Linear Parameter Uncertainty Quantification using Surrogate Gaussian Processes

Romcholo Yulo Macatula

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

In

Mathematics

Matthias Chung, Chair

Johnthan Bardsley

Robert Gramacy

Serkan Gugercin

June 22, 2020

Blacksburg, VA

Keywords: Uncertainty Quantification, Surrogate Models, Linear Parameter Estimation, Tomography, Bayesian Inference, Gaussian Process

Linear Parameter Uncertainty Quantification using Surrogate Gaussian Processes

Romcholo Yulo Macatula

Abstract

We consider uncertainty quantification using surrogate Gaussian processes. We take a previous sampling algorithm and provide a closed form expression of the resulting posterior distribution. We extend the method to weighted least squares and a Bayesian approach both with closed form expressions of the resulting posterior distributions. We test methods on 1D deconvolution and 2D tomography. Our new methods improve on the previous algorithm, however fall short in some aspects to a typical Bayesian inference method.

Linear Parameter Uncertainty Quantification using Surrogate Gaussian Processes

Romcholo Yulo Macatula

General Audience Abstract

Parameter uncertainty quantification seeks to determine both estimates and uncertainty regarding estimates of model parameters. Example of model parameters can include physical properties such as density, growth rates, or even deblurred images. Previous work has shown that replacing data with a surrogate model that approximates the underlying model can provide promising estimates with low uncertainty. We extend the previous methods in the specific field of linear models. Theoretical results are tested on simulated computed tomography problems.

Contents

1 Parameter Estimation and Uncertainty Quantification

Dynamical systems are physical models where a state or set of states evolve over time. Examples of dynamical systems can include fluid flows, population dynamics, or motion of a physical body [1]. The *forward problem* is to use initial state information and model parameters to make estimates of future state measurements. Considering population dynamics for instance, the state measurements can be the population of animal species at particular times, while model parameters can include the interaction rates between species or birth and death rates for particular species.

In comparison, parameter estimation can be seen as an *inverse problem*. Given measurements of the system, we wish to find model parameters that characterize the system. Some inverse problems result from a system of differential equations, or result from an integral equation. Here, we deal primarily with linear inverse problems, which often arise when discretizing the underlying differential or integral equations. Physical examples of linear inverse problems include imaging [2], elastodynamics [3], and computed tomography [4].

According to Sullivan, uncertainty quantification (UQ) is the combination of statistical methods to measure uncertainty and "real-world" physics that model underlying dynamics [5]. An example of UQ can be as simple as single parameters estimates supplemented with confidence intervals. A typical technique is Bayesian inference, where the user provides, protentially empirically, a prior distribution on the parameters of interest, and the likelihood is influenced by model dynamics.

The idea of surrogate modeling is to approximate some underlying model with another, less computationally intensive model. While not limited to optimization, often these models are used to speed up methods where function evaluations are costly [6, 7, 8]. One particular example of surrogate modeling is kriging, or by another name, gaussian processes [9]. Some applications include structural reliability analysis [10] and hydrosciences [11].

Chung and Gramacy et al. proposed an algorithm that uses surrogate stochastic processes for uncertainty quantification on parameter estimation problems [12]. Model observations are approximated by some problem appropriate stochastic process, often with tuned hyperparameters. Then random draws from this distribution are used in classical least squares invesions. The distribution of these inversions are then used to create confidence intervals on parameter estimates. The pseudocode is provided in Algorithm 2.

Here, we provide the closed form distribution of parameter estimates following Algorithm 2 on linear models. We go further and improve estimates by introducing additional information of the surrogate Gaussian process into parameter estimates, then extend results to a general Bayesian framework. The rest of the paper is organized as follows: Section 2 provides a brief overview of Gaussian processes, Section 3 introduces various new closed form posterior parameter distributions with linear inverse problems, and Section 4 provides numerical tests with synthetic data.

2 Gaussian Processes

The main idea of the algorithm for fast and informative UQ in parameter estimation is to approximate the data, $\mathbf{d}$, using a surrogate stochastic process. Here, we will only consider Gaussian processes.

2.1 Random Fields and Gaussian Processes

We begin with a definition of a stochastic process.

Definition 2.1. *A **stochastic process** is a collection of random variables, $\{y(x)\}_{x\in\Omega}$, indexed by $x \in \Omega$, where Ω is some indexing set or domain.*

It is common to refer to a stochastic process as a *random field* or *random function* if the indexing set is multidimensional, i.e. $\mathbb{R}^d$ or some manifold. The term *stochastic process* is reserved when the domain is the integers, or some interval in $\mathbb{R}$.

A Gaussian process is a particular stochastic process that models spatial data through joint multivariate normal distributions. One can think of a Gaussian process as a extension of a multivariate normal to "infinite dimensions". A useful application of a Gaussian process is to make predictions at new input locations given data.

Definition 2.2. *A Gaussian process is a random field, $\{y(x)\}_{x\in\Omega}$, where any finite collection of random variables, $\{Y(x_i) : x_i \in \Omega\}_{i=1}^m$ follows a multivariate normal distribution.*

A Gaussian process is characterized by a choice of mean function, $\mu : \Omega \to \mathbb{R}$ and kernel (or covariance) function $\kappa : \Omega \times \Omega \to \mathbb{R}$. Let $\mathbf{x} = [x_1, x_2, ..., x_m]^\top$, where $x_i \in \Omega$ for $i = 1, 2, ..., m$. We define the mean vector

$$\boldsymbol{\mu}(\mathbf{x}) := [\mu(x_1), \mu(x_2), ..., \mu(x_m)]^\top \in \mathbb{R}^m$$

and the covariance matrix

$$\boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x}) := \begin{bmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \dots & \kappa(x_1, x_m) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & & \vdots \\ \vdots & & \ddots & \vdots \\ \kappa(x_m, x_1) & \dots & \dots & \kappa(x_m, x_m) \end{bmatrix} \in \mathbb{R}^{m\times m}$$

We denote the Gaussian process as

$$\mathbf{y}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x})).$$

2.2 Kernel Functions

We define kernel and covariance functions which serve a critical role in characterizing Gaussian processes.

Definition 2.3. *A kernel function, also called a covariance function, is a map $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that also satisfies,*

i) For $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa(\mathbf{x}_2, \mathbf{x}_1)$ (symmetric)

ii) For $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, $\kappa(\mathbf{x}_1, \mathbf{x}_2) \geq 0$ (semi-positive definite)

Moving forward, we will assume that kernel functions are additionally positive definite. Note, we will use the use the terms "kernel function" and "covariance function" interchangeably. Kernel functions often have additional properties depending on author conventions. We follow definitions from [13].

Definition 2.4. *Let $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel function, and let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$.*

- *We say the kernel is **stationary**, or **homogeneous**, if the kernel depends only on the lag between two inputs, rather than the inputs themselves. Namely,*

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa(\mathbf{x}_1 - \mathbf{x}_2).$$

- *We say the kernel is **isotropic** if the kernel depends only on the distance between two inputs, defined with some metric, $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Namely,*

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \kappa(d(\mathbf{x}_1, \mathbf{x}_2)).$$

Proposition 2.1. *All isotropic kernels are also stationary.*

One should also note, stationary kernels are invariant of translations of the inputs. However, isotropic kernels are invariant of both translation and rotation of inputs. Properties of symmetric positive definite functions carry over to kernel functions, allowing us to combine kernel functions.

Proposition 2.2. *Let $\kappa_1, \kappa_2, ..., \kappa_N$ be kernel functions. Then,*

i) $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{N} \kappa_k(\mathbf{x}_1, \mathbf{x}_2)$ *is also a kernel function.*

ii) $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \prod_{i=1}^{N} \kappa_k(\mathbf{x}_1, \mathbf{x}_2)$ *is also a kernel function.*

Proof. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$. Suppose for all $i = 1, 2, ..., N$, κ_i is a kernel. Then $\kappa_i(\mathbf{x}_1, \mathbf{x}_2) \geq 0$ and $\kappa_i(\mathbf{x}_1, \mathbf{x}_2) = \kappa_i(\mathbf{x}_2, \mathbf{x}_1)$. We show that the sum of kernel functions are symmetric and semi-positive definite.

$$\begin{aligned} \kappa(\mathbf{x}_1, \mathbf{x}_2) &= \sum_{i=1}^{N} \kappa(\mathbf{x}_1, \mathbf{x}_2) \geq 0 && \text{(Semi-positive definite)} \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) &= \sum_{i=1}^{N} \kappa_i(\mathbf{x}_2, \mathbf{x}_1) = \sum_{i=1}^{N} \kappa_i(\mathbf{x}_1, \mathbf{x}_2) \\ &= \kappa(\mathbf{x}_1, \mathbf{x}_2). && \text{(Symmetry)} \end{aligned}$$

The proof for products of kernels is similarly done. ■

We provide a few example kernel functions.

- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \tau^2 \exp\left(-\frac{1}{2\ell^2} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right), \quad \tau, \ell \in \mathbb{R}$

  Squared exponential kernel. This is a commonly used kernels for Gaussian processes. We see that this is an isotropic kernel. Additionally, this kernel is infinitely differentiable, which is critical when trying to differentiate a GP. In other contexts, this is also called a *Radial Basis Function.*

- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \tau^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\ell^2}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\ell^2}\right), \quad \tau, \ell \in \mathbb{R}, \quad \nu \in \mathbb{R}^+$

  Matérn kernel. Above, Γ is the Gamma function and K_ν is the modified Bessel function of the second kind. This kernel is isotropic, and is $\lceil \nu \rceil - 1$ times differentiable. As a special case, when $\nu \to \infty$, the Matérn kernel converges to the Squared Exponential kernel. In practice, ν is chosen to specify the level of smoothness of the kernel.

- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \tau^2 \exp\left(-\frac{1}{\ell^2}\|\mathbf{x}_i - \mathbf{x}_j\|_2\right), \quad \tau, \ell \in \mathbb{R}$

  Ornstein-Uhlenbeck kernel. This is seemingly similar to the squared exponential kernel; it is also isotropic. However, by not squaring the norm, this kernel is not differentiable when $\mathbf{x}_1 = \mathbf{x}_2$. One can show that this is the Matérn kernel with a particular choice of ν.

- $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \delta_{ij}, \quad \sigma \in \mathbb{R}, \quad \delta_{ij} = \begin{cases} 1, \text{if } \mathbf{x}_1 = \mathbf{x}_2 \\ 0, \text{otherwise} \end{cases}$

  This is the *Gaussian noise kernel*, or *nugget*, often used when assuming data contains additive white noise.

Samples of Gaussian processes using these kernel functions are given in Figure 1. Note that while all samples are continuous, they exhibit different levels of smoothness with different kernel functions. In particular, the squared exponential kernel's smoothness can be attributed to the differentiability of the kernel functions. More details can be found in [14].

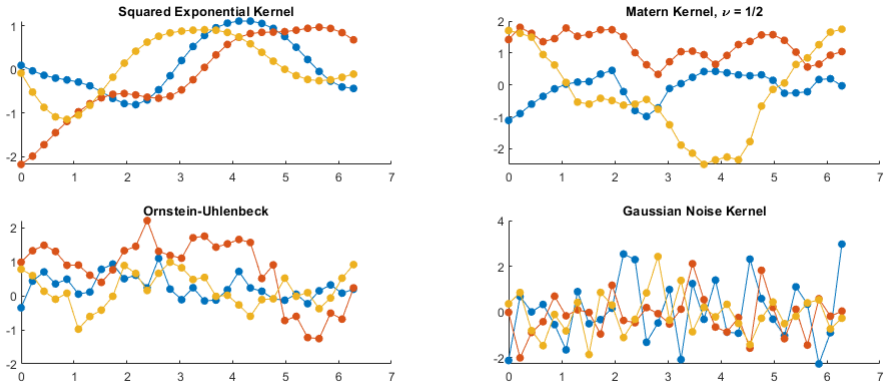

Figure 1: Various samples from Gaussian processes with different kernel functions. Points are shown to emphasize that samples are a collection of function values.

One should notice that many of the given kernel functions above have additional parameters involved. These are called hyperparameters. For example, the squared exponential kernel has two,

$$\kappa(\mathbf{x}_1, \mathbf{x}_2 | \tau, \ell) = \tau^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2\right).$$

Often, one can interpret the effect of each hyperparameters. For instance, τ is the amplitude of the kernel function, and thus of the uncertainty throughout the domain. Additionally ℓ is the length scale, which determines how fast the covariance decreases as two inputs are farther apart. The squared exponential kernel with various amplitudes and length scales are provided in Figure 2.

Example 1: Consider the states of a 1D function, $y(x)$, to be a random variable. The collection of function values $\{y(x) | x \in \mathbb{R}\}$ may be uncountable. However, if you consider any finite collection

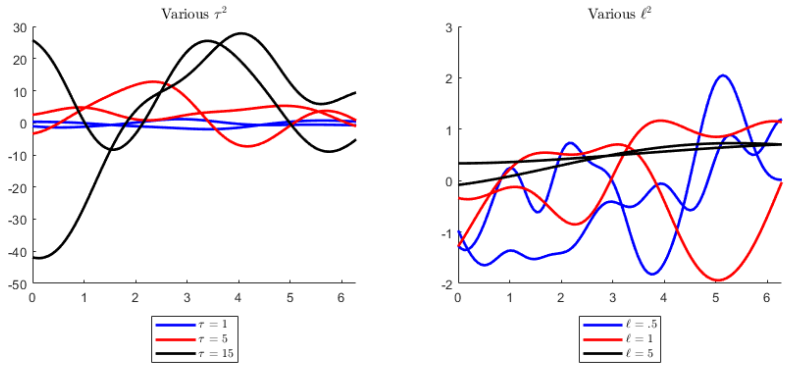

Figure 2: We note that the higher the value of the amplitude, τ, the greater the magnitude of samples. For the length scale, a higher ℓ value forces the samples to be closer together over longer stretches. The less ℓ is, the more the samples are allowed to "wiggle".

of function values, for instance, function values $y(x_i)$ at particular points $\mathbf{x} = [x_1, x_2, ..., x_m]^\top$, then they follow a m dimensional multivariate normal distribution with some mean $\boldsymbol{\mu}(\mathbf{x})$ and covariance $\boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x})$.

$$\begin{bmatrix} y(x_1) \\ \vdots \\ y(x_m) \end{bmatrix} \sim \mathcal{N}_m(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x}))$$

Next we select a mean function $\mu(x)$ and a kernel function $\kappa(x, x)$. Suppose we select a zero mean function and a squared exponential kernel,

$$\mu(t) = 0, \qquad \kappa(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|_2^2}{2}\right),$$

then we can now construct the mean vector and covariance matrix. We can then take random samples from this multivariate normal distribution and plot the function values at the input locations $\mathbf{x}$. Samples of 20 linearly spaced points on the interval $[0, 2\pi]$ are given in Figure 3.

2.3 Calculating Optimal Hyperparameters

The hyperparameters of the kernel function control particular properties of the Gaussian process. For some given set of data, we have the freedom to choose any set of values for the hyperparameters. However, one can find optimal parameters based on the data at hand, using maximum likelihood estimation.

