{\bar{s}}\log(N_{\bar{s}})\right)$ flops, computing $\tilde{k}_{f,\bar{s}}$ takes $N_{\bar{s}}$ flops, and $\tilde{k}_{f,\bar{s}} \in \mathbb{R}^{N_{\bar{s}}}$ and $k_{f,\bar{s}} \in \mathbb{R}^{N_{\bar{s}}}$ occupy around 2MB of memory. In contrast, $\Gamma_{\bar{s}} \in \mathbb{R}^{N_{\bar{s}} \times N_{\bar{s}}}$ occupies around 540 GB of memory.

The MAP estimate found with the model $d = \bar{A}\bar{s} + e$ and the above prior is

$$\bar{s}_{\text{MAP}} = \min_{\bar{s}} \left\{ \left\| \tilde{L}_e(d - \bar{A}\bar{s} - \mu_d) \right\|_2^2 + \left\| \tilde{L}_{\bar{s}}(\bar{s} - \mu_{\bar{s}}) \right\|_2^2 \right\} \tag{9.21}$$

$$= \begin{pmatrix} \tilde{L}_e\bar{A} \\ \tilde{L}_{\bar{s}} \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_e(d - \mu_d) \\ \tilde{L}_{\bar{s}}\mu_{\bar{s}} \end{pmatrix} \tag{9.22}$$

$$= \hat{A}_{\bar{s}}^{\dagger}\hat{d}_{\bar{s}}. \tag{9.23}$$

Recall that computations in this thesis are performed on a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB hard drive. The matrix $\hat{A}_{\bar{s}}$ stored as full would not even fit on the hard drive. The MAP estimate is instead computed by Landweber iterations, similar to Chapter 8. The iterates are computed as

$$\bar{s}_{k+1} = \bar{s}_k + \beta(\hat{A}_{\bar{s}}^T\hat{d}_{\bar{s}} - \hat{A}_{\bar{s}}^T\hat{A}_{\bar{s}}\bar{s}_k) \tag{9.24}$$

where

$$\hat{A}_{\bar{s}}^T\hat{d}_{\bar{s}} = \begin{pmatrix} \bar{A}^T\tilde{L}_e^T & \tilde{L}_{\bar{s}}^T \end{pmatrix} \begin{pmatrix} \tilde{L}_e(d - \mu_d) \\ \tilde{L}_{\bar{s}}\mu_{\bar{s}} \end{pmatrix} \tag{9.25}$$

$$= \bar{A}^T\Gamma_e^{-1}(d - \mu_d) + \Gamma_{\bar{s}}^{-1}\mu_{\bar{s}} \tag{9.26}$$

$$= \frac{1}{\sigma_e^2}\bar{A}^T(d - \mu_d) + \Gamma_{\bar{s}}^{-1}\mu_{\bar{s}} \tag{9.27}$$

and

$$\hat{A}_{\bar{s}}^T\hat{A}_{\bar{s}}\bar{s} = \begin{pmatrix} \bar{A}^T\tilde{L}_e^T & \tilde{L}_{\bar{s}}^T \end{pmatrix} \begin{pmatrix} \tilde{L}_e\bar{A} \\ \tilde{L}_{\bar{s}} \end{pmatrix} \bar{s} \tag{9.28}$$

$$= \bar{A}^T\Gamma_e^{-1}\bar{A}\bar{s} + \Gamma_{\bar{s}}^{-1}\bar{s} \tag{9.29}$$

$$= \frac{1}{\sigma_e^2}\bar{A}^T\bar{A}\bar{s} + \Gamma_{\bar{s}}^{-1}\bar{s} \tag{9.30}$$

where $\Gamma_{\bar{s}}^{-1}\bar{s}$ is evaluated by the fast Fourier transform algorithm. The sparsity of $\bar{A}$ and the FFT form of $\Gamma_{\bar{s}}$ allows Landweber iterations to be computed rapidly, with each iterate requiring $\mathcal{O}\left(N_{\bar{s}}\log(N_{\bar{s}}) + n_d n_{\bar{s}}\right)$ flops due to the FFT evaluation of $\Gamma_{\bar{s}}^{-1}\bar{s}$ and the sparsity of $\bar{A}$.

The stepsize β was chosen by the same method used in Chapter 8. I computed several $b_j = \frac{\|\hat{A}^T\hat{A}w_j\|}{\|w_j\|}$ where w_j is a white noise sample. I found that $\max_j\{b_j\} \approx 100$, so took $\beta = \frac{1}{1{,}000}$. I computed 100 Landweber iterates, however the iterates were found to change little after iterate 15. For online estimation of the MAP, I chose to take 20 iterates.

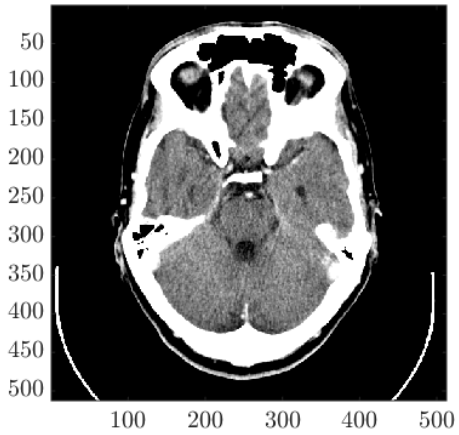


Figure 9.15: Ground truth $\bar{s}_t$

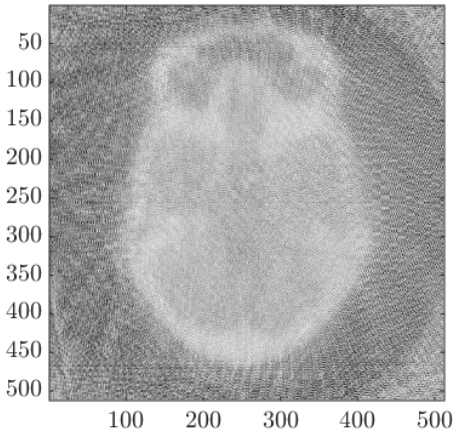


Figure 9.16: Solution $\bar{s}_{MAP}$

The ground truth $\bar{s}_t$ is reproduced in Figure 9.15. The MAP estimate computed in 1.82 seconds with 20 Landweber iterations is shown in Figure 9.16. While the ground truth can be partially discerned from the MAP, there appears to be "nonsmooth" terms. This appears to be a consequence of pencils being relatively sparse relative to the density of pixels.

Smoothness can be more explicitly enforced in the estimation by "re-regularising" each Landweber iterate $\bar{s}_k$. The iteration becomes

$$\hat{s}_{j+1} = L_{\bar{s}}\bar{s}_{j+1} = \bar{s}_j + \beta(\hat{A}_{\bar{s}}^T\hat{d}_{\bar{s}} - \hat{A}_{\bar{s}}^T\hat{A}_{\bar{s}}\bar{s}_j) \tag{9.31}$$

with the computational time of computing 20 iterates rising to 2.1 seconds. The ground truth is reproduced in Figure 9.17. The MAP estimate found with the above iteration scheme is shown in Figure 9.18. The streaks in the image are typical of sparse angle CT [5, 10]. Note that many of the features in the ground truth can be resolved in the MAP estimate.

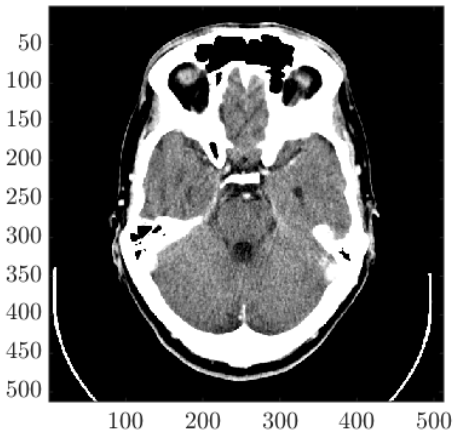


Figure 9.17: Ground truth $\bar{s}_t$

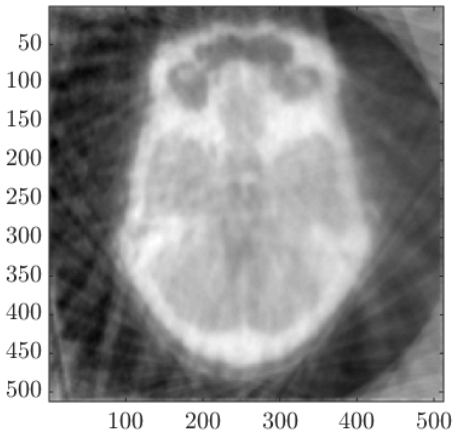


Figure 9.18: Solution $\bar{s}_{MAP}$, re-regularised

9.3 Local Tomography

This section considers the local tomography problem, in which only a a subset of the object density distribution is estimated [124, 128, 129, 130]. The advantage of local tomography is that x-ray exposure can be better restricted to the region of interest. However, the exposures used in this section are the same as in the non-local tomography problem of Subsection 9.2.1. This is done to maintain a level of consistency when comparing the quality of the estimates. Note that the local sparse angle tomography problem is even more ill-posed than the non-local sparse CT problem [10]. This section considers estimating $x_t \in \mathbb{R}^{N_x}$ as shown in Figure 9.14 without estimating all of $\bar{s}_t \in \mathbb{R}^{N_{\bar{s}}}$. Note that a coarse discretisation is also used. This is done to further reduce computational cost at the online stage and further demonstrate the effectiveness of the BAE approach.

This section makes use of samples $\bar{s}_j$, $x_j = P_{x,\bar{s}}\bar{s}_j$, $\bar{y}_j = \bar{A}\bar{s}_j$ and $y_j = Ax_j$ where $\bar{s}_j$ is drawn using Fourier transforms as described in the previous section. Samples are stored in terms of QR decompositions as described in Section 6.4.2 e.g. samples of x are used to form $Q_x \in \mathbb{R}^{N_x \times r_x}$ and $R_x \in \mathbb{R}^{r_x \times m}$ where r_x is the "effective rank" of the sample covariance of x and m is the number of samples. This is similar to the approach of Section 8.2.

9.3.1 Conditional Mean with BAE

Let us first consider the conditional mean

$$\mu_{x|d} = \mu_x + \Gamma_{xd}\Gamma_d^{-1}(d - \mu_d) \tag{9.32}$$

which we approximate as

$$\hat{\mu}_{x|d} = \mu_x + \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}(d - \hat{\mu}_d) \tag{9.33}$$

$$= \mu_x + \frac{1}{m}Q_x R_x R_{\bar{y}}^T Q_{\bar{y}}^T \left(\Gamma_e + \frac{1}{m}Q_{\bar{y}} R_{\bar{y}} R_{\bar{y}}^T Q_{\bar{y}}^T\right)^{-1}(d - \hat{\mu}_d) \tag{9.34}$$

$$= \mu_x + Q_x M_{x\bar{y}} Q_y^T \left(\Gamma_e + Q_{\bar{y}} M_{\bar{y}} Q_{\bar{y}}^T\right)^{-1}(d - \hat{\mu}_d) \tag{9.35}$$

noting that samples of x and $\bar{y} = \bar{A}\bar{s}$ are used. Note that approximate model A is not used in the construction of this estimate, and the approximation error does not explicitly appear. Construction of the recursive QR sample conditional mean above is covered in greater detail in Section 7.1. Noting that $\Gamma_e = \sigma_e^2 I_{n_d}$, $\hat{\Gamma}_d^{-1}$ can be expanded as

$$\hat{\Gamma}_d^{-1} = \left(\Gamma_e + Q_{\bar{y}} M_{\bar{y}} Q_{\bar{y}}^T\right)^{-1} \tag{9.36}$$

$$= \left(\sigma_e^2 I_{n_d} + Q_{\bar{y}} M_{\bar{y}} Q_{\bar{y}}^T\right)^{-1} \tag{9.37}$$

$$= \frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T \tag{9.38}$$

so the sample conditional mean estimate can be expressed as

$$\hat{\mu}_{x|d} = \mu_x + Q_x M_{x\bar{y}} Q_y^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T\right)(d - \hat{\mu}_d) \tag{9.39}$$

where care should be taken in the order of operations to avoid forming large matrices e.g. $Q_x M_{x\bar{y}} Q_y^T$ should not be explicitly formed.

Each of the m samples used in the construction of the above sample conditional mean estimate requires $\mathcal{O}\left(N_{\bar{s}} \log(N_{\bar{s}}) + n_d n_{\bar{s}}\right)$ flops to compute. The matrices $M_{\bar{y}} \in \mathbb{R}^{r_{\bar{y}} \times r_{\bar{y}}}$ and $(M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}}) \in \mathbb{R}^{r_{\bar{y}} \times r_{\bar{y}}}$ need to be inverted. This can be done in parallel with samples being computed as described in Section 6.4.2. Otherwise, the inverses can be computed with $\mathcal{O}(r_{\bar{y}}^3)$ additional flops. At the online stage, computing $\hat{\mu}_{x|d}$ requires $\mathcal{O}\left(n_x r_x + n_d r_{\bar{y}} + r_{\bar{y}}(r_{\bar{y}} + r_x)\right)$ flops and numbers stored in memory.

