Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming

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

[1] FISABIO-Public Health, Spain
[2] Excellence Cluster for Simulation Technology, University of Stuttgart, Germany
Most of work was done while at Dept. of Computer Science at Aalto University
[2] Department of Computer Science, Aalto University, Finland
[3] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark
* Corresponding author, Email: gabriuma@gmail.com

Abstract

Gaussian processes are powerful non-parametric probabilistic models for stochastic functions. However, the direct implementation entails 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 low-rank approximate Bayesian Gaussian processes, based on a basis function approximation via Laplace eigenfunctions for stationary covariance functions. The main contribution of this paper is a detailed analysis of the performance, and practical recommendations for how to select the number of basis functions and the boundary factor. Intuitive visualizations and recommendations, make it easier for users to improve approximation accuracy and computational performance. We also propose diagnostics for checking that the number of basis functions and the boundary factor are adequate given the data. The approach is simple and exhibits an attractive computational complexity due to its linear structure, and it is easy to implement 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 (Rasmussen and Williams, 2006; Neal, 1997).

Their name stems from the fact that any finite set of function values is jointly distributed as a multivariate Gaussian distribution. 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 (Rasmussen and Williams, 2006; Neal, 1997). Among others, they find application in the fields of spatial epidemiology (Diggle, 2013; Carlin et al., 2014), 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 (Roberts, 2010; Briol et al., 2015; Hennig et al., 2015).

The key element of a GP is the covariance function that defines the dependence structure between function values at different inputs. However, computing the posterior distribution of a GP comes with a computational issue because of the need to invert the covariance matrix. 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 by Quiñonero-Candela and Rasmussen

1

(2005), while a general review is provided by Rasmussen and 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 (Loève, 1977; Van Trees, 1968; Adler, 1981; Cramér and Leadbetter, 2013). Sparse spectrum GPs are based on a sparse approximation to the frequency domain representation of a GP (Lázaro Gredilla, 2010; Quiñonero-Candela et al., 2010; Gal and Turner, 2015). 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 and Recht, 2008, 2009). Certain spline smoothing basis functions are equivalent to GPs with certain covariance functions (Wahba, 1990; Furrer and Nychka, 2007). 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) is presented by Jo et al. (2019). Yet another approach is to present Gaussian process using precision matrix, which is the inverse of the covariance matrix. If the precision matrix is sparse, computation taking the benefit of that sparsity can scale much better than $O(n^3)$. See, for example, review by Lindgren et al. (2022).

In this paper, we focus on a recent framework for fast and accurate inference for fully Bayesian GPs using basis function approximations based on approximation via Laplace eigenfunctions for stationary covariance functions proposed by Solin and Särkkä (2020). 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 building blocks 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 precomputation cost of the basis functions is $O(m^2 n)$, 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 differentiation methods used in modern probabilistic programming frameworks, such as Stan (Carpenter et al., 2017) and others. This is because the memory requirements of automatic differentiation scale with the size of the autodiff expression tree which in direct implementations is simpler for basis function than covariance matrix-based approaches. 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 and Särkkä (2020) 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, domain of the prediction space, and properties of the true functional relationship between covariates and response variable, are investigated and analyzed in detail in relation to the computational performance and accuracy of the method. Practical recommendations are given for the values of the key factors based on simple diagnostic values and intuitive graphical summaries that encode the recognized relationships. Our recommendations help users to choose valid and optimized values for these factors, improving computational performance without sacrificing modeling accuracy. The proposed diagnostic can be used to check whether the chosen values for the number of basis functions and the domain of the prediction space are adequate to model the data well. On that basis, we also develop an iterative procedure to achieve accuracte approximation performance with minimal computational costs.

We have implemented the approach in the probabilistic programming language Stan (Carpenter et al., 2017) as well as subsequently in the *brms* package (Bürkner, 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 and 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 a small set of specific GP models, but instead we 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 and Särkkä (2020) is described. In Section 4, the accuracy of these approximations under several conditions is studied using both analytical and numerical methods. Practical diagnostics are developed there as well. 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 the 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 online 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 of a collection of random variables indexed by a continuous variable, that is, $\{f(t) : t \in \mathcal{T}\}$ for some index set $\mathcal{T}$. GPs have the defining property that the marginal distribution of any finite subset of random variables, $\{f(t_1), f(t_2), \ldots, f(t_N)\}$, 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 $p(y_n \mid f(\boldsymbol{x}_n), \phi)$, where p is some parametric distribution with parameters ϕ, and f is an unknown function with a GP prior, which depends on an input $\boldsymbol{x}_n \in \mathbb{R}^D$. This generalizes readily to more complex models depending on several unknown functions, for example such as $p(y_n \mid f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ 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 $\mu : \mathbb{R}^D \to \mathbb{R}$ and $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ are the mean and covariance functions, respectively,

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

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 $\boldsymbol{f} \sim \text{Normal}(\boldsymbol{\mu}, \boldsymbol{K})$, 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,

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

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 $p(y_n \mid f(\boldsymbol{x}_n), \phi) = \text{Normal}(y_n \mid f(\boldsymbol{x}_n), \sigma)$ 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

$p(y_n \mid f(\boldsymbol{x}_n), g(\boldsymbol{x}_n)) = \text{Normal}(y_n \mid f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ or $p(y_n \mid f(\boldsymbol{x}_n), \phi)$ 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 (Williams and Barber, 1998; Rasmussen and Williams, 2006), expectation propagation (Minka, 2001), or variational inference methods (Gibbs and MacKay, 2000; Csató et al., 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 characterizes the correlations between function values at different locations in the input space. A covariance function needs to be symmetric and positive semi-definite (Rasmussen and Williams, 2006). A stationary covariance function is a function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' \in \mathbb{R}^D$, 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 depend only on the input points through the norm of the difference, $k(\boldsymbol{x}, \boldsymbol{x}') = k(|\boldsymbol{x} - \boldsymbol{x}'|) = k(r), r \in \mathbb{R}$, 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,

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

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 $\nu = \infty$, $\nu = 3/2$ and $\nu = 5/2$ are probably the most commonly used kernels (Rasmussen and Williams, 2006),

$$k_\infty(r) = \alpha \exp\left(-\frac{1}{2}\frac{r^2}{\ell^2}\right),$$

$$k_{\frac{3}{2}}(r) = \alpha\left(1 + \frac{\sqrt{3}r}{\ell}\right)\exp\left(-\frac{\sqrt{3}r}{\ell}\right),$$

$$k_{\frac{5}{2}}(r) = \alpha\left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}r}{\ell}\right).$$

The former is commonly known as the squared exponential or exponentiated quadratic covariance function. As an example, assuming the Euclidean distance between observations, $r =$

$|\boldsymbol{x} - \boldsymbol{x}'|_{L2} = \sqrt{\sum_{i=1}^{D}(x_i - x_i')^2}$, the kernel K_ν written above takes the form

$$k_\infty(r) = \alpha \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i - x_i')^2}{\ell_i^2}\right).$$

Notice that the previous expressions $k_\infty(r)$ has been easily generalized to using a multidimensional length-scale $\boldsymbol{\ell} \in \mathbb{R}^D$. Using individual length-scales for each dimension turns an 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 (see, e.g., Rasmussen and 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 and 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 and Williams, 2006). The spectral density functions associated with the Matérn class of covariance functions are given by

$$S_\nu(\boldsymbol{\omega}) = \alpha\,\frac{2^D \pi^{D/2}\Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\,\ell^{2\nu}}\left(\frac{2\nu}{\ell^2} + 4\pi^2\boldsymbol{\omega}^\intercal\boldsymbol{\omega}\right)^{-(\nu+D)/2}$$

in D dimensions, where vector $\boldsymbol{\omega} \in \mathbb{R}^D$ is in the frequency domain, and ℓ and α are the length-scale and magnitude (marginal variance), respectively, of the kernel. The particular cases, where $\nu = \infty$, $\nu = 1/2$, $\nu = 3/2$ and $\nu = 5/2$, take the form

$$S_\infty(\boldsymbol{\omega}) = \alpha\,(\sqrt{2\pi})^D \ell^D \exp(-\frac{1}{2}\ell^2\boldsymbol{\omega}^\intercal\boldsymbol{\omega}), \qquad (1)$$

$$S_{\frac{3}{2}}(\boldsymbol{\omega}) = \alpha\,\frac{2^D\pi^{D/2}\Gamma(\frac{D+3}{2})3^{3/2}}{\frac{1}{2}\sqrt{\pi}\ell^3}\left(\frac{3}{\ell^2} + \boldsymbol{\omega}^\intercal\boldsymbol{\omega}\right)^{-\frac{D+3}{2}}, \quad (2)$$

$$S_{\frac{5}{2}}(\boldsymbol{\omega}) = \alpha\,\frac{2^D\pi^{D/2}\Gamma(\frac{D+5}{2})5^{5/2}}{\frac{3}{4}\sqrt{\pi}\ell^5}\left(\frac{5}{\ell^2} + \boldsymbol{\omega}^\intercal\boldsymbol{\omega}\right)^{-\frac{D+5}{2}}. \quad (3)$$

For instance, with input dimensionality $D = 3$ and $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\intercal$, the spectral densities written above take the form

$$S_\infty(\boldsymbol{\omega}) = \alpha\,(2\pi)^{3/2}\prod_{i=1}^{3}\ell_i\exp\left(-\frac{1}{2}\sum_{i=1}^{3}\ell_i^2\omega_i^2\right),$$

$$S_{\frac{3}{2}}(\boldsymbol{\omega}) = \alpha\,32\,\pi 3^{3/2}\prod_{i=1}^{3}\ell_i\left(3 + \sum_{i=1}^{3}\ell_i^2\omega_i^2\right)^{-3},$$

$$S_{\frac{5}{2}}(\boldsymbol{\omega}) = \alpha\,\frac{64}{3}\,\pi 5^{5/2}\prod_{i=1}^{3}\ell_i\left(5 + \sum_{i=1}^{3}\ell_i^2\omega_i^2\right)^{-4}.$$

where individual length-scales ℓ_i for each frequency dimension ω_i have been used.

3 Hilbert space approximate Gaussian process model

The approximate GP method, developed by Solin and Särkkä (2020) and further analysed in this paper, is based on consid-

ering 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 $\Omega \subset \mathbb{R}^D$ 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 on the theoretical background are provided by Solin and Särkkä (2020). Our starting point for presenting the method is 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.

