Hilbert space approximate Bayesian Gaussian processes using Stan

Gabriel Riutort-Mayol[1*], Michael R. Andersen[2], Paul-Christian Bürkner[3], Aki Vehtari[2]

[1] Department of Cartographic Engineering, Geodesy, and Photogrammetry, Universitat Politècnica de València, Spain.
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark [3] Department of Computer Science, Aalto University, Finland
[*] Corresponding author, Email: gabriuma@gmail.com

Abstract

Gaussian processes are powerful non-parametric probabilistic models for stochastic functions, however they entail a complexity that is computationally intractable when the number of observations is large, especially when estimated with fully Bayesian methods such as Markov-Chain Monte-Carlo. In this paper, we focus on a novel approach for low-rank approximate Bayesian Gaussian processes, based on a basis functions approximation via Laplace eigenfunctions for stationary covariance functions. The main contribution of this paper is a detailed analysis of the performance and practical implementation of the method in relation to key factors such as the number of basis functions, desired prediction space and smoothness of the function to be learned. Intuitive visualizations and useful recommendations for the values of these factors which help users to improve performance and computation are provided. A diagnosis procedure for whether the chosen values for the number of basis functions and the desired prediction space are adequate to fit to the actual data is also proposed. The proposed method is simple and exhibits an attractive computational complexity due to its linear structure and it is easy to implmenent in probabilistic programming frameworks. Several illustrative examples of the performance and applicability of the method in the probabilistic programming language Stan are presented together with the underlying Stan model code.

Keywords— Gaussian process; Low-rank Gaussian process; Hilbert space methods; Sparse Gaussian process; Bayesian statistics; Stan.

1 Introduction

Gaussian processes (GPs) are flexible statistical models for specifying probability distributions over multi-dimensional non-linear functions [Neal, 1997, Rasmussen & Williams, 2006]. Their name stems from the fact that any finite set of function values is jointly distributed as a multivariate Gaussian. GPs are defined by a mean and a covariance function. The covariance function encodes our prior assumptions about the functional relationship, such as continuity, smoothness, periodicity and scale properties. GPs not only allow for non-linear effects but can also implicitly handle interactions between input variables (covariates). Different types of covariance functions can be combined for further increased flexibility. Due to their generality and flexibility, GPs are of broad interest across machine learning and statistics [Neal, 1997, Rasmussen & Williams, 2006]. Among others, they find application in the fields of spatial epidemiology

[Carlin *et al.* , 2014, Diggle, 2013], robotics and control [Deisenroth *et al.* , 2015], signal processing [Särkkä *et al.* , 2013], neuroimaging [Andersen *et al.* , 2017] as well as Bayesian optimization and probabilistic numerics [Briol *et al.* , 2015, Hennig *et al.* , 2015, Roberts, 2010].

The key element of a GP is the covariance function that defines the dependence structure between function values at different inputs. However, evaluating the covariance function **[evaluating the covariance function is not a problem, but computing the posterior is]** comes with a computational issue because of the need of inverting its Gram matrix to optimize the hyperparameters. Given n observations in the data, the computational complexity and memory requirements of computing the posterior distribution for a GP in general scale as $O(n^3)$ and $O(n^2)$, respectively. This limits their application to rather small data sets of a few tens of thousands observations at most. The problem becomes more severe when performing full Bayesian inference via sampling methods, where in each sampling step we need $O(n^3)$ computations when inverting the Gram matrix of the covariance function, usually through Cholesky factorization. To alleviate these computational demands, several approximate methods have been proposed.

Sparse GPs are based on low-rank approximations of the covariance matrix. The low-rank approximation with $m \ll n$ *inducing points* implies reduced memory requirements of $O(nm)$ and corresponding computational complexity of $O(nm^2)$. A unifying view on sparse GPs based on approximate generative methods is provided in Quiñonero-Candela & Rasmussen [2005], while a general review can be found in Rasmussen & Williams [2006]. Burt *et al.* [2019] show that for regression with normally distributed covariates in D dimensions and using the squared exponential covariance function, $M = O(\log^D N)$ is sufficient for an accurate approximation.

An alternative class of low-rank approximations is based on forming a basis function approximation with $m \ll n$ basis functions. The basis functions are usually presented explicitly, but can also be used to form a low rank covariance matrix approximation. Common basis function approximations rest on the spectral analysis and series expansions of GPs [Adler, 1981, Cramér & Leadbetter, 2013, Loève, 1977, Trees, 1968]. Sparse spectrum GPs are based on a sparse approximation to the frequency domain representation of a GP [Gal & Turner, 2015, Lázaro Gredilla, 2010, Quiñonero-Candela *et al.* , 2010]. Recently, Hensman *et al.* [2017] presented a variational Fourier feature approximation for GPs that was derived for the Matérn class of kernels. Another related method for approximating kernels relies on random Fourier features [Rahimi & Recht, 2008, 2009]. Further, certain spline smoothing basis functions are equivalent to GPs with certain covariance functions [Furrer & Nychka, 2007, Wahba, 1990]. Furthermore, a recent related work based on a spectral representation of GPs as an infinite series expansion with the Karhunen-Loève representation (see, e.g. Grenander [1981]) can be found in Jo *et al.* [2019].

In this paper we propose and evaluate a new framework for fast and accurate inference for fully Bayesian GPs using basis function approximations. We focus on the basis function approximation via Laplace eigenfunctions for stationary covariance functions proposed by Solin & Särkkä [2018]. Using a basis function expansion, a GP is approximated with a linear model which makes inference considerably faster. The linear model structure makes GPs easy to implement as a building block in more complicated models in modular probabilistic programming frameworks, where there is a big benefit if the approximation specific computation is simple. Furthermore, a linear representation of a GP makes it easier to be used as latent function in non-Gaussian observational models allowing for more modelling flexibility. The basis function approximation via Laplace eigenfunctions can be made arbitrary accurate and the trade-off between computational complexity and approximation accuracy can easily be controlled.

The Laplace eigenfunctions can be computed analytically and they are independent of the particular choice of the covariance function including the hyperparameters. While the pre-computation cost of the basis functions is $O(m^2n)$, the computational cost of learning the covariance function parameters is $O(mn + m)$ in every step of the optimizer or sampler. This is a big advantage in terms of speed for iterative algorithms such as Markov chain Monte Carlo (MCMC). Another advantage is the reduced memory requirements of automatic differentation methods used in modern probabilistic programming frameworks, such as Stan [Carpenter *et al.* , 2017], WinBUGS [Lunn *et al.* , 2000] and others. This is because the memory requirements of automatic differentation rather scale with the computational complexity instead of with the usual memory requirements for the posterior density computation **[I don't understand the sentence about autodiff and computational complexity]**. The basis function approach also provides an easy way to apply a non-centered parameterization of GPs, which reduces the posterior dependency between parameters representing the estimated function and the hyperparameters of the covariance function, which further improves MCMC efficiency.

While Solin & Särkkä [2018] have fully developed the mathematical theory behind this specific approximation of GPs, further work is needed for its practical implementation in probabilistic programming frameworks. In this paper, the interactions among the key factors of the method such as the number of basis functions, desired prediction space, or properties of the true functional relationship between covariates and response variable, are investigated and analyzed in detail in relation to the performance and accuracy of the method. Practical recommendations are given for the values of the key factors based on intuitive graphical summaries that encode the recognized relationships. Our recommendations will help users to choose valid and optimized values for these factors, improving performance, and saving time of computation. A diagnosis of whether the chosen values for the number of basis functions and the desired prediction space are adequate to fit to the actual data is proposed.

We have implemented the approach in the probabilistic programming language Stan [Carpenter *et al.* , 2017] as well as subsequently in the *brms* package [Bürkner *et al.* , 2017] of the R software [R Core Team, 2019]. Several illustrative examples of the performance and applicability of the method are shown using both simulated and real datasets. All examples are accompanied by the corresponding Stan code.

Although there are several GP specific software packages available to date, for example, GPML [Rasmussen & Nickisch, 2010], GPstuff [Vanhatalo *et al.* , 2013], GPy [GPy, 2012], and GPflow [Matthews *et al.* , 2017]), each provide efficient implementations only for a restricted range of GP-based models. In this paper, we do not focus on the fastest possible inference for some specific GP models, but instead are interested in how GPs can be easily used as modular components in probabilistic programming frameworks.

The remainder of the paper is structured as follows. In Section 2, we introduce GPs, covariance functions and spectral density functions. In Section 3, the reduced rank approximation to GPs proposed by Solin & Särkkä [2018] is described. In Section 4, the accuracy of these approximations under several conditions using analytical and numerical methods is analyzed. Several case studies in which we fit exact and approximate GPs to real and simulated data are provided in Section 5. A brief conclusion of the work is made in Section 6. Appendix A includes a brief presentation of the mathematical details behind this Hilbert space approximation of a stationary covariance function, and Appendix B presents a low-rank representation of a GP for the particular case of a periodic covariance function. Online supplemental material with more case studies illustrating the performance and applicability of the method can be found at `https://github.com/gabriuma/basis_functions_approach_to_GP` in the subfolder `Paper/online_supplemental_material`.

2 Gaussian process as a prior

A GP is a stochastic process which defines the distribution over a collection of random variables indexed by a continuous variable, i.e. $\{f(t) : t \in \mathcal{T}\}$ for some index set $\mathcal{T}$. GPs have the defining property that the marginal distribution of any finite subset of random variables, $\{f(t_1), f(t_2), \dots, f(t_K)\}$, is a multivariate Gaussian distribution.

In this work, GPs will take the role of a prior distribution over function spaces for non-parametric latent functions in a Bayesian setting. Consider a data set $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where y_n is modelled conditionally as $p(y_n|f(\boldsymbol{x}_n), \phi)$, where p is some parametric distribution with parameters ϕ, and f is an unknown function with GP prior, which depends on an input $\boldsymbol{x}_n \in {\rm I\!R}^D$. This generalizes readily to more complex models depending on several unknown functions, for example such as $p(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ or multilevel models.

Our goal is to obtain the posterior distribution for the value of the function $\tilde{f} = f(\tilde{\boldsymbol{x}})$ evaluated at a new input point $\tilde{\boldsymbol{x}}$.

We assume a GP prior for $f \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, where $\mu : \mathrm{I\!R}^D \to \mathrm{I\!R}$ and $k : \mathrm{I\!R}^D \times \mathrm{I\!R}^D \to \mathrm{I\!R}$ are the mean and covariance functions, respectively,

$$\begin{aligned} \mu(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})]\,, \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}[(f(\boldsymbol{x}) - \mu(\boldsymbol{x}))\,(f(\boldsymbol{x}') - \mu(\boldsymbol{x}'))]\,. \end{aligned}$$

The mean and covariance functions completely characterize the GP prior, and control the a priori behavior of the function f. Let $\boldsymbol{f} = \{f(\boldsymbol{x}_n)\}_{n=1}^N$, then the resulting prior distribution for $\boldsymbol{f}$ is a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{\mu} = \{\mu(\boldsymbol{x}_n)\}_{n=1}^N$ is the mean and $\boldsymbol{K}$ the covariance matrix, where $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In the following, we focus on zero-mean Gaussian processes, that is set $\mu(\boldsymbol{x}) = 0$. The covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ might depend on a set of hyperparameters, $\boldsymbol{\theta}$, but we will not write this dependency explicitly to ease the notation. The joint distribution of $\boldsymbol{f}$ and a new $\tilde{f}$ is also a multivariate Gaussian as,

$$p(\boldsymbol{f}, \tilde{f}) = \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{f} \\ f^* \end{array} \right] \middle| \boldsymbol{0}, \left[\begin{array}{cc} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{k}_{\boldsymbol{f},\tilde{f}} \\ \boldsymbol{k}_{\tilde{f},\boldsymbol{f}} & k_{\tilde{f},\tilde{f}} \end{array} \right] \right),$$

where $\boldsymbol{k}_{\boldsymbol{f},\tilde{f}}$ is the covariance between $\boldsymbol{f}$ and $\tilde{f}$, and $k_{\tilde{f},\tilde{f}}$ is the prior variance of $\tilde{f}$.

If $p(y_n|f(\boldsymbol{x}_n), \phi) = \mathcal{N}(y_n|f(\boldsymbol{x}_n), \sigma)$ then $\boldsymbol{f}$ can be integrated out analytically (with a computational cost of $O(n^3)$ for exact GPs and $O(nm^2)$ for sparse GPs). If $p(y_n|f(\boldsymbol{x}_n), \phi) = \mathcal{N}(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ **[Is something missing here? The function g is missing on the left hand side.]** or $p(y_n|f(\boldsymbol{x}_n), \phi)$ is non-Gaussian, the marginalization does not have a closed form solution. Furthermore, if a prior distribution is imposed on ϕ and $\boldsymbol{\theta}$ to form a joint posterior for ϕ, $\boldsymbol{\theta}$ and $\boldsymbol{f}$, approximate inference such as Markov chain Monte Carlo (MCMC) [Brooks *et al.* , 2011], Laplace approximation ([Rasmussen & Williams, 2006, Williams & Barber, 1998], expectation propagation [Minka, 2001], or variational Bayes methods [Csató *et al.* , 2000, Gibbs & MacKay, 2000] are required. In this paper, we focus on the use of MCMC for integrating over the joint posterior. MCMC is usually not the fastest approach, but it is flexible and allows accurate inference and uncertainty estimates for general models in probabilistic programming settings. We consider the computational costs of GPs specifically from this point of view.

2.1 Covariance functions and spectral density

The covariance function is the crucial ingredient in a GP as it encodes our prior assumptions about the function, and defines a correlation structure which characterizes the correlations between function values at different inputs. A covariance function needs be symmetric and positive semi-definite [Rasmussen & Williams, 2006]. A stationary covariance function is a function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' \in \mathrm{I\!R}^D$, such that it can be written $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau})$, which means that the covariance is invariant to translations. Isotropic covariance functions are only depends on the input points through the norm of the difference, $k(\boldsymbol{x}, \boldsymbol{x}') = k(|\boldsymbol{x} - \boldsymbol{x}'|) = k(r), r \in \mathrm{I\!R}$, which means that the covariance is both translation and rotation invariant. The most commonly used distance between observations is the L2-norm ($|\boldsymbol{x} - \boldsymbol{x}'|_{L2}$), also known as Euclidean distance, although other types of distances can be considered.