Suppose we are given data, $\mathbf{d} = [d_1, d_2, ..., d_m]^\top$, at input locations $\mathbf{x} = [x_1, x_2, ..., x_m]^\top$. Suppose our mean function has hyperparameters $\boldsymbol{\theta}_1$ and our covariance functions has hyperparameters $\boldsymbol{\theta}_2$. Denote the vector of all hyperparameters, in the mean and covariance function, as $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]^\top$. Recall that our data is modeled through a Gaussian process, thus the distribution of $\mathbf{d}$ is given by

$$\mathbf{d} \sim \mathcal{N}\left(\mu(\mathbf{x}|\boldsymbol{\theta}_1), \kappa(\mathbf{x}, \mathbf{x}|\boldsymbol{\theta}_2)\right).$$

Since $\mathbf{x}$ is fixed, we write the mean and covariance as a function of hyperparameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ respectively,

$$\boldsymbol{\mu}(\mathbf{x}|\boldsymbol{\theta}_1) = \boldsymbol{\mu}(\boldsymbol{\theta}_1) \in \mathbb{R}^m, \quad \boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x}|\boldsymbol{\theta}_2) = \boldsymbol{\Sigma}(\boldsymbol{\theta}_2) \in \mathbb{R}^{m \times m}.$$

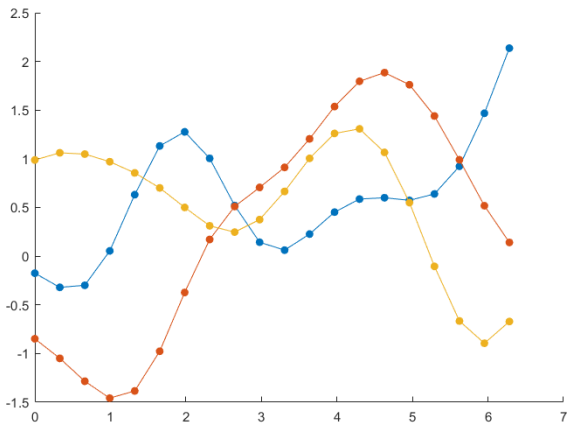

Figure 3: Various samples from our example Gaussian process. Here we display the individual points to emphasize that samples are a discrete set of points.

The log-likelihood of observing our data given hyperparameters $\boldsymbol{\theta}$ is

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2}\log\det(\boldsymbol{\Sigma}(\boldsymbol{\theta}_2)) - \frac{1}{2}(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1))^\top \boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1)) - \frac{m}{2}\log 2\pi.$$

We calculate the hyperparameters by maximizing the likelihood with respect to the hyperparameters $\boldsymbol{\theta}$. Standard optimization methods can be used to optimize the log-likelihood. Efficient optimization schemes require the gradient of the objective function.

We denote $\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}(\boldsymbol{\theta}_2)$ as a matrix in $\mathbb{R}^{n\times n}$ where each element is the partial derivative with respect to θ_j of the kernel function evaluated at $\boldsymbol{\theta}_2$,

$$\left[\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}(\boldsymbol{\theta}_2)\right]_{ik} = \begin{bmatrix} \frac{\partial \kappa}{\partial \theta_j}(\mathbf{x}_1,\mathbf{x}_1,\boldsymbol{\theta}_2) & \dots & \dots & \dots & \frac{\partial \kappa}{\partial \theta_j}(\mathbf{x}_1,\mathbf{x}_n,\boldsymbol{\theta}_2) \\ \vdots & \ddots & & & \vdots \\ \vdots & & \frac{\partial \kappa}{\partial \theta_j}(\mathbf{x}_i,\mathbf{x}_k,\boldsymbol{\theta}_2) & & \vdots \\ \vdots & & & \ddots & \vdots \\ \frac{\partial \kappa}{\partial \theta_j}(\mathbf{x}_n,\mathbf{x}_i,\boldsymbol{\theta}_2) & \dots & \dots & \dots & \frac{\partial \kappa}{\partial \theta_j}(\mathbf{x}_n,\mathbf{x}_n,\boldsymbol{\theta}_2) \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

The jth element of the gradient for the log-likelihood, or the partial derivative with respect to θ_j, of an arbitrary kernel function is given by the following

$$\begin{aligned}\frac{\partial}{\partial \theta_j}\ell(\boldsymbol{\theta}) &= \frac{1}{2}(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1))^\top \boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}\boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1)) - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1))(\mathbf{d}-\boldsymbol{\mu}(\boldsymbol{\theta}_1))^\top \boldsymbol{\Sigma}(\boldsymbol{\theta}_2)^{-1}\frac{\partial \boldsymbol{\Sigma}}{\partial \theta_j}\right).\end{aligned}$$

In the appendix, we list the partial derivatives of particular kernel functions for use in optimization schemes.

2.4 Predictive Distribution

Suppose we are given observations, $\mathbf{d} = [d_1, ..., d_m]^\top \in \mathbb{R}^m$ at inputs $\mathbf{x} = [x_1, .., x_m]^\top \in \mathbb{R}^m$. Our goal is to make predictions of the data at new inputs in our domain, $\mathbf{X} = [X_1, ..., X_M] \in \mathbb{R}^M$. First, choose a mean and kernel function and select hyperparameters to the kernel function.

Let $\mathbf{g} \in \mathbb{R}^M$ be the output values at points $\mathbf{X}$. Following notation from Section 2.2, let

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \boldsymbol{\Sigma}(\mathbf{x}, \mathbf{x}) \in \mathbb{R}^{m \times m}, \qquad \boldsymbol{\Sigma}_{\mathbf{x},\mathbf{X}} = \boldsymbol{\Sigma}(\mathbf{x}, \mathbf{X}) \in \mathbb{R}^{m \times M},$$

$$\boldsymbol{\Sigma}_{\mathbf{X}} = \boldsymbol{\Sigma}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{M \times M}, \qquad \boldsymbol{\Sigma}_{\mathbf{X},\mathbf{x}} = \boldsymbol{\Sigma}(\mathbf{X}, \mathbf{x}) \in \mathbb{R}^{M \times m}.$$

If we model our system as a Gaussian process, then the joint distribution is given by

$$\begin{bmatrix} \mathbf{d} \\ \mathbf{g} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_M \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}} & \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{X}} \\ \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{x}} & \boldsymbol{\Sigma}_{\mathbf{X}} \end{bmatrix} \right). \tag{1}$$

We seek the conditional distribution of $(\mathbf{g}|\mathbf{X}, \mathbf{d}, \mathbf{x})$. We use a commonly known theorem in statistics about joint multivariate normal distributions.

Theorem 2.1 (**Conditional Distribution of Joint MVNs**). *Suppose* $\mathbf{y}$ *follows a multivariate normal distribution with the partitioning,*

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right), \quad \mathbf{y}_1 \in \mathbb{R}^p, \quad \mathbf{y}_2 \in \mathbb{R}^m.$$

Then the conditional distribution of $\mathbf{y}_1$ *conditional on* $\mathbf{y}_2$ *is also multivariate normal with the distribution,*

$$(\mathbf{y}_1|\mathbf{y}_2) \sim \mathcal{N}(\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}).$$

Proof. See Appendix for the proof. ■

Applying Theorem 2.1, the conditional distribution of the new data $\mathbf{g}$ at inputs $\mathbf{X}$, given our data $\mathbf{d}$ is

$$(\mathbf{g}|\mathbf{X}, \mathbf{d}, \mathbf{x}) \sim \mathcal{N}(\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}\mathbf{d}, \boldsymbol{\Sigma}_{\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{X}}). \tag{2}$$

To visualize predictions, we can plot our predictions, $\mathbf{g}$, at time points, $\mathbf{X}$. Additionally at each input, X_i, for $i = 1, ..., M$, we can calculate a pointwise confidence interval using the covariance matrix.

Non-Zero Mean Functions

Recall that we do not need to assume a zero mean function for a Gaussian process. If we have prior information on the data, then we can encode that information into the mean function. Suppose you wish to have assume deterministic mean function $\boldsymbol{\mu}(x)$. Using the same theorem as above, we find that the conditional variance does not change, however the conditional mean changes to the following,

$$\mathbb{E}(\mathbf{g}|\mathbf{X}, \mathbf{d}, \mathbf{x}) = \boldsymbol{\mu}(\mathbf{X}) + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}(\mathbf{d} - \boldsymbol{\mu}(\mathbf{x})). \tag{3}$$

3 Linear Model Uncertainty Quantification

We start with the general notation. We consider a general parameter estimation problem,

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|s(\mathbf{y}(t,\mathbf{u})) - \mathbf{d}\|_2^2 \quad \text{s.t. } \mathbf{y}' = \mathbf{f}(t,\mathbf{y},\mathbf{u}) \tag{4}$$

where $\mathbf{f}$ represents the underlying dynamical model, $\mathbf{y}$ the model output at points t, and $\mathbf{u}$ the parameters of interest in our model. The function $s(\cdot)$ represents the projection of model output to the same space of our observations, $\mathbf{d}$. Sometimes, one is only able to measure or observe a portion of states or model outputs at a limited set of points. We also assume that observations are corrupted with unknown additive measurement noise.

Suppose that our underlying dynamical model is linear with respect to our parameters, $\mathbf{u}$, and the projections of the model output to data is simply the identity map. If we let $\mathbf{A}$ represent the forward model that maps the input parameters to model output, then we have a classical linear inverse, or least squares, problem

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{A}\mathbf{u} - \mathbf{d}\|_2^2 . \tag{5}$$

For our later numerical experiments, we consider the particular problem of function deconvolution in one dimension and computed tomography.

Supposing $\mathbf{A}$ has full column rank, solutions to Equation (5) provide single point estimates, $\hat{\mathbf{u}}$. The field of uncertainty quantification (UQ) is concerned not only with point estimates, but in also determining uncertainty within our estimates. Bayesian inference techniques are common methods for UQ. Let $p(\mathbf{u}|\mathbf{d})$ be the *posterior* density, $p(\mathbf{d}|\mathbf{u})$ the *likelihood* density, $p(\mathbf{u})$ the *prior* density, and $p(\mathbf{d})$ is the *marginal* density. Then Bayes Theorem states,

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u}) \cdot p(\mathbf{u})}{p(\mathbf{d})}, \text{ or omitting the marginal, } p(\mathbf{u}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{u}) \cdot p(\mathbf{u}). \tag{6}$$

While the likelihood depends on the underlying statistical model, the choice of prior is determined by prior knowledge of the user. This prior knowledge could include initial guesses of the distribution of $\mathbf{u}$ or assumptions on the possible range of parameters. In this way, the prior distribution acts as a regularization of our solutions. Details for a basic Bayesian approach are left in a later section.

For particular choices of likelihood and prior distributions, the posterior can be easily derived. For more general distributions, one is only able to take samples of the posterior distribution. Without knowledge of the support of the posterior density, it would be naive to calculate a grid of samples of the posterior distribution. In these situations, one may use Markov Chain Monte Carlo (MCMC) methods to sample the posterior distribution. Examples of MCMC methods include the adaptive Metropolis algorithm given as Algorithm 1. MCMC methods are often computationally intensive; Markov chains are serial in nature. To parallelize MCMC techniques, at best, one can compute separate chains in parallel.

Algorithm 1 Adaptive Metropolis

Require: p_{like}, p_{prior}, p_{prop}, $\mathbf{d}$, $\mathbf{u}_0$

$i = 0$, compute posterior $p_{\text{post}}(\mathbf{u}_0|\mathbf{d})$
while not done **do**
  $\mathbf{u}_{\text{prop}} \sim \mathcal{N}(\mathbf{u}_i, \mathbf{C}_i)$
  compute posterior $p_{\text{post}}(\mathbf{u}_{\text{prop}}|\mathbf{d})$
  compute $c = \min\left(1, \frac{p_{\text{post}}(\mathbf{u}_{\text{prop}}|\mathbf{d})}{p_{\text{post}}(\mathbf{u}_i|\mathbf{d})}\right)$
  sample $u \sim \mathcal{U}([0,1])$
  if $u < c$ **then**
    $\mathbf{u}_{i+1} = \mathbf{u}_{\text{prop}}$
  else
    $\mathbf{u}_{i+1} = \mathbf{u}_i$
  end if
  update $\mathbf{C}_i \to \mathbf{C}_{i+1}$ using $\mathbf{u}_{i+1}$
  $i = i + 1$
end while

Ensure: $\{\mathbf{u}_i\}_{i=1}^K$ samples from posterior

Chung et al.[12] proposed a new approach to Bayesian UQ by replacing the data with surrogate data from an appropriate stochastic process such as a Gaussian process. By taking samples from the Gaussian process, we generate sample paths of our dynamical model. Then we perform standard parameter estimation techniques on the samples rather than the data, which can allow one to find a distribution of parameter estimates. Even without knowledge of the support of the resulting posterior distribution, we can perform uncertainty quantification by finding the distribution of our parameter estimates with respect to GP samples. The algorithm from Chung et al. is outlined in Algorithm 2.

Algorithm 2 UQ using GPs

Require: GP kernel function $\kappa(\cdot,\cdot)$, GP mean function $\mu(\cdot)$, data $\mathbf{d}$, input locations of data $\mathbf{t}$, input locations of GP approximation $\mathbf{T}$, number of samples K

Compute hyperparameters $\hat{\boldsymbol{\theta}} = \max_{\mathbf{u}} \ell(\boldsymbol{\theta})$
Compute $\boldsymbol{\Sigma}_{\mathbf{t}}$,$\boldsymbol{\Sigma}_{\mathbf{T}}$, $\boldsymbol{\Sigma}_{\mathbf{T},\mathbf{t}}$
Compute $\boldsymbol{\mu} \leftarrow \boldsymbol{\Sigma}_{\mathbf{Tt}}\boldsymbol{\Sigma}_{\mathbf{t}}^{-1}\mathbf{d}$
Compute $\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma}_{\mathbf{T}} - \boldsymbol{\Sigma}_{\mathbf{Tt}}\boldsymbol{\Sigma}_{\mathbf{t}}^{-1}\boldsymbol{\Sigma}_{\mathbf{tT}}$
for k = 1, 2, ..., K **do**
  Sample $\mathbf{g}^{(k)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  Compute $\hat{\mathbf{u}}^{(k)} \leftarrow \min_{\mathbf{u}} \left\|\mathbf{g}^{(k)} - s(\mathbf{y}(\mathbf{T}|\mathbf{u}))\right\|_2^2$
end for

Ensure: $\left\{\mathbf{u}^{(k)}\right\}_{i=1}^K$ samples of parameter estimates

In Section 3.2, we consider Algorithm 2 with linear models and provide a closed form sampling distribution of the resulting estimator. In Section 3.3 we show that using a surrogate GP actually corresponds to solving a weighted least squares problem and provide the corresponding maximum likelihood estimator. In Section 3.4, we extend results from Section 3.3 to include regularization with a prior distribution and provide the closed form posterior distribution.

3.1 Linear Models, Ordinary Least Squares Estimator, and Bayesian MAP Estimator

Suppose our forward model is a linear model and the projection of the model output to data is simply the identity map. Then our model output is given by

$$s(\mathbf{y}(\mathbf{t}|\mathbf{u})) = \mathbf{A}\mathbf{u},$$

where $\mathbf{A}$ represents the forward model. If we assume observations also contain Gaussian additive noise, then our statistical model is given by,

$$\mathbf{d} = \mathbf{A}\mathbf{u} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon),$$

where $\boldsymbol{\Sigma}_\epsilon$ is the covariance matrix of the noise.

