DTU Compute
Department of Applied Mathematics and Computer Science

Hilbert space approximations of Gaussian processes in Bayesian modelling

A flexible approach to shape-constrained modelling using Gaussian process approximations

Master Thesis

Hilbert space approximations of Gaussian processes in Bayesian modelling
A flexible approach to shape-constrained modelling using Gaussian process approximations

Master Thesis
January, 2024

By
Morten Rahbæk Boilesen& Johanne Hvidberg Conradsen

Approval

This thesis has been prepared over five months at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Science in Mathematical modelling and computation.

Morten Rahbæk Boilesen - s222303

Signature

26 January 2025

Date

Johanne Hvidberg Conradsen - s223190

Signature

26 January 2025

Date

Abstract

In many fields such as medicine and econometrics, we encounter data where domain knowledge specifies a particular shape in the data, such as monotonicity or concavity. Incorporating these constraints into regression models can potentially improve prediction accuracy, especially in the presence of limited data. This thesis explores the development of shape-constrained models, enforcing monotonic and u-shaped behaviour by leveraging a reduced rank approximation of Gaussian processes, known as the Hilbert space approximation. In the first part of the thesis, we derive this approximation and investigate how it affects both predictive precision and computational complexity compared to a full Gaussian process. Through theoretical analysis, we illuminate how the trade-off between accuracy and complexity depends on the number of basis functions and the domain size of the approximation. In the second part of the thesis, we examine the advantages and limitations of using the Hilbert space approximation for constructing shape-constrained functions. We find that while it offers flexible and analytically tractable models, challenges remain in selecting appropriate priors and hyperparameters when data is scarce. In terms of prediction accuracy, the incorporation of shape constraints generally improves performance in interpolation tasks. We find that extrapolation tasks are more difficult, particularly for u-shaped models, due to an inherent exponentially growing prior variance. Finally, we explore whether shape-constrained models enhance data efficiency, observing some improvements on small datasets, although the results remain inconclusive and further investigation is warranted. Overall, this work provides useful insights into the potential and challenges of using the Hilbert space approximation for shape-constrained modelling.

Acknowledgements

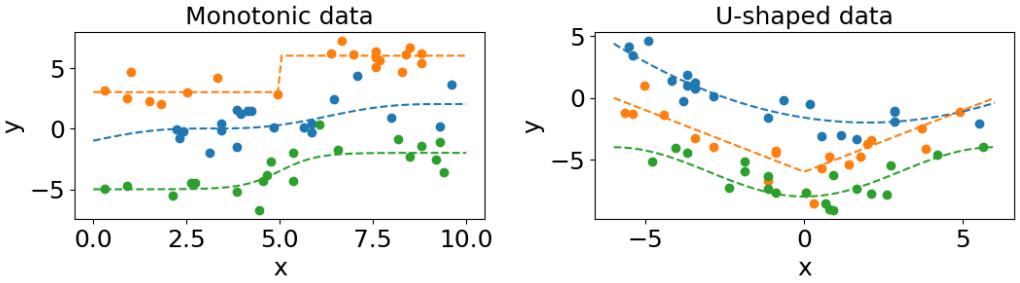
We would like to sincerely thank our supervisor, Michael, for his support and guidance throughout this project. His constructive feedback and thoughtful suggestions have been incredibly helpful, and his patience and encouragement kept us motivated along the way.

Contents

1 Introduction

When working in the field of data science, we may have specific domain knowledge available to guide us. When the domain knowledge shows that the data has a particular shape, we have *shape-constrained data*. It could be that the process generating the data is monotonically increasing or decreasing, as we see it in dose-response curves in medicine (Kelly and Rice 1990), or the data could be u-shaped concave utility functions in econometrics (R. F. Meyer and Pratt 1968) or increasing growth curves, non-increasing survival function or u-shaped hazard function (Reboul 2005) in survival analysis. Sometimes, the shape constraints follow from the laws of nature, as in the case of dose-response curves. Other times, they are based on different assumptions, as we see it in sociological or economical studies. When we have a lot of data available to us, it is generally not necessary to enforce the shape constraints in the model, as these will be learned directly from the data. The challenges arise when only very little data is available and are further complicated when the observations are noisy. In this case, we may benefit from incorporating shape-constraints into the models. Both in the sense that we ensure that the models comply with the shape-constraints, but also based on the assumption that we will have a better fit if we utilize our prior knowledge.

In this work, we aim to develop shape-constrained models to fit monotonic data and u-shaped data. In their simplest form, these correspond to a linear model or a parabola, but come in many variations, see figure 1.1. We will attempt to create flexible, non-parametric models in order to capture this variety based on the data itself. We will do this by utilizing a low-rank approximation of a Gaussian process (GP). This approximation is known as the Hilbert space (HS) approximation and was first described in Solin and Särkkä (2020). It is based on a basis function representation of the Gaussian process and this essentially enables us to sample positive functions with analytically tractable integrals, which we use to model monotonic or convex functions.



(a) An example of datasets generated from different monotonic processes.

(b) Three datasets generated from u-shaped processes.

Figure 1.1

The content of this master thesis can be divided into two main pillars. The first pillar consists of a thorough analysis and investigation of the Hilbert space approximation proposed in Solin and Särkkä (ibid.). We will follow in their footsteps to derive the HS approximation as well as elaborate on some of the proofs in the original article. The second pillar will tackle the problem of estimating shape-constrained functions from noisy observations by constructing strict shape-constrained functions utilising the advantageous analytical properties from the Hilbert space approximation.

Overall, this thesis aims to answer the following research questions:

1. *How does the Hilbert space approximation affect predictive precision and computational complexity compared to a full Gaussian process?*
2. *What are the advantages and limitations of using the Hilbert space approximation for enforcing shape-constrained functions, and how does it compare to other shape-constrained models?*
3. *How does incorporating shape-constraints affect prediction accuracy in the two regression regimes, interpolation and extrapolation?*
4. *To what extent do the use of shape-constrained functions improve data efficiency?*

The first pillar introduces the Hilbert space approximation and covers the first research question. First, we will derive the Hilbert space approximation. Chapter 2 contains preliminary results that we will use throughout the thesis. Section 2.1 is an introduction to the basic concepts of Gaussian process regression and section 2.2 introduces the reader to relevant topics from Functional Analysis. Together, they provide the necessary fundamental theory for this thesis. Section 2.3 is a brief introduction to the Hamiltonian Monte Carlo algorithm, which we will use in the experimental part of the thesis. Chapter 3 contains the derivation of the Hilbert space approximation of Gaussian processes. In sections 3.1, 3.2, and 3.3, we will show how to approximate covariance functions by a series expansion consisting of orthogonal functions. In section 3.4, we apply this approximation to Gaussian processes. This concludes the derivation of the Hilbert space approximation, which forms the basis of the shape-constrained models. In chapter 4, we illuminate research question 1 through a deeper theoretical analysis of convergence and generalization of the Hilbert space approximation. In section 4.1 we will give a thorough convergence analysis of the approximation by expanding the results presented in Solin and Särkkä (2020). In section 4.2 we investigate the behaviour of the average-case learning curve of the HS approximation compared to the full model.

In the second pillar of the thesis, we will apply the HS approximation in order to model shape-constrained functions. In chapter 5, we set out to answer the second research question by using the approximation to construct three analytically tractable models. That is, a strictly positive (section 5.1), monotonic (section 5.2) and u-shaped (section 5.3) model. As they are based on Gaussian process priors, they are non-parametric and highly flexible. We derive the prior mean and variance for each of the three shape-constrained models. In section 5.3.3, we explain how to configure the prior distributions on the model parameters. In section 5.3.4 we propose an alternative modelling approach by relaxing the strictness of the convexity in the u-shaped model. Finally, in section 5.4.3, we discuss the possibilities and limitations of our three proposed models. In chapter 6 we set up experiments in order to test our proposed monotonic and u-shaped models. We will compare them with existing shape-constrained models for interpolation- and extrapolation problems on both synthetic and real data. Section 6.0.4 serves as a small introduction to the models we use for comparison, the model selection process and the inference methods. Sections 6.1 through 6.4 present the four experiments, where we evaluate how our proposed models compare to other shape-constrained models in terms of data efficiency and general prediction performance. In each section, the results of each experiment are discussed briefly. The comparison will help us answer research question 2, and the overall analysis across all of the shape-constrained models help us answer research questions 3 and 4.

Our findings from chapters 5 and 6 are summarized and discussed further in chapter 7, where we also will discuss the inference method and how to improve the model going

forward.

The data, implementations and code used in this master thesis can be found in the public Github repository by Conradsen and Boilesen (2025).

1.1 Literature review

Many different approaches have been used to fit shape constrained functions. One approach is to make spline-based models, which can also be viewed as a basis function approximation of the underlying process. Ramsay (1988) fitted monotonic models by using a monotonic spline basis. This method has been further developed by e.g. M. C. Meyer (2008), who extended it to convex functions. A different approach to shape constrained splines is constraining the spline coefficients. Köllmann (2016) uses a semi-parametric spline regression method to perform unimodal regression. B-splines are used as the basis, and the unimodality is ensured by constraining the B-spline coefficients to be a unimodal sequence. A drawback of spline-based methods is that we have to choose the spline basis and of the number and placements of the knots. Methods based on Gaussian processes have an advantage to these methods as they are non-parametric and inherently Bayesian.

In recent years, Gaussian process priors has been used in a multitude of ways as basis for shape constrained models. In Riihimäki and Vehtari (2010), monotonicity/convexity is ensured by utilizing that the derivatives of a Gaussian process also is a Gaussian process. Riihimäki and Vehtari (ibid.) fit monotonic functions by modelling the joint covariance of the GP evaluated at the data points and the first derivative evaluated at a number of virtual points. By considering the second derivative instead, convex/concave functions can be modelled, as described in Wang and Berger (2016). A drawback of this method is that the constraints are only locally enforced at the virtual points and therefore the model isn't guaranteed to hold the shape constraint. In Ustyuzhaninov et al. (2020), globally monotonic functions are modelled by solving a stochastic differential equation, with drift and diffusion functions governed by a Gaussian Process specified on a set of inducing points. With this method, it is possible to model monotonicity, but not other shape constraints.

In Maatouk and Bay (2016), a variety of global shape constraints are enforced by creating an alternative basis function expansion, constructed by discretizing the domain. Global shape constraints are ensured by imposing equivalent constraints on the coefficients. Different choices of basis lead to different shape constraints. However, the basis functions rely on a discretization of the domain similar to the spline-based methods. A different approach for finding basis functions can be found in Lenk and Choi (2017). Here, the spectral properties of the covariance function is utilized to construct monotonic, convex, unimodal and s-shaped functions. By using a Karhunen-Loéve series expansion of the kernel function, the GP prior is approximated by a linear model, making it possible to transform it into a function with the desired shape constraints. This is very similar to our approach, however it requires us to be able to compute the Karhunen-Loéve expansion of the covariance function, which limits our choices.

In our proposed model, it is only necessary to be able to compute the spectral density of the covariance function, which means that the basis functions are the same regardless of the choice of covariance function. Also, they do not depend on a discretization of the domain, and the shape-constraints are still enforced globally by construction. The method was first introduced in Andersen et al. (2018), where a monotonic model was created by using the Hilbert Space approximation introduced by Solin and Särkkä (2020). The Hilbert space approximation has already been used in a variety of research e.g. for physics-informed Gaussian process learning in Jones, Rogers, and Cross (2023), deep Gaussian processes

regression in Emzir et al. (2019), as reduced-rank approximation of spectral mixtures Gaussian processes in Fradi and Daoudi (2024) and for ice sheet modelling in Brinkerhoff (2022).

2 Preliminaries

The preliminaries chapter has two purposes: One is to provide the theoretical foundation and context needed for developing the concepts in the latter chapters. This is done by introducing the reader to Gaussian processes regression and concepts from Functional Analysis that are fundamental to understanding the Hilbert space approximation. The other is to introduce the reader to the Hamiltonian Monte Carlo algorithm, which will be used in the practical part of the thesis.

2.1 Introduction to Gaussian processes

In this section, we will introduce the key concepts regarding Gaussian processes and Gaussian process regression. A Gaussian process is a stochastic process commonly used in machine learning and statistics for regression and classification problems. The method is non-parametric, which means that it does not assume a fixed functional form or a predefined number of parameters. The non-parametric approach provides flexible functions and tractable uncertainty quantification, making it a powerful tool in statistical modelling for smaller datasets. One drawback is that the exact solution scales poorly when the size or dimension of the data increases. Gaussian processes has been thoroughly described and this chapters relies on the definitions and theorems from Rasmussen and Williams (2006).

One common formulation of a Gaussian process (GP) is that it describes a distribution over functions, which utilises the properties of the multivariate Gaussian distribution.

To understand this, we can consider a function, $f : \mathbb{R}^d \to \mathbb{R}, d \in \mathcal{N}$, as an infinitely long vector, where each entry represents the function value, $f(\boldsymbol{x})$, at a particular input $\boldsymbol{x} \in \mathbb{R}^d$. A Gaussian process deals with these functions by only considering a finite amount of these entries at once, but still maintaining the function's properties across infinitely many points.

A common definition of a Gaussian process is given as,

Definition 2.1.1 *A Gaussian process, $f(\boldsymbol{x})$, is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

This means that for any finite set of points on which we evaluate the process, the outcomes will be consistent. Therefore, we only need to consider the values of the function at the discrete set of input values $\boldsymbol{x}_n$, making it possible to work in a finite space. Furthermore, the Gaussian process, $f(\boldsymbol{x})$, can be described completely by the two components from the Gaussian multivariate distribution, that is, the mean function, $\mu(\boldsymbol{x})$, and covariance function, $k(\boldsymbol{x}, \boldsymbol{x}')$. This non-parametric nature also provides an attractive foundation for building flexible models that are easy to configure and from which we have direct access to uncertainty estimates.

Formally we specify a Gaussian process as

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{2.1}$$

We will consider a zero-mean Gaussian process throughout this section such that $\mu(\boldsymbol{x}) = \boldsymbol{0}$.

The covariance kernel defines the covariance between pairs of inputs and must be symmetric positive semi-definite. There are numerous examples of different kernels, and a

canonical example is the *squared exponential* covariance function, which, due to it being both stationary and easy to parametrise, has a wide range of uses. It is given by

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa^2 \exp\left(\frac{,|\boldsymbol{x}_i - \boldsymbol{x}_j,|^2}{2\ell^2}\right), \tag{2.2}$$

where the parameter ℓ is called the length scale and parameter κ is called the magnitude and
$,|\cdot,|$ denotes the Euclidean norm. In fact, the *squared exponential* covariance function is part of a class of kernels called *Matérn kernels* given by

$$C_\nu(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{,|\boldsymbol{x}_i - \boldsymbol{x}_j,|}{\ell}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{,|\boldsymbol{x}_i - \boldsymbol{x}_j,|}{\ell}\right) \tag{2.3}$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind and ν is the smoothness parameter. $\nu = 1/2$ corresponds to the exponential Ornstein–Uhlenbeck covariance function, and $\nu \to \infty$ to the *squared exponential* covariance function.

In fig 2.1 we have visualised the *squared exponential* (SE) covariance function with different length scales ℓ with samples from the corresponding GP. Here, it is clear to see how the smoothness of the samples is affected by the choice of ℓ.

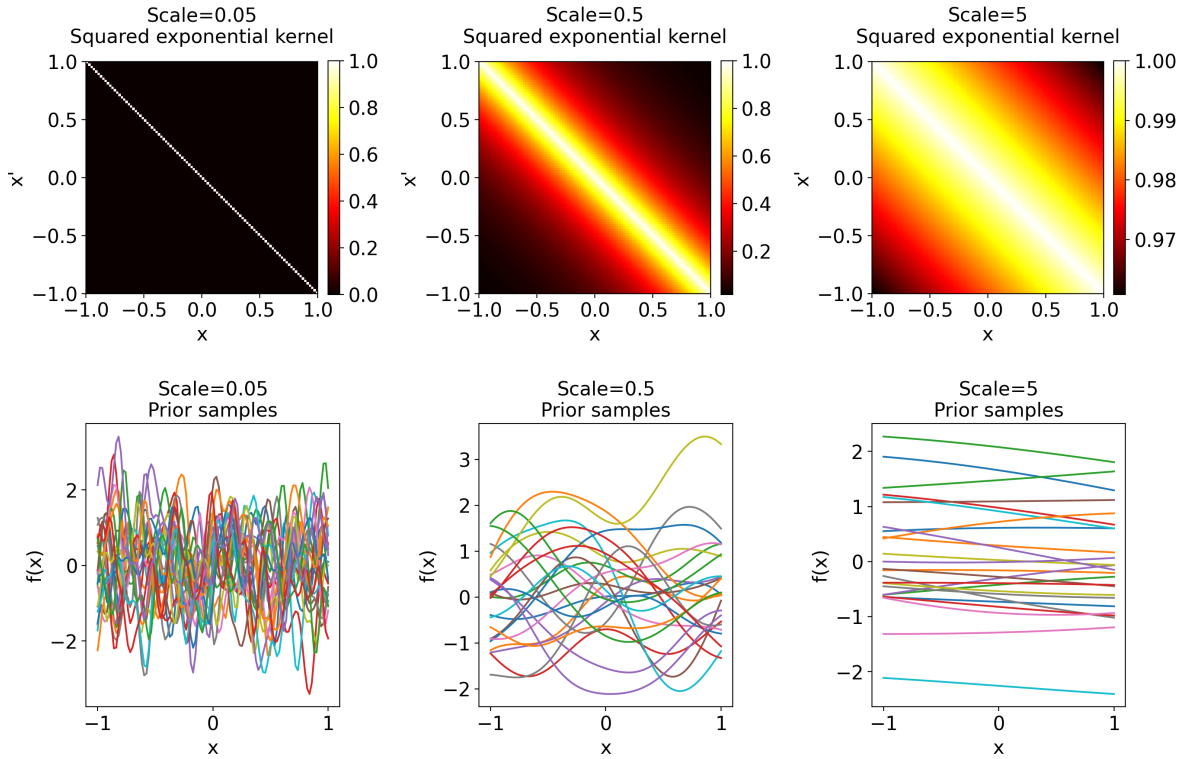


Figure 2.1: Top: heatmaps of the SE kernels showing the covariance between two points $x_1, x_2 \in [-1, 1]$. The length scales are respectively $\ell_1 = 0.05, \ell_2 = 0.5$ and $\ell_3 = 5$. $\kappa = 1$ in all plots.
Bottom: Prior samples from the corresponding GP for each of the three SE kernels.

2.1.1 Gaussian Process regression

A common application of Gaussian processes is in tackling regression problems.

Say we have observed data $\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1, 2, \ldots, n\right\} = (X, \boldsymbol{y})$. We assume that the data has some noise contribution and therefore we describe our model by

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i \tag{2.4}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$ and f follows a Gaussian process prior given by $f \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}'))$. If we assume a Gaussian likelihood, it is given by

$$p(\boldsymbol{y} \mid \boldsymbol{f}) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid f(\boldsymbol{x}_i), \sigma^2) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \sigma^2 I) \tag{2.5}$$

and the prior on $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^\top$ is

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f} \mid \mathbf{0}, K), \quad \text{where } K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ for } i, j \in \{1, \ldots, n\}. \tag{2.6}$$

Given a test point $\boldsymbol{x}_*$, we wish to derive the posterior predictive distribution $p(f(\boldsymbol{x}_*) \mid \boldsymbol{y}, \boldsymbol{x}_*)$ given the observations $\boldsymbol{y}$. Since the Gaussian is conjugate prior to itself, this has an analytically tractable solution. In order to do this, we write the joint distribution of $\boldsymbol{y}$ and $f(\boldsymbol{x}_*)$. Since $\mathrm{Cov}(\boldsymbol{y}, f(\boldsymbol{x}_*)) = \mathrm{Cov}(\boldsymbol{f}, f(\boldsymbol{x}_*)) + \mathrm{Cov}(\boldsymbol{\epsilon}, f(\boldsymbol{x}_*)) = \mathrm{Cov}(\boldsymbol{f}, f(\boldsymbol{x}_*))$, we have

$$\begin{pmatrix} \boldsymbol{y} \\ f(\boldsymbol{x}_*) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K + \sigma^2 & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{pmatrix}\right), \tag{2.7}$$

where $\boldsymbol{k}_* = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), k(\boldsymbol{x}_*, \boldsymbol{x}_2), \ldots, k(\boldsymbol{x}_*, \boldsymbol{x}_n)]^\top$. Note that for other likelihoods, the solution is not necessarily tractable.

Now we may use the general rules for marginalizing Gaussian distributions to derive the analytically tractable posterior predictive distribution, which also is a multivariate Gaussian distribution given by

$$p(f(\boldsymbol{x}_*) \mid \boldsymbol{y}, \boldsymbol{x}_*) = \mathcal{N}\left(\boldsymbol{f}^* \mid \mathbb{E}[\boldsymbol{f}^* \mid \boldsymbol{y}, \boldsymbol{x}_*], \mathbb{V}[\boldsymbol{f}^* \mid \boldsymbol{y}, \boldsymbol{x}_*]\right), \tag{2.8}$$

where the predictive mean and variance are given as

$$\mathbb{E}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}, \boldsymbol{x}_*] = \boldsymbol{k}_*^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{y} \quad \text{and} \quad \mathbb{V}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}, \boldsymbol{x}_*] = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{k}_*. \tag{2.9}$$

See chapter 2.2 in Rasmussen and Williams (2006) for more details.

A simple example is visualised in figure 2.2, where we fit a zero-mean GP with SE kernel on a small simulated dataset.

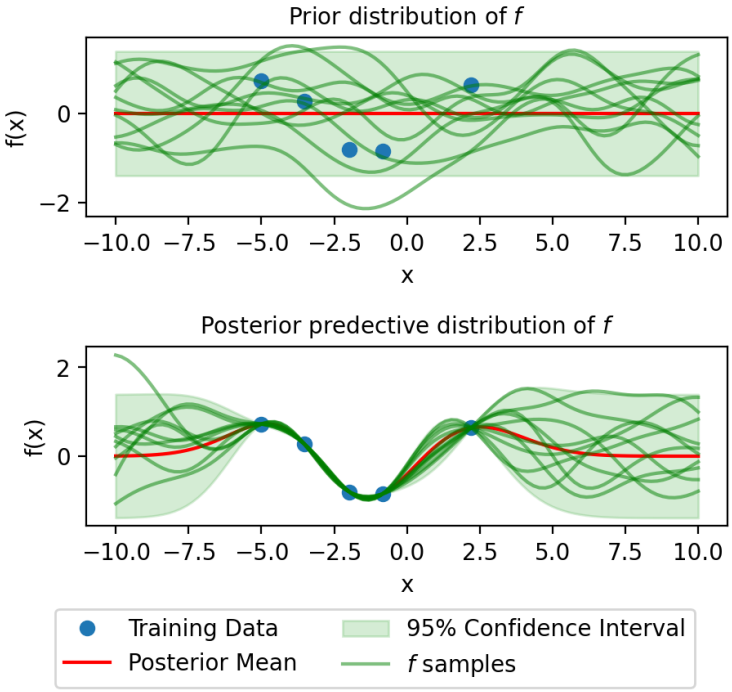


Figure 2.2: Visualisation of the prior and the posterior predictive distribution of a Gaussian process trained on five data points with $x_n \sim \mathcal{U}(-5, 5)$ and $y_n = \sin(x_n) + \epsilon_n$ where $\epsilon_n \sim \mathcal{N}(0, 0.1)$. The prior and posterior GPs are realised in 100 prediction points X_* uniformly spread between -10 and 10. The hyperparameters are optimised by maximising the marginal likelihood using numerical optimisation.

When we incorporate the noise, the posterior predictive distribution of the new observation $y^* = f(\boldsymbol{x}_*) + \epsilon_*$ becomes

$$p(y_* \,|\, \boldsymbol{y}, \boldsymbol{x}_*) = \mathcal{N}\left(y_* \,|\, \mathbb{E}[y_* \,|\, \boldsymbol{y}, \boldsymbol{x}_*], \mathbb{V}[y_* \,|\, \boldsymbol{y}, \boldsymbol{x}_*]\right) \tag{2.10}$$

for

$$\mathbb{E}[y_* \,|\, \boldsymbol{y}, \boldsymbol{x}_*] = \boldsymbol{k}_*^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{y} \quad \text{and} \quad \mathbb{V}[y_* \,|\, \boldsymbol{y}, \boldsymbol{x}_*] = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top \left(K + \sigma^2 I\right)^{-1} \boldsymbol{k}_* + \sigma^2. \tag{2.11}$$

2.1.2 Hyperparameter optimisation

We can add one more layer to the model specification by introducing hyperparameter learning. Given a dataset, $\mathcal{D}$, our objective is to find the optimal hyperparameters so that the model best fits the data. If we consider the set of possible hyperparameters for our model, $\theta = \{\sigma, \theta_K\}$, where θ_K denotes the free parameters of the covariance function and σ^2 is the noise variance, we can derive the marginal likelihood $p(\boldsymbol{y} \,|\, \theta)$ of the data given θ. By marginalising over $\boldsymbol{f}$ we find

$$\begin{aligned} p(\boldsymbol{y} \,|\, \theta) &= \int p(\boldsymbol{y} \,|\, \boldsymbol{f}) p(\boldsymbol{f} \,|\, \theta_K) \,\mathrm{d}\boldsymbol{f} \\ &= \int \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{f}, \sigma^2 I) \mathcal{N}(\boldsymbol{f} \,|\, \boldsymbol{0}, K) \,\mathrm{d}\boldsymbol{f} \\ &= \mathcal{N}(\boldsymbol{y} \,|\, \boldsymbol{0}, \sigma^2 I + K) \end{aligned} \tag{2.12}$$

where we have used the sum rule and inserted the likelihood (2.5) and (2.6) and standard rules for marginalising Gaussians.

Now, we are able to evaluate the likelihood of the data given a set of hyperparameters, θ. The idea is now to find the θ, giving the highest likelihood of the data. This means that we can learn the hyperparameters by maximizing the marginal likelihood of the data given a set of parameters. In practice, we do this by computing or estimating the gradient of $\log(p(\boldsymbol{y} \mid \theta))$ with respect to θ and using a numerical optimisation algorithm.

If we let $\boldsymbol{Q} = \sigma^2 I + K$ then the negative log marginal likelihood, $\log(p(\boldsymbol{y} \mid \theta))$, is given by

$$\mathcal{L} = \frac{1}{2} \log |\, \boldsymbol{Q} \,| + \frac{1}{2} \boldsymbol{y}^T \boldsymbol{Q}^{-1} \boldsymbol{y} + \frac{n}{2} \log(2\pi), \tag{2.13}$$

and the derivates are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta_k} =& \frac{1}{2} \mathrm{Tr} \left(\boldsymbol{Q}^{-1} \frac{\partial \boldsymbol{Q}}{\partial \theta_k} \right) - \frac{1}{2} \boldsymbol{y}^T \boldsymbol{Q}^{-1} \frac{\partial \boldsymbol{Q}}{\partial \theta_k} \boldsymbol{Q}^{-1} \boldsymbol{y}, \\ \frac{\partial \mathcal{L}}{\partial \sigma^2} =& \frac{1}{2} \mathrm{Tr} \left(\boldsymbol{Q}^{-1} \right) - \frac{1}{2} \boldsymbol{y}^T \boldsymbol{Q}^{-1} \boldsymbol{Q}^{-1} \boldsymbol{y}. \end{aligned} \tag{2.14}$$

where Tr denotes the trace of a matrix.

2.1.3 Connection to Bayesian linear regression

So far, we have considered the Gaussian processes from a function space point of view. However, we can also describe a Gaussian process from a weight space view, which has a clear connection to Bayesian linear regression. It turns out that this way of formulating the Gaussian process will be convenient when we construct the Hilbert space approximation in section 3.3.

We start by explaining the Bayesian linear regression setup, where we model the observed target values as

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i,$$

such that

$$f(\boldsymbol{x}_i) = \sum_{j=1}^{m} \alpha_j \phi_j(\boldsymbol{x}_i) = \boldsymbol{\alpha}^\top \phi(\boldsymbol{x}_i), \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

Here $\boldsymbol{x} \in \mathbb{R}^d$ is the input, $\boldsymbol{\alpha} \in \mathbb{R}^m$ are the weights of the linear model and $\phi : \mathbb{R}^d \to \mathbb{R}^m$ is a feature map.

We assume a Gaussian prior over $\boldsymbol{\alpha}$ given by

$$p(\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{0}, \Lambda), \tag{2.15}$$

where Λ is the $m \times m$ covariance matrix of the α's.

We define the following $n \times m$ matrix

$$\Phi(\boldsymbol{x}) = \begin{pmatrix} \phi(\boldsymbol{x}_1)^\top \\ \phi(\boldsymbol{x}_2)^\top \\ \vdots \\ \phi(\boldsymbol{x}_n)^\top \end{pmatrix}.$$

Since $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^\top$ is a linear function of $\boldsymbol{\alpha}$, $\boldsymbol{f}$ also has a Gaussian distribution given by

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{0}, \Phi \Lambda \Phi^\top). \tag{2.16}$$

The posterior predictive distribution of $f(\boldsymbol{x}_*)$ is then given by This enables us to write the joint distribution of y^* and $f(\boldsymbol{x}_*)$ and marginalize as we did for equation (2.9):

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}] &= \phi(\boldsymbol{x}_*)^\top (\Phi^\top \Phi + \sigma^2 \Lambda^{-1})^{-1} \Phi^\top \boldsymbol{y}, \\ \mathbb{V}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}] &= \sigma^2 \phi(\boldsymbol{x}_*)(\Phi^\top \Phi + \sigma^2 \Lambda^{-1})^{-1} \phi(\boldsymbol{x}_*)^\top\end{aligned} \tag{2.17}$$

The derivations of the posterior predictive mean and variance of $\boldsymbol{f}$ are explained in Quinonero-Candela and Rasmussen (2005).

The model we have defined in (2.16) corresponds to a Gaussian process with zero-mean function and covariance function defined by $k(\boldsymbol{x}_i, \boldsymbol{x}_k) = \sum_{j=1}^{m} \Lambda_{jj} \phi_j(\boldsymbol{x}_i) \phi_j(\boldsymbol{x}_k)$, given that Λ is a diagonal matrix.

It is an *m-rank* GP since it consists of m basis functions given in the design matrix ϕ. Please note that this way of formulating the Gaussian process only covers a subset of Gaussian processes in general, since we pick a finite number of basis functions, which reduces the family of possible covariance functions. Also note that $\operatorname{rank}\left(\Phi \Lambda \Phi^\top\right) \leq \operatorname{rank}(\Phi) = m$ which means that when $m < n$, $p(\boldsymbol{f})$ will be a distribution over an m-dimensional subspace of the n-dimensional space $\boldsymbol{f}$ belongs to.

2.2 Linear operators in Hilbert spaces

As the name implies, the Hilbert space approximation relies on the theory of Hilbert spaces, which stems from functional analysis. This section briefly introduces the relevant topics of Functional Analysis used in this thesis. We will cover Hilbert spaces, linear operators, and eigenfunction expansions. The end goal is to state the eigenvalues and eigenfunctions of the negative Laplace operator, as these eigenfunctions form the basis for the Hilbert space approximation.

For a more in-depth treatment of the topics discussed in this section, we refer to e.g. Kreyszig (1991), which is the source of most of the definitions in the following.

2.2.1 Hilbert space

Functional analysis is a branch of abstract mathematics originating from classical analysis. It studies abstract vector spaces and the linear functions defined on these spaces. It generalises results from fields such as linear algebra, linear differential equations, and linear integral equations, as they all turn out to be different flavours of the same underlying structures.

In this thesis, we are concerned with the corner of Functional Analysis regarding the so-called Hilbert spaces. A Hilbert space can be viewed as a generalisation of Euclidean space, $\mathbb{R}^d$. That is, we have taken essential properties of $\mathbb{R}^d$ and used them to define the abstract notion of a "Hilbert space". We warm up to the definition of a Hilbert space by giving two definitions. They should be recognisable from real/complex analysis, albeit in a more abstract setting.

Definition 2.2.1 (Inner product space) *An inner product space X is a vector space with an inner product $\langle x, y \rangle$, i.e. an operation satisfying*

i) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$

ii) $\langle ax, y \rangle = a \langle x, y \rangle$

iii) $\langle x, y \rangle = \overline{\langle y, x \rangle}$ *(the bar denoting complex conjugation)*

iv) $\langle x, x \rangle \geq 0, \langle x \text{ and } x \rangle = 0 \iff x = 0.$

for all vectors $x, y, z \in X$ and scalars a.

The inner product defines a norm on X given by $\|x\| = \sqrt{\langle x, x \rangle}$ and a metric $d(x, y) = \|x - y\|$.

Definition 2.2.2 (Cauchy-sequences and completeness) *Let $(x_1, x_2, \dots)$ be a sequence in a metric space (X, d). The sequence is a Cauchy-sequence if, for every $\varepsilon > 0$, there exists an N such that*

$$d(x_n, x_m) < \varepsilon$$

for $n, m > N$. (X, d) is said to be a complete metric space if every Cauchy sequence in X converges to an element in X.

We may now define the notion of a Hilbert space:

Definition 2.2.3 (Hilbert space) *A Hilbert space is a complete inner product space.*

In Hilbert spaces (and normed spaces in general), we have the notion of a basis:

Definition 2.2.4 (Basis) *A basis of a normed space X is a linearly independent set $B \subset X$ such that $\text{span}(B) = X$. Note that B may not be finite – or even countable. However, if B is finite with d elements, we say that X is d-dimensional. If B is countably infinite, we say that X is infinite-dimensional.*

The d-dimensional Euclidean space is an example of a finite-dimensional Hilbert space with respect to the ℓ^2-norm given by $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^d x_i^2}$ for $\boldsymbol{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_d \end{pmatrix}^\top \in \mathbb{R}^d$.

Another important example of a Hilbert space is the space of square-integrable functions. Let $\Omega \subseteq \mathbb{R}^d$ and let $L^2(\Omega)$ be the space containing functions $f : \Omega \to \mathbb{R}$ such that

$$\left(\int_\Omega f(\boldsymbol{x})^2 \, \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{2}} < \infty, \tag{2.18}$$

with inner product

$$\langle f, g \rangle = \int_\Omega f(\boldsymbol{x}) g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \tag{2.19}$$

Then $L^2(\Omega)$ is a Hilbert space, and it is infinite-dimensional.

2.2.2 Linear Operators on Hilbert Spaces and the negative Laplace operator

Functional analysis is concerned with functions between abstract spaces. When the spaces are normed, these functions are often called operators. Especially important is the class of *linear operators*.

Definition 2.2.5 (Operator) *Let X and Y be vector spaces of the same field. $T : X \to Y$ is said to be a linear operator if for all $x \in X$, $y \in Y$ and scalars a*

i) $T(x + y) = Tx + Ty$

ii) $T(ax) = aTx$

The simplest example of linear operators are matrices. Let $H_1 = \mathbb{R}^d$ and $H_2 = \mathbb{R}^k$. The linear operators from R^d to R^k operators are defined by $k \times d$ matrices, A, such that

$$T(x) = Ax. \tag{2.20}$$

In fact, for any linear operator $T : X \to Y$ where X and Y are finite dimensional normed spaces, T has a matrix representation with respect to a given basis for X and a given basis for Y.

Another example is operators induced by kernels, so-called Hilbert-Schmidt integral operators. Let $k : \Omega \times \Omega \to \mathbb{R}_+$ be a continuous kernel function $k : \Omega \times \Omega \to \mathbb{R}_+$. We may define a linear operator $T_k : L^2(\Omega) \to L^2(\Omega)$ by

$$(T_k f)(x) = \int_\Omega k(x, x') f(x') \, \mathrm{d}x'.$$

The final example of an operator is the negative Laplace operator, defined as the divergence of the gradient of some vector function $f : \mathbb{R}^n \to \mathbb{R}$. We denote it by $-\nabla^2$, and it is given by

$$-\nabla^2 f(x) = -\sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2} f(x).$$

In this thesis, we consider a compact set $\Omega \subset \mathbb{R}^d$ and define the negative Laplacian on a subset of $L^2(\Omega)$ consisting of sufficiently smooth functions. Furthermore, we impose zero Dirichlet boundary conditions on these functions. Loosely speaking, we consider the domain

$$D = \left\{ f \,\middle|\, \begin{matrix} f \text{ sufficiently smooth} \\ f\,(\partial\Omega) = 0 \end{matrix} \right\} \subset L^2(\Omega), \tag{2.21}$$

where $\partial\Omega$ denotes the boundary of Ω.

Thus can view the negative Laplacian a linear operator from $L^2(\Omega)$ to itself, $-\nabla^2 : D \to L^2(\Omega)$.

2.2.3 Eigenfunctions and eigenvalues of linear operators

As we saw in the previous section, linear operators in finite-dimensional Hilbert spaces can be represented by matrices. Using this analogy, it's not surprising that we can generalise the notion of eigenvectors and eigenvalues to finite-dimensional Hilbert spaces in general. As it turns out, one can generalise further and define 'eigenvectors' and values in infinite dimensional space. When considering function space, we have the notion of *eigenfunctions*.

Definition 2.2.6 (Eigenvalues and eigenvectors of operators) *Let H be a Hilbert space and T be a linear operator $T : H \to H$. We say that ϕ is an eigenvector and λ is an eigenvalue of T if they are a solution to the eigenvalue problem $T\phi = \lambda\phi$.*

When H is a function space, we call the eigenvectors *eigenfunctions*.

Consider a linear operator $T : R^d \to \mathbb{R}^d$ defined by a real-valued $d \times d$ matrix A. We know from linear algebra that if A is symmetric, there exists an orthonormal basis for $\mathbb{R}^d$ consisting of normalised eigenvalues of A. We can extend this to a general finite Hilbert space H by requiring that the operator $T : H \to H$ is *self-adjoint*, so that $\langle Tx, y\rangle = \langle x, Ty\rangle$ for all $x, y \in H$ or, slightly weaker, *Hermitian*, meaning $\langle Tx, y\rangle = \langle x, Ty\rangle$ for all x, y in the domain of T denoted $D(T)$.

When H is infinite-dimensional, it is not enough to assume that T is self-adjoint. Stronger assumptions are required, such as *boundedness* or *compactness*. Rather than going into detail, we consider an example, namely the negative Laplacian with Dirichlet boundary conditions, which is a Hermitian operator (see proof in appendix A.1.1).

Lets consider the eigenvalue problem of the negative Laplacian, $-\nabla^2$,

$$-\nabla^2 \phi = \lambda\phi, \tag{2.22}$$

under the additional constraint

$$\phi(\partial\Omega) = 0, \tag{2.23}$$

which is called the Dirichlet eigenvalue problem. First, we note that the negative Laplacian with Dirichlet boundary conditions is a Hermitian operator (see proof in appendix A.1.1).

It can be shown that when Ω is bounded, there is a countable set of solutions $\{(\phi_j, \lambda_j)\}_{j=1}^{\infty}$ and that the set of ϕ_j's is an orthonormal basis of $L^2(\Omega)$ (see Evans (1998) for a rigorous treatment or S. Holland (2007) for a more conceptual argument). Thus

$$f = \sum_i \langle f, \phi_i \rangle \phi_i, f \in L^2(\Omega) \tag{2.24}$$

and

$$\langle \phi_i, \phi_j \rangle = \int_\Omega \phi_i(x)\phi_j(x)\,\mathrm{d}x = \delta_{ij} \tag{2.25}$$

where δ_{ij} is the Kronecker delta function.

If we consider a hyperbox $\Omega = [-L_1, L_1] \times \cdots \times [-L_d, L_d] \subset \mathbb{R}^d$, the solutions to the Dirichlet eigenvalue problem are

$$\lambda_{j_1,\ldots,j_d} = \sum_{k=1}^{d} \lambda_{j_k}, \text{ for } \lambda_{j_k} = \left(\frac{\pi j_k}{2L_k}\right)^2 \tag{2.26}$$

and

$$\phi_{j_i,\ldots,j_d}(\mathbf{x}) = \prod_{k=1}^{d} \phi_{j_k}(x_k), \text{ for } \phi_{j_k}(x_k) = \frac{1}{\sqrt{L_k}} \sin\left(\sqrt{\lambda_{j_k}}(x_k + L_k)\right). \tag{2.27}$$

The notation $_{j_1,\ldots,j_k,\ldots,j_d}$ indicates that for each dimension k, we have a designated index $j_k \in \mathbb{N}$. Thus $_{j_1,j_2,\ldots,j_d}$ denotes a set of indexes for each of the dimensions where $j_k \in \mathbb{N}$. For instance, if $d = 2$ and $j_1 = 1$ and $j_2 = 3$, we have that λ_{j_1,j_2} and ϕ_{j_1,j_2} corresponds to

$$\lambda_{j_1,j_2} = \lambda_{1,3} = \left(\frac{\pi \cdot 1}{2L_1}\right)^2 + \left(\frac{\pi \cdot 3}{2L_2}\right)^2 \tag{2.28}$$

$$\phi_{j_1,j_2} = \phi_{1,3} = \frac{1}{\sqrt{L_1}} \sin\left(\left(\frac{\pi \cdot 1}{2L_1}\right)(x_1 + L_1)\right) \times \frac{1}{\sqrt{L_2}} \sin\left(\left(\frac{\pi \cdot 3}{2L_2}\right)(x_2 + L_2)\right) \tag{2.29}$$

Figure 2.3. illustrates ϕ_{j_k} for $j_k = 1, \ldots, 4$ and the two-dimensional eigenfunctions for different combinations of j_1 and j_2.

One can verify these are, in fact, solutions to the eigenvalue problem by using the product rule, the fact that $\frac{\partial^2}{\partial x^2} \sin(a(x+b)) = -a^2 \sin(a(x+b))$ and by collecting the terms. As for the Dirichlet boundary conditions, we have $0 = \sin(0) = \sin\left(\pi j_k \frac{2L_k}{2L_k}\right)$.

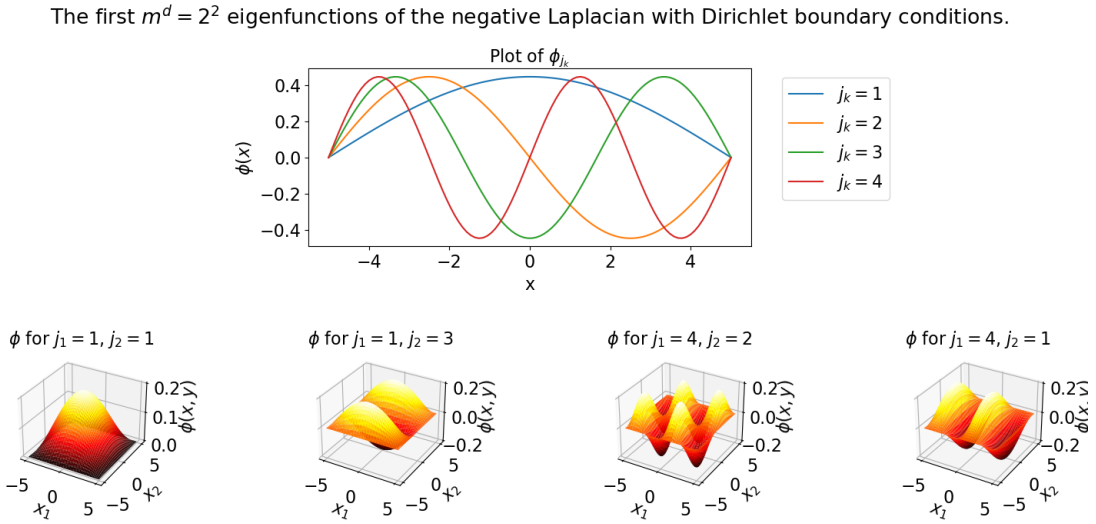


Figure 2.3: An illustration of four eigenfunctions of the negative 2-dimensional Laplacian with Dirichlet boundary condition. The functions correspond to (from the left) $j_1 = j_2 = 1$, $j_1 = 1$ and $j_2 = 3$, $j_1 = 4$ and $j_2 = 2$, $j_1 = 4$ and $j_2 = 1$. The domain is $\Omega = [-5, 5]^2$.

In the next section, we will use the theory of operators and the fact that the eigenfunctions of the negative Laplacian constitute a basis of $L^2(\Omega)$ to form a series expansion of covariance functions. The basis functions will be the Dirichlet eigenfunctions of the negative Laplacian. Thus, the basis does not depend on the specific covariance function, and since it consists of sine functions, it is differentiable, integrable and well-behaved.

2.3 The Hamiltonian Monte Carlo algorithm

In this section, we will take a break from the theoretical perspectives to introduce the Hamiltonian Monte Carlo algorithm, which is a useful tool for obtaining samples from intractable posterior distributions. This will be especially important in the second part of the thesis, where we will test the shape-constrained models in application.

2.3.1 Markov chain Monte Carlo methods and the Metropolis-Hastings algorithm

The Hamiltonian Monte Carlo (HMC) method is a Markov chain Monte Carlo (MCMC) method based on the Metropolis-Hastings algorithm. The goal of MCMC sampling is to obtain samples from a target distribution $p(\boldsymbol{\theta})$. The target distribution may be unknown but must be proportional to a function $\tilde{p}(\boldsymbol{\theta})$ that we can evaluate. The basic idea of MCMC sampling is to construct a Markov chain that has p as stationary distribution. This way, we can obtain samples from the distribution by letting the process evolve sufficiently long. The samples obtained before the Markov Chain has reached the stationary distributions should be discarded. These are often called the *warm-up* samples or *warm-up period*.

The simplest Markov Chain Monte Carlo method is the Metropolis-Hastings algorithm. Conceptually, the Metropolis-Hastings algorithm does this by 'walking' around the parameter space. Given an initial position $\boldsymbol{\theta}_0$, the steps from $\boldsymbol{\theta}_n$ to $\boldsymbol{\theta}_{n+1}$ are taken in two phases:

1) The proposal phase, where a potential next sample $\boldsymbol{\theta}^*$ is drawn from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_n)$. Since the proposal distribution depends only on the previous step, the process is a Markov chain.

2) An acceptance step, where the proposed sample is either accepted or rejected according to an acceptance probability $\alpha = \frac{\tilde{p}(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}_n \mid \boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}_n)}$. If accepted, we set $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}^*$. Otherwise $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$. It is the construction of the acceptance probability that ensures that the stationary distribution of the Markov chain is p.

Note that the resulting samples can be highly correlated.

2.3.2 Hamiltonian Monte Carlo sampling

The Hamiltonian Monte Carlo method uses the same acceptance step as the Metropolis-Hastings algorithm but incorporates Hamiltonian Dynamics into the proposal step. Where the Metropolis-Hastings algorithm could be considered a 'walk' around the distribution, the HMC algorithm is often compared to a satellite orbiting a planet (Betancourt 2018). The Hamiltonian dynamics ensure that the sampler stays in orbit investigating the distribution without crashing into the planet (the mode of the distribution) or flying out into space (outside of the distribution). The HMC method also has the advantage that the samples are less correlated, which results in a better exploration of the target distribution. A detailed explanation of the Metropolis-Hastings algorithm, HMC and other MCMC methods can be found in Murphy (2023), chapter 12.

2.3.3 Convergence diagnostics

Previously, we said that we must discard *warm-up* samples from before the Markov Chain has converged to the stationary distribution. In practice, there is no definite way of determining when that has happened. We must use *convergence* diagnostics to try and determine convergence.

A tool often used for assessing the convergence of Markov chain Monte Carlo samples is the estimated potential scale reduction or $\hat{R}$ (ibid.). Given C chains of MCMC samples, the $\hat{R}$ value is computed by estimating the between chain variance B and within chain variance W.

$$B = \frac{N}{C-1}\sum_{c=1}^{C}\left(\bar{\boldsymbol{\theta}}_{\cdot c} - \bar{\boldsymbol{\theta}}_{\cdot\cdot}\right)^2, \text{ for } \bar{\boldsymbol{\theta}}_{\cdot c} = \frac{1}{N}\sum_{n=1}^{N} x_{nc} \text{ and } \bar{\boldsymbol{\theta}}_{\cdot\cdot} = \frac{1}{M}\sum_{c=1}^{C}\bar{\boldsymbol{\theta}}_{\cdot c},$$

$$W = \frac{1}{M}\sum_{c=1}^{C} s_c^2, \text{ for } s_c^2 = \frac{1}{N-1}\sum_{n=1}^{N}\left(x_{nc} - \bar{\boldsymbol{\theta}}_{\cdot c}\right)^2.$$

The between- and within chain variances are averaged and used to compute the $\hat{R}$ value:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}} \text{ where } \hat{V} = \frac{N-1}{N}W + \frac{1}{N}B. \tag{2.30}$$

If the $\hat{R}$ value is close to one, this is a sign that the chains are essentially the same, indicating that they have converged to the stationary distribution. We say that the chains have mixed. However, this way of computing $\hat{R}$ does not detect non-stationarity within a single chain. This can be remedied by splitting each of the chains in the middle and computing the $\hat{R}$ value on these two chains. We will refer to this as the split-$\hat{R}$ test.

3 Hilbert Space Methods for Reduced-Rank Gaussian Process Regression

As we mentioned in section 2.1, Gaussian process regression scales poorly when the data size, n, grows. There has been a lot of research on how to solve this scaling problem, and one approach is reduced-rank methods. The general idea in reduced-rank methods is to represent the covariance function using a smaller set of basis functions or inducing points in order to reduce the computational complexity while preserving key properties of the covariance kernel. In Solin and Särkkä (2020), such an approach is presented, where a Hilbert space perspective is utilised in order to construct a reduced-rank approximation scheme for the covariance function.

The two main components of the HS approximation are the spectral density of the covariance function and the eigenfunctions and eigenvalues of the negative Laplacian. One of the core advantages of this duality between the spectral density and the negative Laplacian is that the basis functions are independent of the choice of covariance function, which is beneficial in hyperparameter optimisation and in computational complexity. In this chapter, we will derive the Hilbert space approximation proposed in Solin and Särkkä (ibid.) in the following steps:

In section 3.1, we will introduce the reader to the relationship between spectral densities and covariance functions based on Bochner's theorem and the Fourier transform.

In section 3.2, we introduce the covariance operator and demonstrate how, for isotropic covariance functions, it can be expressed as a series involving the negative Laplace operator.

In section 3.3 we will collect the threads from section 3.1 and 3.2 and show how to express a given covariance function into a series expansion in terms of its spectral density and the Laplace eigenvalues and eigenfunctions.

In section 3.4, we show how to directly apply the approximation scheme in a Gaussian process regression problem such as the one described in section 2.1. Finally, we show how the HS approximation can be used for hyperparameter optimisation in section 3.5 and how to enhance numerical stability when implementing the model in section 3.6.

3.1 Spectral Densities of covariance functions

In this section, we will introduce the concept of spectral densities and their relation to covariance functions, as the spectral density is a core component of the construction of the Hilbert space approximation. We start by defining the notion of *stationary* and *isotropic* covariance functions.