The Matérn class of isotropic covariance functions is given by,

$$k_\nu(r) = \alpha \, \frac{2^{1-\nu}}{\Gamma(\nu)} \Big(\frac{\sqrt{2\nu} r}{\ell} \Big)^{\nu} K_\nu \Big(\frac{\sqrt{2\nu} r}{\ell} \Big),$$

where $\nu > 0$ is the order the kernel, K_ν the modified Bessel function of the second kind, and the $\ell > 0$ and $\alpha > 0$ are the length-scale and magnitude (marginal variance), respectively, of the kernel. The particular case where $\nu = \infty$ and $\nu = 3/2$ are probably the most commonly used kernels [Rasmussen & Williams, 2006],

$$\begin{aligned} k_\infty(r) &= \alpha \exp\Big(-\frac{1}{2}\frac{r^2}{\ell^2}\Big), \\ k_{\frac{3}{2}}(r) &= \alpha\Big(1 + \frac{\sqrt{3}r}{\ell}\Big)\exp\Big(-\frac{\sqrt{3}r}{\ell}\Big). \end{aligned}$$

The former is commonly known as the squared exponential (exponentiated quadratic) covariance function. Assuming the Euclidean distance between observations, $r = |\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{\sum_{i=1}^{D}(x_i - x_i')^2}$, the kernels written above take the form

$$\begin{aligned} k_\infty(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}) &= \alpha \exp\Big(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i - x_i')^2}{\ell_i^2}\Big), \\ k_{\frac{3}{2}}(|\boldsymbol{x} - \boldsymbol{x}'|_{L2}) &= \alpha\Bigg(1 + \sqrt{\sum_{i=1}^{D}\frac{3(x_i - x_i')^2}{\ell_i^2}}\Bigg)\exp\Bigg(-\sqrt{\sum_{i=1}^{D}\frac{3(x_i - x_i')^2}{\ell_i^2}}\Bigg). \end{aligned}$$

Notice that the previous expressions have been easily generalized to using a multidimensional length-scale $\boldsymbol{\ell} \in \mathbb{R}^D$. Using individual length-scales for each dimension turns the isotropic covariance function into a non-isotropic covariance function. That is, for a non-isotropic covariance function, the smoothness may vary across different input dimensions.

Stationary covariance functions can be represented in terms of their spectral densities [Rasmussen & Williams, 2006]. In this sense, the covariance function of a stationary process can be represented as the Fourier transform of a positive finite measure (*Bochner's theorem*, see, e.g. Akhiezer & Glazman [1993]). If this measure has a density, it is known as the spectral density of the covariance function, and the covariance function and the spectral density are Fourier duals, known as the *Wiener-Khintchine theorem* [Rasmussen & Williams, 2006]. The spectral density functions associated with the Matérn class of covariance functions is given by

$$S_\nu(\omega) = \alpha\,\frac{2^D\pi^{D/2}\Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\,l^{2\nu}}\Big(\frac{2\nu}{l^2} + 4\pi^2\omega^2\Big)^{(\nu + D/2)}$$

in D dimensions, where variable $\omega \in \mathbb{R}$ denotes the frequency, and ℓ and α are the lengthscale and magnitude (marginal variance), respectively, of the kernel. The particular cases where $\nu = \infty$ and $\nu = 3/2$ take the form

$$S_\infty(\omega) = \alpha\sqrt{2\pi}^D\,\ell^D\exp(-0.5\ell^2\omega^2), \tag{1}$$

$$S_{\frac{3}{2}}(\omega) = \alpha\,\frac{2^D\pi^{D/2}\Gamma(\frac{D+3}{2})\sqrt{3}^3}{\frac{1}{2}\sqrt{\pi}\,\ell^3}\Big(\frac{3}{\ell^2} + \omega^2\Big)^{-\frac{D+3}{2}}. \tag{2}$$

[Is it only purpose that only some equations have numbers?] For input dimension $D = 3$ and Euclidean distance $\omega = \sqrt{\sum_{i=1}^{3}s_i^2}$, and considering a multidimensional lengthscale $\boldsymbol{\ell} \in \mathbb{R}^3$, the spectral densities written above take the form

$$\begin{aligned} S_\infty(\omega) &= \alpha\sqrt{2\pi}^3\prod_{i=1}^{3}\ell_i\exp\Big(-\frac{1}{2}\sum_{i=1}^{3}\ell_i^2 s_i^2\Big), \\ S_{\frac{3}{2}}(\omega) &= \alpha\,32\pi\sqrt{3}^3\prod_{i=1}^{3}\ell_i\big(3 + \sum_{i=1}^{3}\ell_i^2 s_i^2\big)^{-3}. \end{aligned}$$

3 Hilbert space approximate Gaussian process model

The approximate GP method, developed by Solin & Särkkä [2018] and implemented in this paper, is based on considering the covariance operator of a stationary covariance function as a pseudo-differential operator constructed as a series of Laplace operators. Then, the pseudo-differential operator is approximated with Hilbert space methods on a compact subset $\Omega \subset {\rm I\!R}^D$ subject to boundary conditions. For brevity, we will refer to these approximate Gaussian processes as HSGPs. Below, we will present the main results around HSGPs relevant for practical applications. More details and mathematical proofs are provided in Solin & Särkkä [2018]. Our starting point for presenting the method is the main result obtained by Solin & Särkkä [2018] of the definition of the covariance function as a series expansion of eigenvalues and eigenfunctions of the Laplacian operator. The mathematical details of this approximation are briefly presented in Appendix A.

We begin by focusing on the case of a unidimensional input space (i.e. on GPs with just a single covariate) such that $\Omega \in [-L, L] \subset {\rm I\!R}$, where L is some positive real number to which we also refer as boundary condition. As Ω describes the interval in which the approximations are valid, L plays a critical role in the accuracy of HSGPs. We will come back to this issue in Section 4.

Within Ω, we can write any stationary covariance function with input values $\{x, x'\} \in \Omega$ as

$$k(x, x') = \sum_{j=1}^{\infty} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x'), \tag{3}$$

where S_θ is the spectral density of the stationary covariance function k (see Section 2.1) and θ the set of hyperparameters of k [Rasmussen & Williams, 2006]. The terms $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(x)\}_{j=1}^{\infty}$ are the sets of eigenvalues and eigenvectors, respectively, of the Laplacian operator in the given domain Ω. Namely, they satisfy the following eigenvalue problem in Ω when applying the Dirichlet boundary condition (other boundary conditions could be used as well)

$$\begin{aligned} -\nabla^2 \phi_j(x) &= \lambda \phi_j(x), \qquad & x \in \Omega \\ \phi_j(x) &= 0, \qquad & x \notin \Omega. \end{aligned} \tag{4}$$

The eigenvalues $\lambda_j > 0$ are real and positive because the Laplacian is a positive definite Hermitian operator, and the eigenfunctions ϕ_j for the eigenvalue problem in equation (4) are sinusoidal functions. The solution to the eigenvalue problem is independent of the specific choice of covariance function and is given by

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2, \tag{5}$$

$$\phi_j(x) = \sqrt{\frac{1}{L}}\, \sin\!\left(\sqrt{\lambda_j}(x + L)\right). \tag{6}$$

If we truncate the sum in eq. (3) to the first m terms, the approximate covariance function becomes

$$k(x, x') \approx \sum_{j=1}^{m} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x') = \boldsymbol{\phi}(x)^\intercal \Delta\, \boldsymbol{\phi}(x'),$$

where $\boldsymbol{\phi}(x) = \{\phi_j(x)\}_{j=1}^{m} \in {\rm I\!R}^m$ is the column vector of basis functions, and $\Delta \in {\rm I\!R}^{m \times m}$ is a diagonal matrix of the spectral density evaluated at the square root of the eigenvalues, i.e. $S_\theta(\sqrt{\lambda_j})$,

$$\Delta = \begin{bmatrix} S_\theta(\sqrt{\lambda_1}) & & \\ & \ddots & \\ & & S_\theta(\sqrt{\lambda_m}) \end{bmatrix}.$$

Thus, the Gram matrix K of the covariance function k for a set of observations $i = 1, \ldots, n$ and corresponding input values $\{x_i\}_{i=1}^n \in \Omega^n$ can be represented as

$$\mathrm{K} = \Phi \Delta \Phi^\intercal,$$

where $\Phi \in \mathbb{R}^{n \times m}$ is the matrix of eigenfunctions $\phi_j(x_i)$

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_m(x_n) \end{bmatrix}.$$

As a result, the model for f can be written as

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \Phi \Delta \Phi^\intercal).$$

This equivalently leads to a linear representation of f via

$$f(x) \approx \sum_{j}^{m} \left(S_\theta(\sqrt{\lambda_j})\right)^{\frac{1}{2}} \phi_j(x) \beta_j, \tag{7}$$

where $\beta_j \sim \mathrm{Normal}(0, 1)$. Thus, the function f is approximated with a finite basis function expansion (using the eigenfunctions ϕ_j of the Laplace operator), scaled by the square root of spectral density values. A key property of this approximation is that the eigenfunctions ϕ_j do not depend on the hyperparameters of the covariance function θ. Instead, the only dependence of the model on θ is through the spectral density S_θ. The eigenvalues λ_j are monotonically increasing with j and S_θ goes rapidly to zero for bounded covariance functions. Therefore, eq. (7) can be expected to be a good approximation for a finite number of m terms in the series as long as the inputs values x_i are not too close to the boundaries $-L$ and L of Ω. The computational cost of univariate HSGPs scales as $O(nm + m)$, where n is the number of observations and m the number of basis functions.

The parameterization in eq. (7) is naturally in the non-centered parameterization form with independent prior distribution on β_j, which makes the posterior inference easier. Furthermore, all dependencies on the covariance function and the hyperparameters is through the prior distribution of the regression weights β_j. The posterior distribution of the parameters $p(\boldsymbol{\beta}|\boldsymbol{y})$ is a distribution over a m-dimensional space, where m is much smaller than the number of observations n. Therefore, the parameter space is greatly reduced and this makes inference faster, especially when sampling methods are used.

3.1 Generalization to multidimensional GPs

The results from the previous section can be generalizes to a multidimensional input space with compact support, $\Omega = [-L_1, L_1] \times \cdots \times [-L_d, L_d]$ and Dirichlet boundary conditions. In a D-dimensional input space, the total number of eigenfunctions and eigenvalues in the approximation is equal to the number of D-tuples, that is possible combinations of univariate eigenfunctions over all dimensions. The number of D-tuples is given by

$$m^* = \prod_{d=1}^{D} m_d, \tag{8}$$

where m_d is the number of basis function for the dimension d. Let $\mathbb{S} \in \mathbb{N}^{m^* \times D}$ be the matrix of all those D-tuples. For example, suppose we have $D = 3$ dimensions and use $m_1 = 2$, $m_2 = 2$ and $m_3 = 3$

eigenfunctions and eigenvalues for the first, second and third dimension, respectively. Then, the number of multivariate eigenfunctions and eigenvalues is $m^* = m_1 \cdot m_2 \cdot m_3 = 12$ and the matrix $\mathbb{S} \in \mathbb{N}^{12\times 3}$ is given by

$$\mathbb{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}.$$

Each multivariate eigenfunction ϕ^*_j corresponds to the product of the univariate eigenfunctions whose indices corresponds to the elements of the D-tuple $\mathbb{S}_{j\cdot}$, and each multivariate eigenvalue $\boldsymbol{\lambda}^*_j$ is a D-vector with elements that are the univariate eigenvalues whose indices correspond to the elements of the D-tuple $\mathbb{S}_{j\cdot}$. Thus, for $\boldsymbol{x} = \{x_d\}_{d=1}^D \in \Omega$ and $j = 1, \dots, m^*$, we have

$$\boldsymbol{\lambda}^*_j = \left\{\lambda_{\mathbb{S}_{jd}}\right\}_{d=1}^D = \left\{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\right\}_{d=1}^D, \tag{9}$$

$$\phi^*_j(\boldsymbol{x}) = \prod_{d=1}^D \phi_{\mathbb{S}_{jd}}(x_d) = \prod_{d=1}^D \sqrt{\frac{1}{L_d}} \sin\!\Big(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\Big). \tag{10}$$

The approximate covariance function is then represented as

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j=1}^{m^*} S^*_\theta\Big(\sqrt{\boldsymbol{\lambda}^*_j}\Big)\phi^*_j(\boldsymbol{x})\phi^*_j(\boldsymbol{x}'), \tag{11}$$

where S^*_θ is the spectral density of the D-dimensional covariance function (see Section 2.1). We can now write the approximate series expansion of the multivariate function f as

$$f(\boldsymbol{x}) \approx \sum_{j=1}^{m^*} \Big(S^*_\theta\Big(\sqrt{\boldsymbol{\lambda}^*_j}\Big)\Big)^{\frac{1}{2}} \phi^*_j(\boldsymbol{x})\beta_j, \tag{12}$$

where, again, $\beta_j \sim \text{Normal}(0, 1)$. The computational cost, in learning the covariance function hyperparameters, of multivariate HSGPs scales as $O(nm^* + m^*)$, where n is the number of observations and m^* is the number of multivariate basis functions **[I dont think the first of part of this sentence is correct. Do you mean the cost of evaluating the joint distribution in a single iteration rather than ”the cost of learnaing the covariance function hyperparameters?]**. Although this still implies linear scaling in n, the approximation is more costly than in the univariate case, as m^* is the product of the number of univariate basis functions over the input dimensions and grows exponentially with respect to the number of dimensions.

4 The accuracy of the approximation

The accuracy and speed of the HSGP model depends on several interrelated factors, most notably on the number of basis functions and on the boundary condition of the Laplace eigenfunctions. Furthermore, appropriate values for these factors will depend on the degree of non-linearity of the estimated function, which is in turn characterized by the lengthscale of the covariance function. In this section, we analyze the effects of the number of basis functions and the boundary condition on the approximation accuracy. We present recommendations on how they should be chosen and diagnostics to check the accuracy of the obtained approximation.