Assume $\mathbf{A}$ has full column rank, then the solution to the least squares problem is given by,

$$\hat{\mathbf{u}} = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{d}. \tag{7}$$

For a frequentist approach, the given statistical model gives us the sampling distribution of the least squares solution as

$$\hat{\mathbf{u}}_{\text{LS}} \sim \mathcal{N}(\mathbf{u}, (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \boldsymbol{\Sigma}_\epsilon \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}) \tag{8}$$

For a Bayesian approach, we use Bayes rule with a likelihood given by the statistical model and a given prior distribution. A basic prior distribution is given by

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \lambda\mathbf{I}), \quad \lambda > \mathbf{0}$$

and the likelihood is given by

$$\mathbf{d}|\mathbf{u} \sim \mathcal{N}(\mathbf{A}\mathbf{u}, \boldsymbol{\Sigma}_\epsilon).$$

Together, the posterior distribution will be normal and is given by

$$\mathbf{u}|\mathbf{d} \sim \mathcal{N}((\mathbf{A}^\top \boldsymbol{\Sigma}_\epsilon \mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^\top \boldsymbol{\Sigma}_\epsilon \mathbf{d}, (\mathbf{A}^\top \boldsymbol{\Sigma}_\epsilon \mathbf{A} + \lambda\mathbf{I})^{-1}). \tag{9}$$

The Bayesian maximum a posteriori (MAP) estimate in this case is the same as the posterior mean.

3.2 Ordinary Least Squares Estimator with GP Surrogate

Suppose we have a linear parameter estimation problem, as above,

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{A}\mathbf{u} - \mathbf{d}\|_2^2, \quad \mathbf{u} \in \mathbb{R}^n, \quad \mathbf{d} \in \mathbb{R}^m, \quad \mathbf{A} \in \mathbb{R}^{m\times n}.$$

Given observations, $\mathbf{d}$, we generate a Gaussian process, potentially with a different model design. Samples from the GP have the following distribution,

$$\mathbf{g}|\mathbf{d} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Sigma}_\mathbf{g}), \quad \mathbf{g}, \boldsymbol{\mu}_\mathbf{g} \in \mathbb{R}^M, \boldsymbol{\Sigma}_\mathbf{g} \in \mathbb{R}^{M\times M}$$

with $\boldsymbol{\mu}_\mathbf{g}$ and $\boldsymbol{\Sigma}_\mathbf{g}$ defined in the previous section. We define a new model matrix, $\mathbf{A}_\mathbf{g} \in \mathbb{R}^{M\times n}$ that represents the forward model with the new model design of Gaussian process surrogate data. If the design of $\mathbf{g}$ is the same as the original observations, then $\mathbf{A}_\mathbf{g} = \mathbf{A}$. We replace the data with a realization of the Gaussian process, $\mathbf{g}^{(k)}$. The least square problem with the surrogate data is given by

$$\hat{\mathbf{u}}^{(k)} = \arg\min_{\mathbf{u}} \|\mathbf{A}_\mathbf{g}\mathbf{u} - \mathbf{g}\|_2^2$$

with solution,

$$\hat{\mathbf{u}}^{(k)} = (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \mathbf{g}^{(k)}.$$

We also wish to quantify the uncertainty of our estimates. In our case, we seek the sampling distribution of $\hat{\mathbf{u}}^{(k)}$. We note that $\mathbf{g}^{(k)}$ follows a multivariate normal distribution and $(\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top$ is deterministic. We now provide a closed form of the posterior distribution of $\hat{\mathbf{u}}^{(k)}$.

Theorem 3.1 (Sampling Distribution of Least Squares Estimator)**.** *Suppose* $\mathbf{g} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Sigma}_\mathbf{g})$*, the matrix* $\mathbf{A}_\mathbf{g}$ *has full rank and deterministic. If*

$$\hat{\mathbf{u}} = (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \mathbf{g},$$

then $\hat{\mathbf{u}}$ *follows the distribution,*

$$\begin{aligned}\hat{\mathbf{u}} \sim \mathcal{N}((\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \boldsymbol{\mu}_\mathbf{g}, \\ (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \boldsymbol{\Sigma}_\mathbf{g}\mathbf{A}_\mathbf{g}^\top (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}).\end{aligned} \tag{10}$$

Proof. We approximate the data, $\mathbf{d}$, with a Gaussian process for some prior mean and covariance functions, $\mu(\cdot)$ and $\kappa(\cdot,\cdot)$. Then the posterior conditional distribution of $\mathbf{g}$ is given by

$$\mathbf{g}|\mathbf{d} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{g}, \boldsymbol{\Sigma}_\mathbf{g})$$

We assume $\mathbf{g}$ is computed at design points $\mathbf{T} \in \mathbb{R}^M$. Then we form a new forward model matrix, $\mathbf{A}_\mathbf{g} \in \mathbb{R}^{M\times n}$. We assume the new model matrix, $\mathbf{A}_\mathbf{g}$, corresponding to the design of the Gaussian process. The solution to the least squares problem,

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} ||\mathbf{A}_\mathbf{g}\mathbf{u} - \mathbf{g}||_2^2$$

is given by,

$$\hat{\mathbf{u}} = (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \mathbf{g}.$$

Here, $(\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top$ is a deterministic a linear transformation. Since $\mathbf{g}$ is normal, $\hat{\mathbf{u}}$ is also normal. Additionally, the distribution of a linear transformation of a multivariate normal random variable is known and is given by,

$$\begin{aligned}\hat{\mathbf{u}} \sim \mathcal{N}((\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \boldsymbol{\mu}_\mathbf{g}, \\ (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}\mathbf{A}_\mathbf{g}^\top \boldsymbol{\Sigma}_\mathbf{g}\mathbf{A}_\mathbf{g}^\top (\mathbf{A}_\mathbf{g}^\top \mathbf{A}_\mathbf{g})^{-1}).\end{aligned}$$

■

In our numerical results, we refer to these technique as LSGP (least squares with Gaussian process surrogate).

3.3 Maximum Likelihood Estimator and Weighted Least Squares

Instead of solving the least squares problem, we look at finding the maximum likelihood estimator. In the typical context of multiple linear regression, the least squares estimator and the maximum likelihood estimator are equivalent. However, in the context of a Surrogate Gaussian process, we can include information from the posterior Gaussian process covariance matrix.

Using the same notation as in the previous section, we are given a linear model and approximate the data, $\mathbf{d}$, with a Gaussian process, with some mean and covariance function, to obtain a posterior

conditional distribution of $\mathbf{g}$. Since the model design can differ with our Gaussian process, we again define a new model matrix $\mathbf{A_g} \in \mathbb{R}^{M\times n}$ that represents the forward model with the new design of observations. Now we define the misfit between the GP and the model output, $\boldsymbol{\delta}(\mathbf{u})$,

$$\boldsymbol{\delta}(\mathbf{u}) = \mathbf{A_g}\mathbf{u} - \mathbf{g}.$$

To find parameter estimates for $\mathbf{u}$, we find the maximum likelihood estimator. The distribution of $\boldsymbol{\delta}$ conditional on $\mathbf{u}$ is multivariate normal with,

$$\boldsymbol{\delta}|\mathbf{u} \sim \mathcal{N}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g}, \boldsymbol{\Sigma_g}).$$

The likelihood function is given by

$$L(\mathbf{u}|\boldsymbol{\delta}) \propto \exp\left(-\frac{1}{2}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g})\right).$$

Theorem 3.2. *The maximum likelihood estimator is equivalent to solving the following weighted least squares problem*

$$\hat{\mathbf{u}} = \min_{\mathbf{u}} ||\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g}||^2_{\boldsymbol{\Sigma_g}^{-1}} \tag{11}$$

and is given by

$$\hat{\mathbf{u}} = (\mathbf{A_g}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A_g})^{-1}\mathbf{A_g}^\top \boldsymbol{\Sigma_g}^{-1}\boldsymbol{\mu_g}. \tag{12}$$

Proof. To find the maximum likelihood estimate, we maximize the equation for the likelihood with respect to $\mathbf{u}$.

$$\begin{aligned}\hat{\mathbf{u}} &= \max_{\mathbf{u}} \exp\left(-\frac{1}{2}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g})\right) \\ &\Leftrightarrow \min_{\mathbf{u}} \frac{1}{2}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g}) = \min_{\mathbf{u}} ||\mathbf{A_g}\mathbf{u} - \boldsymbol{\mu_g}||^2_{\boldsymbol{\Sigma_g}^{-1}}.\end{aligned}$$

We actually find that this is equivalent to solving a weighted least squares problem with the weights given by the posterior covariance matrix, $\boldsymbol{\Sigma_g}$. The solution to the weighted least squares problem, and thus the maximum likelihood estimate, is given by

$$\hat{\mathbf{u}} = (\mathbf{A_g}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A_g})^{-1}\mathbf{A_g}^\top \boldsymbol{\Sigma_g}^{-1}\boldsymbol{\mu_g}$$

■

3.4 Bayesian Inference: MAP with GP Surrogate

For a Bayesian approach, one can set a prior distribution on $\mathbf{u}$ and find a maximum a posteriori (MAP) estimate.

We use the same notation as above, given a linear model, we compute a posterior Gaussian process on the observations. We also generate a new model matrix $\mathbf{A_g} \in \mathbb{R}^{M\times n}$ that represents the forward model with the new design of observations. Bayes rule tells us,

$$p(\mathbf{u}|\boldsymbol{\delta}) \propto p(\boldsymbol{\delta}|\mathbf{u})p(\mathbf{u})$$

The likelihood is the same as in the previous section. We define the misfit between the GP and the model output, $\boldsymbol{\delta}(\mathbf{u})$,

$$\boldsymbol{\delta}(\mathbf{u}) = \mathbf{A_g}\mathbf{u} - \mathbf{g}$$

with distribution

$$\boldsymbol{\delta}|\mathbf{u} \sim \mathcal{N}(\mathbf{A_g u} - \boldsymbol{\mu_g}, \boldsymbol{\Sigma_g}).$$

The resulting density is

$$p(\boldsymbol{\delta}|\mathbf{u}) \propto \exp\left(-\frac{1}{2}(\mathbf{A_g u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A_g u} - \boldsymbol{\mu_g})\right).$$

We specify a Gaussian prior on $\mathbf{u}$, with general mean and covariance matrix,

$$\mathbf{u} \sim \mathcal{N}(\boldsymbol{\mu_u}, \boldsymbol{\Sigma_u}), \quad p(\mathbf{u}) \propto \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu_u})^\top \boldsymbol{\Sigma_u}^{-1}(\mathbf{u} - \boldsymbol{\mu_u})\right).$$

This prior has the effect of smoothing inversions. Then the posterior density is proportional to the product,

$$\begin{aligned} p(\mathbf{u}|\boldsymbol{\delta}) &\propto p(\boldsymbol{\delta}|\mathbf{u})p(\mathbf{u}) \\ &\propto \exp\left(-\frac{1}{2}(\mathbf{A_g u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A_g u} - \boldsymbol{\mu_g})\right) \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu_u})^\top \boldsymbol{\Sigma_u}^{-1}(\mathbf{u} - \boldsymbol{\mu_u})\right) \\ &\propto \exp\left(-\frac{1}{2}\left[(\mathbf{A_g u} - \boldsymbol{\mu_g})^\top \boldsymbol{\Sigma_g}^{-1}(\mathbf{A u} - \boldsymbol{\mu_g}) + (\mathbf{u} - \boldsymbol{\mu_u})^\top \boldsymbol{\Sigma_u}^{-1}(\mathbf{u} - \boldsymbol{\mu_u})\right]\right). \end{aligned}$$

Since the product involves Gaussian densities, then we can write the posterior as a Gaussian distribution. Similar techniques to derive the posterior mean and covariance are shown, without derivation, in [15, 16].

Theorem 3.3 (Posterior Gaussian Distribution)**.** *The posterior distribution is given by*

$$\mathbf{u}|\boldsymbol{\delta} \sim \mathcal{N}(\boldsymbol{\mu}_{Post}, \boldsymbol{\Sigma}_{Post})$$

with the following mean and covariance matrix,

$$\boldsymbol{\mu}_{Post} = \boldsymbol{\mu_u} + (\mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A} + \boldsymbol{\Sigma_u}^{-1})^{-1}\mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}(\boldsymbol{\mu_g} - \mathbf{A}\boldsymbol{\mu_u}), \quad \boldsymbol{\Sigma}_{Post} = (\mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A} + \boldsymbol{\Sigma_u}^{-1})^{-1}. \quad (13)$$

Proof. We write both the likelihood and the prior as proportional to another Gaussian in canonical form with respect to $\mathbf{u}$. More details on the canonical form of a Gaussian is given in Appendix D.

$$\begin{aligned} \exp\left(-\frac{1}{2}(\boldsymbol{\mu_g} - \mathbf{A u})^\top \boldsymbol{\Sigma_g}^{-1}(\boldsymbol{\mu_g} - \mathbf{A u})\right) &\propto \exp\left(-\frac{1}{2}[\boldsymbol{\mu_g}^\top \boldsymbol{\Sigma_g}^{-1}\boldsymbol{\mu_g} - 2\boldsymbol{\mu_g}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A u} + \mathbf{u}^\top \mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A u}]\right) \\ &\propto \exp\left(\alpha_1 + \boldsymbol{\eta}_1^\top \mathbf{u} - \frac{1}{2}\mathbf{u}^\top \mathbf{P}_1 \mathbf{u}\right) \end{aligned}$$

with $\boldsymbol{\eta}_1 = \mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}\boldsymbol{\mu_g}$, $\mathbf{P}_1 = \mathbf{A}^\top \boldsymbol{\Sigma_g}^{-1}\mathbf{A}$, and α_1 as a negligible constant.

$$\begin{aligned} \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu_u})^\top \boldsymbol{\Sigma_u}^{-1}(\mathbf{u} - \boldsymbol{\mu_u})\right) &\propto \exp\left(-\frac{1}{2}[\boldsymbol{\mu_u}^\top \boldsymbol{\Sigma_u}^{-1}\boldsymbol{\mu_u} - 2\boldsymbol{\mu_u}^\top \boldsymbol{\Sigma_u}^{-1}\mathbf{u} + \mathbf{u}^\top \boldsymbol{\Sigma_u}^{-1}\mathbf{u}]\right) \\ &\propto \exp\left(\alpha_2 + \boldsymbol{\eta}_2^\top \mathbf{u} - \frac{1}{2}\mathbf{u}^\top \mathbf{P}_2 \mathbf{u}\right) \end{aligned}$$

with $\boldsymbol{\eta}_2 = \boldsymbol{\Sigma_u}^{-1}\boldsymbol{\mu_u}$, $\mathbf{P}_2 = \boldsymbol{\Sigma_u}^{-1}$, α_2 as a negligible constant.

Now the product of the Gaussians can be written as the following,

$$\begin{aligned}
\text{(Likelihood)} \times \text{(Prior)} &\propto \exp\left(-\frac{1}{2}(\mathbf{A}\mathbf{u}-\boldsymbol{\mu}_{\mathbf{g}})^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}(\mathbf{A}\mathbf{u}-\boldsymbol{\mu}_{\mathbf{g}})\right)\exp\left(-\frac{1}{2}(\mathbf{u}-\boldsymbol{\mu}_{\mathbf{u}})^{\top}\boldsymbol{\Sigma}_{\mathbf{u}}^{-1}(\mathbf{u}-\boldsymbol{\mu}_{\mathbf{u}})\right)\\
&\propto \exp\left(\alpha_1+\boldsymbol{\eta}_1^{\top}\mathbf{u}-\frac{1}{2}\mathbf{u}^{\top}\mathbf{P}_1\mathbf{u}\right)\exp\left(\alpha_2+\boldsymbol{\eta}_2^{\top}\mathbf{u}-\frac{1}{2}\mathbf{u}^{\top}\mathbf{P}_2\mathbf{u}\right)\\
&= \exp\left((\alpha_1+\alpha_2)+(\boldsymbol{\eta}_1+\boldsymbol{\eta}_2)^{\top}\mathbf{u}-\frac{1}{2}\mathbf{u}^{\top}(\mathbf{P}_1+\mathbf{P}_2)\mathbf{u}\right)\\
&= \exp\left(-\frac{1}{2}[\alpha_3+\boldsymbol{\eta}_3^{\top}\mathbf{u}-\frac{1}{2}\mathbf{u}^{\top}\mathbf{P}_3\mathbf{u}]\right)
\end{aligned}$$

with $\boldsymbol{\eta}_3=\boldsymbol{\eta}_1+\boldsymbol{\eta}_2$, $\mathbf{P}_3=\mathbf{P}_1+\mathbf{P}_2$, and $\alpha_3=\alpha_1+\alpha_3$ as another negligible constant.