The sample conditional mean as expressed in Equation (9.39) was computed with $m = 5,000$ samples in 73 minutes. A relatively low tolerance $\tau = \frac{R(1,1)}{10,000}$ was used, resulting in $r_{\bar{y}} = n_d = 3,600$ and $r_x = m = 5,000$. Further samples were not computed as the offline computational cost is already unacceptably high. Note that the prior in this problem has correlation length $l \approx 2$, and the discretisation is $\Delta_t = 2$. The prior covariance Γ_x is therefore

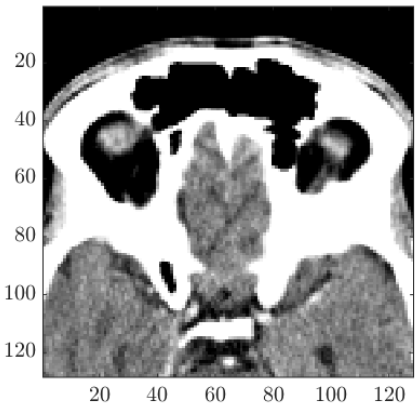


Figure 9.19: Ground truth x_t

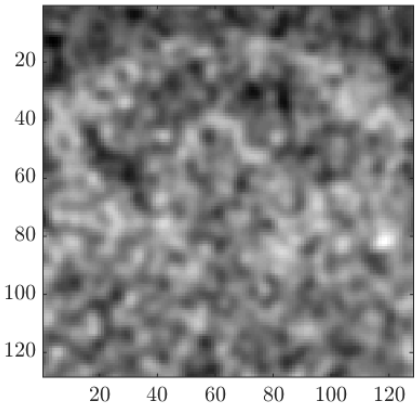


Figure 9.20: Sample conditional mean estimate $\hat{\mu}_{x|d=d_t}$

effectively diagonal, as described in Section 5.1.1. The prior has high effective rank, so sample entities such as Q_x and $M_{x\bar{y}}$ may need $\mathcal{O}(N_x)$ samples to be estimated effectively.

The ground truth x_t is reproduced in Figure 9.19. The sample conditional mean estimate with data simulated by $\bar{\bar{A}}$ and 1% additive white noise is shown in Figure 9.20. The sample conditional mean estimate is quite poor, although the offline stage does compute in only 1.5 seconds. The high effective rank of Γ_x means that the sample conditional mean $\hat{\mu}_{x|d}$ requires $\mathcal{O}(N_x)$ samples be used if it is to be representative of the true conditional mean $\mu_{x|d}$. Given that $N_x = 16,384$ and $m = 5,000$ samples was already relatively computationally costly to compute, a sample conditional mean type approach seems badly suited to this problem.

9.3.2 MAP with BAE

The sample entities used in the construction of a sample mean estimate as described in Section 7.1 were found to be of too high rank to be estimated from an acceptably small number of samples. This is similar to the case of Chapter 8. This subsection implements an iterative MAP estimate with BAE as described in Section 7.4 and applied in Section 8.2.

In this section, the sparse approximate forward model $A = \bar{A}P_{\bar{s},x}$ is used with BAE. The model used for estimating x_t is

$$d = Ax + \hat{\mu}_{\varepsilon|x} + e \tag{9.40}$$

leading to the MAP estimate

$$x_{MAP} = \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^{\dagger} \begin{pmatrix} \tilde{L}_{\nu|x}(d_t + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x - \mu_d - \hat{\mu}_\varepsilon) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{9.41}$$

$$= \hat{A}^{\dagger}\hat{d}_t \tag{9.42}$$

which is found by Landweber iterations

$$x_{k+1} = x_k + \beta(\hat{A}^T\hat{d}_t - \hat{A}^T\hat{A}x_k) \tag{9.43}$$

where

$$\hat{A}^T\hat{d} = \begin{pmatrix} (A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\tilde{L}_{\nu|x}^T & \tilde{L}_x^T \end{pmatrix} \begin{pmatrix} \tilde{L}_{\nu|x}(d_t + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x - \mu_d - \hat{\mu}_\varepsilon) \\ \tilde{L}_x\mu_x \end{pmatrix} \tag{9.44}$$

$$= (A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\Gamma_{\nu|x}^{-1}(d_t + \mu_\omega) + \Gamma_x^{-1}\mu_x \tag{9.45}$$

where

$$\mu_\omega = \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x - \mu_d - \hat{\mu}_\varepsilon \tag{9.46}$$

with

$$\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1} = Q_\varepsilon R_\varepsilon R_x^T(R_x R_x^T)^{-1}Q_x^T \tag{9.47}$$

and

$$\Gamma_{\nu|x}^{-1} = (\Gamma_e + \hat{\Gamma}_{\varepsilon|x})^{-1} \tag{9.48}$$

$$= (\Gamma_e + Q_\varepsilon M_{\varepsilon|x}Q_\varepsilon^T)^{-1} \tag{9.49}$$

$$= \Gamma_e^{-1} - \Gamma_e^{-1}Q_\varepsilon(M_{\varepsilon|x}^{-1} + Q_\varepsilon^T\Gamma_e^{-1}Q_\varepsilon)^{-1}Q_\varepsilon^T\Gamma_e^{-1} \tag{9.50}$$

$$= \frac{1}{\sigma_e^2}I_{n_d} - \frac{1}{\sigma_e^4}Q_\varepsilon(M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2}I_{r_\varepsilon})^{-1}Q_\varepsilon^T \tag{9.51}$$

so

$$\hat{A}^T\hat{d}_t = (A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\Gamma_{\nu|x}^{-1}(d_t + \mu_\omega) + \Gamma_x^{-1}\mu_x \tag{9.52}$$

$$= (A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\left(\frac{1}{\sigma_e^2}I_{n_d} - \frac{1}{\sigma_e^4}Q_\varepsilon(M_{\varepsilon|x}^{-1} + \frac{1}{\sigma_e^2}I_{r_\varepsilon})^{-1}Q_\varepsilon^T\right)(d_t + \mu_\omega) + \Gamma_x^{-1}\mu_x \tag{9.53}$$

also

$$\hat{A}^T\hat{A} = \begin{pmatrix}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\tilde{L}_{\nu|x}^T & \tilde{L}_x^T\end{pmatrix}\begin{pmatrix}\tilde{L}_{\nu|x}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})\\ \tilde{L}_x\end{pmatrix} \tag{9.54}$$

$$= (A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\Gamma_{\nu|x}^{-1}(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) + \Gamma_x^{-1} \tag{9.55}$$

$$= (A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1})^T\left(\frac{1}{\sigma_e^2}I_{n_d} - \frac{1}{\sigma_e^4}Q_\varepsilon(M_{\varepsilon|x}^{-1}+\frac{1}{\sigma_e^2}I_{r_\varepsilon})^{-1}Q_\varepsilon^T\right)(A+\hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) + \Gamma_x^{-1} \tag{9.56}$$

which can be computed efficiently with careful application of the order of operations. Further details of this approach can be found in Sections 7.4 and 8.2.

The same stepsize $\beta = \frac{1}{1{,}000}$ from Section 9.2.1 is used for the Landweber iteration. However, the iterates were found to effectively converge after 8 iterates, so in this section the online stage is computed with 10 Landweber iterates to further reduce computational cost.

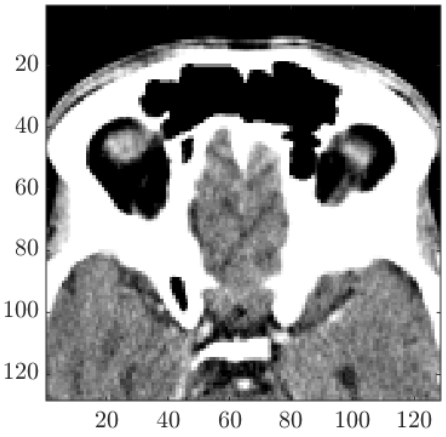


Figure 9.21: Ground truth x_t

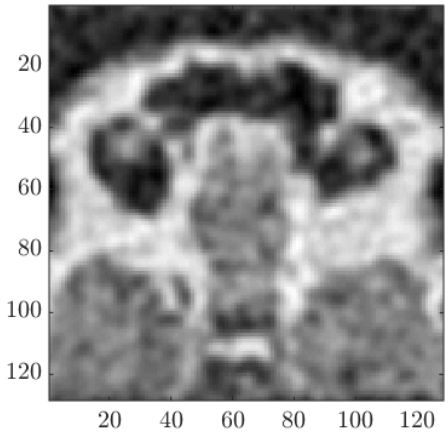


Figure 9.22: x_{MAP} with BAE

In this section, evaluations of $\Gamma_x^{-1}x$ is formed from samples. Note that draws of x are computed as $x_j = P_{x,\bar{s}}\bar{s}_j = P_{x,\bar{s}}\mathcal{F}^{-1}(\mathcal{F}(k_s)\odot\mathcal{F}(w_j))$. The whitening filter $\tilde{k}$ on x is constructed such that $w_j \approx \mathcal{F}^{-1}(\mathcal{F}(\tilde{k})\odot\mathcal{F}(\bar{s}_j))$ from the samples. Forming the filter in this way is similar to the deconvolution filter described in Section 6.8. Constructing the filter in this way ensures discretisation invariance and allows the whitening filter to be implemented with the FFT algorithm.

The offline stage of computing $m = 200$ samples and forming the quantities $\hat{A}^T\hat{A}$ and $\hat{A}^T\hat{d}$ as above took 24.2 seconds on a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB hard drive. The data d_t was simulated as $d_t = \bar{\bar{A}}\bar{s}_t + e_t$ as described in the

introduction to Section 9.2 to avoid committing an inverse crime. Note that the ground truth $\bar{s}_t$ comes from real data, but the data d_t used in the estimation is synthetic. The online stage of estimating x_{MAP} from data d_t with 10 Landweber iterations took 0.27 seconds. The ground truth x_t is reproduced in Figure 9.21. The MAP estimate x_{MAP} is shown in Figure 9.21. Note that many of the features can be resolved. I am pleased with this result, especially considering the offline and online computational cost, and the quality of estimates expected from sparse local tomography.

9.3.3 Uncertainty Quantification by Posterior Residuals

In this section, uncertainty in the estimate x_{MAP} is quantified by the posterior residual approach described in Section 6.9 and applied in Section 8.3.3. This approach was chosen as MAP estimates can be computed rapidly, and the memory requirements are low. Residuals $\zeta_j = x_{\mathrm{MAP,j}} - x_j$ are computed from $\hat{m} = 200$ samples, adding 63.5 seconds to the offline computation stage. The estimate variance is estimated as

$$\hat{\sigma}^2 = \frac{1}{\hat{m}} \sum_{j=1}^{\hat{m}} (\zeta_j - \hat{\mu}_\zeta)^2 \tag{9.57}$$

from which posterior error intervals are formed. Recall that $\hat{\mu}_\zeta \approx 0_{n_x,1}$ when m and $\hat{m}$ are large enough. The residual mean $\hat{\mu}_\zeta$ is shown in Figure 9.23. Note that $\hat{\mu}_\zeta \approx 0$, implying that m and $\hat{m}$ are sufficiently large. The MAP estimate with $3\hat{\sigma}$ posterior error intervals along the 20'th column of pixels is shown in Figure 9.24. Note that the MAP estimate captures major features of the ground truth, and the posterior error intervals tightly contain the ground truth. I consider this to be a good result.

9.4 Discussion

This chapter primarily considers local tomography, estimating x_t from d_t. We found that the effective rank of the prior covariance Γ_x was too high for approximations such as

$$\hat{\mu}_{x|d=d_t} = \hat{\mu}_x + Q_x M_{xy} Q_y^T \left(\Gamma_e^{-1} + \Gamma_e^{-1} Q_y \big(M_y^{-1} + Q_y^T \Gamma_e^{-1} Q_y \big)^{-1} Q_y^T \Gamma_e^{-1} \right) (d_t - \hat{\mu}_d) \tag{9.58}$$

with approximate posterior covariance

$$\hat{\Gamma}_{x|d} = Q_x \left(M_x - M_{xy} Q_y^T (\Gamma_e + Q_{y,\varepsilon} M_{y,\varepsilon|x} Q_{y,\varepsilon}^T)^{-1} Q_y M_{xy}^T \right) Q_x^T \tag{9.59}$$

to be computed reliably from a reasonable number of samples. The estimate $\hat{\mu}_{x|d=d_t}$ found from $5,000$ samples is shown in Figure 9.20, and is clearly lacking.