Definition 3.1.1 (Stationarity covariance functions) *A covariance function, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is said to be stationary if it is a function of $\boldsymbol{r} = |\boldsymbol{x} - \boldsymbol{x}'|$ for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$. If k is stationary, we write $k(\boldsymbol{r})$.*

Definition 3.1.2 (Isotropic covariance functions) *A covariance function, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, is said to be isotropic if it is a function of $\|\boldsymbol{r}\|$ for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$. If k is isotropic, we write $k(\|\boldsymbol{r}\|)$.*

More intuitively, a stationary covariance function is translation invariant, and an isotropic covariance function is rotation invariant.

In order to derive the *spectral density* of a covariance function, we will need Bochner's theorem (Rasmussen and Williams 2006), which holds for the broader class of complex-valued stationary covariance functions.

To do so, we will need to define *mean square continuity*, which is one way of defining continuity for stochastic processes.

Definition 3.1.3 *A stochastic process, X_t, is said to be mean square continuous at time t^* if*

$$\lim_{t \to t^*} \mathbb{E}\left[(X_t - X_{t^*})^2\right] = 0. \tag{3.1}$$

If this is true for all $t \in \mathbb{R}$, we say the process is mean square continuous.

Theorem 3.1.1 (Bochner's theorem) *Consider a complex-valued function $k : \mathbb{R}^d \to \mathbb{C}$. k is the covariance function of a weakly stationary mean square continuous complex stochastic process, X_t, on $\mathbb{R}^d$ if and only if there exists a positive finite measure μ such that*

$$k(\boldsymbol{r}) = \frac{1}{(2\pi)^d} \int e^{i\omega^\top \boldsymbol{r}} \, \mu(d\omega). \tag{3.2}$$

If the measure, μ, has a density $S(\omega)$, then S is called the spectral density of k. Writing out the integral in Bochner's theorem, we obtain

$$k(\boldsymbol{r}) = \frac{1}{(2\pi)^d} \int S(\omega) e^{i\omega^\top \boldsymbol{r}} \, \mathrm{d}\omega = \mathcal{F}^{-1}\left[S(\omega)\right](\boldsymbol{r}), \tag{3.3}$$

where $\mathcal{F}$ denotes the Fourier transform and $\mathcal{F}^{-1}$ its inverse (for more details on the Fourier transform, we refer to appendix A.8 in Rasmussen and Williams (ibid.)). Thus, the spectral density is the Fourier transform of $k(\boldsymbol{r})$

$$S(\omega) = \mathcal{F}\left[k(\boldsymbol{r})\right](\omega) = \int k(\boldsymbol{r}) e^{-i\omega^\top \boldsymbol{r}} \, \mathrm{d}\boldsymbol{r}. \tag{3.4}$$

The duality between (3.3) and (3.4) is known as the Wiener-Khinchin Theorem (ibid.).

To give some intuition on the spectral density, we can view it as the average weight given to frequencies of the covariance function. For example, let's take a relatively smooth function. The corresponding spectral density will have a rapid decay towards zero as $|\omega| \to \infty$ since a smooth function primarily consists of lower frequencies. However, if we take a covariance function that goes quickly towards zero, then the corresponding spectral density will be smoother with a slower decay towards zero. An example with the squared exponential kernel with $\ell = 0.1$ and $\ell = 3$ is visualised in figure (3.1), where the one dimensional squared exponential kernel is given by

$$k(x, x') = \kappa^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right), \tag{3.5}$$

and the corresponding spectral density is given by

$$S(\omega) = \kappa^2 \sqrt{2\pi}\ell \exp\left(-\frac{\ell^2 \omega^2}{2}\right). \tag{3.6}$$

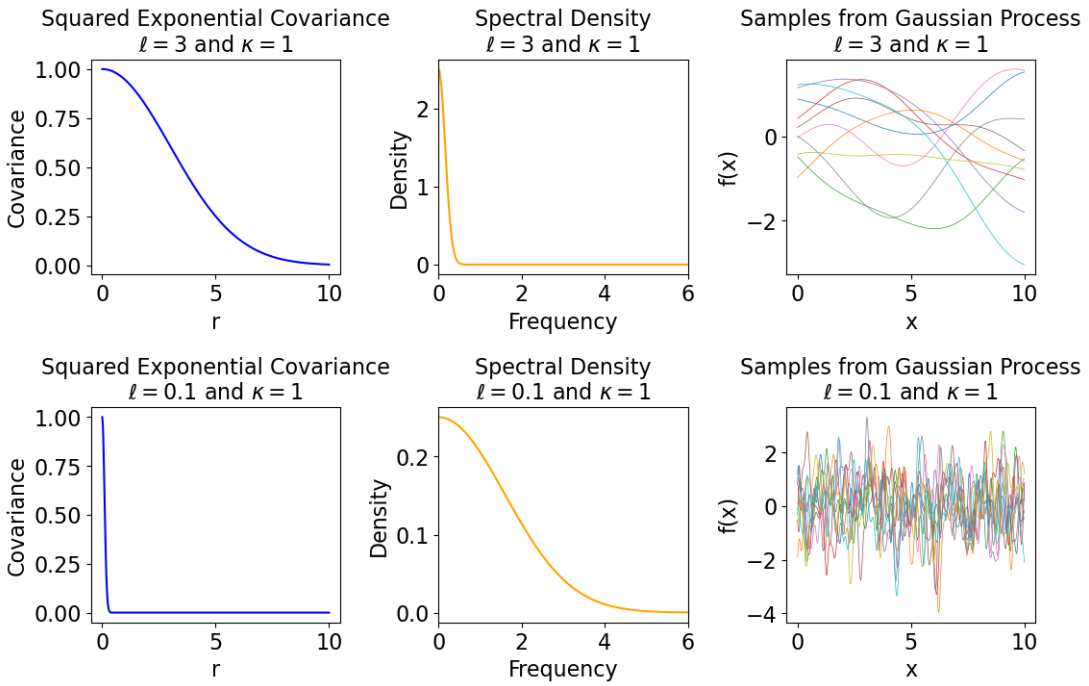


Figure 3.1: A figure illustrating the spectral density of the squared exponential kernel for different length scales. Each row presents a squared exponential covariance function and the corresponding spectral density with length scale $\ell = 3$ in the first row and $\ell = 0.1$ in the second row.

If the covariance function is isotropic, the spectral density will also be isotropic. This is shown in Theorem 2.5.2 in Adler (1981). In such cases, we write $S(\|\omega\|)$.

3.2 The covariance operator

In this section, we introduce the covariance operator and show that isotropic covariance functions can be expressed in terms of the negative Laplace operator described in 2.2.
As we saw in section 2.2, a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ may be used to define a Hilbert Schmidt integral operator $\mathcal{K} : L^2(\Omega) \to L^2(\Omega)$, $\Omega \subseteq \mathbb{R}^d$ by

$$(\mathcal{K}\phi)(\boldsymbol{x}) = \int k(\boldsymbol{x}, \boldsymbol{x}')\phi(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'. \tag{3.7}$$

Assuming k is stationary, we may write

$$(\mathcal{K}\phi)(\boldsymbol{x}) = \int k(\boldsymbol{x} - \boldsymbol{x}')\phi(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'. \tag{3.8}$$

This corresponds to a convolution of k and ϕ. By applying the Fourier transform $\mathcal{F}$ and considering the operator in the Fourier domain instead, we can use the convolution theorem (e.g. see Theorem 7.3.4 in Christensen (2010)), which states that convolution can be written as multiplication in the Fourier domain, i.e.

$$\mathcal{F}[\mathcal{K}\phi](\boldsymbol{\omega}) = \mathcal{F}[k](\boldsymbol{\omega})\mathcal{F}[\phi](\boldsymbol{\omega}) = S(\boldsymbol{\omega})\mathcal{F}[\phi](\boldsymbol{\omega}). \tag{3.9}$$

The last equality follows from the Wiener-Khinchin Theorem (equation 3.4).

If we also assume that k is isotropic, we may consider the spectral density as a function of $\|\boldsymbol{\omega}\|$ instead. We denote this by $S(\|\boldsymbol{\omega}\|)$, which we can again rewrite as a function ψ of

$\|\boldsymbol{\omega}\|^2$ such that $S(\|\boldsymbol{\omega}\|) = \psi(\|\boldsymbol{\omega}\|^2)$. If we require that ψ is an analytic function, we can write it as a power series

$$S(\|\boldsymbol{\omega}\|) = \psi(\|\boldsymbol{\omega}\|^2) = \sum_{k=0}^{\infty} a_k (\|\boldsymbol{\omega}\|^2)^k, \tag{3.10}$$

and insert it in (3.9):

$$\mathcal{F}[\mathcal{K}\phi](\boldsymbol{\omega}) = \sum_{k=0}^{\infty} a_k (\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi](\boldsymbol{\omega}). \tag{3.11}$$

Taking the inverse Fourier transform, we obtain

$$\begin{aligned} \mathcal{K}\phi(x) &= \mathcal{F}^{-1}\left[\sum_{k=0}^{\infty} a_k (\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi](\boldsymbol{\omega})\right](x) \\ &\overset{\text{linearity of } \mathcal{F}^{-1}}{=} \sum_{k=0}^{\infty} a_k \mathcal{F}^{-1}\left[(\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi](\boldsymbol{\omega})\right](x). \end{aligned} \tag{3.12}$$

In order to deal with the $\mathcal{F}^{-1}\left[(\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi](\boldsymbol{\omega})\right]$ term, it is useful to realize $\|\boldsymbol{\omega}\|^2$ is the transfer function of the negative Laplace operator. That is

$$\mathcal{F}\left(-\nabla^2\phi\right) = \mathcal{F}\left(-\sum_k \frac{\partial^2}{\partial x_k^2}\phi\right) = -\sum_k (i\omega_k)^2 \mathcal{F}(\phi) = -\sum_k -\omega_k^2 \mathcal{F}(\phi) = \|\boldsymbol{\omega}\|^2 \mathcal{F}(\phi). \tag{3.13}$$

If we take the Fourier inverse transformation on both sides of (3.13), we get

$$-\nabla^2\phi = \mathcal{F}^{-1}\left(\|\boldsymbol{\omega}\|^2 \mathcal{F}(\phi)\right). \tag{3.14}$$

Furthermore, we also have that

$$(-\nabla^2)^k \phi = \mathcal{F}^{-1}\left((\|\boldsymbol{\omega}\|^2)^k \mathcal{F}(\phi)\right), \tag{3.15}$$

since

$$\begin{aligned} (\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi] &= (\|\boldsymbol{\omega}\|^2)^{k-1} (\|\boldsymbol{\omega}\|^2 \mathcal{F}[\phi]) = (\|\boldsymbol{\omega}\|^2)^{k-1} \mathcal{F}[-\nabla^2\phi] \\ &= (\|\boldsymbol{\omega}\|^2)^{k-2} \mathcal{F}[(-\nabla^2)(-\nabla^2\phi)] = \dots \\ &= \mathcal{F}[(-\nabla^2)^k \phi] \end{aligned} \tag{3.16}$$

By inserting this in (3.12) we can express the covariance operator as

$$\begin{aligned} \mathcal{K}\phi(x) &= \sum_{k=0}^{\infty} a_k \mathcal{F}^{-1}\left[(\|\boldsymbol{\omega}\|^2)^k \mathcal{F}[\phi](\boldsymbol{\omega})\right](x) \\ &= \sum_{k=0}^{\infty} a_k (-\nabla^2)^k \phi(x). \end{aligned} \tag{3.17}$$

The advantage of this series expansion is that we have expressed the covariance operator (and, through that, the covariance function) in terms of the well-studied Laplace operator. This will allow us to approximate the covariance function by utilizing the eigenvalues and eigenfunctions of the Laplace operator, as we shall see in the next section.

3.3 Hilbert space approximation

The goal of this section is to derive an approximation of a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ based on its spectral and the eigenfunctions and eigenvalues from the Laplace Operator. The method described will provide a framework for approximating a multitude of covariance functions and will be the foundation of the Hilbert space approximation of Gaussian processes.

Let $\Omega \subset \mathbb{R}^d$ be compact. Consider solutions to the eigenvalue problem for the Laplace Operator with Dirichlet boundary conditions as described in section 2.2.

$$\begin{aligned} -\nabla^2 \phi_j(\boldsymbol{x}) &= \lambda_j \phi_j(\boldsymbol{x}) & \boldsymbol{x} &\in \Omega \\ \phi_j(\boldsymbol{x}) &= 0 & \boldsymbol{x} &\in \partial\Omega. \end{aligned} \tag{3.18}$$

If we consider

$$l(\boldsymbol{x}, \boldsymbol{x}') = \sum_j \lambda_j \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'), \tag{3.19}$$

we want show that $l(\boldsymbol{x}, \boldsymbol{x}')$ works as a kernel for f in the sense that

$$-\nabla^2 f(\boldsymbol{x}) = \int_\Omega l(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \tag{3.20}$$

for sufficiently (weakly) differentiable functions f in the domain Ω assuming Dirichlet boundary conditions. This is reminiscent of the kernel-induced Hilbert Schmidt integral operators from section 2.2.2, and therefore, we say that l is a *formal kernel* of $-\nabla^2$.

To show 3.20, we insert (3.19) in the right-hand side of (3.20). We may switch the integration and sum and use that $\lambda_j \phi_j(\boldsymbol{x}') = -\nabla^2 \phi_j(\boldsymbol{x})$ to obtain

$$\int_\Omega \sum_j \lambda_j \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' = \sum_j \left(\int \lambda_j \phi_j(\boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \right) \phi_j(\boldsymbol{x}) \tag{3.21}$$

$$= \sum_j \left(\int -\nabla^2 \phi_j(\boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \right) \phi_j(\boldsymbol{x}). \tag{3.22}$$

We recognize the term in the parenthesis as the inner product $\langle f, -\nabla^2 \phi_j \rangle$. Now we may use that $-\nabla^2$ is a hermitian operator (see (A.1.3), which means that $\langle f, -\nabla^2 \phi_j \rangle = \langle -\nabla^2 f, \phi_j \rangle$. This enables us to use that the set of ϕ_j's is an orthonormal basis of $L^2(\Omega)$. Thus

$$= \sum_j \langle f, -\nabla^2 \phi_j \rangle \phi_j(\boldsymbol{x}). \tag{3.23}$$

$$= \sum_j \langle -\nabla^2 f, \phi_j \rangle \phi_j(\boldsymbol{x}) \tag{3.24}$$

$$= -\nabla^2 f(\boldsymbol{x}). \tag{3.25}$$

And thus, l is a formal kernel of $-\nabla^2$.

Next, we also define

$$l^s(\boldsymbol{x}, \boldsymbol{x}') = \sum \lambda_j^s \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'). \tag{3.26}$$

Similarly, we want to show that this can be considered a formal kernel of $(-\nabla^2)^s$. We start by considering

$$\begin{aligned} \int_\Omega l^s(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' &= \int_\Omega \sum_j \lambda_j^s \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \\ &= \sum_j \left(\int_\Omega \lambda_j^s \phi_j(\boldsymbol{x}') f(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' \right) \phi_j(\boldsymbol{x}). \end{aligned} \tag{3.27}$$

By using the properties of the eigenvalues and eigenfunctions repeatedly, we can show

$$\begin{aligned}\lambda_j^s \phi_j(\boldsymbol{x}') =& \lambda^{s-1}(-\nabla^2)\phi_j(\boldsymbol{x}') = \lambda^{s-2}(-\nabla^2)\lambda_j\phi_j(\boldsymbol{x}') \\ =& \lambda^{s-2}(-\nabla^2)^2\phi_j(\boldsymbol{x}') = \dots \\ =& (-\nabla^2)^s\phi_j(\boldsymbol{x}').\end{aligned} \tag{3.28}$$

Inserting this back into (3.27), we have

$$\begin{aligned}\int l^s(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' &= \sum_j \left(\int (-\nabla^2)^s\phi_j(\boldsymbol{x}')f(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\right)\phi_j(\boldsymbol{x}) \\ &= \sum_j \langle f, (-\nabla^2)^s\phi_j\rangle\phi_j(\boldsymbol{x}) \\ &= \sum_j \langle (-\nabla^2)^s f, \phi_j\rangle\phi_j(\boldsymbol{x}) \\ &= (-\nabla^2)^s f(\boldsymbol{x}).\end{aligned} \tag{3.29}$$

We have used that $(-\nabla^2)^s$ also is hermitian, which is shown in detail (A.1.3).

Now we are ready to return to (3.17), where we expressed the covariance operator $\mathcal{K}$ in terms of the negative Laplacian:

$$\mathcal{K}f(\boldsymbol{x}) = \sum_{k=0}^{\infty} a_k(-\nabla^2)^k f(\boldsymbol{x}).$$

Inserting our newfound expression for $-\nabla^2 f(x)$ gives us

$$\begin{aligned}\mathcal{K}f(\boldsymbol{x}) &= \sum_{k=0}^{\infty} a_k \int_\Omega l^k(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}' \\ &= \int_\Omega \sum_{k=0}^{\infty} a_k l^k(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'.\end{aligned} \tag{3.30}$$

Recalling that we defined $\mathcal{K}$ in section 3.2 by

$$\mathcal{K}f(\boldsymbol{x}) = \int_\Omega k(\boldsymbol{x},\boldsymbol{x}')\phi(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}',$$

we conclude that $k(\boldsymbol{x},\boldsymbol{x}') = \sum_{k=0}^{\infty} a_k l^k(\boldsymbol{x},\boldsymbol{x}')$ on Ω when the boundary conditions are met. Therefore, we can use the right-hand side as an approximation of $k(\boldsymbol{x},\boldsymbol{x}')$ such that

$$\begin{aligned}k(\boldsymbol{x},\boldsymbol{x}') &\overset{\text{for } \boldsymbol{x},\boldsymbol{x}'\in\Omega}{=} \sum_{k=0}^{\infty} a_k l^k(\boldsymbol{x},\boldsymbol{x}') \\ &= \sum_{k=0}^{\infty} a_k \underbrace{\sum_j \lambda_j^k \phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}')}_{l^k(\boldsymbol{x},\boldsymbol{x}')} \\ &= \sum_j \left(\sum_{k=0}^{\infty} a_k \lambda_j^k\right)\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}') \\ &= \sum_j S(\sqrt{\lambda_j})\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}')\end{aligned} \tag{3.31}$$

where the last equation follow by using (3.10) with $\|\omega\| = \sqrt{\lambda_j}$.

By truncating this to a finite sum, we may define the Hilbert space approximation of $k(\boldsymbol{x}, \boldsymbol{x}')$ of rank m by

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j=1}^{m} S(\sqrt{\lambda_j})\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'). \tag{3.32}$$

In figure (3.2), we used the HS approximation on Matérn covariance functions with different hyperparameters ν and different numbers of basis functions. For higher values of ν, the approximation needs lower amounts of basis functions m compared to lower values of ν.

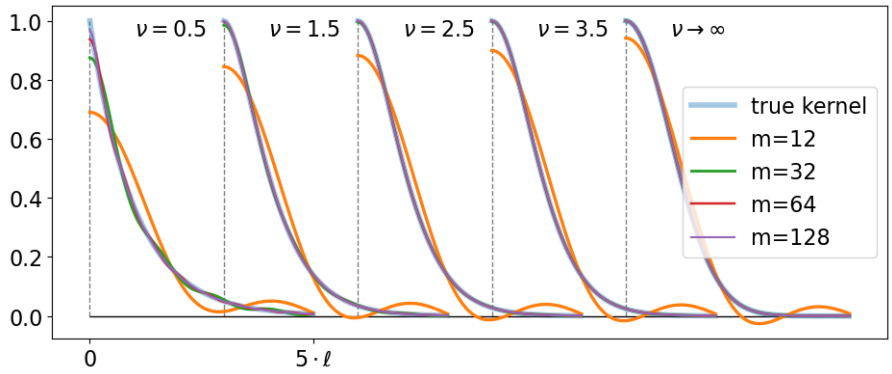


Figure 3.2: Example of Hilbert space approximations of Matérn covariance functions for different numbers of basis functions m. We compare with the full Matérn covariance functions for different values of ν.

3.4 Application to Gaussian process regression

In this section, we will show how to apply the approximation in a Gaussian process regression setting.

We recall from section 2.1 that the problem we want to find a solution to is that given observed data $\mathcal{D} = (X, \boldsymbol{y})$ we want to find the posterior $p(f(\boldsymbol{x}_*) \mid \boldsymbol{y})$. As in section 2.1 we assume a Gaussian likelihood.

In a standard GP regression setting, we would then specify an appropriate covariance function, $k(\boldsymbol{x}, \boldsymbol{x}')$, and find the closed-form solution given by

$$p(f(\boldsymbol{x}_*) \mid \boldsymbol{y}) = \mathcal{N}\left(f(\boldsymbol{x}_*) \mid \mathbb{E}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}], \mathbb{V}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}]\right), \tag{3.33}$$

where

$$\mathbb{E}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}] = \boldsymbol{k}_*^\top (K + \sigma^2 I)^{-1} \boldsymbol{y} \tag{3.34}$$

$$\mathbb{V}[f(\boldsymbol{x}_*) \mid \boldsymbol{y}] = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top (K + \sigma^2 I)^{-1} \boldsymbol{k}_*, \tag{3.35}$$

which involves inverting the $n \times n$ matrix, which has a computational complexity of $\mathcal{O}(n^3)$.

Using the HS approximation, we can estimate $k(\boldsymbol{x}, \boldsymbol{x}')$ with $\tilde{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{m} S(\sqrt{\lambda_j})\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}')$, such that we obtain $K \approx \tilde{K} = \Phi\Lambda\Phi^\top$ for a diagonal $m \times m$ matrix Λ with $\Lambda_{ii} = S(\sqrt{\lambda_i})$

and $n \times m$ matrix Φ where $\Phi_{i,j} = \phi_j(\boldsymbol{x}_i)$. Now we may sample f from $GP(0, \tilde{K})$ as

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \alpha_j \phi_j(\boldsymbol{x}) = \Phi\boldsymbol{\alpha}, \quad p(\alpha_j) = \mathcal{N}(\alpha_j \,|\, 0, S(\sqrt{\lambda_j})).$$

This corresponds to a Bayesian linear model as discussed in section 2.1.3, meaning that we now have projected the infinite-dimensional Gaussian process to an m-dimensional linear model using the eigenfunctions of the negative Laplacian as the feature maps.

We can now find the posterior predictive distribution of f directly from equation (2.17).

$$\begin{aligned} \mathbb{E}[f(\boldsymbol{x}_*) \,|\, \boldsymbol{y}] &= \phi_*^\top (\Phi^\top \Phi + \sigma^2 \Lambda^{-1})^{-1} \Phi^\top \boldsymbol{y} \\ \mathbb{V}[f(\boldsymbol{x}_*) \,|\, \boldsymbol{y}] &= \sigma^2 \phi_*^\top (\Phi^\top \Phi + \sigma^2 \Lambda^{-1})^{-1} \phi_*. \end{aligned} \tag{3.36}$$

When computing the posterior mean and variance, we now only have to find the inverse of an $m \times m$ matrix instead of an $n \times n$ matrix. As the $\Phi^\top \Phi$ matrix product is independent of hyperparameters, you only need to calculate this product once, which will be beneficial in hyperparameter optimisation. We will give more details on this in section 3.5.

There are two new parameters introduced with the HS approximation that you have to be aware of. The domain length of the Dirichlet boundary problem, given by $2L$ and the number of basis functions m. In figure 3.3, we have visualised a simple example where we compare the application of a full Gaussian process with the HS approximation for different numbers of basis functions on a simple dataset. On the left side, we see the prior distributions, and on the right side, we see the posterior distributions after fitting to the five data points. We use a squared exponential covariance function with the same hyperparameters for all models. These have been found by hyperparameter optimisation on the full model and are $\sigma = 0.0498$, $\kappa = 0.81$ and $\ell = 1.61$. The plot serves to illustrate the practical consequences of the choice of L and m. Regarding m, it is easy to see that $m = 2$ isn't able to capture the shape of the data as the basis functions are too smooth. However, letting m grow to $4, 8$ and 16, we get closer and closer to the full model inside the domain $x \in [-10, 10]$.

In this example, we have $L = 10$, which means the prior and posterior variance converges to zero when $x \to \pm 10$. If we where to investigate the process for $x \in [5, 10)$ we should probably have chosen a larger L. In section 4.1, we give a more thorough analysis of the convergence for $L, m \to \infty$.

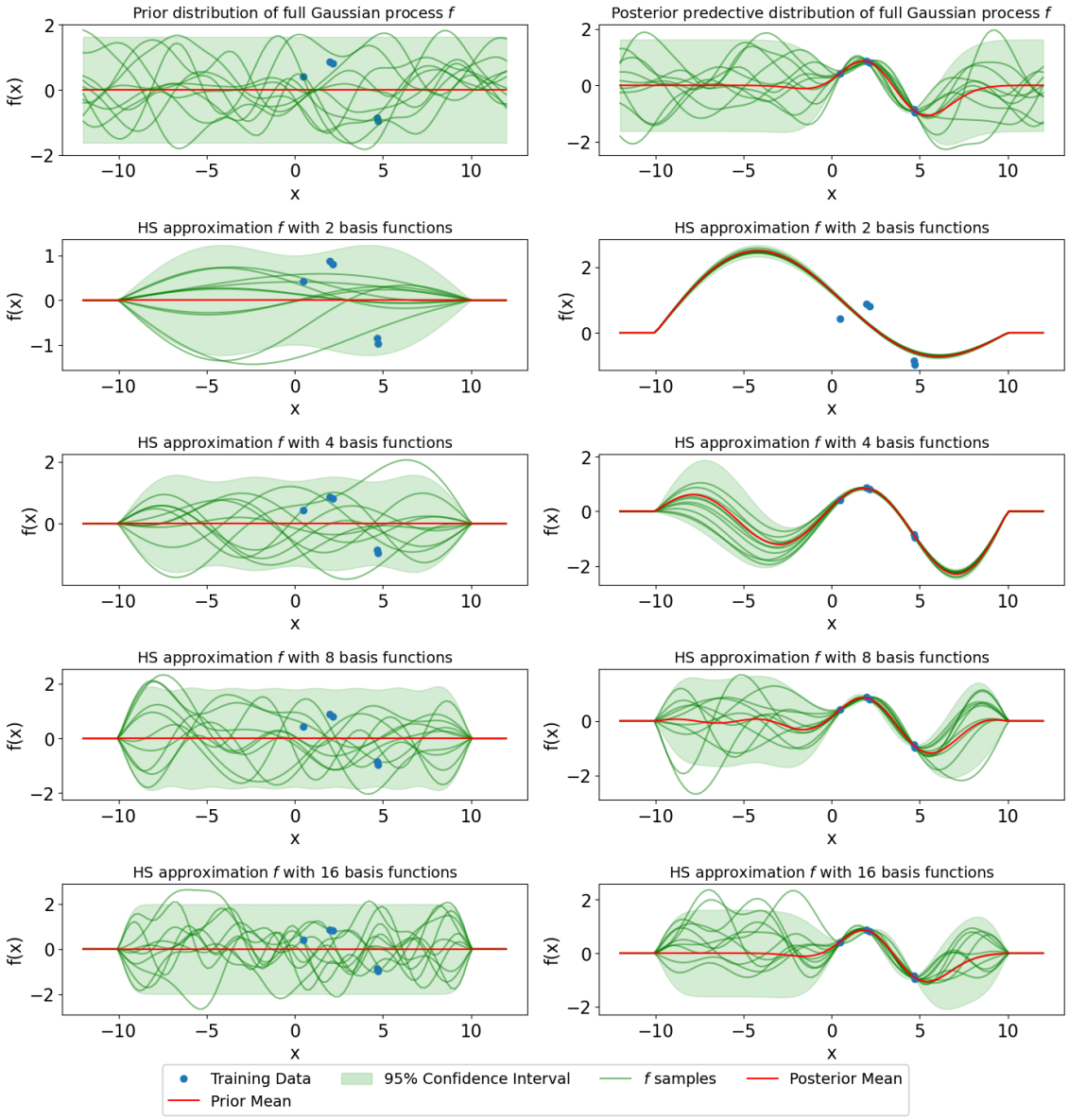


Figure 3.3: Example of applying the HS approximation for regression for different numbers of basis functions m. In the first row, we have a standard full Gaussian process. The prior distribution is plotted in the left column, and the posterior distribution is plotted in the right column.

3.5 Hyperparameter optimization

Solin and Särkkä (2020) show how to reduce the computational cost when optimizing the hyperparameters using the HS approximation. When running a hyperparameter optimization of the full rank Gaussian process from section 2.1.2, the cost of each step of is $\mathcal{O}(n^3)$ since we have to find the inverse of an $n \times n$ matrix. If we instead insert $\tilde{\boldsymbol{Q}} = \Phi\Lambda\Phi^\top + \sigma^2 I$ into equation (2.13) and (2.14) we obtain

$$\mathcal{L} = \frac{1}{2}\log|\tilde{\boldsymbol{Q}}| + \frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y} + \frac{2}{n}\log(2\pi), \tag{3.37}$$

with the derivates

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\theta_k} =& \frac{1}{2}\mathrm{Tr}\left(\tilde{\boldsymbol{Q}}^{-1}\frac{\partial\tilde{\boldsymbol{Q}}}{\partial\theta_k}\right) - \frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\frac{\partial\tilde{\boldsymbol{Q}}}{\partial\theta_k}\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y},\\ \frac{\partial\mathcal{L}}{\partial\sigma^2} =& \frac{1}{2}\mathrm{Tr}\left(\tilde{\boldsymbol{Q}}^{-1}\right) - \frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y}.\end{aligned} \tag{3.38}$$

We can find the following expressions for terms that involve $\log|\tilde{\boldsymbol{Q}}|$:

$$\begin{aligned}\log|\tilde{\boldsymbol{Q}}| &= (n-m)\log(\sigma^2) + \log|\boldsymbol{Z}| + \sum_{j=1}^{m}\log\left(S(\sqrt{\lambda_i})\right),\\ \frac{\partial\log|\tilde{\boldsymbol{Q}}|}{\partial\theta_k} &= \sum_{j=1}^{m} S\left(\sqrt{\lambda_i}\right)^{-1}\frac{\partial S\left(\sqrt{\lambda_i}\right)}{\partial\theta_k} - \sigma^2\mathrm{Tr}\left(\boldsymbol{Z}^{-1}\boldsymbol{\Lambda}^{-2}\frac{\partial\boldsymbol{\Lambda}}{\partial\theta_k}\right),\\ \frac{\partial\log|\tilde{\boldsymbol{Q}}|}{\partial\sigma^2} &= \frac{n-m}{\sigma^2} + \mathrm{Tr}(\boldsymbol{Z}^{-1}\boldsymbol{\Lambda}^{-1}),\end{aligned} \tag{3.39}$$

and for the terms that involve $\tilde{\boldsymbol{Q}}^{-1}$:

$$\begin{aligned}\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y} &= \frac{1}{\sigma^2}(\boldsymbol{y}^\top\boldsymbol{y} - \boldsymbol{y}^\top\Phi\boldsymbol{Z}^{-1}\Phi^\top\boldsymbol{y}),\\ \frac{\partial\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y}}{\partial\theta_k} &= -\boldsymbol{y}^\top\Phi\boldsymbol{Z}^{-1}\left(\boldsymbol{\Lambda}^{-2}\frac{\partial\boldsymbol{\Lambda}}{\partial\theta_k}\right)\boldsymbol{Z}^{-1}\Phi^\top\boldsymbol{y},\\ \frac{\partial\boldsymbol{y}^\top\tilde{\boldsymbol{Q}}^{-1}\boldsymbol{y}}{\partial\sigma^2} &= \frac{1}{\sigma^2}\boldsymbol{y}^\top\Phi\boldsymbol{Z}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^{-1}\Phi^\top\boldsymbol{y} - \frac{1}{\sigma^4}\boldsymbol{y}^\top\boldsymbol{y},\end{aligned} \tag{3.40}$$

where $\boldsymbol{Z} = \sigma^2\boldsymbol{\Lambda}^{-1} + \Phi^\top\Phi$.

These expressions are computationally superior to (2.13) and (2.14) when $m < n$ as we only need to find the inverse of an $m \times m$ matrix, $\boldsymbol{Z}$, instead of the $n \times n$ matrix $\boldsymbol{Q}$. In practice, we use Cholesky factorization to derive the inverse of $\boldsymbol{Z}$. Another benefit of the expressions above is that we only need to calculate the matrix product $\Phi^\top\Phi$ once, as it is not dependent on the hyperparameters. Therefore, the initial cost of computational complexity is $\mathcal{O}(nm^2)$. The cost for each step in the hyperparameter optimization consists of evaluating the marginal likelihood and the corresponding gradient, which, due to the Cholesky factorization, is an $\mathcal{O}(m^3)$ operation.

3.6 Numerical stability and implementation

In this section, we wish to enlighten some aspects of the implementation of the model. In equation (3.36) and in the hyperparameter optimisation described in section 3.5, we need to find the inverse of $\boldsymbol{Z} = \sigma^2\boldsymbol{\Lambda}^{-1} + \Phi^T\Phi$, which we in practice find using Cholesky

factorisation. However, in order to perform the Cholesky factorisation, we need the matrix $\boldsymbol{Z}$ to be positive definite, and this might not be guaranteed due to numerical instability. Therefore, we use a reformulation presented in section 3.4.3 in Rasmussen and Williams (2006), which secures that the matrix is indeed positive definite.
That is, if we define a matrix, B, such that

$$B = W^{\frac{1}{2}} K W^{\frac{1}{2}} + I \tag{3.41}$$

where W is a diagonal matrix. Then B is well conditioned, and it is numerically safe to compute its Cholesky decomposition $LL^T = B$.
If we let $K = \Phi^T \Phi$ and $W = \sigma^{-2}\Lambda$ we can obtain the desired expression by the following reformulation

$$\begin{aligned}
(\Phi^T\Phi + \sigma^2\boldsymbol{\Lambda}^{-1})^{-1} &= \underbrace{\left(\sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\left(\sigma\boldsymbol{\Lambda}^{-\frac{1}{2}}\right)}_{I = W^{\frac{1}{2}} W^{-\frac{1}{2}}} (\Phi^T\Phi + \sigma^2\boldsymbol{\Lambda}^{-1})^{-1} \left(\sigma\boldsymbol{\Lambda}^{-\frac{1}{2}}\right)\left(\sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}}\right) \\
&= \sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}} \left(\left(\sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)\Phi^T\Phi\left(\sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}}\right) + I\right)^{-1} \sigma^{-1}\boldsymbol{\Lambda}^{\frac{1}{2}} \\
&= W^{\frac{1}{2}} B^{-1} W^{\frac{1}{2}}
\end{aligned} \tag{3.42}$$

Then we can safely perform the Cholesky factorisation on B (defined as in 3.41) in order to find the inverse, B^{-1}. Also, note that this does not increase the computational complexity.

4 Analytical insights on the properties of the Hilbert space approximation

In this chapter, we will delve deeper into the properties of Hilbert space approximation in order to answer our first research question:

How does the Hilbert space approximation affect predictive precision and computational complexity compared to a full Gaussian process?

We will answer the question through a detailed examination of the model's convergence properties and an investigation of the model's generalization error through *average-case learning curves*. The aim of section 4.1 is to unfold the mathematical arguments in the proofs on the convergence theorems in Solin and Särkkä (2020). Therefore, this section consists of several mathematical derivations and proofs, which we have tried to present in a way that is as readable as possible without compromising the mathematical details. In section 4.2 we wish to make a general investigation on how the HS approximation acts on different sizes of datasets compared to the asymptotic behaviour of a full Gaussian process. The results in this section are relevant with respect to the Hilbert space approximation but not directly linked to the construction of shape-constrained models in chapter 5, and not necessary to read in order to understand the underlying theory on the models we propose in that chapter.

4.1 Convergence theorems

In this section, we wish to show that the approximated Gaussian process converges to the full Gaussian process when the number of terms in the series expansions goes towards infinity and the domain goes towards $\mathbb{R}$. That is when $L \to \infty$ and $m \to \infty$. First, we show it for the univariate case and then generalise to a multidimensional setting. The following results are presented in Solin and Särkkä (ibid.), and this section aims to restate the results with some more in-depth argumentation.

4.1.1 Univariate case

We start by considering the one-dimensional case where $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a stationary covariance function. The squared exponential and the Matérn kernels are both examples of stationary covariance functions.

Throughout this section, we further assume that

i) $S(\omega)$ is bounded and integrable with $\int_0^\infty S(\omega)\,\mathrm{d}\omega = A$,

ii) $|S'(\omega)|$ is bounded and integrable with $\int_0^\infty |S'(\omega)|\,\mathrm{d}\omega = B$.

We also assume that the training and test data is contained within the interval $\left[-\tilde{L}, \tilde{L}\right]$ such that $\tilde{L} \leq L$.

With the assumptions above, we formulate the univariate convergence in the following theorem:

Theorem 4.1.1 (Univariate convergence) *There exists a constant E which is independent of m, x and x' such that*

$$\left|k(x, x') - \tilde{k}(x, x')\right| \leq \frac{E}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega, \tag{4.1}$$

from which it follows that

$$\lim_{L\to\infty}\left[\lim_{m\to\infty}\tilde{k}_m(x,x')\right] = k(x,x'). \tag{4.2}$$

It is important to note that we cannot switch the order of the limits. This is easy to verify analytically as

$$\lim_{m\to\infty}\left[\lim_{L\to\infty}\left[\frac{E}{L}+\frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right]\right] = \lim_{m\to\infty}\left[\frac{2}{\pi}\int_0^{\infty} S(\omega)\,\mathrm{d}\omega\right] = \frac{2}{\pi}\int_0^{\infty} S(\omega)\,\mathrm{d}\omega, \tag{4.3}$$

where we see that the limit doesn't converge to zero, when switching the order of m and L.

In practice, this means that although increasing m always leads to a better approximation, an increase in L may lead to a worse approximation if m is not increased with it. Ideally, we need to increase m and L in a way such that $\frac{m}{L} \to \infty$. We illustrate this issue in figure 4.1 by plotting the Kullback-Leibler (KL) divergence between the true and approximated posterior as a function of m for different values of L. The KL divergence between two distributions p and q is

$$\mathrm{KL}(p\,\|\,q) = -\int p(x)\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x \tag{4.4}$$

and measure the dissimilarity between the two distributions p and q. If $\mathrm{KL}(p\,\|\,q) = 0$, it means that $p = q$ almost everywhere. See Bishop (2006) page 56-57 for more details.

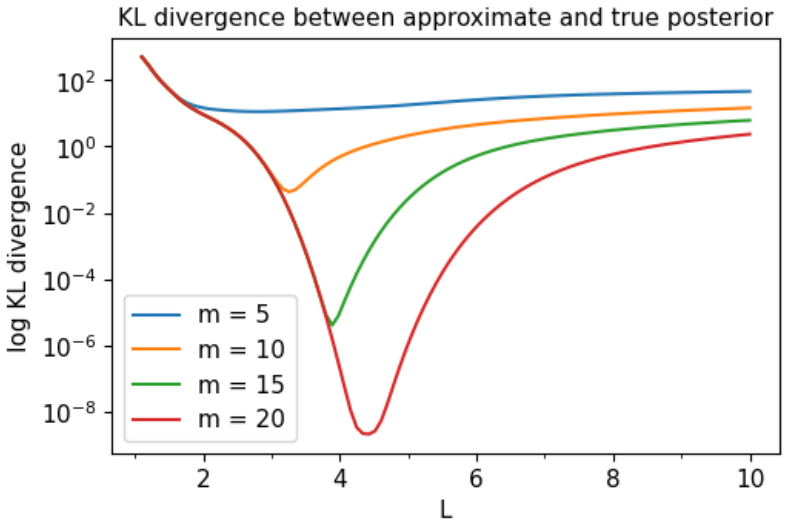


Figure 4.1: The KL divergence between the approximate and true posterior of a Gaussian process regression on a simple dataset. The KL divergence is plotted as a function of L for different values of m. We generate training data by sampling a Gaussian process prior and adding Gaussian noise to it. Although the KL decreases for increasing m, increasing L only causes a decrease in the KL divergence up to a certain point.

We will now give a proof of theorem 4.1.1. The proof is somewhat technical but can be broken into the following four steps:

Rewriting the covariance function in terms of the spectral density. In this step, we will use the Wiener-Khinchin identity to write the covariance function as an integral of the spectral density multiplied by a cosine function.

Showing that $\left|\tilde{k}_\infty(x,x') - k(x,x')|\right| \leq \frac{D_1}{L}$. We assume that we have infinitely many basis functions at our disposal and look at the convergence as a function of L. It is the most demanding part of the proof. We rewrite the Hilbert space approximation and split it into three terms. Each term consists of an infinite series. Using the triangle inequality, we can treat each term individually.

Showing that $\left|\tilde{k}_\infty(x,x') - \tilde{k}_m(x,x')\right| \leq \frac{D_5}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega$*:* Here we investigate the convergence as a function of m.

Proving theorem 1: With the two previous results in hand, we can use the triangle inequality to prove that $\left|\tilde{k}_m(x,x') - k(x,x')|\right| \to 0$ for $m \to infty$, $L \to \infty$.

Rewriting the covariance function.
We start by looking at the true covariance function.
Using the Wiener–Khinchin identity, we can write the covariance function as

$$k(x,x') = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)\exp(-i\omega(x-x'))\,\mathrm{d}\omega. \tag{4.5}$$

Let us rewrite this, first by using Euler's formula,

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)(\cos(\omega(x-x')) + i\sin(\omega(x-x')))\,\mathrm{d}\omega. \tag{4.6}$$

Since $\sin(x)$ is an odd function and $S(\omega)$ of a stationary covariance function is even, the integral on $(-\infty,\infty)$ vanishes and we can write

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega. \tag{4.7}$$

The symmetry of the spectral density now allows us to write

$$= \frac{1}{\pi}\int_{0}^{\infty} S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega. \tag{4.8}$$

We will use this expression of the covariance in the following. We are ready to continue to the next step.

Showing that $\left|\tilde{k}_\infty(x,x') - k(x,x')|\right| \leq \frac{D_1}{L}$**.**
In a one-dimensional domain $\Omega = [-L,L]$ with zero-Dirichlet boundary conditions we formulate the explicit m-term approximation by inserting the eigenvalues and eigenfunctions from (2.26) and (2.27) with $d = 1$ directly into the covariance function approximation from (3.32).

$$\tilde{k}_m(x,x') = \sum_{j=1}^{m} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\sin\left(\frac{\pi j(x+L)}{2L}\right)\sin\left(\frac{\pi j(x'+L)}{2L}\right). \tag{4.9}$$

Now we are ready to let $m = \infty$ and analyse the convergence for a growing domain, $\Omega = [-L,L]$,

$$\tilde{k}_\infty(x,x') = \sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\sin\left(\frac{\pi j(x+L)}{2L}\right)\sin\left(\frac{\pi j(x'+L)}{2L}\right). \tag{4.10}$$

We will now rewrite the summation in (4.10) into three separate terms. The final equations might seem a bit messy, but they allow us to show the convergence of each term individually. First we use that $\sin(a)\sin(b) = \cos(a-b) - \cos(a)\cos(b)$

$$\tilde{k}_\infty(x,x') = \sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\left[\cos\left(\frac{\pi j(x-x')}{2L}\right) - \cos\left(\frac{\pi j(x+L)}{2L}\right)\cos\left(\frac{\pi j(x'+L)}{2L}\right)\right]. \tag{4.11}$$

Then we have that $\cos(a)\cos(b) = \frac{\cos(a-b)+\cos(a+b)}{2}$ which gives us

$$\begin{aligned}\tilde{k}_\infty(x,x') &= \sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\left[\cos\left(\frac{\pi j(x-x')}{2L}\right) - \frac{\cos\left(\frac{\pi j(x-x')}{2L}\right)}{2} - \frac{\cos\left(\frac{\pi j(x+x'+2L)}{2L}\right)}{2}\right] \\ &= \sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{2L}\left[\cos\left(\frac{\pi j(x-x')}{2L}\right) - \cos\left(\frac{\pi j(x+x')}{2L} + \pi j\right)\right].\end{aligned} \tag{4.12}$$

Now, we split the sum into three sums. One containing the first cosine term, and two consisting of respectively the even and odd j's of the second cosine term in order to use that $\cos(a+j\pi) = -\cos(a)$ for odd j's and $\cos(a+j\pi) = \cos(a)$ for even j's.

$$\begin{aligned}\tilde{k}_\infty(x,x') = &\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{2L}\cos\left(\frac{\pi 2j(x-x')}{2L}\right) \\ &- \sum_{j=1}^{\infty} S\left(\frac{\pi 2j}{2L}\right)\frac{1}{2L}\cos\left(\frac{\pi 2j(x+x')}{2L}\right) \\ &+ \sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\frac{1}{2L}\cos\left(\frac{\pi(2j-1)(x+x')}{2L}\right).\end{aligned} \tag{4.13}$$

Now we add and subtract the sum $\sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\cos\left(\frac{\pi 2j(x+x')}{2L}\right)$ and by recognizing common factors we achieve the following three sums.

$$\begin{aligned}\tilde{k}_\infty(x,x') = &\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\cos\left(\frac{\pi j(x-x')}{2L}\right)\frac{1}{2L} \\ &- \frac{1}{2L}\sum_{j=1}^{\infty}\left[S\left(\frac{\pi 2j}{2L}\right) - S\left(\frac{\pi(2j-1)}{2L}\right)\right]\cos\left(\frac{\pi 2j(x+x')}{2L}\right) \\ &- \frac{1}{2L}\sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\left[\cos\left(\frac{\pi 2j(x+x')}{2L}\right) - \cos\left(\frac{\pi(2j-1)(x+x')}{2L}\right)\right].\end{aligned} \tag{4.14}$$

After this initial preparation, we will compare (4.8) and (4.14) in the following proposition.

Proposition 4.1.1 *Assume that for $\omega \geq 0$, $S(\omega)$ is a positive bounded integrable function such that $\int_0^\infty S(\omega)\,\mathrm{d}\omega = A < \infty$ and that for $\omega > 0$, $|S'(\omega)|$ is bounded and integrable with $\int_0^\infty |S'(\omega)|\,\mathrm{d}\omega = B < \infty$.*

Then there exists a constant D_1, such that for all $x, x' \in [-\tilde{L}, \tilde{L}]$ we have

$$\left|\tilde{k}_\infty(x,x') - k(x,x')\right| = \left|\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\sin\left(\frac{\pi j(x+L)}{2L}\right)\sin\left(\frac{\pi j(x'+L)}{2L}\right) - \frac{1}{\pi}\int_0^\infty S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega\right| \leq \frac{D_1}{L}. \tag{4.15}$$

Proof. Using (4.14) and the triangle inequality, we get the following on three terms.

$$\left|\tilde{k}_\infty(x,x') - k(x,x')\right| \leq$$

$$\left|\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\cos\left(\frac{\pi j(x-x')}{2L}\right)\frac{1}{2L} - \frac{1}{\pi}\int_0^\infty S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega\right| \tag{4.16}$$

$$-\left|\frac{1}{2L}\sum_{j=1}^{\infty}\left[S\left(\frac{\pi 2j}{2L}\right) - S\left(\frac{\pi(2j-1)}{2L}\right)\right]\cos\left(\frac{\pi 2j(x+x')}{2L}\right)\right| \tag{4.17}$$

$$-\left|\frac{1}{2L}\sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\left[\cos\left(\frac{\pi 2j(x+x')}{2L}\right) - \cos\left(\frac{\pi(2j-1)(x+x')}{2L}\right)\right]\right| \tag{4.18}$$

$$\leq \frac{D_2}{L} + \frac{D_3}{L} + \frac{D_4}{L} = \frac{D_1}{L}. \tag{4.19}$$

In order to find the three bounds, D_2, D_3 and D_4 in (4.19) we will now analyse the sum terms (4.16), (4.17) and (4.18) one by one.

We start by looking at (4.16). To begin with, we wish to state the following Lemma.

Lemma 4.1.1 *Let $\Delta > 0$ and $\alpha \in [0,1)$ be given constants, $m = 0, 1, 2, \cdots$ some non-negative integer, and assume that $f(\omega)$ is a bounded integrable function defined on $\omega \geq m\Delta$ with bounded derivative on the same domain such that $\int_{m\Delta}^\infty |f'(\omega)|\,\mathrm{d}\omega = K^{(m)} < \infty$, where $K^{(m)} = \int_{m\Delta}^\infty |f'(\omega')|\,\mathrm{d}\omega'\Delta$.*

Then we have

$$\left|\int_{m\Delta}^\infty f(\omega)\,\mathrm{d}\omega - \sum_{j=m+1}^{\infty} f(j\Delta - \alpha\Delta)\Delta\right| \leq K^{(m)}\Delta. \tag{4.20}$$

Furthermore, provided that $\int_0^\infty |f'(\omega)|\,\mathrm{d}\omega = K^{(0)} < \infty$, this bound can be made independent of m.

$$\left|\int_{m\Delta}^\infty f(\omega)\;\mathrm{d}\omega - \sum_{j=m+1}^{\infty} f(j\Delta - \alpha\Delta)\Delta\right| \leq K^{(0)}\Delta. \tag{4.21}$$

The proof of lemma 4.1.1 can be found in appendix A.2.1

Next, we use the lemma to show that there exists a constant D_2 such that for all $x, x' \in [-\tilde{L}, \tilde{L}]$ we have

$$\left|\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\cos\left(\frac{\pi j(x-x')}{2L}\right)\frac{1}{2L} - \frac{1}{\pi}\int_0^\infty S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega\right| \leq \frac{D_2}{L} \tag{4.22}$$

By using Lemma 4.1.1 with $\Delta = \frac{\pi}{2L}$, $f(\omega) = \frac{1}{\pi}S(\omega)\cos(\omega(x-x'))$, $m = 0$, and $\alpha = 0$ as well as the assumptions on $S(\omega)$ and boundedness of sine and cosine we get that

$$\begin{aligned}
&\frac{1}{\pi}\left|\sum_{j=1}^{\infty} S\left(\frac{\pi j}{2L}\right)\cos\left(\frac{\pi j(x-x')}{2L}\right)\frac{\pi}{2L} - \int_0^{\infty} S(\omega)\cos(\omega(x-x'))\,\mathrm{d}\omega\right| \\
&\leq \frac{1}{\pi}\int_0^{\infty}\left|\left(S(\omega)\cos(\omega(x-x'))\right)'\right|\,\mathrm{d}\omega\frac{\pi}{2L} \\
&= \frac{1}{2L}\int_0^{\infty}\left|S'(\omega)\cos(\omega(x-x')) - S(\omega)(x-x')\sin(\omega(x-x'))\right|\,\mathrm{d}\omega \\
&\leq \frac{1}{2L}\int_0^{\infty}\left|S'(\omega)\right|d\omega + \frac{(x-x')}{2L}\int_0^{\infty}\left|S(\omega)\right|d\omega \\
&\leq \frac{1}{2L}\int_0^{\infty}\left|S'(\omega)\right|d\omega + \frac{\tilde{L}}{L}\int_0^{\infty}\left|S(\omega)\right|d\omega < \frac{1}{2L}B + \frac{\tilde{L}}{L}A
\end{aligned} \tag{4.23}$$

where we have used the triangle inequality and that $x, x' \in [-\tilde{L}, \tilde{L}]$. Then we have that $D_2 = \frac{B}{2} + \tilde{L}A$

Now we look at (4.17), and just as before, we begin by stating the following Lemma.