Now in canonical form, one can find the mean and variance of the resulting Gaussian density. The covariance is given by

$$\begin{aligned}
\boldsymbol{\Sigma}_{\text{Post}} &= \mathbf{P}_3^{-1}\\
&= (\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})^{-1}.
\end{aligned}$$

The mean is related to canonical form by following,

$$\begin{aligned}
\mathbf{P}_3\boldsymbol{\mu}_{\text{Post}} &= \boldsymbol{\eta}_3\\
\implies \boldsymbol{\mu}_{\text{Post}} &= \mathbf{P}_3^{-1}\boldsymbol{\eta}_3\\
&= (\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})^{-1}(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\boldsymbol{\mu}_{\mathbf{g}}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1}\boldsymbol{\mu}_{\mathbf{u}})\\
&= (\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})^{-1}(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\boldsymbol{\mu}_{\mathbf{g}}-\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}\boldsymbol{\mu}_{\mathbf{u}}+\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}\boldsymbol{\mu}_{\mathbf{u}}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1}\boldsymbol{\mu}_{\mathbf{u}})\\
&= (\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})^{-1}[\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}(\boldsymbol{\mu}_g-\mathbf{A}\boldsymbol{\mu}_{\mathbf{u}})+(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})\boldsymbol{\mu}_{\mathbf{u}}]\\
&= \boldsymbol{\mu}_{\mathbf{u}}+(\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1})^{-1}\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}(\boldsymbol{\mu}_{\mathbf{g}}-\mathbf{A}\boldsymbol{\mu}_{\mathbf{u}})
\end{aligned}$$

■

Any number of choices exist for the mean vector of the prior. Generally, the choice of the prior covariance matrix is without restriction. However, now we require both $\boldsymbol{\Sigma}_{\mathbf{u}}$ and the matrix $\mathbf{A}^{\top}\boldsymbol{\Sigma}_{\mathbf{g}}^{-1}\mathbf{A}+\boldsymbol{\Sigma}_{\mathbf{u}}^{-1}$ to be invertible. One can see that $\boldsymbol{\Sigma}_{\mathbf{u}}$ represents the regularization we impose on $\mathbf{u}$. The posterior mean can actually be broken down into different components. We update the prior mean, $\boldsymbol{\mu}_{\mathbf{u}}$ with a weighted least squares solutions with regularization. Instead of the weighted least squares being performed on the original data, it is instead performed on the difference between the posterior mean of the surrogate GP and the prior mean, $\boldsymbol{\mu}_{\mathbf{g}}-\mathbf{A}\boldsymbol{\mu}_{\mathbf{u}}$.

4 Applications

4.1 Deconvolution in 1D

Suppose the relationship between an input, u, and model output, $y(s|u)$, is given by a convolution with a given kernel, a,

$$y(s|u) = (a \star u)(s) = \int_{-\infty}^{\infty} a(s-t)u(t)\ \mathrm{d}t. \tag{14}$$

Deconvolution is the process of reconstructing an unknown input function, u, given a true kernel, a and noisy measured observations, $\mathbf{d} = [d_1, ..., d_n]^\top$, of model output. By using a quadrature rule to approximate the integral, often one is led to the typical linear inverse problem,

$$\mathbf{d} = \mathbf{A}\mathbf{u} + \boldsymbol{\epsilon}, \tag{15}$$

where $\mathbf{A}$ and $\mathbf{u}$ represent the quadrature rule and $\boldsymbol{\epsilon}$ represents the noise in observations. Note, in this context, a kernel for convolution is not the same as the kernel used in a Gaussian process, as stated in previous sections.

We follow Example 1.2 from [17] and consider a true function, or signal, $u(t)$, defined on $\mathbb{R}$ with support on $[0, 1]$. We assume a zero boundary condition at $u(0)$ and $u(1)$. Data generated from the true signal is also normalized for numerical experiments. Suppose the model output is generated by a convolution process,

$$y(s|u) = \int_{-\infty}^{\infty} a(s-t)u(t)\ \mathrm{d}t, \quad s \in [0, 1].$$

In our numerical tests, we use for $a(\cdot)$, a Gaussian kernel with support on $[-1, 1]$,

$$a(r) = \begin{cases} \frac{1}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{r^2}{2\gamma^2}\right), & \text{for } r \in [-1, 1], \\ 0, & \text{otherwise.} \end{cases}$$

Now our integral reduces to

$$y(s|u) = \int_{-1}^{2} a(s-t)u(t)\ \mathrm{d}t, \quad s \in [0, 1].$$

We approximate the integral through the midpoint method. Suppose we consider a uniform grid on $[0, 1]$ with $n + 1$ points. Let $t_j = jh$, $j = 0, 1, 2, ..., n$ with $h = \frac{1}{n}$. With the midpoint rule, we calculate the integrand evaluated on $t'_j = (j - \frac{1}{2})h$, for $j = 1, 2, ..., n$. Then the integral is given by

$$y(s|u) = \int_{-1}^{2} a(s-t)u(t)\ \mathrm{d}t = \sum_{j=1}^{n} a(s-t'_j)u(t'_j)h + E_n, \tag{16}$$

where E_n is the quadrature error due to the midpoint rule. If we use the same grid for the sampling of observations, $s'_i = t'_i$, with $i = 1, 2, ..., n$, then we find the following system of equations,

$$y(s'_i|u) \approx h\sum_{j=1}^{n} a(s'_i - t'_j)u(t'_j) = h\sum_{j=1}^{n} a((i-j)h)u(t'_j), \quad i = 1, 2, ..., n.$$

If we define our unknown parameters to be $\mathbf{u} = [u(t'_1), ..., u(t'_n)]^\top$, our model matrix to be $[\mathbf{A}]_{ij} = ha((i-j)h)$, and our observations to be $d_i = d(s'_i) = y(s'_i|u) + \epsilon$, then the above system of equations can be written in matrix-vector form as $\mathbf{d} = \mathbf{A}\mathbf{u}$.

Given observations, $\mathbf{d} = [d(s_1'), ..., d(s_n')]^\top$, we approximate $d(s)$ using a Gaussian process evaluated at a potentially different set of points S_i', for $i = 1, 2, ..., M$. One should consider an appropriate kernel function to measure distance between two observations. We investigate this question in the proceeding numerical experiments by comparing the squared exponential and Matérn kernels.

After computing a surrogate Gaussian process on the data, $\mathbf{g} \in \mathbb{R}^N$, we now need to consider a new model matrix, $\mathbf{A_g}$, that corresponds to the new model design, S_i', for $i = 1, .., M$ that must be of full column rank. Recalling from the midpoint rule in (16), if we assume the design S_i' is uniform on the interval $[0, 1]$, we simply compute the model matrix as

$$[\mathbf{A_g}]_{ij} = ha(S_i' - t_j'), \quad \mathbf{A_g} \in \mathbb{R}^{M \times n}.$$

Once $\mathbf{A_g}$ is generated, we follow the formulas from Section 3 to compute the posterior distributions of the inverted signal.

To illustrate the effect of the choice of the GP kernel, we consider three different input signals. For the kernel function used to blur our true signals, we use $\gamma = 0.03$. The true signal, $\mathbf{u}$, is then sampled uniformly on $[0, 1]$ with $n = 30$ points, and normalized. To simulate observations, We form the model matrix, $\mathbf{A}$, generate the true model output $\mathbf{Au}$, and add noise. The noise vector, $\mathbf{e}$, is generated from a Gaussian distribution such that $\|\mathbf{e}\|_2 = 0.05 \|\mathbf{Au}\|_2$. Then following (5), we generate the observations, $\mathbf{d}$.

For comparisons, we consider multiple methods for inversion. We compute the ordinary least squares estimate (OLS) from (7), an ordinary least squares estimate using a GP Surrogate (OLSGP) from (10), and Maximum Likelihood Estimate using a GP surrogate (MLEGP) from (12). For the Bayesian approaches, we will use a standard Bayesian MAP Estimate (Bayes) from (9) and a Bayesian MAP Estimate using a GP Surrogate (BayesGP) from (13). For both Bayesian methods, we will use the similar prior distributions,

$$\mathbf{x} \sim \mathcal{N}_n(\mathbf{0}, \lambda \mathbf{I}).$$

For simplicity, to determine λ, we perform a direct search in the interval $[10^{-5}, 10^2]$ that minimizes the relative error between the posterior MAP estimate and the true solution. In the following sections, we investigate the inversions performed on various underlying functions, a C^∞ function, a step function, and a function that is the mixture of both.

C^∞ Input Function

First we consider a C^∞ input function,

$$f_1(t) = \begin{cases} \sin^4(2\pi t), & \text{for } 0 < t < 1, \\ 0, & \text{otherwise.} \end{cases}$$

To compute a Gaussian process on the noisy blurred observations, we consider a smooth kernel, the squared exponential kernel with a term for a nugget. Hyperparameters are computed using maximum likelihood. The true signal, blurred observations, and posterior GP are shown in Figure 4.

We perform various inversions in Figure 5 to compute posterior distributions of our true signals. We note that the least squares solutions performs relatively poorly. The other three methods have very competitive estimates for the true signal. We provide relative error tables in Table 1. For this example, we see that both MLEGP and the posterior mean of BayesGP outperform least squares and a Bayesian MAP estimate.

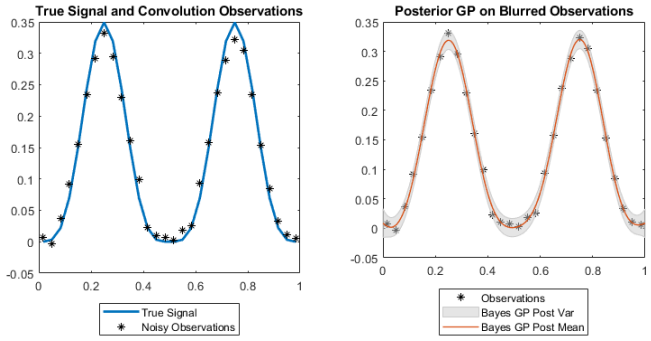

Figure 4: The true input signal is shown on the left plot. The blurred observations along with a posterior Gaussian process using a squared exponential kernel with a nugget.

In UQ, we also consider the variance in our estimates. We show the posterior means and variances of both Bayesian methods in Figure 5. While the posterior means of both approaches are relatively similar, we can see that the Bayesian approach with a GP surrogate provides much lower variance in our solutions.

The previous results used a kernel that was appropriate for our observations. To illustrate the importance of the choice of kernel function, we now look at inversions using a poor choice of kernel function. In this case, we consider a Matérn kernel with $\nu = 2$ without a nugget term. We note that ν determines the differentiability of the kernel function; with $\nu = 2$, the kernel is 1 times differentiable in the mean square sense.

In Figure 6, we compare the two kernels. Without a nugget, there is less variance in our posterior GP. We would expect our GP to have no variance at isolated observations. With this posterior GP, we see our inversions are less smooth. We observe this in both the MLEGP and the posterior mean of BayesGP.

Upon further investigation, the cause of the rough inversions can be attributed to the lack of a nugget. Gramacy et al. argue for the inclusion of a nugget term in nearly all cases of mathematical modeling with GPs [18]. It is no surprise then, that failing to include the nugget when our observations are known to have noise results in poor inversions. In Figure 7, we do a similar comparison, this time, instead, we include the nugget term with the Matérn 2 kernel. Here we see comparable inversions.

Step Input Function

Next we consider a step function with two separated steps,

$$f_2(t) = \begin{cases} 1, & \text{for } 0.2 < t < 0.4, \\ 1, & \text{for } 0.6 < t < 0.8, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly as before, we compute a posterior GP on our blurred observations. This time, we use a Matérn Kernel with $\nu = 2$ and a nugget. We compute various inversions in Figure 8. We note that our inversions are not smooth. In fact, at the jumps, we observe oscillations reminiscent of Gibbs's Phenomenon. We can also observe a relationship between MLEGP and BayesGP mentioned earlier: BayesGP can be seen as MLEGP with added regularization from a prior distribution.

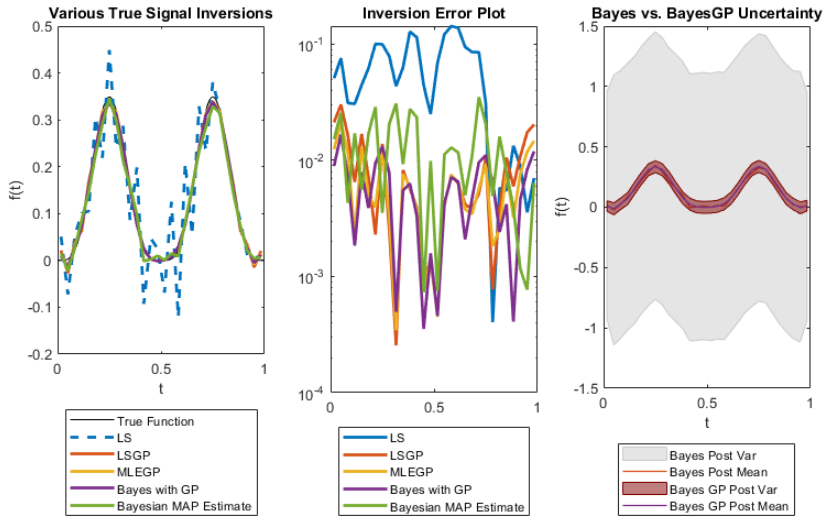

Figure 5: The true signal of the C^∞ function and four different inversions. Both Bayesian techniques plot the posterior mean. As expected, the standard least squares solution performs relatively poor. On the right, we compare the variances between the purely Bayesian and Bayesian with GP surrogate and note the much smaller variance.

Relative errors are provided in Table 1. Additionally, while we see the posterior means of Bayes and BayesGP overlap, one observes that the variance of BayesGP outperforms the typical Bayesian approach.

We investigated with the Matérn Kernel first since we hypothesized that the blurred observations of step function may be better approximated from the Matérn than the squared exponential, which enforces smoothness. We compare results between the Matérn kernel and the squared exponential kernel in Figure 9. When comparing just MLEGP, we still observe oscillations around the jumps. However, the magnitude of oscillations are slightly smaller. After some regularization in BayesGP, we can see that the Squared Exponential relatively comparable. When we look into the relative error of the posterior means, however, BayesGP with the squared exponential loses out to the Matérn Kernel. With the Squared Exponential, the relative error of the posterior mean of BayesGP is 1.793×10^{-1}, just a little over the relative error of the Matérn Kernel of 1.397×10^{-1}.