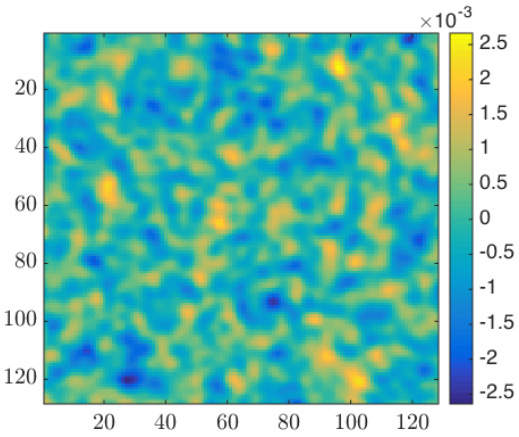


Figure 9.23: Posterior residual mean $\hat{\mu}_\zeta$

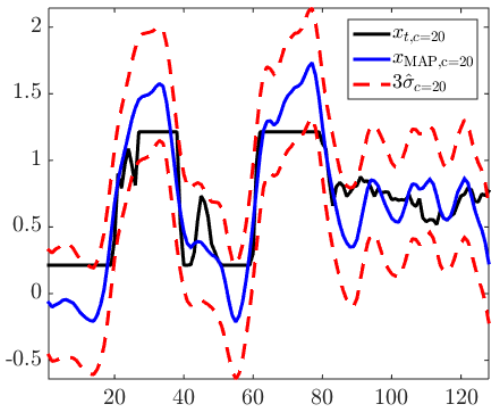


Figure 9.24: MAP estimate, posterior error interval and ground truth along pixel column 20

The problem of high rank entities is similar to that of Chapter 8, so a similar approach was taken. The MAP

$$x_{\mathrm{MAP}} = \begin{pmatrix} \tilde{L}_{\nu|x}(A + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}) \\ \tilde{L}_x \end{pmatrix}^\dagger \begin{pmatrix} \tilde{L}_{\nu|x}(d + \hat{\Gamma}_{\varepsilon x}\hat{\Gamma}_x^{-1}\mu_x - \hat{\mu}_\varepsilon) \\ \tilde{L}_x \mu_x \end{pmatrix} \tag{9.60}$$

$$= \hat{A}^\dagger \hat{d} \tag{9.61}$$

was estimated by Landweber iterations

$$x_{k+1} = x_k + \beta(\hat{A}^T\hat{d} - \hat{A}^T\hat{A}x_k) \tag{9.62}$$

starting at $x_0 = \mu_x$. The BAE sample entities e.g. $\hat{\Gamma}_{\nu|x}$ were computed from $m = 200$ samples with recursive QR in 24.2 seconds on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive. An additional $\hat{m} = 50$ samples $\zeta_j = x_{\mathrm{MAP,j}} - x_j$ were computed, and the diagonal posterior residual approximation

$$\hat{\sigma}_2 \approx \frac{1}{m}\sum_{j=1}^{m}(\zeta_j - \hat{\mu}_\zeta) \odot (\zeta_j - \hat{\mu}_\zeta) \tag{9.63}$$

was used. This added 63.5 seconds of computation to the offline stage. The MAP estimate computes in 0.27 seconds at the online stage, and is shown in Figure 9.22. The MAP estimate with posterior error estimates down the 20'th column of pixels is shown in Figure 9.24. The major features of the ground truth can be discerned from the MAP estimate and the posterior

error estimates support the ground truth. Considering the offline and online computational cost of the local CT implementation in this chapter alongside the inherent limitations of sparse local CT, I consider this chapter a powerful demonstration of the effectiveness of the methods proposed in this thesis.

Chapter 10

Simplified Conductivity Imaging

This chapter considers a simplified problem related to Electrical Impedance Tomography (EIT). The purpose of this chapter is to demonstrate application of some of the methods of this thesis to a nonlinear inverse problem. More details on EIT in the context of inverse problems can be found in [4, 5]. The problem considered in this chapter is discussed in greater detail in [131], the author of which assisted in constructing the forward problem of this chapter.

Consider the Poisson equation

$$-\nabla \cdot \Big(\boldsymbol{\psi}(\mathbf{t}) \nabla \mathbf{u}(\mathbf{t}) \Big) = 0 \qquad \text{in } \Omega \tag{10.1}$$

$$\mathbf{u}(\mathbf{t}) = g_D(\mathbf{t}) \qquad \text{on } \partial\Omega_D \tag{10.2}$$

$$\boldsymbol{\psi} \nabla \mathbf{u}(\mathbf{t}) \cdot \eta = g_N(\mathbf{t}) \qquad \text{on } \partial\Omega_N \tag{10.3}$$

where $\mathbf{t} \in \Omega = [0,1] \times [0,1]$ is a spatial variable, $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ is the boundary of Ω, $\partial\Omega_D$ is the Dirichlet boundary, $\partial\Omega_N$ is the Neumann boundary, $\eta = \eta(\mathbf{t})$ is the outward facing unit normal vector at $\mathbf{t}$, $g_D(\mathbf{t})$ is the Dirichlet boundary condition and $g_N(\mathbf{t})$ is the Neumann boundary condition. Recall that

$$\nabla \mathbf{u}(\mathbf{t}) = \begin{pmatrix} \frac{\partial \mathbf{u}(\mathbf{t})}{\partial \mathbf{t}(1)} \\ \frac{\partial \mathbf{u}(\mathbf{t})}{\partial \mathbf{t}(2)} \end{pmatrix} \tag{10.4}$$

is the gradient of $\mathbf{u}$ and

$$\nabla \cdot \Big(\boldsymbol{\psi}(\mathbf{t}) \nabla \mathbf{u}(\mathbf{t}) \Big) = \frac{\partial}{\partial \mathbf{t}(1)} \Big(\boldsymbol{\psi}(\mathbf{t}) \frac{\partial \mathbf{u}(\mathbf{t})}{\partial \mathbf{t}(1)} \Big) + \frac{\partial}{\partial \mathbf{t}(2)} \Big(\boldsymbol{\psi}(\mathbf{t}) \frac{\partial \mathbf{u}(\mathbf{t})}{\partial \mathbf{t}(2)} \Big) \tag{10.5}$$

is the divergence of $\boldsymbol{\psi}(\mathbf{t}) \nabla \mathbf{u}(\mathbf{t})$.

Let $t_N \in \mathbb{R}^{n_{d_N} \times 2}$ and $t_D \in \mathbb{R}^{n_{d_D} \times 2}$, where $t_N(j,:) \in \partial\Omega$ and $t_D(j,:) \in \partial\Omega$. Let

$$t_d = \begin{pmatrix} t_D \\ t_N \end{pmatrix} \in \mathbb{R}^{n_d \times 2} \tag{10.6}$$

where $n_d = n_{d_D} + n_{d_N}$. Let

$$\mathbf{u}(t_D) = \mathbf{y}_D \in \mathbb{R}^{n_{d_D}} \tag{10.7}$$

$$\boldsymbol{\psi}(t_N)\nabla\mathbf{u}(t_N) \cdot \eta = \mathbf{y}_N \in \mathbb{R}^{n_{d_N}} \tag{10.8}$$

$$\begin{pmatrix} \mathbf{y}_D \\ \mathbf{y}_N \end{pmatrix} = \mathbf{y} \in \mathbb{R}^{n_d} \tag{10.9}$$

where η is the outward facing unit vector normal to $\partial\Omega$ at $t_N(j,:)$. Let

$$d_D = \mathbf{y}_D + e_D \tag{10.10}$$

$$d_N = \mathbf{y}_N + e_N \tag{10.11}$$

$$d = \mathbf{y} + e = \begin{pmatrix} \mathbf{y}_D \\ \mathbf{y}_N \end{pmatrix} + \begin{pmatrix} e_D \\ e_N \end{pmatrix} \tag{10.12}$$

be the data i.e. boundary measurements of $\mathbf{u}$ or $\frac{\partial \mathbf{u}(t_N)}{\partial \eta}$ with additive noise. d_D is called the Dirichlet data and d_N is the Neumann data. The problem of estimating $\boldsymbol{\psi}$ from boundary measurements such as $\mathbf{y}$ is related to *Calderóns Problem* [132, 133].

In this chapter, $\boldsymbol{\psi}(\mathbf{t})$ is the *conductivity*, and $\mathbf{u}(\mathbf{t})$ is the *potential*. The terms conductivity and potential come from EIT [4, 5, 15, 134]. Let

$$\boldsymbol{\psi}(\mathbf{t}) = \exp\big(\mathbf{x}(\mathbf{t})\big) \tag{10.13}$$

and

$$\boldsymbol{\psi}(t(j,:)) = \psi(j) = \exp\big(x(j)\big) = \exp\big(\mathbf{x}(t(j,:))\big) \tag{10.14}$$

and the problem considered in this chapter is estimating x_t. In EIT electrodes are attached to the boundary of an object to apply and measure electric fields. This data is then used to estimate the internal conductivity $\boldsymbol{\psi}$ of the object. A review of the connection between Calderóns Problem and EIT can be found in [135].

In this chapter,

$$\partial\Omega = \partial\Omega_N \tag{10.15}$$

and

$$\mathbf{y} = \mathbf{y}_D \tag{10.16}$$

i.e. Neumann boundary condition and Dirichlet data. In order for $\mathbf{u}$ to be unique, we additionally impose

$$\int_{\Omega} \mathbf{u}(\mathbf{t}) d\mathbf{t} = 0 \tag{10.17}$$

and the *charge conservation* condition [4, 5, 131]

$$\int_{\partial\Omega} g_N(\mathbf{t}) d\mathbf{t} = 0 \tag{10.18}$$

to arrive at the the *continuum model* [136]. Let $g_N(\mathbf{t}) = 1$ on the left and right boundary of Ω and $g_N(\mathbf{t}) = -1$ on the top and bottom boundary of Ω. Let t_d be $n_d = 156$ equispaced points on the boundary $\partial\Omega$. Let $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$ i.e. 1% additive white noise. The resolution of the estimates in this chapter will be lower than the estimates typically found in EIT. This is a consequence of EIT making multiple measurements in which different electrodes are used as the "ground". This leads to several current patterns $\mathbf{u}$ corresponding to the same internal conductivity $\mathbf{x}$. In this chapter, only a single current pattern is used.

This problem is modelled with finite element method (FEM). Modelling the Poisson equation with FEM is well understood, see for example [137, 138]. Development of FEM is often attributed to the work of Hrennikoff [139], Courant [140] and Babuška [141, 142, 143]. A more thorough discussion of FEM can be found in [144, 138]. Given that this thesis is more concerned with the implementation of probabilistic algorithms for computing low rank approximations than with FEM, the technical details of the FEM implementation are provided but not elaborated on.

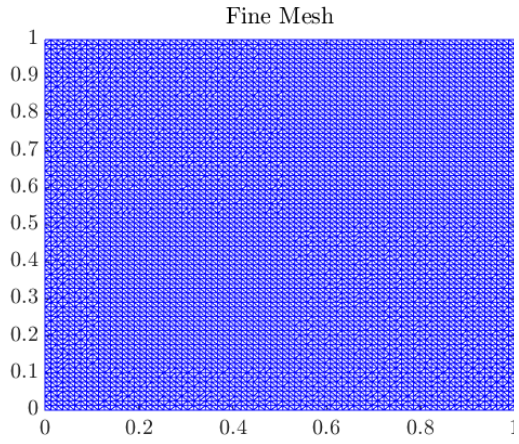


Figure 10.1: Fine mesh

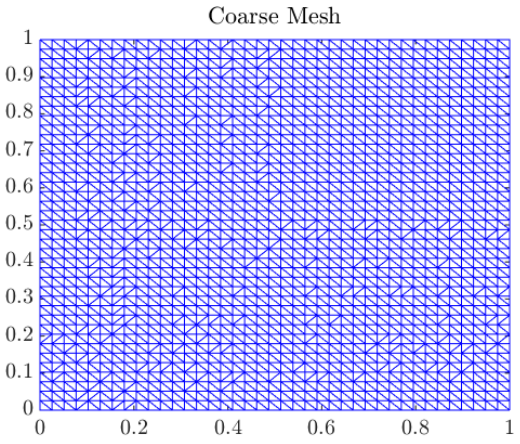


Figure 10.2: Coarse mesh

In this chapter FEM is implemented with linear basis functions and the Ritz-Galerkin approximation. The FEM implementation leads to the expression

$$K_x u = F \tag{10.19}$$

where $u(j) \approx \mathbf{u}\big(t(j,:)\big)$, $u \in \mathbb{R}^{n_x}$, $t \in \mathbb{R}^{n_x \times 2}$ are *node locations*, $x(j) = \mathbf{x}\big(t(j,:)\big)$, $x \in \mathbb{R}^{n_x}$, $K_x \in \mathbb{R}^{n_x \times n_x}$ is the *stiffness matrix* corresponding to a particular x and $F \in \mathbb{R}^{n_x}$ is the *load* or *forcing vector*. The data model is

$$d = Mu + e = M(K_x^{-1}F) + e = A(x) + e \tag{10.20}$$

where $M \in \mathbb{R}^{n_d \times n_x}$ is the measurement operator. The stiffness matrix is typically not explicitly inverted, instead computing e.g. $u = K \setminus F$ in MATLAB. The forward mapping

$$A(x) = \begin{pmatrix} A_{d(1)}(x) \\ A_{d(2)}(x) \\ \vdots \\ A_{d(n_d)}(x) \end{pmatrix} \tag{10.21}$$

is nonlinear. Recall the Jacobian matrix

$$J_x \in \mathbb{R}^{n_d \times n_x} \text{ where } J_x(j,k) = \frac{\partial A_{d(j)}(x)}{\partial x(k)} \tag{10.22}$$

which can be used to construct the affine approximation

$$A(x + \Delta x) \approx A(x) + J_x \Delta x \tag{10.23}$$

where Δx is some small perturbation. Section 6.6.2 outlined how to construct a low rank sample approximation to the Jacobian. However, for this problem the Jacobian is known analytically and can be explicitly formed rapidly, so this chapter takes that approach instead.