3.1 Unidimensional GPs

We begin by focusing on the case of a unidimensional input space (i.e., on GPs with just a single covariate) such that $\Omega \in [-L, L] \subset \mathbb{R}$, 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

$$k(x, x') = \sum_{j=1}^{\infty} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x'), \qquad (4)$$

where S_θ is the spectral density of the stationary covariance function k (see Section 2.1) and θ is the set of hyperparameters of k (Rasmussen and 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)

$$\begin{aligned} -\nabla^2\phi_j(x) &= \lambda_j\phi_j(x), \qquad & x \in \Omega \\ \phi_j(x) &= 0, \qquad & x \notin \Omega. \end{aligned} \qquad (5)$$

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 eq. (5) are sinusoidal functions. The solution to the eigenvalue problem is independent of the specific choice of covariance function and is given by

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2, \qquad (6)$$

$$\phi_j(x) = \sqrt{\frac{1}{L}}\sin\left(\sqrt{\lambda_j}(x + L)\right). \qquad (7)$$

If we truncate the sum in eq. (4) 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 $\boldsymbol{\phi}(x) = \{\phi_j(x)\}_{j=1}^m \in \mathbb{R}^m$ is the column vector of basis functions, and $\Delta \in \mathbb{R}^{m \times m}$ is a diagonal matrix of the spectral density evaluated at the square root of the eigenvalues, that is, $S_\theta(\sqrt{\lambda_j})$,

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

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

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

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

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

This equivalently leads to a linear representation of f via

$$f(x) \approx \sum_{j=1}^m \left(S_\theta(\sqrt{\lambda_j})\right)^{\frac{1}{2}} \phi_j(x)\beta_j, \quad (8)$$

where $\beta_j \sim \text{Normal}(0,1)$. 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. (8) 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 evaluating the log posterior density 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. (8) is naturally in the non-centered parameterization form with independent prior distribution on β_j, which can make the posterior inference easier (see, e.g., Betancourt and Girolami, 2019). 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.

3.2 Generalization to multidimensional GPs

The results from the previous section can be generalized 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

$$m^* = \prod_{d=1}^D m_d, \quad (9)$$

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

$$\mathbb{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}^\intercal. \quad (10)$$

Each multivariate eigenfunction $\phi_j^* : \Omega \to \mathbb{R}$ 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 $j = 1, 2, \ldots, m^*$, we have

$$\boldsymbol{\lambda}_j^* = \{\lambda_{\mathbb{S}_{jd}}\}_{d=1}^D = \left\{ \left(\frac{\pi\mathbb{S}_{jd}}{2L_d}\right)^2 \right\}_{d=1}^D, \quad (11)$$

$$\phi_j^*(\boldsymbol{x}) = \prod_{d=1}^D \phi_{\mathbb{S}_{jd}}(x_d) = \prod_{d=1}^D \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\right). \quad (12)$$

The approximate covariance function is then represented as

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j=1}^{m^*} S_\theta^* \left(\sqrt{\boldsymbol{\lambda}_j^*}\right) \phi_j^*(\boldsymbol{x})\phi_j^*(\boldsymbol{x}'), \quad (13)$$

where S_θ^* is the spectral density of the D-dimensional covariance function (see Section 2.1) as a function of $\sqrt{\boldsymbol{\lambda}_j^*}$ that denotes the element-wise square root of the vector $\boldsymbol{\lambda}_j^*$. We can now write the approximate series expansion of the multivariate function f as

$$f(\boldsymbol{x}) \approx \sum_{j=1}^{m^*} \left(S_\theta^*\left(\sqrt{\boldsymbol{\lambda}_j^*}\right)\right)^{\frac{1}{2}} \phi_j^*(\boldsymbol{x})\beta_j, \quad (14)$$

where, again, $\beta_j \sim \text{Normal}(0,1)$. The computational cost of evaluating the log posterior density of multivariate HSGPs

5

scales as $O(nm^* + m^*)$, where n is the number of observations and m^* is the number of multivariate basis functions. 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.

3.3 Linear representation of a periodic squared exponential covariance function

A GP model with a periodic covariance function does no fit in the framework of the HSGP approximation covered in this study as a periodic covariance function has not a spectral representation, but it has also a low-rank representation. In Appendix B, we briefly present the approximate linear representation of a periodic squared exponential covariance function as developed by Solin and Särkkä (2014), analyze the accuracy of this approximation and, finally, derive the GP model with this approximate periodic squared exponential covariance function.

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 function to be estimated, which is in turn characterized by the length-scale 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 m, the boundary condition L, and the length-scale ℓ, which depend on the particular choice of the kernel function. In this work we investigate these relationships for the squared exponential and the Matérn ($\nu = 3/2$ and $\nu = 5/2$) covariance functions in the present section, and for the periodic squared exponential covariance function in Appendix B. For other kernels, the relationships will be slightly different depending on the smoothness or wigglyness of the covariance function.

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 $\Omega = [-L, L] \subset \mathbb{R}$

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

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 (6) and (7) for the unidimensional case and in equations (11) and (12) for the multidimensional case. The number of basis functions can be truncated at some finite positive value m such that the total variation difference between the exact and approximate covariance functions is less than a predefined threshold $\varepsilon > 0$:

$$\int |k(\tau) - \sum_{j=1}^{m} S_\theta(\sqrt{\lambda_j})\phi_j(\tau)\phi_j(0)| \, \mathrm{d}\tau < \varepsilon. \qquad (15)$$

This inequality can be satisfied for arbitrary small ϵ provided that L and m are sufficiently large (Solin and Särkkä, 2020, Theorem 1 and 4). The specific number of basis functions m needed depends on the degree of non-linearity of the function to be estimated, that is on its length-scale ℓ, which constitutes a hyperparameter of the GP. The approximation also depends on the boundary L (see equations (6), (7), (11) and (12)), 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 this work, we choose L such that the domain $\Omega = [-L, L]$ contains all the inputs points x_i. Without loss of generality, we can assume all data points are contained in a symmetric interval around zero. Let $S = \max_i |x_i|$, then it follows that $x_i \in [-S, S]$ for all i. We now define L as

$$L = c \cdot S, \qquad (16)$$

where $S > 0$ 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.

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 a squared exponential covariance function of length-scale $\ell = 0.3$ and marginal variance $\alpha = 1$, using input values from the zero-mean input domain with half-range $S = 1$. Several HSGP models with varying m and c are fitted to this data. In this example, the length-scale 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 predictive mean and standard deviation of the estimated function as well as on the covariance function itself. For a sufficiently large fixed value of c, Figure 1 shows clearly how m affects the accuracy on the approximation for both the posterior mean or uncertainty. It is seen that if the number of basis functions m is too small, the estimated function tend to be overly smooth because the necessary high frequency components are missing. In general, the higher the degree of wigglyness of the function to be estimated, the larger number
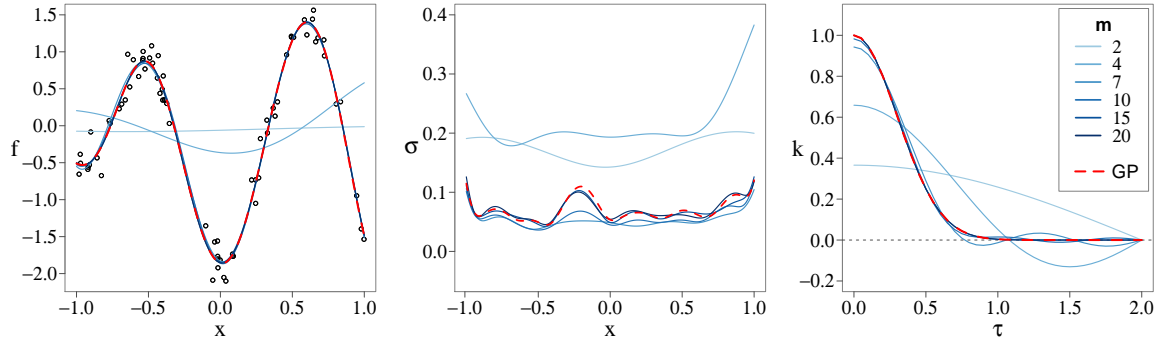
Figure 1: Mean posterior predictive functions (left), posterior standard deviations (center), and covariance functions (right) of both the exact GP model (dashed red line) and the HSGP model for different number of basis functions m, with the boundary factor fixed to a large enough value.
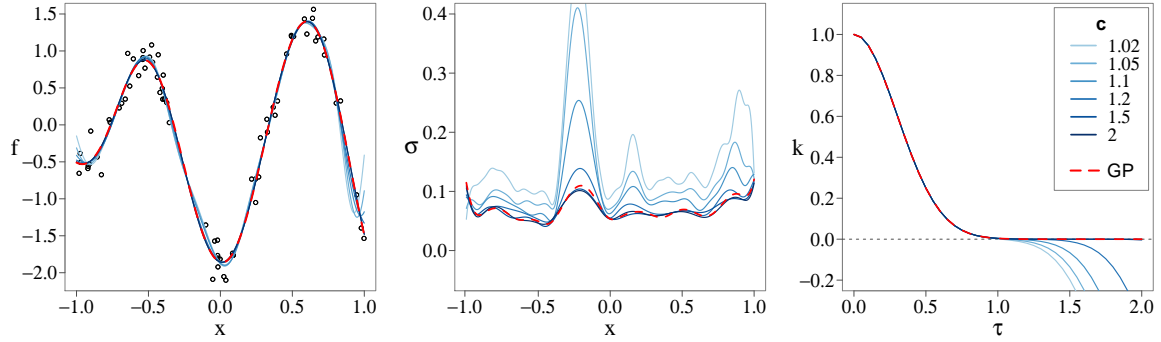

Figure 2: Mean posterior predictive functions (left), posterior standard deviations (center), and covariance functions (right) of both the exact GP model (dashed red line) and the HSGP model for different values of the boundary factor c, with a large enough fixed number of basis functions.

of basis functions will be required. If m is fixed to a sufficiently large value, Figure 2 shows that c affects the approximation of the mean mainly near the boundaries, while the approximation of the standard deviation is affected across the whole domain. The approximation error tends to be bigger for the standard deviation than for the mean.

Next, we analyze how the interaction between m and c affects the quality of the approximation. The length-scale and marginal variance of the covariance function will no longer be fixed but instead we compute the joint posterior distribution of the function values and the hyperparameters using the dynamic HMC algorithm implemented in Stan (Stan Development Team, 2021) for both the exact GP and the HSGP models. Figure 3 shows the posterior predictive mean and standard deviation of the function as well as the covariance function obtained after fitting the model for varying m and c. Figure 4 shows the root mean square error (RMSE) of the HSGP models computed against the exact GP model. Figure 5 shows the estimated length-scale and marginal variance for the exact 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 computation time for this example 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. (6), where L appears in the denominator.

- There exists a minimum c below which an accurate approximation will never be achieved regardless of the number of basis functions m.