Composite Function

Finally, we consider a function with both C^∞ and step segments, and try to observe the effect of trying to invert the signal with the different kernel functions. The true signal is given by,

$$f_3(t) = \begin{cases} 1, & \text{for } 0.1 < t < 0.25, \\ 0.5, & \text{for } 0.3 < t < 0.4, \\ \sin^4(2\pi t), & \text{for } 0.5 < t < 1, \\ 0, & \text{otherwise.} \end{cases}$$

We consider the use of just the Squared Exponential kernel with a nugget. We compute a posterior GP on the blurred observations, again using maximum likelihood to determine optimal

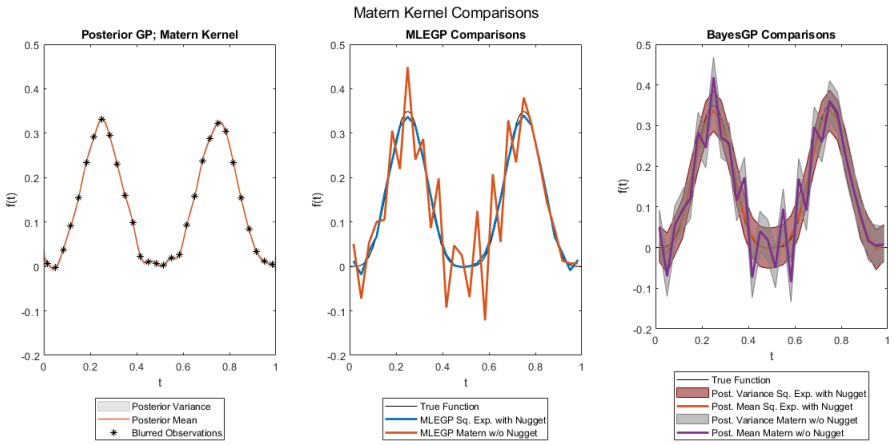

Figure 6: Comparison between the squared exponential kernel with nugget and a $\nu = 2$ Matérn kernel without a nugget on the C^∞ function inversions. Note the less smooth MLEGP and BayesGP inversions. This is actually due to the lack of the nugget rather than the use of the Matérn kernel.

	C^∞ Function	Step Function	Composite Function
Least Squares	4.056×10^{-1}	3.860×10^{-1}	3.846×10^{-1}
Least Squares GP	6.014×10^{-2}	2.789×10^{-1}	3.198×10^{-1}
MLE GP	4.481×10^{-2}	2.921×10^{-1}	2.277×10^{-1}
Bayes GP	3.941×10^{-2}	1.397×10^{-1}	1.975×10^{-1}
Bayes	8.858×10^{-2}	1.405×10^{-1}	1.594×10^{-1}
Bayes Median Variance	4.529×10^{-1}	1.040×10^{0}	8.116×10^{-1}
Bayes GP Median Variance	9.095×10^{-4}	5.179×10^{-4}	5.614×10^{-4}

Table 1: Relative error rates and median variance of the three functions with different inversions. The squared exponential kernel with nugget were used for both the C^∞ and composite function. A Matérn kernel with nugget was used for the step function.

hyperparameters.

We look at various inversions in Figure 10 and a relative error table is given in Table 1. Inversions provide encouraging results, and the posterior distribution with BayesGP provides smaller variance in our inversion of the true signal, shown in Figure 10. Similarly as before, we also test inversions with the Matérn kernel with $\nu = 2$ and a nugget. The comparisons are given in Figure 11.

The squared exponential kernel outperforms the Matérn kernel in the MLEGP inversion, with relative errors of 2.277×10^{-1} and 3.124×10^{-1} respectively. However, the opposite is true for the BayesGP inversion, with relative errors of 1.975×10^{-1} and 1.596×10^{-1}. While the presence of both a step function and a C^∞ function encourages us to use a kernel function that can approximate both properties, we cannot definitively conclude that a particular kernel is suited for particular functions. The squared exponential kernel shows promising results for most of the deconvolution examples.

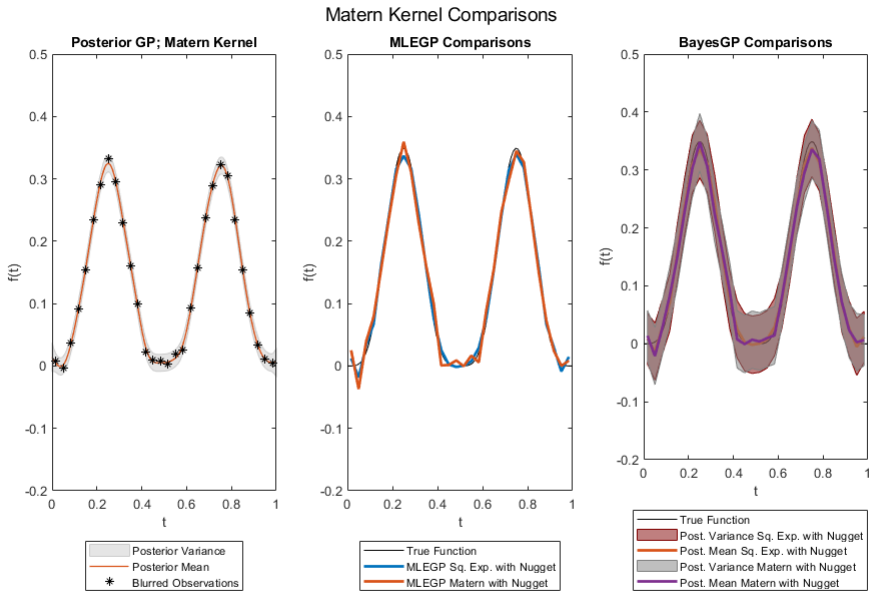

Figure 7: Inversions with the Matérn and squared exponential, both with nuggets, are comparable.

4.2 Linear Tomography

One example for a linear inverse problem is tomography, which is a technique to reconstruct an object by measuring intensities of waves (x-rays, acoustic waves, etc.) sent through the object. The amount by which the object can absorb waves throughout its body is described by an attenuation coefficient function. We collect data of the intensities of the waves before and after entering the object. The change in intensity due to the attenuation can then be computed through Beer's Law, where observations are line integrals through the object. Particular details on the physical model can be found in [4]. Given intensity measurements and the physical model, our goal is to reconstruct the attenuation coefficient throughout the object. In this example, we will be working with tomography in 2D. We begin with some notation, following similar definitions from [17, 4]. Our domain of interest is the unit box, $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$. Even if our object is not simply a box, we can embed the object into Ω. Defined on the unit box is a function $u : \Omega \to \mathbb{R}$ which maps each location in Ω to an *attenuation coefficient* in $\mathbb{R}$. We assume, any location outside the object provides an attenuation coefficient of 0. We next discretize Ω into an $n_1 \times n_2$ grid of pixels. Each pixel is a rectangular subset of the domain Ω. The center of the ij-th pixel, P_{ij}, is given by

$$(x_i, y_j) = (i - 1/2, j - 1/2), \quad i = 1, 2, ..., n_1, \quad j = 1, 2, ..., n_2.$$

In total we have $n_1 \times n_2 = N$ pixels. We assume on each pixel, P_{ij}, the attenuation coefficient is constant, denoted by the value u_{ij},

$$\forall (x, y) \in P_{ij}, \quad u(x, y) = u_{ij}.$$

Observations are described as line integrals through the domain Ω. So first, we describe a parameterization of all lines through Ω. Any line in $\mathbb{R}^2$ can be described by the equation, $ax + by = c$ for

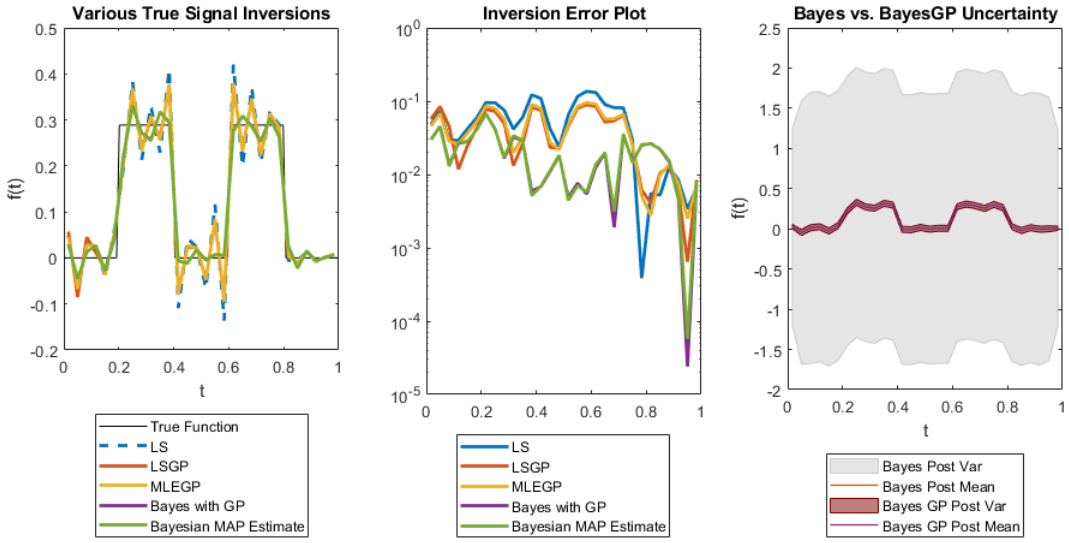

Figure 8: Various inversions of the step function using a GP with a Matérn kernel with a nugget. Here we see overall both Bayesian methods outperform least squares, LSGP, and MLEGP. As before, the variance of Bayes GP is far below the variance of the Bayes method.

some $a, b, c \in \mathbb{R}$. Following [17], define

$$\boldsymbol{\omega}_\rho = [\cos\rho, \sin\rho]^\top, \quad z = \frac{c}{\sqrt{a^2+b^2}}.$$

Then we can similarly parameterize any line as

$$\ell_{\rho,z} = \left\{(x,y) \in \mathbb{R}^2 : \boldsymbol{\omega}_\rho^\top \begin{bmatrix} x \\ y \end{bmatrix} = z\right\}$$

or, by defining $\boldsymbol{\omega}_\rho^\perp = [-\sin\rho, \cos\rho]^\top$,

$$\ell_{\rho,z} = \left\{z\boldsymbol{\omega}_\rho + s\boldsymbol{\omega}_\rho^\perp : s \in \mathbb{R}\right\}.$$

Let $I_{\rho,z}(s)$ be the intensity of a beam along the line $\ell_{\rho,z}$ and $u(\ell_{\rho,z}(s))$ be the value of the attenuation coefficient along the line $\ell_{\rho,z}$. Let s_0 and s_{end} represent the beginning and end of the line as it crosses through our domain Ω. Through Beer's Law, the change in intensity can be described as

$$\mathrm{d}I_{\rho,z}(s) = -u(\ell_{\rho,z}(s))I_{\rho,z}(s)\mathrm{d}s.$$

Solving the differential equation provides us the relationship,

$$-\log\left(\frac{I_{\rho,z}(s_{\text{end}})}{I_{\rho,z}(s_0)}\right) = \int_{s_0}^{s_{\text{end}}} u(\ell_{\rho,z}(s))\ \mathrm{d}s.$$

Model output is specified by the angle of the beam, ρ, and distance from the center of Ω, z, and is the negative log of the change in intensity,

$$y(\rho,z) = -\log\left(\frac{I_{\rho,z}(s_{\text{end}})}{I_{\rho,z}(s_0)}\right) = -\int_{s_0}^{s_{\text{end}}} \frac{\mathrm{d}I}{I}\ \mathrm{d}s = \int_{s_0}^{s_{\text{end}}} u(\ell_{\rho,z}(s))\ \mathrm{d}s. \tag{17}$$

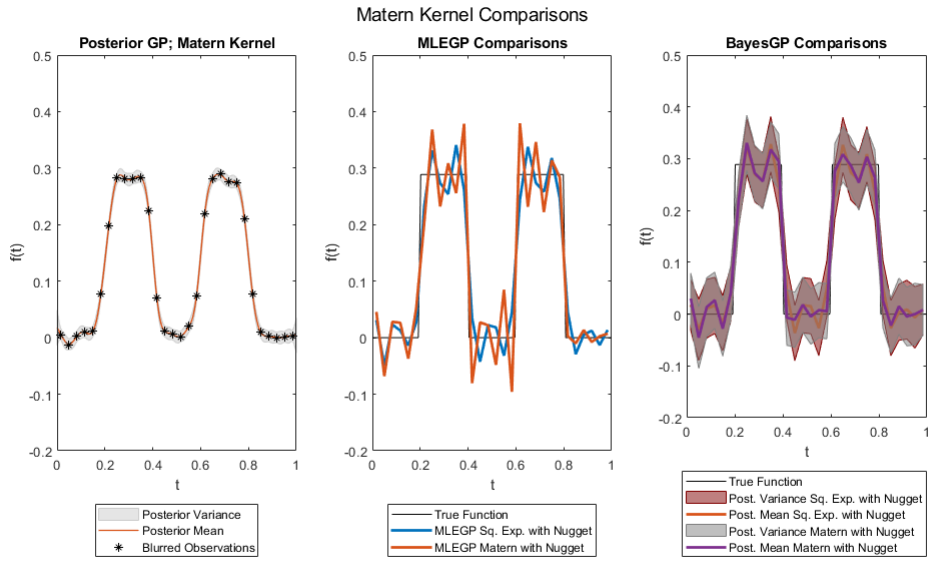

Figure 9: Comparison between squared exponential and Matérn kernels, both with nuggets, on inversions of a blurred step function. There does not seem to be any discernible differences between kernels for this particular problem.

Thus, model outputs are values of the Radon transformation [4] of the attenuation function, $u(x, y)$. We next define the full discretized problem. Suppose we have a collection of model outputs corresponding to a collection of lines,

$$\{\ell_{\rho_k, z_r} : k = 1, 2, ..., K, \quad r = 1, 2, .., R\}, \quad y_{kr} = y(\rho_k, z_r).$$

In total we have $KR = M$ model outputs. Let $\Delta\ell_{kr}^{ij}$ represent the length of line ℓ_{ρ_k, z_r} through pixel $P_{i,j}$. Then the linearization of the line integral is given by

$$y_{kr} = \sum_{i,j=1}^{n} \Delta\ell_{kr}^{ij} u_{ij}.$$

Define the matrices, $\mathbf{U} \in \mathbb{R}^{n_1 \times n_2}$ with ij-th element u_{ij}, and $\mathbf{D} \in \mathbb{R}^{R \times K}$ with kr-th element d_{kr}. Define the vectors, $\mathbf{u} = \text{vec}(\mathbf{U}) \in \mathbb{R}^N$ and $\mathbf{d} = \text{vec}(\mathbf{D}) \in \mathbb{R}^M$. We assume that observations are model outputs corrupted with additive Gaussian noise, $\boldsymbol{\epsilon}$, with zero mean and variance matrix, $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$. Then we can write

$$\mathbf{d} = \mathbf{A}\mathbf{u} + \boldsymbol{\epsilon}, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}). \tag{18}$$

where $\mathbf{A}$ contains $\Delta\ell_{kl}^{ij}$ in the corresponding locations.

Observations can be seen as a surface over a 2D domain, as in Equation (17), corrupted with noise. We use a squared exponential kernel with a nugget for our Gaussian processes, however other options can be considered. We identify optimal hyperparameters, $\hat{\boldsymbol{\theta}}$, via Maximum Likelihood. We calculate the posterior GP conditional on the observations, $\mathbf{b}$, and optimal hyperparameters, $\hat{\boldsymbol{\theta}}$

$$(\mathbf{g}|\mathbf{b}, \hat{\boldsymbol{\theta}}) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{g}}, \boldsymbol{\Sigma}_{\mathbf{g}}).$$

Here, $\boldsymbol{\mu}_{\mathbf{g}}$ and $\boldsymbol{\Sigma}_{\mathbf{g}}$ are the posterior mean and covariance matrix as defined in Equation (2).