This chapter will apply BAE to the estimation of x_t. Let

$$d = \bar{\bar{A}}(\bar{\bar{x}}) + e \tag{10.24}$$

where $\bar{\bar{x}} \in \mathbb{R}^{n_{\bar{\bar{x}}}}$ be the "very fine" model, with $n_{\bar{\bar{x}}} = 160^2 = 25,600$. This is only evaluated once to simulate data and test the estimation scheme without committing an inverse crime. Let

$$d = \bar{A}(\bar{x}) + e \tag{10.25}$$

where $\bar{x} \in \mathbb{R}^{n_{\bar{x}}}$ be the "fine" model, with $n_{\bar{x}} = 80^2 = 6,400$. Let

$$d \approx A(x) + e \tag{10.26}$$

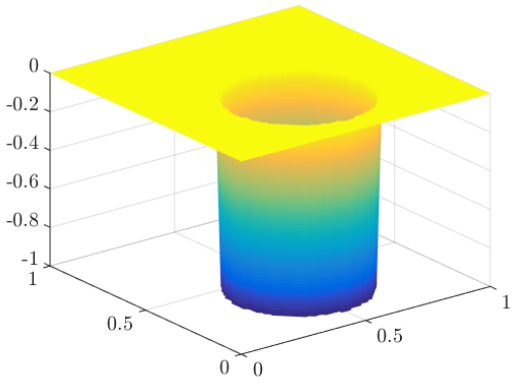


Figure 10.3: Fine mesh ground truth $\bar{x}_t$

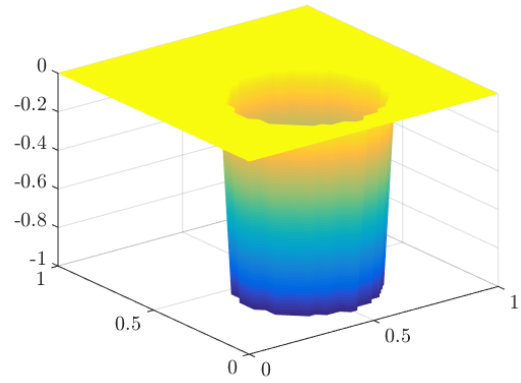


Figure 10.4: Coarse mesh ground truth x_t

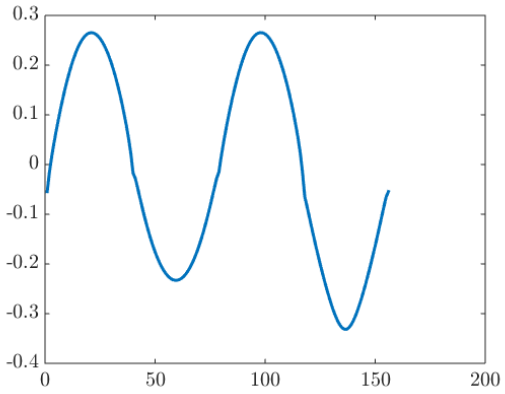


Figure 10.5: Noiseless data/model prediction $\bar{y}_t = \bar{A}(\bar{x}_t)$

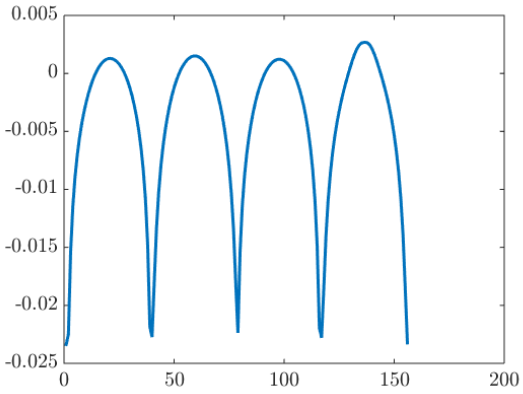


Figure 10.6: Model prediction difference $\bar{y}_t - y_t = \bar{A}(\bar{x}) - A(x)$

where $x \in \mathbb{R}^{n_x}$ be the "coarse" model, with $n_x = 40^2 = 1,600$. For BAE, the models $\bar{A}$ and A are used to generate m samples of ε at the offline stage. Only A and the Gaussian sample approximation to ε are used at the online stage. Note that the only difference between the models is how fine the discretisations are.

The FEM mesh used in the fine model is shown in Figure 10.1. The FEM mesh used in the coarse model is shown in Figure 10.2. The measurement points t_d correspond to the boundary nodes of the coarse mesh, and every second boundary node of the fine mesh.

The fine mesh ground truth $\bar{x}_t$ is shown in Figure 10.3. The data in the simplified conductivity imaging problem considered in the chapter was found in [131] to carry little information on $\mathbf{x}$. This motivates the choice of $\mathbf{x}_t$, as smaller internal structures are unlikely to be recon-

structed. The coarse mesh ground truth is shown in Figure 10.4. The fine model prediction $\bar{y}_t = \bar{A}(\bar{x}_t)$ is shown in Figure 10.5. The difference between the fine and coarse model prediction $\bar{y}_t - y_t = \bar{A}(\bar{x}) - A(x)$ is shown in Figure 10.6. Note the scales.

10.1 Nonlinear Inversion

This section considers estimating x_t from data $d_t = A(x_t) + e_t$ using nonlinear methods. The model A is used both in simulating the data and estimating x_t, so an inverse problem is committed. The purpose of this section is to demonstrate the structure of the simplified conductivity imaging problem. By stating the problem in terms of the coarse model A, the estimation can be approached in a straightforward manner, without using the proposed methods of Chapters 6 and 7. For example, $n_x = 1,600$ means that Γ_x only occupies around 20MB of memory as a full matrix. We are using a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB hard drive, so the dimensionality of this problem provides relatively little obstacle.

Let $x \sim \mathcal{N}(\mu_x, \Gamma_x)$ where Γ_x is constructed explicitly from a smoothness covariance function as described in Section 5.1.1. Given the size of the feature in the known ground truth and the relatively low level of information in the data, a long correlation length is used. The factor L_x such that $L_x L_x^T = \Gamma_x$ is computed as the Cholesky factor. The factor $\tilde{L}_x$ such that $\tilde{L}_x^T \tilde{L}_x = \Gamma_x^{-1}$ is computed by back substitution from L_x. The MAP estimate

$$x_{\mathrm{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - A(x)) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{10.27}$$

is computed with Gauss-Newton iterations. The Gauss-Newton algorithm is briefly reviewed in Section 4.3.2, and further details can be found in numerical optimisation texts e.g. [68]. Once the MAP has been computed, the Laplace approximation to the posterior covariance

$$\Gamma_{x|d} \approx \hat{\Gamma}_{x|d} = \Gamma_x - \Gamma_x J_{\mathrm{MAP}}^T (\Gamma_e + J_{\mathrm{MAP}} \Gamma_x J_{\mathrm{MAP}}^T)^{-1} J_{\mathrm{MAP}} \Gamma_x \tag{10.28}$$

is constructed where J_{MAP} is the Jacobian of A at x_{MAP}. The diagonal $\hat{\sigma}^2$ of $\hat{\Gamma}_{x|d}$ is then used to construct posterior error intervals.

For this problem, the MAP estimate has converged after 12 Gauss-Newton iterations, computing in 8.98 seconds on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive. The posterior covariance is constructed in 2.21 seconds. This gives a total online computation time of 11.2 seconds. A side view of x_t is shown in Figure 10.7 and a top down view in Figure 10.9. A side view of the estimate x_{MAP} is shown in Figure 10.8 and a top down view in Figure 10.10. The posterior uncertainty estimate $\hat{\sigma}$ plotted on Ω as $\hat{\sigma}(j)$ at $t(j,:)$ is shown in Figure 10.11. Note that the uncertainty is lower near the boundary. The ground truth, MAP estimate and $3\hat{\sigma}$ posterior error interval along $\mathbf{t}(2) = 0.5$ i.e. the vertical centre line is shown in Figure 10.12.

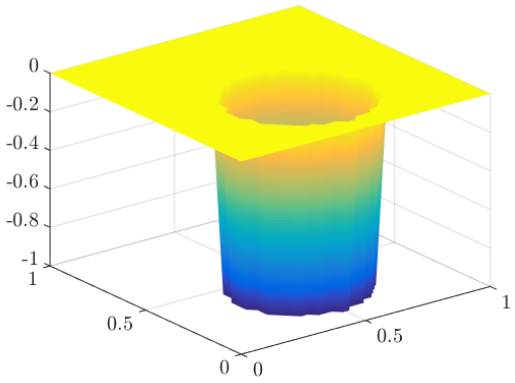


Figure 10.7: Ground truth x_t

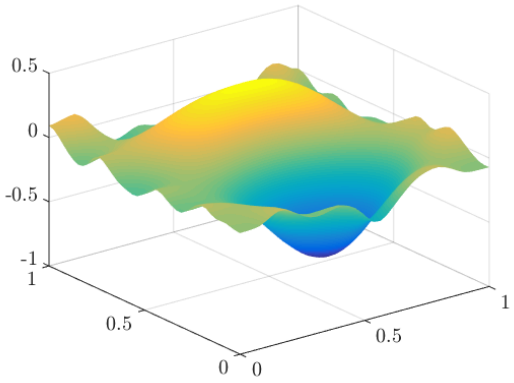


Figure 10.8: Map estimate x_{MAP}

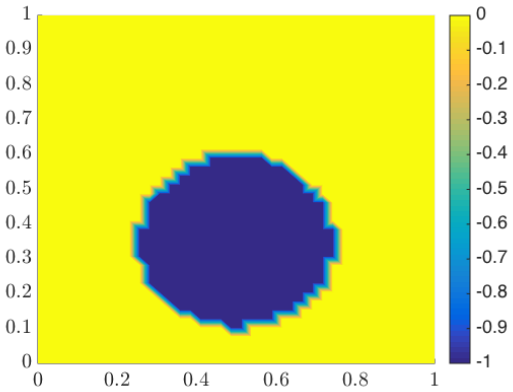


Figure 10.9: Ground truth x_t

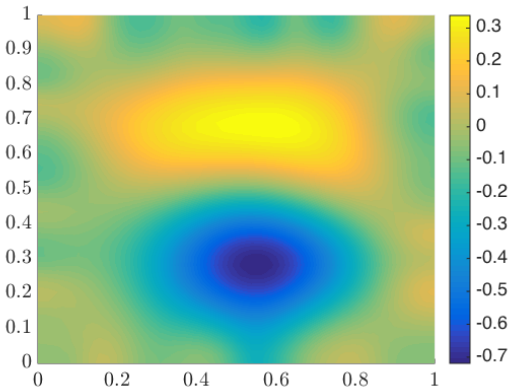


Figure 10.10: Map estimate x_{MAP}

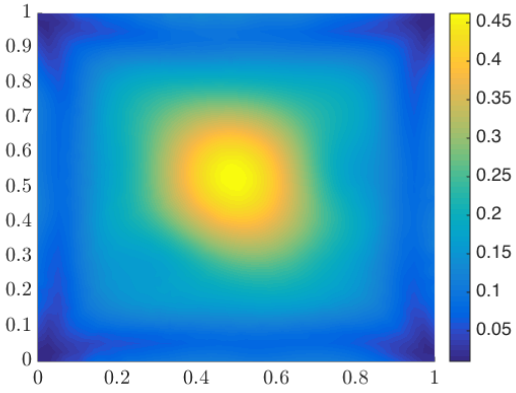


Figure 10.11: Posterior diagonal $\hat{\sigma}$

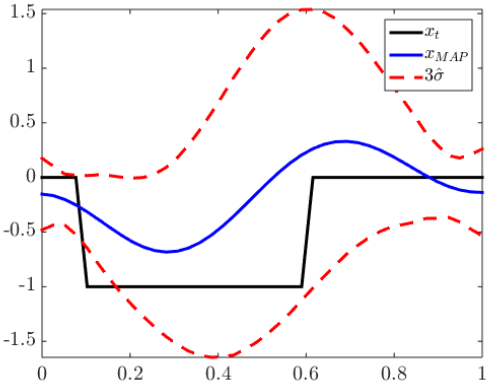


Figure 10.12: Map estimate, posterior error interval and ground truth along $\mathbf{t}(2) = 0.5$

Note that the MAP estimate approaches the negative values of the ground truth, and then adds a positive part. The posterior error intervals seem representative, containing most but not all of the ground truth. While the estimate does not capture the sharp edges of the ground truth feature, this is not surprising given a very smooth prior is used in the estimation. Experiments with smoothness priors with different correlation lengths and heights did not produce particularly better results, with the above estimates being a good balance of detecting features without amplifying noise in this problem.