Lemma 4.1.2 *Assume that $f(\omega)$ is a bounded integrable function defined on $\omega \geq 0$ with bounded derivative on $\omega > 0$ and $g(\omega)$ is a bounded function defined on $\omega \geq 0$ such that $|g(w)| \leq K_2$. Further assume that $\int_0^{\infty}|f'(\omega)|\,\mathrm{d}\omega = K_1 < \infty$. Then for any $\alpha, \beta \in [0, 1)$ and $\Delta > 0$ we have*

$$\left|\sum_{j=1}^{\infty}[f(j\Delta) - f(j\Delta - \alpha\Delta)]g(j\Delta - \beta\Delta)\right| \leq K_1K_2. \tag{4.24}$$

The proof of lemma 4.1.2 can be found in appendix A.2.2.

We apply Lemma 4.1.2 to show that there exists a constant D_3 such that.

$$\left|\frac{1}{2L}\sum_{j=1}^{\infty}\left[S\left(\frac{\pi 2j}{2L}\right) - S\left(\frac{\pi(2j-1)}{2L}\right)\right]\cos\left(\frac{\pi 2j(x+x')}{2L}\right)\right| \leq \frac{D_3}{L}. \tag{4.25}$$

We use Lemma 4.1.2 with $\Delta = \frac{\pi}{L}, \alpha = \frac{1}{2}, \beta = 0, f(\omega) = S(\omega)$ and $g(\omega) = \cos(\omega(x-x'))$. Since $|\cos(x)| \leq 1$ for all $x \in \mathbb{R}$ the result follows directly from Lemma 4.1.2 with $D_3 = \frac{B}{2}$.

For the third sum (4.18), we begin by stating the following Lemma.

Lemma 4.1.3 *Assume that $f(\omega) \geq 0$ is a positive bounded integrable function defined on $\omega \geq 0$ with bounded derivative on $\omega > 0$ such that $\int_0^{\infty} f(\omega)\,\mathrm{d}\omega = K_0 < \infty$ and $\int_0^{\infty}|f'(\omega)|\,\mathrm{d}\omega = K_1 \leq \infty$, and $g(\omega)$ is a bounded integrable function defined on $\omega \geq 0$ with bounded derivative on $\omega > 0$ such that $|g'(\omega)| \leq K_2$. Then for any $\alpha, \beta \in [0, 1)$ and $\Delta > 0$ we have for $K = K_1 + K_0$:*

$$\left|\sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)[g(j\Delta) - g(j\Delta - \beta\Delta)]\right| \leq KK_2. \tag{4.26}$$

The proof of lemma 4.1.3 can be found in appendix A.2.3.

Now we are ready to show that there exists a constant D_4 such that

$$\left|\frac{1}{2L}\sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\left[\cos\left(\frac{\pi 2j(x+x')}{2L}\right)-\cos\left(\frac{\pi(2j-1)(x+x')}{2L}\right)\right]\right| \leq \frac{D_4}{L}. \quad (4.27)$$

This follows by using Lemma 4.1.3 with $\Delta = \frac{\pi}{L}, \alpha = \frac{1}{2}, \beta = \frac{1}{2}, f(\omega) = S(\omega)$ and $g(\omega) = \cos(\omega(x+x'))$. Then we get

$$\begin{aligned}\frac{1}{2L}\left|\sum_{j=1}^{\infty} S\left(\frac{\pi(2j-1)}{2L}\right)\left[\cos\left(\frac{\pi 2j(x+x')}{2L}\right)-\cos\left(\frac{\pi(2j-1)(x+x')}{2L}\right)\right]\right| \\ \leq \frac{(A+B)\tilde{L}}{L},\end{aligned} \quad (4.28)$$

where $2\tilde{L}$ is an upper bound for $|(x+x')\sin(\omega(x+x'))|$ since sine is bounded by 1 and $(x+x')$ is bounded by $2\tilde{L}$. Thus $D_4 = (A+B)\tilde{L}$.

Finally, we can combine the three above results to show that

$$\left|\tilde{k}_\infty(x,x') - k(x,x')\right| \leq \frac{D_2}{L} + \frac{D_3}{L} + \frac{D_4}{L} = \frac{D_1}{L}. \quad (4.29)$$

The bound D_1 is explicitly given from $D_2 = \frac{B}{2} + \tilde{L}A$, $D_3 = \frac{B}{2}$ and $D_4 = (A+B)\tilde{L}$ yielding $D_1 = B + (2A+B)\tilde{L}$, which finishes the proof of Proposition 4.1.1.

This brings us to the next step.

Showing that $\left|\tilde{k}_\infty(x,x') - \tilde{k}_m(x,x')\right| \leq \frac{D_5}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega$.
Until now, we have assumed $m = \infty$, but we still need to consider what happens when we consider a finite sum consisting of m terms. We will do this by proving the following proposition.

Proposition 4.1.2 *Assume that on $\omega \geq 0$, $S(\omega)$ is bounded and integrable and on $\omega > 0$ it has a bounded derivative. Then there exists a constant D_5 such that for $x, x' \in [-\tilde{L}, \tilde{L}]$ we have*

$$|\tilde{k}_\infty(x,x') - \tilde{k}_m(x,x')| \leq \frac{D_5}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega. \quad (4.30)$$

Proof. We have that

$$|\tilde{k}_\infty(x,x') - \tilde{k}_m(x,x')| = \left|\sum_{j=m+1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\sin\left(\frac{\pi j(x+L)}{2L}\right)\sin\left(\frac{\pi j(x'+L)}{2L}\right)\right| \quad (4.31)$$

$$\leq \left|\sum_{j=m+1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L}\right|. \quad (4.32)$$

since $\sin(x) \leq 1$ for all $x \in \mathbb{R}$. We notice that we can use Lemma 4.1.1 on the following equation with $f(\omega) = \frac{2}{\pi}S(\omega)$ and $\Delta = \frac{\pi}{2L}$

$$\left|\sum_{j=m+1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L} - \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right| \leq B\frac{\pi}{2L} = \frac{D_5}{L}. \quad (4.33)$$

Therefore we continue from (4.31) by adding and subtracting $\frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega$ such that we have

$$\left|\sum_{j=m+1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L} - \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right|. \tag{4.34}$$

By using the triangle inequality, we can now obtain the desired result

$$\leq \left|\sum_{j=m+1}^{\infty} S\left(\frac{\pi j}{2L}\right)\frac{1}{L} - \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right| + \left|\frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right| \tag{4.35}$$

$$\leq \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} |S'(\omega)|\,\mathrm{d}\omega + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega = \frac{D_5}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega. \tag{4.36}$$

We are now ready to move on to the final step.

Proving theorem 1.
Proof of Theorem 1.
The first part follows from Proposition 4.1.1 and 4.1.2 since we have

$$\begin{aligned}\left|k(x,x') - \tilde{k}_m(x,x')\right| &\leq |k(x,x') - \tilde{k}_\infty(x,x')| + |\tilde{k}_\infty(x,x') - \tilde{k}_m(x,x')| \\ &\leq \frac{D_1}{L} + \frac{D_5}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega \qquad (4.37)\\ &= \frac{E}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega \qquad (4.38)\end{aligned}$$

where $D_1 + D_5 = E$.

For the second part, we only need to show that (4.38) goes towards zero when we first let $m \to \infty$ and then let $L \to \infty$. From the assumptions on $S(\omega)$, stated at the beginning of the section, we have that $\lim_{m\to\infty} \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega = 0$, and therefore we have

$$\lim_{L\to\infty}\left[\lim_{m\to\infty}\left(\frac{E}{L} + \frac{2}{\pi}\int_{\frac{\pi m}{2L}}^{\infty} S(\omega)\,\mathrm{d}\omega\right)\right] = \lim_{L\to\infty}\left[\frac{E}{L}\right] = 0. \tag{4.39}$$

Thus

$$\lim_{L\to\infty}\left[\lim_{m\to\infty}\tilde{k}_m(x,x')\right] = k(x,x') \tag{4.40}$$

which concludes the proof.

4.1.2 Multivariate case

Next, we consider the multivariate case. This section builds on top of section 4.1.1 by showing that we can apply theorem 4.1.1 on one dimension at a time. However, as the number of eigenfunctions and eigenvalues scales exponentially for growing dimensions, there are several more terms to deal with, which results in rather tedious derivations.

Let $\boldsymbol{x} \in \mathbb{R}^d$ and consider a hyperbox $\Omega = [-L_1, L_1]\times\cdots\times[-L_d, L_d]$ with Dirichlet boundary conditions. We recall that the solution to the eigenvalue problem for the Laplace Operator is given by

$$\phi_{j_i,\ldots,j_d}(\boldsymbol{x}) = \prod_{k=1}^{d}\phi_{j_k}(x_k), \qquad \lambda_{j_1,\ldots,j_d} = \sum_{k=1}^{d}\lambda_{j_k}, \tag{4.41}$$

where ϕ_{j_k} and λ_{j_k} are defined as in 2.26 and 2.27. This means that if we consider an approximation with $\hat{m}$ basis functions, this will correspond to $m = \hat{m}^d$ combinations of ϕ_{j_k}'s. We compute the explicit multivariate m-term HS approximation by inserting (2.26) and (2.27) directly. Then we obtain

$$\begin{aligned}\tilde{k}_m(\boldsymbol{x}, \boldsymbol{x}') &= \sum_{j_1,\ldots,j_d=1}^{\hat{m}} S\left(\sqrt{\lambda_{j_1,\ldots,j_d}}\right) \phi_{j_1,\ldots,j_d}(\boldsymbol{x})\phi_{j_1,\ldots,j_d}(\boldsymbol{x}') \\ &= \sum_{j_1,\ldots,j_d=1}^{\hat{m}} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \prod_{k=1}^{d} \frac{1}{L_k} \sin\left(\sqrt{\lambda_{j_k}}(x_k + L)\right) \sin\left(\sqrt{\lambda_{j_k}}(x_k' + L)\right).\end{aligned} \tag{4.42}$$

Note that $\sum_{j_1,\ldots,j_d=1}$ is a shorthand for $\sum_{j_1=1}^{\hat{m}} \cdots \sum_{j_d=1}^{\hat{m}}$, and that we have abused notation slightly by writing $S\left(\lambda_{j_1},\ldots,\lambda_{d}\right)$ rather than $S\left(\sqrt{\lambda_{j_1,\ldots,j_d}}\right)$. This is because it is more convenient to consider S as a function of the individual λ_{j_k}'s in the following.

We assume that the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ is isotropic as we are in a multivariate setting. We further assume that if we fix $\omega_1, \ldots, \omega_{k-1}, \omega_{k+1}, \ldots, \omega_d$ and consider $S(\omega_1, \ldots, \omega_d)$ only as a function of ω_k, it satisfies the assumptions from the previous section, namely:

i) $S(\omega_1, \ldots, \omega_d)$ is bounded and integrable with $\int_0^\infty S(\omega_1, \ldots, \omega_d)\,\mathrm{d}\omega_k = A_k$,

ii) $\left|\frac{\partial}{\partial\omega_k} S(\omega_1, \ldots, \omega_d)\right|$ is bounded and integrable with $\int_0^\infty \left|\frac{\partial}{\partial\omega_k} S(\omega_1, \ldots, \omega_d)\right| \mathrm{d}\omega_k = B_k$.

We also assume that the training and test datasets are contained within a hypercube $\boldsymbol{x}, \boldsymbol{x}^* \in \left[-\tilde{L}, \tilde{L}\right]^d$ such that $\tilde{L} \leq L_k$ for each k.

With the assumptions above, we have the following result:

Theorem 4.1.2 (Multivariate convergence) *Let $L_{\min} = \min_k L_k$ ad $L_{\max} = \max_k L_k$. There exists a constant E which is independent of m, $\boldsymbol{x}$ and $\boldsymbol{x}'$ such that*

$$\left|k(\boldsymbol{x}, \boldsymbol{x}') - \tilde{k}(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \frac{Ed}{L_{min}} + \frac{1}{\pi^d} \int_{\|\omega\| \geq \frac{\pi\hat{m}}{2L_{max}}} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}, \tag{4.43}$$

from which it follows that

$$\lim_{L_1,\ldots,L_d \to \infty} \left[\lim_{m\to\infty} \tilde{k}_m(\boldsymbol{x}, \boldsymbol{x}')\right] = k(\boldsymbol{x}, \boldsymbol{x}'). \tag{4.44}$$

The proof of this roughly follows the same steps as in the univariate case, namely

1. *Rewriting the covariance function in terms of the spectral density.*
2. *Letting $m \to \infty$ and showing that $\left|\tilde{k}_\infty(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \frac{D_1 d}{L}$.*
3. *Considering the convergence as a function of m and showing that $\left|\tilde{k}_\infty(\boldsymbol{x}, \boldsymbol{x}') - \tilde{k}_m(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \frac{dD_2}{L} + \frac{1}{\pi^d}\int_{\|\boldsymbol{\omega}\| \geq \frac{\pi m}{2L}} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}$.*
4. *Proving theorem 2 using the triangle inequality.*

Rewriting the covariance function in terms of the spectral density.
We start by rewriting the covariance function in the same manner as equation (4.8):

$$
\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{x}') &= \frac{1}{(2\pi)^d} \int_{R^d} S(\boldsymbol{\omega}) \exp(-i\boldsymbol{\omega}^\top (\boldsymbol{x} - \boldsymbol{x}')) \, \mathrm{d}\boldsymbol{\omega} \\
&= \frac{1}{\pi^d} \int_0^\infty \cdots \int_0^\infty S(\omega_1, \ldots, \omega_d) \prod_{k=1}^{d} \cos(\omega_k (x_k - x_k')) \, \mathrm{d}\omega_1 \ldots \, \mathrm{d}\omega_d.
\end{aligned} \tag{4.45}
$$

which we can do by arguments analogous to those in the one-dimensional case.

This brings us to the second step, which, once again, is the most comprehensive.

Showing that $\left|\tilde{k}_\infty(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \frac{D_1 d}{L}$.
As before, we begin by assuming that $\hat{m} = \infty$ and consider the convergence for a growing domain Ω.

Proposition 4.1.3 *Let k be an isotropic covariance function satisfying the assumptions in theorem 4.1.2.*

Then there exists a constant D_1, such that for all $\boldsymbol{x}, \boldsymbol{x}' \in \left[-\tilde{L}, \tilde{L}\right]^d$ we have

$$
\left|\tilde{k}_\infty(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \frac{D_1 d}{L}, \tag{4.46}
$$

for $L = \min_k(L_k)$.

Proof: We begin the proof by noting that for each $k = 1, \ldots, d$ we can use lemma 4.1.1 in the same manner as in (4.22) to show that

$$
\begin{aligned}
&\left| \sum_{j_k=1}^{\infty} S\left(\lambda_{j_1}, \ldots, \lambda_{j_d}\right) \frac{1}{L_k} \sin\left(\sqrt{\lambda_i}(x_k + L_k)\right) \sin\left(\sqrt{\lambda_i}(x_k' + L_k)\right) \right. \\
&\quad \left. - \frac{1}{\pi} \int_0^\infty S(\lambda_{j_1}, \ldots, \omega_k, \ldots, \lambda_d) \cos(\omega_k (x_k - x_k')) \, \mathrm{d}\omega_k \right| \leq \frac{B_k + (2A_k + B_k)\tilde{L}}{L_k},
\end{aligned} \tag{4.47}
$$

where $A_k = \int_0^\infty S(\lambda_{j_1}, \ldots, \omega_k, \ldots, \lambda_{j_d}) \, \mathrm{d}\omega_k$ and $B_k = \int_0^\infty \left|\frac{\partial}{\partial \omega_k} S(\lambda_{j_1}, \ldots, \omega_k, \ldots, \lambda_d)\right| \, \mathrm{d}\omega_k$. Note that A_k and B_k are functions of $j_2, \ldots, j_d$.

In order to use the lemma, we need to 'pull out' the sums one by one, i.e. starting with rewriting $\sum_{j_1, \ldots, j_d}$ as $\sum_{j_2, \ldots, j_d} \sum_{j_1}$. From here, the proof is essentially considering only one dimension at a time and using lemma 4.1.1 repeatedly. The crux of the proof is showing that $\sum_{j_1, \ldots, j_d} A_1$ and $\sum_{j_1, \ldots, j_d} B_1$ are bounded.

We will use the shorthand $\phi_k(x_k)$ for the basis function $\sin\left(\sqrt{\lambda_{j_k}}(x_k + L_k)\right)$. Now

$$
\begin{aligned}
\left|\tilde{k}_\infty(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \boldsymbol{x}')\right| = \Bigg| &\sum_{j_2, \ldots, j_d = 1}^{\infty} \left(\sum_{j_1=1}^{\infty} S\left(\lambda_{j_1}, \ldots, \lambda_{j_d}\right) \frac{1}{L_1} \phi_1(x_1) \, \phi_1\left(x_1'\right) \right) \prod_{k=2}^{d} \frac{1}{L_k} \phi_k(x_k) \, \phi_k\left(x_k'\right) \\
&- \frac{1}{\pi^d} \int_0^\infty \cdots \int_0^\infty S(\omega_1, \ldots, \omega_d) \prod_{k=1}^{d} \cos(\omega_k (x_k - x_k')) \, \mathrm{d}\omega_1 \ldots \, \mathrm{d}\omega_d \Bigg|.
\end{aligned} \tag{4.48}
$$

The next step to be able to use (4.47) is to add and subtract $\frac{1}{\pi} \int_0^\infty S(\omega_1, \ldots, \lambda_d) \cos(\omega_1(x_1 -$

$x_1')\,\mathrm{d}\omega_1$ inside the innermost sum and use the triangle inequality:

$$
\begin{aligned}
&= \Bigg| \sum_{j_2,\ldots,j_d=1}^{\infty} \Bigg(\sum_{j_1=1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \frac{1}{L_1}\phi_1\left(x_1\right)\phi_1\left(x_1'\right) - \frac{1}{\pi}\int_0^{\infty} S(\omega_1,\ldots,\lambda_d)\cos(\omega_1(x_1-x_1')\,\mathrm{d}\omega_1 \\
&+ \frac{1}{\pi}\int_0^{\infty} S(\omega_1,\ldots,\lambda_d)\cos(\omega_1(x_1-x_1')\,\mathrm{d}\omega_1 \Bigg) \prod_{k=2}^{d} \frac{1}{L_k}\phi_k\left(x_k\right)\phi_k\left(x_k'\right) \\
&\qquad\qquad - \frac{1}{\pi^d}\int_0^{\infty}\cdots\int_0^{\infty} S(\omega_1,\ldots,\omega_d)\prod_{k=1}^{d}\cos(\omega_k(x_k-x_k'))\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d \Bigg| \\
&\leq \underbrace{\Bigg| \sum_{j_2,\ldots,j_d=1}^{\infty} \Bigg(\sum_{j_1=0}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \frac{1}{L_1}\phi_1\left(x_1\right)\phi_1\left(x_1'\right) - \frac{1}{\pi}\int_0^{\infty} S(\omega,\ldots,\lambda_d)\cos(\omega_1(x_1-x_1')\,\mathrm{d}\omega_1 \Bigg) \prod_{k=2}^{d}\frac{1}{L_k} \Bigg|}_{G} \\
&+ \Bigg| \sum_{j_2,\ldots,j_d=1}^{\infty} \Bigg(\frac{1}{\pi}\int_0^{\infty} S(\omega_1,\ldots,\lambda_d)\cos(\omega_1(x_1-x_1')\,\mathrm{d}\omega_1 \Bigg) \prod_{k=2}^{d} \frac{1}{L_k}\phi_k\left(x_k\right)\phi_k\left(x_k'\right) \\
&\qquad\qquad - \frac{1}{\pi^d}\int_0^{\infty}\cdots\int_0^{\infty} S(\omega_1,\ldots,\omega_d)\prod_{k=1}^{d}\cos(\omega_k(x_k-x_k'))\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d \Bigg|.
\end{aligned}
\tag{4.49}
$$

We want to bound the G term by using the result in (4.47). In order to do so, we have to show

$$
\sum_{j_2,\ldots,j_d=1}^{\infty} A_1 \prod_{k=2}^{d}\frac{1}{L_k} < \infty \qquad \text{and} \qquad \sum_{j_2,\ldots,j_d=1}^{\infty} B_1 \prod_{k=2}^{d}\frac{1}{L_k} < \infty
$$

Let's start by noting that $S(\omega_1,\ldots,\omega_d) \geq 0$ and therefore

$$
\begin{aligned}
\sum_{j_2,\ldots,j_d=1}^{\infty}\int_0^{\infty} S(\omega_1,\ldots,\lambda_{j_d})\,\mathrm{d}\omega_1 &= \sum_{j_3,\ldots,j_d=1}^{\infty}\left(\sum_{j_2=1}^{\infty}\int_0^{\infty} S(\omega_1,\ldots,\lambda_{j_d})\,\mathrm{d}\omega_1\right) \\
&\leq \sum_{j_3,\ldots,j_d=1}^{\infty}\left(\int_0^{\infty}\int_0^{\infty} S(\omega_1,\omega_2,\ldots,\lambda_{j_d})\,\mathrm{d}\omega_1\,\mathrm{d}\omega_2\right) \\
&\leq \ldots \\
&\leq \int_0^{\infty}\cdots\int_0^{\infty} S(\omega_1,\ldots,\omega_d)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d \\
&= \int_0^{\infty} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega} < K_1
\end{aligned}
\tag{4.50}
$$

The last equality holds since $\int_0^{\infty} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega} \leq \int_{-\infty}^{\infty} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega} = (2\pi)^d k(0) = K_1$ by the Wiener-Khinchin theorem, and then equality follows from the Fubini/Tonelli theorem.

Considering B_1, we can similarly show that

$$
\sum_{j_2,\ldots,j_d=1}^{\infty}\int_0^{\infty}\left|\frac{\partial}{\partial\omega_1}S(\omega_1,\lambda_{j_2},\ldots,\lambda_{j_d})\right|\,\mathrm{d}\omega_1 \leq \int_0^{\infty}\cdots\int_0^{\infty}\left|\frac{\partial}{\partial\omega_1}S(\omega_1,\ldots,\omega_d)\right|\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d
\tag{4.51}
$$

Consider the multiple integral

$$
\int_{(0,\infty)^d}\left|\frac{\partial}{\partial\omega_1}S(\boldsymbol{\omega})\right|\,\mathrm{d}\boldsymbol{\omega}. \tag{4.52}
$$

The restriction of this to any finite region $(0,\xi)^d$ will be finite, as $\left|\frac{\partial}{\partial\omega_1}S(\omega_1,\ldots,\omega_d)\right|$ is bounded. Furthermore, since $\int_{(0,\infty)^d} S(\omega_1,\ldots,\omega_d)\,\mathrm{d}\omega$ is finite, the tail regions of the integral will also be bounded due to the decay of S.

Therefore it follows by Fubini/Tonelli that

$$\int_0^\infty \cdots \int_0^\infty |S'(\omega_1,\ldots,\omega_d)| d\omega_1 \ldots d\omega_d = \int_{(0,\infty)^d} \left|\frac{\partial}{\partial\omega_1} S(\boldsymbol{\omega})\right| \mathrm{d}\omega \leq K_2. \tag{4.53}$$

We can use these bounds to obtain

$$\begin{aligned}
&\left| \sum_{j_2,\ldots,j_d=1}^{\infty} \left(\sum_{j_1=0}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \frac{1}{L_1} \phi_1\left(x_1\right) \phi_1\left(x_1'\right) - \frac{1}{\pi} \int_0^\infty S(\omega,\ldots,\lambda_d) \cos(\omega_1(x_1 - x_1')\,\mathrm{d}\omega_1 \right) \prod_{k=2}^{d} \frac{1}{L_k} \right| \\
&\quad \leq \sum_{j_2,\ldots,j_d=1}^{\infty} \frac{1}{L_1} \left(B_1 + (2A_1 + B_1)\tilde{L} \right) \prod_{k=2}^{d} \frac{1}{L_k} \\
&\quad \leq \frac{1}{L_1} \left(K_2 + \tilde{L}(2K_1 + K_2) \right) \prod_{k=2}^{d} \frac{1}{L_k} = \frac{D_{1,1}}{L_1}
\end{aligned} \tag{4.54}$$

Then we can continue by bounding (4.49) with

$$\begin{aligned}
&\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - k(\boldsymbol{x},\boldsymbol{x}')\right| \leq \\
&\frac{D_{1,1}}{L_1} + \left| \sum_{j_2,\ldots,j_d}^{\infty} \left(\frac{1}{\pi} \int_0^\infty S(\omega_1,\ldots,\lambda_d) \cos(\omega_1(x_1 - x_1')\,\mathrm{d}\omega_1 \right) \prod_{k=2}^{d} \frac{1}{L_k} \phi_k\left(x_k\right) \phi_k\left(x_k'\right) \right. \\
&\qquad \left. - \frac{1}{\pi^d} \int_0^\infty \cdots \int_0^\infty S(\omega_1,\ldots,\omega_d) \prod_{k=1}^{d} \cos(\omega_k(x_k - x_k'))\,\mathrm{d}\omega_1 \ldots\,\mathrm{d}\omega_d \right|
\end{aligned} \tag{4.55}$$

We wish to repeat these steps process of using (4.47), the triangle inequality and bounding by $\frac{D_{1,k}}{L_k}$ for $k = 2,\ldots,d$. The steps are not entirely analogous to the case where $k = 1$, but the underlying ideas are the same. We have included $k = 2$ in appendix A.2.4 and the remaining can be done analogously to obtain

$$\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - k(\boldsymbol{x},\boldsymbol{x}')\right| \leq \sum_{k}^{d} \frac{D_{1,k}}{L_k} \leq \frac{dD_1}{L} \tag{4.56}$$

where $D_1 = \max_k(D_{1,k})$ and $L = \min_k(L_k)$. This concludes the proof of 4.1.3.

Showing that $\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - \tilde{k}_m(\boldsymbol{x},\boldsymbol{x}')\right| \leq \frac{dD_2}{L} + \frac{1}{\pi^d}\int_{\|\boldsymbol{\omega}\|\geq\frac{\pi m}{2L}} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}$.
As before, it is time to investigate what happens for fixed L and varying m. The proof of the following proposition also relies heavily on lemma 4.1.1.

Proposition 4.1.4 *Let the covariance function k satisfy the assumptions from proposition 4.1.3.*

Then there exists a constant D_2, such that for all $\boldsymbol{x},\boldsymbol{x}' \in \left[-\tilde{L},\tilde{L}\right]^d$ we have

$$\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - \tilde{k}_m(\boldsymbol{x},\boldsymbol{x}')\right| \leq \frac{dD_2}{L_{min}} + \frac{1}{\pi^d}\int_{\|\omega\|\geq\frac{\pi\hat{m}}{2L_{max}}} S(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega} \tag{4.57}$$

for $L_{min} = \min_k(L_K)$ *and* $L_{max} = \max k(L_k)$.

Proof: We have

$$\begin{aligned}
&\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - \tilde{k}_m(\boldsymbol{x},\boldsymbol{x}')\right| \\
&= \left|\sum_{j_1,\ldots,j_d=1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right)\prod_{k=1}^{d}\frac{1}{L_k}\phi_k\left(x_k\right)\phi_k\left(x_k'\right) - \sum_{j_1,\ldots,j_d=1}^{\hat{m}} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right)\prod_{k=1}^{d}\frac{1}{L_k}\phi_k\left(x_k\right)\phi_k\left(x_k'\right)\right| \\
&\leq \left|\sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right)\prod_{k=1}^{d}\frac{1}{L_k}\right|.
\end{aligned} \tag{4.58}$$

Consider

$$\left|\sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right)\prod_{k=1}^{d}\frac{1}{L_k} - \left(\frac{2}{\pi}\right)^d\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty}\cdots\int_{\frac{\pi\hat{m}}{2L_d}}^{\infty} S\left(\omega_1,\ldots,\omega_d\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d\right|, \tag{4.59}$$

which we may expand into a telescoping sum in the following manner (*remark*: the following equation (4.60) on the next page):

$$\begin{aligned}
&\left|\sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right)\prod_{k=1}^{d}\frac{1}{L_k}\right. \\
&\qquad - \frac{2}{\pi}\sum_{j_2,\ldots,j_d=\hat{m}+1}^{\infty}\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} S\left(\omega_1,\lambda_{j_2},\ldots,\lambda_{j_d}\right)\,\mathrm{d}\omega_1\prod_{k=2}^{d}\frac{1}{L_k} \\
&\qquad + \frac{2}{\pi}\sum_{j_2,\ldots,j_d=\hat{m}+1}^{\infty}\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} S\left(\omega_1,\lambda_{j_2},\ldots,\lambda_{j_d}\right)\,\mathrm{d}\omega_1\prod_{k=2}^{d}\frac{1}{L_k} - \\
&\qquad\qquad\qquad\qquad \vdots \\
&\qquad - \left(\frac{2}{\pi}\right)^{d-1}\sum_{j_d=\hat{m}+1}^{\infty}\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty}\cdots\int_{\frac{\pi\hat{m}}{2L_{d-1}}}^{\infty} S\left(\omega_1,\ldots,\omega_{d-1},\lambda_{j_d}\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_{d-1}\frac{1}{L_d} \\
&\qquad + \left(\frac{2}{\pi}\right)^{d-1}\sum_{j_d=\hat{m}+1}^{\infty}\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty}\cdots\int_{\frac{\pi\hat{m}}{2L_{d-1}}}^{\infty} S\left(\omega_1,\ldots,\omega_{d-1},\lambda_{j_d}\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_{d-1}\frac{1}{L_d} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left. - \left(\frac{2}{\pi}\right)^{d}\int_{\frac{\pi\hat{m}}{2L_1}}^{\infty}\cdots\int_{\frac{\pi\hat{m}}{2L_d}}^{\infty} S\left(\omega_1,\ldots,\omega_d\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d\right|
\end{aligned}$$

$$
\begin{aligned}
\leq \Bigg| & \sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \prod_{k=1}^{d} \frac{1}{L_k} \\
& - \frac{2}{\pi} \sum_{j_2,\ldots,j_d=\hat{m}+1}^{\infty} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} S\left(\omega_1,\lambda_{j_2},\ldots,\lambda_{j_d}\right) \mathrm{d}\omega_1 \prod_{k=2}^{d} \frac{1}{L_k} \Bigg| + \\
& \qquad\qquad \vdots \\
+ \Bigg| & \left(\frac{2}{\pi}\right)^{d-1} \sum_{j_d=\hat{m}+1}^{\infty} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_{d-1}}}^{\infty} S\left(\omega_1,\ldots,\omega_{d-1},\lambda_{j_d}\right) \mathrm{d}\omega_1 \ldots \mathrm{d}\omega_{d-1} \frac{1}{L_d} \\
& - \left(\frac{2}{\pi}\right)^{d} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_d}}^{\infty} S\left(\omega_1,\ldots,\omega_d\right) \mathrm{d}\omega_1 \ldots \mathrm{d}\omega_d \Bigg|
\end{aligned}
\tag{4.60}
$$

We use the triangle inequality in the last step by pairing all the terms.

We bound each of these terms upwards by using lemma 4.1.1. For instance, using lemma 4.1.1 with $f(\omega_1) = \frac{2}{\pi} S(\omega_1, \lambda_{j_2}, \ldots, \lambda_{j_d})$, $\alpha = 0$ and $\Delta = \frac{\lambda_{j_k}}{j_k} = \frac{\pi}{2L_1}$ we obtain

$$
\left| \int_{\hat{m}\frac{\pi}{2L_1}}^{\infty} \frac{2}{\pi} S(\omega_1, \lambda_{j_2}, \ldots, \lambda_{j_k}) \, \mathrm{d}\omega - \sum_{j_1=\hat{m}+1}^{\infty} \frac{1}{L_1} S(\lambda_{j_1}, \ldots, \lambda_{j_d}) \right| \leq \frac{1}{L_1} \int_0^{\infty} \left| \frac{\partial}{\partial \omega_1} S(\omega_1) \right| \mathrm{d}\omega_1 = \frac{1}{L_1} B_1,
\tag{4.61}
$$

and derive the bound

$$
\begin{aligned}
& \left| \sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \prod_{k=1}^{d} \frac{1}{L_k} - \frac{2}{\pi} \sum_{j_2,\ldots,j_d=\hat{m}+1}^{\infty} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} S\left(\omega_1,\ldots,\lambda_{j_d}\right) \mathrm{d}\omega_1 \prod_{k=2}^{d} \frac{1}{L_k} \right| \\
& \qquad\qquad \leq \frac{1}{L_1} \sum_{j_2,\ldots,j_d=\hat{m}+1}^{\infty} B_1 \prod_{k=2}^{d} \frac{1}{L_k}.
\end{aligned}
\tag{4.62}
$$

We can bound this by a constant $\frac{D_{2,1}}{L_1}$ by using similar arguments as in (4.50).

More generally, we can use the lemma with $\alpha = 0$ and $\Delta = \frac{\pi}{2L_i}$ and

$$
f(\omega_i) = \left(\frac{2}{\pi}\right)^{i} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_{i-1}}}^{\infty} S(\omega_1,\ldots,\omega_i,\lambda_{j_{i+1}},\ldots\lambda_{j_d}) \, \mathrm{d}\omega_1 \ldots \omega_{i-1}
$$

$$
\left| \int_{m\Delta}^{\infty} f(\omega) \, \mathrm{d}\omega - \sum_{j=m+1}^{\infty} f(j\Delta - \alpha\Delta)\Delta \right| \leq K^{(0)}\Delta.
\tag{4.63}
$$

The condition that f and $|f'|$ are bounded integrable follows from the same considerations that lead to equation (4.50) and (4.53). Using the lemma, we obtain

$$
\begin{aligned}
& \Bigg| \left(\frac{2}{\pi}\right)^{i} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_i}}^{\infty} S(\omega_1,\ldots,\omega_i,\lambda_{j_{i+1}},\ldots\lambda_{j_d}) \, \mathrm{d}\omega_1 \ldots \omega_i \\
& \quad - \sum_{j_i=\hat{m}+1}^{\infty} \frac{1}{L_i} \left(\frac{2}{\pi}\right)^{i-1} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_{i-1}}}^{\infty} S(\omega_1,\ldots,\omega_{i-1},\lambda_{j_i},\ldots\lambda_{j_d}) \, \mathrm{d}\omega_1 \ldots \omega_{i-1} \Bigg| \\
& \qquad\qquad \leq \frac{1}{L_i} \tilde{B}_i
\end{aligned}
\tag{4.64}
$$

and derive the bound

$$\left| \left(\frac{2}{\pi}\right)^{i-1} \sum_{j_i,\ldots,j_d=\hat{m}+1}^{\infty} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_{i-1}}}^{\infty} S\left(\omega_1,\ldots,\omega_{i-1},\ldots,\lambda_{j_d}\,\mathrm{d}\omega_1\ldots\omega_{i-1}\right) \prod_{k=i}^{d} \frac{1}{L_k} \right.$$
$$\left. - \left(\frac{2}{\pi}\right)^{i} \sum_{j_{i+1},\ldots,j_d=\hat{m}+1}^{\infty} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_i}}^{\infty} S\left(\omega_1,\ldots,\omega_i,\ldots,\lambda_{j_d}\,\mathrm{d}\omega_1\ldots\omega_i\right) \prod_{k=i+1}^{d} \frac{1}{L_k} \right| \quad (4.65)$$
$$\leq \frac{1}{L_i} \sum_{j_{i+1},\ldots,j_d=\hat{m}+1}^{\infty} \tilde{B}_i \prod_{k=i+1}^{d} \frac{1}{L_k}.$$

As before, this can be bounded by some constant $\frac{D_{2,i}}{L_i}$.

Now

$$\left| \sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \prod_{k=1}^{d} \frac{1}{L_k} - \left(\frac{2}{\pi}\right)^{d} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_d}}^{\infty} S\left(\omega_1,\ldots,\omega_d\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d \right| \leq \sum_{i=1}^{d} \frac{D_{2,i}}{L_i}, \quad (4.66)$$

and if we use the triangle inequality and set $D_2 = \max_k(D_{2,i})$ and $L_{min} = \min_k(L_k)$ we obtain

$$\left| \sum_{j_1,\ldots,j_d=\hat{m}+1}^{\infty} S\left(\lambda_{j_1},\ldots,\lambda_{j_d}\right) \prod_{k=1}^{d} \frac{1}{L_k} \right| \leq \frac{dD_2}{L} + \left(\frac{2}{\pi}\right)^{d} \int_{\frac{\pi\hat{m}}{2L_1}}^{\infty} \cdots \int_{\frac{\pi\hat{m}}{2L_d}}^{\infty} S\left(\omega_1,\ldots,\omega_d\right)\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d. \quad (4.67)$$

Finally, we realize that the integral corresponds to integrating over the area outside the hypercube $\left[-\frac{\pi\hat{m}}{2L_1},\frac{\pi\hat{m}}{2L_1}\right] \times \cdots \times \left[-\frac{\pi\hat{m}}{2L_d},\frac{\pi\hat{m}}{2L_d}\right]$. We can bound this by integrating over the larger area outside the inscribed hypersphere $\left\{\boldsymbol{\omega} \,|\, \|\boldsymbol{\omega}\| < \frac{\pi\hat{m}}{2L_{max}}\right\}$, $L_{max} = \max_k(L_k)$. This finishes the proof of proposition 4.1.4. Now, there is only one step left.

Proving theorem 2.
With proposition 4.1.3 and proposition 4.1.4 in hand, we can proof 4.1.2 analogously to the one-dimensional case (beginning at equation (4.38)).

4.1.3 Convergence between posterior mean and variance

Finally, we will show that the convergence also holds for the posterior distributions, i.e. that given 4.1.1 or 4.1.2 then the corresponding posterior mean and covariance also converges uniformly for $L, m \to \infty$. This is shown in theorem 2.2 from Särkkä and Piché (2014). In this section we wish to give a more elaborate proof of this result.

Theorem 4.1.3 *If it holds that the prior $\tilde{m}$-rank approximation converges uniformly to the true covariance function, that is*

$$\lim_{L_1,\ldots,L_d\to\infty} \left[\lim_{m\to\infty} \tilde{k}_m(\boldsymbol{x},\boldsymbol{x}') \right] = k(\boldsymbol{x},\boldsymbol{x}'), \quad (4.68)$$

then it follows that the posterior mean and covariance functions converge uniformly to the exact solutions when $m, L_1,\ldots,L_d \to \infty$ that is

$$\lim_{L_1,\ldots,L_d\to\infty} \left[\lim_{m\to\infty} \mathbb{E}[f_m(\boldsymbol{x}_*)|\,\boldsymbol{y},\boldsymbol{x}_*] \right] = \mathbb{E}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] \quad (4.69)$$

and

$$\lim_{L_1,\ldots,L_d\to\infty} \left[\lim_{m\to\infty} \mathbb{V}[f_m(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] \right] = \mathbb{V}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)]. \quad (4.70)$$

Proof.
Recall that $\mathbb{E}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] = \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{y}$ and $\mathbb{E}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] = \boldsymbol{k}_{m*}^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}$ where $[\boldsymbol{k}_{m*}]_{i,j} = \tilde{k}_m(x_i^*, x_j)$ (i.e. the approximate prior covariance between the new data and the training data) and $[\mathcal{K}_m]_{i,j} = \tilde{k}_m(x_i^*, x_j^*)$ (i.e. the approximate prior covariance of the new observations).

We look at

$$
\begin{aligned}
|\mathbb{E}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] - \mathbb{E}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)]| &= \left|\boldsymbol{k}_{m*}^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y} - \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\leq \left|\boldsymbol{k}_{m*}^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y} - \boldsymbol{k}_*^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\qquad + \left|\boldsymbol{k}_*^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y} - \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\leq \left|(\boldsymbol{k}_{m*}^\top - \boldsymbol{k}_*^\top)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\qquad + \left|\boldsymbol{k}_*^\top\left[(\mathcal{K}_m+\sigma_n^2 I)^{-1} - (\mathcal{K}+\sigma_n^2 I)^{-1}\right]\boldsymbol{y}\right| \\
&= \left|(\boldsymbol{k}_{m*}^\top - \boldsymbol{k}_*^\top)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\qquad + \left|\boldsymbol{k}_*^\top\left[(\mathcal{K}_m+\sigma_n^2 I)^{-1}(\mathcal{K}+\sigma_n^2 I)(\mathcal{K}+\sigma_n^2 I)^{-1}\right.\right. \\
&\qquad \left.\left. - (\mathcal{K}+\sigma_n^2 I)^{-1}(\mathcal{K}_m+\sigma_n^2 I)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\right]\boldsymbol{y}\right| \\
&= \left|(\boldsymbol{k}_{m*}^\top - \boldsymbol{k}_*^\top)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\qquad + \left|\boldsymbol{k}_*^\top\left[(\mathcal{K}_m+\sigma_n^2 I)^{-1}\left((\mathcal{K}+\sigma_n^2 I) - (\mathcal{K}_m+\sigma_n^2 I)\right)(\mathcal{K}+\sigma_n^2 I)^{-1}\right]\boldsymbol{y}\right| \\
&= \left|(\boldsymbol{k}_{m*}^\top - \boldsymbol{k}_*^\top)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{y}\right| \\
&\qquad + \left|\boldsymbol{k}_*^\top\left[(\mathcal{K}_m+\sigma_n^2 I)^{-1}(\mathcal{K}-\mathcal{K}_m)(\mathcal{K}+\sigma_n^2 I)^{-1}\right]\boldsymbol{y}\right|
\end{aligned}
\tag{4.71}
$$

Letting both m and $L_1, \ldots, L_d$ go towards infinity both terms will be zero, which means that the Hilbert space aproximated posterior mean converges to the true posterior mean when $L_1, \ldots, L_d, m \to \infty$.
The argument for the posterior variance is analogous. We have

$$
\begin{aligned}
|\mathbb{V}[f_m(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)] - \mathbb{V}[f(\boldsymbol{x}_*|\,\boldsymbol{y},\boldsymbol{x}_*)]| &= \left|k_m(x_*,x_*) - \boldsymbol{k}_{m*}^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*} - \left(k(x_*,x_*) - \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{k}_*\right)\right| \\
&= \big|k_m(x_*,x_*) - k(x_*,x_*) \\
&\quad - \boldsymbol{k}_{m*}^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*} + \boldsymbol{k}_*^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*} \\
&\quad - \boldsymbol{k}_*^\top(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*} + \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{k}_* \\
&\quad - \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*} + \boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*}\big| \\
&\leq |k_m(x_*,x_*) - k(x_*,x_*)| \\
&\quad + \left|(\boldsymbol{k}_*^\top - \boldsymbol{k}_{m*}^\top)(\mathcal{K}_m+\sigma_n^2 I)^{-1}\boldsymbol{k}_{m*}\right| \\
&\quad + \left|\boldsymbol{k}_*^\top(\mathcal{K}+\sigma_n^2 I)^{-1}(\boldsymbol{k}_* - \boldsymbol{k}_{m*})\right| \\
&\quad + \left|\boldsymbol{k}_*^\top\left((\mathcal{K}_m+\sigma_n^2 I)^{-1}(\mathcal{K}-\mathcal{K}_m)(\mathcal{K}+\sigma_n^2 I)^{-1}\right)\boldsymbol{k}_{m*}\right|
\end{aligned}
\tag{4.72}
$$

Letting m and $L_1, \ldots, L_d$ go towards infinity both terms will be zero, which means that the Hilbert space approximated posterior variance converges to the true posterior variance when $m, L_1, \ldots, L_d \to \infty$.

4.2 Average-case Learning Curves of the HS approximation

The more times we have to solve a specific problem, the more we learn and the better we become at solving that problem. In the same way, one would hope that the more data we have available, the better our machine-learning models will be at predicting the true underlying functions. A *learning curve* illustrates how models learn when presented with more information, and *average-case learning curves* is one way of quantifying this.

Formally, the average-case learning curve illustrates how the predictor's, $\bar{f}_{\mathcal{D}}$, accuracy improves as the number of data points, n, increases. In that way, we can understand the average-case learning curve as the average generalisation error of a training set of size n, being a function of n.

The case where the underlying process is a Gaussian process has been widely studied and quantified, as described in Rasmussen and Williams (2006), chapter 7.3. In this section, we are interested in applying some of these results to the HS approximation and seeing how much is 'lost' when using the approximation as opposed to the full model. We will derive a lower bound of the average-case learning curves for Gaussian processes in general and compare it to the HS model. We are interested in investigating whether the approximation is more sensitive to different data sizes than a full Gaussian process.

In the following, we will start by going through the theoretical details on the subject of learning curves in a standard Gaussian process setting following Rasmussen and Williams (ibid.). Secondly, we wish to compare the results from chapter 7.3 in Rasmussen and Williams (ibid.) with simulations where we have replaced the standard Gaussian process predictor with our HS approximation predictor.

4.2.1 Average-case Learning curves for standard Gaussian processes

Assume that we have observed a set of noisy observations $\mathcal{D} = (\boldsymbol{y}, X)$ of an underlying target function $f \sim \mathcal{GP}(0, k_0)$ with noise variance σ_0^2, such that $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ where $\epsilon \sim \mathcal{N}(0, \sigma_0^2)$. We use $\mathcal{D}$ to compute an estimator of the function, $\bar{f}_{\mathcal{D}}$. We assume the data has input density $p(\boldsymbol{x}_*)$. The generalisation error is given by

$$E_{\mathcal{D}}^g(f) = \int \mathcal{L}\left(f(\boldsymbol{x}_*), \bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right) p(\boldsymbol{x}_*)\, \mathrm{d}\boldsymbol{x}_*, \tag{4.73}$$

where $\mathcal{L}$ is some loss function. In other words, it is the expected prediction loss when we predict f based on $\mathcal{D}$. Clearly, $E_{\mathcal{D}}^g(f)$ depends on both f and the observed X. Since we assume that f is generated from a Gaussian process prior, we can marginalise out the target functions. This gives us

$$E^g(X) = \int E_{\mathcal{D}}^g(f) p(f)\, \mathrm{d}f. \tag{4.74}$$

Lastly, we can average over the observed training data to obtain the *learning curve*.

$$E^g(n) = \int E^g(X) p(\boldsymbol{x}_1) p(\boldsymbol{x}_2) \dots p(\boldsymbol{x}_n)\, \mathrm{d}\boldsymbol{x}_1\, \mathrm{d}\boldsymbol{x}_2 \dots\, \mathrm{d}\boldsymbol{x}_n. \tag{4.75}$$

In general, it is rather difficult to obtain $E^g(n)$ since we need to average $E^g(X)$ over the (possibly unknown) input density, which is not always analytically tractable. However, for Gaussian process priors and specific loss functions, it is possible.

We want to analyse the above equations with $\mathcal{L}$ as the squared error. That is

$$\mathcal{L}\left(f(\boldsymbol{x}_*), \bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right) = \left(f(\boldsymbol{x}_*) - \bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)^2$$

at a test location $\boldsymbol{x}_*$.

We assume a particular test location, $\boldsymbol{x}_*$. We use the posterior mean as the predictor $\bar{f}_{\mathcal{D}}$ with covariance function k_1 and σ_1^2. Averaging over f we obtain

$$\begin{aligned}
&\mathbb{E}_f\left[\left(f(\boldsymbol{x}_*)-\bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)^2\right]\\
=&\mathbb{E}_f\left[\left(f(\boldsymbol{x}_*)-\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}y\right)^2\right]\\
=&\mathbb{E}_f\left[f^2(\boldsymbol{x}_*)-2f(\boldsymbol{x}_*)\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}y+\left(\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}y\right)^2\right]\\
=&\mathbb{E}_f\left[f^2(\boldsymbol{x}_*)\right]-2\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}\mathbb{E}_f\left[f(\boldsymbol{x}_*)y\right]+\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}\mathbb{E}_f\left[yy^\top\right]K_{1,y}^{-1}\mathbf{k}_1(\boldsymbol{x}_*)\\
=&k_0(\boldsymbol{x}_*,\boldsymbol{x}_*)-2\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}\mathbf{k}_0(\boldsymbol{x}_*)+\mathbf{k}_1(\boldsymbol{x}_*)^\top K_{1,y}^{-1}K_{0,y}K_{1,y}^{-1}\mathbf{k}_1(\boldsymbol{x}_*),
\end{aligned} \tag{4.76}$$

where $K_{i,y}=K_{i,f}+\sigma_i^2$. If we assume that $k_0=k_1$, that is, the predictor is correctly specified with respect to covariance function and hyperparameters, then the above equation reduces to

$$\mathbb{E}_f\left[\left(f(\boldsymbol{x}_*)-\bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)^2\right]=k_0(\boldsymbol{x}_*,\boldsymbol{x}_*)-\mathbf{k}_0(\boldsymbol{x}_*)^\top K_{0,y}^{-1}\mathbf{k}_0(\boldsymbol{x}_*). \tag{4.77}$$

Now we wish to obtain the generalization error $E^g(X)$ by averaging over the input density $p(\boldsymbol{x}_*)$ such that

$$\begin{aligned}
E^g(X)&=\int\mathbb{E}_f\left[\left(f(\boldsymbol{x}_*)-\bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)^2\right]p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\\
&=\int k_0(\boldsymbol{x}_*,\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*-\int\mathbf{k}_0(\boldsymbol{x}_*)^\top K_{0,y}^{-1}\mathbf{k}_0(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\\
&=\int k_0(\boldsymbol{x}_*,\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*-\mathrm{Tr}\left(K_{0,y}^{-1}\int\mathbf{k}_0(\boldsymbol{x}_*)\mathbf{k}_0(\boldsymbol{x}_*)^\top p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\right).
\end{aligned} \tag{4.78}$$

Remark that we in equation (4.77) and (4.78) reversed the order of integration, compared to (4.73) and (4.74). This can be done because

$$\begin{aligned}
E^g(X)&=\int E_{\mathcal{D}}^g(f)p(f)\,\mathrm{d}f\\
&=\int\left(\int\mathcal{L}\left(f(\boldsymbol{x}_*),\bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\right)p(f)\,\mathrm{d}f\\
&=\int\left(\int\mathcal{L}\left(f(\boldsymbol{x}_*),\bar{f}_{\mathcal{D}}(\boldsymbol{x}_*)\right)p(f)\,\mathrm{d}f\right)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*.
\end{aligned} \tag{4.79}$$

Now, we wish to analyse the learning curve formula (4.75) in the same manner as above. If we sustain the assumption that $k_0=k_1$, we can simplify the form of $E^g(X)$ considerably by making an eigenfunction expansion of k_0. As we have seen in section 3.2, we can use the covariance function to define an operator. Only this time, we have to take the input density into account and define the linear operator

$$T_kf=\int_\Omega^\infty k(\boldsymbol{x},\boldsymbol{x}')f(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

Since k is a covariance function, it is positive semidefinite, and one can show that eigenfunctions exist for this operator and are orthonormal[1], i.e. we have λ_i's and ϕ_i's such

[1]This follows from *compactness* of T_k, see Kreyszig (1991).

that

$$T_k\phi_j = \lambda_j\phi_j \quad \text{and} \quad \int_\Omega \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) = \delta_{i,j},$$

where $\delta_{i,j}$ is the Kronecker delta.