Ultimately, these recommendations are based on the relationships among the number of basis functions, the boundary condition and the lengthscale of the function, which depend on the particular choice of the kernel function. In this work we investigate these relationships for the square exponential covariance function and Matérn (ν=3/2) covariance function in the present section, and for the periodic squared exponential covariance function in Appendix B. For other kernels, the relationships will be slightly different, in function of mainly the smoothness or wigglyness of the kernel effects. **[The last sentence is not very clear.]**

4.1 Dependency on the number of basis functions and the boundary condition

As explained in Section 3, the approximation of the covariance function is a series expansion of eigenfunctions and eigenvalues of the Laplace operator in a given domain Ω, for instance in a one-dimensional input space $\Omega = [-L, L] \subset {\rm I\!R}$

$$k(\tau) = \sum_{j=1}^{\infty} S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0),$$

where L describes the boundary condition, j is the index for the eigenfunctions and eigenvalues, and $\tau = x - x'$ is the difference between two covariate values x and x' in Ω. The eigenvalues λ_j and eigenfunctions ϕ_j are given in equations (5) and (6) for the unidimensional case and in equations (9) and (10) for the multidimensional case. The number of basis functions can be truncated at some finite positive value m such that the difference between the densities of the exact and approximate covariance functions is less than a predefined threshold $\varepsilon > 0$

$$\Big(\int k(\tau)\mathrm{d}\tau - \int \sum_{j=1}^{m} S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0)\mathrm{d}\tau\Big) < \varepsilon. \tag{13}$$

[I think the equation above is wrong, it should compared the areas under the curves and not the difference between the exact and approximate covariance function. It should the total variation difference between the true kernel and the approximate kernel, i.e. something like

$$\int |k(\tau) - S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0)\mathrm{d}\tau|\mathrm{d}\tau$$

. How did you compute it in the code?]

The specific numbe of basis functions m in the approximation needed to satisfy equation (13) depends on the degree of non-linearity of the function to be estimated, that is on its lengthscale ℓ, which constitutes a hyperparameter of the GP. The approximation also depends on the boundary L (see equations (5), (6), (9) and (10)), which will affect its accuracy especially near the boundaries. As we will see later on, L will also influence the number of basis functions required in the approximation. In the present paper, we will set L an extension of the desired covariate input domain $\Psi = \max_i |x_i|$. Without loss of generality, we can assume Ψ to be symmetric around zero, that is $\Psi = [-S, S] \subset \Omega$. We now define L as

$$L = c \cdot S, \tag{14}$$

where S (for $S > 0$) represents the half-range of the input space, and $c \geq 1$ is the proportional extension factor. In the following, we will refer to c as the boundary factor of the approximation. The boundary factor can also be regarded as the boundary L normalized by the half-range S of the input space.

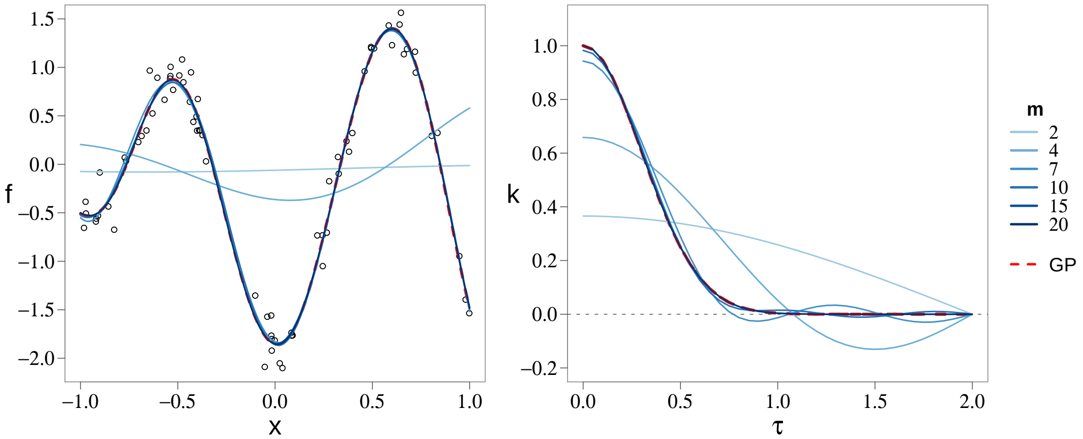

Figure 1: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model (dashed red line) and the HSGP model for different number of basis functions m, with the boundary factor fixed to a large enough value. Notice that the dashed red line of the regular GP can hardly be seen in the plots because it is under the blue lines.

We start by illustrating how the number of basis functions m and boundary factor c influence the accuracy of the HSGP approximations individually. For this purpose, a set of noisy observations are drawn from an exact GP model with lengthscale $\ell = 0.3$ and marginal variance $\alpha = 1$ of the covariance function **[which covarariance function?]**, using input values from the zero-mean input domain with half-range $S = 1$. Several HSGP models with varying m and L are fitted to this data. In this example, the lengthscale and marginal variance parameters used in the HSGPs are fixed to the true values of the data-generating model. Figures 1 and 2 illustrate the individual effects of m and c, respectively, on the posterior predictions of the estimated function and on the covariance function itself. For c fixed to a large enough value, Figure 1 shows clearly how m affects the accuracy on the approximation and the non-linearity of the estimated function, in the sense that fewer basis functions inaccurately imply larger lengthscales and consequently more linear functional forms. The higher the "wigglyness" of the function to be estimated, the more basis functions will be required. If m fixed to a large enough value, Figure 2 shows that c mainly affects the approximation near the boundaries.

Next, we analyze how the interaction effects between m and c affects the quality of the approximation. The lengthscale and marginal variance of the covariance function will no longer be fixed but instead estimated in both regular GP and HSGP models. **[Specify how the hyperparameters are handled]** Figure 3 shows the posterior predictive mean of the function and the covariance function obtained after fitting the data for varying m and c. Figure 4 shows the root mean square error (RMSE) of the HSGP models computed against the regular GP model. Figure 5 shows the estimated lengthscale and marginal variance for the regular GP model and the HSGP models. Looking at the RMSEs in Figure 4, we can conclude that the optimal choice in terms of precision and computations would be $m = 15$ basis functions and a boundary factor between $c = 1.5$ and $c = 2.5$. Further, the less conservative choice of $m = 10$ and $c = 1.5$ could also produce a sufficiently accurately approximation depending on the application. We may also come to the same conclusion by looking at the posterior predictions and covariance function plots in Figure 3. From these results, some general conclusions may be drawn:

- As c increases, m has to increase as well (and vice versa). This is consistent with the expression for the eigenvalues in eq. (5), where L appears in the denominator .
- There exists a minimum c below which a accurate approximation will never be achieved regardless of the number of basis functions m.

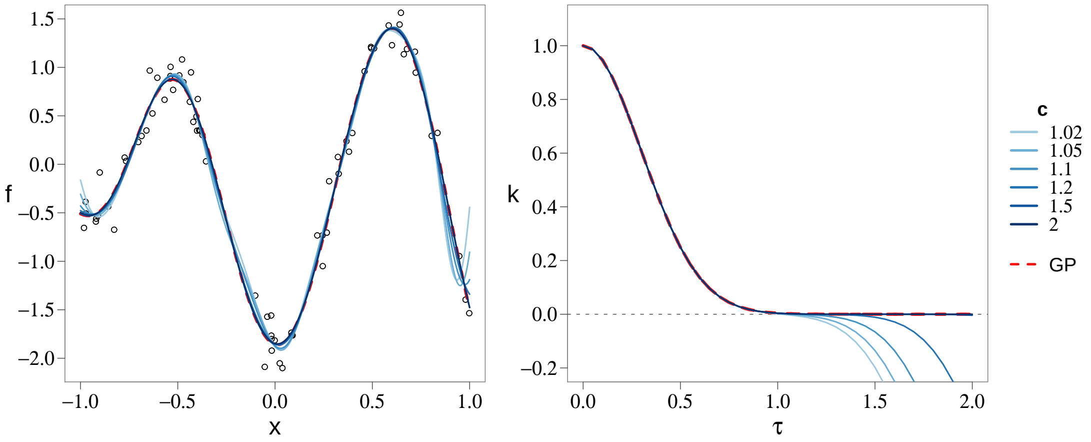

Figure 2: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model (dashed red line) and the HSGP model for different values of the boundary factor c, with a large enough fixed number of basis functions. Notice that the dashed red line of the regular GP can hardly be seen in the plots because it is under the blue lines.

Additionally, there is a clear relation between the number of basis functions m and the boundary factor c with the lengthscale ℓ of the approximated function. Figures 6 and 7 depict how these three factors interact with each other in relation to a close approximation of the HSGP model, in the cases of a GP with square exponential covariance function and Matérn(ν=3/2) covariance function, respectively, and a single input dimension. More precisely, for a given GP model (with a square exponential covariance function) with lengthscale ℓ and given a boundary factor c, Figure 6 shows the minimum m required to achieve a accurate approximation in the sense of satisfying equation (13). Similarly for Figure 7 in the case of a Matérn(ν=3/2) covariance function. We considered an approximation to be a close enough when the difference between densities of the approximate covariance function and the exact covariance function, ε in equation (13), is below 1% of the density of the exact covariance function **[1) I don't fully understand the statement in the sentence about. Which density are you referring to? 2) I also think the is a problem with the equation below. The integral $\int k(\tau)\mathrm{d}\tau$ is just a constant (for fixed hyperparameters). Furthermore, we need to use different symbols to represent the exact covariance function and the approximate covariance function, you could use k and $\tilde{k}$ for example.]**

$$\frac{\varepsilon}{\int k(\tau)\mathrm{d}\tau} < 0.01.$$

Alternatively, these figures could be understood as providing the minimum c that we should use for given ℓ and m. Of course, we may also read it as providing the minimum ℓ that can be closely approximated given m and c. We obtain the following main conclusions:

- As ℓ increases, c and m required for a close enough approximation decrease.
- The lower c, the smaller m can and ℓ must be to achieve a close approximation.
- For a given ℓ there exist a minimum c under which a close approximation is never going to be achieved regardless of m. This fact can be seen in Figures 6 and 7 as the contour lines which represent c have an end in function of ℓ (Valid c are restricted in function of ℓ).

As stated above, Figures 6 and 7 provide the minimum lengthscale that can be closely approximated given m and c. This information serves as a powerful diagnostic tool in determining if the obtained accuracy is

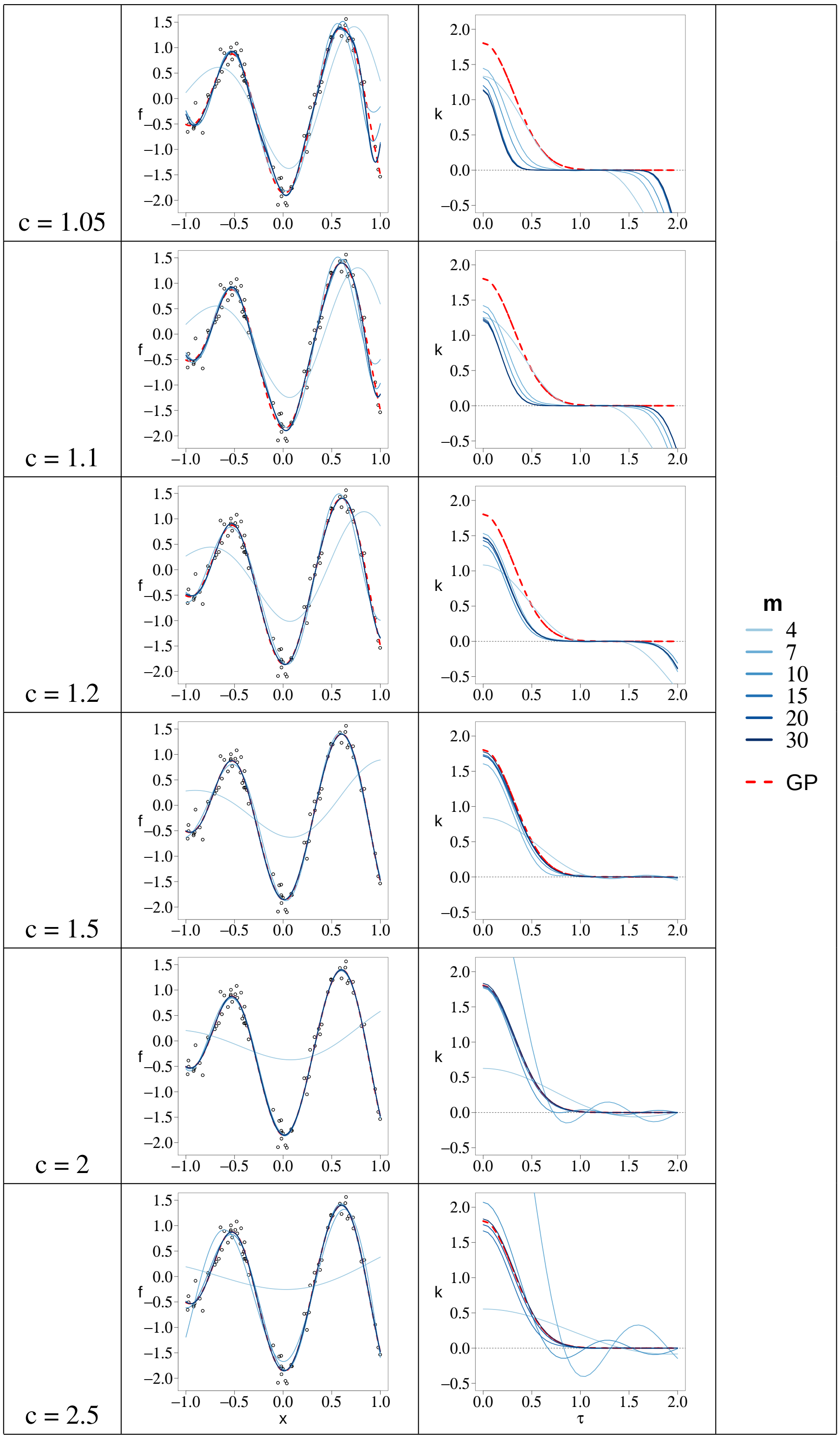

Figure 3: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model and the HSGP model for different number of basis functions m and for different values of the boundary factor c. Notice that the dashed red line of the regular GP can hardly be seen in the some of the plots because it is under the blue lines.

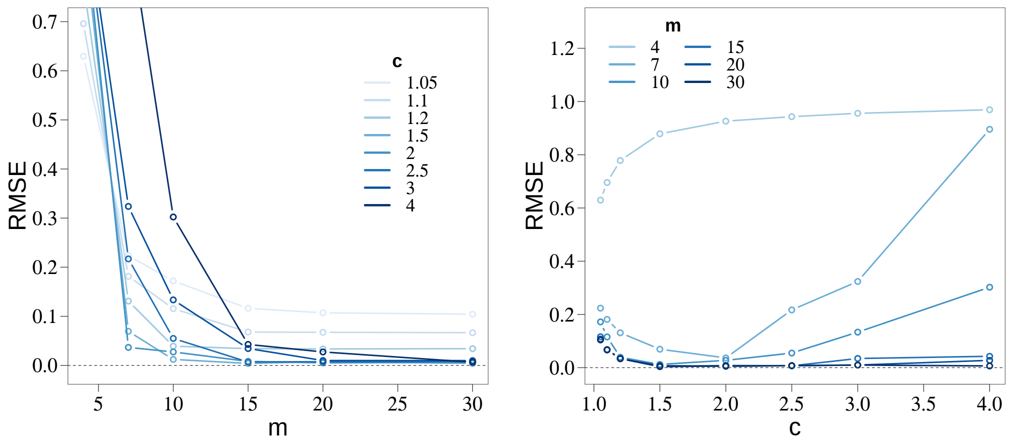

Figure 4: Root mean square error (RMSE) of the proposed HSGP models computed against the regular GP model. RMSE versus the number of basis functions m and for different values of the boundary factor c (left). RMSE versus the boundary factor c and for different values of the number of basis functions m (right).