10.2 Inversion by Local Sample Approximation

In this section a local sample approximation is used to estimate x_t. The fine model $\bar{A}$ is used for computing samples, and the very fine model $\bar{\bar{A}}$ is used to simulate data used when analysing the quality of estimates, avoiding an inverse crime. The approach being used is similar to that described in Section 7.1. A discussion of linear sample approximations to nonlinear operators can be found in Section 6.6.2.

Let $\bar{x} \sim \mathcal{N}(\mu_{\bar{x}}, \Gamma_{\bar{x}})$ where $\Gamma_{\bar{x}} = L_{\bar{x}} L_{\bar{x}}^T$ and $L_{\bar{x}}$ is a smoothing filter constructed as

$$\bar{x}_j = \mu_{\bar{x}} + \mathcal{F}^{-1}\left(k_{f,\bar{x}} \odot \mathcal{F}(w_j) \right) \tag{10.29}$$

where $k_{f,\bar{x}} \in \mathbb{R}^{n_{\bar{x}}}$ is the Fourier transform of a Gaussian kernel. This allows the use of the FFT algorithm when filtering, and only the filters e.g. $k_{f,\bar{x}} \in \mathbb{R}^{n_{\bar{x}}}$ need to be stored in memory. This approach is outlined in Section 5.1.3 and applied in Chapters 8 and 9 with more detail. A draw $\bar{x}_j$ is shown side on in Figure 10.13 and top down in Figure 10.15. The Gaussian kernel width and height were chosen so that draws were consistent with those of the prior in Section 10.1. The coarse scale sample $x_j = P_{x,\bar{x}} \bar{x}_j$ is shown side on in Figure 10.14 and top down in Figure 10.16.

The filter is used to compute m samples $\bar{x}_j$, from which samples $x_j = P_{x,\bar{x}} \bar{x}_j$ and $\bar{y}_j = \bar{A}(\bar{x})$ are computed. These samples are used to construct the QR factors Q_x, R_x, $Q_{\bar{y}}$ and $R_{\bar{y}}$ as described in Section 6.4.1. These factors are then used to construct the sample conditional mean as

$$\begin{aligned}
\hat{\mu}_{x|d} &= \mu_x + \hat{\Gamma}_{xd} \hat{\Gamma}_d^{-1} (d - \hat{\mu}_d) && (10.30)\\
&= \mu_x + \frac{1}{m} Q_x R_x R_{\bar{y}}^T Q_{\bar{y}}^T (\Gamma_e + \frac{1}{m} Q_{\bar{y}} R_{\bar{y}} R_{\bar{y}}^T Q_{\bar{y}}^T)^{-1} (d - \hat{\mu}_{\bar{y}}) && (10.31)\\
&= \mu_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T (\Gamma_e + Q_{\bar{y}} M_{\bar{y}} Q_{\bar{y}}^T)^{-1} (d - \hat{\mu}_{\bar{y}}) && (10.32)\\
&= \mu_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{\bar{y}} (M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}})^{-1} Q_{\bar{y}}^T \right) (d - \hat{\mu}_{\bar{y}}) && (10.33)\\
&= \mu_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T \right) (d - \hat{\mu}_{\bar{y}}) && (10.34)
\end{aligned}$$

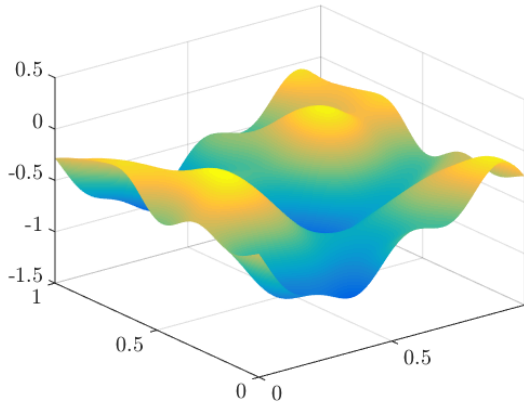


Figure 10.13: Sample $\bar{x}_j$ of $\bar{x}$

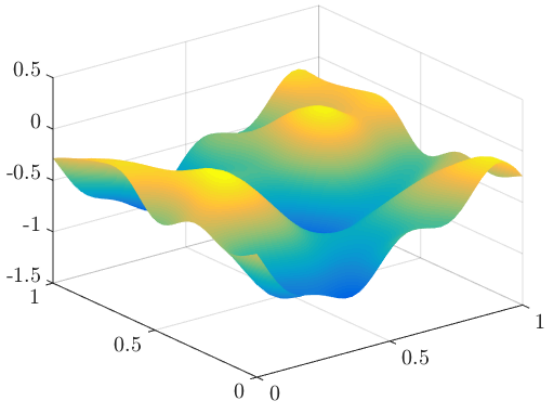


Figure 10.14: Sample $x_j = P_{x,\bar{x}}\bar{x}_j$ of x

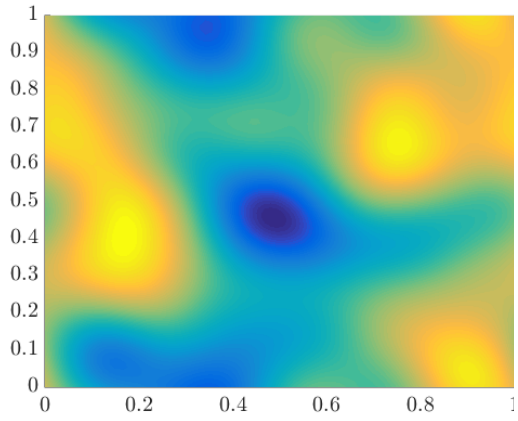


Figure 10.15: Sample $\bar{x}_j$ of $\bar{x}$

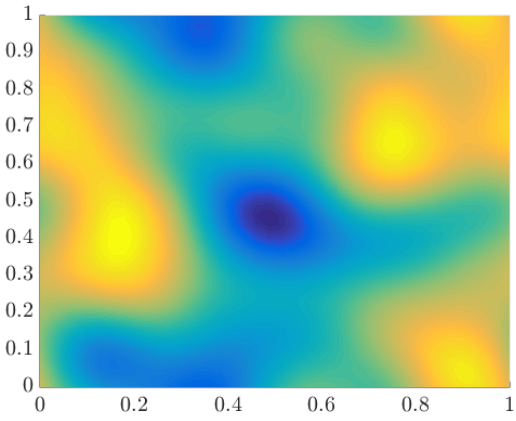


Figure 10.16: Sample $x_j = P_{x,\bar{x}}\bar{x}_j$ of x

where we make use of the orthonormal columns of $Q_{\bar{y}}$, the matrix inversion lemma, and $\Gamma_e = \sigma_e^2 I_{n_d}$. Note that only the accurate model $\bar{A}$ is used in constructing this estimator, and that the estimator does not estimate $\mathbf{u}$. The validity of the sample conditional mean as an estimate of x_t depends on whether $\bar{A}$ is "effectively linear" in the "effective support" of $\pi_{\bar{x}}(\bar{x})$. This is discussed in more detail in Section 6.6.2.

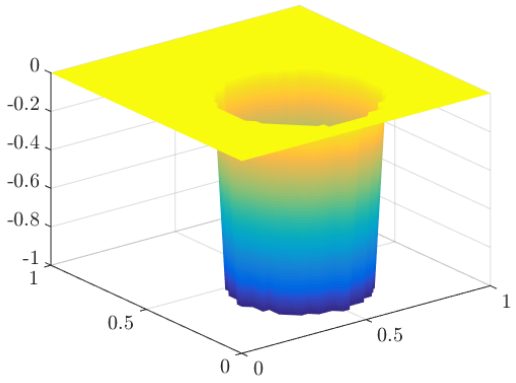


Figure 10.17: Ground truth x_t

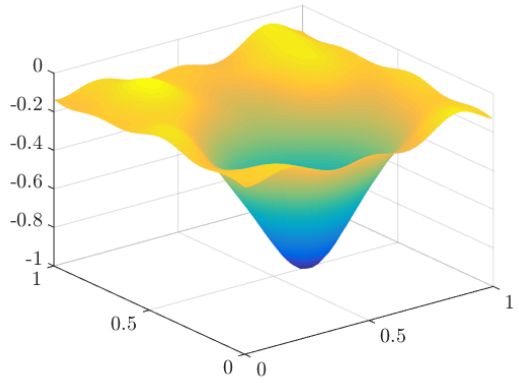


Figure 10.18: Sample conditional mean estimate $\hat{\mu}_{x|d=d_t}$

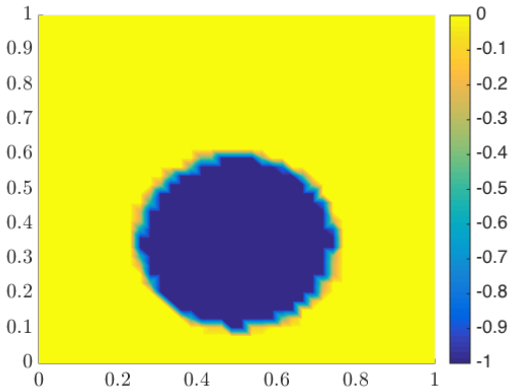


Figure 10.19: Ground truth x_t

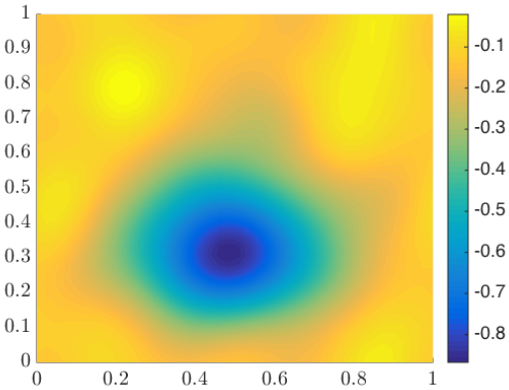


Figure 10.20: Sample conditional mean estimate $\hat{\mu}_{x|d=d_t}$

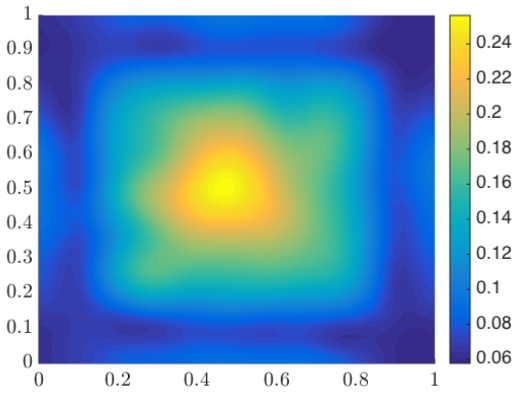


Figure 10.21: Posterior diagonal $\hat{\sigma}$

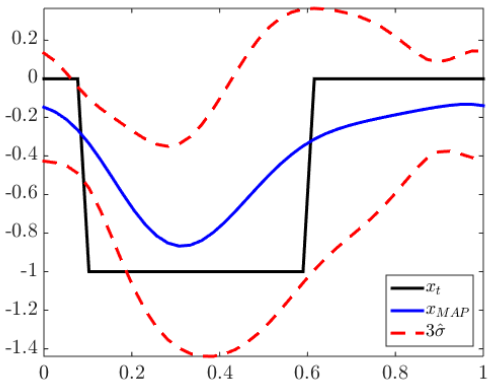


Figure 10.22: Sample CM estimate, posterior error interval and ground truth along $\mathbf{t}(2) = 0.5$