Furthermore, we can express k in terms of the eigenfunctions and eigenvalues of T_k. We have

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_j \lambda_j\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}').$$

These results are known as Mercer's theorem (see section 4.3 in Rasmussen and Williams (2006)).

It is important to note that although we have used the same notation, ϕ and λ are eigenfunctions and eigenvalues of the covariance operator and not the negative Laplace operator. Thus they are not the same eigenvalues and eigenfunctions that we used in the HS approximation. Because of the input density, it is not immediately possible to derive a bound for approximated $\tilde{k}$ specifically.

We may write $k_0(\boldsymbol{x}, \boldsymbol{x}') = \sum_i \lambda_i\phi_i(\boldsymbol{x})\phi_i(\boldsymbol{x}')$ where $\int k_0(\boldsymbol{x}, \boldsymbol{x}')\phi_i(\boldsymbol{x})p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \lambda_i\phi_i(\boldsymbol{x}')$. We can use this to obtain

$$\begin{aligned}
&\int k_0(\boldsymbol{x}_*, \boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* - \operatorname{Tr}\left(K_{0,y}^{-1}\int \mathbf{k}_0(\boldsymbol{x}_*)^\top \mathbf{k}_0(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\right) \\
&= \int \sum_i \lambda_i\phi_i(\boldsymbol{x}_*)\phi_i(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* - \operatorname{Tr}\left((\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\int Ap(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\right),
\end{aligned} \tag{4.80}$$

where Λ is a diagonal matrix with $\Lambda_{ii} = \lambda_i$. Here

$$A = \begin{pmatrix} \sum_i \lambda_i\phi_i(\boldsymbol{x}_1)\phi_i(\boldsymbol{x}_*) \\ \dots \\ \sum_i \lambda_i\phi_i(\boldsymbol{x}_n)\phi_i(\boldsymbol{x}_*) \end{pmatrix}\left(\sum_i \lambda_i\phi_i(\boldsymbol{x}_1)\phi_i(\boldsymbol{x}_*), \dots, \sum_i \lambda_i\phi_i(\boldsymbol{x}_n)\phi_i(\boldsymbol{x}_*)\right),$$

such that $A_{k,l} = \sum_i\sum_j \lambda_i\lambda_j\phi_i(x_k)\phi_i(\boldsymbol{x}_*)\phi_i(x_l)\phi_j(\boldsymbol{x}_*)$. We start by analysing the first term in (4.80) and by using the orthonormality of ϕ wrt. the inner product we obtain

$$\int \sum_i \lambda_i\phi_i(\boldsymbol{x}_*)\phi_i(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* = \sum_i \lambda_i\int \phi_i(\boldsymbol{x}_*)\phi_i(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* = \sum_i \lambda_i = \operatorname{Tr}(\Lambda). \tag{4.81}$$

Now, we deal with the integral in the second term of (4.80). We start component wise for each entry $A_{k,l}$ and obtain

$$\begin{aligned}
\int A_{k,l}p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* &= \int \sum_i\sum_j \lambda_i\lambda_j\phi_i(x_k)\phi_i(\boldsymbol{x}_*)\phi_j(x_l)\phi_j(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* \\
&= \sum_i\sum_j \lambda_i\lambda_j\phi_i(x_k)\phi_j(x_l)\int \phi_i(\boldsymbol{x}_*)\phi_j(\boldsymbol{x}_*)p(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_* \\
&= \sum_i \lambda_i^2\phi_i(x_k)\phi_i(x_l) \\
&= \Phi_k^\top\Lambda^2\Phi_l.
\end{aligned} \tag{4.82}$$

If we look at the whole second term of 4.80, we then obtain

$$\operatorname{Tr}\left((\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\int Ap(\boldsymbol{x}_*)\,\mathrm{d}\boldsymbol{x}_*\right) = \operatorname{Tr}\left((\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda^2\Phi\right). \tag{4.83}$$

We wish to simplify this equation by using the inverse matrix theorem from (A.9) in Rasmussen and Williams (2006), and therefore, we do this simple reformulation of (4.83).

$$= \text{Tr}\left(\Lambda\Phi(\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda\right). \tag{4.84}$$

Now we can use the inverse matrix theorem directly on 4.78 by putting $Z = \Lambda^{-1}, U = V = \Phi$ and $W = I\sigma^{-2}$ and we get

$$\begin{aligned} E^g(X) &= \text{Tr}\left(\Lambda\right) - \text{Tr}\left(\Lambda\Phi(\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda\right) \\ &= \text{Tr}\left(\Lambda - \Lambda\Phi(\sigma_n^2 I + \Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda\right) \\ &= \text{Tr}\left(\Lambda^{-1} + \sigma_n^{-2}\Phi\Phi^\top\right)^{-1}. \end{aligned} \tag{4.85}$$

This reformulation of $E^g(n)$ makes the computation remarkably more simple. If we want to introduce one more simplification, then a naive implementation of $E^g(n)$ would be to simply input the expectation of $\Phi\Phi^\top$ in (4.85) such that

$$E^g(n) \approx \text{Tr}\left(\Lambda^{-1} + \sigma_n^{-2} nI\right)^{-1} = \sum_i \frac{\sigma_n^2\lambda_i}{\sigma_n^2 + n\lambda_i}. \tag{4.86}$$

We have that $\mathbb{E}_X\left[\Phi\Phi^\top\right] = nI$ since

$$\begin{aligned} \mathbb{E}_X\left[(\Phi\Phi^\top)_{k,l}\right] &= \int\cdots\int (\Phi\Phi^\top)_{k,l} p(\boldsymbol{x}_1)\ldots p(\boldsymbol{x}_n)\,\mathrm{d}\boldsymbol{x}_1\ldots\,\mathrm{d}\boldsymbol{x}_n \\ &= \int\cdots\int \sum_{i=1}^n \phi_k(x_i)\phi_l(x_i) p(\boldsymbol{x}_1)\ldots p(\boldsymbol{x}_n)\,\mathrm{d}\boldsymbol{x}_1\ldots\,\mathrm{d}\boldsymbol{x}_n \\ &= \sum_{i=1}^n \int\cdots\int \phi_k(x_i)\phi_l(x_i) p(\boldsymbol{x}_1)\,\mathrm{d}\boldsymbol{x}_1\ldots p(\boldsymbol{x}_n)\,\mathrm{d}\boldsymbol{x}_n \\ &= n\delta_{k,l}, \end{aligned} \tag{4.87}$$

due to the orthonormality of ϕ.
It has been showed by Opper and Vivarelli in Opper and Vivarelli (1999) that the native implementation of $E^g(n)$ in (4.86) is in fact a lower bound such that

$$E^g(n) \geq \sum_i \frac{\sigma_n^2\lambda_i}{\sigma_n^2 + n\lambda_i}. \tag{4.88}$$

Solin and Särkkä (2020) mention in the discussion section of their article that the Opper-Vivarelli bound can be approximated by

$$\sum_i \frac{\sigma_n^2\lambda_i}{\sigma_n^2 + n\lambda_i} \approx \sum_i \frac{\sigma_n^2 S\left(\sqrt{\tilde{\lambda}_i}\right)}{\sigma_n^2 + nS\left(\sqrt{\tilde{\lambda}_i}\right)}, \tag{4.89}$$

where $\tilde{\lambda}_i$ are the Dirichlét eigenvalues of the Laplacian. However, we found this difficult to confirm since the eigenfunctions of the Laplacian are not generally orthogonal with respect to the input density $p(\boldsymbol{x})$. In section 2.4 in the article from Solin and Särkkä (ibid.), it is described how the Hilbert space approximation can be considered in terms of an inner product defined by an input density, leading to a different set of eigenfunctions that solve the Dirichlet eigenvalue problem of the Laplacian with respect to this other inner product. In that case, showing (4.89) is analogous to the previous section. We have not pursued this any further.

4.2.2 Simulations on the average-case learning curve

In this section, we will simulate the generalisation error of our Hilbert space Gaussian process with respect to different numbers of modes and hyperparameters and compare these with a standard Gaussian process and the lower bound from 4.88. We also plot the approximated lower bound proposed by Solin and Särkkä (ibid.).

Our procedure is inspired by section 7 in Opper and Vivarelli (1999). The training and test data follow a standard Gaussian distribution. We use a Monte Carlo estimator to compute the average learning curves. For each $n = 5, \ldots, N$, we sample 3-tuples $\left(\boldsymbol{x}_{train}^{(i)}, f^{(i)}, \boldsymbol{x}_{*}^{(i)}\right)$ for $i = 1, \ldots, D$, and compute

$$\hat{\mathcal{E}}_{\mathcal{D}_i}^{g}(f_i) = \frac{1}{n_{test}} \sum_{k=1}^{n_{test}} \mathcal{L}\left(f^{(i)}\left(x_{*k}^{(i)}\right), \bar{f}_{\mathcal{D}_i}\left(x_{*k}^{(i)}\right)\right)$$

as an estimator of the generalisation error and

$$\hat{\mathcal{E}}^{g}(n) = \frac{1}{D} \sum_{i=1}^{D} \hat{\mathcal{E}}_{\mathcal{D}_i}^{g}(f_i)$$

as the final estimator of the learning curve.

Data: m and L, N, n_{test} and D.
for $i \in \{1, 2, \ldots, D\}$ **do**
  $\tilde{N} = N + n_{test}$ Sample $X_i = \{x_{i1}, x_{i2}, \ldots, x_{i\tilde{N}}\} \sim \mathcal{N}(0, I)$.
  Sample f_i from a GP prior with squared exponential kernel function with $\kappa = 1$ and $\ell = l$.
  Let $\boldsymbol{f}_i = \{f_i(x_{i1}), \ldots, f_{i\tilde{N}}\}$
  Compute $y_i = \boldsymbol{f}_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 0.1 \cdot I)$.
  Define test data
    $\boldsymbol{x}_{*}^{(i)} = \{x_{i(N+1)}, \ldots, x_{i\tilde{N}}\}$.
    $\boldsymbol{f}_{*}^{(i)} = \{\boldsymbol{f}_{i(N+1)}, \ldots, \boldsymbol{f}_{i[\tilde{N}]}\}$.
  for $n \in \{5, \ldots, N\}$ **do**
    Subset the data to obtain $\mathcal{D}_i^{(n)}$
      $\boldsymbol{x}_{train}^{(i)} = \{x_{i1}, \ldots, x_{in}\}$
      $\boldsymbol{y}_{train}^{(i)} = \{y_{i1}, \ldots, y_{in}\}$
    Fit HS-model with m basis functions and L on $\boldsymbol{x}_{train}^{(i)}, \boldsymbol{y}_{train}^{(i)}$ using the squared exponential kernel with the true parameters σ, ℓ.
    Compute $\bar{\boldsymbol{f}}_{\mathcal{D}_i^{(n)}}$ on $\boldsymbol{x}_{*}^{(i)}$.
    Compute

$$\hat{\mathcal{E}}_{\mathcal{D}_i^{(n)}}^{g}(f_i) = \frac{1}{n_{test}} \sum_{k=1}^{n_{test}} \left(\boldsymbol{f}_{*k}^{(i)} - \bar{\boldsymbol{f}}_{\mathcal{D}_i^{(n)}k}\right)^2$$

  end
end

Algorithm 1: Pseudocode for simulation of the average case learning curve for the HS approximation given m basis functions and domain parameter L.

The procedure for sampling the generalisation error is described in algorithm 1. We use $N = n_{test} = 1000$, $D = 100$ and $l \in \{0.05, 0.1, 1, 5\}$ and $m \in \{12, 32, 64, 128, 256\}$,

producing D estimates of the generalization error for each (n, m, l). Finally, we take the mean over the datasets to estimate the learning curve for each (n, m, l). We run procedure $R = 20$ times and report the mean and standard deviation over the runs.

The results from the simulation are plotted in figure (4.2). First of all, we see that the

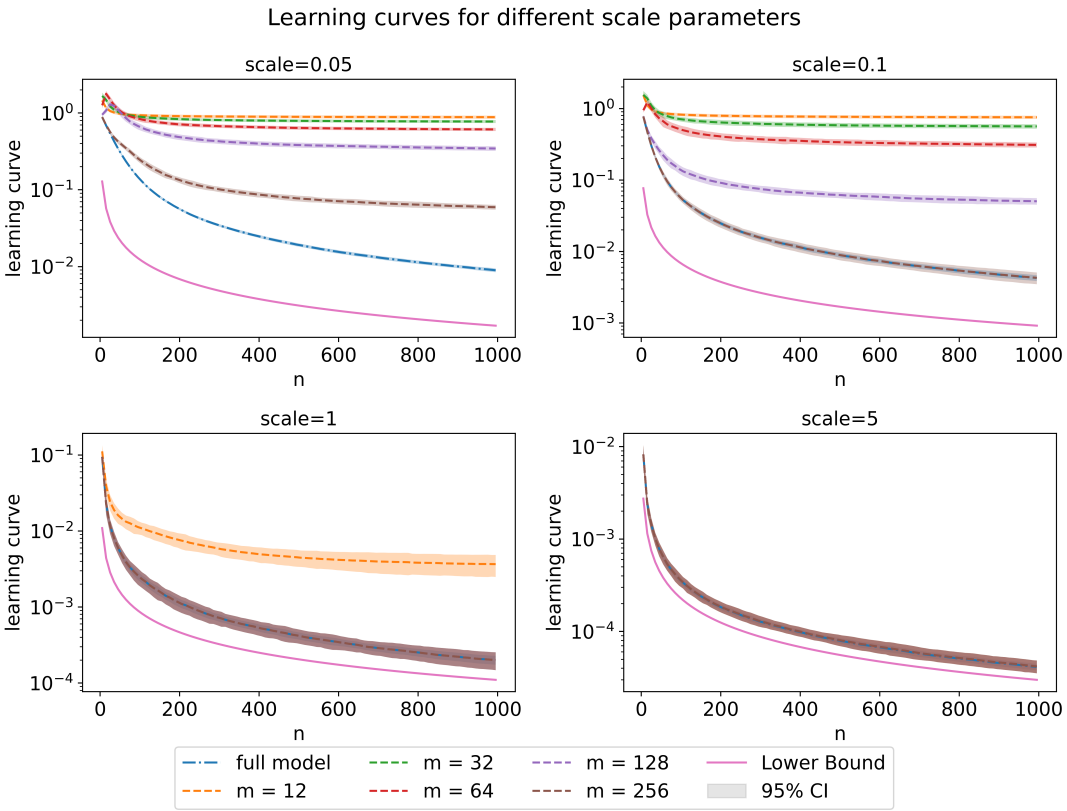


Figure 4.2: Result of 20 simulations of the learning curves. The lines correspond to the mean over the 20 runs. We see that for small length scales, a high number of basis functions is required to match the full model, and for large length scales, all models perform as well as the full model. The lower bound is the Opper-Vivarelli bound given in (4.88)

learning curves of the HS approximations have the same asymptotic behaviour as the full model. Across all the plots, it is clear that the HS model loses precision when we only use a few basis elements, which is exactly as expected based on the discussions and proofs in the previous sections regarding convergence. We also see how the number of basis functions is connected to the length scale. When l is low, the samples f_i fluctuate more quickly. The HS models with fewer basis elements perform badly, as the basis elements used have too low frequencies. However, when we fit on data sampled with a large length scale, we see that the HS model easily competes with the standard GP and that we can choose m to be quite low. This indicates that the HS model is especially beneficial for data with a long length scale, as we don't need to take out a lot of basis elements, and the approximation will be both exact and cheap computationally. The plots also show that for length scales 0.1, 1 and 5, the HS model achieves the same precision as the standard GP if we select a high enough number of basis elements. This is a direct consequence of the results in the section 4.1 on the convergence of the HS-model.

4.3 Summary of findings

In this section, we will contextualize the theoretical findings from chapter 4 in a modelling setting. In this chapter, we have investigated the convergence of the Hilbert space approximation. We have proven that for high enough values of m and L, it is possible to make the approximation arbitrarily close to the true distribution, and this is true both for the prior and posterior distributions. We have also seen that the convergence depends on the order in which to take the limits, as we illustrated in figure 4.1 by plotting the KL-divergence between the approximate and true posterior. In practice, this means that the larger we choose the domain $[-L, L]$, the more basis functions are required to obtain the same precision. Using a higher value of m always leads to better convergence. However, it will also impact the computational complexity of the model, which grows $\mathcal{O}(m^3)$, as discussed in section 3.5.

We have also investigated the generalization of the model through average-case learning curves, which measures how a model learns as a function of the number of training data points. This concept is well documented for Gaussian processes in general, and we derived a lower bound of the average case learning curve of a GP, assuming the latent function generating the data was a GP. We made an empirical comparison to the Hilbert space approximation. First of all, we found that for poor choices of m, the average-case learning curves stalled, but also that we were able to get the same performance as the full model, as long as we had enough basis functions – a direct result of the convergence results. Secondly, the result illustrated how the length scale of the data impacts the requirements on m and L. If the data has a short length scale, we need a high number of basis functions to capture the high frequencies. If the data has a long length scale, fewer basis functions are needed to capture the fluctuations. However, we will need to be careful to choose L large enough that we have a 'buffer' zone around $-L$ and L, where the basis functions – and thus, the variance of the posterior – goes to zero. Thereby, the choice of L limits m from below. Overall, in practical uses of the approximation, we want to choose L as large as possible, to avoid issues at the boundaries, and m as small as possible, to keep computational costs down.

5 Shape-constrained modelling using Hilbert space approximation

Originally, the HS approximation was intended to reduce the computational cost of fitting a Gaussian process. However, the nature and properties of the basis functions make it possible to use the approximation for shape-constrained modelling, which is useful in cases where only a few data points are available and some domain knowledge indicates a certain shape of the data. It could be that we know the data to be monotonic, e.g. if we consider plant growth, or u-shaped or unimodal, which is sometimes assumed in developmental psychology. The hypothesis is that incorporating this information into our model will result in better performance on small datasets than when fitting a pure Gaussian process.

In this section, we develop shape-constrained functions in three steps building on top of each other. First, we construct positive functions. These can be integrated in order to construct monotonic functions. Finally, the monotonicity gives rise to convex functions.

5.1 Positive functions

5.1.1 Model derivation

The first shape constraint we will implement is positiveness. Furthermore, we will also show that we can use this approach to fit probability density functions. In order to do so, we must be able to integrate the functions to find the normalizing constants. Due to the construction of the HS approximation, this normalization constant is easy to compute if we choose the right positive transformation.

In order to impose positivity, we transform some Gaussian process g by a positive function $t : \mathbb{R} \to \mathbb{R}_0^+$. In the following work, we have used $t(x) = x^2$.

Thus

$$h(x) = t(g(x)) = g(x)^2. \tag{5.1}$$

The choice of $t(x) = x^2$ is motivated by the properties of the HS approximation, although it also has some drawbacks that will be discussed later.
When inserting the approximation $g \approx \sum_i^m \alpha_i \phi_i(x)$ for $\alpha_i \sim \mathcal{N}\left(0, S(\sqrt{\lambda_i})\right)$ we obtain

$$h(x) \approx \left(\sum_i \alpha_i \phi_i(x)\right)^2 = \sum_{ij} \alpha_i \alpha_j \phi_i(x) \phi_j(x) = \boldsymbol{\alpha}^\top \Phi^\top \Phi \boldsymbol{\alpha}, \tag{5.2}$$

where $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \end{pmatrix}^\top$.

The assuming joint probability probability density of $\boldsymbol{\alpha}$ and $\boldsymbol{y}$ now becomes

$$p(\boldsymbol{\alpha}, \boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}), \tag{5.3}$$

where $p(\boldsymbol{y}|\boldsymbol{\alpha})$ is the likelihood. The posterior $p(\boldsymbol{\alpha}|\boldsymbol{y})$ is analytically intractable due to the quadratic dependency on $\boldsymbol{\alpha}$.

Another useful property inherent in this approach is that we are able to compute a normalizing constant $Z = \int_{-\infty}^{\infty} h(x)\,\mathrm{d}x$ such that we can use this construction to model

probability density functions by $\frac{1}{Z}h(x) = p(x)$. The derivation is simple, and by utilizing the orthogonality of the eigenfunctions, we get that

$$\begin{aligned} Z \approx \int_{-\infty}^{\infty} \left(\sum_i \alpha_i \phi_i(x) \right)^2 \mathrm{d}x &= \int_{-\infty}^{\infty} \sum_{ij} \alpha_i \alpha_j \phi_i(x) \phi_j(x) \, \mathrm{d}x \\ &= \sum_{ij} \alpha_i \alpha_j \int_{-\infty}^{\infty} \phi_i(x) \phi_j(x) \, \mathrm{d}x \quad = \sum_i \alpha_i^2 \end{aligned} \tag{5.4}$$

Finally, this allows us to define

$$p(x) = \frac{1}{Z} h(x) = \frac{\sum_{ij} \alpha_i \alpha_j \phi_i(x) \phi_j(x)}{\sum_k \alpha_k^2}. \tag{5.5}$$

We also have access to the analytical solution of the cdf. This derivation is described in the next section,

In figure 5.1, we have sampled prior pdf and cdf samples using this approach. The result is smooth pdf samples, which integrates to 1 over the whole domain.

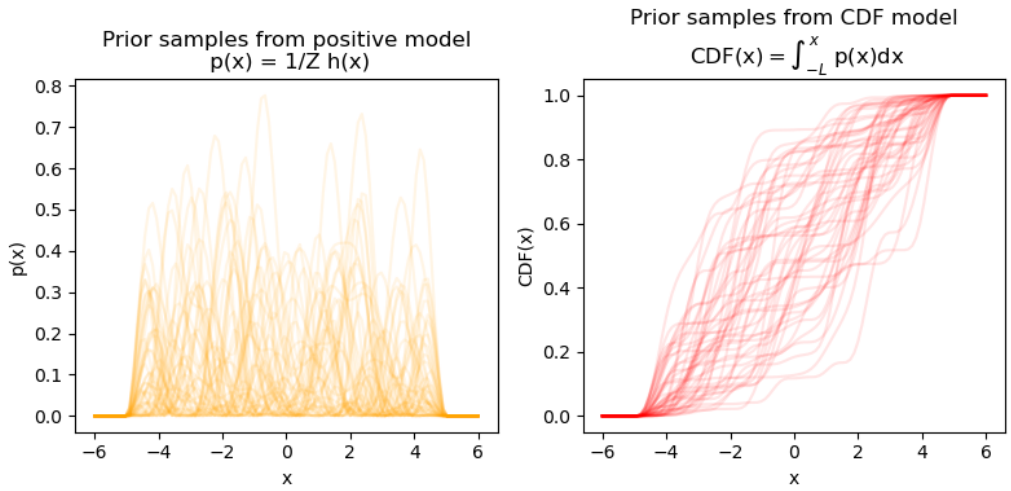


Figure 5.1: Left: Prior samples from the positive function normalized with Z. The number of basis functions is $m = 16$, $L = 5$, and the kernel is squared exponential with $\kappa = 1$ and $\ell = 1$. Right: The corresponding samples from the cdf are calculated using the basis functions given in section 5.2 and normalised with Z.

5.1.2 Choice of positive function *t(x)*

There are several advantages of choosing $t(x) = x^2$. As we have already shown, it simplifies the computation of $\int t(g(x)) \, \mathrm{d}x$, making it analytically tractable. It is also beneficial when calculating the prior mean and variance, as we shall see in the next section. One of the disadvantages of choosing $t(x) = x^2$ is that the parameter space of α will have two solutions that evaluate to the same function h. If

$$h(x) = \sum_{ij} \alpha_i \alpha_j \phi_i(x) \phi_j(x)$$

and we choose $\{\beta_i\}_{i=1}^m$ such that $\beta_i = -\alpha_i$ we also have

$$h(x) = \sum_{ij} \beta_i \beta_j \phi_i(x) \phi_j(x).$$

This can cause problems when using Markov chain Monte Carlo methods for inference, as it can be more tricky to assess convergence if a chain jumps between two modes.

5.1.3 Prior mean and variance for positive functions

Due to the simplicity of the negative Laplacian eigenfunctions, we are able to compute the prior mean and variance analytically. This is useful for understanding model behaviour, the impact of the hyperparameters and the impact on the posterior.

In order to find the mean of $h(x) = g(x)^2$ we assume that $\mathbb{E}[g(x)] = \mu$ and $\mathbb{V}[g(x)] = \eta^2$.

We use that $\mathbb{V}[g(x)] = \mathbb{E}[g(x)^2] - \mathbb{E}[g(x)]^2$ and since the marginal variance of g if η^2, we obtain

$$\mathbb{E}[h(x)] = \mathbb{E}[g(x)^2] = \mathbb{V}[g(x)] + \mathbb{E}[g(x)]^2 = \eta^2 + \mu^2. \tag{5.6}$$

Next we find $\mathbb{V}[h(x)]$. From (5.6) we easily find that

$$\mathbb{E}[h(x)]^2 = (\eta^2 + \mu^2)^2. \tag{5.7}$$

Now we look at $\mathbb{E}[h(x)^2] = \mathbb{E}[g(x)^4]$. Since $g(x)$ is normally distributed, this is the non-central fourth moment and is known to be.

$$\mathbb{E}[g(x)^4] = \mu^4 + 6\mu^2\eta^2 + 3\eta^4. \tag{5.8}$$

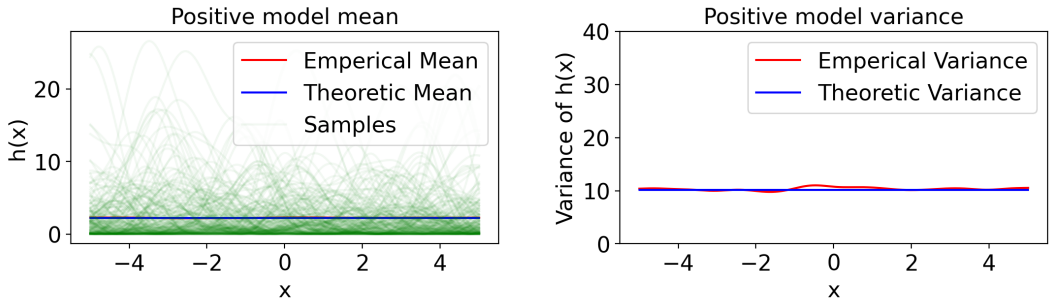
Finally, the variance of $h(x)$ is

$$\begin{aligned}
\mathbb{V}[h(x)] &= \mathbb{E}[h(x)^2] - \mathbb{E}[h(x)]^2 \\
&= \mu^4 + 6\mu^2\eta^2 + 3\eta^4 - (\eta^2 + \mu^2)^2 \\
&= \mu^4 + 6\mu^2\eta^2 + 3\eta^4 - \mu^4 - \eta^4 - 2\mu^2\eta^2 \\
&= 4\mu^2\eta^2 + 2\eta^4.
\end{aligned} \tag{5.9}$$

5.1.4 Simulations of mean and variance

In this section, we simulate the mean and variance of the positive model and compare them to the theoretical results from the previous section.
Method: We sample 10000 functions g_i from a zero-mean Gaussian process evaluated in 1000 uniformly distanced points on the interval [-5,5]. We then transform each sample by $h_i = g_i^2$ and calculate the mean and variance of all 10000 realizations. We use the squared exponential kernel as covariance function with $\ell = 1.1$ and $\kappa = 1.5$. Thus $\mathbb{E}[g(x)] = 0$ and $\mathbb{V}[g(x)] = 1.5^2$.

When we look at figure 5.2, the theoretical and empirical mean seems to be spot on after 10000 samples. The empirical variance appears to have a little bit of uncertainty around the theoretical variance. In conclusion, the plots confirm our analytical derivations.



(a) Visualisation of the theoretical mean with the empirical mean computed on 10000 samples.

(b) Visualisation of the theoretical variance with the empirical variance computed on 10000 samples.

Figure 5.2

5.2 Monotonic functions

In this section, we will build upon the positive model to construct monotonic functions. This method was first described in (Andersen et al. 2018). A defining characteristic of a monotonic function is that it has a strictly positive or negative derivative. Considering the case where f is increasing monotonically, we define

$$f(x) = f_0 + \int_a^x t(g(s))\,\mathrm{d}s \tag{5.10}$$

with $f_0 = f(a)$ for some $a \in \mathbb{R}$ and t and g being defined as in 5.1. This can easily be changed to a monotonically decreasing function by changing the sign of f.

Using the HS approximation and $a = -L$, we obtain

$$\begin{aligned} f(x) &\approx f_0 + \int_{-L}^x \left(\sum_i \alpha_i \phi_i(s)\right)^2 \mathrm{d}s \\ &= f_0 + \sum_{ij} \alpha_i \alpha_j \int_{-L}^x \phi_i(s)\phi_j(s)\,\mathrm{d}s \\ &= f_0 + \boldsymbol{\alpha}^\top \boldsymbol{\psi} \boldsymbol{\alpha}, \end{aligned} \tag{5.11}$$

where

$$[\boldsymbol{\psi}]_{ij} = \int_{-L}^x \phi_i(s)\phi_j(s)\,\mathrm{d}s = \begin{cases} \frac{x+L}{2L} - \frac{\sin\left(\gamma_{ii}^+(x+L)\right)}{2\gamma_{ii}^+ L}, & i = j \\ \frac{\sin\left(\gamma_{ij}^-(x+L)\right)}{2\gamma_{ij}^- L} - \frac{\sin\left(\gamma_{ij}^+(x+L)\right)}{2\gamma_{ij}^+ L}, & i \neq j \end{cases} \tag{5.12}$$

when we let $\gamma_{ij}^{\pm} = \sqrt{\lambda_i} \pm \sqrt{\lambda_j}$. The derivation of this can be found in Appendix A.3.

The joint probability density now becomes

$$p(\boldsymbol{\alpha}, \boldsymbol{y}, f_0) = p(\boldsymbol{y}|\boldsymbol{\alpha}, f_0)p(\boldsymbol{\alpha})p(f_0). \tag{5.13}$$

The posterior, which we want to infer is now $p(\boldsymbol{\alpha}, f_0|\boldsymbol{y})$. As in the positive model from section 5.1, the posterior distribution is analytically intractable due to the quadratic dependency on $\boldsymbol{\alpha}$ and due to a potential non-Gaussian likelihood, $p(\boldsymbol{y}|\boldsymbol{\alpha})$.

Remark: When applying the monotonic function, $f(x)$, for modelling purposes, we need to pay attention to the parameter f_0. We will do so by imposing a prior on f_0. We

will discuss the aspects of dealing with f_0 as a hyperparameter in section 5.3.3. For the following sections regarding the monotonic model, we will consider f_0 as a given constant.

5.2.1 Prior mean and variance

Once again, we are able to compute the prior mean and variance of the model, which means we can reap the benefits mentioned in section A.1.

The results in this section are derived in Andersen et al. (2018). It would also be possible to derive these with similar calculations as in Appendix A.4.

Assume that $g(x)$ has mean μ and a stationary covariance function and that f_0 is constant. Furthermore, we assume that the marginal variance is given by $\mathbb{V}[g(x)] = \eta^2$. Under these assumptions, the mean of $f(x)$ is

$$\mathbb{E}[f(x)] = f_0 + \eta^2(x - a) \tag{5.14}$$

i.e., the mean is linear in x with a slope that depends on the variance of the Gaussian process.

The marginal variance is

$$\mathbb{V}[f(x)] = 2\int_a^x \int_a^x k(s, s')^2 \,\mathrm{d}s \,\mathrm{d}s' \tag{5.15}$$

Using the squared exponential kernel $k(s, s') = \eta^2 \exp\left(-\frac{(s-s')^2}{2\ell^2}\right)$, the marginal variance is

$$\mathbb{V}[f(x)] = 2\eta^4\ell^2 \left[\exp\left(-\left(\frac{x-a}{\ell}\right)^2\right) + \sqrt{\pi}\left(\frac{x-a}{\ell}\right)\operatorname{erf}\left(\frac{x-a}{\ell}\right) - 1\right] \tag{5.16}$$

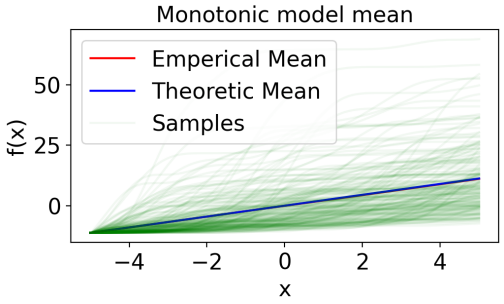
For $x - a \gg \ell$, the variance becomes linear with slope $2\eta^4\ell^2$.

5.2.2 Simulations of mean and variance

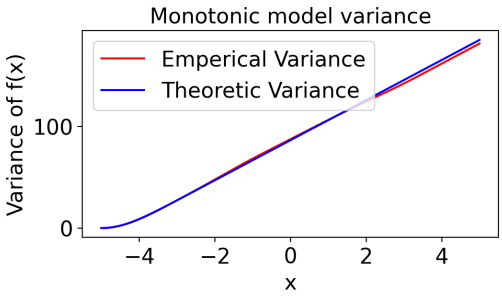
As in the previous section, we verify the above results by simulating prior samples.

Method: We sample 10000 g_i of a zero-mean Gaussian process evaluated in 1000 uniformly distanced points on the interval [-5,5]. We then transform each sample by $h_i = g_i^2$ and use numeric integration to obtain f_i. We set $f_0 = 0$. Finally, we calculate the mean and variance of all 10000 realizations. We use the squared exponential kernel as covariance function with $\ell = 1.1$ and $\kappa = 1.5$.

In figure 5.3, we see the linearity of both mean and variance in both empirical and theoretic functions. The empirical mean seems to be spot on the theoretic mean, and the empirical variance has a little bit of uncertainty around the theoretical, just like for the positive model. Thus we have verified the analytical results.



(a) Visualisation of the theoretical mean with the empirical mean computed on 10000 samples.



(b) Visualisation of the theoretical variance with the empirical mean computed on 10000 samples.

Figure 5.3

5.3 U-shaped functions

Finally, we expand the model to u-shaped functions. By a u-shaped function, F, we understand a convex function with exactly one 'tipping point' x' such that $F'(x) \leq 0$ for $x \leq x'$ and $F'(x) \geq 0$ for $x \geq x'$. A characteristic of a convex function is that its second derivative is non-negative for all x. By integrating the positive function from the previous sections twice, we obtain a convex function F. Furthermore, if we ensure that $F'(x) = 0$ for some x, we obtain a u-shaped function F. In practice, we do this by modelling a monotonically increasing function

$$f(s) = f_0 + \int_a^s t(g(s'))\,\mathrm{d}s', \quad f_0 < 0 \tag{5.17}$$

and integrating it from a to x to obtain

$$\begin{aligned} F(x) &= F_0 + \int_a^x \left(f_0 + \int_a^s t(g(s'))\,\mathrm{d}s' \right) \mathrm{d}s \\ &= F_0 + f_0(x-a) + \int_a^x \int_a^s t(g(s'))\,\mathrm{d}s'\,\mathrm{d}s. \end{aligned} \tag{5.18}$$

Once again, we use the HS approximation of g with $a = -L$ to obtain an analytical expression for the double integral:

$$\begin{aligned} F(x) &\approx F_0 + (x+L)f_0 + \int_{-L}^x \sum_{ij} \alpha_i \alpha_j \psi_{ij}(s)\,\mathrm{d}s \\ &= F_0 + (x+L)f_0 + \sum_{ij} \alpha_i \alpha_j \int_{-L}^x \psi_{ij}(s)\,\mathrm{d}s \\ &= F_0 + (x+L)f_0 + \boldsymbol{\alpha}^T \boldsymbol{\Psi} \boldsymbol{\alpha} \end{aligned} \tag{5.19}$$

where

$$[\boldsymbol{\Psi}]_{ij} = \int_{-L}^x \psi_{ij}(s)\,\mathrm{d}s = \begin{cases} \frac{(x+L)^2}{4L} + \frac{\cos\left(\gamma_{jj}^+(x+L)\right)-1}{2L(\gamma_{jj}^+)^2}, & i = j \\ \frac{1-\cos\left(\gamma_{ij}^-(x+L)\right)}{2L(\gamma_{ij}^-)^2} + \frac{\cos\left(\gamma_{ij}^+(x+L)\right)-1}{2L(\gamma_{ij}^+)^2}, & i \neq j \end{cases} \tag{5.20}$$

where $\gamma_{ij}^{\pm} = \sqrt{\lambda_i} \pm \sqrt{\lambda_j}$. Derivations can be found in appendix A.4.

We can now describe the joint probability density by

$$p(\boldsymbol{\alpha}, \boldsymbol{y}, f_0, F_0) = p(\boldsymbol{y}|\boldsymbol{\alpha}, f_0, F_0)p(\boldsymbol{\alpha})p(f_0)p(F_0) \tag{5.21}$$

The posterior, which we want to infer, is now $p(\boldsymbol{\alpha}, f_0, F_0|\boldsymbol{y})$. The posterior is intractable, and we have to resort to approximation inference.

5.3.1 Prior mean and variance

In this section, we present the prior mean and variance of F given f_0 and F_0. Derivations can be found in appendix A.4.

Assume that $g(x)$ is a Gaussian process with mean μ and a stationary covariance function. Furthermore, assume that the marginal variance is given by $\mathbb{V}[g(x)] = \eta^2$.
Under the given assumptions, the mean of $F(x)$ is

$$\mathbb{E}[F(x)] = F_0 + f_0(x-a) + \frac{\mu^2+\eta^2}{2}(x-a)^2. \tag{5.22}$$

That is, the mean of $F(x)$ is a quadratic function of x, and the coefficient of the quadratic term depends equally on the mean and variance of the original Gaussian process. It is easy to analyse this expression as it is a parabola. The minimum of a parabola is given by $x_{min} = -\frac{B}{2A}$ where we have that

$$A = \frac{\mu^2+\eta^2}{2} \quad \text{and} \quad B = f_0 - a(\mu^2+\eta^2).$$

We can then describe the minima as a function of μ, a, η and f_0.

$$x_{min} = -\frac{f_0 - a(\mu^2+\eta^2)}{\mu^2+\eta^2} \tag{5.23}$$

In practice (for the HS formulation of this model), we have $a = -L$ and $\mu = 0$ such that

$$x_{min} = -\frac{f_0 + L\eta^2}{\eta^2} = -\frac{f_0}{\eta^2} - L \tag{5.24}$$

Now we can already see that we need f_0 to be negative in order to have $x_{min} \in [-L, L]$, and it is easy to see how x_{min} will move around when changing the values of f_0, η and L.

We can also compute the expected values in the boundary of the domain. If we consider $\mu = 0$ and $a = -L$ and compute the prior expectation at the end of the domain, we obtain

$$\mathbb{E}[F(-L)] = F_0 \quad \text{and} \quad \mathbb{E}[F(L)] = F_0 + 2f + 2\eta^2 L^2. \tag{5.25}$$

Thus, the prior mean of the right endpoint depends quadratically on both L and η. Equations (5.24) and (5.25) can provide useful information when choosing prior distributions on f_0 and η given L. In 5.4.1, we will go into more detail on the choice of f_0, η and L.

The prior variance of F given F_0 and f_0 is

$$\mathbb{V}(F(x)) = \int_a^x \int_a^x \int_a^s \int_a^r 4k(r',s')\mu^2 \,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s + \int_a^x \int_a^x \int_a^s \int_a^r 2k(r',s')^2 \,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s. \tag{5.26}$$

Using a squared exponential kernel, it is possible to compute the variance analytically. Detailed derivations of this and equation (5.26) can be found in appendix A.4.

The following function ω is a helping function we define in order to present the result of equation (5.26) in the case of the squared exponential kernel with magnitude η^2 and

lengthscale ℓ.

$$\begin{aligned}\omega(x,a,\ell,\eta) &= \int_a^x \int_a^x \int_a^s \int_a^r k(r',s')\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \\ &= \eta^2\ell^2\sqrt{\frac{\pi}{2}}\Bigg[\left(\frac{2(x-a)^3}{3\ell}+2(x-a)(1-\ell)\right)\operatorname{erf}\left(\frac{(x-a)}{\sqrt{2}\ell}\right) \\ &\quad+\left(\frac{(2\sqrt{\pi}-5)\ell^2\sqrt{2}}{3\sqrt{\pi}}+\frac{2^{\frac{3}{2}}(x-a)^2}{3\sqrt{\pi}}\right)\exp\left(-\frac{(x-a)^2}{2\ell^2}\right) \\ &\quad+\frac{(5-2\sqrt{\pi})\sqrt{2}\ell^2}{3\sqrt{\pi}}-\frac{(x-a)^2\sqrt{2}}{\sqrt{\pi}}\Bigg]\end{aligned} \tag{5.27}$$

Furthermore, we also have that,

$$\int_a^x \int_a^x \int_a^s \int_a^r 2k(r',s')^2\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s = 2\eta^2 \int_a^x \int_a^x \int_a^s \int_a^r \tilde{k}(r',s')\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \tag{5.28}$$

where $\tilde{k}$ is the squared exponential with magnitude η^2 and length scale $\rho = \frac{1}{\sqrt{2}}\ell$. This means

$$\int_a^x \int_a^x \int_a^s \int_a^r 2k(r',s')^2\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s = 2\eta^2\omega(x,a,\rho,\eta) \tag{5.29}$$

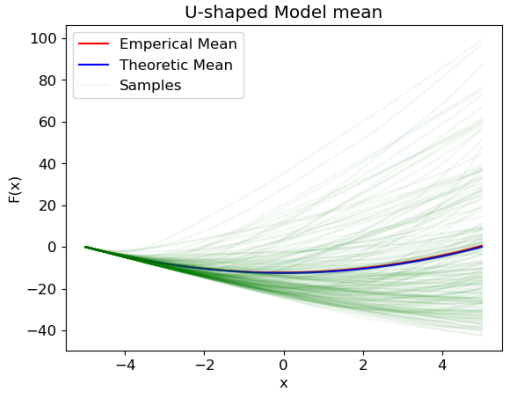
and thus, referring back to (5.26) we obtain

$$\mathbb{V}\left[F(x)\right] = 4\mu^2\omega(x,a,\ell,\eta) + 2\eta^2\omega(x,a,\rho,\eta). \tag{5.30}$$

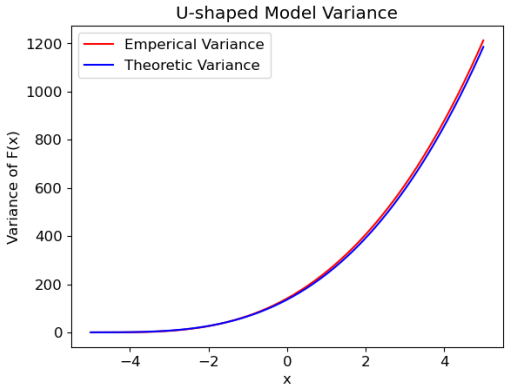
5.3.2 Simulations of mean and variance

In this section, we wish to simulate the mean and variance of our process to verify the theoretical results from the previous section. *Method:* We sample 10000 $g(x)$ of a zero-mean Gaussian process evaluated in 1000 uniformly distanced points on the interval [-5,5]. We then transform each sample by $F(x) = F_0 + f_0(x-a) + \int_a^x \int_a^s g(s')^2\,\mathrm{d}s'\,\mathrm{d}s$ and calculate the mean and variance of all 10000 realisations. We use the squared exponential kernel as covariance function with $\ell = 1.1$ and $\kappa = 1$. We let $F_0 = 0$ and $f_0 = -L$ in order to have $\mathbb{E}[x_{min}] = 0$ as described in equation (5.24). The resulting samples are plotted together with the empirical and theoretical mean function in figure 5.4a, and the empirical and theoretical variance functions are plotted in figure 5.4b.

The theoretical and empirical mean seems to be spot on after 10000 samples. The empirical variance seems to have a little bit of uncertainty around the theoretical variance. Numerical instabilities could cause this, as we use numeric integration to calculate the samples. Generally, it is clear, just like for the monotonic model, that the variance seems to grow drastically when x is growing. However, it is worth mentioning that we have not imposed a prior on F_0 or f_0 in this theoretic derivation. This will be discussed in section 5.3.3.



(a) Visualisation of the theoretical mean with the empirical mean computed on 10000 samples.



(b) Visualisation of the theoretical variance with the empirical variance computed on 10000 samples.

Figure 5.4

5.3.3 Choosing priors for f_0 and F_0

So far, we have considered f_0 and F_0 as given constants. In this section, we will impose suitable priors on f_0 and F_0 in order to gain more model flexibility.

First, we need to understand the role of f_0 and F_0 in the model. As we saw in equation (5.25), the F_0 parameter is the value of $F(-L)$. Therefore, it is crucial to choose a prior for F_0 that provides the model with sufficient flexibility. This is very clear in figure 5.4a, where we see that the prior variance tends to zero in the left part of the domain.

The parameter f_0 is the value of $F'(-L)$, and thereby f_0 denotes the slope of F in $-L$.

By construction, $F'(x)$ is a monotonically increasing function. Since we wish to model functions such that there exists an x_{min} for which $F'(x_{min}) = 0$, we enforce that f_0 is negative, as we saw in equation (5.24). Remark that this doesn't guarantee that there exists an $x_{min} \in [-L, L]$ as we can still have that $x_{min} > L$ for $f_0 \leq -2L\eta^2$ cf. equation (5.24).

In the following, we test the u-shaped model with different priors on a small toy data example. The dataset is given by

$$y_i = h(x_i) + \epsilon_i \tag{5.31}$$

where

$$h(x) = \begin{cases} 10 & \text{for } |x| > 2 \\ 0 & \text{for } |x| \leq 2 \end{cases} \tag{5.32}$$

for $x_1, \ldots, x_{100} \sim \mathcal{U}(-2.5, 2.5)$ and $\epsilon_i \sim \mathcal{N}(0, 1)$. We split up the data randomly into equally sized training and test sets.

Initially, we impose a standard Gaussian prior for F_0 and a negative standard half-normal prior for f_0 and use Hamiltonian Monte Carlo for inference as described in section 2.3. The HMC algorithm has been run with 4000 posterior samples discarding the first 1000 as warm up. We choose $L = 5$ and $m = 10$ basis functions. The result can be seen in figure 5.5.

It appears that although the model manages to capture the shape for high values of x, where the prior variance of F given F_0 and f_0 is high, it fails to fit the data for values of

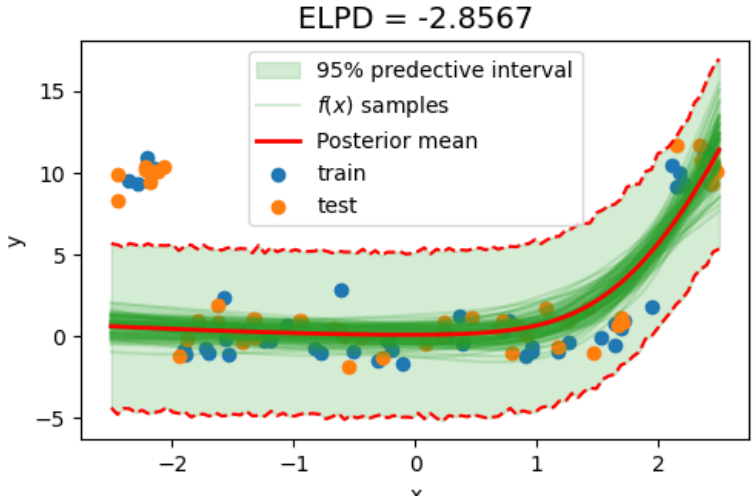


Figure 5.5: Results of HMC inference when imposing a standard Gaussian prior on F_0 and standard half-normal prior on f_0. We see that this model fails to incorporate enough variance in the left endpoint. We have chosen $L = 5$ and $m = 10$ basis functions.

x closer to $-L$. To fix this, we must increase the variance of the F_0 and f_0 priors. We can do this by using more suitable scale parameters in the prior distributions or by choosing distributions with heavier tails. By increasing the variance on the prior on F_0, we ensure higher flexibility of our model for input values near $-L$. By giving the prior on f_0 more variance, we give F more 'space' to decrease before eventually tipping over to increasing.

To try and solve the problems detected in figure 5.5, we will do two things. First, we will consider other distributions as priors for f_0 and F_0. Secondly, we will impose half-normal hyperpriors on the scale parameters of the distribution for f_0 and F_0.

For F_0, an alternative to the Gaussian distribution could be a Student's t distribution, which has heavier tails than a Gaussian. We use 4 degrees of freedom. For f_0, we considered two alternatives to the negative half-normal distribution: The negative Gamma distribution and the negative log-normal distribution. For the Gamma distribution, we choose $k = 2$.

In summary, we test the following priors for F_0:

Model 1	**Model 2**
$F_0 \sim \mathcal{N}(0, \sigma^2_{F_0})$	$F_0 \sim \mathrm{t}(0, \sigma^2_{F_0}, 4)$
$\sigma_{F_0} \sim \mathrm{Halfnormal}(1)$	$\sigma_{F_0} \sim \mathrm{Halfnormal}(1)$,

combined with the following priors for f_0:

Model 1	**Model 2**	**Model 3**
$f_0 \sim \mathrm{Negative\ Halfnorm}(\sigma^2_{f_0})$	$f_0 \sim \mathrm{Negative\ Gamma}(2, \sigma^2_{f_0})$	$f_0 \sim \mathrm{Negative\ Lognormal}(0, \sigma^2_{f_0})$
$\sigma_{f_0} \sim \mathrm{Halfnorm}(1)$	$\sigma_{f_0} \sim \mathrm{Halfnorm}(1)$	$\sigma_{f_0} \sim \mathrm{Halfnorm}(1)$

We measure the accuracy of the models using the expected log predictive density, ELPD, which is presented in section 6.0.2. The results can be seen in figure 5.6. We observe that the greatest improvement is obtained when using the Student's t-distribution as prior

for F_0 rather than a Gaussian. The priors on f_0 perform roughly equally, although the negative Gamma distribution has the lowest ELPD.