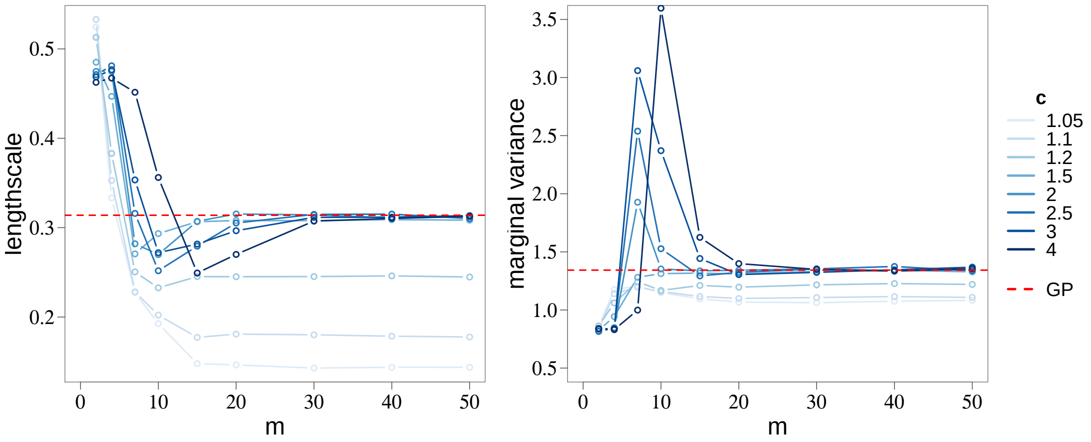

Figure 5: Estimated lengthscale (left) and marginal variance (right) parameters of both regular GP and HSGP models, plotted versus the number of basis functions m and for different values of the boundary factor c.

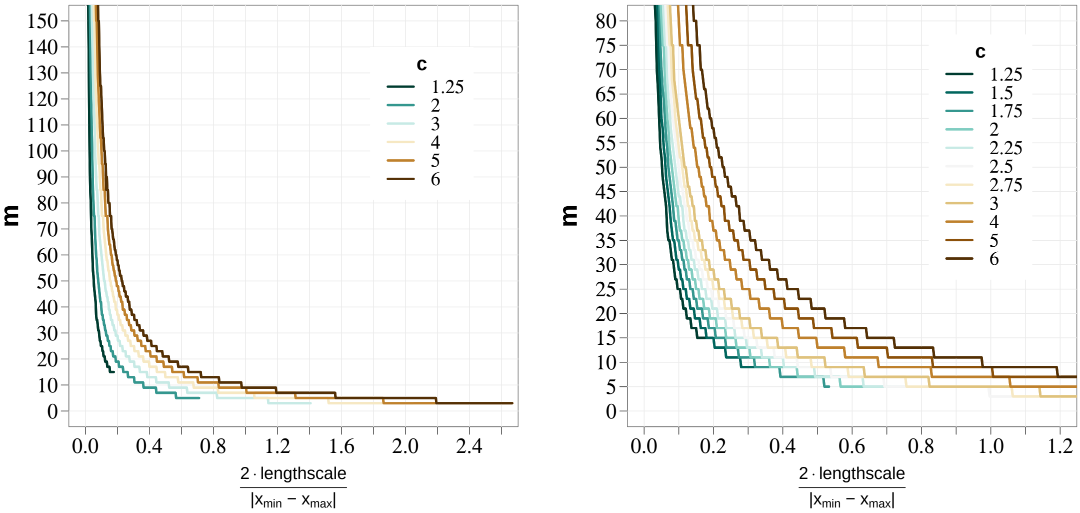

Figure 6: Relation among the minimum number of basis functions m, the boundary factor c $(c = \frac{L}{S})$ and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a square exponential covariance function. The right-side plot is a zoom in of the left-side plot.

acceptable. As the lenghscale ℓ controls the "wigglyness" of the functional relationship, it strongly influences the difficulty of obtaining accurate inference about the function from the data. Basically, if the lengthscale estimate is accurate, we can expect the HSGP approximation to be accurate as well. Thus, having obtained an estimate $\hat{\ell}$ of ℓ from the HSGP model based on prespecified m and c, we can check whether or not $\hat{\ell}$ exceeds the minimum lengthscale provided in Figure 6 or 7 (depending on which kernel is used). If $\hat{\ell}$ exceeds this recommended minimum lengthscale, the approximation should be close enough. If, however, $\hat{\ell}$ does not exceed recommended minimum lengthscale, the approximation may be inaccurate and m should be increased or c decreased. We may also use this diagnostic in a iterative procedure. Starting from some initial guess of ℓ, we can choose initial values for m and c and fit an HSGP model, then check the approximation accuracy, and, if not accurate enough because the estimated $\hat{\ell}$ is below the minimum lengthscale, repeat the process while increasing m or decreasing c. As mentioned earlier, c cannot be decreased as much as desired because it is restricted by the lengthscale, so increasing m may usually the preferred approach.

If we look back to the conclusions drawn from Figures 4 and 5, where $m = 10$ basis functions and a boundary factor of $c = 1.5$ were enough to closely approximate a function with $\ell = 0.3$, we can recognize that these conclusions also matches those obtained from Figure 6.

Figures 6 and 7 were build for a GP with a unidimensional covariance function, which result in a surface depending on three variables, m, c and ℓ. An equivalent figure for a GP model with a two-dimensional covariance function would result in a surface depending on four variables, m, c, ℓ_1 and ℓ_2, which is more difficult to be graphically represented. More precisely, in the multi-dimensional case, whether the approximation is close enough might depend only on the ratio between wiggliness in every dimensions. For instance, in the two-dimensional case, it would depend on the ratio between ℓ_1 and ℓ_2 and could be graphically represented. Future research will focus on building useful graphs or analytical models that provide these relations in multi-dimensional cases. However, as an approximation, we can use the unidimensional GP conclusions in Figures 6 and 7 to check the accuracy by analyze individually the different dimensions of a multidimensional GP model.

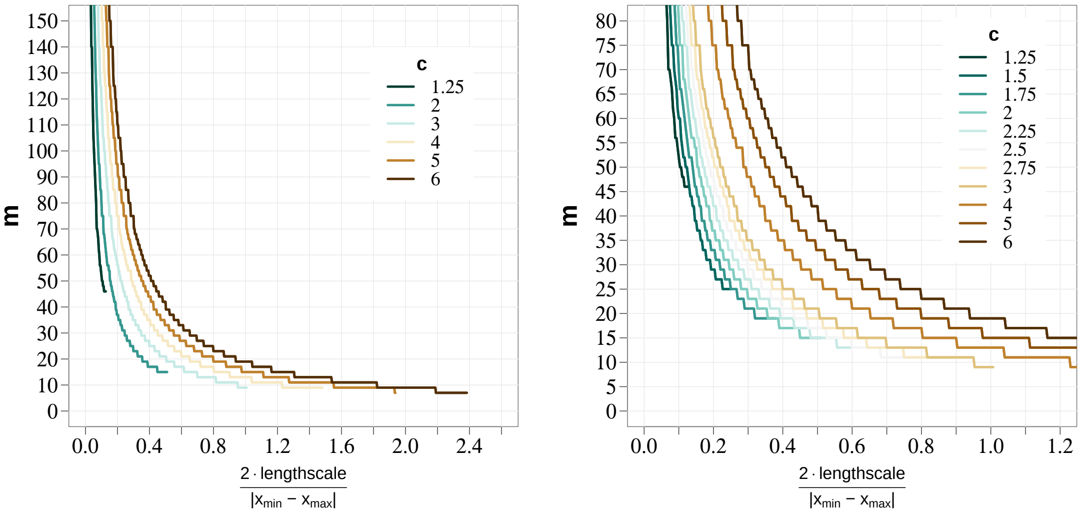

Figure 7: Relation among the minimum number of basis functions m, the boundary factor c $(c = \frac{L}{S})$ and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a Matèrn(ν=3/2) covariance function. The right-side plot is a zoom in of the left-side plot.

4.2 Comparing lengthscale estimates

In this example, we make a comparison of the lengthscale estimates obtained from the regular GP and HSGP models. We also have a look at those recommended minimum lengthscales provided by Figure 6. For this analysis, we will use various datasets consisting of noisy draws from a GP prior model with a squared exponential covariance function and varying lengthscale values. Different values of the number of basis functions m are used when estimating the HSGP models, and the boundary factor c is set to a valid and optimum value in every case.

Figure 8 shows the posterior predictions of both regular GP and HSGP models fitted to those datasets. The lengthscale estimates as obtained by regular GP and HSGP models are depicted in Figure 9. As noted previously, an accurate estimate of the lengthscale can be a good indicator of a close approximation of the HSGP model to the regular GP model. Further, Figure 10 shows the root mean square error (RMSE) of the HSGP models, computed against the regular GP models, as a function of the lengthscale and number of basis functions.

Comparing the accuracy of the lengthscale in Figure 9 to the RMSE in Figure 10, we see that they agree closely with each other for medium lengthscales. That is, a good estimation of the lengthscale implies a small RMSE. This is no longer true for very small or large lengthscales. In small lengthscales, even very small inaccuracies may have a strong influence on the posteriors predictions and thus on the RMSE. In large lengthscales, larger inaccuracies change the posterior predictions only little and may thus not yield large RMSEs. The dashed black line in Figure 9 represents the minimum lengthscale that can be closely approximated under the given condition, according to the results presented in Figure 6. We observe that whenever the estimated lengthscale exceeds the minimially estimable lengthscale, the RMSE of the posterior predictions is small (see Figure 10). Conversely, when the estimated lengthscale is smaller than the minimally estimable one, the RMSE becomes very large.

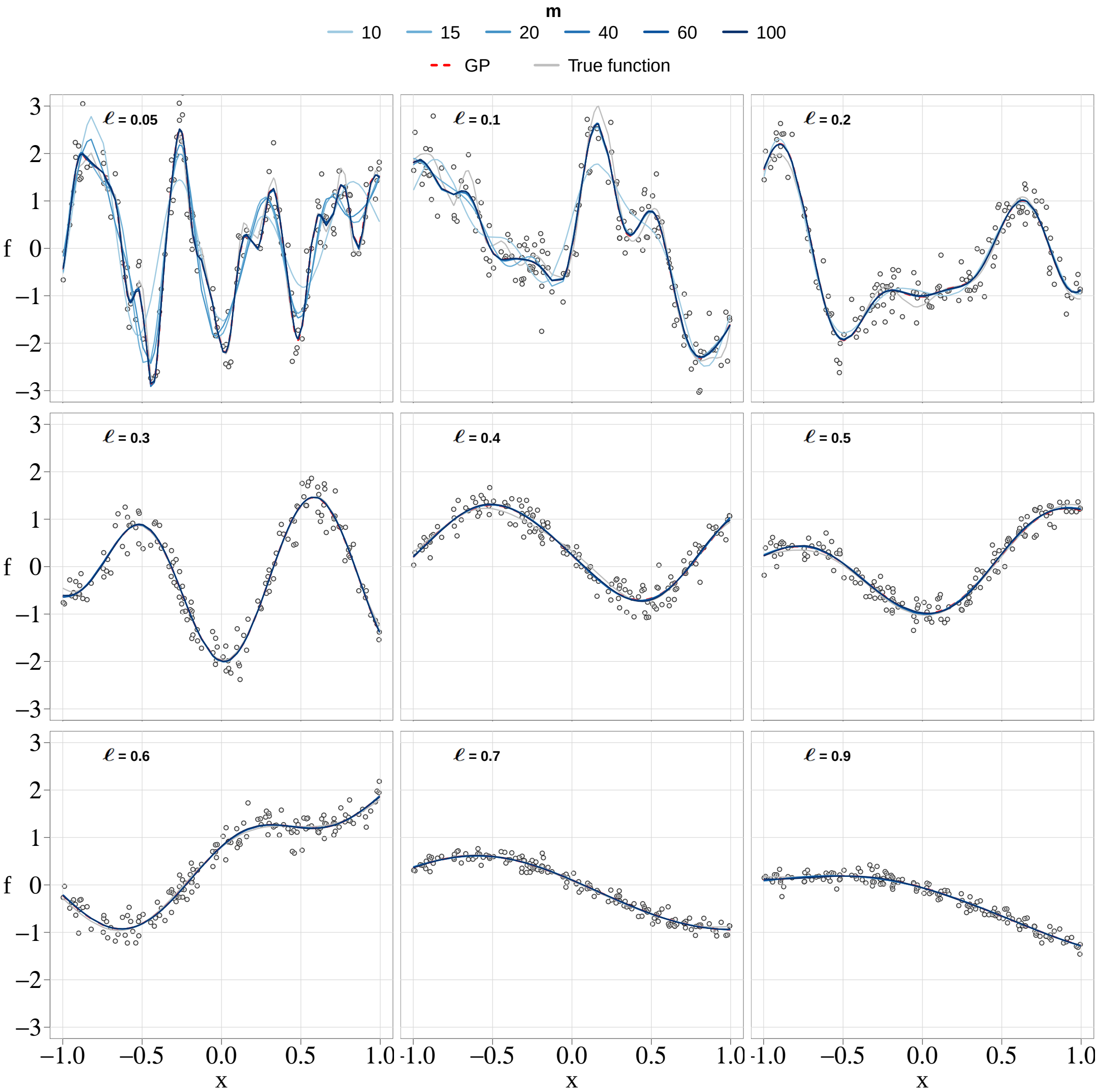

Figure 8: Mean posterior predictions of both regular GP and HSGP models, fitted over various datasets drawn from square exponential GP models with different characteristic lengthscales (ℓ) and same marginal variance (α) as the data-generating functions (*true function*). Notice that the dashed red line of the regular GP can hardly be seen in the plots because it is almost perfectly overlayed by the blue lines.