The sample posterior covariance estimate can be expressed as

$$\hat{\Gamma}_{x|d} = \hat{\Gamma}_x - \hat{\Gamma}_{xd}\hat{\Gamma}_d^{-1}\hat{\Gamma}_{dx} \tag{10.35}$$

$$= \frac{1}{m}Q_x R_x R_x^T Q_x^T - \frac{1}{m^2}Q_x R_x R_{\bar{y}}^T Q_{\bar{y}}^T (\Gamma_e + \frac{1}{m}Q_{\bar{y}} R_{\bar{y}} R_{\bar{y}}^T Q_{\bar{y}}^T)^{-1} Q_{\bar{y}} R_{\bar{y}} R_x^T Q_x^T \tag{10.36}$$

$$= Q_x M_x Q_x^T - Q_x M_{x\bar{y}} Q_{\bar{y}}^T (\Gamma_e + Q_{\bar{y}} M_{\bar{y}} Q_{\bar{y}}^T)^{-1} Q_{\bar{y}} M_{x\bar{y}}^T Q_x^T \tag{10.37}$$

$$= Q_x M_x Q_x^T - Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\Gamma_e^{-1} - \Gamma_e^{-1} Q_{\bar{y}} (M_{\bar{y}}^{-1} + Q_{\bar{y}}^T \Gamma_e^{-1} Q_{\bar{y}})^{-1} Q_{\bar{y}}^T \right) Q_{\bar{y}} M_{x\bar{y}}^T Q_x^T \tag{10.38}$$

$$= Q_x M_x Q_x^T - Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T \right) Q_{\bar{y}} M_{x\bar{y}}^T Q_x^T \tag{10.39}$$

$$= Q_x M_x Q_x^T - Q_x M_{x\bar{y}} \left(\frac{1}{\sigma_e^2} I_{r_{\bar{y}}} - \frac{1}{\sigma_e^4} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} \right) M_{x\bar{y}}^T Q_x^T \tag{10.40}$$

$$= Q_x \left(M_x - M_{x\bar{y}} \left(\frac{1}{\sigma_e^2} I_{r_{\bar{y}}} - \frac{1}{\sigma_e^4} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} \right) M_{x\bar{y}}^T \right) Q_x^T \tag{10.41}$$

reusing results from the construction of $\hat{\mu}_{x|d}$. Note that $\hat{\sigma}^2 = \text{diag}(\hat{\Gamma}_{x|d})$ can be constructed element by element, without explicitly forming $\hat{\Gamma}_{x|d} \in \mathbb{R}^{n_x \times n_x}$ in memory. The posterior error interval is constructed as $\hat{\mu}_{x|d} \pm 3\hat{\sigma}$ in this section.

The computational platform used is a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB hard drive. The offline stage of computing $m = 200$ samples, forming

$$\hat{\mu}_{x|d} = \mu_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T \right) (d - \hat{\mu}_{\bar{y}}) \tag{10.42}$$

and

$$\hat{\sigma}^2 = \text{diag}(\hat{\Gamma}_{x|d}) = \text{diag}\left(Q_x \left(M_x - M_{x\bar{y}} (\frac{1}{\sigma_e^2} I_{r_{\bar{y}}} - \frac{1}{\sigma_e^4} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1}) M_{x\bar{y}}^T \right) Q_x^T \right) \tag{10.43}$$

took 10.8 seconds. The tolerance $\tau = \frac{R(1,1)}{100}$ used in computing the QR factors resulted in $r_{\bar{y}} = 26$ and $r_x = m = 200$. This suggest that for $\bar{x}$ in the "effective support" of $\pi_{\bar{x}}$, the operator $\bar{A}$ is not only approximately linear but also effectively low rank. This could explain why only large scale features in the ground truth can be resolved in estimates for this problem [131].

The ground truth x_t is shown side on in Figure 10.17 and top down in Figure 10.19. Data was simulated as $d_t = \bar{\bar{A}}(\bar{\bar{x}}_t) + e_t$ where e_t is a draw of $e \sim \mathcal{N}(0_{n_d,1}, \sigma_e^2 I_{n_d})$. The estimate

$\hat{\mu}_{x|d=d_t}$ is shown side on in Figure 10.18 and top down in Figure 10.20. This estimate appears to be more representative of x_t than the MAP found by nonlinear optimisation and inverse crimes in Section 10.1. The posterior uncertainty estimate $\hat{\sigma}$ plotted on Ω as $\hat{\sigma}(j)$ at $t(j,:)$ is shown in Figure 10.21. Note that the uncertainty is generally lower near the boundary. The ground truth, CM estimate and $3\hat{\sigma}$ posterior error interval along $\mathbf{t}(2) = 0.5$ i.e. the vertical centre line is shown in Figure 10.22. The CM estimate and posterior error intervals appear representative.

Computations were performed on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive. The estimation procedure of Section 10.1 required 8.98 seconds to compute the MAP and an additional 2.21 seconds to compute the posterior covariance for a total of 11.2 seconds of online computation. In this section, the posterior covariance is fixed and computed at the offline stage. The offline stage computes in 10.8 seconds. The online stage, computing the sample CM estimate and adding posterior error intervals, takes 0.0004 seconds.

10.3 Discussion

This chapter considered computing estimates for a simplified conductivity imaging problem. While related to EIT, this problem was chosen to demonstrate the effectiveness of sample based local approximations to a nonlinear inverse problem.

This problem is known to carry little information of the unknown of interest $\mathbf{x}$ [131], hence the example ground truth was chosen to consist of a single large inclusion. The ground truth at fine discretisation $\bar{x}_t$ is shown in Figure 10.3. The ground truth at coarse discretisation is shown in Figures 10.4, 10.7, 10.9, 10.17, and 10.19.

A long correlation length smoothness prior was found to be suitable for this problem. An example draw $\bar{x}_j$ of the fine discretisation unknown $\bar{x}$ is shown in Figures 10.13 and 10.15. An example draw at the coarse discretisation is shown in Figures 10.14 and 10.16. Note that the features in the draw appear to be of similar scale to the inclusion in $\mathbf{x}_t$.

The MAP estimate x_{MAP} found as the solution to the nonlinear optimisation problem

$$x_{\mathrm{MAP}} = \min_x \left\{ \left\| \tilde{L}_e(d_t - A(x)) \right\|_2^2 + \left\| \tilde{L}_x(x - \mu_x) \right\|_2^2 \right\} \tag{10.44}$$

is shown in Figures 10.8 and 10.10. Experiments with altering the Gaussian smoothness prior parameters were not found to notably improve the quality of the estimates. Note that in the above, $d_t = A(x_t) + e_t$, so an inverse crime is committed in forming the above MAP estimate. Online computation of the above MAP estimate on a 2010 laptop with 2.4 GHz dual core CPU, 4 GB of RAM and 250 GB hard drive with 12 Gauss-Newton iterations took 8.98 seconds. The posterior covariance was approximated as

$$\hat{\Gamma}_{x|d} = \Gamma_x - \Gamma_x J_{\mathrm{MAP}}^T (\Gamma_e + J_{\mathrm{MAP}} \Gamma_x J_{\mathrm{MAP}}^T)^{-1} J_{\mathrm{MAP}} \Gamma_x \tag{10.45}$$

and $\hat{\sigma}^2 = \text{diag}(\hat{\Gamma}_{x|d})$ was used to construct posterior error intervals $x_{\text{MAP}} \pm 3\hat{\sigma}$. The above approximate posterior covariance takes 2.21 seconds to compute, giving a total online cputation time of 11.2 seconds. The quantity $\hat{\sigma}$ on Ω is shown in Figure 10.11, noting that posterior error is lower near the boundary $\partial\Omega$, where data is collected. The MAP estimate with posterior error intervals down the centre line $\mathbf{t}(2) = 0.5$ are shown in Figure 10.12.

Note that the MAP estimate produces a very smooth approximation to x_t, and adds a positive inclusion around $\mathbf{t}(0.5, 0.7)$. The smoothness is unsurprising given that a very smooth prior is used in constructing the estimate. The posterior error interval largely contains the ground truth, however the sharp edges are poorly represented. Experiments with other Gaussian smoothness priors did not particularly improve the quality of the estimates. Future work may involve application of a TV prior to capture the sharp edges of $\mathbf{x}_t$.

A local sample approximation was also applied to this nonlinear estimation problem. The model $\bar{A}$ is used for computing samples, and the model $\bar{\bar{A}}$ is used for simulating the data d_t. The sample conditional mean was formed as

$$\hat{\mu}_{x|d} = \mu_x + Q_x M_{x\bar{y}} Q_{\bar{y}}^T \left(\frac{1}{\sigma_e^2} I_{n_d} - \frac{1}{\sigma_e^4} Q_{\bar{y}} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1} Q_{\bar{y}}^T \right) (d - \hat{\mu}_{\bar{y}}) \tag{10.46}$$

and sample posterior error intervals computed from

$$\hat{\sigma}^2 = \text{diag}(\hat{\Gamma}_{x|d}) = \text{diag}\left(Q_x \left(M_x - M_{x\bar{y}} (\frac{1}{\sigma_e^2} I_{r_{\bar{y}}} - \frac{1}{\sigma_e^4} (M_{\bar{y}}^{-1} + \frac{1}{\sigma_e^2} I_{r_{\bar{y}}})^{-1}) M_{x\bar{y}}^T \right) Q_x^T \right) \tag{10.47}$$

as $\hat{\mu}_{x|d} \pm 3\hat{\sigma}$. The offline stage of computing $m = 200$ samples and forming the above sample entities took 10.8 seconds on a 2010 laptop with 2.4 GHz dual core CPU, 4GB of RAM and 250 GB. The forward operator $\bar{A}$ was found to be effectively low rank and linear for reasonable values of $\pi_{\bar{x}}(\bar{x})$. The sample conditional mean estimate $\hat{\mu}_{x|d=d_t}$ is shown in Figures 10.18 and 10.20. The quantity $\hat{\sigma}$ is plotted over Ω in Figure 10.21. Note that the width of posterior error intervals are fixed in this approach, whereas $\hat{\sigma}$ shown in Figure 10.11 corresponds to that particular estimate x_{MAP} given d_t. Note that the general shape of the posterior error intervals are the same, but the posterior error intervals in the local sample approximation approach are narrower than the nonlinear posterior error interval for this d_t.

The sample conditional mean estimate, posterior error interval and ground truth along the vertical centre line $\mathbf{t}(2) = 0.5$ is shown in Figure 10.22. The online stage of evaluation $\hat{\mu}_{x|d=d_t}$ and adding posterior error intervals takes 0.0004 seconds. Note that the total online and offline computational stage of the local sample approximation inversion takes 10.8 seconds, relative to the online stage taking 11.2 seconds for nonlinear inversion. Note that the sample CM estimate does not have the additional positive inclusion seen in the nonlinear MAP. While this method similarly fails to capture the sharp corners of the ground truth, the local sample estimates

appear to be generally more representative than the nonlinear estimates and compute 28,000 times faster at the online stage.

Computing estimates based on local sample approximations performed well for the nonlinear problem of this chapter, with the local sample approximation estimates appearing more representative of the ground truth and computing 28,000 times than estimates found by nonlinear inversion methods. While the problem of this chapter is based on EIT, it is not a physically meaningful problem. Future work might involve applying the methods outlined in this chapter to a more realistic EIT model. The nonlinear estimates could possibly be improved by e.g. implementing a TV type prior, albeit with an increase in computational cost. The analysis of this chapter as intended just to demonstrate an application of inversions by local sample approximation to a nonlinear problem, and this goal has been achieved.

Chapter 11

Conclusions

The computational costs involved in inverse problems may be prohibitive at the offline and/or online stage. This thesis proposes various methods by which these computational costs can be reduced. The recursive QR method proposed in this thesis allows reduced rank approximations to the entities used in BAE to be constructed in parallel with samples being computed, reducing computational cost and memory requirements at the offline and online stage.

Recursive QR was also applied to more general probabilistic algorithms for computing approximate matrix decompositions. The concept of locally accurate probabilistic decompositions was also proposed, with the intent of allowing lower rank approximations to be found with fewer computations. An extension to approximations of nonlinear operators was also developed.

Recursive QR and local sample approximation were developed in this thesis in the context of solving inverse problems, particularly with BAE. The concepts may prove sufficiently general to be more broadly applied, for example using low rank recursive QR constructed local sample approximations to forward operators to reduce the computational costs of simulating physical systems.

A general approach to inverse problems incorporating the methods of this thesis was proposed. This approach was applied to the linear problems of 2D deconvolution and 2D x-ray CT. A nonlinear simplified conductivity imaging problem was also investigated. The methods of this thesis were shown to reduce the computational cost while still providing satisfactory estimates.

The recursive QR method as proposed in this thesis could be further refined in the future. For example, the current Gram-Schmidt orthogonalisation could be replaced with a more stable algorithm such as Householder reflections. The columns of Q could also be reorthogonalised and products of the form RR^T updated to more accurately approximate sample entities.

The methods of this thesis were applied to synthetic problems albeit related to real world applications. The synthetic problems of this thesis were analysed fairly briefly and with some-

what crude estimation schemes. For example, the MAP estimates in Chapters 8 and 9 were computed with Landweber iteration. Future work could involve application of other iterative solvers such as the generalised minimal residual method. While this was sufficient to demonstrate the methodological improvements, more robust analysis of a wider variety of cases could still prove informative. Further work could include applying the methods of this thesis to real world problems. For example, extending the 2D x-ray CT problem considered in this thesis to 3D CT using clinical data. Another possible future avenue of research would be to apply the local sample approximation approach of Chapter 10 to an actual EIT problem, rather than the simplified conductivity imaging problem considered.