4.2 Near linear proportionality between m, c and ℓ

A priori, the terms in the series expansion (14) with very small spectral density are unlikely to contribute to the approximation. Given the boundary factor c and the length-scale ℓ, we can compute the cumulative sum of the spectral densities and find out how many basis functions are, a priori, explaining almost 100% of the variation. Thus, given c and ℓ, we can estimate a good choice for the number of basis functions m for any covariance function.
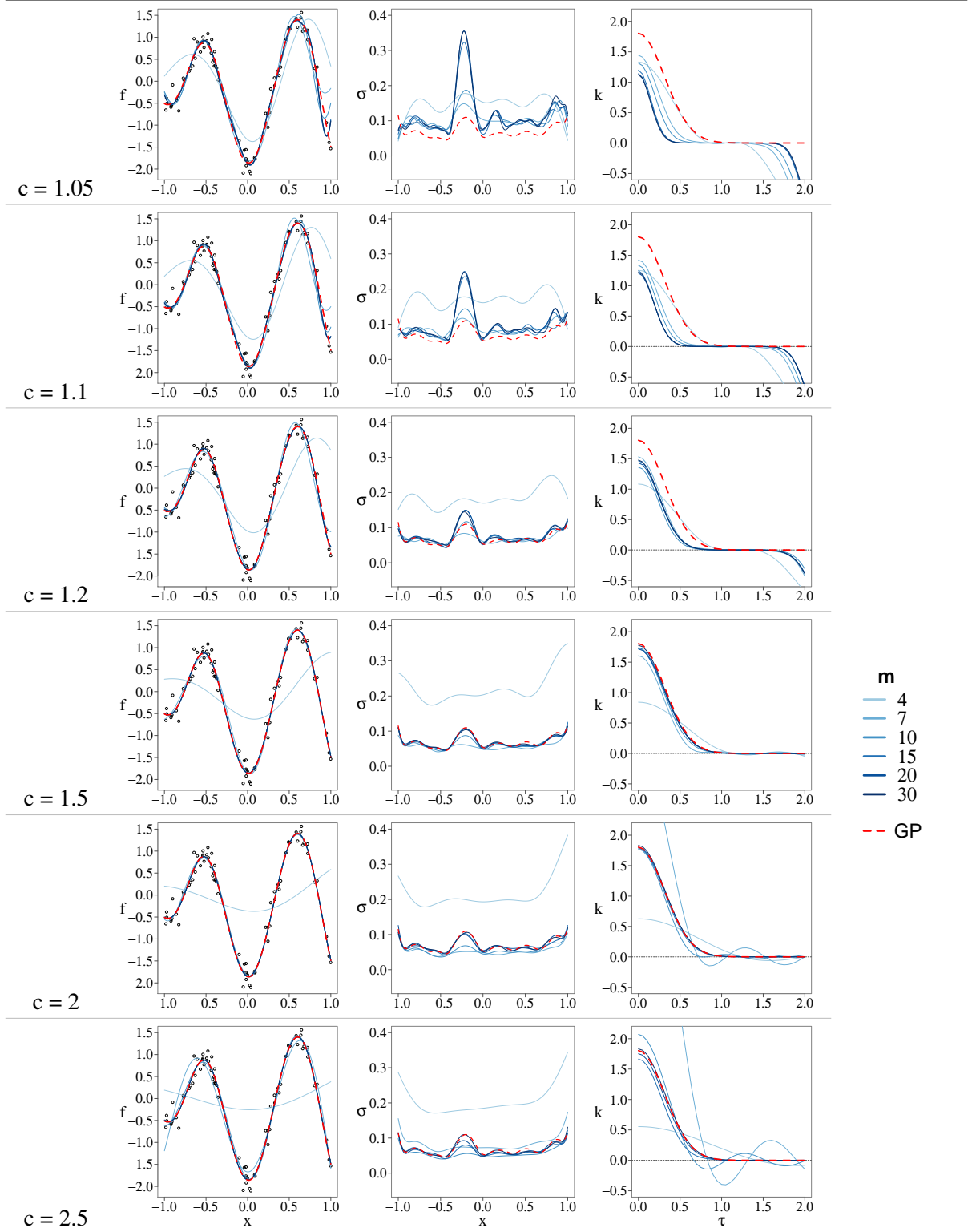
Figure 3: Posterior mean predictive functions (left), posterior standard deviations (center) and covariance functions (right) of both the exact GP model and the HSGP model for different number of basis functions m and for different values of the boundary factor c.
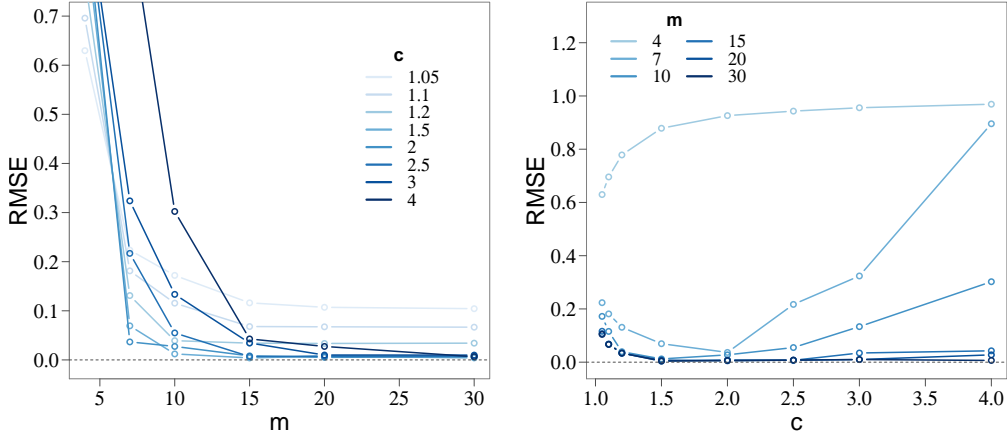
Figure 4: Root mean square error (RMSE) of the proposed HSGP models computed against the exact GP model. RMSE versus the number of basis functions m and for different values of the boundary factor c (left). RMSE versus the boundary factor c and for different values of the number of basis functions m (right).
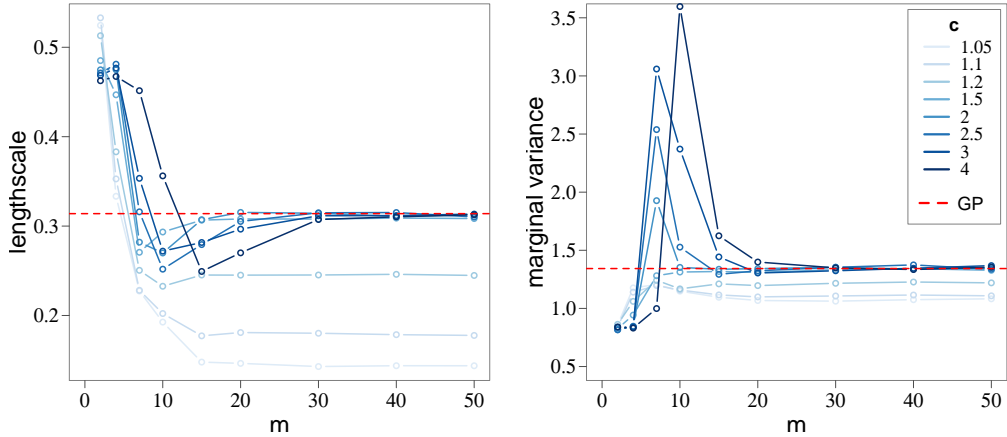
Figure 5: Estimated length-scale (left) and marginal variance (right) parameters of both exact GP and HSGP models, plotted versus the number of basis functions m and for different values of the boundary factor c.

When considering squared exponential and Matérn covariance functions, we can show with simple algebra that when c is larger than the minimal value recommendation, the number of m first terms needed to explain almost 100% of the variation has a near linear relationship with ℓ and c. With decreasing ℓ, the m should grow near linearly with $1/\ell$, and with increasing c, the m should grow near linearly with c. This is natural as with decreasing ℓ, more higher frequency basis functions are needed. With increasing c, as a smaller range of the basis functions are used in the approximation, the expected number of zero up-crossings goes down linearly with c, and thus more higher frequency basis functions are needed to compensate this. When c is below our recommendations given ℓ, the effect of ℓ and c to the recommended m is more non-linear, but as long as we stay in the recommended range the linearity assumption is useful thumb rule for how to change m, if ℓ or c are changed.

4.3 Empirical discovering of the functional form of the relationships between m, c and ℓ

Empirical simulations are carried out to analyze the relationships between m, c and $\frac{\ell}{S}$ (lengthscale ℓ relative to the half-range S of the input domain). Figure 6 depicts how these three factors interact and affect the accuracy of the HSGP approximation for a GP with squared exponential, Matérn ($\nu=3/2$), and Matérn ($\nu=5/2$) covariance functions and a single input dimension. More precisely, for a given GP model (with a particular covariance function) with length-scale $\frac{\ell}{S}$ (From now to later on we will refer to $\frac{\ell}{S}$ by just ℓ) and given a boundary factor c, Figure 6 shows the minimum number of basis functions m required to obtain an accurate approximation in the sense of satisfying eq. (15). We considered an approximation to be a sufficiently accurate when the total variation difference be-
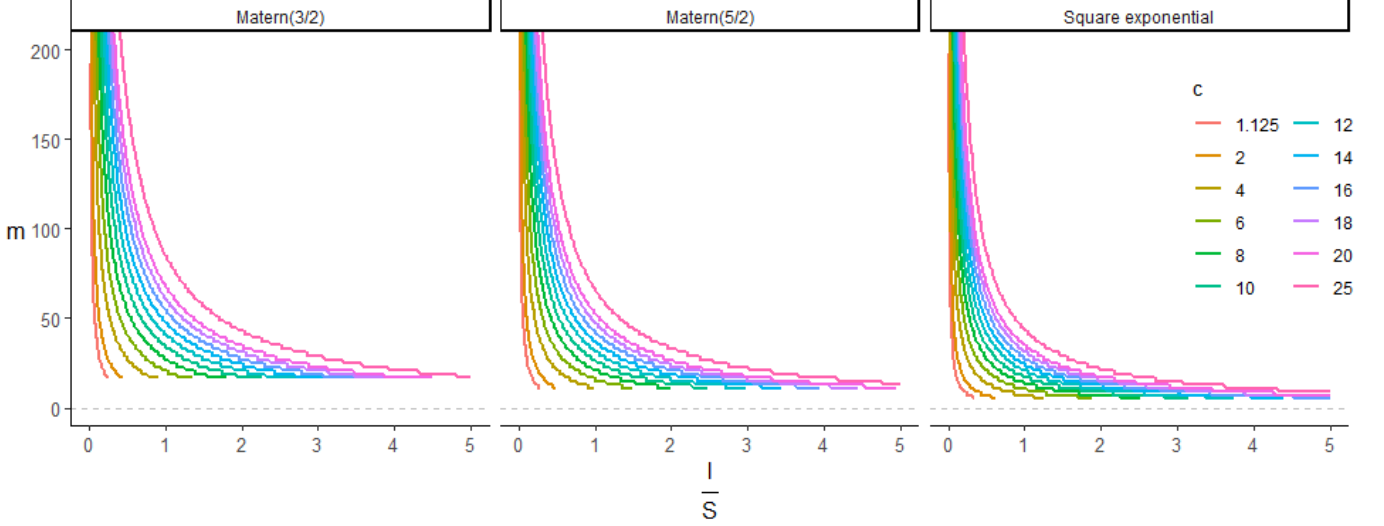
Figure 6: Relation among the minimum number of basis functions m, the boundary factor c ($c = \frac{L}{S}$) and the length-scale normalized by the half-range of the data ($\frac{\ell}{S}$), for the squared exponential, Matérn (ν=3/2) and Matérn (ν=5/2) covariance functions.

tween the approximate and exact covariance functions, ε in eq. (15), is below 1% of the total area under the curve of the exact covariance function k such that

$$\int |k(\tau) - \tilde{k}_m(\tau)|\,\mathrm{d}\tau < 0.01 \int k(\tau)\,\mathrm{d}\tau, \qquad (17)$$

where $\tilde{k}_m$ is the approximate covariance function with m basis functions. Alternatively, these figures can 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 approximated with high accuracy given m and c. We obtain the following main conclusions:

- As ℓ increases, m required for an accurate approximation decrease.