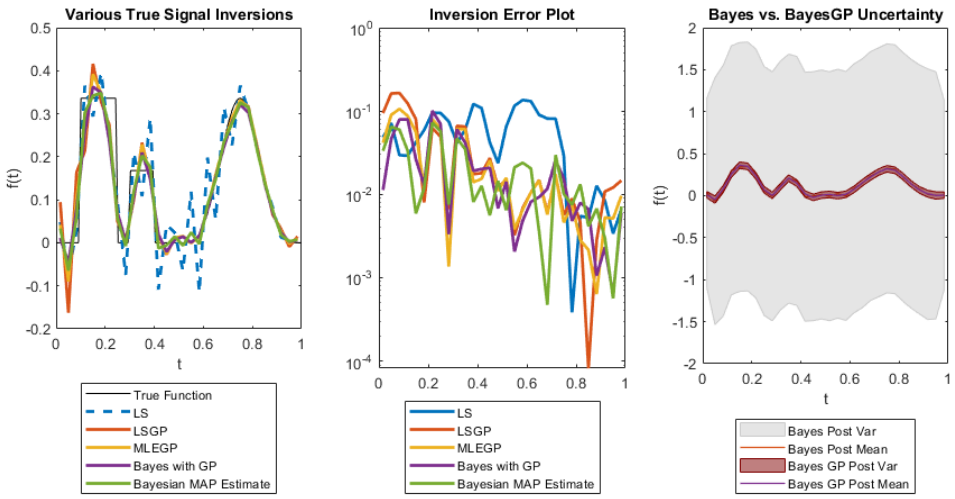

Figure 10: Inversions of a mixed input signal, using a squared exponential kernel with a nugget for the posterior GP. Again, the variance of the BayesGP method outperforms the variance of the purely Bayesian method.

For our numerical experiments, we use a Shepp-Logan phantom of size $N = n \times n = 64 \times 64$. We add independent Gaussian noise with variance $\sigma^2 = 0.001$. For our observations we shoot 91 rays into the phantom at 90 angles evenly spaced on the interval $[0°, 180°]$. Together, we have $M = 91 \times 90$ observations.

Chung et al. noted the effectiveness of a surrogate Gaussian process for missing or censored observations [12]. For tomography examples, we will consider two situations of missing data. First, we will remove a random proportion of observations, then remove contiguous blocks of observations.

For this first example, we remove 40% of the observations. Our new model matrix is simply the original without the rows that correspond to the removed observations. We then compute a posterior GP on our observations. The domain of our GP is the cross product of our angles, and the perpendicular distance of a ray to the origin, $[0°, 180°] \times [-\frac{n}{2}, \frac{n}{2}]$. We use a squared exponential kernel, and find optimal hyperparameters using maximum likelihood. We compute the posterior GP on the original design of the observations. That is, at our given observations as the observations that are missing. A simple method to find the model matrix, $\mathbf{A_g}$, is to use the original model matrix. The full sinogram, sinogram with missing observations, posterior GP, and the relative error heat map are given in Figure 12.

The various inversions are shown in Figure 13 and their relative errors in Figure 14. We note that the MLE GP solution performs better than the LS GP solution, followed by the Bayes GP solution. However, the Bayes MAP estimate has the lowest relative error. We also note that the covariance of the BayesGP solution is orders of magnitude lower than the variance in the Bayes MAP estimate as seen in Figure 15. The exact values are provided in Table 2.

In our next example, rather than removing random observations, we remove contiguous blocks. Here, we remove the rays associated with the angles

$$\theta = \{20°, 21°, 22°, 40°, 41°, 42°, 60°, 61°, 62°, 80°, 81°, 82°\}.$$

This accounts for only 13% of the observations. We construct $\mathbf{A_g}$ in a similar manner to the example above. The various sinograms, and the posterior GP approximation are provided in Figure 16. The

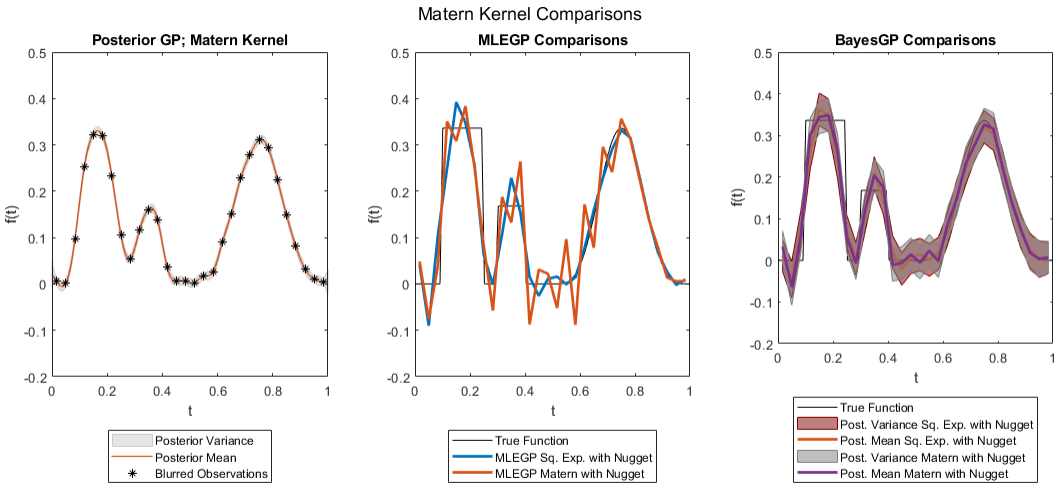

Figure 11: Comparison between squared exponential and Matérn kernels, both with nuggets, on inversions of a blurred composite function. In this case, the squared exponential kernel outperforms the Matérn kernel with MLEGP, but is slightly worse with the BayesGP kernel when considering relative errors.

various inversions are shown in Figure 17, along with the error heat maps in Figure 18 and the variance heat maps in Figure 19. We observe similar patterns to the previous example. The LS GP, MLE GP, and Bayes GP show increasing performance, but the Bayes MAP estimate still have the lowest relative error. Again, we note that the variance in our estimates are orders of magnitude lower in the Bayes GP solution. Interestingly, while we removed less observations as a proportion of all observations, the LS GP solution performed worse in missing contiguous blocks compares to missing random observations. This is intuitive since missing contiguous blocks provide less information about that area of the object, when compared to a single missing observation surrounded by other observations.

Next we investigate how the methods perform over a range of percentages of missing observa-

	Random Removal	Block Removal
Least Squares	2.78e $\times 10^{2}$	1.31 $\times 10^{0}$
LS GP	9.35 $\times 10^{-1}$	1.46 $\times 10^{0}$
MLE GP	7.27 $\times 10^{-1}$	7.17 $\times 10^{-1}$
Bayes GP	5.87 $\times 10^{-1}$	5.77 $\times 10^{-1}$
Bayes MAP Estimate	3.05 $\times 10^{-1}$	2.59 $\times 10^{-1}$
Bayes GP Median Variance	3.25 $\times 10^{-2}$	4.90 $\times 10^{-2}$
Bayes Median Variance	1.89 $\times 10^{0}$	1.46 $\times 10^{0}$

Table 2: The relative errors of the tomography inversions with the various methods. We see that MLEGP and BayesGP both improve on the relative error of LSGP. However, the best method is still the pure Bayesian method. On the other hand, the variance of the BayesGP method is magnitudes lower than the variance of the pure Bayesian method.

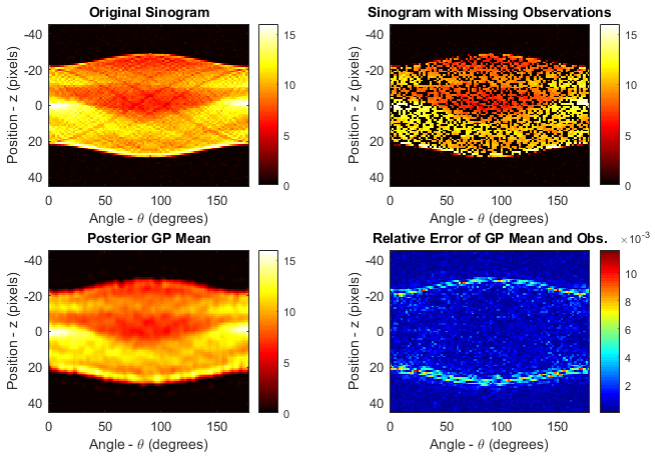

Figure 12: The original sinogram, the sinogram without the removed observations and the posterior GP are shown above. Note that the GP has now made predictions of what the missing data should be. The error heat map shows relative error of the posterior mean of the GP and the observations.

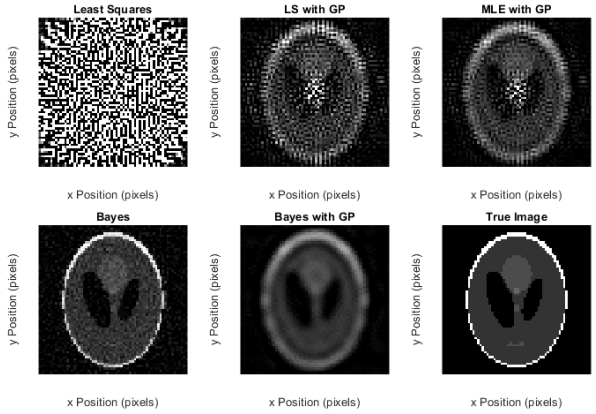

Figure 13: Reconstructions using the various methods. The true image is given in the bottom corner.

tions. We remove a range of proportions of random observations, from 0% to 50%. In our particular example, if we remove more than 50% of our observations we will have less observations than pixels, and our model matrices will no longer be of full column rank for the least squares method. Our Gaussian process is similar to above. We make predictions at the current observations as well as our missing observations. The relative error versus the proportion of missing observations are given in Figure 20. We see similar results from before, in order of accuracy, we have LSGP, MLEGP, BayesGP, and the pure Bayesian method. An interesting result comes from the curve of BayesGP. Removing 50% of the observations provides the same accuracy of removing only 2% of the observations. It would seem to suggest that BayesGP is robust to missing data for this particular problem.

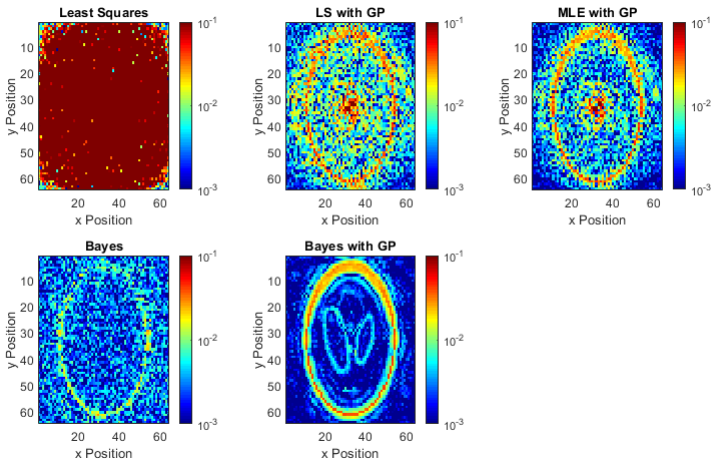

Figure 14: Error maps of the given reconstructions. We note that for the BayesGP method, the majority of the error occurs around the interface of high attenuation. The error of the pure Bayesian method is also spread out throughout the image as compared to the BayesGP method.

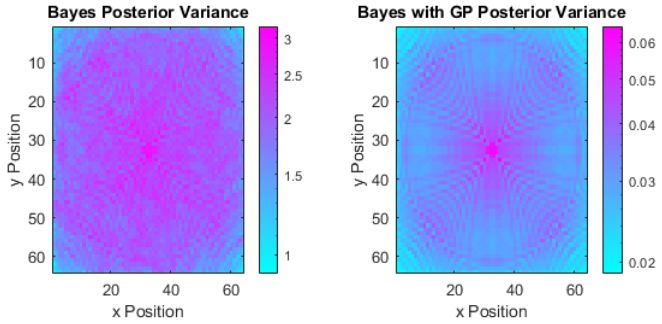

Figure 15: Heat maps of the variance for Bayes and BayesGP. While the pure Bayesian method has a lower relative error, the variance of BayesGP is magnitudes lower compared to the Bayesian method.

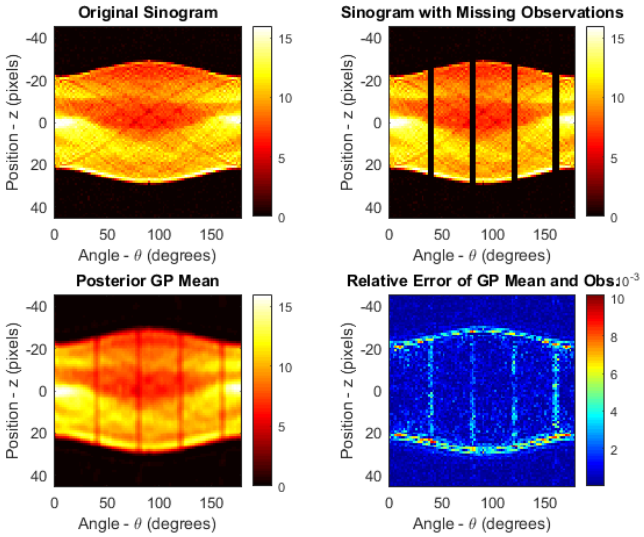

Figure 16: The original sinogram, the sinogram without the removed observations, the posterior GP, and the relative error heat map are shown above. The total relative error of the posterior GP to the original observations is 1.32×10^{-1}.

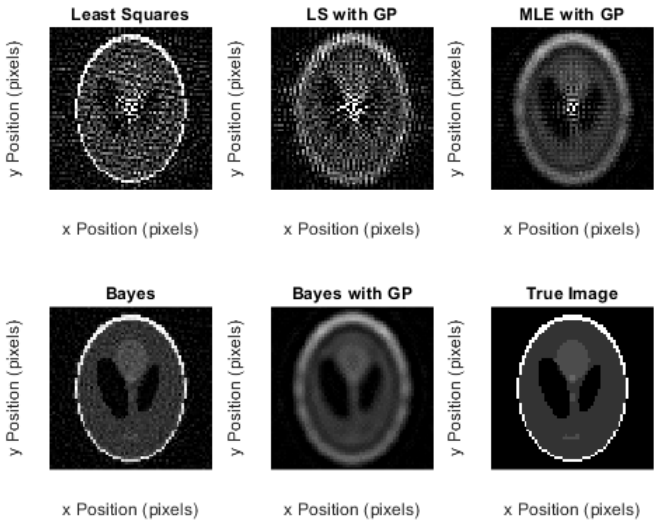

Figure 17: Various reconstructions for the block removal of observations. We see similar patterns for the inversions as compared to the random removals. In order of performance, we have least squares, LSGP, MLEGP, BayesGP, and the pure Bayesian method.