In conclusion, this thesis provides various methods by which computational costs can be reduced, with a particular emphasis on probabilistic algorithms computing matrix decompositions from samples.

Bibliography

[1] J. Hadamard, "Sur les problèmes aux dérivés partielles et leur signification physique," *Princeton University Bulletin*, vol. 13, pp. 49–52, 1902.

[2] P. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Monographs on Mathematical Modeling and Computation, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1998.

[3] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 2005.

[4] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*. Springer-Verlag New York, 2005.

[5] J. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*. Computational Science and Engineering, Society for Industrial and Applied Mathematics, 2012.

[6] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography - theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, pp. 413–497, Apr 1993.

[7] B. Gleich and R. Weizenecker, "Tomographic imaging using the nonlinear response of magnetic particles," *Nature*, vol. 435, pp. 1214–1217, Jun 30 2005.

[8] D. Boas, D. Brooks, E. Miller, C. DiMarzio, M. Kilmer, R. Gaudette, and Q. Zhang, "Imaging the body with diffuse optical tomography," *IEEE Signal Processing Magazine*, vol. 18, pp. 57–75, Nov 2001.

[9] S. Arridge and J. Hebden, "Optical imaging in medicine .2. Modelling and reconstruction," *Physics In Medicine And Biology*, vol. 42, pp. 841–853, May 1997.

[10] V. Kolehmainen, S. Siltanen, S. Jarvenpaa, J. Kaipio, P. Koistinen, M. Lassas, J. Pirttila, and E. Somersalo, "Statistical inversion for medical x-ray tomography with few radiographs: II. Application to dental radiology," *Physics in Medicine and Biology*, vol. 48, pp. 1465–1490, May 21 2003.

[11] K. Hamaguchi, K. Nakaji, and E. Nakamura, "Inverse problem for the propagation equation of cosmic-ray electrons/positrons from dark matter," *Physics Letters B*, vol. 680, pp. 172–178, Sep 21 2009.

[12] T. Nakamura and T. Chiba, "Determining the equation of state of the expanding Universe: inverse problem in cosmology," *Monthly Notices Of The Royal Astronomical Society*, vol. 306, pp. 696–700, Jul 1 1999.

[13] C.-M. Yoo, T. Kai, and K.-i. Nakao, "Solving the Inverse Problem with Inhomogeneous Universes," *Progress of Theoretical Physics*, vol. 120, pp. 937–960, Nov 2008.

[14] E. Tatulli, F. Millour, A. Chelli, G. Duvert, B. Acke, O. H. Utrera, K. H. Hofmann, S. Kraus, F. Malbet, P. Mege, R. G. Petrov, M. Vannier, G. Zins, P. Antonelli, U. Beckmann, Y. Bresson, M. Dugue, S. Gennari, L. Gluck, P. Kern, S. Lagarde, E. Le Coarer, F. Lisi, K. Perraut, P. Puget, F. Rantakyro, S. Robbe-Dubois, A. Roussel, G. Weigelt, M. Accardo, K. Agabi, E. Altariba, B. Arezki, E. Aristidi, C. Baffa, J. Behrend, T. Bloecker, S. Bonhomme, S. Busoni, F. Cassaing, J. M. Clausse, J. Colin, C. Connot, A. Delboulbe, A. D. de Souza, T. Driebe, P. Feautrier, D. Ferruzzi, T. Forveille, E. Fossat, R. Foy, D. Fraix-Burnet, A. Gallardo, E. Giani, C. Gil, A. Glentzlin, M. Heiden, M. Heininger, D. Kamm, M. Kiekebusch, D. Le Contel, J. M. Le Contel, T. Lesourd, B. Lopez, M. Lopez, Y. Magnard, A. Marconi, G. Mars, G. Martinot-Lagarde, P. Mathias, J. L. Monin, D. Mouillet, D. Mourard, E. Nussbaum, K. Ohnaka, J. Pacheco, C. Perrier, Y. Rabbia, S. Rebattu, F. Reynaud, A. Richichi, A. Robini, M. Sacchettini, D. Schertl, M. Scholler, W. Solscheid, A. Spang, P. Stee, P. Stefanini, M. Tallon, I. Tallon-Bosc, D. Tasso, L. Testi, F. Vakili, O. von der Luehe, J. C. Valtier, and N. Ventura, "Interferometric data reduction with AMBER/VLTI. Principle, estimators, and illustration," *Astronomy & Astrophysics*, vol. 464, pp. 29–42, Mar 2007.

[15] J. Kaipio, A. Seppanen, E. Somersalo, and H. Haario, "Posterior covariance related optimal current patterns in electrical impedance tomography," *Inverse Problems*, vol. 20, pp. 919–936, Jun 2004.

[16] P. Raghavendra, T. Sahai, P. Kumar, M. Chauhan, and N. Ananthkrishnan, "Aircraft spin recovery, with and without thrust vectoring, using nonlinear dynamic inversion," *Journal of Aircraft*, vol. 42, pp. 1492–1503, Nov-Dec 2005. AIAA 42nd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan 05-08, 2004.

[17] W. Xudong, W. Z. Shen, W. J. Zhu, J. N. Sorensen, and C. Jin, "Shape Optimization of Wind Turbine Blades," *Wind Energy*, vol. 12, pp. 781–803, Nov 2009.

[18] A. P. Nagy, M. M. Abdalla, and Z. Gurdal, "Isogeometric sizing and shape optimisation of beam structures," *Computer Methods In Applied Mechanics And Engineering*, vol. 199, no. 17-20, pp. 1216–1230, 2010.

[19] G. Fornaro, F. Serafino, and F. Soldovieri, "Three-dimensional focusing with multipass SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, pp. 507–517, Mar 2003.

[20] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, "Sparsity and compressed sensing in radar imaging," *Proceedings of the IEEE*, vol. 98, pp. 1006–1020, Jun 2010.

[21] S. Hoole, S. Subramaniam, R. Saldanha, J. Coulumb, and J. Sabonnadiere, "Inverse problem methodology and finite-elements in the identification of cracks, sources, materials, and their geometry in inaccessible locations," *IEEE Transactions on Magnetics*, vol. 27, pp. 3433–3443, May 1991.

[22] T. Lahivaara, N. F. D. Ward, T. Huttunen, Z. Rawlinson, and J. P. Kaipio, "Estimation of aquifer dimensions from passive seismic signals in the presence of material and source uncertainties," *Geophysical Journal International*, vol. 200, pp. 1662–1675, Mar 2015.

[23] A. G. Journel, "Geostatistics for conditional simulation of ore bodies," *Economic Geology*, vol. 69, no. 5, pp. 673–687, 1974.

[24] J. McKinley, C. Deutscht, C. Neufeldt, M. Patton, M. Cooper, and M. Young, "Use of geostatistical Bayesian updating to integrate airborne radiometrics and soil geochemistry to improve mapping for mineral exploration," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 114, pp. 575 – 575, 08 2014.

[25] J. Eidsvik and S. L. Ellefmo, "The value of information in mineral exploration within a multi-gaussian framework," *Mathematical Geosciences*, vol. 45, pp. 777–798, Oct 2013.

[26] K. Kulpa, "The CLEAN type algorithms for radar signal processing," in *MRRS: 2008 Microwaves, Radar and Remote Sensing Symposium, Proceedings* (Yanovsky, F, ed.), (345 E 47TH ST, NEW YORK, NY 10017 USA), pp. 152–157, IEEE AES SP, Ukraine Joint Chapter; GRSS; IEEE East Ukraine Joint Chapter; AESS; MTT S; IEEE; EuMA; uRSi, IEEE, 2008. Microwaves, Radar and Remote Sensing Symposium, Kiev, Ukraine, Sep 22-24, 2008.

[27] A. Nissinen, L. M. Heikkinen, and J. P. Kaipio, "The Bayesian approximation error approach for electrical impedance tomography - experimental results," *Measurement Science and Technology*, vol. 19, Jan 2008.

[28] H. Tapp, A. Peyton, E. Kemsley, and R. Wilson, "Chemical engineering applications of electrical process tomography," *Sensors and Actuators B-Chemical*, vol. 92, pp. 17–24, Jul 1 2003.

[29] J. Kourunen, R. Kayhko, J. Matula, J. Kayhko, M. Vauhkonen, and L. M. Heikkinen, "Imaging of mixing of two miscible liquids using electrical impedance tomography and linear impedance sensor," *Flow Measurement and Instrumentation*, vol. 19, pp. 391–396, Dec 2008.

[30] H. Zhou, J. Jaime Gomez-Hernandez, H.-J. Hendricks Franssen, and L. Li, "An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering," *Advances in Water Resources*, vol. 34, pp. 844–864, Jul 2011.

[31] B. E. Treeby, E. Z. Zhang, and B. T. Cox, "Photoacoustic tomography in absorbing acoustic media using time reversal," *Inverse Problems*, vol. 26, no. 11, p. 115003, 2010.

[32] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.

[33] V. Kolehmainen, A. Vanne, S. Siltanen, S. Jarvenpaa, J. P. Kaipio, M. Lassas, and M. Kalke, "Parallelized Bayesian inversion for three-dimensional dental x-ray imaging," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 218–228, Feb 2006.

[34] V. Kolehmainen, A. Vanne, S. Siltanen, S. Jarvenpaa, J. Kaipio, M. Lassas, and M. Kalke, "Parallelized Bayesian inversion for three-dimensional dental X-ray imaging," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 218–228, Feb 2006.

[35] K. Niinimaki, S. Siltanen, and V. Kolehmainen, "Bayesian multiresolution method for local tomography in dental x-ray imaging," *Physics in Medicine and Biology*, vol. 52, pp. 6663–6678, Nov 21 2007.

[36] J. Kaipio and V. Kolehmainen, "Approximate marginalization over modelling errors and uncertainties in inverse problems," in *Bayesian Theory and Applications* (P. Damien, P. Dellaportas, N. Polson, and D. Stephens, eds.), Oxford University Press, 2015.

[37] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[38] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2233–2246, Nov 2012.

[39] V. Bonin, M. H. Histed, S. Yurgenson, and R. C. Reid, "Local Diversity and Fine-Scale Organization of Receptive Fields in Mouse Visual Cortex," *Journal of Neuroscience*, vol. 31, pp. 18506–18521, Dec 14 2011.

[40] Y. Chen, D. Zhang, Z. Jin, X. Chen, S. Zu, W. Huang, and S. Gan, "Simultaneous denoising and reconstruction of 5-D seismic data via damped rank-reduction method," *Geophysical Journal International*, vol. 206, pp. 1695–1717, Sep 2016.

[41] A. Tikhonov, A. Goncharsky, V. Stepanov, and A. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*. Mathematics and Its Applications, Springer Netherlands, 2013.

[42] H. Anton and R. Busby, *Contemporary Linear Algebra*. Wiley, 2002.

[43] A. Ben-Israel, *Generalized Inverses: Theory and Applications*. Springer, 2003.

[44] S. Campbell and C. Meyer, *Generalized Inverses of Linear Transformations*. Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 2009.

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

[46] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*. Texts in applied mathematics, Springer, 2000.

[47] C. Vogel, *Computational Methods for Inverse Problems*. Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, 2002.

[48] J. Stewart and D. Busch, *Calculus, International Metric Edition*. Cengage Learning, Incorporated, 2015.

[49] T. Sullivan, *Introduction to Uncertainty Quantification*. Texts in Applied Mathematics, Springer International Publishing, 2015.

[50] K. Karhunen, *Ueber lineare Methoden in der Wahrscheinlichkeitsrechnung*. Annales Academiae scientiarum Fennicae. Series A. 1, Mathematica-physica, 1947.

[51] M. Loeve, *Probability Theory II*. F.W.Gehring P.r.Halmos and C.c.Moore, Springer, 1978.

[52] A. Chatterjee, "An introduction to the proper orthogonal decomposition," *Current Science*, vol. 78, pp. 808–817, Apr 10 2000.

[53] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA Journal*, vol. 40, pp. 2323–2330, Nov 2002. AIAA 15th Computational Fluid Dynamics Conference, ANAHEIM, CA, Jun 11-14, 2001.

[54] G. Berkooz, P. Holmes, and J. Lumley, "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, vol. 25, pp. 539–575, 1993.

[55] K. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Elsevier Science, 2014.

[56] P. Hansen, J. Nagy, and D. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*. Fundamentals of Algorithms, SIAM, Society for Industrial and Applied Mathematics, 2006.

[57] L. Landweber, "An iteration formula for fredholm integral equations of the first kind," *American Journal of Mathematics*, vol. 73, no. 3, pp. 615–624, 1951.

[58] A. Louis, *Inverse und schlecht gestellte Probleme*. Teubner Studienbücher Mathematik, Vieweg+Teubner Verlag, 2013.