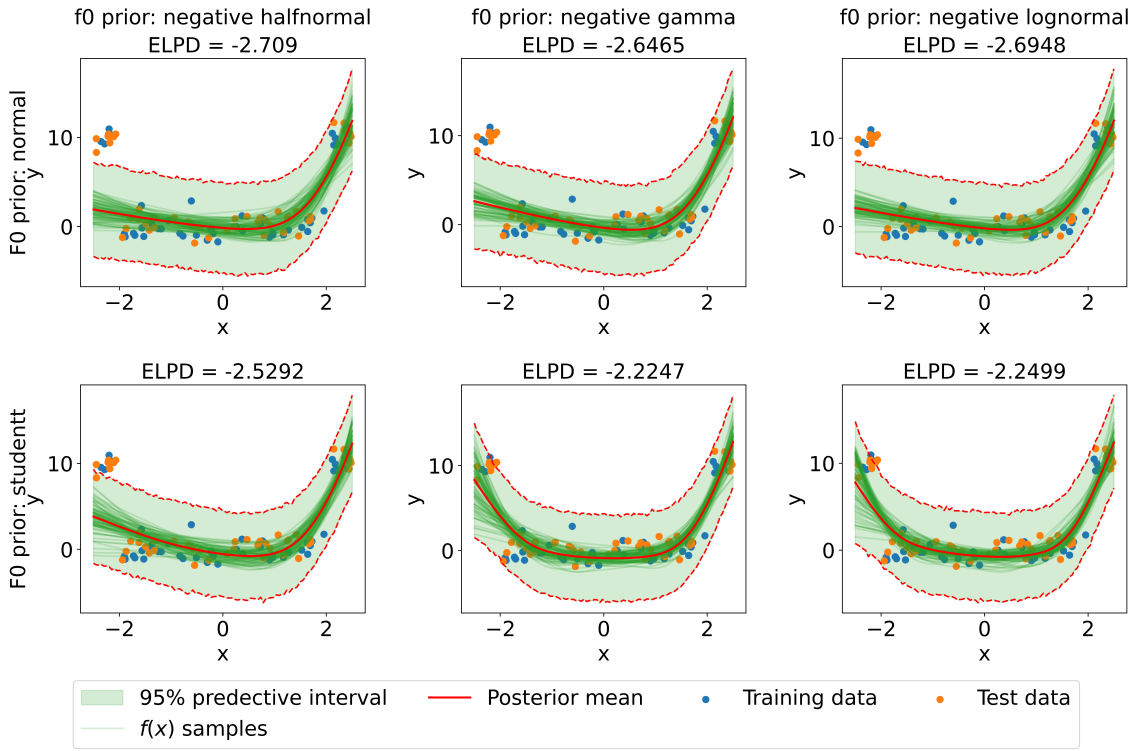


Figure 5.6: Results of HMC inference for different combinations of priors on F_0 and f_0. We have chosen $L = 5$ and $m = 10$ basis functions. The ELPD is the estimated log probability density on the given test data as defined in equation (6.7). We see a large improvement when using a Student's t prior on F_0 rather than a Gaussian.

However, we also see that it is still not enough to capture the data around $x = -2.5$. We attempt to remedy this by adding a mean to the prior, μ_{F_0}. Rather than introduce another hyperparameter to the model, we introduce the following heuristic: Pick out the first n data points and perform a linear regression $l(x) = a + bx$. Compute $l(-L)$ and let this be the mean of F_0, i.e. $\mu_{F_0} = l(-L)$. In this case, we evaluate μ_{F_0} based on the first quarter of the data points. The results are seen in 5.7. We see that the heuristic has improved the fit. Based on this small experiment, we choose to model f_0 with a negative gamma prior and F_0 with a Student's t prior with a mean determined by the heuristic. The choice of priors can also be adapted based on the data at hand, but for simplicity, we used the same priors for all experiments in section 6.

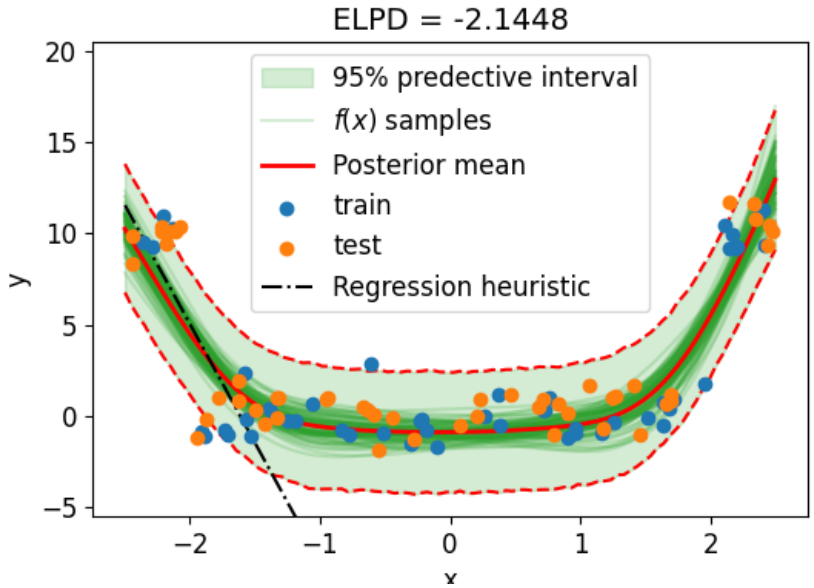


Figure 5.7: Results of imposing a regression heuristic for choosing the prior mean of F_0. F_0 has a Student's t distribution with mean μ_{F_0} determined by the heuristic.

5.3.4 Relaxing the convexity

The model we have defined in section 5.3 is strictly convex by construction. This may be an advantage in cases where data is assumed convex, but it can definitely also be a disadvantage in cases where the data is not strictly convex. It may be that the data we wish to model is *unimodal*. That is, we wish to model F with only one mode, such that there exists an x' where $F(x) \geq F(x')$ for all $x \leq x'$ and $F(x) \leq F(x')$ for $x \geq x'$.

If we switch the sign of F from the previous sections, we obtain a unimodal, concave function. This means that if we can relax the convexity of the u-shaped model, we can use it to model unimodal functions by flipping the sign.

In this section, we start by attempting to relax the convexity by adding a Gaussian process to the strictly convex model. Afterwards, we flip the sign and test our method on unimodal data.

We will relax the convexity by adding a Hilbert space approximated Gaussian process to our convex model defined in (5.18) in three different ways. The most straightforward approach is to obtain a relaxed model by adding a Gaussian process directly to the u-shaped model F:

Model 1:

$$F_{rel}(x) = F(x) + g(x) \approx F_0 - (x + L)f_0 + \sum_{ij} \alpha_i \alpha_j \int_{-L}^{x} \psi_{ij}(s)\,\mathrm{d}s + \sum_i \beta_i \phi_i(x) \tag{5.33}$$

Here, g is assumed to have a zero-mean Gaussian prior with its own individual covariance function, which we estimate using the Hilbert space approximation. Thus $\alpha_i \sim \mathcal{N}\left(0, S_1\left(\sqrt{\lambda_i}\right)\right)$ and $\beta_i \sim \mathcal{N}\left(0, S_2\left(\sqrt{\lambda i}\right)\right)$ where S_1, S_2 denotes the spectral density of the covariance functions of the concave part and the GP part respectively.

Instead of adding the Hilbert space approximated Gaussian process directly to F, we also attempt to add the Gaussian process term to the first and second derivatives of F,

respectively. This gives rise to the following models:

Model 2:

$$F_{rel}(x) = F(x) + \int_a^x g(s)\,\mathrm{d}s \approx F_0 - (x+L)f_0 + \sum_{ij} \alpha_i \alpha_j \int_{-L}^x \psi_{ij}(s)\,\mathrm{d}s + \sum_i \beta_i \int_a^x \phi_i(s)\,\mathrm{d}s \tag{5.34}$$

for $\alpha_i \sim \mathcal{N}\left(0, S_1\left(\sqrt{\lambda_i}\right)\right)$ and $\beta_i \sim \mathcal{N}\left(0, S_2\left(\sqrt{\lambda i}\right)\right)$ with S_1, S_2 defined as above.

Model 3:

$$\begin{aligned} F_{rel}(x) &= F(x) + \int_a^x \int_a^s g(s')\,\mathrm{d}s'\,\mathrm{d}s \\ &\approx F_0 - (x+L)f_0 + \sum_{ij} \alpha_i \alpha_j \int_{-L}^x \psi_{ij}(s)\,\mathrm{d}s + \sum_i \beta_i \int_a^x \int_a^s \phi_i(s')\,\mathrm{d}s'\,\mathrm{d}s \end{aligned} \tag{5.35}$$

for α_i and β_i as before.

Due to the simplicity of the Laplace eigenfunctions, it is easy to compute the basis functions of the Gaussian process terms by

$$\int_{-L}^x \phi_i(s)\,\mathrm{d}s = \int_{-L}^x \frac{1}{\sqrt{L}} \sin\left(\sqrt{\lambda_i}(x+L)\right)\,\mathrm{d}s = \int_0^{\sqrt{\lambda_i}(x+L)} \frac{\sin(u)}{\sqrt{\lambda_i}\sqrt{L}}\,\mathrm{d}u = \frac{1-\cos\left(\sqrt{\lambda_i}(x+L)\right)}{\sqrt{\lambda_i}\sqrt{L}} \tag{5.36}$$

and

$$\int_{-L}^x \int_{-L}^s \phi_i(s')\,\mathrm{d}s'\,\mathrm{d}s = \int_{-L}^x \frac{1-\cos\left(\sqrt{\lambda_i}(s+L)\right)}{\sqrt{\lambda_i}\sqrt{L}}\,\mathrm{d}s = \frac{x+L}{\sqrt{\lambda_i}\sqrt{L}} - \frac{\sin\left(\sqrt{\lambda_i}(x+L)\right)}{\lambda_i\sqrt{L}}. \tag{5.37}$$

We test the three models on the toy data generated in 5.39. This data is uni-modal so we flip the sign on the three model formulas from above. The data is generated using the following function:

$$y_i = h(x_i) + \epsilon_i \tag{5.38}$$

where

$$h(x) = 2\sin\left(\frac{\pi(x+5)}{5} - \frac{\pi}{2}\right) + 2 \tag{5.39}$$

for $x_1, \ldots, x_{100} \sim \mathcal{U}(-5, 5)$ and $\epsilon_i \sim \mathcal{N}(0, 1)$. The data points are randomly divided into equally large test- and training sets. For comparison, we also train a concave model, $-F(x)$, from section 5.3 and a baseline zero-mean Gaussian process.

We let $m = 40$ and $L = 10$ for model 1, 2, 3 and the concave model. The results are presented in figure 5.8. We use HMC inference for approximating the posterior, where we run three chains, each with a warm up on 1000 samples and 3000 posterior samples.

The overall picture in figure 5.8 is that we succeed in modelling functions that are not strictly concave for both models 1, 2 and 3. Since we have only performed a single run on one realization of the data, it is difficult to draw any conclusions ranking the models. Still, it appears that both models 2 and 3 are comparable to the baseline GP, with model 3 outperforming the others in this specific case. The experiment also illustrates the shortcomings of the strictly concave model.

In figure 5.8, we have also plotted the contributing components to the three relaxed models. If we point our attention to model 1 (column 2), we can see in the component plot f) that

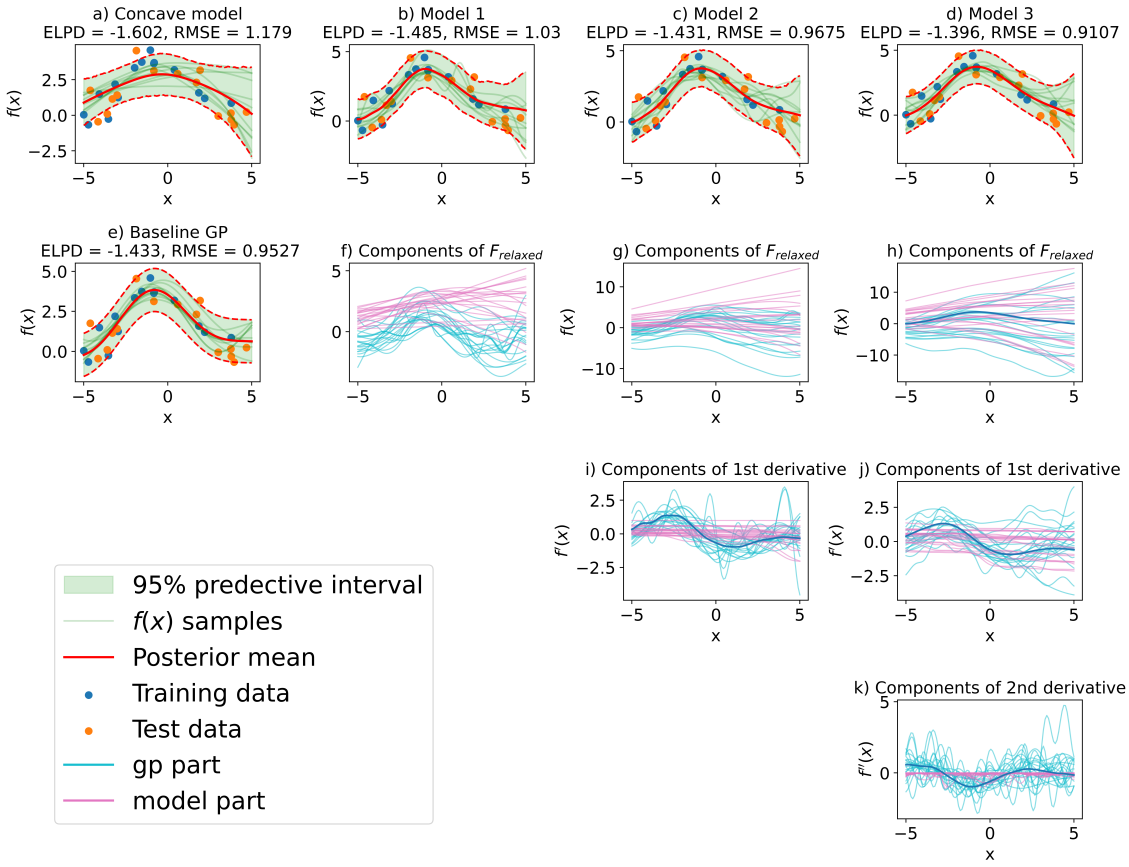


Figure 5.8: Plot of the different "relaxed" u-shaped models. Column 1: a) The strictly concave model described in section 5.3 and e) a baseline Gaussian process. Column 2: b) Model 1 with the fitted model, f) the GP part and the concave part of the model individually. Column 3: c) Model 2 with the fitted model, g) the two components of model 2 coming from the GP part and the concave part individually, i) the derivatives of the two components. Column 4: d) Model 3 with the fitted model, h) the two components of model 3 coming from the GP part and the concave part individually, j) the first derivatives of the two components, k) the second derivatives of the two components. The HMC algorithm (Described in section 6.0.4) has been run with 4000 posterior samples discarding the first 1000 as warm up.

the pink samples, coming from the strict model, are relatively smooth and effectively play the role of an intercept term. The contribution from the Gaussian process term effectively dictates the shape of the fit. The same picture is present for model 2 when considering the components of the first derivative in figure 5.8 i) and for model 3 when considering the components of the second derivative in figure 5.8 k).

Overall, there is a tendency for the Gaussian process to overrule the shape-constrained part of the model. In the 'differentiation' layer where it is added, the GP will be responsible for the largest contribution of the variance. This highlights the main problem with our relaxation approach, which is calibrating the balance between the contribution from the Gaussian process and the contribution from the concave model. However, the variance contributions are evened out by the integration process. This can be seen in figure 5.9, which illustrates the variance of the components in model 3. From this, it is evident that the GP doubles the variance of model 3 and thus adds a significant amount of flexibility.

Conclusively, we have proposed relaxations of the u-shaped model from section 5.3, which suggests a specific shape to a higher or smaller degree. It appears that model 3 is the most balanced between the contributions of the GP term and the concave term.

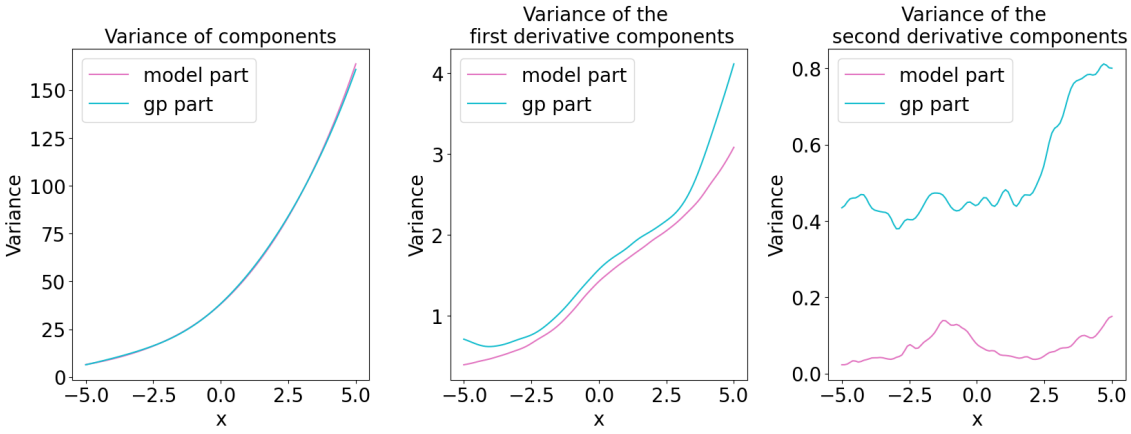


Figure 5.9: The variance of the components of model 3 estimated on 9000 samples. $F_{rel}(x) = g(x) + F(x)$, where $g(x)$ is the GP term and $F(x)$ is the concave term as given in 5.35. The left-hand plot shows the variance of $g(x)$ and $F(x)$, respectively. The middle plot shows the variance of $g'(x)$ and $F'(x)$, and the right-hand plot shows the variance of $g''(x)$ and $f''(x)$.

Based on the observations and insights obtained in this small experiment, we decide to include the relaxed model 3 in the experiment 2 in chapter 6 and see how well it performs in different u-shaped regression problems.

5.4 Discussion on the Application and Configuration of Shape-Constrained HS Models

In this section, we will discuss some general topics of the three models in terms of the effect of m and L and how to configure the models by taking the insights on the prior mean and variance into account. We will also discuss computational complexity and how the models scale in for multiple dimensions. We will primarily focus on the monotonic and the u-shaped, as these are the two models we will use in the experiments in chapter 6.

5.4.1 The effect of m and L

A general challenge you are faced with when dealing with the shape-constrained HS models proposed in this work is the choice of basis functions m and domain length controlled by L.

Riutort-Mayol et al. (2022), investigates how to choose the appropriate m and L for the HS approximation from section 3.4. In the shape-constrained models we have proposed in this section, we face a similar problem. However, the behaviour of the models is different due to the transformations that we apply to the Gaussian process in terms of integration and squaring.

When using the Hilbert space approximation for Gaussian process regression as described in 3.4 we have to be aware that the posterior variance goes to zero in the boundary points. Therefore we usually choose L big enough such that the distribution of the posterior

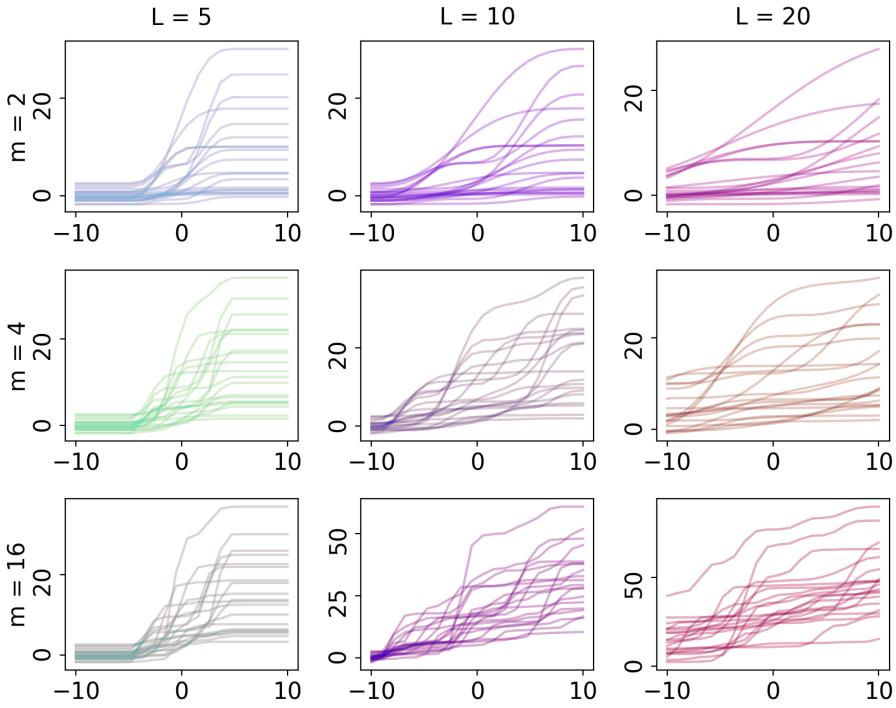


Figure 5.10: Prior samples of monotonic functions f for different values of m and L. We have $f_0 \sim \mathrm{t}(0, \sigma_{f_0}, 4)$. The covariance function is a squared exponential kernel with magnitude $\kappa = 1$ and length scale $\ell = 1$

samples are not affected by being too near the boundary of the domain $[-L, L]$. The shape-constrained functions we have proposed do not necessarily behave like this.

For the positive function, $h(x)$, we still have that the model is a zero function for all $x \notin [-L, L]$ and that $h(-L) = 0$ and $h(L) = 0$. For the monotonic we have that $f(-L) = f_0$ and $f(L) = f_0 + \alpha^\top \alpha$ since

$$\psi_{i,j}(L) = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j. \end{cases}$$

This means that the function now has a trainable distribution in the boundary points given by $f(-L) = f_0 \sim p(f_0 \,|\, \boldsymbol{y})$ and $f(L) = f_0 + \alpha^\top \alpha$ where $f_0 \sim p(f_0 \,|\, \boldsymbol{y})$ and $\alpha^\top \alpha \sim p(\alpha^\top \alpha \,|\, \boldsymbol{y})$. The u-shaped model also has a posterior distribution boundary points, which is even more complex than for the monotonic. We will refrain from going into detail here. The main point is that the boundary values of the monotonic and the u-shaped models don't evaluate to zero. Thus, it is no longer crucial to choose L far away from the data, as the model is not restricted to a constant value at the boundary.

The integration also has a smoothening effect on the GP samples, which makes the interpretation of m and L less intuitive. It is especially difficult to see the effect of m on the model flexibility in the u-shaped model, which is visualised in figure 5.11.

We have plotted prior samples from different configurations of m and L for the monotonic and the u-shaped model, respectively, in order to examine their effect on the prior distributions.

In figure 5.10, we see how m and L affect the prior samples from the monotonic model. The most apparent effect is that lower values of m correspond to flatter functions, whereas

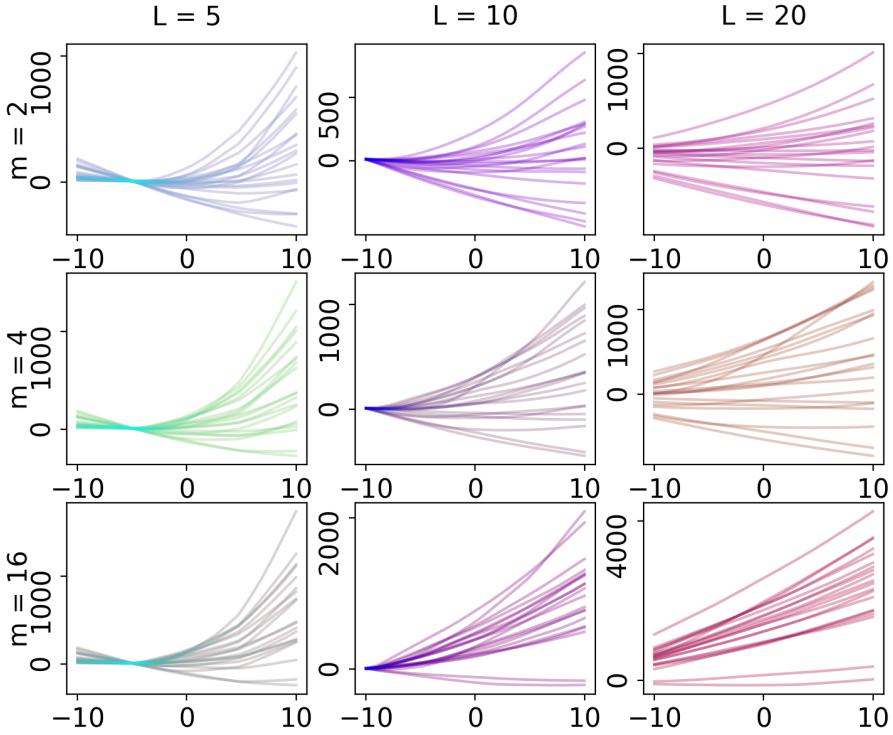


Figure 5.11: Prior samples of u-shaped functions F for different values of m and L. We have $F_0 \sim \text{t}(0, \sigma^2_{F_0}, 4)$ and $f_0 \sim$ Negative Gamma$(2, \sigma^2_{f_0})$. The covariance function is a squared exponential kernel with magnitude $\kappa = 1$ and length scale $\ell = 1$

higher values of m result in more 'wiggly', flexible functions. This is due to the inclusion of eigenmodes with higher frequencies in the approximation of the underlying GP. We note that for $x \notin [-L, L]$, the functions become constant functions, which is caused by the underlying approximated GP being zero outside the domain. The final, subtle effect to note is that a higher value of L also has a smoothing effect. This is because increasing the domain 'stretches' out the basis function, giving it a lower frequency.

For the u-shaped model, the picture is more muddled, as illustrated in 5.11. The clearest effect is that of L, which determines where the minima of the functions are as well as the intersection with f_0. We also see that for $x \notin [-L, L]$, the u-shaped functions become linear functions. It is hard to detect much change in the shape of the functions depending on m, although intuitively, it should provide the model with a more flexible curvature. The smoothening from the integration causes this. We do notice that the functions have larger variances for larger values of m.

In the formula for the minimum of the prior mean in equation (5.24), we also see that the relationship between the marginal variance of the GP and the choice of f_0 is crucial in terms of the shape of the model. In figure 5.12, we have plotted some prior samples for different values of f_0 and κ where κ is the magnitude parameter of the squared exponential kernel. As expected from equation (5.24), we see that low values of κ result in more linear samples in figure 5.12 (a). We also observe less variance in the distribution since the variance depends quadratically on κ, as seen in plot (b). In (a), we also see how a negative f_0 corresponds to a negative slope in the boundary endpoint $-L$. In (b), where f_0 is close to zero, the result is samples looking like constant functions. For larger values of κ, we see that the prior variance grows when $x \to 10$. In (c), we give an example where κ dominates

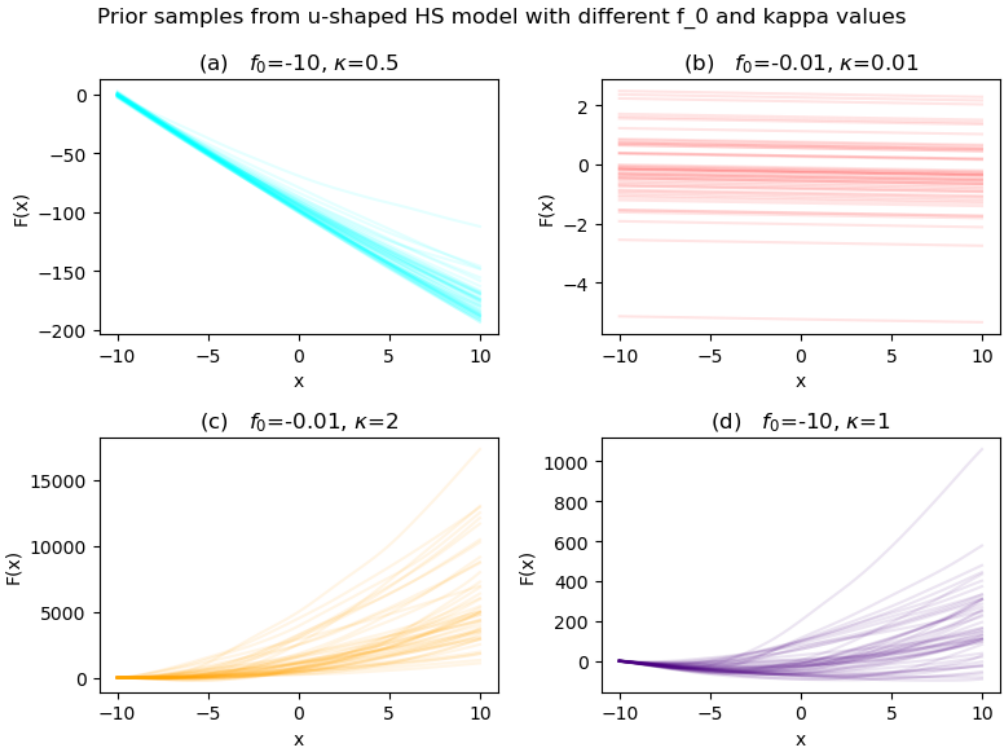


Figure 5.12: Samples from prior distributions of the u-shaped model for different values of κ and f_0. The plots display the dynamics between these two parameters based on equation (5.24).

f_0, which results in increasing samples, and in (d), we have chosen rather big values for f_0 and κ such that $x_{min} = 0$.

In the experiments in section 6, we deal choose the parameters by performing a thorough model selection, searching through different combinations of m and L.

5.4.2 Computation complexity

When evaluating the models we have to compute a matrix product given by $\boldsymbol{\alpha}^\top \Phi^\top \Phi \boldsymbol{\alpha}$ for the positive model, $\boldsymbol{\alpha}^\top \psi \boldsymbol{\alpha}$ for the monotonic model and $\boldsymbol{\alpha}^\top \Psi \boldsymbol{\alpha}$ for the u-shaped model.

The computation of these matrix products requires an initial operation of $\mathcal{O}(nm^2)$ for computing $\mathbf{Q}$, where $\mathbf{Q} \in \{\Phi^\top \Phi, \psi, \Psi\}$.

After this initial cost, evaluating the function for a new $\boldsymbol{\alpha}$ – possible generated by different hyperparameters or kernels – requires us to compute $\boldsymbol{\alpha}^\top \mathbf{Q} \boldsymbol{\alpha} = \sum_{ij} \alpha_i \alpha_j \mathbf{Q}_{ij}$ which has a complexity of $\mathcal{O}(m^2)$.

5.4.3 Multiple dimensions

So far we have only considered the Hilbert space approximation and the shape-constrained models in a one-dimensional setting. In Solin and Särkkä (2020), it is examined how to use the Hilbert space approximation of a standard GP in a multivariate setting. One problem is that the model complexity scales exponentially with the number of dimensions of the data, as you have to evaluate $\hat{m}^d$ combinations of the basis functions. If we were to try to implement our monotonic or u-shaped models in a multivariate setting, it would still be analytically feasible, but it would scale even more drastically than the Hilbert space approximation. If we were to define the monotonic model in a two-dimensional case, we

would have the Hilbert space approximation given by

$$g(x_1, x_2) = \sum_{j_1,j_2}^{\hat{m}} \alpha_{j_1,j_2} \phi_{j_1,j_2}(x_1, x_2) \tag{5.40}$$

where $\alpha_{j_1,j_2} \sim \mathcal{N}(0, S(\sqrt{\lambda_{j_1,j_2}})$ and λ_{j_1,j_2} and $\phi_{j_1,j_2}(x_1, x_2)$ is defined as in (2.26) and(2.27). Then, the multivariate monotonic function is given by

$$\begin{aligned} f(\boldsymbol{x}) =& f_0 + \int_{-L_1}^{x_1} \int_{-L_2}^{x_2} g(x_1, x_2)^2 \, \mathrm{d}s_2 \, \mathrm{d}s_1 \\ =& f_0 + \int_{-L_1}^{x_1} \int_{-L_2}^{x_2} \left(\sum_{j_1,j_2}^{\hat{m}} \alpha_{j_1,j_2} \phi_{j_1}(x_1) \phi_{j_2}(x_2) \right)^2 \mathrm{d}s_2 \, \mathrm{d}s_1 \\ =& f_0 + \int_{-L_1}^{x_1} \int_{-L_2}^{x_2} \sum_{j_1,j_2,i_1,i_2}^{\hat{m}} \left(\alpha_{j_1,j_2} \alpha_{i_1,i_2} \phi_{j_1}(x_1) \phi_{j_2}(x_2) \phi_{i_1}(x_1) \phi_{i_2}(x_2) \right) \mathrm{d}s_2 \, \mathrm{d}s_1 \end{aligned} \tag{5.41}$$

It becomes clear that the scaling is even worse for our shape-constrained models, as the squaring of the GP's results in $(\hat{m}^d)^2$ combinations of basis functions. This scaling issue holds for both the positive, the monotonic and the u-shaped model.

One way to deal with this problem would be to consider another positive function instead of $t(g(\boldsymbol{x})) = g(\boldsymbol{x})^2$, however, as we discussed in 5.1.2 it has proven difficult to find another positive function that provides the same analytical benefits. It would also only reduce the scaling to $\hat{m}^d$.

Another way to deal with the scaling issue could be to look into a more simple model, such as a generalised additive model, given by

$$f(x_1, x_2, \ldots, x_d) = \sum_{k=1}^{d} f_k(x_k), \tag{5.42}$$

where each $f_k(x_k)$ is either positive, monotonic or u-shaped depending on the modelling problem. That would result in a more scalable model. The cost is that we must assume independence across the dimensions, which results in a less flexible model.

6 Experiments

Introduction to experiments

In this section, we wish to apply the monotonic and u-shaped models from the former section in different regression problems to see how they behave in practice. We will also compare the models with other shape-constrained models in order to illuminate the last three research questions:

2. *What are the advantages and limitations of using the Hilbert space approximation for enforcing shape-constrained functions, and how does it compare to other shape constrained models?*

3. *How do shape-constrained functions affect prediction accuracy in the two regression regimes, interpolation and extrapolation?*

4. *To what extent do the use of shape-constrained functions improve data efficiency?*

We have set up four different experiments:

Experiment 1: This experiment is based on 6 different monotonic benchmark functions from which we generate synthetic data by adding noise to the functions. The same functions have been used in Ustyuzhaninov et al. (2020) and Maatouk (2017).

Experiment 2: This is analogous to experiment 1 but in a u-shaped setting, where we have created six u-shaped (convex or unimodal) benchmark functions.

The main purpose of experiments 1 (monotonic model) and 2 (u-shaped model) is to test the HS model on a variety of interpolation tasks in a controlled experiment environment with few data points. We wish to analyse the performance of the model compared to both a non shape-constrained baseline model and other shape-constrained models, which will help us answer research question 2 and 3. By evaluating the shape-constrained models overall, we gain insights for answering research question 4.

Experiment 3: In experiment 3 we want to see how well the monotonic model performs in an extrapolation task, in order to answer research question 2. Therefore we have set up a forecasting problem on time series data (World Bank Group DataBank 2024).

Experiment 4: Here we wish to investigate how the u-shaped model performs in an interpolation problem and how well it performs in terms of data efficiency. We test the models on different sizes of real u-shaped data (Blanchflower and Oswald 2008). We wish to test the data efficiency, of a non constrained baseline model, the HS model and another shape-constrained model on datasets decreasing in size. This will help us answer research question 4.

6.0.1 Models in comparison

In the experiments, we will test the following models. We will not go into detail with the models but describe the core ideas of each of them.

Hilbert space model (HS model): The monotonic and u-shaped Hilbert space model introduced in section 5.2 and 5.3.

The relaxed Hilbert space model (HS relaxed model): Model 3 as described in section 5.3.4.

The virtual point model (VP model): Described in Riihimäki and Vehtari (2010). This method can be used to model both monotonic and convex/concave functions. The model utilizes that derivatives of a Gaussian process is another Gaussian process. In Riihimäki and Vehtari (ibid.), M virtual observations (x_i, m_i) are added to the model, such that

$$m_i = \begin{cases} 1 & \text{to impose } f^{(n)}(x_i) \geq 0 \\ 0 & \text{to impose } f(n)(x_i) < 0 \end{cases} \tag{6.1}$$

where $n \in \{1, 2\}$ determines whether we are modelling the first derivative f' (resulting in a monotonic model) or the second derivative f'' (resulting in a convex/concave model). For instance, if we set $n = 2$ and $m_i = 0$ for all i, this corresponds to a concave model. The joint distribution of the observed data and virtual observations is a Gaussian with zero mean and covariance

$$K = \begin{pmatrix} K_{\boldsymbol{f},\boldsymbol{f}} & K_{\boldsymbol{f},\boldsymbol{f}^{(n)}} \\ K_{\boldsymbol{f}^{(n)},\boldsymbol{f}} & K_{\boldsymbol{f}^{(n)},\boldsymbol{f}^{(n)}} \end{pmatrix}. \tag{6.2}$$

A probit likelihood is used to model the sign of the derivative observations in combination with the likelihood of the data. The joint distribution of the model is

$$p\left(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{f^{(n)}}\right) = \underbrace{p\left(\boldsymbol{y}|\boldsymbol{f}\right)}_{\text{Likelihood of data}} \underbrace{\prod_{i=1}^{M} \text{Bern}\left(\boldsymbol{m}_i \,|\, \Phi\left(\boldsymbol{f}_i^{(n)} \frac{1}{\nu}\right)\right)}_{\text{Likelihood of derivative information}} \underbrace{\mathcal{N}\left(\begin{pmatrix}\boldsymbol{f} \\ \boldsymbol{f}^{(n)}\end{pmatrix} \middle| \boldsymbol{0}, K\right)}_{\text{Joint prior of data and virtual derivatives}}, \tag{6.3}$$

where ν is a scaling factor on the probit function. The advantage of using the probit function instead of the step function as the likelihood is that it tolerates small errors. The probit function approaches the step function when $\nu \to 0$.

The Monotonic Gaussian Process Flows model (SDE model): This model is presented in Ustyuzhaninov et al. (2020) and imposes the monotonicity constraint based on numerical solutions of stochastic differential equations. The core idea is to use a monotonically increasing set of inputs $x_1 \leq x_2 \leq \cdots \leq x_n$ as initial conditions for the same realization ω of a stochastic differential equation (SDE) and use the solutions at time T as the outputs $f(x_i)$. Since the noise driving the system (determined by ω) is the same for all trajectories, the monotonicity will be preserved, i.e. $f(x_1) \leq f(x_2) \leq \ldots f(x_n)$. The drift and diffusion terms of the SDE are defined as the mean and the covariance from a Gaussian process specified via M inducing outputs $\boldsymbol{U}$ corresponding to M inducing points $\boldsymbol{z}_i$ in the spatio-temporal domain. That is, the posterior of the Gaussian process is given by

$$p(g(s,t)|\boldsymbol{U}, \boldsymbol{Z}) \sim \mathcal{N}\left(\mu(s,t), \Sigma(s,t)\right), \tag{6.4}$$

and the SDE is given by

$$dS(t,\omega;x) = \mu\left(S(t,\omega;x),t\right)\, dt + \sqrt{\Sigma\left(S(t,\omega;x),t\right)} dB(t,\omega), \tag{6.5}$$

where $B(t,\omega)$ is Brownian motion.

The joint density of the model is then given by

$$p(\boldsymbol{y}, S(T,\omega;x), g, \boldsymbol{U}) = \underbrace{p(\boldsymbol{y}|S(T,\omega;x))}_{\text{likelihood}} \underbrace{p(S(T,\omega;x)\,|g)}_{\text{SDE}}\; \underbrace{p(g|\boldsymbol{U})p(\boldsymbol{U}).}_{\text{GP prior of g(s,t)}} \tag{6.6}$$

For all experiments, we will also consider a zero-mean Gaussian process as a baseline model.

6.0.2 Evaluation metrics

We will evaluate the performance of the models described in section 6.0.1 with respect to the *expected log probability density* (ELPD) and the *root mean square error* (RMSE). These are given by

$$\begin{aligned} \text{ELPD} &= \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log(p(y_i|\mathcal{D}_{train}), \\ \text{RMSE} &= \sqrt{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(y_i - \hat{f}(x_i)\right)}, \end{aligned} \tag{6.7}$$

where $y_i \in \mathcal{D}_{test}$ is the test data consisting of n_{test} data points, $\mathcal{D}_{train}$ is the training dataset and $\hat{f}$ is the posterior mean. The expected log predictive density measures how well the predictive posterior captures the data. The RMSE measures how well the mean fits the data, and thus how good our predictions will be.

We have to use Monte Carlo estimates in order to derive the two metrics and we find the posterior mean by the following Monte Carlo estimate.

$$\hat{f}(x) = \int f(x)p(f|\mathcal{D}_{train})\,\mathrm{d}f \approx \frac{1}{n_{samples}} \sum_{j=1}^{n_{samples}} f_j(x), \tag{6.8}$$

where $f_j \sim p(f|\mathcal{D}_{train})$.

To estimate the ELPD we need to marginalize with respect to f. We use a Monte Carlo estimator to compute the integral as shown in the following equation.

$$\begin{aligned} \text{ELPD} =& \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log(p(y_i)|\mathcal{D}_{train}) \\ =& \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log\left(\int p(y_i|f)p(f|\mathcal{D}_{train})\,\mathrm{d}f\right) \\ =& \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \log\left(\frac{1}{n_{samples}} \sum_{j=1}^{n_{samples}} p(y_i|f_j)\right) \quad \text{for } f_j \sim p(f_j|\mathcal{D}_{train}). \end{aligned} \tag{6.9}$$

For both the ELPD and the RMSE we will also report the standard deviation of the sample mean in order to assert the robustness of our results. It is given by

$$\text{std}\left[\frac{1}{S}\sum_{i=1}^{S}\mathcal{X}_i\right] = \sqrt{\mathbb{V}\left[\frac{1}{S}\sum_{i=1}^{S}\mathcal{X}_i\right]} = \sqrt{\frac{1}{S^2}\sum_{i=1}^{S}\mathbb{V}\left[\mathcal{X}_i\right]} = \sqrt{\frac{\mathbb{V}[\mathcal{X}_i]}{S}} = \frac{\text{std}[\mathcal{X}_i]}{\sqrt{S}}. \tag{6.10}$$

6.0.3 Model selection

In all the experiments, we attempt to give each model equal chances in performance by finding the best parameter configuration for each model. We do this by performing a model selection on the HS model, the VP model, the SDE model, and the HS relaxed model by looping through selected parameters for each model.

For the HS model, we choose parameters m, the number of basis functions, and L, the size of the domain.

For the HS relaxed model, we also choose m and L, where m denotes the number of basis functions for both the GP and strictly convex parts of the model. Thus, the model consists of $2m$ basis functions in total.

For the VP model, we choose the number of virtual points.

For the SDE model, we choose the parameters M, L, and the kernel to use. M is the number of inducing points, and T is the time span in which we develop the SDE.

We do this by using k-fold cross-validation. However the amount of folds depends on the data. In the simulation experiments (1 and 2), we have unlimited data, so we can repeat the experiment with new seeds as many times as we want. For the data experiments (3 and 4), we have to be more economical. The exact details of the cross-validation are described in each experiment section.

We select the best configuration of model parameters based on the RMSE, following the procedure from Ustyuzhaninov et al. (2020).

6.0.4 Inference

We assume a Gaussian likelihood in all four experiments, i.e.

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I}). \tag{6.11}$$

Hamiltonian Monte Carlo sampling

We use the Hamiltonian Monte Carlo algorithm described in section 2.3 to obtain posterior samples of the baseline GP, the HS model, the relaxed HS model and the VP model. For all other models than the GP, this is a necessity due to the posteriors not having closed forms. For the baseline model, using Hamiltonian Monte Carlo enables us to impose priors on the hyperparameters.

The Hamiltonian Monte Carlo algorithm is implemented in *stan* (Stan Development Team 2024) in the *pystan* integration (Riddell, Hartikainen, and Carter 2021).

For all models we will use the squared exponential kernel with magnitude κ and scale parameter ℓ as covariance function. We impose half normal priors on the kernel magnitude κ and the noise variance σ. In the HS and HS relaxed models, we impose a half-normal prior on the scale parameter ℓ. In the baseline GP and VP model, we use an inverted gamma prior for the scale parameter. Finally, we impose the priors discussed in section 5.3.3 for the HS model and HS relaxed model on f_0 and F_0. We use the same prior for f_0 in the monotonic HS model as we did on F_0 in the u-shaped HS model.

This leaves us with the following joint probabilities of the models:

Baseline GP:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{f}, \sigma, \kappa, \ell) = &\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \sigma)\, \mathcal{N}(\boldsymbol{f} \mid 0, K) \\ &\text{Halfnorm}(\sigma \mid)\, \text{Halfnorm}(\kappa \mid 1)\, \text{InvGamma}(\ell \mid 5, 5), \end{aligned} \tag{6.12}$$

where K is determined by the kernel.

HS monotonic model:

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\alpha}, f_0, \sigma, \kappa, \ell, \sigma_{f_0}) = &\mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma)\, \mathcal{N}(\boldsymbol{\alpha}|0, \Lambda)\, \text{t}\left(f_0|0, \sigma_{f_0}^2, 4\right) \\ &\text{Halfnorm}(\sigma|1)\, \text{Halfnorm}(\ell|1) \\ &\text{Halfnorm}(\kappa|1)\, \text{Halfnorm}(\sigma_{f_0}|1) \end{aligned} \tag{6.13}$$

where $\Lambda = \text{diag}\left(S\left(\sqrt{\lambda_1}\right), \ldots, S\left(\sqrt{\lambda_m}\right)\right)$ and $\boldsymbol{f} = f_0 + \boldsymbol{\alpha}^\top \boldsymbol{\psi} \boldsymbol{\alpha}$ as defined in equation (5.11).

HS u-shaped model:

$$\begin{aligned} p\left(\boldsymbol{y}, \boldsymbol{\alpha}, F_0, f_0, \sigma, \kappa, \ell, \sigma_{F_0}, \sigma_{f_0},\right) = \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{F}, \sigma\right) \mathcal{N}\left(\boldsymbol{\alpha} | 0, \Lambda\right) \mathrm{t}\left(F_0 | \mu_{F_0}, \sigma_{F_0}^2, 4\right) \\ \operatorname{NegLognormal}\left(f_0 | 0, \sigma_{f_0}^2\right) \operatorname{Halfnorm}\left(\sigma | 1\right) \operatorname{Halfnorm}\left(\kappa | 1\right) \\ \operatorname{Halfnorm}\left(\ell | 1\right) \operatorname{Halfnorm}\left(\sigma_{F_0} | 1\right) \operatorname{Halfnorm}\left(\sigma_{f_0} | 1\right) \end{aligned} \tag{6.14}$$

where Λ is as above and $\boldsymbol{F} = F_0 + (\boldsymbol{x} + L) f_0 + \boldsymbol{\alpha}^T \boldsymbol{\Psi} \boldsymbol{\alpha}$ as defined in equation (5.19) and μ_{F_0} is the mean for the prior on F_0. Different ideas for selecting this value is discussed in section 5.3.3.

HS relaxed model:

$$\begin{aligned} p\left(\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}, F_0, f_0, \sigma, \kappa_\alpha, \ell_\alpha, \kappa_\beta, \ell_\beta, \sigma_{F_0}, \sigma_{f_0},\right) \\ = \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{F}_{rel}, \sigma\right) \mathcal{N}\left(\boldsymbol{\alpha} | 0, \Lambda_\alpha\right) \mathcal{N}\left(\boldsymbol{\beta} | 0, \Lambda_\beta\right) \\ \mathrm{t}\left(F_0 | 0, \sigma_{F_0}^2, 4\right) \operatorname{NegLognormal}\left(f_0 | 0, \sigma_{f_0}^2\right) \operatorname{Halfnorm}\left(\sigma | 1\right) \\ \operatorname{Halfnorm}\left(\kappa_\alpha | 1\right) \operatorname{Halfnorm}\left(\ell_\alpha | 1\right) \operatorname{Halfnorm}\left(\kappa_\beta | 1\right) \\ \operatorname{Halfnorm}\left(\ell_\beta | 1\right) \operatorname{Halfnorm}\left(\sigma_{F_0} | 1\right) \operatorname{Halfnorm}\left(\sigma_{f_0} | 1\right) \end{aligned} \tag{6.15}$$

where Λ_α and Λ_β are defined analogously to the previous covariance matrices and $\boldsymbol{F}_{rel}$ is defined as in equation 5.35.

VP model:

$$\begin{aligned} p\left(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{f}^{(\boldsymbol{n})}\right) = \mathcal{N}\left(\boldsymbol{y} | \boldsymbol{f}, \sigma^2\right) \prod_{i=1}^{M} \operatorname{Bern}\left(\boldsymbol{m}_i \,|\, \Phi\left(\boldsymbol{f}_i' \frac{1}{\nu}\right)\right) \mathcal{N}\left(\begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{f}' \end{pmatrix} \middle| \boldsymbol{0}, K\right) \\ \operatorname{Halfnorm}(\kappa \,|\, 1) \operatorname{Halfnorm}(\ell \,|\, 1) \operatorname{InvGamma}(\sigma \,|\, 5, 5) \end{aligned} \tag{6.16}$$

for $n \in \{1, 2\}$ and

$$K = \begin{pmatrix} K_{\boldsymbol{f}, \boldsymbol{f}} & K_{\boldsymbol{f}, \boldsymbol{f}^{(n)}} \\ K_{\boldsymbol{f}^{(n)}, \boldsymbol{f}} & K_{\boldsymbol{f}^{(n)}, \boldsymbol{f}^{(n)}} \end{pmatrix}. \tag{6.17}$$

We set $\nu = 100$. For higher values of ν, the HMC sampler had trouble finding an acceptable initial value.

When performing model selection, we sample 3 chains with 4000 samples in each chain and discard the first 1000 as warm-up samples. When testing the optimal model configuration, we run 4 chains with 20000 samples and discard the first 10000.

Variational inference
For the SDE model, we use variational inference to find the posterior of $\boldsymbol{U}$. We do this by maximising the evidence lower bound of the marginal likelihood given by:

$$\log p(\mathcal{D}|) \geq \mathcal{L} = \mathrm{KL}[q(\boldsymbol{U}) || p(\boldsymbol{U})] + \mathbb{E}_{q(\boldsymbol{U})} \mathbb{E}_{p(S(T, \omega; x) | \boldsymbol{U})}[\log(p(\boldsymbol{y} | S(T, \omega; x))], \tag{6.18}$$

where $q(\boldsymbol{U}) \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$ and $\boldsymbol{m}$ and $\boldsymbol{S}$ are the parameters to optimise. A more detailed explanation of the variational inference procedure can be found in section 3.1 in Ustyuzhaninov et al. (2020).

When performing the model selection, we run 3000 iterations of optimisation and generate 500 posterior samples for the Monte Carlo estimate for the ELPD and the RMSE. When running the best model configuration, we run 10000 iterations of optimisation and generating 500 posterior samples for the Monte Carlo estimate for the ELPD and the RMSE. For numerical solutions of SDE, we use the Euler-Maruyama method with 20 time steps. All of this is done using the implementation from Ustyuzhaninov et al. (2021).

6.1 Experiment 1: Monotonic benchmark functions

In this experiment we will evaluate the performance of the monotonic HS model, the monotonic VP model, the SDE model and the baseline GP on six different monotonic benchmark functions. The functions have been used in several other articles such as Ustyuzhaninov et al. (2020) and Maatouk (2017).

They are given as

$$f_i(x) = \begin{cases} 3 & i = 1 \quad \text{(flat function)} \\ 0.32(x + \sin(x)) & i = 2 \quad \text{(sinusoidal function)} \\ 3 + 3 \cdot \mathbb{1}_{(5,10]}(x) & i = 3 \quad \text{(step function)} \\ 0.3x & i = 4 \quad \text{(linear function)} \\ 0.15 \exp(0.6x - 3) & i = 5 \quad \text{(exponential function)} \\ \frac{3}{1+\exp(-2x+10)} & i = 6 \quad \text{(logistic function)} \end{cases} \tag{6.19}$$

were $\mathbb{1}_{(a,b]}(x)$ is an indicator function over the interval $(a, b]$.