- The lower c, the smaller m can and ℓ must be to achieve an accurate approximation.

- For a given ℓ there exist a minimum c under which an accurate approximation is never going to be achieved regardless of m. This fact can be seen in Figure 6 as the contour lines which represent c have an end in function of ℓ (Valid c are restricted in function of ℓ). As ℓ increases, the minimum valid c also increases.

4.3.1 Numerical equations

As explained in Section 4.2, when c is large enough, there is a near linear proportionality between m, ℓ and c. To obtain practical numerical functions that can be used to guide the selection of these parameters, we have empirically derived the practically useful constant terms. We require a lower bound for c of $c \geq 1.2$ such that the equations below are precise enough for practical application.

- Squared exponential:

$$m = 1.75\,\frac{c}{\ell/S} \quad \Leftrightarrow \quad \ell/s = 1.75\,\frac{c}{m}, \qquad (18)$$

with
$$c \geq 3.2\,\ell/s \ \ \& \ \ c \geq 1.2 \qquad (19)$$

- Matérn (ν=5/2):

$$m = 2.65\,\frac{c}{\ell/S} \quad \Leftrightarrow \quad \ell/s = 2.65\,\frac{c}{m}, \qquad (20)$$

with
$$c \geq 4.1\,\ell/s \ \ \& \ \ c \geq 1.2 \qquad (21)$$

- Matérn (ν=3/2):

$$m = 3.42\,\frac{c}{\ell/S} \quad \Leftrightarrow \quad \ell/s = 3.42\,\frac{c}{m}, \qquad (22)$$

with
$$c \geq 4.5\,\ell/s \ \ \& \ \ c \geq 1.2 \qquad (23)$$

These constants vary monotonically with respect to ν (squared exponential corresponding to Matérn with $\nu \to \infty$). Using the formula for Matérn (ν=3/2) provides the largest m and c, and thus this formula alone could be used as a conservative choice for all Matérn covariance functions with $\nu \geq 3/2$ and likely as a good initial guess for many other covariance functions. If the aim is to find minimal m to speedup the computation, a further refined formula can be obtained for new covariance functions.

Figure 6 and previous equations (18)-(21) were obtained for a GP with a unidimensional covariance function, which result in a surfaces depending on three variables, m, c and ℓ. Equivalent results 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. More precisely, in the multi-dimensional case,

whether the approximation is close enough might depend only on the ratio between wigglyness in every dimensions. For instance, in the two-dimensional case, it would depend on the ratio between ℓ_1 and ℓ_2. 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 Figure 6 or equations (18)-(21) to check the accuracy by analyzing individually the different dimensions of a multidimensional GP model.

4.4 Relationships between m and ℓ for a periodic squared exponential covariance function

As commented in Section 3.3, in Appendix B we present an approximate linear representation of a periodic squared exponential covariance function. In Appendix B, we also analyze the accuracy of this linear representation and derive the minimum number of terms m in the approximation required to achieve a close approximation to the exact periodic squared exponential kernel as a function of the length-scale ℓ of the kernel. Since this is a series expansion of sinusoidal functions, the approximation does not depend on any boundary condition. This relationship between m and ℓ for a periodic squared exponential covariance function is gathered in Figure B.1 and the numerical equation was estimated in eq. (B.6) which is depicted next:

$$ m \geq \frac{3.72}{\ell} \quad \Leftrightarrow \quad \ell \geq \frac{3.72}{m}. $$

4.5 Diagnostics of the approximation

Equations (18), (20), (22) and (B.6) (depending on which kernel is used) provide the minimum length-scale that can be accurately inferred given m and c. This information serves as a powerful diagnostic tool in determining if the obtained accuracy is acceptable. As the length-scale ℓ controls the wigglyness of the function, it strongly influences the difficulty of estimating the latent function from the data. Basically, if the length-scale estimate is accurate, we can expect the HSGP approximation to be accurate as well.

Having obtained an estimate $\hat{\ell}$ for a HSGP model with prespecified m and c, we can check whether $\hat{\ell}$ exceeds the smallest length-scale, that can be accurately inferred, provided as a function of m and c by equations (18), (20), (22) and (B.6) (depending on which kernel is used). If $\hat{\ell}$ exceeds this value, the approximation is assumed to be good. If $\hat{\ell}$ does not exceed this value, the approximation may be inaccurate, and m and/or c need to be increased. In Figures 4 and 5, $m = 10$ and $c = 1.5$ were sufficient for an accurate modeling of function with $\ell = 0.3$, which matches the diagnostic based on equations (18) and (19).

Equations in Section 4.3.1 to update m and c imply:

- c must be big enough for a given ℓ, and

- m must be big enough for given ℓ and c.

If larger than minimal c and m (for a given ℓ) are used in the initial HSGP model, it is likely that the results are already sufficiently accurate. As ℓ is initially unknown, we recommend using this diagnostic in an iterative procedure by starting with c and m based on some initial guess about ℓ, and if the estimated $\hat{\ell}$ is below the diagnostic threshold, select new c and m using $\hat{\ell}$. This can be repeated until

- the estimated $\hat{\ell}$ is larger than the diagnostic threshold given c and m, and

- the predictive accuracy measures, for example, root mean square error (RMSE), coefficient of determination (R^2), or expected log predicitve density (ELPD) do not improve.

As commented above, the estimated $\hat{\ell}$ being larger than diagnostic threshold does not guarantee that the approximate is sufficiently accurate, and thus we recommend to look at the predicitve accuracy measures, too.

Apart from providing a powerful diagnostic tool in determining if the approxiamtion is sufficiently accurate, the equations in the previous Section 4.3.1 also provide the optimal values for m (the minimum m required for an accurate approximation) and c (the minimum c that allows for the minimum m) that van be used to minimize the computational cost in repeated computations (e.g., in cross-validation and simulation based calibration). This is even more useful in multi-dimensional cases ($2 \leq D \leq 4$), where knowing the smallest useful value of m for each dimension has even bigger effect on the total computational cost.

4.5.1 A step-by-step user-guide to apply the diagnostics

Based on the above proposed diagnostics, we obtain a simple, iterative step-by-step procedure that users can follow in order to obtain an accurate HSGP approximation. The procedure is split into two phases, Phase A and B, which have to be completed consecutively.

Phase A:

A1. Make an initial guess on the length-scale ℓ_1. If there is no useful information available, we recommend to start with a large length-scale, about $\ell_1 \in [0.5, 1]$. Set the iteration index to $i = 1$.

A2. Obtain the minimum valid boundary factor c_i determined by $\ell = \ell_i$ for the given kernel as per Section 4.3.1.

A3. Obtain the mimimum valid number of basis functions m_i determined by $\ell = \ell_i$ and $c = c_i$ for the given kernel as per Section 4.3.1.

A4. Fit an HSGP model using m_i and c_i and ensure convergence of the MCMC chains.

11

A5. Perform the length-scale diagnostic by checking if $\hat{\ell}_i - 0.01 \geq \ell_i$.

If the diagnostic is FALSE, the HSGP approximation is not yet sufficiently accurate. Set $\ell_{i+1} = \hat{\ell}_i$, increase the iteration index $i = i + 1$, and go back to A2.

If the diagnostic is TRUE, the HSGP approximation is close to be sufficiently accurate. Continue with Phase B.

Phase B:

B1. For the current HSGP model, compute measures of predictive accuracy, for example, RMSE, R^2, or ELPD.

B2. Set $m_{i+1} = m_i + 5$ and increase the iteration index $i = i + 1$.

B3. Obtain the minimum valid boundary factor c_i determined by $\ell = \hat{\ell}_{i-1}$ for the given kernel as per Section 4.3.1.

B4. Obtain the minimum valid length-scale ℓ_i that can be accurately estimated by $m = m_i$ and $c = c_i$ for the given kernel as per Section 4.3.1.

B5. Fit an HSGP model using m_i and c_i and ensure convergence of the MCMC chains.

B6. Perform the length-scale diagnostic by checking if $\hat{\ell}_i - 0.01 \geq \ell_i$.

Check the stability of both $\hat{\ell}_i$ and the measures of predictive accuracy relative to the previous iteration.

If all the stability checks succeed, the HSGP approximation of the latest model should be sufficiently accurate and the procedure ends here.

Otherwise, go back to B1.

4.6 Performance analysis of the diagnostics

In this section, we first illustrate that accurate estimates of the length-scale implies accurate approximations via HSGPs. Figure 7 left shows a comparison of the length-scale estimates obtained from the exact GP and HSGP models with a squared exponential kernel, from various noisy datasets drawn from underlying functions with varying smoothness. Different values for 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 by using eq. (19). Figure 7 (right) shows the RMSE of the HSGP models with the exact GP model as the reference. It can be seen that accurate estimates of the length-scale imply small RMSEs.

Table 1 shows the iterative steps of applying the diagnostic procedure explained in Section 4.5.1 over some of the data sets also used in the analysis in Figure 7 left. It is clearly visible that by following our recommendations, an optimum solution with minimum computational requirements is achieved in these cases. Figure 8 graphically compares the exact GP length-scale and the estimated HSGP length-scale in every iteration and data set. Between two and four iterations, depending on wigglyness of the function to be learned and the distance between the initial

guess of the length scale and the true length scale, are sufficient to reach the optimal values of m and c.

As concrete examples, the iterative steps applied to perform diagnostic on two of the data sets in Table 1 are described in Appendix C.