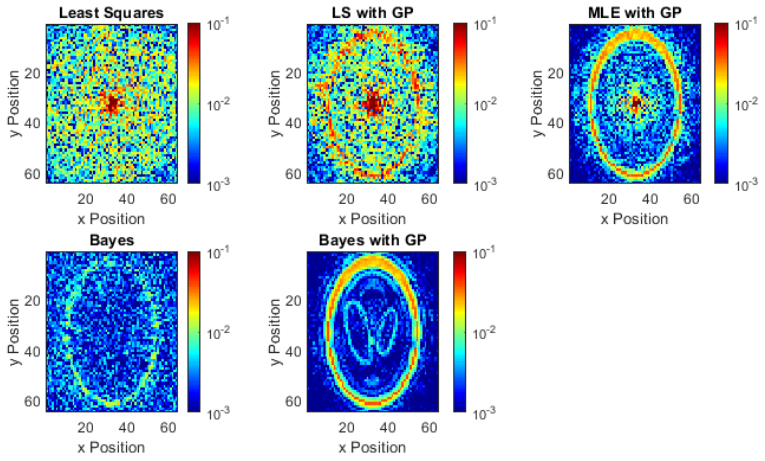

Figure 18: Error heat maps for various reconstructions for the block removal of observations. We observe similar patterns as before. The error of the pure Bayesian method is spread out throughout the image, while the error of BayesGP is concentrated at the interfaces between different attenuation coefficients.

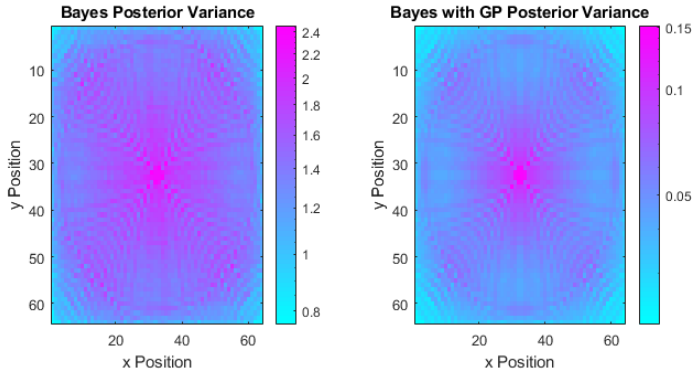

Figure 19: Heat maps of the variance for Bayes and BayesGP. While the pure Bayesian method has a lower relative error, the variance of BayesGP is magnitudes lower compared to the Bayesian method.

5 Conclusion and Future Work

We have developed the techniques for uncertainty quantification using surrogate Gaussian processes, Algorithm 2, as introduced in [12], applied to linear inverse problems. We introduced a closed form distribution of the resulting samples of Algorithm 2. Additionally, we suggest incorporating the covariance of the posterior Gaussian process in a weighted least squares approach. Finally we extend the weighted least squares approach to a Bayesian framework by the inclusion of a prior distribution.

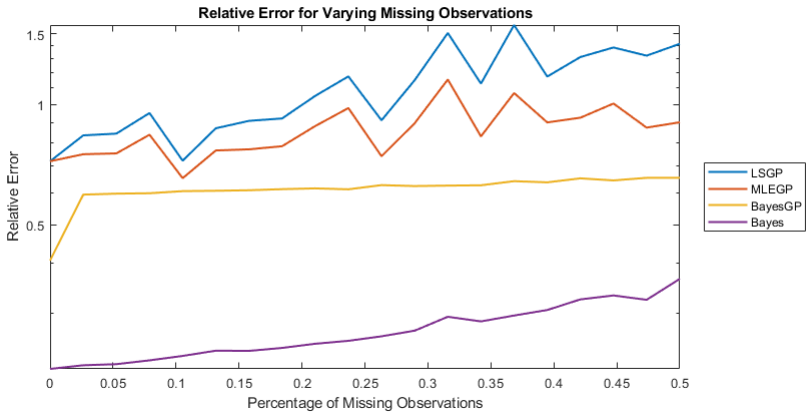

Figure 20: The relative error versus the proportion of missing data. Errors from least squares are omitted due to their magnitude relative to the other methods. While we see similar patterns between the methods as before, we note that BayesGP level out and stop deteriorating after removing 2% of the observations.

We tested methods on two numerical problems: a relatively small 1D deconvolution problem, and a larger tomography problem. In both cases, we acknowledge that one must construct a problem dependent model matrix for the posterior Gaussian process. For the deconvolution problem, we demonstrate the impact of different choices of kernel functions on final uncertainty quantification distributions and confirm the distribution of samples of Algorithm 2. For the tomography problem, we investigate the effect of missing data in our inversions.

We show relative error improvements between the three surrogate approaches for both problems. We observe that between the three surrogate Gaussian process methods, BayesGP performs the best, followed by MLEGP and LSGP with respect to the relative error of the mean of their respective distribution to the true solution. In both problems, we observe vastly decreased variance in our BayesGP inversion compared to the variance from our Bayesian inference method. In all tests for tomography, our Bayesian inference method still outperforms the surrogate Gaussian process methods in relative error to the true solution.

Numerical tests for tomography used a typical squared exponential kernel with hyperparameters optimized via maximum likelihood. This choice of kernel may not be the ideal kernel for the problem of tomography. In particular, the squared exponential kernel does not take advantage of the special sinusoidal structure of the sinogram. Future work includes investigating different choices of kernel functions.

For the sake of comparing the BayesGP and Bayesian inference methods, we used a very basic prior distribution, Gaussian with zero mean and diagonal covariance. Particularly for tomography, there may be a more appropriate prior distributions to consider. Previous work involving Gaussian processes and tomography used the Gaussian random field as the prior distribution, rather than

the likelihood [15]. In that case, it may be more appropriate for the prior to use a Gaussian Markov random field [17], or another stochastic process, as the attenuation function can change drastically at the interface of different densities.

For extremely large inverse problems, methods described here require costly matrix multiplications and storage of covariance matrices. Gaussian processes using globally supported kernel functions will require $O(M^2)$ pair-wise kernel evaluations between the design points. One method to introduce sparsity in our covariance matrices is to use kernels with only local support. Unfortunately, this still does not guarantee that the inversions are sparse as well. Literature on large scale Gaussian processes include the use of an ensemble of local Gaussian processes to approximate an overall Gaussian process [19]. Techniques incorporating techniques in large scale Gaussian processes to the inverse problem are yet to be investigated. If the direct computation of the distributions are infeasible, one can still sample the posterior distributions by applying small alterations to Algorithm 2 to sample the uncertainty quantification distributions of (10) and (2).

Along the lines of large inverse problems, there is also room to explore use of surrogate stochastic processes here as a method in model reduction. We explored sparse data in our tomography examples by varying the amount of available data. However, in the when forming the posterior Gaussian process, and the new model matrix, $\mathbf{A_g}$, on has the freedom to specify a new model design of surrogate observations. One can consider optimal experimental design problems using surrogate stochastic processes. Additionally, one can explore how the effect of varying the number of surrogate observations in the posterior Gaussian process distribution with inversions. Perhaps with an ideal experimental design, one can reduce the number of observations required before performing inversions.

In conclusion, the novel method for uncertainty quantification introduced by Chung and Gramacy et al. approaches the inverse problem in new ways that warrant further investigation. We lay some foundations and extend their method for linear inverse problems through closed form distributions and the inclusion of information from the Gaussian process in the inversion process.

Appendices

Appendix A: Classical Multiple Linear Regression

Suppose you have data $\mathcal{D} = \{(y_i, \mathbf{x}_i)\}_{i=1,...,N}$, where $y_i \in \mathbb{R}$ and $\mathbf{x}_i \in \mathbb{R}^N$. Suppose the data was generated through some function, $f : \mathbb{R}^N \to \mathbb{R}$,,

$$y_i = f(\mathbf{x}_i), \qquad i = 1, 2, ..., N.$$

Now suppose you wish to approximate f through some linear combination of chosen basis functions $\{\phi_i(\mathbf{x})\}_{i=1}^M$, where $\phi_i : \mathbb{R}^N \to \mathbb{R}$. Then our approximation to f is given by

$$\hat{f}(\mathbf{x}, \boldsymbol{\beta}) = \sum_{i=1}^{M} \beta_i \phi_i(\mathbf{x}).$$

Since $\hat{f}$ is parametrized by $\boldsymbol{\beta}$, we wish to find a particular $\boldsymbol{\beta}$ that minimizes the error between the true function f and the approximation $\hat{f}$ with respect to some norm. The problem of least squares is to minimize the squared distance between the true model $f(\mathbf{x})$ and our approximation $\hat{f}(\mathbf{x}, \boldsymbol{\beta})$ at observed data points. Thus, we seek

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^M}{\arg\min} \sum_{i=1}^{N} (y_i - \hat{f}(\mathbf{x}, \boldsymbol{\beta}))^2.$$

A.1 Derivation through Linear Algebra

Define the vectors and matrices,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N, \quad \boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_M(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \dots & \phi_M(\mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{N \times M}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_M \end{bmatrix}$$

Then the function to minimize can be written as a vector norm,

$$\sum_{i=1}^{N} (y_i - \hat{f}(\mathbf{x}, \boldsymbol{\beta}))^2 = \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 .$$

To optimize, we take the derivative with respect to $\boldsymbol{\beta}$,

$$\begin{aligned} \frac{d}{d\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 &= \frac{d}{d\boldsymbol{\beta}} \left[(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}) \right] \\ &= \frac{d}{\boldsymbol{\beta}} \left[\boldsymbol{\beta}^\top \boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{\beta} - 2\boldsymbol{\beta}^\top \boldsymbol{\Phi}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y} \right] \\ &= 2\boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{\beta} - 2\boldsymbol{\Phi}^\top \mathbf{y}. \end{aligned}$$

Setting the derivative to zero, we derive the Standard Equations,

$$\begin{aligned} \frac{d}{d\boldsymbol{\beta}} \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 &\stackrel{!}{=} 0 \\ \implies 2\boldsymbol{\Phi}^\top \boldsymbol{\Phi} \boldsymbol{\beta} - 2\boldsymbol{\Phi}^\top \mathbf{y} &= 0 \\ \implies \hat{\boldsymbol{\beta}} &= (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{y} \end{aligned}$$

A.2 Derivation through Maximum Likelihood Estimation

Suppose observations contain independent, identically distributed additive white noise. Then,

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, \sigma^2)$$

or equivalently,

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_n).$$

The likelihood function is given by

$$\ell(\boldsymbol{\beta}|\mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})\right)$$

The choice of $\hat{\boldsymbol{\beta}}$ that maximized the likelihood function is given by

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^M} \ell(\boldsymbol{\beta}|\mathbf{y}) \\
&= \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})\right) \\
&\Leftrightarrow \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^M} (\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}) \\
\implies \hat{\boldsymbol{\beta}} &= (\mathbf{\Phi}^\top\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{y}
\end{aligned}$$

A.3 Generalized Least Squares (and Weighted Least Squares)

Instead of independent identically distributed additive error, we suppose the distribution of ϵ is more general,

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_\epsilon).$$

Then the distribution of $\mathbf{y}$ is given by

$$\mathbf{y} \sim \mathcal{N}(\mathbf{\Phi}\boldsymbol{\beta}, \mathbf{\Sigma}_\epsilon)$$

with likelihood

$$\ell(\boldsymbol{\beta}|\mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top\mathbf{\Sigma}_\epsilon^{-1}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})\right).$$

The choice of $\boldsymbol{\beta}$ that maximizes the likelihood is equivalent to the following optimization problems,

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^M} \ell(\boldsymbol{\beta}|\mathbf{y}) \\
&\Leftrightarrow \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^M} (\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top\mathbf{\Sigma}_\epsilon^{-1}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}) \\
&= \left\|\mathbf{\Sigma}_\epsilon^{-1/2}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})\right\|_2^2 \\
&= \|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta}\|_{\mathbf{\Sigma}_\epsilon^{-1}}^2.
\end{aligned}$$

We can find the derivative of the objective function, with respect to $\boldsymbol{\beta}$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\beta}}\left[(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})^\top\mathbf{\Sigma}_\epsilon^{-1}(\mathbf{y} - \mathbf{\Phi}\boldsymbol{\beta})\right] &= \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\beta}}\left[\boldsymbol{\beta}^\top\mathbf{\Phi}^\top\mathbf{\Sigma}_\epsilon^{-1}\mathbf{\Phi}\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top\mathbf{\Phi}^\top\mathbf{\Sigma}_\epsilon^{-1}\mathbf{y} + \mathbf{y}^\top\mathbf{\Sigma}_\epsilon^{-1}\mathbf{y}\right] \\
&= 2\mathbf{\Phi}^\top\mathbf{\Sigma}_\epsilon^{-1}\mathbf{\Phi}\boldsymbol{\beta} - 2\mathbf{\Phi}^\top\mathbf{\Sigma}_\epsilon^{-1}\mathbf{y}.
\end{aligned}$$

Setting the derivative equal to zero and solving for $\boldsymbol{\beta}$,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\beta}}\left[(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}_\epsilon^{-1}(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta})\right] \stackrel{!}{=} 0 \\
&\implies 2\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_\epsilon^{-1}\boldsymbol{\Phi}\boldsymbol{\beta} - 2\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_\epsilon^{-1}\mathbf{y} = \mathbf{0} \\
&\implies \hat{\boldsymbol{\beta}} = (\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_\epsilon^{-1}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_\epsilon^{-1}\mathbf{y}
\end{aligned}$$

Appendix B: Tikhonov Regularization

We continue with notation from the previous section. Now we adjust the objective function by penalizing solutions to the optimization problem having large norm. For some $\lambda > 0$, we have the following optimization problem,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^M}{\arg\min} \left\| \mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_2^2 .$$

The derivative of the objective function, with respect to $\boldsymbol{\beta}$,

$$\frac{d}{d\boldsymbol{\beta}} \left[\left\| \mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_2^2 \right] = 2\boldsymbol{\Phi}^\top \boldsymbol{\Phi}\boldsymbol{\beta} - 2\boldsymbol{\Phi}^\top \mathbf{y} + 2\lambda\boldsymbol{\beta}.$$

Setting the derivative equal to zero and solving for $\boldsymbol{\beta}$,

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \lambda \mathbf{I}_M)^{-1} \boldsymbol{\Phi}^\top \mathbf{y}$$

Note, literature on regularization of least squares with general filters, often use $\boldsymbol{\Phi}$ to represent the filter factors.

B.1 Tikhonov Regularization as Bayesian Regression

Bayes rule tells us,

$$p(\boldsymbol{\beta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\beta}) p(\boldsymbol{\beta})$$

Specify a particular prior distribution of $\boldsymbol{\beta}$ by

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \alpha^2 \mathbf{I}_M), \quad \alpha > 0$$

with density

$$p(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2\alpha^2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right).$$

The distribution of $\mathbf{y}$ conditional on $\boldsymbol{\beta}$ is given by

$$\mathbf{Y}|\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$$

with likelihood

$$p(\mathbf{y}|\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}) \right).$$

By Bayes Rule, the posterior density is then proportional to the following,

$$\begin{aligned} p(\boldsymbol{\beta}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}) \right) \exp\left(-\frac{1}{2\alpha^2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta})^\top (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\beta}) - \frac{1}{2\alpha^2} \boldsymbol{\beta}^\top \boldsymbol{\beta}) \right) \end{aligned}$$

The maximum a posteriori (MAP) estimate is the mode of the posterior distribution. In other words, the MAP estimate is the choice of $\boldsymbol{\beta}$ which maximizes the posterior density.

$$
\begin{aligned}
\hat{\boldsymbol{\beta}} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\arg\max}\, p(\boldsymbol{\beta}|\mathbf{y}) \\
&= \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta})^\top(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}) - \frac{1}{2\alpha^2}\boldsymbol{\beta}^\top\boldsymbol{\beta})\right) \\
&\Leftrightarrow \underset{\boldsymbol{\beta}\in\mathbb{R}^M}{\arg\min}\, \frac{1}{2\sigma^2}(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta})^\top(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}) + \frac{1}{2\alpha^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}) \\
&= (\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta})^\top(\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \qquad (\text{With } \lambda = \tfrac{\sigma^2}{\alpha^2}) \\
&= \|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\beta}\|_2^2 + \lambda\,\|\boldsymbol{\beta}\|_2^2\,.
\end{aligned}
$$

The above optimization is equivalent to Tikhonov regularization.