The benchmark functions are visualised in figure 6.1.

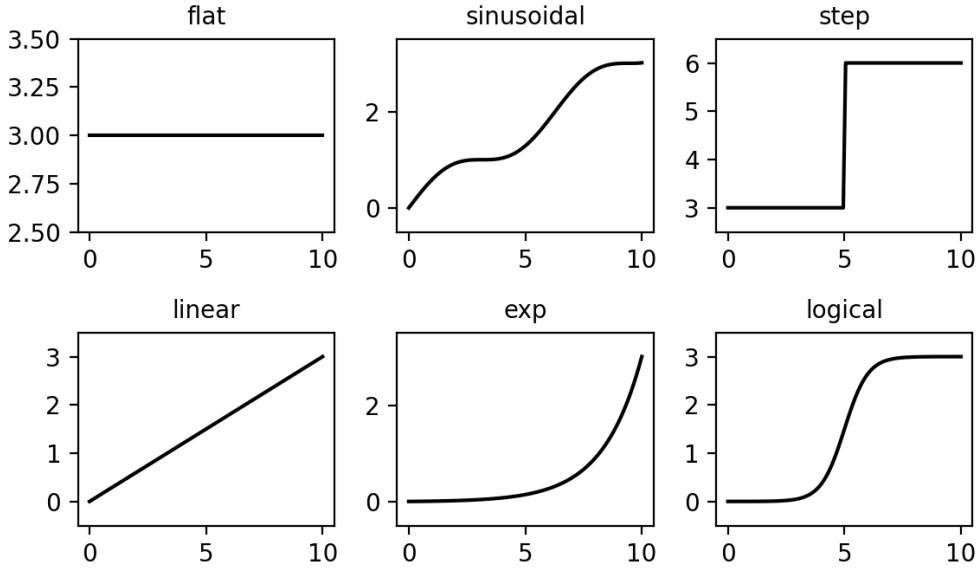


Figure 6.1: Monotonic benchmark functions.

For each benchmark function, we generate data by first sampling N x-points from a uniform distribution over $[0, 10]$ and then sample $y_n = f(x_n) + \epsilon_n$ where $\epsilon \sim \mathcal{N}(0, 1)$. Throughout this experiment, we will always consider 15 training data points and 100 test data points.

6.1.1 Model selection

For the model selection, we run each model configuration for 20 datasets generated with new random seeds.

For the HS model, we perform a grid search over all combinations of the parameters $m \in \{2, 3, 5, 10, 20\}$ and $L \in \{10, 20, 30\}$. For the VP model we choose between $M \in \{5, 10, 20, 50\}$. For the SDE model perform a grid search over all combinations of the

parameters $M \in \{20, 40\}$, $T \in \{1, 3, 5\}$ and two kernels: the squared exponential kernel and the Matérn kernel with $\nu = \frac{3}{2}$.

We report the best model configuration in table 6.1 for each model and benchmark function, that is the configuration with the lowest average RMSE on the 20 runs.

Function	HS model	VP model	SDE model
flat	m: 2, L: 30	virtual points: 50	M: 20, T: 5, kernel: Matérn
sinusoidal	m: 10, L: 10	virtual points: 50	M: 40, T: 5, kernel: Matérn
step	m: 10, L: 10	virtual points: 10	M: 20, T: 5, kernel: squared exponential
linear	m: 10, L: 10	virtual points: 50	M: 40, T: 5, kernel: Matérn
exp	m: 3, L: 20	virtual points: 50	M: 40, T: 5, kernel: Matérn
logical	m: 3, L: 10	virtual points: 50	M: 20, T: 3, kernel: Matérn

Table 6.1: Results of the model selection procedure described in section 6.1.1 for the three models across the six different benchmark functions.

6.1.2 Results

Now we run the experiment again for the best model configurations from table 6.1 and the baseline GP with 20 new random seeds for the training and test data. In table 6.2 and 6.3 you can see the resulting metrics where we take the mean and standard deviation of the RMSE and ELPD for each model across the 20 runs of the experiment for all six benchmark functions. We furthermore present the average ranking for each model across the benchmark functions.

Function	baseline GP		HS model		VP model		SDE model	
flat	1.176	± 0.028	**1.046**	± 0.021	1.092	± 0.021	1.404	± 0.057
sinusoidal	1.239	± 0.036	1.154	± 0.024	**1.087**	± 0.016	1.111	± 0.026
step	1.340	± 0.038	**1.230**	± 0.021	1.319	± 0.024	1.931	± 0.059
linear	1.135	± 0.026	1.112	± 0.015	**1.072**	± 0.02	1.081	± 0.022
exp	1.150	± 0.02	1.160	± 0.02	**1.123**	± 0.013	1.141	± 0.018
logical	1.162	± 0.04	**1.119**	± 0.027	1.137	± 0.025	1.136	± 0.02
mean rank	**3.5**		**2.17**		**1.67**		**2.67**	

Table 6.2: RMSE of experiment 1. The RMSE is given by equation (6.7) and the confidence intervals are given by the std of the sample mean from equation 6.10. The bold font marks the best model on a given benchmark function. The mean rank is the average rank in performance for each model for all six benchmark functions.

Based on the results from this experiment we can see that our model performs competitively with the other models in the experiment. In fact, it has the lowest average in terms of RMSE and ELPD for half of the benchmark functions, as seen in table 6.2 and table 6.3, and it is significantly better than the other models for the step function and the flat function. For the rest of the benchmark functions the VP model has the best average performance. Both the HS and VP model performs better than the baseline GP overall, and the HS is significantly better than the baseline GP with no overlapping confidence intervals on the three first benchmarks functions.

A general observation on the plots in figure 6.2 is that the amount of noise makes it difficult for any of the models to capture the subtle nuances of the benchmark functions, and in general the posterior means are more linear and smooth than the function from which the data is generated.

Function	baseline GP		HS model		VP model		SDE model	
flat	-1.646	± 0.039	**-1.516**	± 0.026	-1.528	± 0.021	-1.829	± 0.037
sinusoidal	-1.654	± 0.024	-1.600	± 0.023	**-1.547**	± 0.026	-1.801	± 0.031
step	-1.741	± 0.036	**-1.652**	± 0.019	-1.735	± 0.022	-2.113	± 0.03
linear	-1.579	± 0.018	-1.550	± 0.015	**-1.543**	± 0.031	-1.733	± 0.028
exp	-1.587	± 0.014	-1.583	± 0.016	**-1.553**	± 0.012	-1.797	± 0.021
logical	-1.610	± 0.031	**-1.565**	± 0.025	-1.585	± 0.023	-1.645	± 0.019
mean rank	**3.0**		**1.5**		**1.5**		**4.0**	

Table 6.3: ELPD of experiment 1. The ELPD is given by equation (6.7) and the confidence intervals are given by the std of the sample mean from equation 6.10. The bold font marks the best model on a given benchmark function. The mean rank is the average rank in performance for each model for all six benchmark functions.

In the plot of the baseline GP in figure 6.2 we can see that the samples are not guaranteed to be monotonic. Especially when the GP is fitted on the sinusoidal and step datasets, you can see that it has even failed to capture a monotonic posterior mean. This is a consequence of the rather low amount of data and the prior zero-mean.

The SDE model performs slightly better than the baseline GP and evenly with the HS and VP model for four of the functions in terms of the RMSE having overlapping confidence intervals. However, looking at the ELPD it performs significantly worse than both the VP and the HS model for all benchmark functions and significantly worse than the baseline GP on five out of six benchmark functions. When you look at the plot in figure 6.2, you can see how it is prone to overestimating the noise variance, e.g. for the linear benchmark function in row 4. In general the SDE model has been unstable in performance, which can be seen in the plot in appendix A.5.1, however, when it works it performs competitively with the other models.

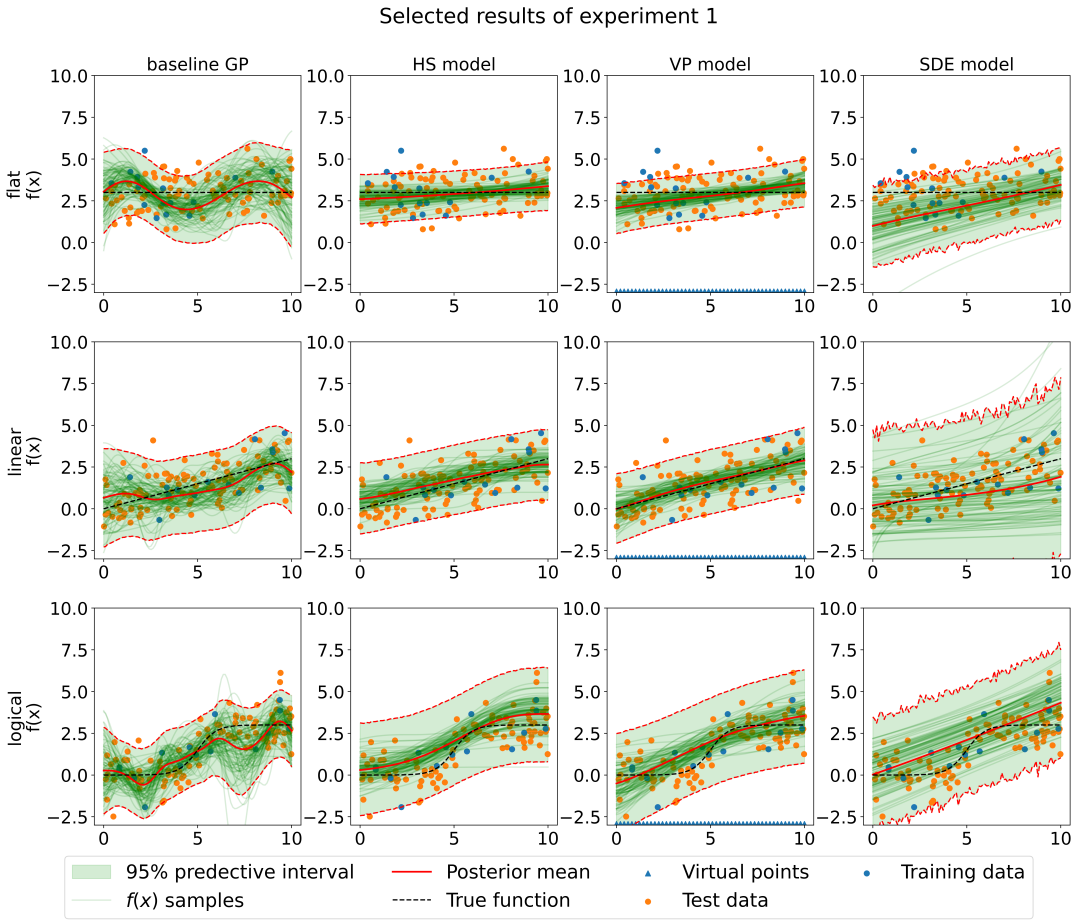


Figure 6.2: Plot of experiment 1. The plots show the posterior predictive distribution of each model fitted on the flat, sinusoidal and step functions from one of the experiment runs. The posterior predictive of the remaining benchmark functions can be seen in A.5.1.

6.2 Experiment 2: U-shaped benchmark functions

In this experiment we will evaluate the performance of the u-shaped HS model, the u-shaped VP model, the HS relaxed model and the baseline GP on six different u-shaped benchmark functions. This experiment serves the same role as experiment 1 in 6.1 in creating a fair and just comparison of the different model on a variety of u-shaped functions in a controlled experimental environment. The u-shaped benchmark functions are given as

$$f_i(x) = \begin{cases} 2 & i = 1 \quad \text{(flat)} \\ 0.1(x-2)^2 & i = 2 \quad \text{(skew)} \\ 0.25x^2 & i = 3 \quad \text{(parabola)} \\ |x| & i = 4 \quad \text{(absolute value)} \\ -2\left(\sin\left(\frac{x+5}{5}\pi - \pi/2\right)\right) + 2 & i = 5 \quad \text{(sine)} \\ 4 - 4 \cdot \mathbb{1}_{[-3,3](x)} & i = 6 \quad \text{(step)} \end{cases} \tag{6.20}$$

Be aware that only the first four of these benchmark functions are actually convex functions. The last two are unimodal functions that have been flipped upside down. We have included these functions to evaluate the consequences of imposing to strict constraints on the data. It also serves to evaluate the performance of the HS relaxed model.

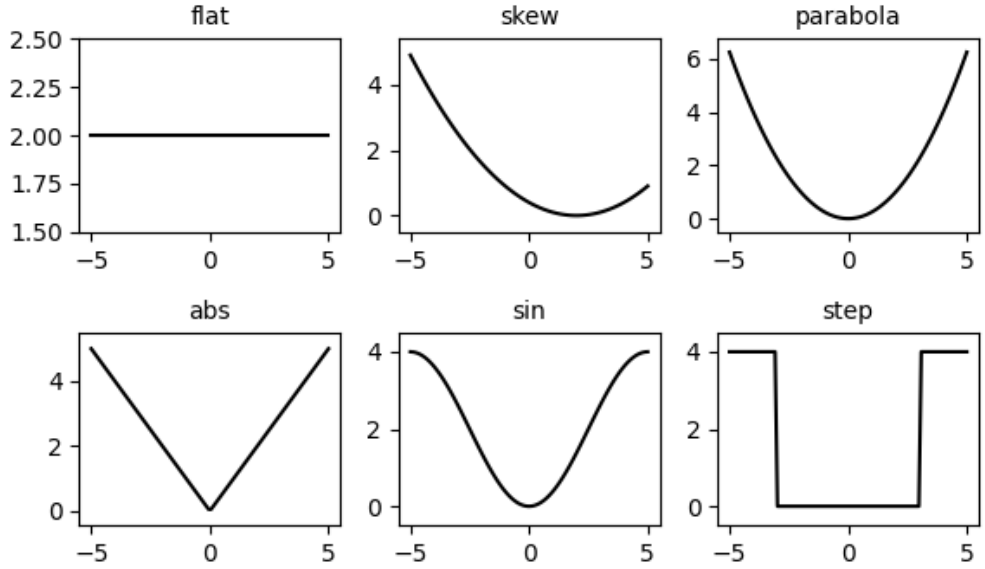


Figure 6.3: U-shaped benchmark functions

Model selection
For the HS model, we perform a grid search over all combinations of $m \in \{2, 3, 5, 10, 20\}$ and $L \in \{10, 20, 30\}$. For the u-shaped HS relaxed model we perform a grid search over the same values for m and L. For the VP model we choose $N_{vp} \in \{5, 10, 20, 50\}$.

The model parameters for each of the three models across the benchmark functions can be seen in table 6.4.

As described in 5.3.3 we impose a Student's t distribution as prior on the intercept term F_0. We find the mean parameter, μ_{F_0}, using the heuristic from section 5.3.3: Fit a straight line l to the first half of the data using linear regression. Compute $l(-L)$ for the different values of L and use this as the μ_{F_0} parameter.

Benchmark function	HS model	HS relaxed model	VP model
flat	m: 2, L: 20	m: 5, L: 30	virtual points: 10
skew	m: 5, L: 10	m: 5, L: 20	virtual points: 50
parabola	m: 5, L: 10	m: 5, L: 20	virtual points: 5
abs	m: 5, L: 10	m: 20, L: 20	virtual points: 5
sine	m: 10, L: 10	m: 20, L: 20	virtual points: 5
step	m: 2, L: 10	m: 10, L: 20	virtual points: 5

Table 6.4: Results of the model selection procedure described in section 6.2 for the three models across the six different benchmark functions.

6.2.1 Results

In table 6.4 we are presented the results of the model selection. We notice that for the convex HS model, the model selection favours relatively few basis functions compared to the relaxed model, which uses up to 20 basis functions. This indicates that the two models need to be configured in different ways even though the models in construction are quite similar. In the VP model, we only select five virtual points for most of the benchmark functions. This means that the shape constraints are not really enforced in the VP model apart from on the flat and skewed benchmark functions. Choosing the number of virtual

points purely based on performance may not be the best idea if you want to preserve the shape-constraint. We will discuss this later in this section.

The general picture for the results in table 6.5 and 6.6 is that the HS relaxed model and the VP model outperform the baseline GP and HS models in terms of mean RMSE and ELPD. Even though the HS model has the lowest mean rank for both ELPD and RMSE it still has overlapping confidence intervals compared to the best model for five out of six of the datasets in terms of RMSE and four out of six in terms of ELPD. This suggests that there is no significant difference in performance for these cases. Viewing it this way our model performs equally with the baseline GP, as the GP performs significantly worse than the best model in terms of RMSE of the parabola function. The GP also performs significantly worse than the best model in terms of ELPD on the flat and the parabola function. The overall picture of the HS model is that it performs competitively with the models in comparison.

Function	baseline GP		HS model		HS relaxed model		VP model	
flat	1.171	± 0.026	1.107	± 0.032	1.113	± 0.031	**1.059**	± 0.021
skew	1.240	± 0.038	1.227	± 0.045	1.192	± 0.04	**1.180**	± 0.037
parabola	1.516	± 0.071	1.193	± 0.032	**1.127**	± 0.022	1.601	± 0.062
abs	1.309	± 0.051	1.392	± 0.056	**1.294**	± 0.056	1.383	± 0.029
sine	1.256	± 0.039	1.316	± 0.046	**1.252**	± 0.047	1.297	± 0.033
step	1.545	± 0.057	1.625	± 0.047	1.582	± 0.043	**1.495**	± 0.049
mean rank	**2.83**		**3.17**		**1.83**		**2.17**	

Table 6.5: RMSE of experiment 2. The RMSE is given by equation (6.7) and the confidence intervals are given by the std of the sample mean from equation 6.10. The bold font marks the best model on a given benchmark function. The mean rank is the average rank in performance for each model for all six benchmark functions.

Function	baseline GP		HS model		HS relaxed model		VP model	
flat	-1.649	± 0.04	-1.561	± 0.031	-1.563	± 0.025	**-1.525**	± 0.026
skew	-1.655	± 0.026	-1.644	± 0.034	-1.622	± 0.03	**-1.610**	± 0.03
parabola	-1.820	± 0.048	-1.632	± 0.03	**-1.580**	± 0.028	-1.926	± 0.051
abs	**-1.678**	± 0.026	-1.804	± 0.07	-1.694	± 0.044	-1.740	± 0.019
sine	-1.669	± 0.027	-1.709	± 0.036	**-1.667**	± 0.035	-1.703	± 0.025
step	-1.843	± 0.032	-1.920	± 0.029	-1.890	± 0.022	**-1.830**	± 0.033
mean rank	**2.67**		**3.17**		**2.0**		**2.17**	

Table 6.6: ELPD of experiment 2. The ELPD is given by equation (6.7) and the confidence intervals are given by the std of the sample mean from equation 6.10. The bold font marks the best model on a given benchmark function. The mean rank is the average rank in performance for each model for all six benchmark functions.

In the initial trials of this experiment, we chose a zero mean prior on F_0 instead of imposing the linear regression heuristic. This resulted in a too-high noise variance and too-flat samples for most of the benchmark functions. The results in this section show that the regression heuristic has provided some relevant prior information to the model. However, there is an unexplored potential here in finding even better ways of configuring the prior distribution on F_0. In figure 6.5, we have imposed the Student's t-distribution with mean= 15 and trained the HS model on a single dataset from the *sine* and the *step* benchmark function for $m = 5$ and $m = 20$ basis functions. Even though these examples have not

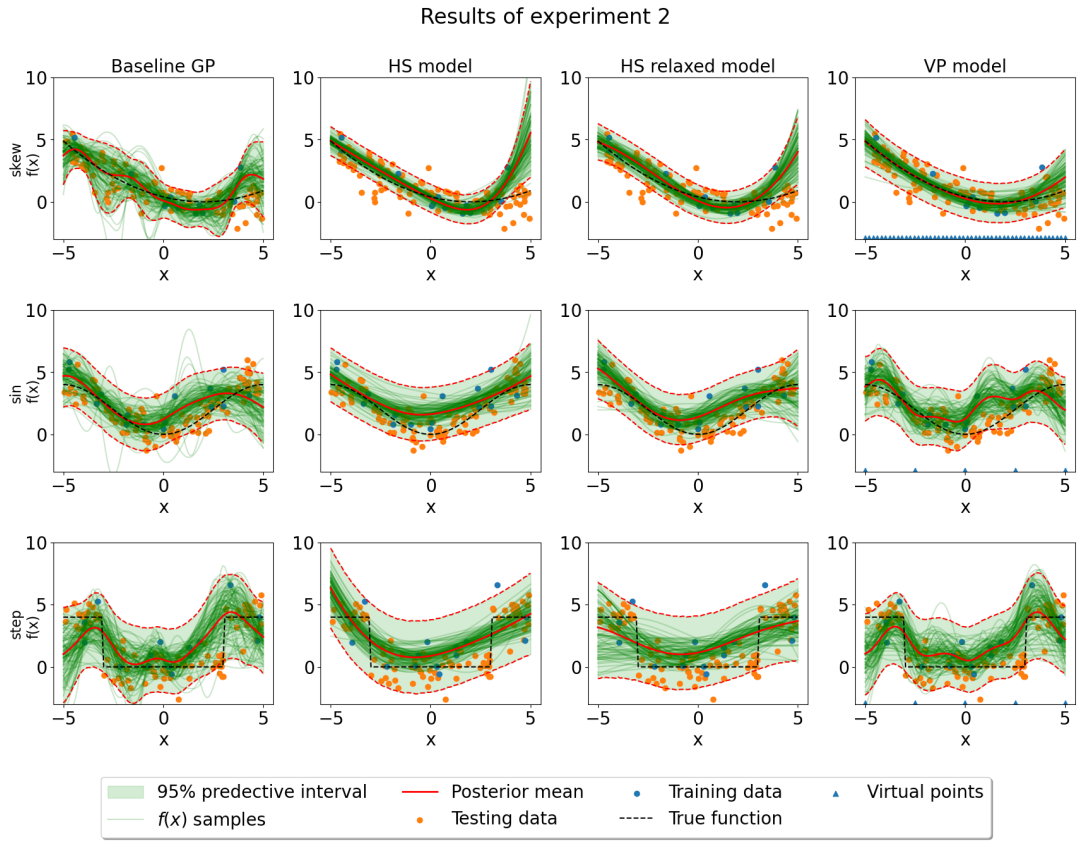


Figure 6.4: Plot of experiment 2. The plots show the posterior predictive distribution of each model fitted on the six benchmark functions from one of the experiment runs. A full figure of all the models and benchmark functions can be found in appendix A.5.2

been run with the same number of runs, chains and iterations as in the experiment in this chapter, making the results less reliable, we still see that the RMSE and ELPD samples are significantly better than the ones in table 6.5 and 6.6.

When we look at the plots in figure 6.4, we can see that both the VP model and the HS relaxed model are having non-convex samples on some of the benchmark functions. For the HS relaxed model, this is expected, however the VP model is not behaving as intended, which probably is due to the low number of virtual points chosen in the model selection. When we look at the VP plots of the last four benchmark functions in figure 6.4, it seems that the samples are too convex in the virtual points. An interpretation could be that the convexity demand is to high in the virtual points, and it simply recovers in the space between the points. An idea could be to train the model on a higher values of ν, as this will loosen the convexity demand in the virtual points. It seems, however, that the convexity demand is well calibrated in the skew function where the number of virtual points is 50. Even though the VP model in some cases fails to preserve the u-shape constraint in this experiment, it still performs better than the baseline GP. So it seems that there still is something to win in terms of a small number of virtual points.

For the HS relaxed model the posterior samples and the posterior means seems to capture the u-shaped form and provides quite good fits on the benchmark functions. Based on the results from this experiment it seems to have worked in providing some flexibility to the convexity constraint. As in the discussion in section 5.3.4, we will also investigate the contributions from respectively the GP term and the convex term of the model by plotting

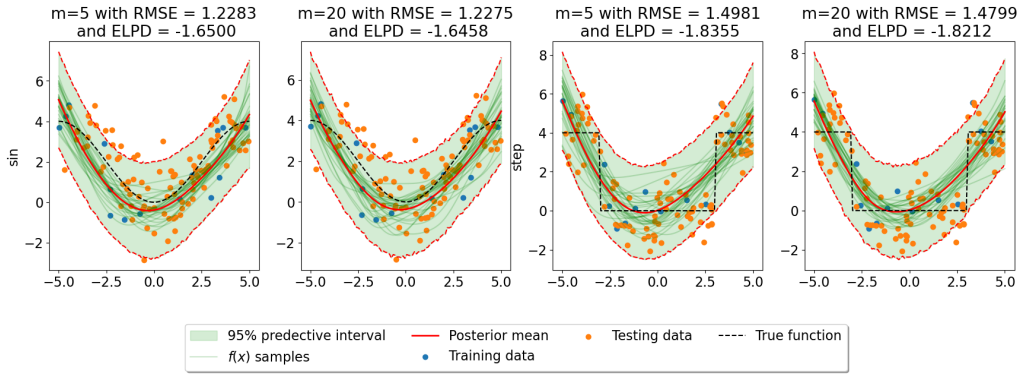


Figure 6.5: Samples from the HS model trained on one dataset with $m = 5$ and $m = 20$ and where $F_0 \sim \text{t}(15, \sigma_{F_0}, 4)$. The results in the plot are only calculated on a single dataset.

the variance from each terms. This can be seen in figure 6.6. It is clear to see that the contributions from the GP term and the convex term are even, so it seems to be well balanced in terms of preserving some of the u-shaped constraint but also adding flexibility from the GP. We have not given the prior mean of the intercept the same attention as with the HS model. Based on the results here, we can conclude that the HS relaxed model is not as sensitive towards the prior distribution on the intercept F_0, since the model isn't globally convex, although it would be interesting to investigate if imposing a similar heuristic on the HS relaxed model could improve the performance.

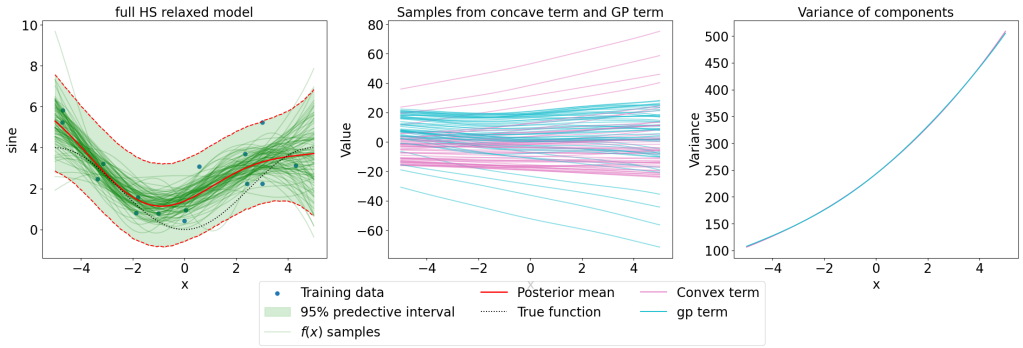


Figure 6.6: Visualisation on the contribution from each of the two terms in the HS relaxed model for the step function dataset. Visualisation of the variance of the two components of the HS relaxed for the rest of the benchmark functions can be found in A.3.

6.3 Experiment 3: Monotonic real data

In this experiment, we test the monotonic model on a dataset from the World Banks World Development Indicators (World Bank Group DataBank 2024). The dataset consists of the global fertility rate (births per woman) from 1963 to 2022, which has been strictly monotonically decreasing throughout that period. We use the data to test how the monotonic models perform at a forecasting task. The data has been normalized.

We use the years 1963-2010 as training data and 2011-2022 as test data. Furthermore, we select the model parameters by using 5-fold cross-validation in the following manner:

1. Use 36 data points from 1963-1998 as training data and validate on 6 data points from 1999-2004

2. Move this split two years into the future, i.e. train on 1965-2000 and validate on 2001-2006

3. Repeat step 2 4 times in total. The final training set will be 1971-2006, and the final validation set will be 2007-2010.

Model selection
For the HS model, we perform a grid search over all combinations of the parameters $m \in \{2, 3, 5, 10, 20\}$ and $L \in \{10, 20, 30\}$. For the VP model choose $N_{vp} \in \{5, 10, 20, 50\}$. For the SDE model perform a grid search over all combinations of $M \in \{20, 40\}$, $T \in \{1, 2, 3, 4, 5\}$ and two kernels: the squared exponential kernel and the Matérn kernel with $\nu = \frac{3}{2}$.

6.3.1 Results

The results can be seen in table 6.7.

	baseline GP	HS model m: 5, L: 10	VP model virtual points: 5	SDE model M: 20, T: 2, kernel: squared exponential
RMSE	0.382	0.125	0.593	**0.069**
RMSE rank	**3.0**	**2.0**	**4.0**	**1.0**
ELPD	-0.175	0.694	0.046	**0.855**
ELPD rank	**4.0**	**2.0**	**3.0**	**1.0**

Table 6.7: Model selection, RMSE and ELPD of experiment 3. Results of the model selection procedure described in section 6.3 for the three models i described underneath the model name. The RMSE and ELPD is given by equation (6.7). The bold font marks the best model with respect to the RMSE and the ELPD.

Overall, the SDE performs best in the forecasting task, followed by the HS model. The baseline model comes in third when evaluating the RMSE, but is beaten by the VP model when measuring performance by the ELPD. Inspecting figure 6.7, we can explain this by comparing the means and sample paths of the GP prior model and VP model. Although the mean of the baseline model is not monotonic, the mean of the VP model decreases too fast, thus making a worse prediction of the test data and causing it to underpeform measured by RMSE. However, since the sample paths follow the monotonicity constraints, it performes better when it comes to the ELPD.

The first evident thing is that the choice of the zero-mean Gaussian process for the baseline model is questionable for this particular dataset, as it shows a clear decreasing trend and thus is not stationary. Incorporating a linear mean function or adding several kernels with different length scales as covariance function would be more appropriate to capture the decreasing trend as well as the smaller fluctuations in the data.

As for the VP model, it undershoots on the test data, which only has a slight descent. Even though this 'flattening' of the curve is also present in the training data, the VP model predicts a much steeper descent. We hypothesize that the virtual points, M, could cause this. The model selection has chosen the lowest number of virtual points. To test our hypothesis, we run the VP model for $M = 1, 2, 3, 4$ to see if this will change the steepness of the extrapolation. The posterior predictive distributions are plotted in figure 6.8, where the RMSE of the fits are also reported.

This confirms our suspicions that the virtual points cause the VP model to 'undershoot'. The best performance is obtained at only 2 virtual points; however, the 95% confidence

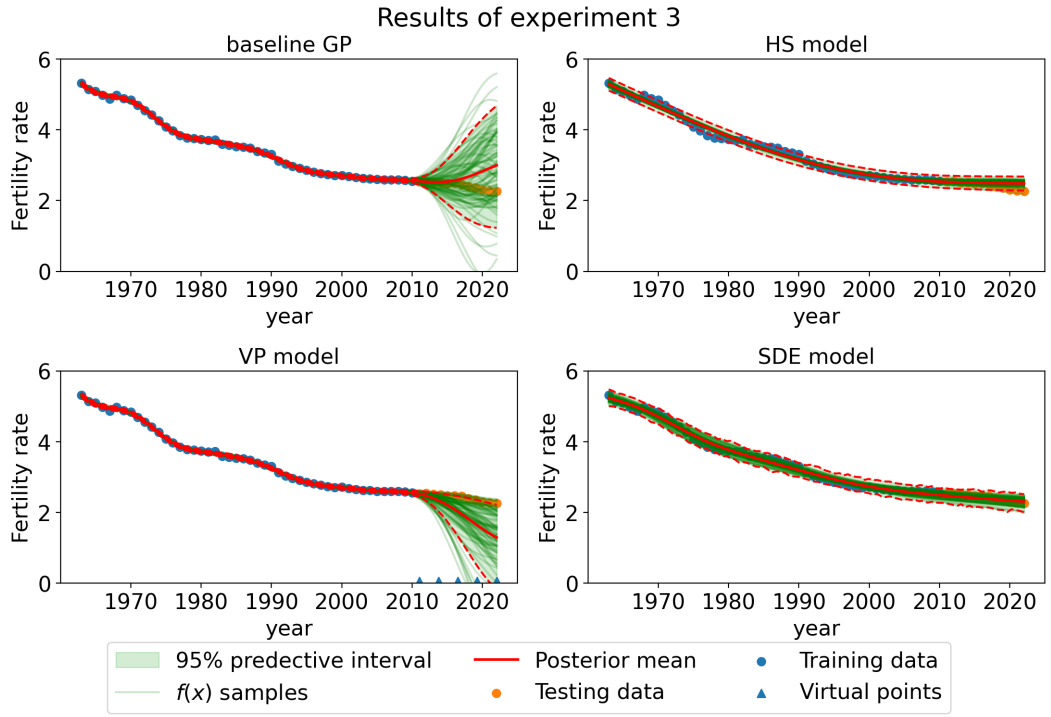


Figure 6.7: The result of experiment 3. We see that both the HS model and the SDE model tend to oversmooth. The GP performs well on the training data, but fails when it comes to forecasting. The data has been normalised for training and scaled back to original values in this plot.

intervals show that this drastically influences the monotonicity of the samples.

Both the SDE and HS models have found smoother solutions with a larger noise variance. This is acceptable in for extrapolation tasks where the expected value is used for prediction. However, these fits will be a poor choice for interpolation as the two models don't capture the small fluctuations in the data. Both models have surprisingly low variance outside of the training data, and the HS model in particular. Overall, the f samples from the HS model have a very low variance. In Ustyuzhaninov et al. (2020), it is hypothesized that the low model uncertainty of the HS model is caused by the restricted domain, $[-L, L]$. However, in this case the normalized data is in $[-2, 2]$, and the length of the domain is $[-10, 10]$. In figure 6.9, we have extended the f samples to a (normalized) domain $[-12, 12]$. Here, we see that the model eventually returns to a larger variance but that it happens rather slowly. In this case, it seems more likely that the number of basis functions restricts how quickly the model recovers rather than the length of the domain.

A general comment on our model selection is that this specific dataset is a calculated mean over a large population and, therefore, hardly exhibits any noise. A further investigation could be to model the problem without any noise and, by that, remove the risk that the SDE and HS model converge to a parameter mode with a too-high noise variance.

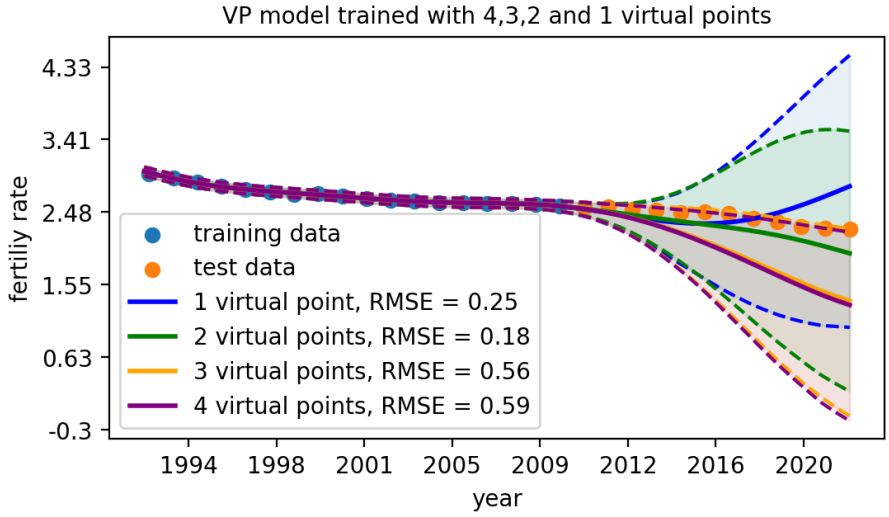


Figure 6.8: Posterior distribution obtained by training the VP model for $M = 1, 2, 3, 4$ on the World Bank dataset.

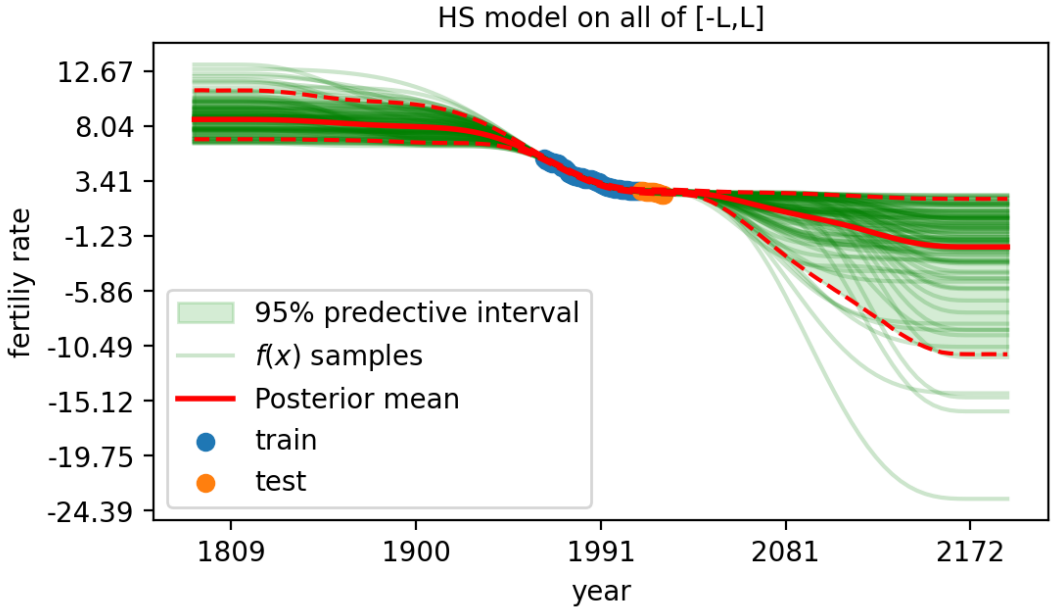


Figure 6.9: Results of extrapolating the monotonic HS model to the entire domain from -L to L.

6.4 Experiment 4: U-shaped real data

In this experiment, we will test the convex Hilbert space model on a real dataset and compare it to a baseline GP regression model and the VP model. The data is from Blanchflower and Oswald (2008) and measures the probability of depression as a function of age. In the article Blanchflower and Oswald (ibid.), it is shown that there is a u-shaped (concave) relationship between age and the probability of depression. In this experiment, we normalised the data.

We attempt to investigate the performance of the models as a function of the size of the training dataset in order to assess the benefit of the shape constraints.

In order to do this, we make 7 different training runs by making 7 random splits of the dataset into training and test data. We use 42 data points for training (75%) and 13 for testing (25%). For each run, we fit the model to 7 subsets of the training data containing $2, 3, 4, 5, 10$ and 21 data points, respectively. We use 5-fold cross-validation to select the model parameters. However, in order to do this for the very small datasets, we use a full validation set for all subsets. This is illustrated in figure 6.10. Finally, we evaluate the performance on the test set for each run.

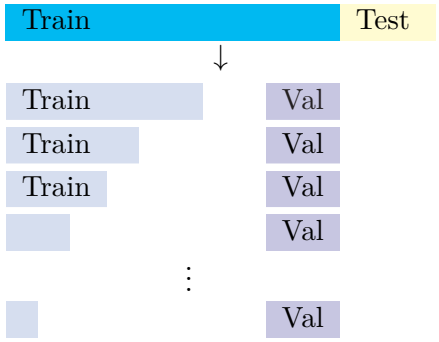


Figure 6.10: Illustration of a single fold of a single run. The full data set is shuffled and split into a training- and test dataset. Afterwards, the training data is split into a validation dataset and a sequence of sub-training sets.

Model selection

For the HS model we choose between $m \in \{2, 3, 5, 10, 20\}$ and $L \in \{2, 3, 5\}$. For the VP model, we loop through the number of virtual points, $M \in \{5, 10, 20, 50\}$.

Remark: For this problem, we perform a distinct model selection for each run of the data. We have chosen this approach because the random selection of the training and test data finds the model configuration which is suited to the distinct split of the data and not necessarily across all the different splits.

6.4.1 Results

The RMSE and ELPD have been plotted as a function of the number of data points in 6.11. Based on these 7 runs, it is difficult to say something definite due to the high standard deviation, and the models appear to have equivalent performance. If we measure performance on the RMSE, we see that although it is only the best model on one of the 6 training sets, it has the highest ranking overall. The HS model comes second, even though it has the worst performance on the very small datasets. Although the VP model comes in last, it is the least sensitive to the size of the dataset overall. When measuring performance based on ELPD, we see a similar picture, where the VP model has steady performance overall, but is outperformed by the HS model on the larger datasets.

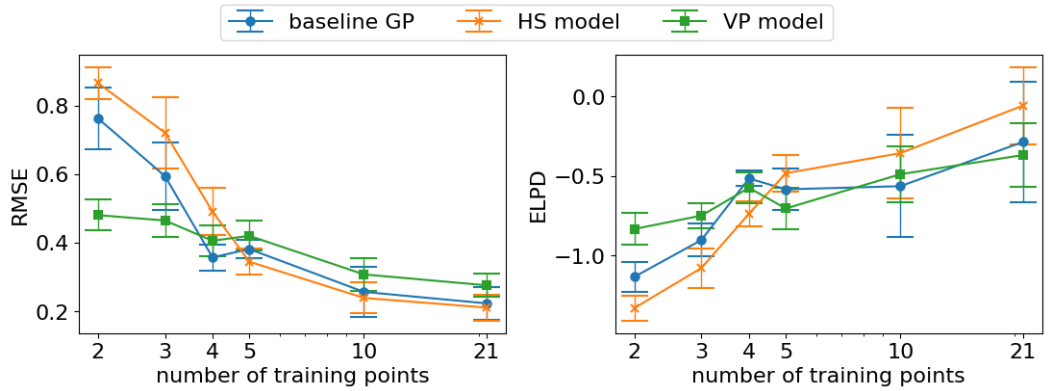


Figure 6.11: A plot of the RMSE and ELPD, respectively, as a function of the number of training points. Note that as the error bars indicate, it is hard to tell if there is any significant difference between the models for all but the two smallest datasets. However, the plots give an overall idea of the performance.

In order to evaluate the performance and diagnose why the HS model performs so badly on the small datasets, we have illustrated the posterior distributions of the three models in 6.12. We have plotted the distributions for models trained on 4 and 10 data points.

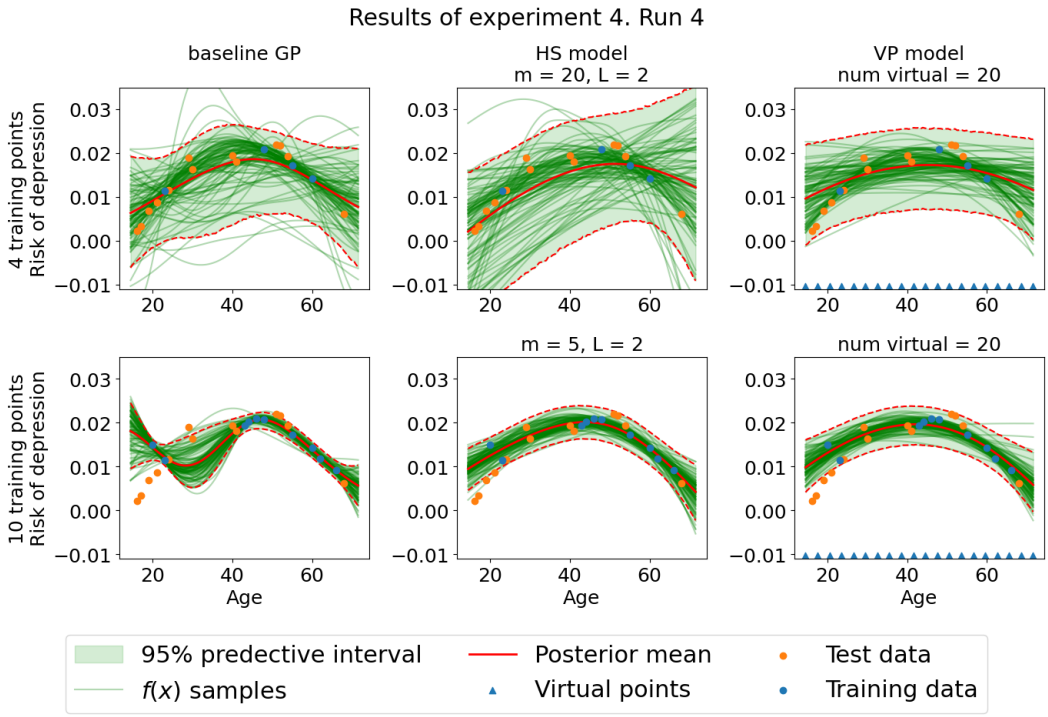


Figure 6.12: Results on experiment 4. The plots illustrate the posterior distribution of the baseline GP, HS model and VP model trained on 4 and 10 training points, respectively.

In the figure, we see that when there is extremely little data available, the posterior resembles the prior distribution more closely. Since the prior variance scales exponentially as x moves further away from $-L$, the posterior does the same and thus 'overshoots' the data by fitting F's with a mode outside of $[-L, L]$. These functions essentially become monotonic.

The potential benefit of the shape constraints is also visible in figure 6.12. In the case with 10 training points, we see how sensitive the baseline model is to the placements of the data points. In figure 6.11, we also see an indication that imposing shape constraints is a good idea. Overall, the largest datasets in the experiment are still rather small, so, in general, it seems that when calibrated correctly, there is a benefit of incorporating shape constraints into the model in order to avoid over fitting as we see, the GP is prone to in figure 6.12. However, it is only for the extremely small datasets that we see a significant difference between a shape-constrained model (the VP model) and the baseline model. As such, we can only conclude that the benefit of adding the shape constraints is more robustness in the model rather than significantly better prediction precision.

If we where to broaden out the investigation of data efficiency of shape constrained models, it would be beneficial to repeat experiment with more runs on different datasets and include larger sizes of datasets as well. It would be informative to train the models on the data with different levels of noise variance, as the averaged data from Blanchflower and Oswald (2008) we used in experiment 4 is rather smooth. Our suggestion is that when fitting to data with more noise variance, the baseline GP will be more prone to overfitting, as we have seen in experiment 1 and 2, and that the shape constrained models would be a better choice for such data.

7 Discussion

In this chapter, we will discuss some of the themes and trends across the former sections. We will give a summary of the learnings from the experiments in chapter 6 and relate these to the results from chapter 5 and the research questions in the introduction. We will discuss the HMC sampling method applied in the experiments section, and lastly we will mention several topics for further investigation.

7.1 Discussion on shape-constrained HS model and comparison with existing methods

In general the experiments show that the monotonic and the u-shaped HS models perform competitively with the other models in terms of both the ELPD and RMSE. We have thereby shown that it is possible to utilise the HS approximation from section 3 in order to enforce global monotonic or u-shaped constraints and still achieve good fits on shape-constrained data. In contrast we saw that the VP model generally performs well, but frequently fails to preserve the desired shape inherited from the data. In general it seems like the VP and HS model are equal in performance, but we want to emphasize some other aspects that favours the HS model. Firstly the HS model has strictly monotonic samples whereas the VP model doesn't. In this experiment the VP model seems to be quite good in preserving the monotonicity, but that is also a result of the model selection where the VP model ends up having 50 virtual points in five out of the six cases. This is expensive as the computation complexity is given by $\mathcal{O}((n+M)^3)$ for each optimisation step where n is the amount of training data and M is the number of virtual points (Riihimäki and Vehtari 2010). Therefore Riihimäki and Vehtari suggests that you should try to lower the number of virtual points as much as possible, but doing so relaxes the guarantee of monotonicity. The HS model, on the other hand, only requires an initial operation of $\mathcal{O}(nm^2)$ and for each iteration in optimization it costs $\mathcal{O}(m^2)$ where m is the number of basis elements as discussed in 5.4.2. The SDE-model is guaranteed to be monotonic by the model construction, and it has shown the potential to achieve good fits on par with the models in comparison. Based on the experiences from experiment 1 and 3, it has been more difficult to configure the model and to obtain appropriate fits on the data, as we see that it sometimes fails to capture the trend in the data, either because it is prone to overestimating the noise variance or because of unstable sampling.

In experiment 2 we also test the proposed relaxation of the u-shaped HS model from section 5.3.4. Throughout all six benchmark functions it seems to perform quite well, resulting in samples and posterior means which overall fits the u-shaped shape in the data and scores the best average RMSE in three out of six of the benchmark functions. Our u-shaped HS relaxed model, has been easier to configure in terms of finding a suitable prior on the intercept term, due to the flexibility of the model. However, it is twice as slow to run since we deal with double the amount of basis functions compared to the u-shaped HS model. Lastly, it is still difficult to control the balance between the GP term and the convex term, and even though the two terms seems to provide an even amount of variance as seen in A.3, we are not guaranteed that the GP term will not dominate the convex term.

7.1.1 Model selection

In all of the experiments, we have used a grid-search over the model parameters and picked the model with the lowest average RMSE.

For the monotonic HS model, we observe that lower values of L are preferred, with the exception of the flat function and the exponential function. This is in accordance with the observations made in the discussion of the effect of m and L in section 5.4.1. It seems that choosing an L as small as possible is generally a good idea for interpolation tasks, but one should be careful when doing extrapolation since the monotonic functions become constant outside the domain. Interestingly, it appears that a higher number of basis functions does not necessarily lead to a better fit in the monotonic model. Across all monotonic experiments, we never choose more than 5 basis functions.

We observe that the U-shaped HS model never selects more than 10 basis functions. We might expect that the smoothing from the double integration in the model construction, would create a need for a high number of basis functions, but this doesn't happen in practice. In general, the best L is the lowest L, apart from the flat benchmark function in experiment 2. This suggests that the model performs best when the model domain, $[-L, L]$, is very close to the data domain. If we were to enhance the model for bigger L's, one could investigate other methods for configuring the prior mean on the intercept, F_0, as we have shown it to be closely connected with the configuration on L. This is especially relevant for extrapolation tasks.

In the HS relaxed model, we see a different picture. Here, the model favours larger values of L and, in some cases, also m. This could be caused by the fact that we have chosen the same m and L for both the convex and the GP part. It is possible that the GP is 'dominating' so the model parameters are chosen to better approximate the GP component.

Across all four experiments, we experienced the VP models sensitivity to the placement of the virtual points. In experiment 1 and 4 the placement of the virtual points works well and the desired shape constraint is enforced. In experiment 3 we observed how sensitive the model is to the number of virtual points in extrapolation tasks, where adding a single virtual point contributed to a large change in the posterior mean of the samples. In experiment 2 we have observed that in interpolation it is also difficult to figure out how many virtual points to use, and that the model selection in some examples favours the model configuration where the shape constraint demand on convexity fails.

7.1.2 Did we succeed in modelling shape-constrained models for small datasets?

Overall, we set out to try to investigate the effect of incorporating shape constraints into regression models. In addition, we wanted to use the HS approximation to construct flexible, non-parametric models that are globally shape-constrained and analytically tractable. Based on the experiments, both the monotonic and the u-shaped models have shown promising properties. The monotonic HS model performed well in both interpolation and extrapolation tasks although we experienced some problems with low sample variance outside of the data. The u-shaped HS model performed competitively with the comparison models but also presented some more challenges overall. We believe that these challenges are caused by the double integration, which both smoothens out the flexibility of the underlying GP and causes the variance to grow exponentially.