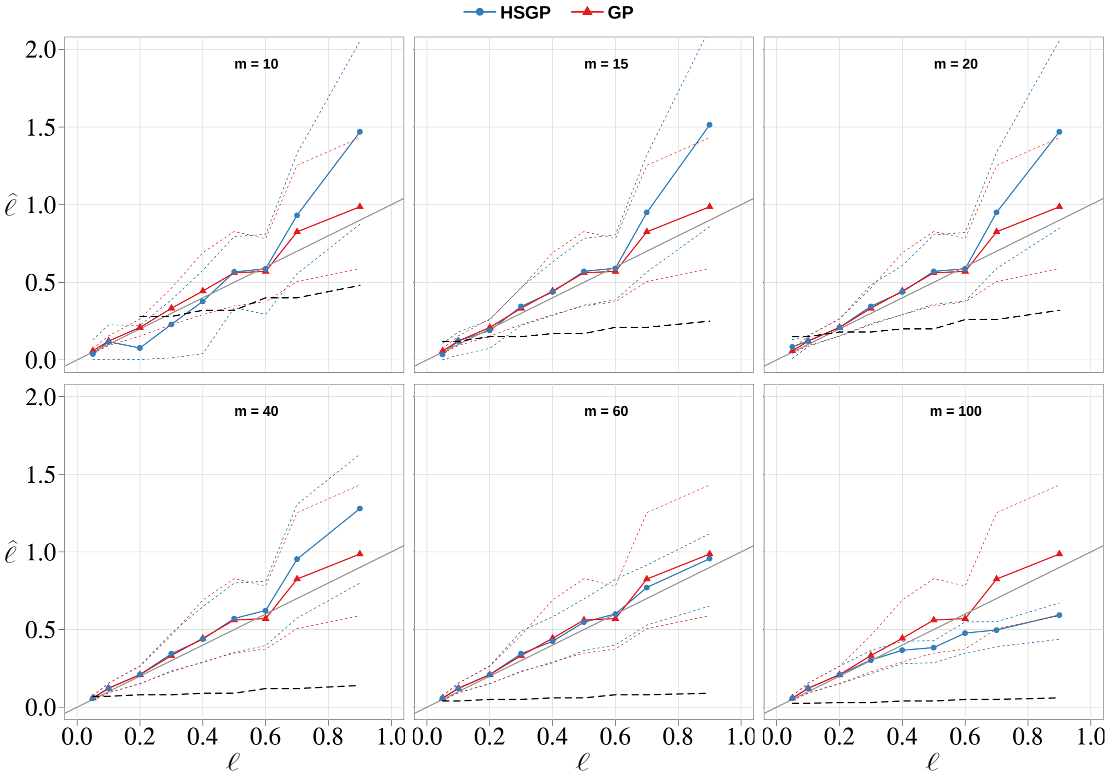

Figure 9: Data-generating functional lengthscales (ℓ), of the various datasets illustrated in Figure 8, versus the corresponding lengthscale estimates ($\hat{\ell}$) from the regular GP and HSGP models. 95% confident intervals of the lengthscale estimates are plotted as dot lines. The different plots represent the use of different number of basis functions m in the HSGP model. The dashed black line represents the recommended minimum lengthscales provided by Figure 6 that can be closely approximated by the HSGP model in every case.

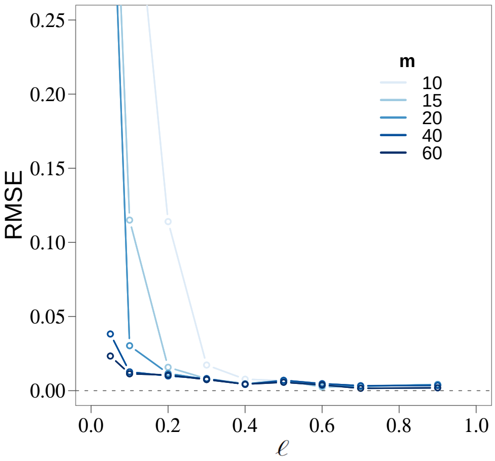

Figure 10: RMSE of the HSGP models with different number of basis functions m, for the various datasets with different wiggly effects (ℓ).

5 Case studies

In this section, we will present several simulated and real case studies in which we apply the developed HSGP models. More case studies are presented in the online supplemental materials.

5.1 Simulated data for a univariate function

This example consists of a simulated dataset with $n = 250$ ($i = 1, \dots, n$) single draws from a Gaussian process prior with a Matérn(ν=3/2) covariance function and hyperparameters marginal variance $\alpha = 1$ and lengthscale $\ell = 0.15$, with corresponding inputs values $\boldsymbol{x} = (x_1 \dots, x_n)$ with $x_i \in [-1, 1] \subset \mathrm{I\!R}$. To form the final noisy dataset $\boldsymbol{y}$, Gaussian noise standard deviation $\sigma = 0.2$ was added to the GP draws.

The regular GP model for fitting this simulated dataset $\boldsymbol{y}$ can be written as follows,

$$\begin{aligned}\boldsymbol{y} &= \boldsymbol{f} + \boldsymbol{\epsilon}\\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 I)\\ f(x) &\sim \mathcal{GP}(0, k(x, x', \theta)),\end{aligned}$$

where I represents the identity matrix and $\boldsymbol{f} = \{f(x_i)\}_{i=1}^n$ represents the underlying function values to the noisy data. The previous formulation corresponds to the latent form of a GP model. The function $f : \mathrm{I\!R} \to \mathrm{I\!R}$ is a GP prior with a Matérn(ν=3/2) covariance function k. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ is multivariate Gaussian distributed with covariance matrix K, where $K_{ij} = k(x_i, x_j, \theta)$, with $i, j = 1, \dots, n$.

A more computationally efficient formulation of a GP model with Gaussian likelihood, and for probabilistic inference using sampling methods such as HMC, would be its marginalized form,

$$\boldsymbol{y} \sim \mathcal{N}(0, K + \sigma^2 I),$$

where the function values $\boldsymbol{f}$ have been integrated out, yielding a lower-dimensional parameter space over which to do inference, reducing the time of computation and improving the sampling and the effective number of samples.

In the HSGP model, the latent function values $f(x)$ are approximated as in equation (7), with the Matérn(ν=3/2) spectral density S as in equation (2), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (5) and (6), respectively. In order to do model comparison, in addition to the regular GP model and HSGP model, a spline-based model is also fitted using the thin plate regression spline approach in Wood [2003] and implemented in the R-package *mgcv* [Wood & Wood, 2015]. A Bayesian approach is used to fit this spline model using the R-package *brms* [Bürkner *et al.* , 2017].

Figure 11 shows the posteriors predictive distributions of the three models, the regular GP, the HSGP with $m = 80$ basis functions and boundary factor $c = 1.2$ ($L = c \cdot 1 = 1.2$; see equation (14)), and the spline model with 80 knots. The true data-generative function and the noisy observations are also plotted. The sample observations are plotted as circles and the out-of-sample or test data, which have not been taking part on training the models, are plotted as crosses. The test data located at the extremes of the plot are used for assessing model extrapolation, and the test data located in the middle are used for assessing model interpolation. The posteriors of the three models, regular GP, HSGP and spline, are pretty similar within the interpolation input space. However, when extrapolating the spline model solution clearly differs from the regular GP and HSGP models as well as the actual observations.

In order to assess the performance of the models as a function of the number of basis functions and number of knots, different models with different number of basis functions for the HSGP model, and different

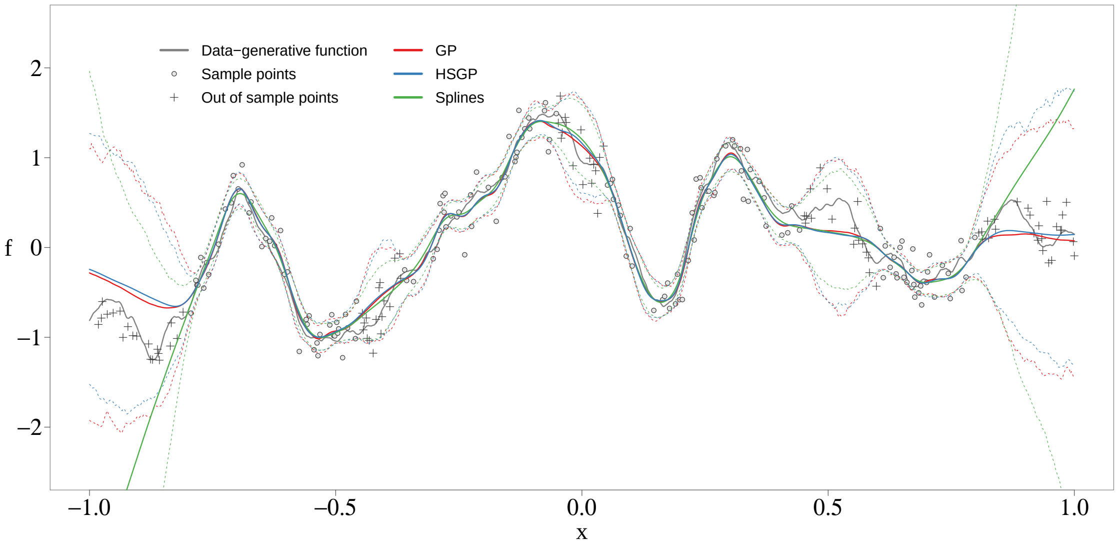

Figure 11: Posterior predictive means of the proposed HSGP model, the regular GP model, and the spline model. 95% credible intervals are plotted as dashed lines.

number of knots for the spline model, have been fitted. Figure 12 shows the standardized root mean squared error (SRMSE) for interpolation and extrapolating data as a function of the number of basis functions and knots. The SRMSE is computed against the data-generating model. From Figures 11 and 12, it can be seen a close approximation of the HSGP model to the regular GP model for interpolating and extrapolating data. However, the spline model does not extrapolate data properly. Both models show roughly similar interpolating performance.

Figure 13 shows computational times, in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the spline model. The HSGP model is on average roughly 400 times faster than the regular GP and 10 times faster than the spline model, for this particular application with a univariate input space. Also, the increase in computation time as a function of the number of basis functions in a univariate input space is relatively slight.

The Stan model code for the exact GP, the approximate GP and the spline models of this case study can be found online at `https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_1D-Simulated-data`.

In the online supplemental material, some other case studies are presented. From those examples, it can be seen how computation time of the HSGP model increases rapidly with the number of input dimensions (D) since the number of basis functions in the approximation increases exponentially with D (see equation (8)). Even though, in a bivariate input space, the computation time increases significantly with D, the HSGP model works significantly faster than the regular GP for most of the non-linear $2D$ functions (even highly wiggly functions; see Figures B.3-right and C.3 in the online material). However, HSGPs tend to be slower than exact GPs for $D > 3$ with a relatively low number of basis functions ($m \gtrsim 5$), as well as even for $D = 3$ with a moderate high number of basis functions ($m \gtrsim 20$; see Figure C.3 in the online material). In all of the investigated cases, choosing the optimal boundary factor in the HSGP approximation reduces the number of required basis functions noticably (see Figures A.3, B.3-left and C.2 in the online material) and therefore also reduces computational time drastically in particular in multivariate input spaces.

Roughly similar or even worse behavior was found for splines where serious difficulties with computation time were encountered in building spline models for $D = 3$ and *knots* > 10, or even for $D = 2$ and *knots*

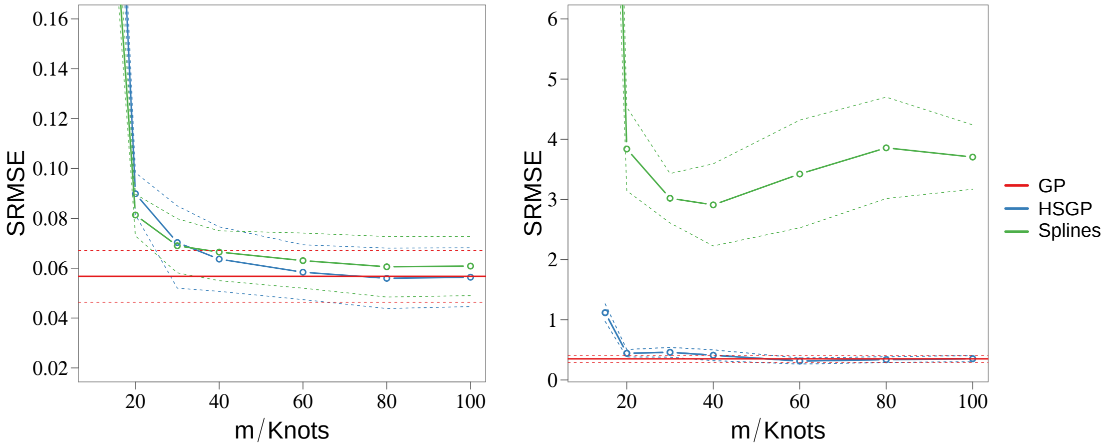

Figure 12: Standardized root mean square error (SRMSE) of the different methods against the data-generating function. SRMSE for interpolation (left) and SRMSE for extrapolation (right). The standard deviation of the mean of the SRMSE is plotted as dashed lines.

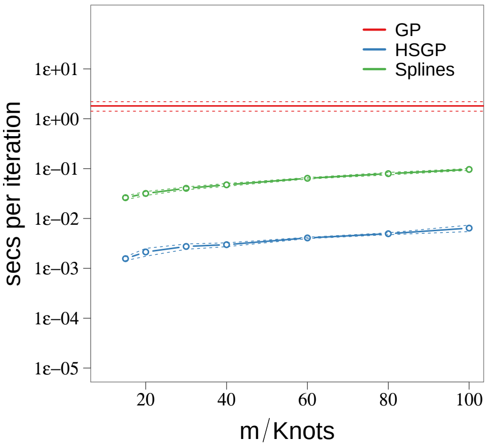

Figure 13: Computational time (y-axis), in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the spline model. The y-axis is on a logarithmic scale. The standard deviation of the computational time is plotted as dashed lines.

> 40 (see Figures B.3-right and C.3 in the online material).

5.2 Birthday data

This example is an analysis of patterns in birthday frequencies in a dataset containing records of all births in the United States on each day during the period 1969–1988. The model decomposes the number of births along all the period in longer-term trend effects, patterns during the year, day-of-week effects, and special days effects. The special days effects cover patterns such as possible fewer births on Halloween, Christmas or new year, and excess of births on Valentine's Day or the days after Christmas (due, presumably, to choices involved in scheduled deliveries, along with decisions of whether to induce a birth for health reasons). This analysis was originally addressed in Gelman *et al.* [2013]. The total number of days within the period is $T = 7305$ ($t = 1, \dots, T$), then a regular GP model is unfeasible to be fitted on this dataset as we know inference scales $O(T^3)$ in covariance matrix inversion. Therefore, an approximate approach has to be used to fit a GP model on this data. We will use the HSGP method as well as the low-rank GP model with a periodic covariance function introduced in Appendix B which is based on expanding the periodic covariance function into a series of stochastic resonators [Solin & Särkkä, 2014].