4.7 Other covariance functions

Above, we thoroughly studied the relationship between the number of basis functions in the approximation and the approximation accuracy across different configurations. We specifically focused on the Matérn and squared exponential families of covariance functions, yet there exists other families of stationary covariance functions. The basis function approximation can easily be implemented for any stationary covariance function, where the spectral density function is available. The assess the accuracy of a basis function approximation for a kernel, where the diagnostic plots like Figure 6 or equations (18)-(21) are not available, we suggest to use the relative total variational distance between the true covariance function and the approximate covariance function as given in eq. (17). Ensuring that the relative distance is bounded by a small constant for the relevant lengthscale implies a high quality approximation. Another possibility to asses the accuracy of the approximation is to look at the cumulative sum of the spectral densities terms used in the series expansion and find out how much of the total variation they are actually explaining.

To select c for many other covariance functions, users can be guided by equations (19), (21) and (23), as pointed in Section 4.3.1.

4.8 The computational cost in the multi-dimensional setting

The HSGP model is computationally superior in 1D and 2D even for highly wiggly functions, except when the number of data points is so small ($n \lesssim 300$, i.e., n smaller than some value around 300) that exact GPs are already reasonably fast themselves. However, the computation time of the HSGP model increases rapidly with the number of input dimensions (D) since the number of multivariate basis functions $m^* = m_1 \times \cdots \times m_D$ in the approximation increases exponentially with D (see eq. (9)). Yet, the HSGP method can still be computationally faster than the exact GP for larger datasets due the latter's cubic scaling in n.

In our experiments of multivariate problems (see Section 5.3.1), the computation time for the HSGP model was faster than for the exact GP for most of the non-linear $2D$ functions or moderate-to-large sized 3D datasets ($n \gtrsim 1000$, i.e., n greater than some value around 1000), even for highly wiggly 3D functions (e.g., $\ell_1, \ell_2, \ell_3 \approx 0.1$).

For small sized datasets ($n \lesssim 1000$), HSGPs are likely to be slower than exact GPs already for highly to moderated wiggly 3D functions (e.g., $\ell_1 \lesssim 0.1$, and $\ell_2, \ell_3 \gtrsim 0.3$) and for overall smooth 4D functions (e.g., $\ell_1 \lesssim 0.1$, and $\ell_2, \ell_3, \ell_4 \gtrsim 0.4$).
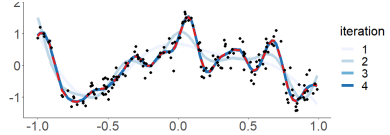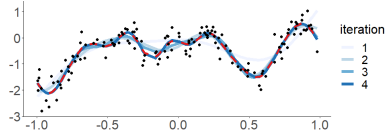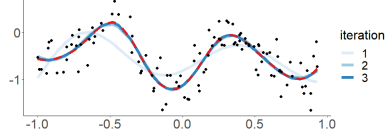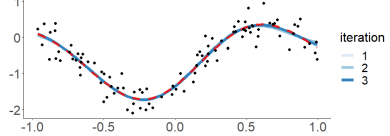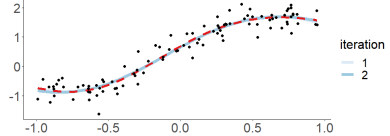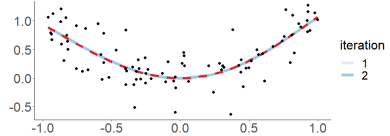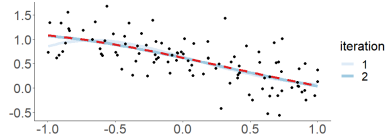
ℓ_{GP}	iter.	ℓ	c	m	$\hat{\ell}$	$\hat{\ell}+0.01 \geq \ell$	RMSE	R^2	ELPD	RMSE*	
0.08	1	0.50	1.60	6	0.17	FALSE	0.45	0.61	-0.40	0.36	
	2	0.17	1.20	13	0.07	FALSE	0.36	0.74	-0.18	0.24	
	3	0.07	1.20	31	0.08	TRUE	0.24	0.87	0.20	0.01	
	4	0.06	1.20	36	0.08	TRUE	0.24	0.87	0.21	0.01	
0.13	1	0.50	1.60	6	0.25	FALSE	0.46	0.63	-0.49	0.33	
	2	0.25	1.20	9	0.15	FALSE	0.35	0.78	-0.14	0.15	
	3	0.15	1.20	15	0.15	TRUE	0.32	0.81	-0.07	0.08	
	4	0.11	1.20	20	0.12	TRUE	0.30	0.83	-0.01	0.01	
0.25	1	0.50	1.60	6	0.24	FALSE	0.37	0.40	-0.15	0.21	
	2	0.24	1.20	9	0.26	TRUE	0.30	0.60	-0.09	0.02	
	3	0.25	1.20	14	0.26	TRUE	0.29	0.62	-0.05	0.01	
0.54	1	0.50	1.60	6	0.38	FALSE	0.27	0.86	0.03	0.01	
	2	0.38	1.20	6	0.47	TRUE	0.28	0.86	0.03	0.05	
	3	0.24	1.50	11	0.53	TRUE	0.27	0.86	0.04	0.01	
0.80	1	1.00	3.20	6	0.52	FALSE	0.29	0.92	0.13	0.03	
	2	0.52	1.65	6	0.98	TRUE	0.29	0.92	0.11	0.03	
	3	0.50	3.12	11	0.77	TRUE	0.28	0.92	0.11	0.01	
1.40	1	1.00	3.20	6	1.02	TRUE	0.28	0.56	0.14	0.01	
	2	0.52	3.27	11	1.23	TRUE	0.28	0.57	0.14	0.00	
2.94	1	0.50	1.60	6	1.19	TRUE	0.34	0.45	-0.10	0.06	
	2	0.61	3.81	11	2.58	TRUE	0.34	0.45	-0.07	0.01	

Table 1: Diagnostic on various data sets with varying smoothness. ℓ_{GP} refers to the exact GP length-scale estimate. ℓ refers to the minimum valid length-scale that can be accurately estimated determined by m and c as per Section 4.3.1, except for the first iteration which is the initial guess of the length-scale. $\hat{\ell}$ refers to the HSGP length-scale estimate. RMSE* refers to the root mean squared error computed with the exact GP model as the reference.
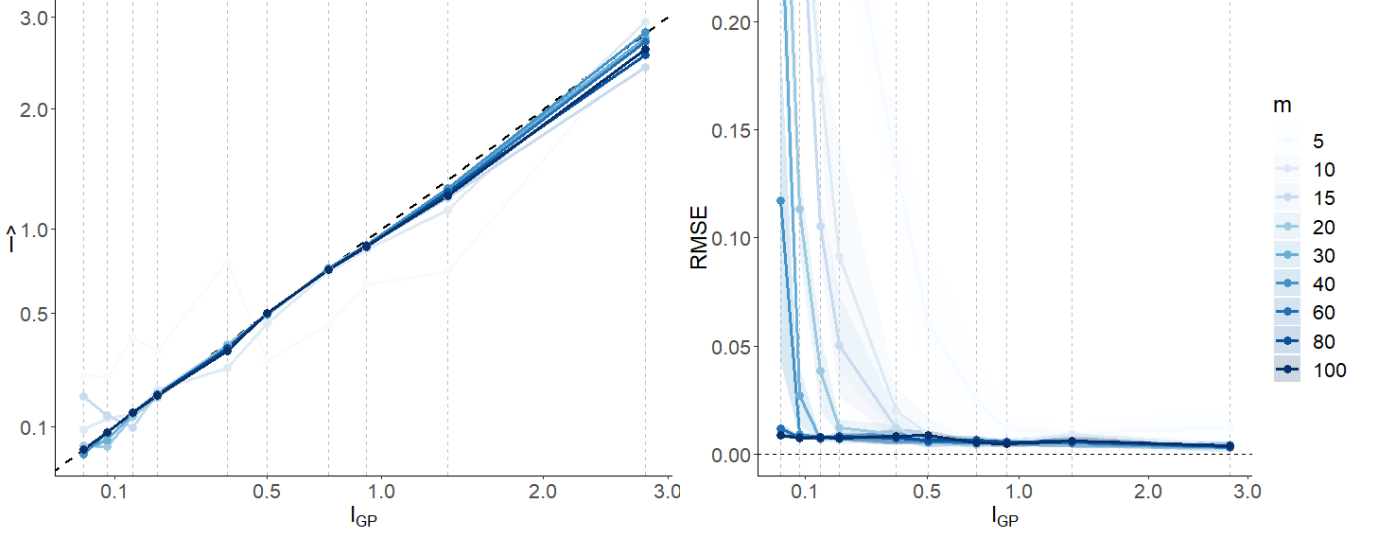
Figure 7: Exact GP length-scale (ℓ_{GP}) and approximate HSGP length-scale ($\hat{\ell}$) for various datasets with varying smoothness (left) and their corresponding root mean squared errors (RMSE) computed with the exact GP model as the reference.
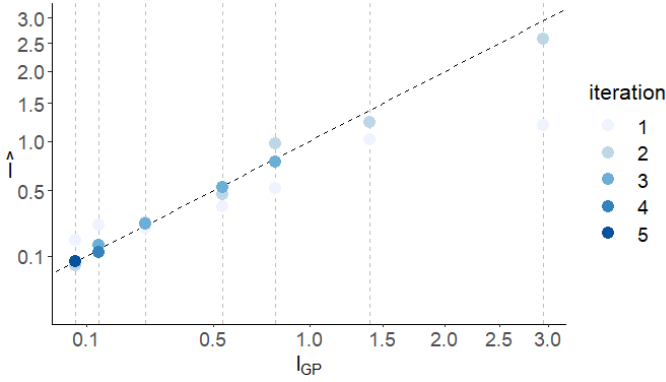

Figure 8: Comparison of the true and estimated length-scales in every iteration for the different data sets varying the length-scale.

As it has been shown in case study III (Section 5.3), the proposed diagnostic tool can be very useful for multivariate problems as it allows one to reduce m^* to the minimum sufficient value, reducing computational time drastically, and still getting an accurate approximation. For example, assuming a squared exponential covariance function, choosing the optimal value for c allows one to use few basis functions in every single dimension (i.e., $m \lesssim 10$ for $\ell \gtrsim 0.3$; $20 \gtrsim m \gtrsim 10$ for $0.3 \gtrsim \ell \gtrsim 0.1$; and $m \gtrsim 20$ for $\ell \lesssim 0.1$), which, from results presented in Figure 15, implies that the HSGP model can be, in general terms, useful for highly wiggly 3D functions and smooth 4D functions.

Whether HSGP or exact GP is faster will also depend on the specific implementation details, which can have big effects on the scaling constants. Thus, more detailed recommendations would depend on the specific software implementation.