Appendix C: MVN Conditional Distribution

Theorem C.1. *Suppose* $\mathbf{y}$ *follows a multivariate normal distribution. If we partition* $\mathbf{y}$ *by*

$$\mathbf{y} = \begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}\right), \quad \mathbf{y}_1 \in \mathbb{R}^p, \quad \mathbf{y}_2 \in \mathbb{R}^m$$

Then the conditional distribution of $\mathbf{y}_1$ *conditional on* $\mathbf{y}_2$ *is also multivariate normal with the following mean and variance matrix*

$$\mathbf{y}_1|\mathbf{y}_2 \sim \mathcal{N}(\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})$$

We state a few lemmas without proof.

Lemma C.1.1. *A linear combination of Multivariate Normal (MVN) random variables is also Multivariate Normal.*

Lemma C.1.2. *If two MVN random variables have zero covariance, then they are independent.*

Lemma C.1.3. *Suppose* $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are independent MVN random variables. The joint distribution is then given as the product of the individual distributions,*

$$p(\mathbf{y}_1, \mathbf{y}_2) = p(\mathbf{y}_1)p(\mathbf{y}_2).$$

Proof. (*Statistical Methods*) First we define a new random variable, $\mathbf{z} = \mathbf{y}_1 + \mathbf{C}\mathbf{y}_2$ where $\mathbf{C} \in \mathbb{R}^{p\times m}$ is deterministic. By Lemma C.1.1, we know $\mathbf{z}$ is Multivariate Normal. We will enforce the condition that $\mathbf{z}$ and $\mathbf{y}_2$ are independent. By Lemma C.1.2, it is sufficient to enforce $\text{Cov}(\mathbf{z}, \mathbf{y}_2) = 0$.

$$\begin{aligned}
\text{Cov}(\mathbf{z}, \mathbf{y}_1) &= \text{Cov}(\mathbf{y}_2 + \mathbf{C}\mathbf{y}_1, \mathbf{y}_1) \\
&= \text{Cov}(\mathbf{y}_2, \mathbf{y}_1) + \mathbf{C}\text{Cov}(\mathbf{y}_1, \mathbf{y}_1) \\
&= \boldsymbol{\Sigma}_{21} + \mathbf{C}\boldsymbol{\Sigma}_{11} \stackrel{!}{=} \mathbf{0} \\
\implies \mathbf{C} &= -\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}
\end{aligned}$$

By rearrangement, $\mathbf{y}_1 = \mathbf{z} - \mathbf{C}\mathbf{y}_2$. Since the RHS is a linear combination of Multivariate Normal variables, we know $\mathbf{y}_1$ is also Multivariate Normal. Then it is sufficient to calculate the mean and variance of $\mathbf{y}_1$ to characterize the distribution.

$$\begin{aligned}
\mathbb{E}[\mathbf{y}_2|\mathbf{y}_1] &= \mathbb{E}[\mathbf{z} - \mathbf{C}\mathbf{y}_1|\mathbf{y}_1] \\
&= \mathbb{E}[\mathbf{z}|\mathbf{y}_1] - \mathbf{C}\mathbb{E}[\mathbf{y}_1|\mathbf{y}_1] \\
&= \mathbb{E}[\mathbf{z}] - \mathbf{C}\mathbf{y}_1 \\
&= \boldsymbol{\mu}_2 + \mathbf{C}\boldsymbol{\mu}_1 - \mathbf{C}\mathbf{y}_1 \\
&= \boldsymbol{\mu}_2 - \mathbf{C}(\mathbf{y}_1 - \boldsymbol{\mu}_1) \\
&= \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1)
\end{aligned}$$

$$\begin{aligned}
\text{Var}(\mathbf{y}_2|\mathbf{y}_1) &= \text{Var}(\mathbf{z} - \mathbf{C}\mathbf{y}_1|\mathbf{y}_1) \\
&= \text{Var}(\mathbf{z}|\mathbf{y}_1) - \mathbf{C}\text{Var}(\mathbf{y}_1|\mathbf{y}_1)\mathbf{C}^\top \\
&= \text{Var}(\mathbf{z}) = \text{Cov}(\mathbf{z}, \mathbf{z}) \\
&= \text{Cov}(\mathbf{z}, \mathbf{y}_2 + \mathbf{C}\mathbf{y}_1) \\
&= \text{Cov}(\mathbf{z}, \mathbf{y}_2) + \text{Cov}(\mathbf{z}, \mathbf{y}_1)\mathbf{C}^\top = \text{Cov}(\mathbf{z}, \mathbf{y}_2) \\
&= \text{Cov}(\mathbf{y}_2, \mathbf{y}_2) + \mathbf{C}\text{Cov}(\mathbf{y}_1, \mathbf{y}_2) \\
&= \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}
\end{aligned}$$

■

We note that the conditional covariance matrix is the Schur Complement of $\mathbf{\Sigma}_{11}$ of $\mathbf{\Sigma}$. To see this connection, we give another proof of the conditional distribution.

Proof. With properties of probability distributions, we can write the joint density as a product of the conditional and marginal densities.

$$p(\mathbf{y}_1, \mathbf{y}_2) = p(\mathbf{y}_1|\mathbf{y}_2)p(\mathbf{y}_2)$$

The full joint density is given by

$$p(\mathbf{y}_1, \mathbf{y}_2) \propto \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}\right)^\top \begin{bmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}\\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{bmatrix}^{-1} \left(\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{bmatrix}\right)\right)$$

Consider the inverse of the block covariance matrix. Assuming $\mathbf{\Sigma}_{11}^{-1}$ exists, then the Schur complement of $\mathbf{\Sigma}_{11}$ of $\mathbf{\Sigma}$, is given by

$$\mathbf{S} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}.$$

■

Appendix D: Properties of Gaussian Distributions

Definition D.1 (Canonical Form of Gaussian Density)**.** *Let* $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}$*. The density is given by*

$$p(\mathbf{x}) = c \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where c *is a normalizing constant.*

The density in canonical form is given by

$$p(\mathbf{x}) = \exp\left(\alpha + \boldsymbol{\eta}^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x}\right)$$

where α *is a normalizing constant, and* $\boldsymbol{\eta}$ *and* $\mathbf{P}$ *are related to* $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$ *by*

$$\mathbf{P} = \boldsymbol{\Sigma}^{-1}, \quad \mathbf{P}\boldsymbol{\mu} = \boldsymbol{\eta}$$

Theorem D.1 (Linear Combination of Gaussian RVs)**.** *Suppose* $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$*, and* $\mathbf{A}$ *and* $\mathbf{B}$ *are deterministic. Then the linear transformation follows the distribution,*

$$\mathbf{A} + \mathbf{B}\mathbf{y} \sim \mathcal{N}(\mathbf{A} + \mathbf{B}\boldsymbol{\mu}, \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\top)$$

Proof. From a previous lemma, we know that $\mathbf{A} + \mathbf{B}\mathbf{y}$ must be multivariate normal. Thus it is sufficient to calculate the expected value and variance.

$$\begin{aligned}
\mathbb{E}[\mathbf{A} + \mathbf{B}\mathbf{y}] &= \mathbb{E}[\mathbf{A}] + \mathbb{E}[\mathbf{B}\mathbf{y}] \\
&= \mathbf{A} + \mathbf{B}\mathbb{E}[\mathbf{y}] \\
&= \mathbf{A} + \mathbf{B}\boldsymbol{\mu} \\
\mathrm{Var}(\mathbf{A} + \mathbf{B}\mathbf{y}) &= \mathrm{Cov}(\mathbf{A} + \mathbf{B}\mathbf{y}, \mathbf{A} + \mathbf{B}\mathbf{y}) \\
&= \mathrm{Cov}(\mathbf{A}, \mathbf{A}) + \mathrm{Cov}(\mathbf{A}, \mathbf{B}\mathbf{y}) + \mathrm{Cov}(\mathbf{B}\mathbf{y}, \mathbf{A}) + \mathrm{Cov}(\mathbf{B}\mathbf{y}, \mathbf{B}\mathbf{y}) \\
&= \mathrm{Cov}(\mathbf{B}\mathbf{y}, \mathbf{B}\mathbf{y}) \\
&= \mathbf{B}\mathrm{Cov}(\mathbf{y}, \mathbf{y})\mathbf{B}^\top \\
&= \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\top
\end{aligned}$$

■

Bromiley [20] provides a memo for general properties of products of Gaussian PDFs.

Theorem D.2 (Product of Gaussian PDFs of the same dimension)**.** *Suppose* $\mathbf{x}_1 \sim \mathcal{N}_n(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ *and* $\mathbf{x}_2 \sim \mathcal{N}_n(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ *are two Multivariate Gaussian Random Variables of the same dimension. The product of the corresponding densities is also proportional to the density of Gaussian Random Variable* $\mathbf{x}_3$*, that is,*

$$p_1(\mathbf{x})p_2(\mathbf{x}) \propto \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_3)^\top \boldsymbol{\Sigma}_3^{-1}(\mathbf{x}-\boldsymbol{\mu}_3)\right),$$

where the mean and covariance matrix of $\mathbf{x}_3$ *are given by*

$$\boldsymbol{\Sigma}_3 = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}, \quad \boldsymbol{\mu}_3 = \boldsymbol{\Sigma}_3\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_3\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2$$

Proof. We follow [20]. The PDF of $\mathbf{x}_1$ is given by

$$p_1(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_1|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right).$$

In canonical notation, the pdf is expressed in a different way. Define the variables,

$$\mathbf{P}_1 = \boldsymbol{\Sigma}_1^{-1}, \quad \boldsymbol{\eta}_1 = \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1, \quad \alpha_1 = -\frac{1}{2}(n\log 2\pi - \log|\mathbf{P}_1| + \boldsymbol{\eta}_1^\top \mathbf{P}_1^{-1}\boldsymbol{\eta}_1).$$

Then the pdf of $\mathbf{x}_1$ in canonical notation is

$$p_1(\mathbf{x}) = \exp(\alpha_1 + \boldsymbol{\eta}_1^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \mathbf{P}_1 \mathbf{x})$$

Similarly for $\mathbf{x}_2$,

$$p_2(\mathbf{x}) = \exp(\alpha_2 + \boldsymbol{\eta}_2^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \mathbf{P}_2 \mathbf{x})$$

Now we calculate the product,

$$\begin{aligned} p_1(\mathbf{x})p_2(\mathbf{x}) &= \exp((\alpha_1+\alpha_2) + (\boldsymbol{\eta}_1+\boldsymbol{\eta}_2)^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top(\mathbf{P}_1+\mathbf{P}_2)\mathbf{x}) \\ &= \exp(\alpha_1+\alpha_2-\alpha_3)\exp(\alpha_3 + \boldsymbol{\eta}_3^\top \mathbf{x} - \frac{1}{2}\mathbf{x}^\top \mathbf{P}_3 \mathbf{x}) \end{aligned}$$

where

$$\mathbf{P}_3 = \mathbf{P}_1 + \mathbf{P}_2, \quad \boldsymbol{\eta}_3 = \boldsymbol{\eta}_1 + \boldsymbol{\eta}_2$$

and

$$\alpha_3 = -\frac{1}{2}(n\log 2\pi - \log|\mathbf{P}_3| + \boldsymbol{\eta}_3^\top \mathbf{P}_3^{-1}\boldsymbol{\eta}_3)$$

Now we note, $p_1(\mathbf{x})p_2(\mathbf{x})$ is proportional to another Gaussian in canonical form. We find the covariance matrix,

$$\begin{aligned} \boldsymbol{\Sigma}_3^{-1} &= \mathbf{P}_3 \\ &= \mathbf{P}_1 + \mathbf{P}_2 \\ &= \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1} \\ \implies \boldsymbol{\Sigma}_3 &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \end{aligned}$$

and the mean vector,

$$\begin{aligned} \boldsymbol{\eta}_3 &= \boldsymbol{\Sigma}_3^{-1}\boldsymbol{\mu}_3 \\ \implies \boldsymbol{\mu}_3 &= \boldsymbol{\Sigma}_3\boldsymbol{\eta}_3 \\ &= \boldsymbol{\Sigma}_3(\boldsymbol{\eta}_1+\boldsymbol{\eta}_2) \\ &= \boldsymbol{\Sigma}_3(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \\ &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2) \end{aligned}$$

■

Corollary D.2.1. *An alternative form of the mean vector and covariance matrix of* $\mathbf{x}_3$ *is given by*

$$\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_1(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}\boldsymbol{\Sigma}_2, \quad \boldsymbol{\mu}_3 = \boldsymbol{\Sigma}_2(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_1(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}\boldsymbol{\mu}_2$$

References

[1] Gerald Teschl. *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc., 2012.

[2] A. Ribes and F. Schmitt. Linear inverse problems in imaging. *IEEE Signal Processing Magazine*, 25(4):84–99, 2008.

[3] Fatih Yaman, Valery G Yakhno, and Roland Potthast. A survey on inverse problems for applied sciences. *Mathematical problems in engineering*, 2013, 2013.

[4] Charles L Epstein. *Introduction to the mathematics of medical imaging*. SIAM, 2007.

[5] Timothy John Sullivan. *Introduction to uncertainty quantification*, volume 63. Springer, 2015.

[6] José A Caballero and Ignacio E Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE journal*, 54(10):2633–2650, 2008.

[7] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

[8] Zongzhao Zhou, Yew Soon Ong, Prasanth B Nair, Andy J Keane, and Kai Yew Lum. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(1):66–76, 2006.

[9] Jack PC Kleijnen. Kriging metamodeling in simulation: A review. *European journal of operational research*, 192(3):707–716, 2009.

[10] B. Gaspar, A.P. Teixeira, and C. Guedes Soares. Assessment of the efficiency of kriging surrogate models for structural reliability analysis. *Probabilistic Engineering Mechanics*, 37:24 – 34, 2014.

[11] JP Delhomme. Kriging in the hydrosciences. *Advances in water resources*, 1:251–266, 1978.

[12] M. Chung, M. Binois, R. Gramacy, J. Bardsley, D. Moquin, A. Smith, and A. Smith. Parameter and uncertainty estimation for dynamical systems using surrogate stochastic processes. *SIAM Journal on Scientific Computing*, 41(4):A2212–A2238, 2019.

[13] Marc G Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, 2(Dec):299–312, 2001.

[14] Petter Abrahamsen. A review of gaussian random fields and correlation functions, 1997.

[15] T Wang, D Mazon, J Svensson, D Li, A Jardin, and G Verdoolaege. Gaussian process tomography for soft x-ray spectroscopy at west without equilibrium information. *Review of Scientific Instruments*, 89(6):063505, 2018.

[16] Dong Li, J Svensson, H Thomsen, F Medina, A Werner, and R Wolf. Bayesian soft x-ray tomography using non-stationary gaussian processes. *Review of Scientific Instruments*, 84(8):083506, 2013.

[17] Johnathan M Bardsley. *Computational Uncertainty Quantification for Inverse Problems*, volume 19. SIAM, 2018.

[18] Robert B Gramacy and Herbert KH Lee. Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3):713–722, 2012.

[19] Jun Wei Ng and Marc Peter Deisenroth. Hierarchical mixture-of-experts model for large-scale gaussian process regression, 2014.

[20] Paul Bromiley. Products and convolutions of gaussian probability density functions. *Tina-Vision Memo*, 3(4):1, 2003.