Let's denote y_t as the number of births of day t. The observational model is a normal model with parameters the mean function $\mu(t)$ and noise variance σ^2,

$$y_t \sim \mathcal{N}(\mu(t), \sigma^2).$$

The mean function $\mu(t)$ will be defined as an additive model in the form:

$$\mu(t) = f_1(t) + f_2(t) + f_3(t) + f_4(t). \tag{15}$$

The component $f_1(t)$ represents the long-term trends modeled by a GP with squared exponential covariance function,

$$f_1(t) \sim \mathcal{GP}(0, k_1), \qquad k_1(t, t') = \sigma_1^2 \exp\Big(-\frac{1}{2}\frac{(t-t')^2}{\ell_1^2}\Big),$$

which means the function values $\boldsymbol{f}_1 = \{f_1(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_1, where $K_{1_{t,s}} = k_1(t, s)$, with $t, s = 1, \dots, T$.

The component $f_2(t)$ represents the yearly smooth seasonal pattern, using a periodic squared exponential covariance function (with period 365.25 to match the average length of the year) in a GP model,

$$f_2(t) \sim \mathcal{GP}(0, k_2), \qquad k_2(t, t') = \sigma_2^2 \exp\Big(-\frac{2\sin^2(\pi(t-t')/365.25}{\ell_2^2}\Big),$$

which means the function values $\boldsymbol{f}_2 = \{f_2(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_2, where $K_{2_{t,s}} = k_2(t, s)$, with $t, s = 1, \dots, T$.

The component $f_3(t)$ represents the weekly smooth pattern using a periodic squared exponential covariance function (with period 7 of length of the week) in a GP model,

$$f_3(t) \sim \mathcal{GP}(0, k_3), \qquad k_3(t, t') = \sigma_3^2 \exp\Big(-\frac{2\sin^2(\pi(t-t')/7}{\ell_3^2}\Big),$$

which means the function values $\boldsymbol{f}_3 = \{f_3(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_3, where $K_{3_{t,s}} = k_3(t, s)$, with $t, s = 1, \dots, T$.

The component $f_4(t)$ represents the special days effects, modeled as a horse-shoe prior model [Piironen *et al.* , 2017]:

$$f_4(t) \sim \mathcal{N}(0, \lambda_t^2 \tau^2), \qquad \lambda_t^2 \sim \mathcal{C}^+(0,1).$$

A horse-shoe prior allows for sparse distributed effects. Its global parameter τ pulls all the weights (effects) globally towards zero, while the thick half-Cauchy tails for the local scales λ_t allow some of the weights to escape the shrinkage. Different levels of sparsity can be accommodated by changing the value of τ: with large τ all the variables have very diffuse priors with very little shrinkage towards zero, but letting $\tau \to 0$ will shrink all the weights $f_4(t)$ to zero [Piironen & Vehtari, 2016].

GP priors have been defined over the components $f_1(t)$, $f_2(t)$ and $f_3(t)$. Then, low-rank representations of the GP priors have to be used in the modeling and inference. The component $f_1(t)$ will be approximated using the HSGP model. Thus, the function values $f_1(t)$ are approximated as in equation (7), with the squared exponential spectral density S as in equation (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (5) and (6).

The year effects $f_2(t)$ and week effects $f_3(t)$, as they use a periodic covariance function, they do no fit under the main framework of the HSGP approximation covered in this chapter. However, they do have a representation based on expanding periodic covariance functions into a series of stochastic resonators (Appendix B). Thus, the functions $f_2(t)$ and $f_3(t)$ are approximated as in equation (B.7), with variance coefficients $\tilde{q}_j^2$ as in equation (B.5).

For the component $f_1(t)$, $m = 30$ basis functions and a boundary factor $c = 1.5$ were used. The lengthscale estimate $\hat{\ell}_1$, for this component, normalized by half of the range of the input x_1, is bigger than the minimum lengthscale reported by Figure 6 as a function of m and c. This means that the chosen number of basis functions and the boundary factor are suitable values for modeling the input effects sufficiently accurate.

For the components $f_2(t)$ and $f_3(t)$, $J = 10$ cosine terms were used. The lengthscales estimates $\hat{\ell}_2$ and $\hat{\ell}_3$, for the GP components $f_2(t)$ and $f_3(t)$, respectively, are bigger than the minimum lengthscale reported by Figure B.1 as function of the number of cosine terms J, which means that the approximations are accurate enough.

Figure 14 shows the posterior means of the long-term trend $f_1(t)$ and year patterns $f_2(t)$ for the whole period, jointly with the observed data. Figure 15 show the process for one year (1972) only. In this figure, the special days effects $f_4(t)$ in the year can be clearly represented. The posterior means of the the function $\mu(t)$ and the components $f_1(t)$ (long-term trend) and $f_2(t)$ (year pattern) are also plotted in this Figure 15. Figure 16 show the process in the month of January of 1972 only, where the week pattern $f_3(t)$ can be clearly represented. The mean of the the function $\mu(t)$ and components $f_1(t)$ (long-term trend), $f_2(t)$ (year pattern) and $f_4(t)$ (special-days effects) are also plotted in this Figure 16.

The Stan model code for the approximate GP model of this case study can be found at `https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_Birthday-data`.

5.3 Leukemia data

The next example presents a survival analysis in acute myeloid leukemia (AML) in adults, with data recorded between 1982 and 1998 in the North West Leukemia Register in the United Kingdom. The data set consists of survival times t_i and censoring indicator z_i (0 for observed and 1 for censored) for $n = 1043$ cases ($i = 1, \ldots, n$). About 16% of cases were censored. Predictors are *age* (x_1), *sex* (x_2), *white blood cell* (WBC) (x_3) count at diagnosis with 1 unit = 50 × $109/L$, and the *Townsend deprivation index* (TDI) (x_4) which is a measure of deprivation for district of residence. We denote $\boldsymbol{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}) \in \mathbb{R}^4$ as the vector of predictor values for observation i.

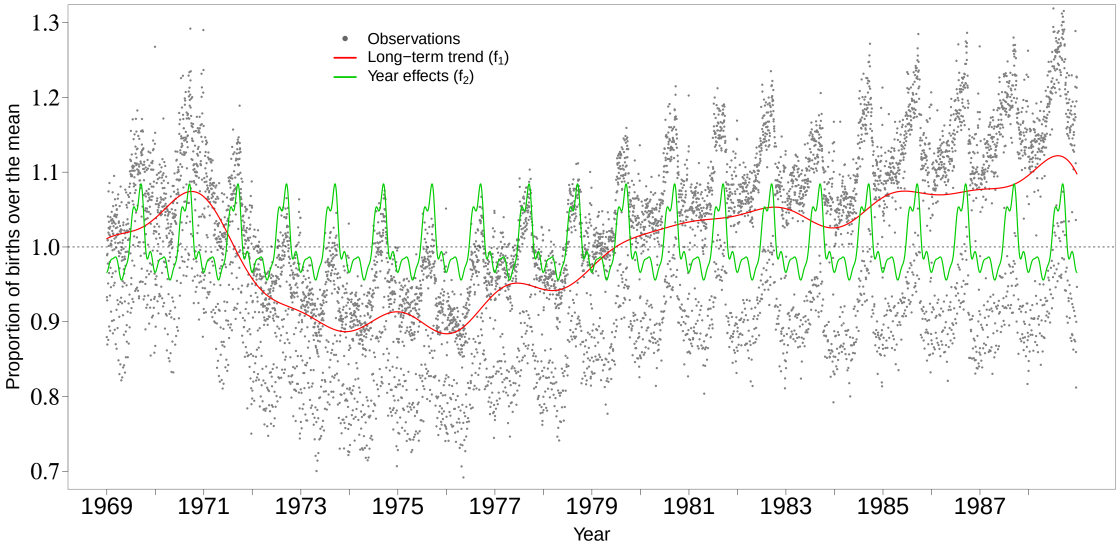

Figure 14: Posterior means of the long-term trend ($f_1(\cdot)$) and year effects pattern ($f_2(\cdot)$) for the whole series.

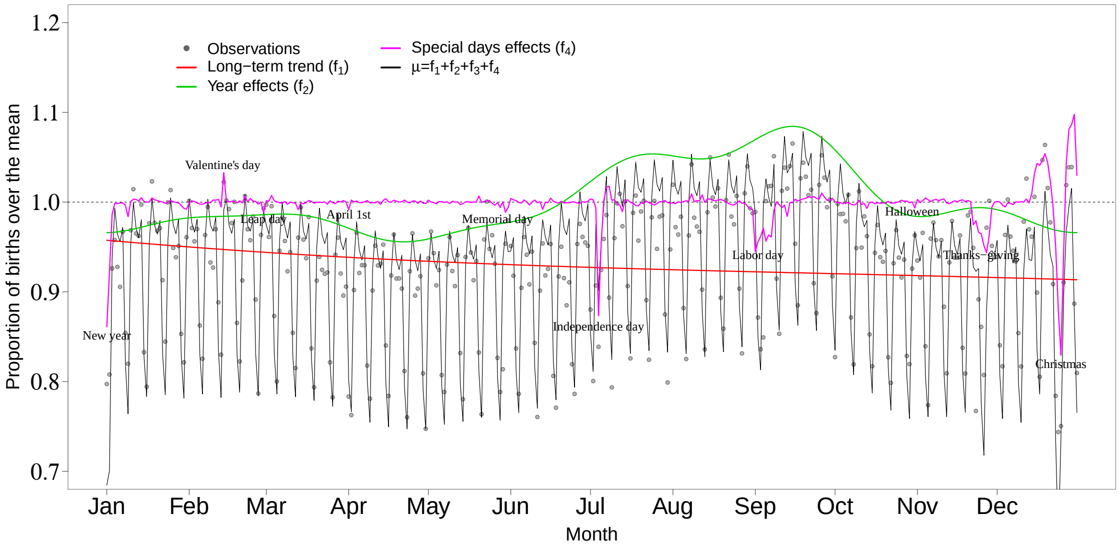

Figure 15: Posterior means of the function $\mu(\cdot)$ for the year 1972 of the series. The special days effects pattern ($f_4(\cdot)$) in the year is also represented, as well as the long-term trend ($f_1(\cdot)$) and year effects pattern ($f_2(\cdot)$).

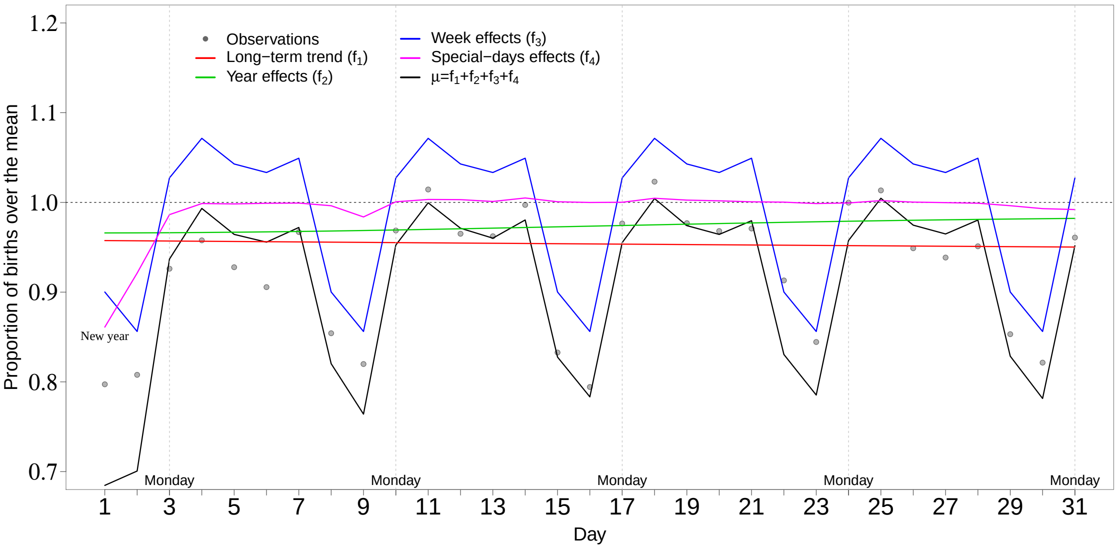

Figure 16: Posterior means of the function $\mu(\cdot)$ for the month of January of 1972. The week effects pattern ($f_3(\cdot)$) in the month is also represented, as well as the long-term trend ($f_1(\cdot)$), year effects pattern ($f_2(\cdot)$) and special days effects pattern ($f_4(\cdot)$).

As the WBC measurements were strictly positive and highly skewed, we fit the model to its logarithm. Continuous predictors were normalized to have zero mean and unit standard deviation. We assume a log-normal observation model for the observed survival time, t_i, with a function of the predictors, $f(\boldsymbol{x}_i) : \mathbb{R}^4 \to \mathbb{R}$, as the location parameter, and σ as the Gaussian noise:

$$p(t_i) = \text{LogNormal}(t_i | f(\boldsymbol{x}_i), \sigma^2).$$

As we do not have a model for the censoring process, we do not have a full observation model, and the observational model for the censored data t_i is assumed to be the complementary cumulative normal probability distribution:

$$p(y_i > t_i) = \int_{t_i}^{\infty} \text{LogNormal}(y_i | f(\boldsymbol{x}_i), \sigma^2) \mathrm{d}y_i = 1 - \Phi\Big(\frac{\log(y_i) - f(\boldsymbol{x}_i)}{\sigma}\Big),$$

where y_i denotes the uncensored time.

The latent function $f(\cdot)$ is modeled as a Gaussian process, centered around a linear model of the predictors $\boldsymbol{x}$, and with a squared exponential covariance function k depending on predictors x and hyperparameters $\theta = (\alpha, \ell)$,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(c + \boldsymbol{\beta}\boldsymbol{x}, k(\boldsymbol{x}, \boldsymbol{x}', \theta)),$$

where c and $\boldsymbol{\beta}$ are the intercept and vector of coefficients, respectively, of the linear model. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ are multivariate Gaussian distributed with mean function $\mu(\cdot)$ and covariance matrix K, where $\mu(\boldsymbol{x}_i) = c + \boldsymbol{\beta}\boldsymbol{x}_i$ and $K_{rs} = k(\boldsymbol{x}_r, \boldsymbol{x}_s, \theta)$, with $r, s = 1, \ldots, n$. The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process. Notice that a scalar lengthscale is considered in the multivariate covariance function assuming an isotropic function.