In our experiments, we observed roughly similar or even worse

scaling behavior for the thin plate splines in comparison to HSGPs, with serious challenges with computation time when doing inference with spline models for $D = 3$ and more than 10 basis functions per dimension, or even for $D = 2$ and more than 40 basis functions per dimension (see Figures B.3 right and C.3 in the online supplement).

5 Case studies

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

5.1 Simulated data for a univariate function

In this experiment, we analyze a synthetic dataset with $n = 250$ observations, where the true data generating process is a Gaussian process with additive noise. The data points are simulated from the model $y_i = f(x_i) + \epsilon_i$, where f is a sample from a Gaussian process $f(x) \sim \mathcal{GP}(0, k(x, x', \theta))$ with a Matérn (ν=3/2) covariance function k with marginal variance $\alpha = 1$ and length-scale $\ell = 0.2$ at inputs values $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ with $x_i \in [-1, 1]$. ϵ_i is additive Gaussian noise with standard deviation $\sigma = 0.2$.

In the HSGP model, the latent function values $f(x)$ are approximated as in eq. (8), with the Matérn (ν=3/2) spectral density S as in eq. (2), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (6) and (7), respectively.

The joint posterior parameter distributions are estimated by sampling using the dynamic HMC algorithm implemented in Stan (Stan Development Team, 2021). Normal$(0, 1)$, Normal$(0, 3)$ and Gamma$(1.2, 0.2)$ prior distributions has been used for the observation noise σ, covariance function

iter.	ℓ	c	m	$\hat{\ell}$	$\hat{\ell} + 0.01 \geq \ell$	RMSE	RMSE*
1	0.50	2.25	16	0.12	FALSE	0.26	0.17
2	0.12	1.20	35	0.32	TRUE	0.18	0.01
3	0.02	1.20	40	0.28	TRUE	0.18	0.01

Table 2: Diagnostic on various data sets with varying smoothness. ℓ refers to the minimum valid length-scale that can be accurately estimated determined by m and c, except for the first iteration which is the initial guess of the length-scale. RMSE* refers to the root mean squared error computed with the exact GP model as the reference.

marginal variance α, and length-scale ℓ, respectively. We use the same prior distributions to fit the exact GP model.

The HSGP model is fitted following the recommended iterative steps as in Section 4.5.1. A initial value for the minimum lengthscale ℓ to use at the first iteration is guessed to be 0.5. While diagnostic $\hat{\ell} + 0.01 \geq \ell$ is *false*, c and m are updated by equations (23) and (22), respectively, and the minimum ℓ is updated with the estimated $\hat{\ell}$. After the diagnostic generated the first *true*, m is updated by increasing the m of the previous iteration by 5 additional basis functions, c is updated by eq. (23) as a function of the estimated $\hat{\ell}$ at previous iteration, and the minimum ℓ is set by eq. (22) as function of c and m. Table 2 contains the values for the parameters ℓ, c, m, the estimated $\hat{\ell}$, the diagnostic $\hat{\ell} + 0.01 \geq \ell$ and the RMSE compute with both the data and the GP as the reference for every iterative steps of the fitting process.

Figure 9 shows the posteriors predictive distributions of the exact GP and the HSGP models, the later using the parameter values as the third iterative step, $c = 1.2$ ($L = c \cdot 1 = 1.2$; see eq. (16)) and $m = 40$ basis functions. In order to do model comparison a spline-based model with 60 knots is also fitted and plotted jointly with the GP and HSGP models. The spline-based model is fitted using the thin plate regression spline approach by Wood (2003) and implemented in the R-package *mgcv* (Wood, 2011). A Bayesian approach is used to fit this spline model using the R-package *brms* (Bürkner, 2017). The sample observations are plotted as circles and the out-of-sample or test data are plotted as crosses. The posteriors of the three models, exact GP, HSGP and spline, are similar in the interpolation regions of the input space. However, when extrapolating the spline model solution clearly differs from the exact GP and HSGP models as well as the actual observations.

Figure 10 shows both the standardized root mean squared error (SRMSE) of the models for the sample data and the 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. Both the HSGP and splines models show roughly similar performance at sample data. The HSGP method yields a good approximation of the exact GP model for both interpolation and extrapolation. However, the spline model does not extrapolate data properly (Figure 9). The HSGP model is on average
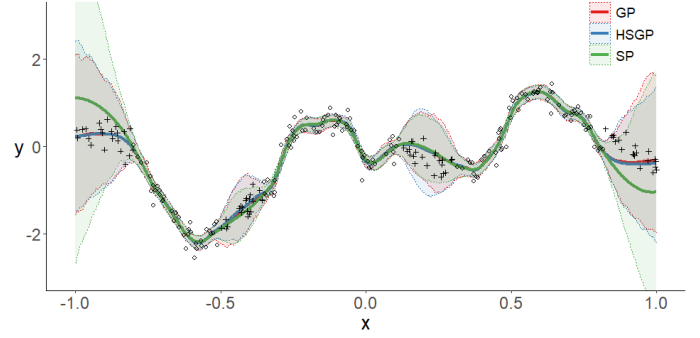

Figure 9: Posterior predictive means of the proposed HSGP model, the exact GP model, and the spline model (SP). 95% credible intervals are plotted as dashed lines.

roughly 400 times faster than the exact GP and 10 times faster than the spline model for this particular model and data. Also, it is seen that the computation time increases slowly as a function of the number of basis functions.
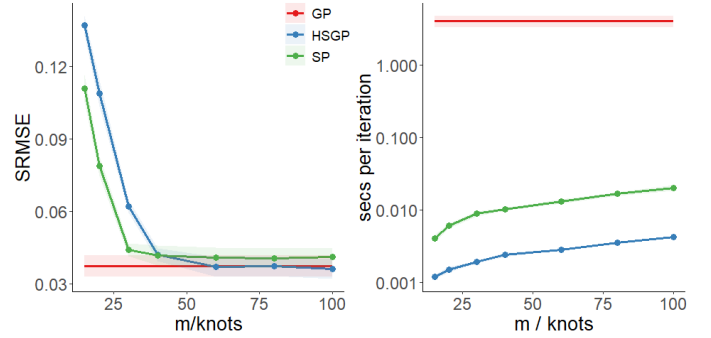

Figure 10: Computational time (y-axis), in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the spline model. The y-axis is on a logarithmic scale. The standard deviation of the computational time is plotted as dashed lines.

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

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

15

a birth for health reasons). Gelman et al. (2013) presented an analysis using exact GP and maximum a posteriori inference. As the total number of days within the period is $T = 7305$ ($t = 1, 2, \ldots, T$), a full Bayesian inference with MCMC for a exact GP model is memory and time consuming. We will use the HSGP method as well as the low-rank GP model with a periodic covariance function described in Appendix B which is based on expanding the periodic covariance function into a series of stochastic resonators (Solin and Särkkä, 2014).

Let y_t denote the number of births on the t'th day. The observation model is a normal distribution with mean function $\mu(t)$ and noise variance σ^2,

$$y_t \sim \mathrm{Normal}(\mu(t), \sigma^2).$$

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

$$\mu(t) = f_1(t) + f_2(t) + f_3(t) + f_4(t). \tag{24}$$

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

$$f_1(t) \sim \mathcal{GP}(0, k_1), \quad k_1(t, t') = \alpha_1 \exp\left(-\frac{1}{2}\frac{(t - t')^2}{\ell_1^2}\right),$$

which means the function values $\boldsymbol{f}_1 = \{f_1(t)\}_{t=1}^T$ are multi-variate Gaussian distributed with covariance matrix $\boldsymbol{K}_1$, where $K_{1_{t,s}} = k_1(t, s)$, with $t, s = 1, \ldots, T$. α_1 and ℓ_1 represent the marginal variance and length-scale, respectively, of this GP prior component. 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,

$$f_2(t) \sim \mathcal{GP}(0, k_2),$$
$$k_2(t, t') = \alpha_2 \exp\left(-\frac{2 \sin^2(\pi(t - t')/365.25)}{\ell_2^2}\right).$$

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,

$$f_3(t) \sim \mathcal{GP}(0, k_3),$$
$$k_3(t, t') = \alpha_3 \exp\left(-\frac{2 \sin^2(\pi(t - t')/7)}{\ell_3^2}\right).$$

The component $f_4(t)$ represents the special days effects, modeled as a Student's t prior model with 1 degree of freedom and variance τ^2:

$$f_4(t) \sim t(1, 0, \tau^2).$$

The component f_1 will be approximated using the HSGP model and the function values f_1 are approximated as in eq. (8), with the squared exponential spectral density S as in eq. (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (6) and

(7). The year effects f_2 and week effects f_3 use a periodic covariance function and thus do no fit under the main framework of the HSGP approximation covered in this paper. 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 and f_3 are approximated as in eq. (B.8), with variance coefficients $\tilde{q}_j^2$ as in eq. (B.5).

The joint posterior parameter distributions are estimated by sampling using the dynamic HMC algorithm implemented in Stan (Stan Development Team, 2021). Normal$(0, 1)$, Normal$(0, 10)$ and Normal$(0, 2)$ prior distributions has been used for the observation noise σ, covariance function marginal variances $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3\}$, and length-scales $\boldsymbol{\ell} = \{\ell_1, \ell_2, \ell_3\}$, respectively. A Normal$(0, 0.1)$ prior distribution has been used for the standard deviation τ of the Student's t distribution with 1 degree of freedom used to model f_4 (i.e., the special days effects).

The HSGP model is fitted following the recommended iterative steps as in Section 4.5.1, where in each iteration the diagnosis is applied on f_1, f_2 and f_3, where for each one these functions the parameters c, m, minimum ℓ, estimated $\hat{\ell}$ and diagnostic $\hat{\ell} + 0.01 \geq \ell$ are updated. For functions f_2 and f_3 there are not boundary factor c as they use periodic covariance functions, and m and minimum ℓ are updated by eq. (B.6). A initial value for the minimum lengthscale ℓ_1 of f_1 is guessed to correspond to around 3 years (i.e., $\ell_1 = 3$ years $= 3 \cdot 365 = 1095$ days) of the input dimension. Initial values for the minimum lengthscales ℓ_2 and ℓ_3 of f_2 and f_3, respectively, are guessed to correspond to half of the period (i.e., $l_2 = l_3 = 0.5$). After the diagnostic generated the first *true*, m is updated by increasing the m of the previous iteration by 5 additional basis functions and c and the minimum ℓ updated accordingly as explained in Section 4.5.1. The full diagnosis process is applied until two *trues* are achieved for each function. Table 3 contains the values for the parameters ℓ, c, m, the estimated $\hat{\ell}$, the diagnostic $\hat{\ell} + 0.01 \geq \ell$ for each function, and the RMSE compute with the data as the reference for every iterative steps of the fitting process.