[59] G. Vainikko and A. Y. Veretennikov, "Iteration procedures in ill-posed problems," 1986.

[60] M. Hanke, A. Neubauer, and O. Scherzer, "A convergence analysis of the landweber iteration for nonlinear ill-posed problems," *Numerische Mathematik*, vol. 72, no. 1, pp. 21–37, 1995.

[61] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*. Mathematics and Its Applications, Springer Netherlands, 1996.

[62] A. M. Stuart, "Inverse problems: a Bayesian perspective," *Acta Numerica*, vol. 19, pp. 451–559, 2010.

[63] D. Calvetti and E. Somersalo, *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Surveys and Tutorials in the Applied Mathematical Sciences, Springer New York, 2007.

[64] T. Bayes, R. Price, and J. Canton, "An essay towards solving a problem in the doctrine of chances," 1763.

[65] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. Communications and signal processing, McGraw-Hill, 1991.

[66] Y. Tong, *The Multivariate Normal Distribution*. Springer Series in Statistics, Springer Verlag GMBH, 1990.

[67] M. Slotani, "Tolerance regions for a multivariate normal population," *Annals of the Institute of Statistical Mathematics*, vol. 16, pp. 135–153, Dec 1964.

[68] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, Springer New York, 2000.

[69] P. S. Laplace, "Memoir on the probability of the causes of events," *Statistical Science*, vol. 1, pp. 364–378, 08 1986.

[70] D. J. MacKay, "Choice of basis for Laplace approximation," *Machine Learning*, vol. 33, pp. 77–86, Oct 1998.

[71] D. V. Lindley, "Approximate Bayesian methods," *Trabajos de Estadistica Y de Investigacion Operativa*, vol. 31, pp. 223–245, Feb 1980.

[72] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, pp. 63–71, Springer, 2004.

[73] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series, University Press Group Limited, 2006.

[74] S. W. Anzengruber and R. Ramlau, "Morozov's discrepancy principle for Tikhonov-type functionals with nonlinear operators," *Inverse Problems*, vol. 26, no. 2, p. 025001, 2010.

[75] S. Pereverzev and E. Schock, "Morozov's discrepancy principle for Tikhonov," *Numerical Functional Analysis and Optimization*, vol. 21, no. 7-8, pp. 901–916, 2000.

[76] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the l-curve," *SIAM Review*, vol. 34, no. 4, pp. 561–580, 1992.

[77] M. Belge, M. E. Kilmer, and E. L. Miller, "Efficient determination of multiple regularization parameters in a generalized l-curve framework," *Inverse Problems*, vol. 18, no. 4, p. 1161, 2002.

[78] P. C. Hansen, "The l-curve and its use in the numerical treatment of inverse problems," 1999.

[79] B. Anderson and J. Moore, *Optimal Filtering*. Dover books on engineering, Dover Publications, 2004.

[80] J. Cooley, P. Lewis, and P. Welch, "Application of the fast Fourier transform to computation of Fourier integrals, Fourier series, and convolution integrals," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 79–84, June 1967.

[81] A. Papoulis, *The Fourier Integral and Its Applications*. Dover Books on Engineering, Dover Publications, Incorporated, 2018.

[82] L. Devroye, *Non-Uniform Random Variate Generation*. Springer New York, 1986.

[83] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[84] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[85] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[86] F. Andersson, M. Carlsson, and L. Tenorio, "On the representation of functions with gaussian wave packets," *Journal of Fourier Analysis and Applications*, vol. 18, pp. 146–181, Feb 2012.

[87] F. J. Herrmann, P. Moghaddam, and C. C. Stolk, "Sparsity-and continuity-promoting seismic image recovery with curvelet frames," *Applied and Computational Harmonic Analysis*, vol. 24, no. 2, pp. 150–173, 2008.

[88] I. Loris, G. Nolet, I. Daubechies, and F. Dahlen, "Tomographic inversion using L1-norm regularization of wavelet coefficients," *Geophysical Journal International*, vol. 170, no. 1, pp. 359–370, 2007.

[89] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, no. 263-340, p. 227, 2010.

[90] M. Lassas and S. Siltanen, "Can one use total variation prior for edge-preserving bayesian inversion?," *Inverse Problems*, vol. 20, no. 5, p. 1537, 2004.

[91] J. Kaipio and E. Somersalo, "Statistical inverse problems: Discretization, model reduction and inverse crimes," *Journal of Computational and Applied Mathematics*, vol. 198, no. 2, pp. 493 – 504, 2007. Special Issue: Applied Computational Inverse Problems.

[92] Z. Yao, Z. Hu, and J. Li, "A TV-Gaussian prior for infinite-dimensional Bayesian inverse problems and its numerical implementations," *Inverse Problems*, vol. 32, p. 075006, July 2016.

[93] H. Zhu, C. K. Williams, R. Rohwer, and M. Morciniec, "Gaussian regression and optimal finite dimensional linear models," 1998.

[94] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 4, pp. 423–498, 2011.

[95] A. Seppanen, M. Vauhkonen, P. Vauhkonen, E. Somersalo, and J. Kaipio, "State estimation with fluid dynamical evolution models in process tomography - an application to impedance tomography," *Inverse Problems*, vol. 17, pp. 467–483, Jun 2001.

[96] D. Scott, R. Williams, and E. F. (U.S.), *Frontiers in Industrial Process Tomography*. Engineering Foundation Conference Proceedings Series, Engineering Foundation, 1995.

[97] R. Williams and M. Beck, *Process Tomography: Principles, Techniques, and Applications*. Measurement science & technology, Butterworth-Heinemann, 1995.

[98] N. Stokes, *Probability and Statistics*. Larsen and Keller Education, 2017.

[99] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini, "Likelihood-informed dimension reduction for nonlinear inverse problems," *Inverse Problems*, vol. 30, no. 11, p. 114015, 2014.

[100] T. Cui, Y. M. Marzouk, and K. E. Willcox, "Data-driven model reduction for the Bayesian solution of inverse problems," *International Journal for Numerical Methods in Engineering*, vol. 102, pp. 966–990, May 4 2015.

[101] M. Mozumder, T. Tarvainen, S. Arridge, J. P. Kaipio, C. D'Andrea, and V. Kolehmainen, "Approximate marginalization of absorption and scattering in fluorescence diffuse optical tomography," *Inverse Problems & Imaging*, vol. 10, no. 1, pp. 227–246, 2016.

[102] J. Koponen, T. Huttunen, T. Tarvainen, and J. P. Kaipio, "Bayesian approximation error approach in full-wave ultrasound tomography," *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 61, pp. 1627–1637, Oct 2014.

[103] W. Xie, Y. Deng, D. Yan, X. Yang, and Q. Luo, "Sparsity-promoting Bayesian approximation error method for compensating for the mismodeling of optical properties in fluorescence molecular tomography," *Optics Letters*, vol. 42, pp. 3024–3027, Aug 2017.

[104] T. Tarvainen, A. Pulkkinen, B. T. Cox, J. P. Kaipio, and S. R. Arridge, "Bayesian image reconstruction in quantitative photoacoustic tomography," *IEEE Transactions on Medical Imaging*, vol. 32, pp. 2287–2298, Dec 2013.

[105] M. Simon, "Bayesian anomaly detection in heterogeneous media with applications to geophysical tomography," *Inverse Problems*, vol. 30, Nov 2014.

[106] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*. Applied Multivariate Statistical Analysis, Pearson Prentice Hall, 2007.

[107] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM Journal on Applied Mathematics*, vol. 24, no. 3, pp. 315–323, 1973.

[108] F. Sullivan and J. Dongarra, "The top 10 algorithms," *Computing in Science and Engineering*, vol. 2, pp. 22–23, 01 2000.

[109] G. W. Stewart, "The decompositional approach to matrix computation," *Computing in Science and Engineering*, vol. 2, pp. 50–59, Jan 2000.

[110] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," *CoRR*, vol. abs/0812.4293, 2008.

[111] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 158–183, 2006.

[112] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, no. 2, pp. 217 – 235, 2000.

[113] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 143–152, Oct 2006.

[114] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices," *Proceedings of the National Academy of Sciences*, vol. 104, no. 51, pp. 20167–20172, 2007.

[115] Y. Chen and D. S. Oliver, "Ensemble randomized maximum likelihood method as an iterative ensemble smoother," *Mathematical Geosciences*, vol. 44, pp. 1–26, Jan 2012.

[116] Y. Chen and D. S. Oliver, "Levenberg–Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification," *Computational Geosciences*, vol. 17, pp. 689–703, Aug 2013.

[117] R. G. Brown, P. Y. Hwang, *et al.*, *Introduction to Random Signals and Applied Kalman Filtering*, vol. 3. Wiley New York, 1992.

[118] A. Gelman, D. B. Rubin, *et al.*, "Inference from iterative simulation using multiple sequences," *Statistical science*, vol. 7, no. 4, pp. 457–472, 1992.

[119] A. Gelman, X.-L. Meng, and H. Stern, "Posterior predictive assessment of model fitness via realized discrepancies," *Statistica Sinica*, pp. 733–760, 1996.

[120] G. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[121] L. Eriksson, *Design of Experiments: Principles and Applications*. Umetrics Academy - training in multivariate technology, Umetrics Academy, 2000.

[122] J. D. Dixon, "Estimating extremal eigenvalues and condition numbers of matrices," *SIAM Journal on Numerical Analysis*, vol. 20, no. 4, pp. 812–814, 1983.

[123] S. Siltanen, V. Kolehmainen, S. Jarvenpaa, J. P. Kaipio, P. Koistinen, M. Lassas, J. Pirttila, and E. Somersalo, "Statistical inversion for medical x-ray tomography with few radiographs: I. general theory," *Physics in Medicine and Biology*, vol. 48, no. 10, p. 1437, 2003.

[124] J. Hsieh, *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*. Press Monograph Series, SPIE, 2015.

[125] F. Natterer, *The Mathematics of Computerized Tomography*. Wiley, 1986.

[126] R. Rangayyan, A. P. Dhawan, and R. Gordon, "Algorithms for limited-view computed tomography: an annotated bibliography and a challenge," *Applied Optics*, vol. 24, no. 23, pp. 4000–4012, 1985.

[127] K. M. Hanson, "Bayesian and related methods in image reconstruction from incomplete data," *Image Recovery: Theory and Application*, pp. 79–125, 1987.

[128] K. T. Smith and F. Keinert, "Mathematical foundations of computed tomography," *Applied Optics*, vol. 24, no. 23, pp. 3950–3957, 1985.

[129] P. Kuchment, K. Lancaster, and L. Mogilevskaya, "On local tomography," *Inverse Problems*, vol. 11, no. 3, p. 571, 1995.

[130] A. Faridani, E. L. Ritman, and K. T. Smith, "Local tomography," *SIAM Journal on Applied Mathematics*, vol. 52, no. 2, pp. 459–484, 1992.

[131] R. Nicholson, *Approaches to Multiscale Inverse Problems (PhD. thesis)*. 2016.

[132] A. P. Calderón, "On an inverse boundary value problem," *Seminar on Numerical Analysis and its Applications to Continuum Physics*, pp. 65–73, 1980.

[133] A. P. Calderón, "On an inverse boundary value problem (reprint)," *Computational & Applied Mathematics*, vol. 25, no. 2-3, pp. 133–138, 2006.

[134] L. Borcea, "Electrical impedance tomography," *Inverse Problems*, vol. 18, pp. R99–R136, Dec 2002.

[135] G. Uhlmann, "Electrical impedance tomography and Calderon's problem," *Inverse Problems*, vol. 25, Dec 2009.

[136] M. Cheney, D. Isaacson, and J. Newell, "Electrical impedance tomography," *SIAM Review*, vol. 41, pp. 85–101, Mar 1999.

[137] L. Evans and A. M. Society, *Partial Differential Equations*. Graduate studies in mathematics, American Mathematical Society, 2010.

[138] M. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*. Texts in Computational Science and Engineering, Springer Berlin Heidelberg, 2013.

[139] A. Hrennikoff, "Solution of problems of elasticity by the framework method," *Journal of Applied Mechanics*, 1941.

[140] R. Courant *et al.*, "Variational methods for the solution of problems of equilibrium and vibrations," *Lecture Notes in Pure and Applied Mathematics*, pp. 1–1, 1994.

[141] I. Babuška, "Error-bounds for finite element method," *Numerische Mathematik*, vol. 16, no. 4, pp. 322–333, 1971.

[142] I. Babuška, R. B. Kellogg, and J. Pitkäranta, "Direct and inverse error estimates for finite elements with mesh refinements," *Numerische Mathematik*, vol. 33, no. 4, pp. 447–471, 1979.

[143] B. Szabo, B. A. Szabo, and I. Babuška, *Finite Element Analysis*. John Wiley & Sons, 1991.

[144] S. Brenner and R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15. Springer Science & Business Media, 2007.