Due to the predictor *sex* (x_2) being a categorical variable ('1' for female and '2' for male), we apply dummy/treatment coding for the GP functions, in a similar way such coding is applied in linear models. A

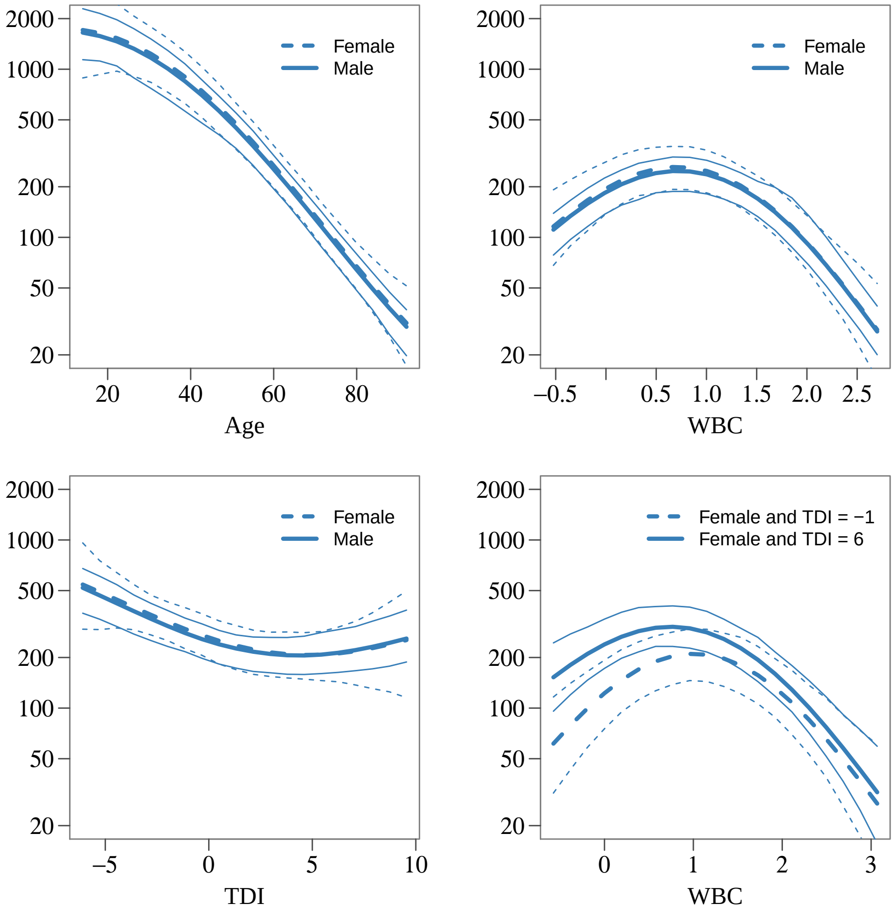

Figure 17: Expected lifetime conditional comparison for each predictor with other predictors fixed to their mean values. The thick line in each graph is the posterior mean estimated using a HSGP model, and the thin lines represent pointwise 95% credible intervals.

general mean GP function is defined for all observations, while a second GP function only applies to one of the predictor levels ('male' in this case) and is set to zero otherwise. More formally, this construction reads as follows:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(c + \boldsymbol{\beta x}, k(\boldsymbol{x}, \boldsymbol{x}', \theta_1)) + g(\boldsymbol{x}), \tag{16}$$

with

$$g(\boldsymbol{x}) = \begin{cases} 0 & \text{if} \quad x_2 = \text{'1'} \\ \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}', \theta_2)) & \text{if} \quad x_2 = \text{'2'} \end{cases} \tag{17}$$

In the former equations, θ_1 contains the hyperparameters α_1 and ℓ_1 which are the marginal variance and lengthscale, respectively, of the general mean GP function, and θ_2 contains the hyperparameters α_2 and ℓ_2 which are the marginal variance and lengthscale, respectively, of a specific GP function restricted to the male sex ($x_2 = $ '2').

Using the HSGP approximation, the functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are approximated as in equation (12), with the D-dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (1), and the multivariate eigenfunctions ϕ_j and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ as in equations (10) and (9), respectively.

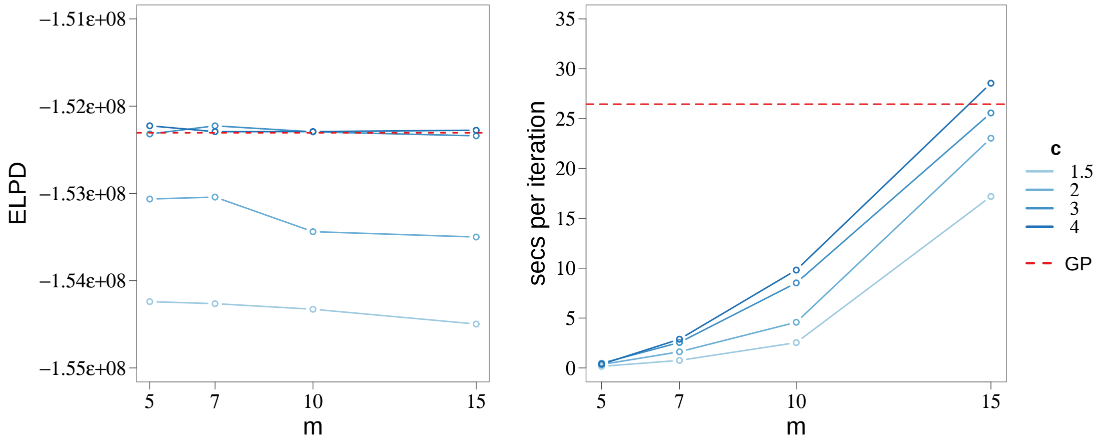

Figure 18: Expected log predictive density (ELPD; left) and time of computation in seconds per iteration (iteration of the HMC sampling method; right) as a function of the number of basis functions m and boundary factor c.

Figure 17 shows estimated conditional comparison of each predictor with all others fixed to their mean values. These posterior estimates correspond to the HSGP model with $m = 10$ basis functions and $c = 3$ boundary factor. The model has found smooth non-linear patterns and the right bottom subplot also shows that the conditional comparison associated with WBC has an interaction with TDI.

Figure 18 shows the expected log predictive density (ELPD; see Vehtari *et al.* [2012]) and time of computation as function of the number of univariate basis functions m ($m^* = m^D$ in equation (12)) and boundary factor c. As the functions are smooth, a few number of basis functions and a large boundary factor are required to obtain a good approximation (Figure 18-left); Small boundary factors are not allowed when large lengthscales, as can be seen in Figure 6. Increasing the boundary factor also significantly increases the time of computation (Figure 18-right). With a moderate number of univariate basis functions ($m = 15$), the HSGP model becomes slower than the exact GP model, in this specific application with 3 input variables, as the total number of multivariate basis functions becomes $15^3 = 3375$ and is therefore quite high.

The Stan model code for the exact GP and the approximate GP models of this case study can be found at `https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_Leukemia-data`.

6 Conclusion

Modeling unknown functions using exact GPs is computationally intractable for a lot of applications. This problem becomes especially severe when performing full Bayesian inference using sampling-based methods. In this paper, a novel approach for a low-rank representation of stationary GPs, originally proposed by Solin & Särkkä [2018], has been implemented and analyzed in detail. The method is based on a basis function approximation via Laplace eigenfunctions. The method has an attractive computational cost as it effectively approximates GPs by linear models, which is also an attractive property in modular probabilistic programming programming frameworks. The dominating cost per log density evaluation (during sampling) is $O(nm + m)$, which is a big benefit in comparison to $O(n^3)$ of a regular GP model. The obtained design matrix is independent of hyperparameters and therefore only needs to be constructed once, at cost $O(nm)$. All dependencies on the kernel and the hyperparameters are through the prior distribution of the regression weights. The parameters' posterior distribution is m-dimensional, where m is usually much smaller than the number of observations n.

The main contribution of this paper is an in-depth analysis and diagnosis of the performance and accuracy of the approximation in relation to the key factors of the method, that is, the number of basis functions, the boundary condition of the Laplace eigenfunctions, and the non-linearity of the function to be learned. Recommendations for the values of these key factors based on the recognized relations among them have been provided along with illustrations of these relations. These illustrations will not only help users to improve performance and save computation time, but also serve as a powerful diagnosis tool whether the chosen values for the number of basis functions and the boundary condition are adequate to fit to the data at hand with sufficient accuracy.

The developed approximate GPs can be easily applied as modular components in probabilistic programming frameworks such as Stan in both Gaussian and non-Gaussian observational models. Using several simulated and real datasets, we have demonstrated the practical applicability and improved sampling effiency, as compared to regular GPs, of the developed method. The main drawback of the approach is that its computational complexity scales exponentially with the number of input dimensions. Hence, choosing optimal values for the number of basis functions and the boundary factor, using the recommendations and diagnostics provided in Figures 6 and 7, is essential to avoid a excessive computational time especially in multivariate input spaces. However, in practice, input dimensionalities larger than 3 start to be quite computationally demanding even for moderately wiggly functions and few basis functions per input dimension. In these high dimensional cases, the proposed approxiamte GP methods may still be used for low-dimensional components in an additive modeling scheme but without modeling very high dimensional interactions.

The obtained functional relationships between the key factors influencing the approximation not only help users to visually assess the accuracy of the method but can also serve an automatic diagnostic tool, if appropriately implemented. In this paper, we primarily studied the functional relationships for univariate inputs. Accordingly, investigating the functional relationships more thoroughly for multivariate inputs remains a topic for future research.

A Approximation of the covariance function using Hilbert space methods

In this section, we briefly present a summary of the mathematical details of the approximation of a stationary covariance function as a series expansion of eigenvalues and eigenfunctions of the Laplacian operator. This statement is basically an extract of the work Solin & Särkkä [2018], where the authors fully develop the mathematical theory behind the Hilbert Space approximation for stationary covariance functions.

Associated to each covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ we can also define a covariance operator $\mathcal{K}$ over a function $f(\boldsymbol{x})$ as follows:

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

From the Bochner's and Wiener-Khintchine theorem, the spectral density of a stationary covariance function $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau})$, $\boldsymbol{\tau} = (\boldsymbol{x} - \boldsymbol{x}')$, is the Fourier transform of the covariance function,

$$S(\boldsymbol{w}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \boldsymbol{w} \boldsymbol{\tau}} \mathrm{d}\boldsymbol{\tau},$$

where $\boldsymbol{w}$ is in the frequency domain. The operator $\mathcal{K}$ will be translation invariant if the covariance function is stationary. This allows for a Fourier representation of the operator $\mathcal{K}$ as a transfer function which is the spectral density of the Gaussian process. Thus, the spectral density $S(\boldsymbol{w})$ also gives the approximate eigenvalues of the operator $\mathcal{K}$.

In the isotropic case $S(\boldsymbol{w}) = S(||\boldsymbol{w}||)$ and assuming that the spectral density function $S(\cdot)$ is regular enough, then it can be represented as a polynomial expansion:

$$S(||\boldsymbol{w}||) = a_0 + a_1||\boldsymbol{w}||^2 + a_2(||\boldsymbol{w}||^2)^2 + a_3(||\boldsymbol{w}||^2)^3 + \cdots . \tag{A.1}$$

The Fourier transform of the Laplace operator ∇^2 is $-||\boldsymbol{w}||$, thus the Fourier transform of $S(||\boldsymbol{w}||)$ is

$$\mathcal{K} = a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + a_3(-\nabla^2)^3 + \cdots , \tag{A.2}$$

defining a pseudo-differential operator as a series of Laplace operators.

If the negative Laplace operator $-\nabla^2$ is defined as the covariance operator of the formal kernel l,

$$-\nabla^2 f(\boldsymbol{x}) = \int l(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}',$$

then the formal kernel can be represented as

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

where $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(\boldsymbol{x})\}_{j=1}^{\infty}$ are the set of eigenvalues and eigenvectors, respectively, of the Laplacian operator. Namely, they satisfy the following eigenvalue problem in the compact subset $\boldsymbol{x} \in \Omega \subset \mathbb{R}^D$ and with the Dirichlet boundary condition (another boundary condition could be used as well):

$$\begin{aligned} -\nabla^2 \phi_j(\boldsymbol{x}) &= \lambda \phi_j(\boldsymbol{x}), && x \in \Omega \\ \phi_j(\boldsymbol{x}) &= 0, && x \notin \Omega. \end{aligned}$$

Because $-\nabla^2$ is a positive definite Hermitian operator, the set of eigenfunctions $\phi_j(\cdot)$ are orthonormal with respect to the inner product

$$< f, g >= \int_\Omega f(\boldsymbol{x})g(\boldsymbol{x})\mathrm{d}(\boldsymbol{x})$$

that is,

$$\int_\Omega \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})\mathrm{d}(\boldsymbol{x}) = \delta_{ij},$$

and all the eigenvalues λ_j are real and positive.

Due to normality of the basis of the representation of the formal kernel $l(\boldsymbol{x}, \boldsymbol{x}')$, its formal powers $s = 1, 2, \ldots$ can be written as

$$l(\boldsymbol{x}, \boldsymbol{x}')^s = \sum_j \lambda_j^s \phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'), \tag{A.3}$$

which are again to be interpreted to mean that

$$(-\nabla^2)^s f(\boldsymbol{x}) = \int l^s(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}'.$$

This implies that we also have

$$[a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \cdots]f(\boldsymbol{x}) = \int [a_0 + a_1 l^1(\boldsymbol{x},\boldsymbol{x}') + a_2 l^2(\boldsymbol{x},\boldsymbol{x}') + \cdots]f(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}'.$$

Then, looking at equations (A.2) and (A.3), it can be concluded

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_j [a_0 + a_1\lambda_j^1 + a_2\lambda_j^2 + \cdots]\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'). \tag{A.4}$$

By letting $||\boldsymbol{w}||^2 = \lambda_j$ the spectral density in Equation (A.1) becomes

$$S(\sqrt{\lambda_j}) = a_0 + a_1\lambda_j + a_2\lambda_j^2 + a_3\lambda_j^3 + \cdots ,$$

and substituting in equation (A.4) then leads to the final searched approximation

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_j S(\sqrt{\lambda_j})\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'), \tag{A.5}$$

where $S(\cdot)$ is the spectral density of the covariance function, λ_j is the jth eigenvalue and $\phi_j(\cdot)$ the eigenfunction of the Laplace operator in a given domain.