Figure 11 shows the posterior means of the long-term trend f_1 and yearly pattern f_2 for the whole period, jointly with the observed data. Figure 12 shows the model for one year (1972) only. In this figure, the special days effects f_4 in the year can be clearly represented. The posterior means of the the function μ and the components f_1 (long-term trend) and f_2 (year pattern) are also plotted in this Figure 12. Figure 13 show the process in the month of January of 1972 only, where the week pattern f_3 can be clearly represented. The mean of the the function μ and components f_1 (long-term trend), f_2 (year pattern) and f_4 (special-days effects) are also plotted in this Figure 13.

R code and the Stan model code for the HSGP model of this case study can be found online at `https://github.com/gabriuma/basis_` `functions_approach_to_GP/tree/master/` `Paper/Case-study_Birthday-data.`

iter.	f	ℓ	c	m	$\hat{\ell}$	$\hat{\ell}+0.01 \geq \ell$	RMSE	ELPD
	f_1	0.52	1.20	7	0.53	FALSE		
1	f_2	0.50	-	8	0.34	FALSE	0.32	0.05
	f_3	0.50	-	8	1.62	TRUE		
	f_1	0.50	1.20	8	0.54	TRUE		
2	f_2	0.34	-	11	0.29	FALSE	0.32	0.06
	f_3	0.29	-	13	1.53	TRUE		
	f_1	0.28	1.20	13	0.33	TRUE		
3	f_2	0.29	-	13	0.24	FALSE	0.30	0.12
	f_3	0.29	-	13	1.64	TRUE		
	f_1	0.28	1.20	18	0.23	TRUE		
4	f_2	0.24	-	16	0.23	TRUE	0.30	0.15
	f_3	0.29	-	13	1.68	TRUE		
	f_1	0.16	1.20	23	0.20	TRUE		
5	f_2	0.24	-	16	0.23	TRUE	0.29	0.18
	f_3	0.29	-	13	1.63	TRUE		

Table 3: Iterative steps in HSGP model fitting of *Birthday data* case study. The diagnosis procedure updating the parameters minimum ℓ, c, m, estimated $\hat{\ell}$ and diagnostic $\hat{\ell}+0.01 \geq \ell$ is applied on the three underlying functions f_1, f_2 and f_3. Notice that f_2 and f_3 do not have boundary factor c as they use periodic covariance functions. RMSE and ELPD are evaluated for the model underlying function $\mu = f_1 + f_2 + f_3 + f_4$ and sample data. The full diagnosis process is applied until two *trues* are achieved for each function.
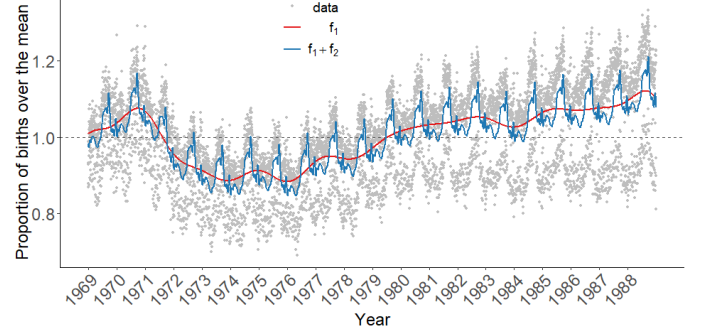

Figure 11: Posterior means of the long-term trend f_1 and year effects pattern (f_2) for the whole series.
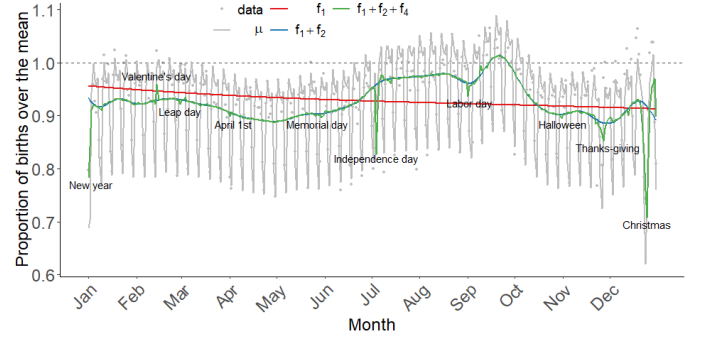

Figure 12: Posterior means of the function μ for the year 1972 of the series. The special days effects pattern f_4 in the year is also represented, as well as the long-term trend f_1 and year effects pattern f_2.

5.3 Case study: Simulated data for 2D and 3D functions

In this case study, we apply the diagnostic tool to fit and diagnose two different data sets, one data set simulated from a bivariate (D=2) function and another data set simulated from a 3-dimensional (D=3) function. Furthermore, in Section 5.3.1 we present results of the computational time required to fit the HSGP model in 2D, 3D and 4D input spaces, with different sized data sets and as a function of the number of multivariate basis functions $m^* = m_1 \times \cdots \times m_D$ used in the approximation.

2D and 3D synthetic functions were drawn from 2D and 3D GP priors, with input values $\boldsymbol{x}_i \in [-1, 1]^2$ and $\boldsymbol{x}_i \in [-1, 1]^3$, respectively. Squared exponential covariance functions with marginal variance $\alpha = 1$ and length-scales $\ell_1 = 0.10$, $\ell_2 = 0.3$, and $\ell_3 = 0.4$, where ℓ_i is the length scale for the i'th dimension, were used in the GP priors. 200 and 1000 data points were sampled from the 2D and 3D drawn functions, respectively, and independent Gaussian noise with standard deviation $\sigma = 0.2$ was added to the data points to form the final noisy sets of observations.

In the HSGP models with 2 and 3 input dimensions, the underlying functions $f(\boldsymbol{x})$ are approximated as in eq. (13), with the D-dimensional squared exponential spectral densities S as in eq. (1), and the D-vectors of eigenvalues $\boldsymbol{\lambda}_j$ and the multivariate eigenfunctions ϕ_j as in equations (12) and (11), respectively.

The joint posterior parameter distributions are estimated via the dynamic HMC sampling algorithm implemented in Stan (Stan Development Team, 2021). $\mathrm{Normal}(0, 1)$, $\mathrm{Normal}(0, 3)$, and $\mathrm{InverseGamma}(2, 0.5)$ priors were used for the observation noise σ, marginal variance α, and length-scales ℓ, respectively. We used the same priors to fit the exact GP model serving as benchmark.

The HSGP models are fitted following the recommended iterative procedure detailed in Section 4.5.1, where the diagnostic is applied on every dimension, separately. For each dimension, the parameters c, m, minimum ℓ, estimated $\hat{\ell}$, and the diagnostic $\hat{\ell}+0.01 \geq \ell$ are updated using the equations in Section 4.3.1. The values 0.5, 1, and 1 are set as initial values for the minimum lengthscales ℓ_1, ℓ_2 and ℓ_3, respectively. In order to be as efficient as possible, after the diagnostic generated the first *true* for a certain dimension, its corresponding m is updated by increasing the m of the previous iteration by only 2 additional basis functions and, after the second *true*, m is no longer increased. The full diagnostic process is applied until two *trues* are achieved for each dimension.
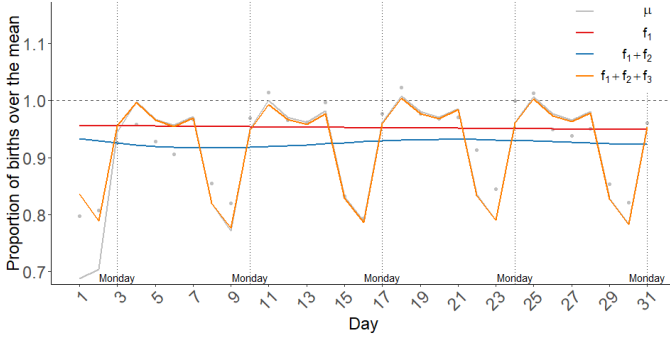
Figure 13: Posterior means of the function μ for the month of January of 1972. The week effects pattern f_3 in the month is also represented, as well as the long-term trend f_1, year effects pattern f_2 and special days effects pattern f_4.

Table 4 and 5 contain the iterative steps to fit and diagnose the 2D and 3D data sets, respectively. The minimum requirements to fit the models were easily achieved by performing 4 iterations: from a 1st iteration that uses few multivariate basis functions m^* ($m^* = m_1(= 6) \times m_2(= 6) = 36$ and $m^* = m_1(= 6) \times m_2(= 6) \times m_3(= 6) = 216$ for the 2D and 3D data sets, respectively) to the 4th iteration that uses the minimum multivariate basis functions ($m^* = m_1(= 22) \times m_2(= 11) = 242$ and $m^* = m_1(= 21) \times m_2(= 10) \times m_3(= 12) = 2520$ for the 2D and 3D data sets, respectively) required to accurately approximate these notably wiggly functions. Without using our recommended guidelines and diagnostics, it would likely have taken the user several more trials to get the optimal solution and to spend a significant amount of time, as using a larger number of basis functions in multivariate cases increases time of computation drastically.

iter.	Dim.	ℓ	c	m	$\hat{\ell}$	$\hat{\ell} + 0.01 \geq \ell$	RMSE	RMSE*
1	1	0.50	1.6	6	0.19	FALSE	0.75	0.72
	2	1.00	3.2	6	0.25	FALSE		
2	1	0.19	1.2	12	0.11	FALSE	0.18	0.12
	2	0.25	1.2	9	0.25	TRUE		
3	1	0.11	1.2	20	0.11	TRUE	0.11	0.03
	2	0.19	1.2	11	0.27	TRUE		
4	1	0.10	1.2	22	0.11	TRUE	0.10	0.02
	2	0.19	1.2	11	0.27	TRUE		

Exact GP length-scales: $\ell_{1_{GP}} = 0.10$, $\ell_{2_{GP}} = 0.29$

Table 4: Iterative steps in HSGP model fitting on the 2D data. The diagnostic procedure updating the parameters minimum ℓ, c, m, estimated $\hat{\ell}$, and diagnostic $\hat{\ell} + 0.01 \geq \ell$ is applied on the two input dimensions. The full diagnostic process is applied until two consecutive *trues* in the diagnostic are achieved for each dimension. RMSE and RMSE* refer to the root mean squared error of the HSGP computed against the true function and the exact GP, respectively.

iter.	Dim.	ℓ	c	m	$\hat{\ell}$	$\hat{\ell} + 0.01 \geq \ell$	RMSE	RMSE*
1	1	0.50	1.6	6	0.14	FALSE		
	2	0.50	1.6	6	0.27	FALSE	0.63	0.58
	3	1.00	3.2	6	0.23	FALSE		
2	1	0.13	1.2	16	0.11	FALSE		
	2	0.27	1.2	8	0.29	TRUE	0.13	0.07
	3	0.23	1.2	19	0.40	TRUE		
3	1	0.11	1.2	19	0.12	TRUE		
	2	0.21	1.2	10	0.29	TRUE	0.11	0.04
	3	0.19	1.3	12	0.43	TRUE		
4	1	0.10	1.2	21	0.12	TRUE		
	2	0.21	1.2	10	0.29	TRUE	0.11	0.03
	3	0.19	1.3	12	0.43	TRUE		

Exact GP length-scales: $\ell_{1_{GP}} = 0.11$, $\ell_{2_{GP}} = 0.32$, $\ell_{3_{GP}} = 0.43$

Table 5: Iterative steps in HSGP model fitting on the 3D data. The diagnostic procedure updating the parameters minimum ℓ, c, m, estimated $\hat{\ell}$ and diagnostic $\hat{\ell} + 0.01 \geq \ell$ is applied on the three input dimensions. The full diagnostic process is applied until two consecutive *trues* in the diagnostic are achieved for each dimension. RMSE and RMSE* refer to the root mean squared error of the HSGP computed against the true function and the exact GP, respectively.