While working with the u-shaped HS model, we also found that choosing an appropriate prior on F_0 is very important due to the model's convexity. In the initial trials of experiment 2, we only imposed a zero-mean prior on F_0, resulting in very flat posteriors across all datasets in the experiment. It is therefore advisable to choose the prior for F_0 based on the specific dataset. The heuristic described in section 5.3.3 has proven to be effective, providing the HS model with a fair chance of identifying suitable F_0 samples for the data fitting. Using this heuristic, we saw good performance in the experiments with more data

such as experiment 2 and the larger datasets in experiment 4. However, there is room for improvement, as seen in experiment 2 in figure 6.5, where we chose an even higher prior mean on F_0. Ideas for further investigation on this is discussed in section 7.3.

A big drawback of the u-shaped HS model is from the exponentially growing prior variance. This becomes evident for the very small datasets in experiment 4. Due to the small amount of information provided in the data, the posterior is very similar to the prior distribution and thus inherits the large variance. This is seen in figure 6.12. The prior variance also makes the model badly suited for extrapolation when predicting test points x_* where $x_* \gg \max(x_i)$ for $x_i \in X_{train}$. When we consider extrapolation on the left side of the data, we have the opposite problem. In this case, there is almost no variance inherent in the models. The main variance contribution comes from the prior distribution on the intercept term, and it is then merely a question of finding the prior on the intercept term. Therefore, the u-shaped model is not the most suitable model for extrapolation problems.

Another main topic we wanted to investigate was to what extent shape-constrained modelling can enhance the model precision on small datasets. In general, all experiments are based on rather small datasets. Some with a lot of noise and some on real data with a rather low noise level. Especially for the synthetic datasets, we see that our models perform better than the baseline GP on average. This is also confirmed by visual inspection of the fits. We put this question to the test in experiment 4 where we train the models on datasets consisting of only 4, 3 and 2 training points. In those extreme cases, the u-shaped HS model fails due to the exponentially growing prior variance. However, the VP model manages to keep a steady performance over all the small datasets, indicating that the addition of shape constraints adds to the robustness of the models.

As mentioned in the discussion section 6.4, one idea to broaden out the investigation of the interplay between shape constrain and data efficiency, is to perform experiment 4 with more runs in order to increase validity of the results, test the models on different datasets with different noise levels and explore the performance on even larger sizes of datasets than we have used in experiment 4.

7.2 Sampling and convergence

In the experiments in chapter 6, we have used Hamiltonian Monte Carlo sampling as a tool for obtaining samples from the posterior. However, even though the Markov chain theoretically will reach the stationary distribution, it is not guaranteed that it has within the sampled period. In this section we will take a step back and discuss whether the HMC algorithm has succeeded in providing reliable posterior samples. We will use the split-$\hat{R}$ diagnostic defined in 2.3 to discuss the convergence of the HS model across the 4 experiments.

In the case of the HS model, the convergence assessment is further complicated by the multimodality of the posterior caused by the α's. This introduces the risk that two chains have converged but to two different modes. Because of this, we initially compute the split-$\hat{R}$ for the individual chains. This means that for each run, we have 4 split-$\hat{R}$ for each parameter.

We have summarized the split-$\hat{R}$ values for the monotonic model in table 7.1 and the split $\hat{R}$ values for the u-shaped model in table 7.2. A parameter is considered converged if it has a split-$\hat{R}$ value below 1.01. A chain is considered converged if all parameters in the run have converged and a run is considered converged if all chains in the run are converged.

The tables clearly show that there are issues with convergence. The u-shaped HS model

Monotonic model			
data	converged parameters	converged chains	converged runs
flat	324/560	20/80	0/20
sinusoidal	1145/1200	60/80	7/20
step	1161/1200	68/80	12/20
linear	1100/1200	61/80	8/20
exp	581/640	52/80	3/20
logical	499/640	30/80	2/20
fertility	40/40	4/4	na

Table 7.1: Convergence in the monotonic Hilbert space model. A chain is considered converged if all parameters from that chain are converged. A run is considered converged if all chains for the run are converged.

U-shaped model			
data	converged parameters	converged chains	converged runs
flat	564/720	38/80	1/20
skew	894/960	54/80	5/20
parabola	829/960	43/80	2/20
abs	858/960	53/80	4/20
sine	1301/1360	65/80	8/20
step	546/720	16/80	0/20
depression $n_{train} = 2$	256/292	5/28	0/7
depression $n_{train} = 3$	254/332	3/28	0/7
depression $n_{train} = 4$	273/332	4/28	0/7
depression $n_{train} = 5$	225/292	5/28	0/7
depression $n_{train} = 10$	248/332	5/28	0/7
depression $n_{train} = 21$	178/260	6/28	1/7

Table 7.2: Convergence of the convex Hilbert space model. A chain is considered converged if all parameters from that chain are converged. A run is considered converged if all chains for the run are converged.

has particular challenges, as we experience problems with convergence in almost every run. We have slightly better convergence in the monotonic HS model. Notably, it appears that the 'success' of the convergence varies from problem to problem.

For instance, the data from the flat benchmark function appears very prone to convergence issues, whereas the data generated by the step function is successful in around half of the 20 runs, and the fertility data experiment appears very successful (although this is only based on a single experiment).

In order to investigate what causes the high split $\hat{R}$ values, we have made trace- and scatter plots of the parameters obtained from fitting the monotonic model to the 'flat' data. We find that the run with the highest split-$\hat{R}$ value is run 11. Figure 7.1 shows a chain from run 11 that has successfully converged and figure 7.2 shows a chain from run 11 that has failed to converge. Similar plots can be seen in appendix A.5.4 for the u-shaped model. We will focus on the plots from the monotonic model as they are easier to interpret due to the lower number of parameters. However, analogous observations can be made in the u-shaped model.

When looking in figure 7.1, we can see what appears to be the multimodality of the alphas,

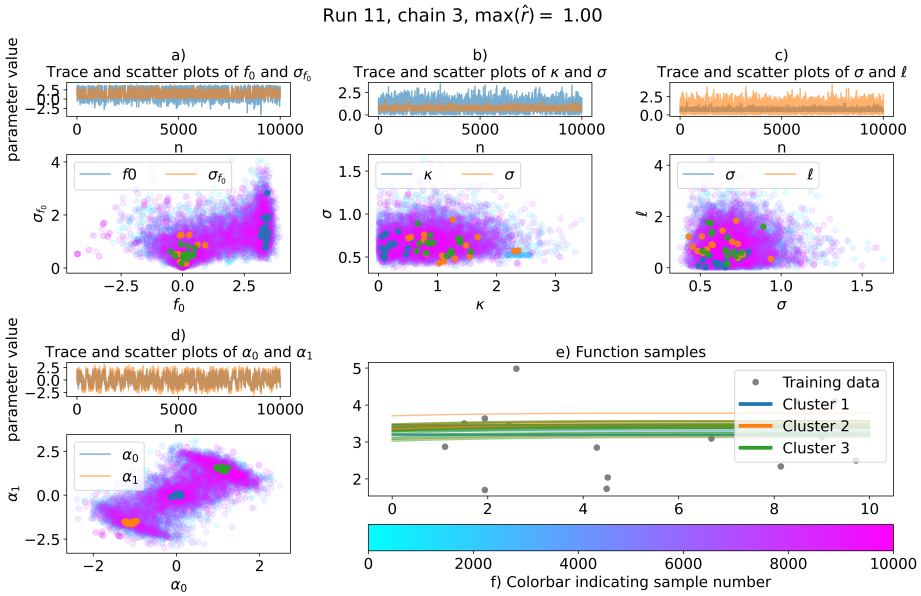


Figure 7.1: Convergence analysis plots of a successful chain. The converged chain is from training the monotonic HS model on the flat benchmark function. Plot a)-d): Trace and scatter plots of different pairs of parameters. Based on the α's, three 'clusters' of samples have been marked in blue, green and orange across the scatter plots. Plot e): posterior samples drawn from the three 'clusters' together with the training data. f) Colour bar used in the scatter plots. Light blue samples are sampled in the beginning of the chain, whereas pink samples stem from the end of the chain.

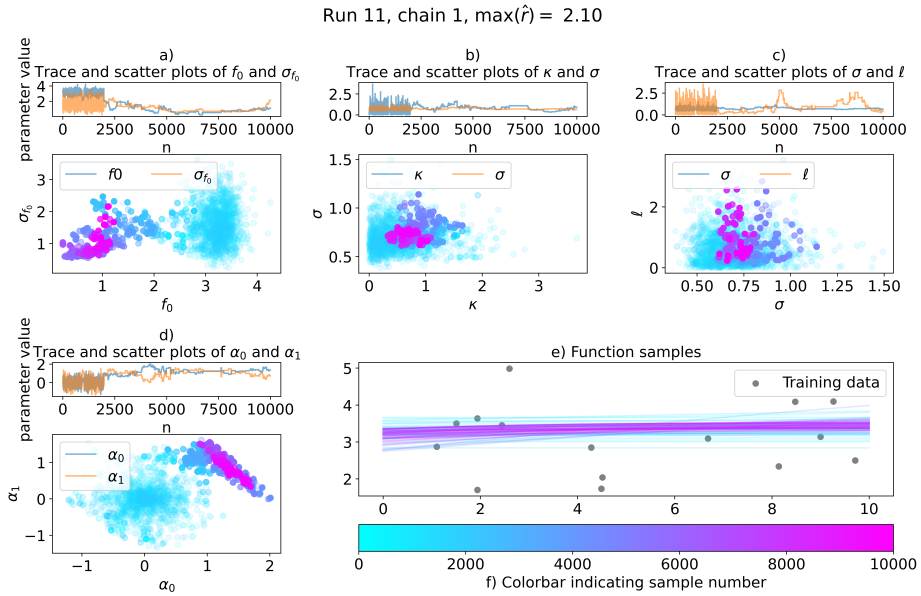


Figure 7.2: Convergence analysis plots of a failed chain. The chain is from training the monotonic HS model on the flat benchmark function. Plot a)-d): Trace and scatter plots of different pairs of parameters. Plot e): posterior samples together with the training data. f) Colour bar used in the plots. Light blue samples are sampled at the beginning of the chain, whereas pink samples stem from the end of the chain.

as there appears to be mode around $\alpha_0 = \alpha_1 = 0$ and two symmetric modes. The pattern could also be caused by the sampler not having investigated the parameter space. We coloured a few samples from clusters close to these modes throughout the plot. In figure 7.1 e), it is difficult to distinguish the posterior function samples across the different modes. However, in figure 7.1 a), it appears that the $\alpha = 0.0$ has a higher value of f_0, whereas the positive/negative α_0 modes have f_0 closer to zero. Thus, a function from cluster 2 essentially fits the flat function using the intercept, and a function from cluster 1 or 3 fits using the α values. Judging by figure 7.1 there are also differences in the kappa parameters.

In figure 7.2, we see clearly what makes the $\hat{R}$ value so large. Namely, it appears the HS model, has a tendency to get 'stuck' at certain values, where many of the proposals get rejected and the samples get very correlated. However, it is also possible that the chain has still not explored the parameter space sufficiently and would eventually converge to the true posterior. If this is the case, the 'multimodality' seen in figure 7.1 could simply be a product of the fact that the HMC algorithm has not explored the parameter space sufficiently yet. All in all, it is difficult to tell the difference between the posterior samples in figure 7.1 and 7.2.

Regardless, we should be aware the convergence issues and the indication that there might be issues with the model. As such, the results may not represent the model fully. A suggestion for a better sampling scheme could be to use an optimizer to find a mode in the distribution and initialize the sampler from there.

Another explanation for the convergence issues might be related to the number and placement of data points, which will contribute to how well-defined the posterior is. In the case of the 'flat' data in experiments 1 and 2, we only have 15 data points and quite a lot of noise, which results in an ill-defined posterior 'landscape'. In contrast, the extrapolation experiment is done on a much larger dataset with very little noise, which generally results in a more well-defined posterior landscape. This underlines the point that when dealing with data scarcity, choosing good priors is key.

7.3 Topics for further investigation

In this section, we present subjects and ideas for further investigation.

In section 7.1 we argue that the u-shaped HS model is not the best choice for extrapolation problems as the prior variance grows exponentially when the input of the model, x, grows. Therefore it would be interesting to see if there is a way of controlling the prior variance. One idea is to utilise that the prior variance is analytically tractable for the squared exponential kernel in section 5.3. Thus, we can use the inverse of the prior variance as a normalizing constant to try and reduce the posterior variance. It would be interesting to see if this could improve the performance in extrapolation problems.

It could also be worthwhile to find methods for deciding the prior on the intercept parameter, f_0 (monotonic) and F_0 (u-shaped). This is especially interesting for the u-shaped model, as it has turned out to be a rather crucial step in the model design. So far we have proposed to find a mean parameter on the prior distribution using linear regression on a relevant part of the u-shaped data. One expansion of this idea could be to do polynomial regression instead of linear regression. Another idea could be to investigate other priors than the normal distribution or Student's t as we have done in this work.

In our work, we have tried to investigate the role of m and L both for the Hilbert space approximation in general and for the shape-constrained models. However, it would be

insightful to investigate even further how the relationship between m and L affects the behaviour of the monotonic and u-shaped model, and especially, how fast the flexibility of the model increases when m grows. In this investigation, it might also be relevant to investigate other stationary kernels, e.g. from the Matérn kernel family.

Lastly, it could be interesting to investigate if there is a way to calibrate the relationship between the GP part and the convex part for the HS relaxed model. One idea could be to impose different priors on the two magnitude kernel hyperparameters. So far we have only considered the same number of basis functions for respectively the convex part and the GP part. It would be interesting to explore what happens if we choose different numbers of basis functions m and domain lengths L for each of the components. If we aim for a smooth convex trend in the model while allowing sufficient flexibility for small non-convex fluctuations, it might be effective to use a low number of basis functions for the convex component and a higher number for the GP component. We would still face the problem of controlling the magnitude of the contributions from each term. However, we might be able to control the flexibility contribution from each of the model components.

8 Conclusion

In this work, we have derived the m-rank Hilbert space approximation of a Gaussian process. We have shown convergence to a full Gaussian process for an unlimited amount of basis functions. As a result, the average-case learning curve of the approximation converges to the average-case learning curve of the full GP. Through empirical investigation, we have confirmed this and studied the effect of m on the average-case learning curves. We have utilised the series approximation expression of the GP to construct strict shape-constrained models that are analytically tractable. We have done so for positive, monotonic and u-shaped models, and we have derived their prior mean and variance. We have compared our proposed models with other shape-constrained models from relevant literature and found that our model construction performs competitively. We have discussed the advantages and limitations of our models and the consequences of the choice of m, L and the prior distributions on the intercept terms. This thesis addresses four research questions, each summarised below with our findings.

How does the Hilbert space approximation affect predictive precision and computational complexity compared to a full Gaussian process? In chapters 3 and 4, we examined the role of m and L and showed how these both control the precision of the model and the computation complexity. In terms of precision, we saw that it is important that $m \to \infty$ before $L \to \infty$ in order to have convergence to the full model. In terms of computational complexity, we have shown that it is very beneficial to use the HS approximation for big datasets. After an initial cost of $\mathcal{O}(nm^2)$, the computational complexity of deriving the posterior distribution scales as $\mathcal{O}(m^3)$ as opposed to $\mathcal{O}(n^3)$ in full Gaussian process regression. Thus, the computational complexity is controlled by the number of basis functions m, resulting in a trade-off between precision and complexity. If we consider m isolated, we have that when the length scale of the data is high, fewer basis functions are required to capture the fluctuations in the data. For data with a low length scale, we will need higher values of m, resulting in more flexible functions. This trade-off is also affected by the choice of L. For data with a low length scale, we will need a larger domain so the posterior predictive has a 'buffer' before its variance goes to zero, requiring a larger L and, therefore, a larger m.

What are the advantages and limitations of using the Hilbert space approximation for enforcing shape-constrained functions? The series expansion of the HS approximation of a GP can be transformed by integration into a basis $\{\psi_{ij}\}_{i,j=1}^{m}$ for constructing monotonic functions and a basis $\{\Psi_{ij}\}_{i,j=1}^{m}$ for constructing convex functions respectively. Both of these bases are analytically tractable, and we can even derive the prior distributions of the positive, monotonic and convex functions analytically. The properties of m and L are, to a degree, inherited from the HS approximation. However, the integration transformation has a smoothening effect on the samples. The advantage is that once we have chosen appropriate m and L, there are no additional parameters in contrast to, e.g. the monotonic model in Riihimäki and Vehtari (2010), which is sensitive to the number and placement of the virtual points. An issue is finding appropriate priors of the hyperparameters and intercept terms, especially in the convex model and when data is scarce. The computational complexity of evaluating the functions from the shape-constrained HS models is squared compared to the cost of evaluating the functions from the HS approximation. In multiple-dimension data, this scales very quickly.

How does incorporating shape constraints affect prediction accuracy in the two regression

regimes, interpolation and extrapolation? Overall, we see a positive effect of incorporating shape-constraints in the interpolation regime. For the monotonic data, all three shape-constrained models perform better on the benchmark functions in experiment 1 than the baseline when measuring performance with the RMSE and ELPD. The effect is less prevalent for the u-shaped models in experiment 2 – here, the problem is more difficult, and none of the models stand out significantly. In the extrapolation regime, incorporating shape constraints can significantly improve the prediction accuracy, but it is also more challenging to calibrate the shape-constrained models. We saw this issue in the virtual points model (Riihimäki and Vehtari 2010), where the number of virtual points greatly affected the posterior predictions on the test data. We hypothesise that the exponentially increasing variance of the u-shaped model can be a problem, and a topic for further investigation is to attempt to reduce the variance of the u-shaped model in order to use it for extrapolation.

To what extent do the use of shape-constrained functions improve data efficiency? Overall, we have shown that shape-constrained models generally perform better than a baseline GP on relatively small datasets. Throughout the experiments, the shape constraints models performed better than the non-constrained model, indicating that incorporating shape constraints has a positive effect on data efficiency. However, our attempts at quantifying this improvement have had ambiguous results, and for very small datasets, our u-shaped HS model fails due to the exponentially growing prior variance. In contrast, the VP model has a steady performance on the smallest dataset, and thus, the most clear benefit of using shape constraints is the added robustness of the models. However, this requires that the models themselves are robust enough to handle small amounts of data. To examine the topic of the interplay between data efficiency and shape-constrained modelling more thoroughly, one suggestion is to raise the number of repetitions of experiment 4 in order to increase the validity of the results and to train the models on more datasets with different levels of noise variance and a wider variety of different training data sizes.

Bibliography

Adler, R. J. (1981). *The Geometry of Random Fields*. Wiley.

Andersen, Michael Riis et al. (2018). "A non-parametric probabilistic model for monotonic functions." In: *BNP@NeurIPS 2018 workshop. All of Bayesian Nonparametrics (Especially the Useful Bits)*.

Betancourt, Michael (2018). *A Conceptual Introduction to Hamiltonian Monte Carlo*. arXiv: 1701.02434 [stat.ME]. URL: https://arxiv.org/abs/1701.02434.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.

Blanchflower, David G. and Andrew J. Oswald (2008). "Is well-being U-shaped over the life cycle?" In: *Social Science & Medicine* 66.8, pp. 1733–1749. ISSN: 0277-9536. DOI: https://doi.org/10.1016/j.socscimed.2008.01.030. URL: https://www.sciencedirect.com/science/article/pii/S0277953608000245.

Brinkerhoff, Douglas J. (2022). "Variational inference at glacier scale". In: *Journal of Computational Physics* 459, p. 111095. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2022.111095. URL: https://www.sciencedirect.com/science/article/pii/S0021999122001577.

Christensen, Ole (2010). *Functions, Spaces, and Expansions: Mathematical Tools in Physics and Engineering*. English. 1st ed. Birkhäuser Verlag. ISBN: 978-0-8176-4979-1.

Conradsen, Johanne Hvidberg and Morten Rahbæk Boilesen (2025). *Hilbert space approximations of Gaussian processes in Bayesian modelling*. https://github.com/MortenBoilesen/HS_approximations_Master_thesis.

Emzir, Muhammad F. et al. (2019). "Hilbert-Space Reduced-Rank Methods For Deep Gaussian Processes". In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. DOI: 10.1109/MLSP.2019.8918874.

Evans, Lawrence C. (1998). *Partial differential equations*. Graduate studies in mathematics, v. 19. American Mathematical Society. ISBN: 0821807722.

Fradi, Anis and Khalid Daoudi (Apr. 2024). "Reduced-rank spectral mixtures Gaussian processes for probabilistic time–frequency representations". In: *Signal Process.* 218.C. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2023.109355. URL: https://doi.org/10.1016/j.sigpro.2023.109355.

Jones, M.R., T.J. Rogers, and E.J. Cross (2023). "Constraining Gaussian processes for physics-informed acoustic emission mapping". In: *Mechanical Systems and Signal Processing* 188, p. 109984. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2022.109984. URL: https://www.sciencedirect.com/science/article/pii/S0888327022010524.

Kelly, Colleen and John Rice (1990). "Monotone smoothing with application to dose–response curves and the assessment of synergism". In: *Biometrics* 46.

Köllmann, Claudia (Jan. 2016). "Unimodal spline regression and its use in various applications with single or multiple modes". PhD thesis. TU Dortmund University. DOI: 10.17877/DE290R-17270.

Kreyszig, E. (1991). *Introductory Functional Analysis with Applications*. Wiley Classics Library. Wiley. ISBN: 9780471504597.

Lenk, Peter and Taeryon Choi (2017). "Bayesian analysis of shape-restricted functions using Gaussian process priors". In: *Statistica Sinica*, pp. 43–69. ISSN: 19968507 and 10170405. DOI: 10.5705/ss.202015.0096.

Maatouk, Hassan (2017). *Finite-dimensional approximation of Gaussian processes with inequality constraints*. arXiv: 1706.02178 [stat.ME]. URL: https://arxiv.org/abs/1706.02178.

Maatouk, Hassan and Xavier Bay (2016). *Gaussian process emulators for computer experiments with inequality constraints*. arXiv: 1606.01265 [math.PR]. URL: https://arxiv.org/abs/1606.01265.

Meyer, Mary C. (2008). "Inference Using Shape-Restricted Regression Splines". In: *The Annals of Applied Statistics* 2.3, pp. 1013–1033. ISSN: 19326157. URL: http://www.jstor.org/stable/30245118 (visited on 07/01/2025).

Meyer, Richard F. and John W. Pratt (1968). "The Consistent Assessment and Fairing of Preference Functions". In: *IEEE Transactions on Systems Science and Cybernetics* 4.3, pp. 270–278. DOI: 10.1109/TSSC.1968.300121.

Murphy, Kevin P. (2023). *Probabilistic Machine Learning: Advanced Topics*. MIT Press. URL: http://probml.github.io/book2.

Opper, Manfred and Francesco Vivarelli (1999). "General Bounds on Bayes Errors for Regression with Gaussian Processes". In: *1999, Advances in Neural Information Processing Systems 11*, pp. 302–308.

Quinonero-Candela, Joaquin and Carl Edward Rasmussen (2005). "In: Murray-Smith, R., Shorten, R. (eds) Switching and Learning in Feedback Systems. Lecture Notes in Computer Science". In: vol. 3355. Springer, Berlin, Heidelberg. Chap. Analysis of Some Methods for Reduced Rank Gaussian Process Regression.

Ramsay, J. O. (1988). "Monotone Regression Splines in Action". In: *Statistical Science* 3.4, pp. 425–441. DOI: 10.1214/ss/1177012761.

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*. the MIT Press.

Reboul, L. (June 2005). "Estimation of a function under shape restrictions. Applications to reliability". In: *The Annals of Statistics* 33.3. ISSN: 0090-5364. DOI: 10.1214/009053605000000138. URL: http://dx.doi.org/10.1214/009053605000000138.

Riddell, Allen, Ari Hartikainen, and Matthew Carter (Mar. 2021). *pystan (3.10.0)*. PyPI.

Riihimäki, Jaakko and Aki Vehtari (2010). "Gaussian processes with monotonicity information". In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Riutort-Mayol, Gabriel et al. (2022). *Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming*. arXiv: 2004.11408 [stat.CO]. URL: https://arxiv.org/abs/2004.11408.

S. Holland Jr., Samuel (2007). *Applied Analysis by the Hilbert Space Method: An Introduction with Applications to the Wave, Heat, and Schrödinger Equations (Dover Books on Mathematics)*. Dover Ed. Dover Publications. ISBN: 0486458016.

Särkkä, Simo and Robert Piché (2014). "On convergence and accuracy of state-space approximations of squared exponential covariance functions". In: *2014 IEEE Iinternational workshop on machine learning for signal processing,* Volume 30.

Solin, Arno and Simo Särkkä (2020). "Hilbert Space Methods for Reduced-Rank Gaussian Process Regression". In: *Statistics and Computing* Volume 30.

Stan Development Team (Dec. 2024). *Stan Modeling Language (2.36)*.

Ustyuzhaninov, Ivan et al. (2020). "Monotonic Gaussian Process Flows". In: *Proceedings of the 23nd International Conference on Artificial Intelligence and Statistics (AISTATS)*.

– (2021). *Monotonic flow*. https://github.com/IevaKazlauskaite/monotonic_flow/.

Wang, Xiaojing and James O. Berger (2016). "Estimating Shape Constrained Functions Using Gaussian Processes". In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1, pp. 1–25. DOI: 10.1137/140955033.

World Bank Group DataBank (2024). *World Development Indicators*. URL: https://databank.worldbank.org/source/world-development-indicators (visited on 01/13/2025).

A Supplementary results and derivations

A.1 Negative Laplace Operator

A.1.1 Proof that $(-\nabla^2)$ is Hermitian

Let $f, g \in C^2$ be functions on the domain, Ω, with the assumed Dirichlet boundary conditions. Then we can write

$$\begin{aligned}
\langle(-\nabla^2)f, g\rangle &= -\sum_i \int_\Omega \frac{\partial^2 f}{\partial x_i^2}(x)g(x)\,\mathrm{d}x \\
&= -\sum_i \left(\underbrace{\left[\frac{\partial f}{\partial x_i}(x)g(x)\right]_{\partial\Omega}}_{=0} - \int_\Omega \frac{\partial f}{\partial x_i}(x)\frac{\partial g}{\partial x_i}(x)\,\mathrm{d}x \right) \\
&= -\sum_i \left(\underbrace{\left[f(x)\frac{\partial g}{\partial x_i}(x)\right]_{\partial\Omega}}_{=0} - \int_\Omega f(x)\frac{\partial^2 g}{\partial x_i^2}(x)\,\mathrm{d}x \right) \\
&= -\sum_i \int_\Omega f(x)\frac{\partial^2 g}{\partial x_i^2}(x)\,\mathrm{d}x \\
&= \langle f, (-\nabla^2)g\rangle
\end{aligned} \tag{A.1}$$

Showing that $(-\nabla^2)$ is indeed Hermitian.

A.1.2 Proof that $(-\nabla^2)$ positive semidefinite

We can also show that $(-\nabla^2)$ is a positive definite operator by inserting f instead of g in (A.1). Then we obtain

$$\langle(-\nabla^2)f, f\rangle = \sum_i \int_\Omega \frac{\partial f}{\partial x_i}(x)\frac{\partial f}{\partial x_i}(x)\,\mathrm{d}x = \sum_i \|\frac{\partial f}{\partial x_i}(x)\|^2 \tag{A.2}$$

This is a sum of norms and therefore strictly positive for $f \neq 0$

A.1.3 Proof that $(-\nabla^2)^k$ is Hermitian

Let $f, g \in C^{2k}$ be functions on the domain, Ω, with the assumed Dirichlet boundary conditions. Then it follows by repetitively using that $(-\nabla^2)$ is Hermitian. First we have

$$\langle(-\nabla^2)^k f, g\rangle = \langle(-\nabla^2)(-\nabla^2)^{k-1} f, g\rangle = \langle(-\nabla^2)^{k-1} f, (-\nabla^2)g\rangle. \tag{A.3}$$

Then we repeat this k times and obtain

$$\langle(-\nabla^2)^k f, g\rangle = \langle f, (-\nabla^2)^k g\rangle. \tag{A.4}$$

A.2 Supplementary proofs for section 4.1

A.2.1 Proof of lemma 4.1.1

By the fundamental theorem of calculus we have

$$f(\omega) - f(j\Delta - \alpha\Delta) = \int_{j\Delta-\alpha\Delta}^{\omega} f'(\omega')\,\mathrm{d}\omega' \tag{A.5}$$

and taking the absolute value

$$|f(\omega) - f(j\Delta - \alpha\Delta)| = \left| \int_{j\Delta-\alpha\Delta}^{\omega} f'(\omega') \, \mathrm{d}\omega' \right| \leq \int_{j\Delta-\alpha\Delta}^{\omega} \left| f'(\omega') \right| \, \mathrm{d}\omega'. \tag{A.6}$$

For $\omega \in ((j-1)\Delta, j\Delta]$ we then have

$$|f(\omega) - f(j\Delta - \alpha\Delta)| \leq \int_{(j-1)\Delta}^{j\Delta} \left| f'(\omega') \right| \, \mathrm{d}\omega'. \tag{A.7}$$

Returning to (4.20), we may split the integral into a sum to obtain

$$\begin{aligned}
\left| \int_{m\Delta}^{\infty} f(\omega) \, \mathrm{d}\omega - \sum_{j=m+1}^{\infty} f(j\Delta - \alpha\Delta)\Delta \right| &\leq \sum_{j=m+1}^{\infty} \int_{(j-1)\Delta}^{j\Delta} |f(\omega) - f(j\Delta - \alpha\Delta)| \, \mathrm{d}\omega \\
&\overset{(A.7)}{\leq} \sum_{j=m+1}^{\infty} \int_{(j-1)\Delta}^{j\Delta} \left[\int_{(j-1)\Delta}^{j\Delta} \left| f'(\omega') \right| \, \mathrm{d}\omega' \right] \mathrm{d}\omega \\
&= \sum_{j=m+1}^{\infty} \int_{(j-1)\Delta}^{j\Delta} \left| f'(\omega') \right| \, \mathrm{d}\omega' \Delta \\
&= \underbrace{\int_{m\Delta}^{\infty} \left| f'(\omega') \right| \, \mathrm{d}\omega' \Delta}_{K^{(m)}} \\
&\leq \underbrace{\int_{0}^{\infty} \left| f'(\omega') \right| \, \mathrm{d}\omega' \Delta}_{K^{(0)}},
\end{aligned} \tag{A.8}$$

which concludes the proof.

A.2.2 Proof of 4.1.2.

By using (A.7) we have

$$|f(j\Delta) - f(j\Delta - \alpha\Delta)| \leq \int_{(j-1)\Delta}^{j\Delta} |f'(\omega)| \, \mathrm{d}\omega,$$

Then by the triangle inequality and boundedness of g we have

$$\begin{aligned}
\left| \sum_{j=1}^{\infty} \left(f(j\Delta) - f(j\Delta - \alpha\Delta) \right) g(j\Delta - \beta\Delta) \right| &\leq \sum_{j=1}^{\infty} |(f(j\Delta) - f(j\Delta - \alpha\Delta))| \, |g(j\Delta - \beta\Delta)| \\
&\leq \sum_{j=1}^{\infty} \left(\int_{(j-1)\Delta}^{j\Delta} |f'(\omega)| \, \mathrm{d}\omega \right) K_2 \\
&\leq \int_{0}^{\infty} |f'(\omega)| \, \mathrm{d}\omega \, K_2 = K_1 K_2.
\end{aligned} \tag{A.9}$$

which finishes the proof.

A.2.3 Proof of 4.1.3.

By applying the mean value theorem to (A.7) we have that for some $\omega_j^* \in [j\Delta - \alpha\Delta, j\Delta]$

$$|g(j\Delta) - g(j\Delta - \beta\Delta)| = |g'(\omega_j^*)||\beta\Delta| \leq |g'(\omega_j^*)|\Delta \leq K_2\Delta. \tag{A.10}$$

The last inequality comes from $|g'(\omega)|$ being bounded for any ω. By using Lemma 4.1.1 and the fact that $f(\omega)$ is bounded we get

$$\begin{aligned}\left|\sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)\Delta\right| &\leq \left|\sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)\Delta - \int_0^{\infty} f(\omega)\,\mathrm{d}\omega\right| + \left|\int_0^{\infty} f(\omega)\,\mathrm{d}\omega\right| \\ &\leq K_1 + K_0 = K.\end{aligned} \tag{A.11}$$

Then we combine the two results above and use the triangle inequality and the positiveness of f to finish the proof of Lemma 4.1.3.

$$\left|\sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)\left[g(j\Delta) - g(j\Delta - \beta\Delta)\right]\right| \tag{A.12}$$

$$\leq \sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)\left|g(j\Delta) - g(j\Delta - \beta\Delta)\right| \tag{A.13}$$

$$\leq \sum_{j=1}^{\infty} f(j\Delta - \alpha\Delta)\Delta K_2 \leq KK_2. \tag{A.14}$$

which proves the lemma.

A.2.4 Additional details on proof on convergence in the multivariate case

Here we give the additional details on the multivariate proof in 4.1

We start from equation (4.55) given by

$$\begin{aligned}&\left|\tilde{k}_{\infty}(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \boldsymbol{x}')\right| \leq \\ &\frac{D_{1,1}}{L_1} + \Bigg|\underbrace{\sum_{j_2,\ldots,j_d}^{\infty}\left(\frac{1}{\pi}\int_0^{\infty} S(\omega_1, \ldots, \lambda_d)\cos(\omega_1(x_1 - x_1')\,\mathrm{d}\omega_1\right)\prod_{k=2}^{d}\frac{1}{L_k}\phi_k\left(x_k\right)\phi_k\left(x_k'\right)}_{H} \\ &\qquad - \frac{1}{\pi^d}\int_0^{\infty}\cdots\int_0^{\infty} S(\omega_1, \ldots, \omega_d)\prod_{k=1}^{d}\cos(\omega_k(x_k - x_k'))\,\mathrm{d}\omega_1 \ldots \mathrm{d}\omega_d\Bigg|\end{aligned} \tag{A.15}$$

To avoid the lines of the calculations becoming too long, we disregard the parts of (4.49) that do not change and thus continue by considering only H. We may rewrite this as

$$
\begin{aligned}
H &= \sum_{j_2,\ldots,j_d}^{\infty} \left(\frac{1}{\pi} \int_0^{\infty} S(\omega_1, \ldots, \lambda_d) \cos(\omega_1 (x_1 - x_1')) \, \mathrm{d}\omega_1 \right) \prod_{k=2}^{d} \frac{1}{L_k} \phi_k(x_k) \, \phi_k(x_k')) \\
&= \sum_{j_3,\ldots,j_d=1}^{\infty} \left(\frac{1}{\pi} \int_0^{\infty} \left(\sum_{j_2=1}^{\infty} S(\omega_1, \ldots, \lambda_d) \frac{1}{L_2} \phi_2(x_2) \, \phi_2(x_2') \right) \cos(\omega_1 (x_1 - x_1')) \, \mathrm{d}\omega_1 \right) \\
&\qquad \times \prod_{k=3}^{d} \frac{1}{L_k} \phi_k(x_k) \, \phi_k(x_k') \, .
\end{aligned} \tag{A.16}
$$

If we take the absolute value, add zero in a convenient way and use the triangle inequality, we obtain

$$
\begin{aligned}
&= \Bigg| \sum_{j_3,\ldots,j_d=1}^{\infty} \frac{1}{\pi} \int_0^{\infty} \Bigg(\sum_{j_2=1}^{\infty} S(\omega_1, \ldots, \lambda_d) \frac{1}{L_2} \phi_2(x_2) \, \phi_2(x_2') \\
&\quad - \frac{1}{\pi} \int_0^{\infty} S(\omega_1, \omega_2 \ldots, \lambda_d) \cos(\omega_2 (x_2 - x_2')) \, \mathrm{d}\omega_2 + \frac{1}{\pi} \int_0^{\infty} S(\omega_1, \omega_2 \ldots, \lambda_d) \cos(\omega_2 (x_2 - x_2')) \, \mathrm{d}\omega_2 \Bigg) \\
&\qquad \times \cos(\omega_1 (x_1 - x_1')) \, \mathrm{d}\omega_1 \times \prod_{k=3}^{d} \frac{1}{L_k} \phi_k(x_k) \, \phi_k(x_k') \Bigg| \\
&\leq \underbrace{\sum_{j_3,\ldots,j_d=1}^{\infty} \Bigg| \frac{1}{\pi} \int_0^{\infty} \Bigg(\sum_{j_2=1}^{\infty} S(\omega_1, \ldots, \lambda_d) \frac{1}{L_2} \phi_2(x_2) \, \phi_2(x_2') \\
&\qquad - \frac{1}{\pi} \int_0^{\infty} S(\omega_1, \omega_2 \ldots, \lambda_d) \cos(\omega_2 (x_2 - x_2')) \, \mathrm{d}\omega_2 \Bigg) \cos(\omega_1 (x_1 - x_1')) \, \mathrm{d}\omega_1 \Bigg| \prod_{k=3}^{d} \frac{1}{L_k}}_{I} \\
&\quad + \left| \sum_{j_3,\ldots,j_d=1}^{\infty} \left(\frac{1}{\pi^2} \int_0^{\infty} \int_0^{\infty} S(\omega_1, \omega_2 \ldots, \lambda_d) \cos(\omega_1 (x_1 - x_1')) \cos(\omega_2 (x_2 - x_2')) \, \mathrm{d}\omega_2 \, \mathrm{d}\omega_1 \right) \prod_{k=3}^{d} \frac{1}{L_k} \phi_k(x_k) \, \phi_k(x_k') \right|.
\end{aligned} \tag{A.17}
$$

Continuing with I, we use that $\left| \int f(x) \, \mathrm{d}x \right| \leq \int |f(x)| \, \mathrm{d}x$ and equation (4.47).

$$
\begin{aligned}
I &\leq \sum_{j_3,\ldots,j_d=1}^{\infty} \frac{1}{\pi} \int_0^{\infty} \Bigg| \sum_{j_2=1}^{\infty} S(\omega_1, \ldots, \lambda_d) \frac{1}{L_2} \phi_2(x_2) \, \phi_2(x_2') \\
&\qquad - \frac{1}{\pi} \int_0^{\infty} S(\omega_1, \omega_2 \ldots, \lambda_d) \cos(\omega_2 (x_2 - x_2')) \, \mathrm{d}\omega_2 \Bigg| \, \mathrm{d}\omega_1 \prod_{k=3}^{d} \frac{1}{L_k} \\
&\leq \sum_{j_3,\ldots,j_d=1}^{\infty} \frac{1}{\pi} \int_0^{\infty} \frac{B_2 + 2(A_2 + B_2)\tilde{L}}{L_2} \, \mathrm{d}\omega_1 \prod_{k=3}^{d} \frac{1}{L_k}.
\end{aligned} \tag{A.18}
$$

Noting that

$$
\int_0^{\infty} \sum_{j_3,\ldots,j_d=1}^{\infty} A_2 \, \mathrm{d}\omega_1 = \sum_{j_3,\ldots,j_d=1}^{\infty} \int_0^{\infty} A_2 \, \mathrm{d}\omega_1 = \sum_{j_3,\ldots,j_d=1}^{\infty} \int_0^{\infty} \int_0^{\infty} S(\lambda_{j_1}, \omega_2, \ldots, \lambda_d) \, \mathrm{d}\omega_2 \, \mathrm{d}\omega_1, \tag{A.19}
$$

and likewise,

$$
\int_0^{\infty} \sum_{j_3,\ldots,j_d=1}^{\infty} B_2 \, \mathrm{d}\omega_1 = \sum_{j_3,\ldots,j_d=1}^{\infty} \int_0^{\infty} \int_0^{\infty} \left| S'(\lambda_{j_1}, \omega_2, \ldots, \lambda_d) \right| \, \mathrm{d}\omega_2 \, \mathrm{d}\omega_1. \tag{A.20}
$$

we have already shown that there exists $D_{1,2}$ such that $I \leq \frac{D_{1,2}}{L_2}$. Putting it all together we may return to (4.55) and obtain

$$\begin{aligned}\left|\tilde{k}_\infty(\boldsymbol{x},\boldsymbol{x}') - k(\boldsymbol{x},\boldsymbol{x}')\right| \leq \frac{D_{1,1}}{L_1} + \frac{D_{1,2}}{L_2} \\ + \left|\sum_{j_3,\ldots,j_d}^{\infty}\left(\frac{1}{\pi^2}\int_0^\infty\int_0^\infty S(\omega_1,\omega_2,\ldots,\lambda_d)\cos(\omega_1(x_1-x_1'))\cos(\omega_1(x_2-x_2'))\,\mathrm{d}\omega_1\,\mathrm{d}\omega_2\right)\right. \\ \times\prod_{k=3}^{d}\frac{1}{L_k}\sin\left(\sqrt{\lambda_{j_k}}(x_k+L)\right)\sin\left(\sqrt{\lambda_{j_k}}(x_k'+L)\right) \\ \left.-\frac{1}{\pi^d}\int_0^\infty\cdots\int_0^\infty S(\omega_1,\ldots,\omega_d)\prod_{k=1}^{d}\cos(\omega_k(x_k-x_k'))\,\mathrm{d}\omega_1\ldots\,\mathrm{d}\omega_d\right|\end{aligned} \tag{A.21}$$

A.3 Monotonic functions

In order to compute the ψ_{ij}'s we note that $\sin(a)\sin(b) = \frac{1}{2}\left(\cos(a-b)+\cos(a+b)\right)$ and

$$\int_{-L}^{x}\cos\left(\gamma(x+L)\right)\,\mathrm{d}x = \int_0^{\gamma(x+L)}\frac{\cos(u)}{\gamma}\,\mathrm{d}u = \frac{\sin\left(\gamma(x+L)\right)}{\gamma} \tag{A.22}$$

we obtain for $i \neq j$

$$\begin{aligned}\psi_{ij} &= \frac{1}{2L}\int_{-L}^{x}\cos\left((\sqrt{\lambda_i}-\sqrt{\lambda_j})(x+L)\right) - \cos\left((\sqrt{\lambda_i}+\sqrt{\lambda_j})(x+L)\right)\,\mathrm{d}x \\ &= \frac{\sin\left((\sqrt{\lambda_i}-\sqrt{\lambda_j})(x+L)\right)}{2(\sqrt{\lambda_i}-\sqrt{\lambda_j})L} - \frac{\sin\left((\sqrt{\lambda_i}+\sqrt{\lambda_j})(x+L)\right)}{2(\sqrt{\lambda_i}+\sqrt{\lambda_j})L} \\ &= \frac{\sin\left(\gamma_{ij}^{-}(x+L)\right)}{2\gamma_{ij}^{-}L} - \frac{\sin\left(\gamma_{ij}^{+}(x+L)\right)}{2\gamma_{ij}^{+}L}\end{aligned} \tag{A.23}$$

when we let $\gamma_{ij}^{\pm} = \sqrt{\lambda_i} \pm \sqrt{\lambda_j}$.

For $i = j$ we obtain

$$\begin{aligned}\psi_{ii} &= \frac{1}{2L}\int_{-L}^{x}\cos(0)\,\mathrm{d}x - \frac{1}{2L}\int_{-L}^{x}\cos\left(\sqrt{2\lambda_i}(x+L)\right)\,\mathrm{d}x \\ &= \frac{x+L}{2L} - \frac{\sin\left(2\sqrt{\lambda_i}(x+L)\right)}{4\sqrt{\lambda_i}L} \\ &= \frac{x+L}{2L} - \frac{\sin\left(\gamma_{ii}^{+}(x+L)\right)}{2\gamma_{ii}^{+}L}.\end{aligned} \tag{A.24}$$

A.4 Ushaped Functions

A.4.1 Deriving Ψ_{ij}

In order to compute $\boldsymbol{\Psi}$, we use that for $\gamma \in \{\gamma_{ij}^{+}, \gamma_{ij}^{-}, \gamma_{jj}^{+}\}$

$$\int_{-L}^{x}\frac{\sin\left(\gamma(s+L)\right)}{2L\gamma}\,\mathrm{d}s = \int_0^{\gamma(x+L)}\frac{\sin(u)}{2L\gamma^2}\,\mathrm{d}u = \frac{1-\cos\left(\gamma(x+L)\right)}{2L\gamma^2} \tag{A.25}$$

for $i = j$

$$\begin{aligned}\Psi_{ij} &= \int_{-L}^{x} \frac{1}{2L}(s+L) - \frac{\sin\left(\gamma_{jj}^{+}(s+L)\right)}{2L\gamma_{jj}^{+}}\,\mathrm{d}s \\ &= \frac{1}{2L}\left[\frac{1}{2}s^2 + sL\right]_{-L}^{x} - \frac{1-\cos\left(\gamma_{jj}^{+}(x+L)\right)}{2L(\gamma_{jj}^{+})^2} \\ &= \frac{\frac{1}{2}x^2 + xL - \frac{1}{2}L^2 + L^2}{2L} - \frac{1-\cos\left(\gamma_{jj}^{+}(x+L)\right)}{2L(\gamma_{jj}^{+})^2} \\ &= \frac{(x+L)^2}{4L} + \frac{\cos\left(\gamma_{jj}^{+}(x+L)\right)-1}{2L(\gamma_{jj}^{+})^2}\end{aligned} \tag{A.26}$$

and for $i \neq j$

$$\begin{aligned}\Psi_{ij} &= \int_{L}^{x} \frac{\sin\left(\gamma_{ij}^{-}(s+L)\right)}{2L\gamma_{ij}^{-}} - \frac{\sin\left(\gamma_{ij}^{+}(s+L)\right)}{2L\gamma_{ij}^{+}}\,\mathrm{d}s \\ &= \frac{1-\cos\left(\gamma_{ij}^{-}(x+L)\right)}{2L(\gamma_{ij}^{-})^2} + \frac{\cos\left(\gamma_{ij}^{+}(x+L)\right)-1}{2L(\gamma_{ij}^{+})^2}\end{aligned} \tag{A.27}$$

A.4.2 Derivation of expected value

Assume that f_0 and F_0 are fixed and that $g(x)$ has mean μ and a stationary covariance function. Furthermore we assume that the marginal variance is given by $\mathbb{V}[g(x)] = \eta^2$. Then

$$\begin{aligned}\mathbb{E}[F(x)] =& \mathbb{E}\left[F_0 - f_0(x-a) + \int_a^x \int_a^s g(s')^2\,\mathrm{d}s'\,\mathrm{d}s\right] \\ =& F_0 - f_0(x-a) + \int_a^x \int_a^s \mathbb{E}\left[g(s')^2\right]\,\mathrm{d}s'\,\mathrm{d}s \\ =& F_0 - f_0(x-a) + \int_a^x \int_a^s \mu^2 + \eta^2 + \,\mathrm{d}s'\,\mathrm{d}s \\ =& F_0 - f_0(x-a) + (\mu^2+\eta^2)\left[\frac{s^2}{2} - sa\right]_a^x \\ =& F_0 - f_0(x-a) + \frac{\mu^2+\eta^2}{2}(x-a)^2\end{aligned} \tag{A.28}$$

A.4.3 Derivation of the variance

In order to calculate the variance, we now need to calculate the second moment of $F(x)$. We start by showing that

$$\mathbb{E}[g(s)^2 g(s')^2] = (\eta^2+\mu^2)^2 + 2k_{ss'}^2 + 4k_{ss'}\mu^2 \tag{A.29}$$

This will be needed later in the variance derivation. We can express the distribution of $g(s)$ and $g(s')$ as multivariate normal distribution

$$\tilde{\mathbf{g}} = \begin{pmatrix} g(s) \\ g(s') \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu \\ \mu\end{bmatrix}, \begin{pmatrix} k_{ss} & k_{ss'} \\ k_{s's} & k_{s's'} \end{pmatrix}\right) \tag{A.30}$$

In order to find $\mathbb{E}[g(s)^2 g(s')^2]$ we can use the moment generating function of $\tilde{\mathbf{g}}$ which is given by

$$M_{\tilde{\mathbf{g}}}(\mathbf{t}) = e^{m(\mathbf{t})} \tag{A.31}$$

with

$$m(\mathbf{t}) = \mathbf{t}^T\boldsymbol{\mu} + \frac{1}{2}\mathbf{t}^T\mathbf{K}\mathbf{t} = t_s\mu + t_{s'}\mu + \frac{1}{2}(k_{ss}t_s^2 + 2k_{ss'}t_st_{s'} + k_{s's'}t_{s'}^2) \tag{A.32}$$

We can now use the following formula

$$\mathbb{E}[X_1^{k_1}\dots X_n^{k_n}] = \frac{\partial^k}{\partial t_1^{k_1}\dots\partial t_n^{k_n}} M_{\mathbf{X}}(\mathbf{t})|_{\mathbf{t}=0} \tag{A.33}$$

such that

$$\mathbb{E}[g(s)^2 g(s')^2] = \frac{\partial^4}{\partial t_s^2 \partial t_{s'}^2} M_{\tilde{\mathbf{g}}}(\mathbf{t})|_{\mathbf{t}=0} \tag{A.34}$$

First we have

$$\begin{aligned}\frac{\partial^4}{\partial t_s^2\partial t_{s'}^2}e^{m(\mathbf{t})} =& \left(\frac{\partial^2 m(\mathbf{t})}{\partial t_s^2}\frac{\partial^2 m(\mathbf{t})}{\partial t_{s'}^2} + \frac{\partial^2 m(\mathbf{t})}{\partial t_s^2}\left(\frac{\partial m(\mathbf{t})}{\partial t_{s'}}\right)^2 + \left(\frac{\partial m(\mathbf{t})}{\partial t_s}\right)^2\frac{\partial^2 m(\mathbf{t})}{\partial t_{s'}^2}\right.\\ &\left. +2\left(\frac{\partial^2 m(\mathbf{t})}{\partial t_s\partial t_{s'}}\right)^2 + 4\frac{\partial^2 m(\mathbf{t})}{\partial t_s\partial t_{s'}}\frac{\partial m(\mathbf{t})}{\partial t_s}\frac{\partial m(\mathbf{t})}{\partial t_{s'}} + \left(\frac{\partial m(\mathbf{t})}{\partial t_s}\right)^2\left(\frac{\partial m(\mathbf{t})}{\partial t_{s'}}\right)^2\right)e^{m(\mathbf{t})}\end{aligned} \tag{A.35}$$

Inserting $t_s = 0$, $t_{s'} = 0$ and $k_{ss} = \eta^2$ we obtain

$$\begin{aligned}\mathbb{E}[g(s)^2 g(s')^2] =& \eta^4 + 2\mu^2\eta^2 + 2k_{ss'}^2 + 4k_{ss'}\mu^2 + \mu^4\\ =& (\eta^2+\mu^2)^2 + 2k_{ss'}^2 + 4k_{ss'}\mu^2\end{aligned} \tag{A.36}$$

since

$$\begin{aligned} m(\mathbf{0}) &= 0\\ \frac{\partial m(\mathbf{0})}{\partial t_n} &= \mu \text{ for } n \in \{s, s'\}\\ \frac{\partial^2 m(\mathbf{0})}{\partial t_{n_1}\partial t_{n_2}} &= k_{n_1,n_2} \text{ for } n_1, n_2 \in \{s, s'\}\end{aligned} \tag{A.37}$$

Now we are ready to compute the second moment of F:

$$\begin{aligned}\mathbb{E}\left[F^2(x)\right] &= \mathbb{E}\left[\left(F_0 - f_0(x-a) + \int_a^x\int_a^s g(s')^2\,\mathrm{d}s'\,\mathrm{d}s\right)^2\right]\\ &= \mathbb{E}\left[\left(F_0 - f_0(x-a) + \int_a^x\int_a^s g(s')^2\,\mathrm{d}s'\,\mathrm{d}s\right)\left(F_0 - f_0(x-a) + \int_a^x\int_a^r g(r')^2\,\mathrm{d}r'\,\mathrm{d}r\right)\right]\\ &= \mathbb{E}\left[(F_0 - f_0(x-a))^2\right] + \mathbb{E}\left[\left(\int_a^x\int_a^s g(r')^2\,\mathrm{d}s'\,\mathrm{d}s\right)\left(\int_a^x\int_a^r g(r')^2\,\mathrm{d}r'\,\mathrm{d}r\right)\right]\\ &\qquad +2\mathbb{E}\left[(F_0 - f_0(x-a))\int_a^x\int_a^s g(r')^2\,\mathrm{d}s'\,\mathrm{d}s\right]\\ &= (F_0 - f_0(x-a))^2 + \mathbb{E}\left[\left(\int_a^x\int_a^s g(r')^2\,\mathrm{d}s'\,\mathrm{d}s\right)\left(\int_a^x\int_a^r g(r')^2\,\mathrm{d}r'\,\mathrm{d}r\right)\right]\\ &\qquad +2\left(F_0 - f_0(x-a)\right)\mathbb{E}\left[\int_a^x\int_a^s g(r')^2\,\mathrm{d}s'\,\mathrm{d}s\right]\end{aligned} \tag{A.38}$$

Using the second moment of the Gaussian distribution we have

$$\mathbb{E}\left[\int_a^x \int_a^s g(r')^2 \,\mathrm{d}s' \,\mathrm{d}s\right] = \int_a^x \int_a^s \mathbb{E}\left[g(r')^2\right] \,\mathrm{d}s' \,\mathrm{d}s = \int_a^x \int_a^s \mu^2+\eta^2 \,\mathrm{d}s' \,\mathrm{d}s = \frac{1}{2}(\mu^2+\eta^2)(x-a)^2 \tag{A.39}$$

and continuing with the product integral term, we note that we are integrating over positive, continuous functions and can use Tonelli's theorem to switch the order of integrands.

$$\begin{aligned}&\mathbb{E}\left[\left(\int_a^x \int_a^s g(r')^2 \,\mathrm{d}s' \,\mathrm{d}s\right)\left(\int_a^x \int_a^r g(r')^2 \,\mathrm{d}r' \,\mathrm{d}r\right)\right] \\ &= \mathbb{E}\left[\int_a^x \int_a^x \int_a^s \int_a^r g(s')^2 g(r')^2 \,\mathrm{d}r' \,\mathrm{d}s' \,\mathrm{d}r \,\mathrm{d}s\right]\end{aligned} \tag{A.40}$$

Collecting the terms from $\mathbb{E}(F^2(x))$ that do not depend on $k(r', s')$ we obtain

$$\begin{aligned}&(F_0 - f_0(x-a))^2 + (F_0 - f_0(x-a)\frac{1}{2}(\mu^2+\eta^2)(x-a)^2) + \frac{1}{4}(\mu^2+\eta^2)^2(x-a)^4 \\ &= \left(F_0 - f_0(x-a) + \frac{1}{2}(\mu^2+\eta^2)(x-a)^2\right)^2 \\ &= \mathbb{E}\left[F(x)\right]^2\end{aligned} \tag{A.41}$$

and thus

$$\begin{aligned}\mathbb{V}\left[F(x)\right] &= \mathbb{E}\left[F^2(x)\right] - \mathbb{E}\left[F(x)\right]^2 \\ &= \int_a^x \int_a^x \int_a^s \int_a^r 4k(r', s')\mu^2 \,\mathrm{d}r' \,\mathrm{d}s' \,\mathrm{d}r \,\mathrm{d}s + \int_a^x \int_a^x \int_a^s \int_a^r 2k(r', s')^2 \,\mathrm{d}r' \,\mathrm{d}s' \,\mathrm{d}r \,\mathrm{d}s\end{aligned} \tag{A.42}$$

A.4.4 Prior variance with squared exponential kernel functions.