B Low-rank Gaussian process with a periodic covariance function

A GP model with a periodic covariance function does no fit in the framework of the HSGP approximation covered in this study. However, it do has a low-rank representation. In this section, we first give a brief presentation of the results from Solin & Särkkä [2014], where the authors obtain an approximate linear representation of a periodic squared exponential covariance function based on expanding the periodic covariance function into a series of stochastic resonators. Secondly, we analyze the accuracy of this approximation and, finally, we derive the GP model with this approximate periodic square exponential covariance function.

The periodic squared exponential covariance function takes the form

$$k(\boldsymbol{\tau}) = \alpha \exp\Big(-\frac{2\sin^2(\omega_0 \frac{\boldsymbol{\tau}}{2})}{\ell^2}\Big), \tag{B.1}$$

where α is the magnitude scale of the covariance, ℓ is the characteristic lengthscale of the covariance, and ω_0 is the angular frequency defining the periodicity.

In Solin & Särkkä [2014], the authors come to a cosine series expansion for the periodic covariance function (B.1) as follows,

$$k(\tau) = \alpha \sum_{j=0}^{J} \tilde{q}_j^2 \cos(j\omega_0\tau), \tag{B.2}$$

which comes basically from a Taylor series representation of the periodic covariance function. The coefficients $\tilde{q}_j^2$ of the previous expression are

$$\tilde{q}_j^2 = \frac{2}{\exp(\frac{1}{\ell^2})} \sum_{j=0}^{\lfloor\frac{J-j}{2}\rfloor} \frac{(2\ell^2)^{-j-2}}{(j+i)!i!}, \tag{B.3}$$

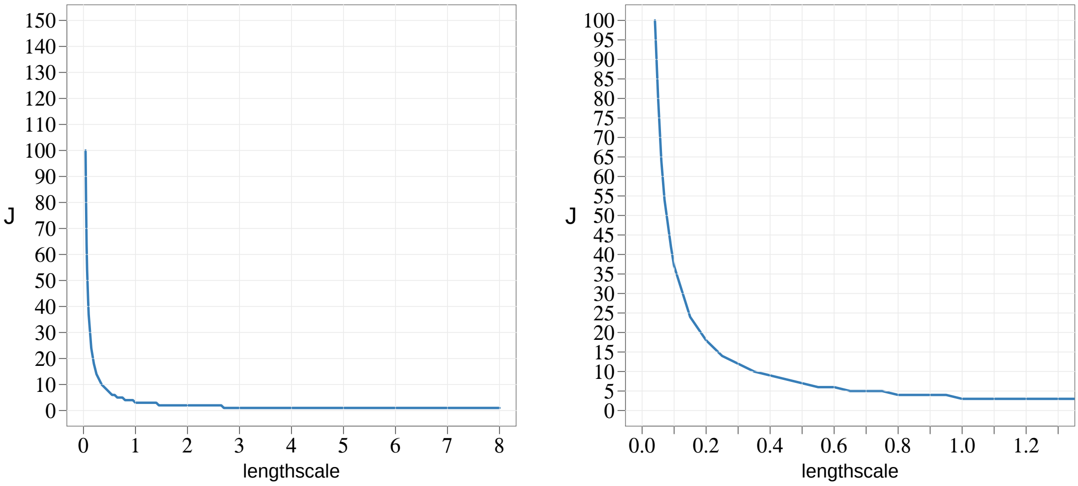

Figure B.1: Relation among the minimum number of terms J in the approximation and the lengthscale (ℓ) of the periodic squared exponential covariance function. The right-side plot is a zoom in of the left-side plot.

where $j = 1, 2, \cdots, J$, and $\lfloor \cdot \rfloor$ denotes the floor round-off operator. For the index $j = 0$, the coefficient is

$$\tilde{q}_0^2 = \frac{1}{2} \frac{2}{\exp(\frac{1}{\ell^2})} \sum_{j=0}^{\lfloor \frac{J-j}{2} \rfloor} \frac{(2\ell^2)^{-j-2}}{(j+i)!i!}. \tag{B.4}$$

Note that the covariance in equation (B.2) is a Jth order truncation of a Taylor series representation. As Solin & Särkkä [2014] argue, this approximation converges to equation (B.1) when $J \to \infty$.

An upper bounded approximation to the coefficients $\tilde{q}_j^2$ and $\tilde{q}_0^2$ can be obtained by taking the limit $J \to \infty$ in the sub-sums in the corresponding equations (B.3) and (B.4), and thus leading to the following variance coefficients:

$$\begin{aligned} \tilde{q}_j^2 &= \frac{2\mathrm{I}_j(\ell^{-2})}{\exp(\frac{1}{\ell^2})}, \\ \tilde{q}_0^2 &= \frac{\mathrm{I}_0(\ell^{-2})}{\exp(\frac{1}{\ell^2})}, \end{aligned} \tag{B.5}$$

for $j = 1, 2, \cdots, J$, and where the $\mathrm{I}_j(z)$ is the modified Bessel function [Abramowitz & Stegun, 1970] of the first kind. This approximation implies that the requirement of a valid covariance function is relaxed and only an optimal series approximation is required [Solin & Särkkä, 2014]. A more detailed explanation and mathematical proofs of this approximation of a periodic covariance function can be found in Solin & Särkkä [2014].

In order to assess the accuracy of this representation as a function of the number of cosine terms J considered in the approximation, an empirical evaluation is carried out in a similar way than that in Section 4 of this work. Thus, Figure B.1 shows the minimum number of terms J required to achieve a close approximation to the exact periodic squared exponential kernel as a function of the lengthscale of the kernel. We have considered an approximation to be close enough in terms of satisfying equation (13) with $\varepsilon = 0.5\%$. Notice that since this is a series expansion of sinusoidal functions, the approximation does not depend on any boundary condition.

The function values of a GP model with this low-rank representation of the periodic exponential covariance function can be easily derived. Considering the identity

$$\cos(j\omega_0(x-x')) = \cos(j\omega_0 x)\cos(j\omega_0 x') + \sin(j\omega_0 x)\sin(j\omega_0 x'),$$

the covariance $k(\tau)$ in equation (B.2) can be re-writting as

$$k(x,x') = \sigma^2 \Big(\sum_{j=0}^{J} \tilde{q}_j^2 \cos(j\omega_0 x)\cos(j\omega_0 x') + \sum_{j=1}^{J} \tilde{q}_j^2 \sin(j\omega_0 x)\sin(j\omega_0 x') \Big). \tag{B.6}$$

where $\tau = x - x'$. With this approximation for the periodic squared exponential covariance function $k(x,x')$, the approximate GP model $f(x) \sim \mathcal{GP}(0, k(x,x')$ equivalently leads to a linear representation of $f(\cdot)$ via

$$f(x) \approx \sigma \Big(\sum_{j=0}^{J} \tilde{q}_j \cos(j\omega_0 x)\beta_j + \sum_{j=1}^{J} \tilde{q}_j \sin(j\omega_0 x)\beta_{J+1+j} \Big), \tag{B.7}$$

where $\beta_j \sim \text{Normal}(0,1)$, with $j = 1, \ldots, 2J+1$. The cosine $\cos(j\omega_0 x)$ and sinus $\sin(j\omega_0 x)$ terms do not depend on the covariance hyperparameters ℓ. The only dependence on the hyperparameter ℓ is through the coefficients $\tilde{q}_j$, which are J-dimensional. The computational cost of this approximation scales as $O\big(n(2J+1) + (2J+1)\big)$, where n is the number of observations and J the number of cosine terms. The parameterization in equation (B.7) is naturally in the non-centered form with independent prior distributions on β_j, which makes posterior inference easier.

Acknowledgment

References

Abramowitz, M., & Stegun, I.A. 1970. *Handbook of mathematical functions*.

Adler, Robert J. 1981. *The geometry of random fields*. Vol. 62. SIAM.

Akhiezer, NI, & Glazman, IM. 1993. Theory of Linear Operators in Hilbert Space (Ungar, New York, 1963). *Vol. II*, 121–126.

Andersen, Michael Riis, Vehtari, Aki, Winther, Ole, & Hansen, Lars Kai. 2017. Bayesian Inference for Spatio-temporal Spike-and-Slab Priors. *Journal of Machine Learning Research*, **18**(139), 1–58.

Briol, François-Xavier, Oates, Chris, Girolami, Mark, Osborne, Michael A, Sejdinovic, Dino, *et al.* . 2015. Probabilistic integration: A role in statistical computation? *arXiv preprint arXiv:1512.00933*.

Brooks, Steve, Gelman, Andrew, Jones, Galin, & Meng, Xiao-Li. 2011. *Handbook of Markov Chain Monte Carlo*. CRC press.

Bürkner, Paul-Christian, *et al.* . 2017. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, **80**(1), 1–28.

Burt, David, Rasmussen, Carl Edward, & van der Wilk, Mark. 2019. Explicit rates of convergence for sparse variational inference in Gaussian process regression. *arXiv preprint arXiv:1903.03571*.

Carlin, Bradley P, Gelfand, Alan E, & Banerjee, Sudipto. 2014. *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC.

Carpenter, Bob, Gelman, Andrew, Hoffman, Matthew D, Lee, Daniel, Goodrich, Ben, Betancourt, Michael, Brubaker, Marcus, Guo, Jiqiang, Li, Peter, & Riddell, Allen. 2017. Stan: A probabilistic programming language. *Journal of statistical software*, **76**(1).

Cramér, Harald, & Leadbetter, M Ross. 2013. *Stationary and related stochastic processes: Sample function properties and their applications*. Courier Corporation.

Csató, Lehel, Fokoué, Ernest, Opper, Manfred, Schottky, Bernhard, & Winther, Ole. 2000. Efficient approaches to Gaussian process classification. *Pages 251–257 of: Advances in neural information processing systems*.

Deisenroth, Marc Peter, Fox, Dieter, & Rasmussen, Carl Edward. 2015. Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, **37**(2), 408–423.

Diggle, Peter J. 2013. *Statistical analysis of spatial and spatio-temporal point patterns*. Chapman and Hall/CRC.

Furrer, Eva M, & Nychka, Douglas W. 2007. A framework to understand the asymptotic properties of kriging and splines. *Journal of the Korean Statistical Society*, **36**(1), 57–76.

Gal, Yarin, & Turner, Richard. 2015. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. *Pages 655–664 of: International Conference on Machine Learning*.

Gelman, Andrew, Carlin, John B, Stern, Hal S, Dunson, David B, Vehtari, Aki, & Rubin, Donald B. 2013. *Bayesian data analysis*. Chapman and Hall/CRC.

Gibbs, Mark N, & MacKay, David JC. 2000. Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, **11**(6), 1458–1464.

GPy. 2012. *GPy: A Gaussian process framework in Python*.

Grenander, U. 1981. *Abstract inference*. John Wiley & Sons.

Hennig, Philipp, Osborne, Michael A, & Girolami, Mark. 2015. Probabilistic numerics and uncertainty in computations. *Proc. R. Soc. A*, **471**(2179), 20150142.

Hensman, James, Durrande, Nicolas, & Solin, Arno. 2017. Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research*, **18**(1), 5537–5588.

Jo, Seongil, Choi, Taeryon, Park, Beomjo, & Lenk, Peter. 2019. bsamGP: An R package for Bayesian spectral analysis models using Gaussian process priors. *Journal of Statistical Software, Articles*, **90**(10), 1–41.

Lázaro Gredilla, Miguel. 2010. Sparse Gaussian processes for large-scale machine learning.

Loève, M. 1977. *Probability theory*.

Lunn, David J, Thomas, Andrew, Best, Nicky, & Spiegelhalter, David. 2000. WinBUGS - a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, **10**(4), 325–337.

Matthews, Alexander G. de G., van der Wilk, Mark, Nickson, Tom, Fujii, Keisuke., Boukouvalas, Alexis, León-Villagrá, Pablo, Ghahramani, Zoubin, & Hensman, James. 2017. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, **18**(40), 1–6.

Minka, Thomas P. 2001. Expectation propagation for approximate Bayesian inference. *Pages 362–369 of: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.

Neal, Radford M. 1997. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. *arXiv preprint physics/9701026*.

Piironen, Juho, & Vehtari, Aki. 2016. On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. *arXiv preprint arXiv:1610.05559*.

Piironen, Juho, Vehtari, Aki, *et al.* . 2017. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, **11**(2), 5018–5051.

Quiñonero-Candela, Joaquin, & Rasmussen, Carl Edward. 2005. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, **6**(Dec), 1939–1959.

Quiñonero-Candela, Joaquin, Rasmussen, Carl Edward, Figueiras-Vidal, Aníbal R, *et al.* . 2010. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, **11**(Jun), 1865–1881.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. http://www.R-project.org/.

Rahimi, Ali, & Recht, Benjamin. 2008. Random features for large-scale kernel machines. *Pages 1177–1184 of: Advances in neural information processing systems*.

Rahimi, Ali, & Recht, Benjamin. 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Pages 1313–1320 of: Advances in neural information processing systems*.

Rasmussen, Carl Edward, & Nickisch, Hannes. 2010. Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, **11**, 3011–3015.

Rasmussen, Carl Edward, & Williams, Christopher KI. 2006. *Gaussian process for machine learning*. MIT press.

Roberts, Stephen J. 2010. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. Ph.D. thesis, University of Oxford.

Solin, Arno, & Särkkä, Simo. 2014. Explicit link between periodic covariance functions and state space models. *Pages 904–912 of: Artificial Intelligence and Statistics*.

Solin, Arno, & Särkkä, Simo. 2018. Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*.

Särkkä, Simo, Solin, Arno, & Hartikainen, Jouni. 2013. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, **30**(4), 51–61.

Trees, HLV. 1968. *Detection, estimation and modulation theory, vol. 1*.

Vanhatalo, Jarno, Riihimäki, Jaakko, Hartikainen, Jouni, Jylänki, Pasi, Tolvanen, Ville, & Vehtari, Aki. 2013. GPstuff: Bayesian modeling with Gaussian processes. *The Journal of Machine Learning Research*, **14**(1), 1175–1179.

Vehtari, Aki, Ojanen, Janne, *et al.* . 2012. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, **6**, 142–228. doi:10.1214/12-SS102.

Wahba, Grace. 1990. *Spline models for observational data*. Vol. 59. SIAM.

Williams, Christopher KI, & Barber, David. 1998. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(12), 1342–1351.

Wood, Simon, & Wood, Maintainer Simon. 2015. Package ‘mgcv’. *R package version*, **1**, 29.

Wood, Simon N. 2003. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **65**(1), 95–114.