Let $k(x, x') = \eta^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$

We begin by considering the first term, i.e.

$$\int_a^x \int_a^x \int_a^s \int_a^r 4k(r', s')\mu^2 \,\mathrm{d}r' \,\mathrm{d}s' \,\mathrm{d}r \,\mathrm{d}s$$

Integrating this with respect to r' and s' we obtain

$$\begin{aligned}\int_a^s \int_a^r 4k(r', s')\mu^2 \,\mathrm{d}r' \,\mathrm{d}s' &= 4\mu^2\eta^2 \int_a^s \int_a^r \exp\left(\frac{-(r'-s')^2}{2\ell^2}\right) \,\mathrm{d}r' \,\mathrm{d}s' \\ &\overset{u(r')=(r'-s')/(\sqrt{2}\ell)}{=} 4\mu^2\eta^2\sqrt{2}\ell \int_a^s \int_{(a-s')/(\sqrt{2}\ell)}^{(r-s')/(\sqrt{2}\ell)} \exp\left(-u^2\right) \,\mathrm{d}u \,\mathrm{d}s' \\ &= 4\mu^2\eta^2\sqrt{2}\ell\frac{\sqrt{\pi}}{2} \int_a^s \left[\mathrm{erf}(u)\right]_{(a-s')/(\sqrt{2}\ell)}^{(r-s')/(\sqrt{2}\ell)} \,\mathrm{d}s' \\ &= 4\mu^2\eta^2\ell\sqrt{\frac{\pi}{2}} \int_a^s \left(\mathrm{erf}\left(\frac{r-s'}{\sqrt{2}\ell}\right) - \mathrm{erf}\left(\frac{a-s'}{\sqrt{2}\ell}\right)\right) \,\mathrm{d}s'\end{aligned} \tag{A.43}$$

Now

$$\begin{aligned}\int_a^s \operatorname{erf}\left(\frac{r-s'}{\sqrt{2}\ell}\right) \mathrm{d}s' &\overset{v(s')=(r-s')/(\sqrt{2}\ell)}{=} -\sqrt{2}\ell \int_{(a-s)/(\sqrt{2}\ell)}^{(r-s)/(\sqrt{2}\ell)} \operatorname{erf}(v)\,\mathrm{d}v \\ &= -\sqrt{2}\ell \left[v\operatorname{erf}(v) + \frac{1}{\sqrt{\pi}}\exp\left(-v^2\right)\right]_{(r-a)/(\sqrt{2}\ell)}^{(r-s)/(\sqrt{2}\ell)} \\ &= \ell\left[\frac{s-r}{\ell}\operatorname{erf}\left(\frac{r-s}{\sqrt{2}\ell}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{r-s}{\sqrt{2}\ell}\right)^2\right)\right. \\ &\quad \left. - \frac{a-r}{\ell}\operatorname{erf}\left(\frac{r-a}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{r-a}{\sqrt{2}\ell}\right)^2\right)\right]\end{aligned} \tag{A.44}$$

and, similarily

$$\begin{aligned}\int_a^s \operatorname{erf}\left(\frac{a-s'}{\sqrt{2}\ell}\right) \mathrm{d}s' &= \ell\left[\frac{s-a}{\ell}\operatorname{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right)\right. \\ &\quad \left. - \frac{a-a}{\ell}\operatorname{erf}\left(\frac{a-a}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{a-a}{\sqrt{2}\ell}\right)^2\right)\right] \\ &= \ell\left[\frac{s-a}{\ell}\operatorname{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right) + \sqrt{\frac{2}{\pi}}\right]\end{aligned} \tag{A.45}$$

and thus

$$\begin{aligned}&\int_a^s \int_a^r 4k(r',s')\mu^2\,\mathrm{d}r'\,\mathrm{d}s' \\ &= 4\mu^2\eta^2\ell^2\sqrt{\frac{\pi}{2}}\left[\frac{s-r}{\ell}\operatorname{erf}\left(\frac{r-s}{\sqrt{2}\ell}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{r-s}{\sqrt{2}\ell}\right)^2\right)\right. \\ &\quad - \frac{a-r}{\ell}\operatorname{erf}\left(\frac{r-a}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{r-a}{\sqrt{2}\ell}\right)^2\right) \\ &\quad \left. - \frac{s-a}{\ell}\operatorname{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right) - \sqrt{\frac{2}{\pi}}\right]\end{aligned} \tag{A.46}$$

Next, we need to integrate over r and s from a to x. We do this term by term, starting with the error function terms that only depend on either s or r.

$$\begin{aligned}\int_a^x \frac{s-a}{\ell}\operatorname{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right) \mathrm{d}s &\overset{u(s)=(a-s)/(\sqrt{2}\ell)}{=} -\sqrt{2}\ell \int_0^{(a-x)/(\sqrt{2}\ell)} -\sqrt{2}u\operatorname{erf}(u)\,\mathrm{d}u \\ &= 2\ell \int_0^{(a-x)/(\sqrt{2}\ell)} u\operatorname{erf}(u)\,\mathrm{d}u\end{aligned} \tag{A.47}$$

Using integration by parts with $v'(u) = u$, $w(u) = \operatorname{erf}(u)$ we obtain

$$\int_0^{(a-x)/(\sqrt{2}\ell)} u\operatorname{erf}(u)\,\mathrm{d}u = \left[\frac{1}{2}u^2\operatorname{erf}(u)\right]_0^{(a-x)/(\sqrt{2}\ell)} - \frac{1}{\sqrt{\pi}}\int_0^{(a-x)/(\sqrt{2}\ell)} u^2\exp\left(-u^2\right)\,\mathrm{d}u \tag{A.48}$$

and by using integration by parts again with $v'(u) = u\exp\left(-u^2\right)$ $(v(u) = -\frac{1}{2}\exp\left(-u^2\right))$ and $w(u) = u$ we obtain

$$\begin{aligned}-\frac{1}{\sqrt{\pi}}\int_0^{(a-x)/(\sqrt{2}\ell)} u\left(u\exp\left(-u^2\right)\right)\,\mathrm{d}u &= -\frac{1}{\sqrt{\pi}}\left[\left[-\frac{1}{2}u\exp\left(-u^2\right)\right]_0^{(a-x)/(\sqrt{2}\ell)} - \int_0^{(a-x)/(\sqrt{2}\ell)} -\frac{1}{2}\exp\left(-u^2\right)\,\mathrm{d}u\right] \\ &= -\frac{1}{\sqrt{\pi}}\left[-\frac{1}{2}\frac{(a-x)}{\sqrt{2}\ell}\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) + \frac{1}{2}\frac{\sqrt{\pi}}{2}\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right)\right]\end{aligned} \tag{A.49}$$

And thus

$$\begin{aligned}2\ell\int_0^{(a-x)/(\sqrt{2}\ell)} u\mathrm{erf}\left(u\right)\,\mathrm{d}u &= \ell\left[\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right) + \frac{1}{\sqrt{\pi}}\frac{(a-x)}{\sqrt{2}\ell}\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) - \frac{1}{2}\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right)\right] \\ &= -\ell\left[\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) + \frac{1}{\sqrt{\pi}}\frac{x-a}{\sqrt{2}\ell}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) - \frac{1}{2}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right]\end{aligned} \tag{A.50}$$

By similar arguments

$$\begin{aligned}\int_a^x \frac{a-r}{\ell}\mathrm{erf}\left(\frac{r-a}{\sqrt{2}\ell}\right)\,\mathrm{d}r &\overset{u(r)=(r-a)/(\sqrt{2}\ell)}{=} -2\ell\int_0^{(x-a)/(\sqrt{2}\ell)} u\mathrm{erf}\left(u\right)\,\mathrm{d}u \\ &= -\ell\left[\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) + \frac{1}{\sqrt{\pi}}\frac{(x-a)}{\sqrt{2}\ell}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right. \\ &\quad \left. - \frac{1}{2}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right]\end{aligned} \tag{A.51}$$

We may now define $\omega_1(x, a, \ell)$ by

$$\begin{aligned}\omega_1(x,a,\ell) &= -\int_a^x\int_a^x\left(\frac{a-r}{\ell}\mathrm{erf}\left(\frac{r-a}{\sqrt{2}\ell}\right) + \int_a^x \frac{s-a}{\ell}\mathrm{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right)\right)\,\mathrm{d}r\,\mathrm{d}s \\ &= -\int_a^x\int_a^x \frac{a-r}{\ell}\mathrm{erf}\left(\frac{r-a}{\sqrt{2}\ell}\right)\,\mathrm{d}r\,\mathrm{d}s - \int_a^x\int_a^x \frac{s-a}{\ell}\mathrm{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right)\,\mathrm{d}s\,\mathrm{d}r \\ &\quad 2\ell(x-a)\left[\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) + \frac{1}{\sqrt{\pi}}\frac{(x-a)}{\sqrt{2}\ell}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) - \frac{1}{2}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right]\end{aligned} \tag{A.52}$$

Now for the error function term that depends both on r and s, we start by integrating over r to obtain (by similar substitutions and partial integration as before)

$$\begin{aligned}
\int_a^x \frac{s-r}{\ell}\mathrm{erf}\left(\frac{r-s}{\sqrt{2}\ell}\right)\,\mathrm{d}r &= -2\ell\Bigg[\frac{1}{2}\left[u^2\mathrm{erf}(u)\right]_{(a-s)/(\sqrt{2}\ell)}^{(x-s)/(\sqrt{2}\ell)} \\
&\quad - \frac{1}{\sqrt{\pi}}\left(-\frac{1}{2}\left[u\exp\left(-u^2\right)\right]_{(a-s)/(\sqrt{2}\ell)}^{(x-s)/(\sqrt{2}\ell)} - \left(-\frac{1}{2}\right)\left[\frac{\sqrt{\pi}}{2}\mathrm{erf}(u)\right]_{(a-s)/(\sqrt{2}\ell)}^{(x-s)/(\sqrt{2}\ell)}\right)\Bigg] \\
&= -\ell\Bigg[\left(\frac{x-s}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-s}{\sqrt{2}\ell}\right) - \left(\frac{a-s}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right) \\
&\quad - \frac{1}{\sqrt{\pi}}\left(-\left(\frac{x-s}{\sqrt{2}\ell}\right)\exp\left(-\left(\frac{x-s}{\sqrt{2}\ell}\right)^2\right) + \left(\frac{a-s}{\sqrt{2}\ell}\right)\exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right)\right. \\
&\quad \left. + \frac{\sqrt{\pi}}{2}\mathrm{erf}\left(\frac{x-s}{\sqrt{2}\ell}\right) - \frac{\sqrt{\pi}}{2}\mathrm{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right)\right)\Bigg]
\end{aligned} \tag{A.53}$$

Next we integrate with respect to s.

The first two terms requires us to use integration by parts twice:

$$\begin{aligned}
\int_A^B u^2\mathrm{erf}\left(u\right)\,\mathrm{d}s &= \left[\frac{1}{3}u^3\mathrm{erf}\left(u\right)\right]_A^B - \int_A^B \frac{1}{3}u^3\frac{2}{\sqrt{\pi}}\exp\left(-u^2\right) \\
&= \left[\frac{1}{3}u^3\mathrm{erf}\left(u\right)\right]_A^B - \frac{2}{3\sqrt{\pi}}\left(\left[u^2\left(-\frac{1}{2}\right)\exp\left(-u^2\right)\right]_A^B - \int_A^B 2u\left(-\frac{1}{2}\right)\exp\left(-u^2\right)\right) \\
&= \left[\frac{1}{3}u^3\mathrm{erf}\left(u\right)\right]_A^B - \frac{2}{3\sqrt{\pi}}\left(\left[u^2\left(-\frac{1}{2}\right)\exp\left(-u^2\right)\right]_A^B + \left[-\frac{1}{2}\exp\left(-u^2\right)\right]_A^B\right) \\
&= \frac{1}{3}\left[u^3\mathrm{erf}(u) + \frac{1}{\sqrt{\pi}}u^2\exp\left(-u^2\right) + \frac{1}{\sqrt{\pi}}\exp\left(-u^2\right)\right]_A^B
\end{aligned} \tag{A.54}$$

Using (A.54) and substituting $u_1 = (x-s)/(\sqrt{2}\ell)$ and $u_2 = (a-s)/(\sqrt{2}\ell)$ we define ω_2 by

$$\begin{aligned}
\omega_2(x,a,\ell) &= \int_a^x -\ell\left[\left(\frac{x-s}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-s}{\sqrt{2}\ell}\right) - \left(\frac{a-s}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right)\right]\,\mathrm{d}s \\
&= \frac{\sqrt{2}\ell^2}{3}\Bigg[\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-x}{\sqrt{2}\ell}\right)^2\right) \\
&\quad - \left(\frac{x-a}{\sqrt{2}\ell}\right)^3\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) - \frac{1}{\sqrt{\pi}}\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) - \frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) \\
&\quad - \left(\frac{a-x}{\sqrt{2}\ell}\right)^3\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right) - \frac{1}{\sqrt{\pi}}\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) - \frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) \\
&\quad + \frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{a-a}{\sqrt{2}\ell}\right)^2\right)\Bigg] \\
&= \frac{2^{3/2}\ell^2}{3\sqrt{\pi}}\left[1 - \sqrt{\pi}\left(\frac{x-a}{\sqrt{2}\ell}\right)^3\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) - \left(\frac{x-a}{\sqrt{2}\ell}\right)^2\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) - \exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]
\end{aligned} \tag{A.55}$$

For the next two terms, we need co compute

$$\int_A^B u \exp\left(-u^2\right)\,\mathrm{d}s = \left[-\frac{1}{2}\exp\left(-u^2\right)\right]_A^B \tag{A.56}$$

and using (A.56) we define ω_3 by

$$\begin{aligned}
\omega_3(x,a,\ell) &= \int_a^x \frac{\ell}{\sqrt{\pi}}\left[-\left(\frac{x-s}{\sqrt{2}\ell}\right)\exp\left(-\left(\frac{x-s}{\sqrt{2}\ell}\right)^2\right)+\left(\frac{a-s}{\sqrt{2}\ell}\right)\exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right)\right]\mathrm{d}s \\
&= \frac{\sqrt{2}\ell^2}{3}\left[-\exp\left(-\left(\frac{x-x}{\sqrt{2}\ell}\right)^2\right)+\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right. \\
&\qquad \left.+\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right)-\exp\left(-\left(\frac{a-a}{\sqrt{2}\ell}\right)^2\right)\right] \\
&= \frac{2^{3/2}\ell^2}{3}\left[-1+\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]
\end{aligned} \tag{A.57}$$

For the last two terms, we will just need to integrate the error function. We define ω_4 by

$$\begin{aligned}
\omega_4(x,a,\ell) &= \int_a^x \frac{\ell}{2}\left[\operatorname{erf}\left(\frac{x-s}{\sqrt{2}\ell}\right)-\operatorname{erf}\left(\frac{a-s}{\sqrt{2}\ell}\right)\right]\mathrm{d}s \\
&= \frac{-\ell^2}{\sqrt{2}}\left[\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-x}{\sqrt{2}\ell}\right)^2\right)\right. \\
&\qquad -\left(\frac{x-a}{\sqrt{2}\ell}\right)\operatorname{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)-\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) \\
&\qquad -\left(\frac{a-x}{\sqrt{2}\ell}\right)\operatorname{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right)-\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) \\
&\qquad \left.+\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{a-a}{\sqrt{2}\ell}\right)^2\right)\right] \\
&= -\sqrt{2}\ell^2\left[\frac{1}{\sqrt{\pi}}-\left(\frac{x-a}{\sqrt{2}\ell}\right)\operatorname{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)-\frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]
\end{aligned} \tag{A.58}$$

Returning to (A.46), we still need to compute three integrals, the first of which is

$$\int_a^x\int_a^x \exp\left(-\frac{(r-s)^2}{2\ell^2}\right)\mathrm{d}r\,\mathrm{d}s = \int_a^x\int_a^x k(r,s)\,\mathrm{d}r\,\mathrm{d}s \tag{A.59}$$

which means that we can reuse equation (A.46) to obtain

$$\begin{aligned}\int_a^x \int_a^x \exp\left(-\frac{(r-s)^2}{2\ell^2}\right) \mathrm{d}r\,\mathrm{d}s &= \ell^2\sqrt{\frac{\pi}{2}}\left[\frac{x-x}{\ell}\mathrm{erf}\left(\frac{x-x}{\sqrt{2}\ell}\right) - \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{x-x}{\sqrt{2}\ell}\right)^2\right)\right. \\ &\quad - \frac{a-x}{\ell}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) \\ &\quad \left. - \frac{x-a}{\ell}\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right) + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{a-x}{\sqrt{2}\ell}\right)^2\right) - \sqrt{\frac{2}{\pi}}\right] \\ &= 2\ell^2\sqrt{\frac{\pi}{2}}\left[-\sqrt{\frac{2}{\pi}} + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) + \left(\frac{x-a}{\ell}\right)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right]\end{aligned} \tag{A.60}$$

For the remaining two we have that

$$\begin{aligned}\int_a^x \int_a^x \exp\left(-\frac{(r-a)^2}{2\ell^2}\right) \mathrm{d}r\,\mathrm{d}s &= \int_a^x \sqrt{\frac{\pi}{2}}\ell\left[\mathrm{erf}\left(u\right)\right]_0^{(x-a)/(\sqrt{2}\ell)} \mathrm{d}s \\ &= \sqrt{\frac{\pi}{2}}(x-a)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\end{aligned} \tag{A.61}$$

and likewise

$$\begin{aligned}\int_a^x \int_a^x \exp\left(-\frac{(a-s)^2}{2\ell^2}\right) \mathrm{d}r\,\mathrm{d}s &= -\sqrt{\frac{\pi}{2}}(x-a)\mathrm{erf}\left(\frac{a-x}{\sqrt{2}\ell}\right) \\ &= \sqrt{\frac{\pi}{2}}(x-a)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\end{aligned} \tag{A.62}$$

Finally, we may define ω_5 as

$$\begin{aligned}\omega_5(x,a,\ell) &= \int_a^x \int_a^x \sqrt{\frac{2}{\pi}}\left[-\exp\left(-\left(\frac{r-s}{\sqrt{2}\ell}\right)^2\right) + \exp\left(-\left(\frac{r-a}{\sqrt{2}\ell}\right)^2\right) + \exp\left(-\left(\frac{a-s}{\sqrt{2}\ell}\right)^2\right)\right] \mathrm{d}r\,\mathrm{d}s \\ &= \sqrt{\frac{2}{\pi}}\left[-2\ell^2\sqrt{\frac{\pi}{2}}\left(-\sqrt{\frac{2}{\pi}} + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) + \left(\frac{x-a}{\ell}\right)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right)\right. \\ &\quad \left. + \sqrt{2\pi}(x-a)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right] \\ &= -2\ell^2\left(-\sqrt{\frac{2}{\pi}} + \sqrt{\frac{2}{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right) + \left(\frac{x-a}{\ell}\right)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right) \\ &\quad + 2(x-a)\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\end{aligned} \tag{A.63}$$

We collect all the terms to define

$$\begin{aligned}\omega(x,a,\ell,\eta) &= \int_a^x \int_a^x \int_a^s \int_a^r k(r',s')\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \\ &= \eta^2\ell^2\sqrt{\frac{\pi}{2}}\left(\omega_1(x,a,\ell)+\omega_2(x,a,\ell)+\omega_3(x,a,\ell)+\omega_4(x,a,\ell)+\omega_5(x,a,\ell)-(x-a)^2\sqrt{\frac{2}{\pi}}\right)\end{aligned} \tag{A.64}$$

and then

$$\int_a^x \int_a^x \int_a^s \int_a^r 4k(r',s')\mu^2\,\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s = 4\mu^2\omega(x,a,\ell,\eta) \tag{A.65}$$

In order to calculate the second integral term in (A.42) we start by doing the following reformulation of the term.

$$\begin{aligned}&\int_a^x \int_a^x \int_a^s \int_a^r 2\left(\eta^2\exp\left(\frac{-(r'-s')^2}{2\ell^2}\right)\right)^2 \mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \\ &= 2\eta^2\int_a^x \int_a^x \int_a^s \int_a^r \eta^2\exp\left(\frac{-2(r'-s')^2}{2\ell^2}\right)\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \\ &\overset{\rho=\frac{1}{\sqrt{2}}\ell}{=} 2\eta^2\int_a^x \int_a^x \int_a^s \int_a^r \eta^2\exp\left(\frac{-(r'-s')^2}{2\rho^2}\right)\mathrm{d}r'\,\mathrm{d}s'\,\mathrm{d}r\,\mathrm{d}s \\ &= 2\eta^2\omega(x,a,\rho,\eta)\end{aligned} \tag{A.66}$$

and thus, referring back to (A.42) we obtain

$$\mathbb{V}\left[F(x)\right] = 4\mu^2\omega(x,a,\ell,\eta) + 2\eta^2\omega(x,a,\rho,\eta) \tag{A.67}$$

For easier and more efficient implementation, we note that $\omega(x,a,\ell,eta)$ corresponds to a linear combination of the terms $\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)$, $\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)$ and 1. We collect the coefficients in front of each of these terms by going thourgh $\omega_1,\ldots,\omega_5$.

$$\omega_1(x,a,\ell) = 2\ell(x-a)\left[\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)+\frac{1}{\sqrt{\pi}}\frac{x-a}{\sqrt{2}\ell}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)-\frac{1}{2}\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right]$$

- error function coefficient:

$$2\ell(x-a)\frac{(x-a)^2}{2\ell^2}+2\ell(x-a)\frac{-1}{2} = \frac{(x-a)^3}{\ell^2}-\ell(x-a)$$

- exponential coefficient:

$$2\ell(x-a)\frac{1}{\sqrt{\pi}}\frac{x-a}{\sqrt{2}\ell} = \frac{\sqrt{2}(x-a)^2}{\sqrt{\pi}}$$

$$\omega_2(x,a,\ell) = \frac{2^{\frac{3}{2}}\ell^2}{3\sqrt{\pi}}\left[1-\sqrt{\pi}\left(\frac{x-a}{\sqrt{2}\ell}\right)^3\mathrm{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)-\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]$$

- error function coefficient:

$$-\frac{2^{3/2}\ell^2}{3\sqrt{\pi}}\sqrt{\pi}\frac{(x-a)^3}{2^{3/2}\ell^3} = -\frac{(x-a)^3}{3\ell}$$

- exponential coefficient:

$$-\frac{2^{3/2}\ell^2}{3\sqrt{\pi}}\frac{(x-a)^2}{2\ell^2} - \frac{2^{3/2}\ell^2}{3\sqrt{\pi}} = -\frac{\sqrt{2}}{3\sqrt{\pi}}(x-a)^2 - \frac{2^{3/2}\ell^2}{3\sqrt{\pi}}$$

- constant:

$$\frac{2^{3/2}\ell^2}{3\sqrt{\pi}}$$

$$\omega_3(x,a,\ell) = \frac{2^{\frac{3}{2}}\ell^2}{3}\left[-1+\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]$$

$$\omega_4(x,a,\ell) = -\sqrt{2}\ell^2\left[\frac{1}{\sqrt{\pi}} - \left(\frac{x-a}{\sqrt{2}\ell}\right)\operatorname{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right) - \frac{1}{\sqrt{\pi}}\exp\left(-\left(\frac{x-a}{\sqrt{2}\ell}\right)^2\right)\right]$$

- error function coefficient:

$$-\sqrt{2}\ell^2\frac{-(x-a)}{\sqrt{2}\ell} = (x-a)\ell$$

- exponential coefficient:

$$-\sqrt{2}\ell^2\frac{-1}{\sqrt{\pi}} = \frac{\sqrt{2}\ell^2}{\sqrt{\pi}}$$

- constant:

$$-\sqrt{2}\ell^2\frac{1}{\pi}$$

$$\omega_5(x,a,\ell) = -2\ell^2\left(-\sqrt{\frac{2}{\pi}} + \sqrt{\frac{2}{\pi}}\exp\left(-\frac{(x-a)^2}{2\ell^2}\right) + \left(\frac{x-a}{\ell}\right)\operatorname{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)\right) + 2(x-a)\operatorname{erf}\left(\frac{x-a}{\sqrt{2}\ell}\right)$$

- error function term:

$$-2\ell^2\frac{x-a}{\ell} + 2(x-a) = -2\ell(x-a) + 2(x-a) = 2(1-\ell)(x-a)$$

- exponential term:

$$-2\ell^2\sqrt{\frac{2}{\pi}} = -\frac{2^{3/2}\ell^2}{\sqrt{\pi}}$$

- constant:

$$-2\ell^2\left(-\sqrt{\frac{2}{\pi}}\right) = \frac{2^{3/2}\ell^2}{\sqrt{\pi}}$$

An additional constant term is added in the form of $-(x-a)^2\sqrt{\frac{2}{\pi}}$

In summary:

	erf	exp	1
ω_1	$\frac{(x-a)^3}{\ell} - \ell(x-a)$	$\frac{\sqrt{2}(x-a)^2}{\sqrt{\pi}}$	0
ω_2	$-\frac{(x-a)^3}{3\ell}$	$-\frac{\sqrt{2}(x-a)^2}{3\sqrt{\pi}} - \frac{2^{3/2}\ell^2}{3\sqrt{\pi}}$	$\frac{2^{3/2}\ell^2}{3\sqrt{\pi}}$
ω_3	0	$\frac{2^{\frac{3}{2}}\ell^2}{3}$	$-\frac{2^{\frac{3}{2}}\ell^2}{3}$
ω_4	$(x-a)\ell$	$\frac{\sqrt{2}\ell^2}{\sqrt{\pi}}$	$-\frac{\sqrt{2}\ell^2}{\sqrt{\pi}}$
ω_5	$2(x-a)(1-\ell)$	$-\frac{2^{3/2}\ell^2}{\sqrt{\pi}}$	$\frac{2^{3/2}\ell^2}{\sqrt{\pi}}$
			$-(x-a)^2\sqrt{\frac{2}{\pi}}$

This enables us to simplify the expression for ω in the following manner:

$$\begin{aligned}\omega(x,a,\ell,\eta) = \eta^2\ell^2\sqrt{\frac{\pi}{2}}\Bigg[&\left(\frac{2(x-a)^3}{3\ell} + 2(x-a)(1-\ell)\right)\operatorname{erf}\left(\frac{(x-a)}{\sqrt{2}\ell}\right)\\ &+\left(\frac{(2\sqrt{\pi}-5)\ell^2\sqrt{2}}{3\sqrt{\pi}} + \frac{2^{\frac{3}{2}}(x-a)^2}{3\sqrt{\pi}}\right)\exp\left(-\frac{(x-a)^2}{2\ell^2}\right)\\ &+\frac{(5-2\sqrt{\pi})\sqrt{2}\ell^2}{3\sqrt{\pi}} - \frac{(x-a)^2\sqrt{2}}{\sqrt{\pi}}\Bigg]\end{aligned} \tag{A.68}$$

A.5 Experiments - supplementary figures

A.5.1 Full plots from experiment 1

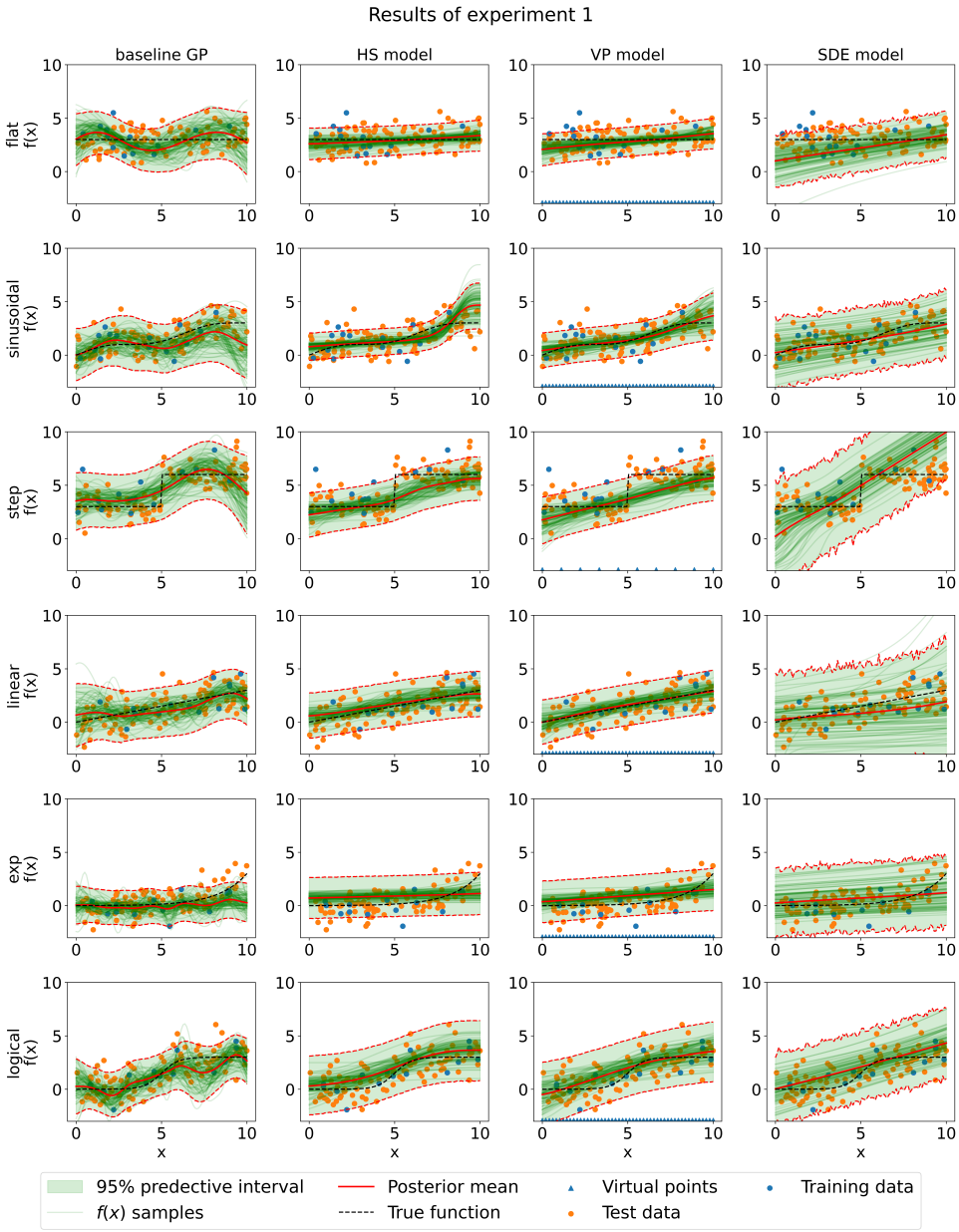


Figure A.1: Plot of experiment 1. The plots show the posterior predictive distributions of the baseline and the monotonic shape-constrained models for all 6 of the benchmark functions.

A.5.2 Full plots from experiment 2

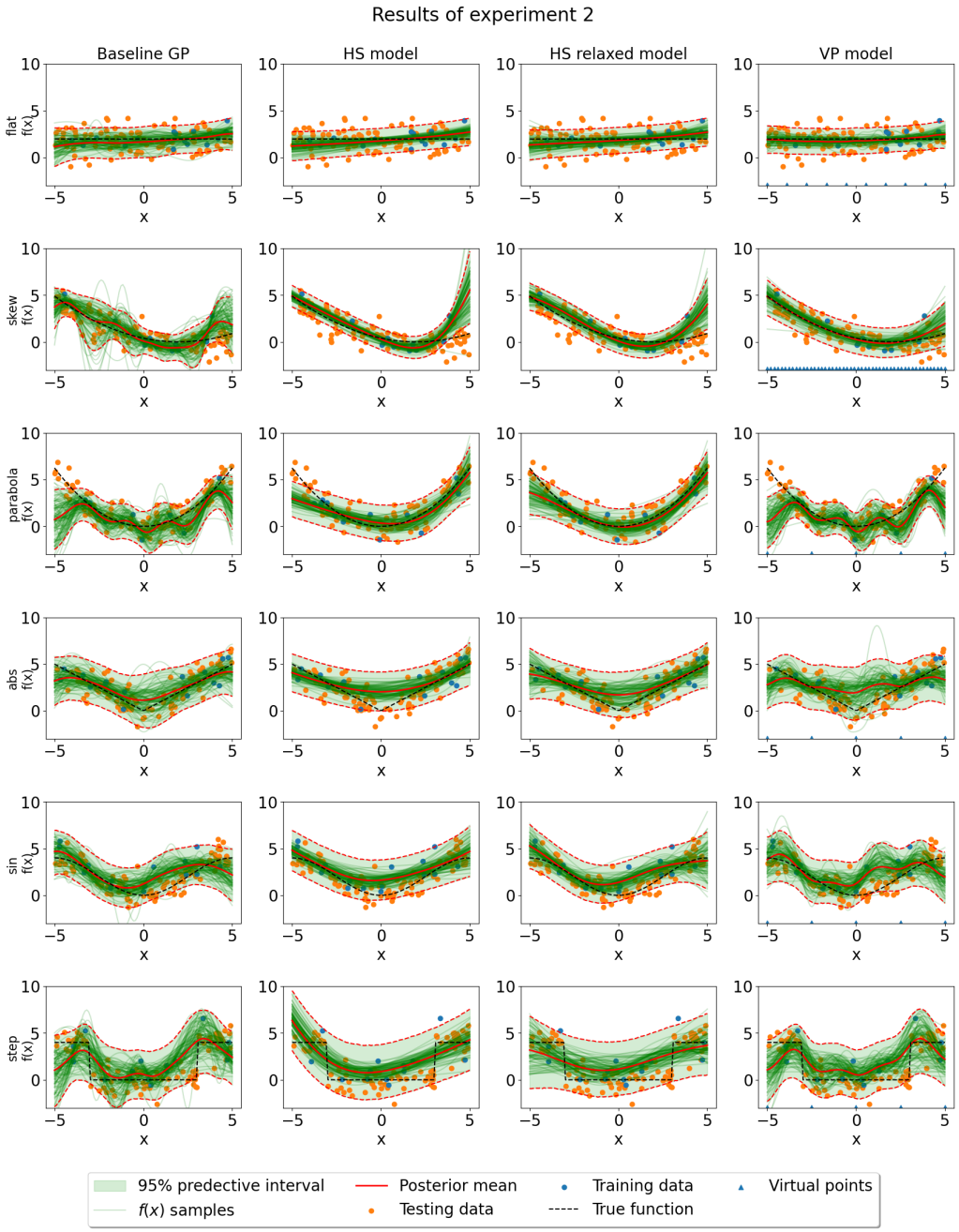


Figure A.2: Plot of experiment 2. The plots show the posterior predictive distributions of the baseline and the u-shaped models for all 6 of the benchmark functions.

A.5.3 Visualisation of contributions in the HS relaxed

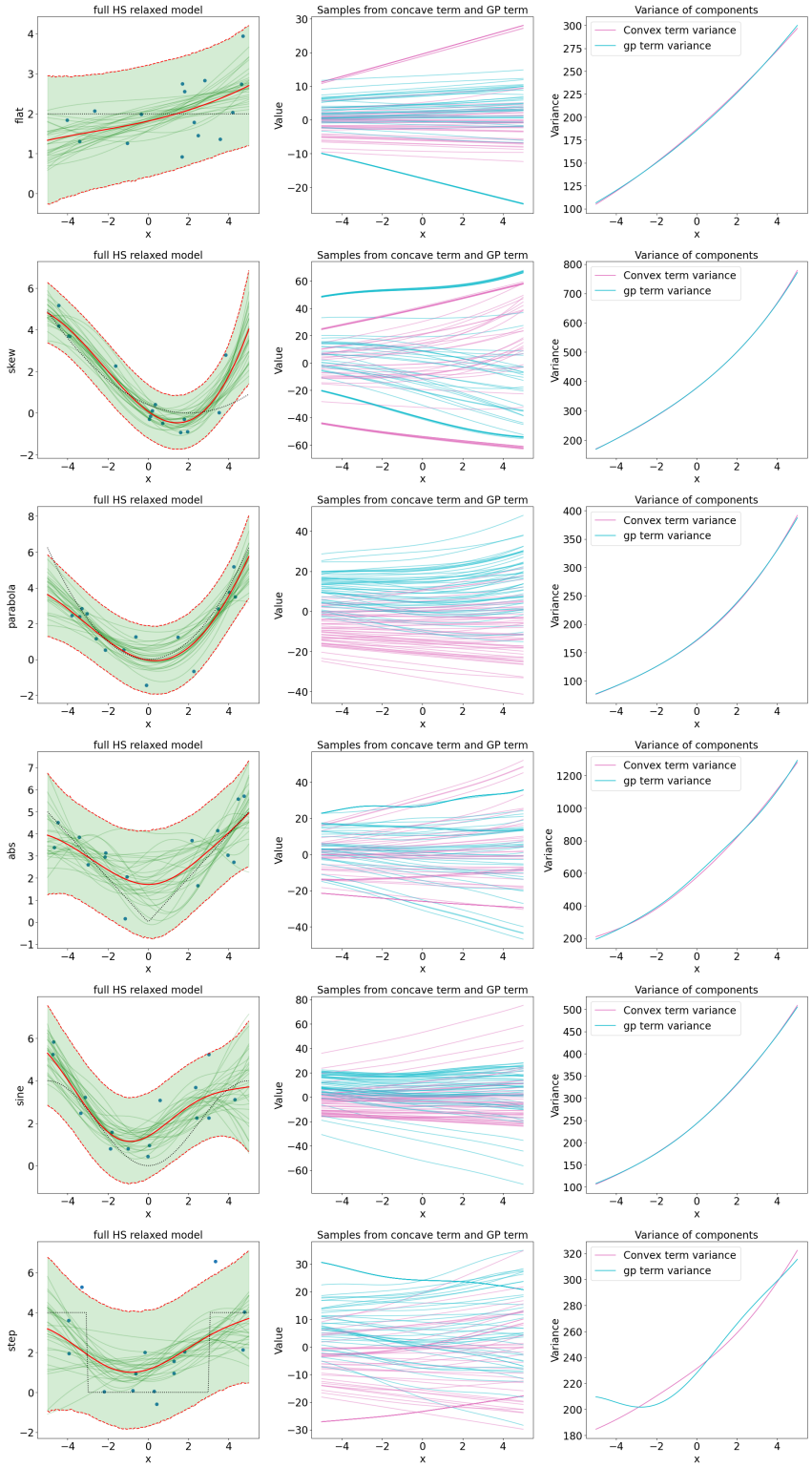


Figure A.3: Visualisation on the contribution from each the two terms in the HS relaxed model for all benchmark function datasets.

A.5.4 Convergence plots for the u-shaped model

Convergence plots similar to the ones discussed in section 4.1.

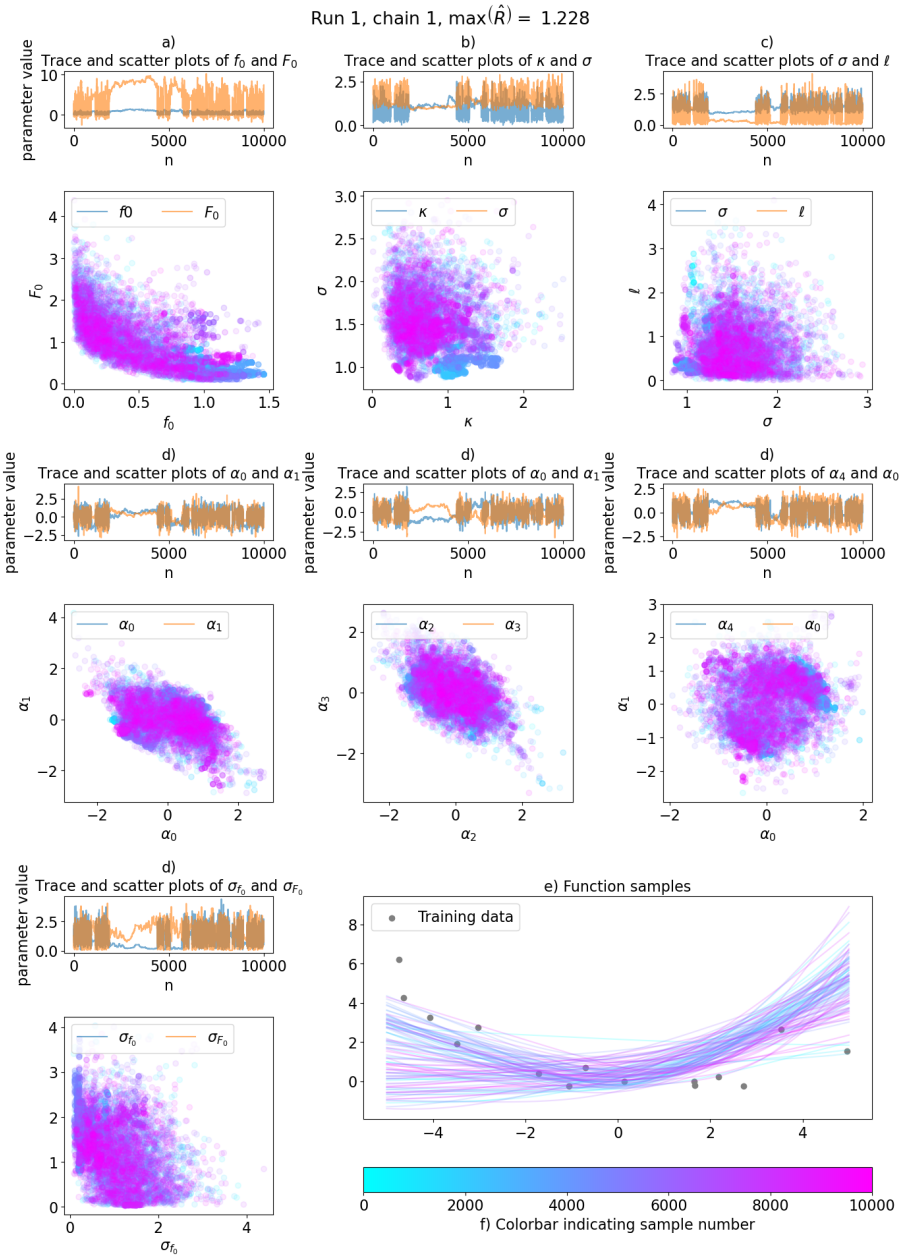


Figure A.4: Convergence diagnostic plot of the HMC samples obtained from fitting the u-shaped model to sinusoidal data (benchmark function 3). This is an example of a chain where the maximal split-$\hat{R}$-value across all parameters is above 1.01.

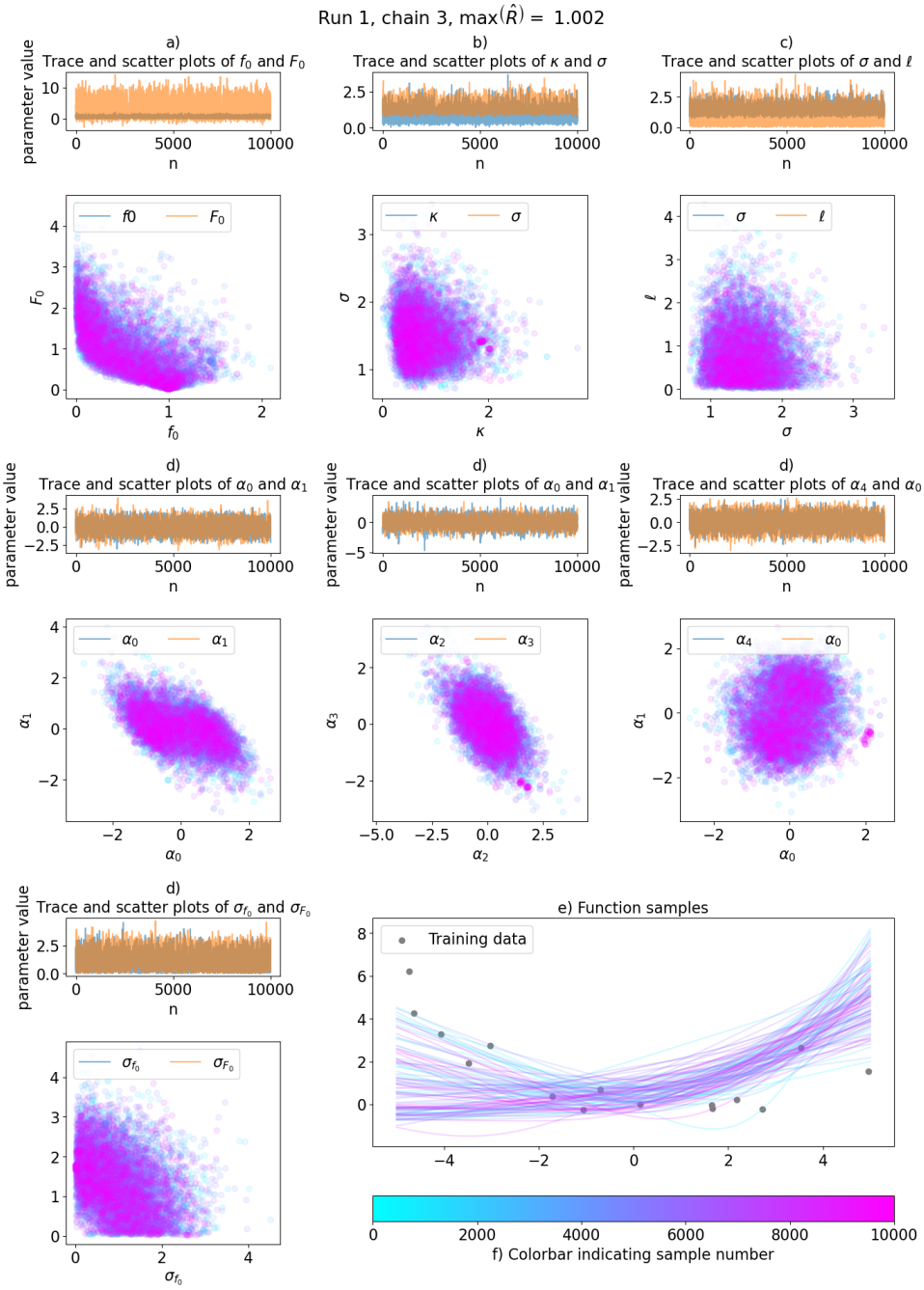


Figure A.5: Convergence diagnostic plot of the HMC samples obtained from fitting the u-shaped model to sinusoidal data (benchmark function 3). This is an example of a chain where the maximal split-$\hat{R}$-value across all parameters is below 1.01.

Technical
University of
Denmark

Asmussens Alle, Build. 322, entrance east
2800 Kgs. Lyngby
Tlf. 4525 1700

www.compute.dtu.dk/