Figure 14 shows the difference between the true underlying data-generating function and the posterior mean for GP, HSGP and spline-based model solutions for the 2D data set. For the 3D data set plots with the posterior functions are not shown because it is difficult to plot functions in a 3D input space. The splines-based model was fitted using a cubic spline basis, penalized by the conventional integrated square second derivative cubic spline penalty (Wood, 2017), and implemented in the R-package *mgcv* (Wood, 2011). A Bayesian approach is used to fit this spline-based model using the R-package *brms* (Bürkner, 2017). For the spline-based model, 40 knots in each dimension were used. Figure 14-left shows the root mean squared error (RMSE), computed against the data-generating function, as a function of the boundary factor c and number of univariate basis functions m, for the HSGP, and knots, for the 2D spline-based model. It can be seen that the posterior mean for HSGP and exact GP accurately represents the true data-generating function. In contrast, the accuracy of the spline-based model is significantly worse.

R code and the Stan model codes for the exact GP, the HSGP and the spline-based models of this case study can be found online at https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_2D&3D-Simulated-data.
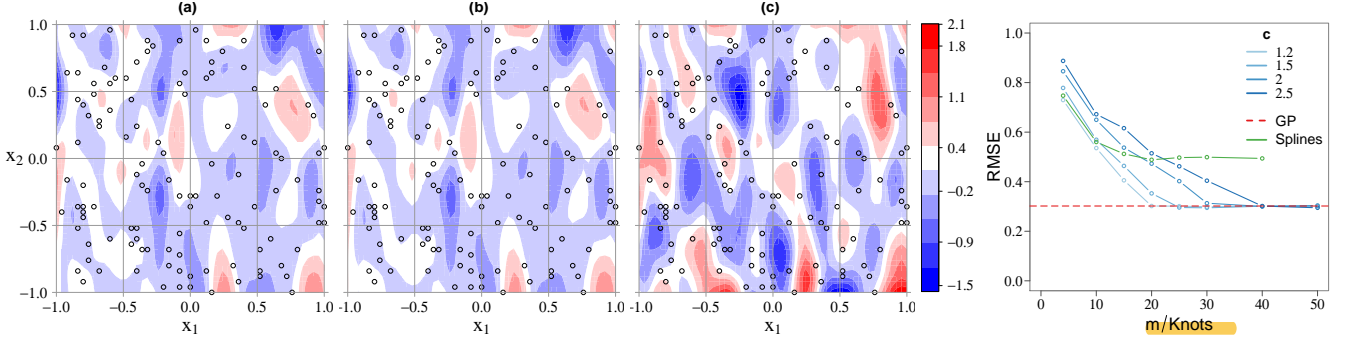
18

Figure 14: Error between the 2D data-generating function and posterior mean functions of the GP (a), HSGP (b) and spline-based (c) models. Sample points are plotted as circles. The right side plot shows the root mean square error (RMSE) of the different methods, and plotted as a function of the boundary factor c, number of univariate basis functions m and knots (the same number of basis functions m and knots per dimension are used, resulting in m^D total number of multivariate basis functions).

5.3.1 Computation requirements in 2D, 3D, and 4D input spaces

Figure 15 shows the computation time for dynamic HMC (in seconds per iteration) in 2D, 3D and 4D input spaces and different sized data sets, $n = 300, n = 500, n = 1000$, and $n = 3000$, as a function of the number of multivariate basis functions $m^* = m_1 \times \cdots \times m_D$ used in the approximation. Both the time of computation and m^* are plotted in the logarithmic scale.

Looking back at Figure 6, or equations (18), (22), and (20), and assuming squared exponential covariance function, any univariate function (or single dimensions of a multivariate function) with true lengthscales bigger than 0.3 can be accurately fitted using only $10 - -12$ basis functions. For lengthscales between 0.1 and 0.3, $10 - -22$ basis functions are sufficient.

For a very wiggly 2D function (say, with $\ell_1 = 0.1$ and $\ell_2 = 0.1$), the approximate number of multivariate basis functions needed is $m^* = 22 \times 22 = 484$, which results in significantly faster computation than the exact GP, even with small data sets (i.e., $n \lesssim 300$) (see Figure 15). For a very wiggly 3D function (say, with $\ell_1 = 0.1$, $\ell_2 = 0.1$ and $\ell_3 = 0.1$), the approximate number of multivariate basis functions needed is $m^* = 22 \times 22 \times 22 = 10648$, which, unless for small data sets (i.e., $n \lesssim 1000$), the method is still significantly faster than the regular GP. A 3D function where each dimension has a lengthscale around 0.1 is not that common in statistical analysis, and thus in many cases the approximation will be significantly faster than excat GP. For 4D data sets, the method can still be more efficient than the exact GP for moderate-to-large data sets (i.e., $n \lesssim 1000$).

Finally, for $D > 5$ the method starts to be impractical even for smooth univariate functions. However, in these cases, the method may still be used for lower dimensional componenets in an additive modeling scheme.

5.4 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 and censoring 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 diagnostic with 1 unit = $50 \times 109/L$, 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.

As the WBC predictor values were strictly positive and highly skewed, a logarithm transformation is used. 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:

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

We do not have a full observation model, as we do not have a model for the censoring process. We use the complementary cumulative log-normal probability distribution for the censored data conditionally on the censoring time t_i:

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

where $y_i > t_i$ denotes the unobserved survival 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. Due to the predictor *sex* (x_2) being a categorical variable ('1' for female and '2' for male), we
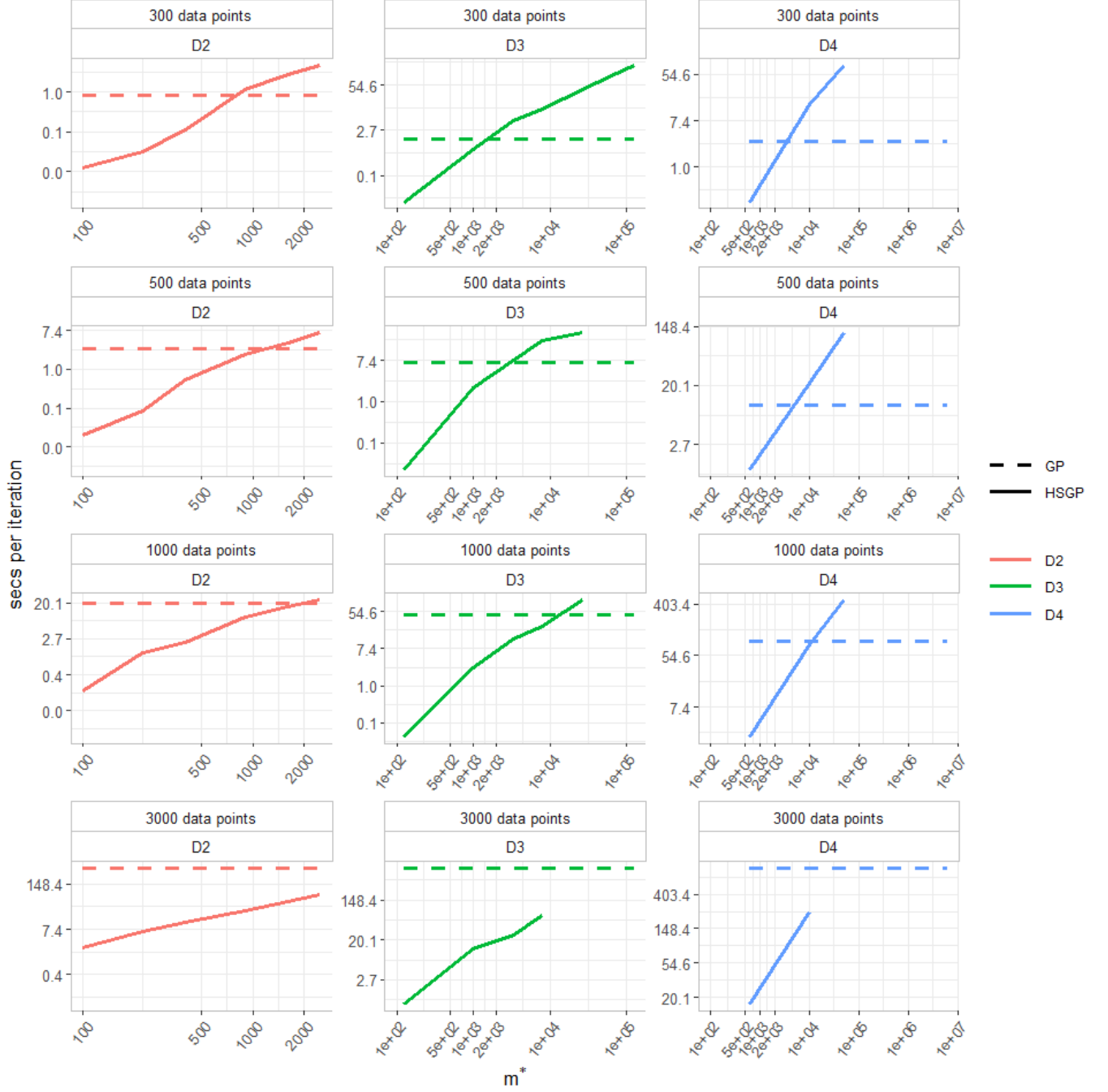
Figure 15: Computational time in seconds per iteration (iteration of the HMC sampling method) for different datasets with different dimensionalities and data points, and plotted as a function of the number of multivariate basis functions $m^* = m_1 \times \cdots \times m_D$.

apply indicator variable coding for the GP functions, in a similar way such coding is applied in linear models (Gelman et al., 2020). The latent function $f(\boldsymbol{x})$, besides of being centered around a linear model, is composed of a general mean GP function, $h(\boldsymbol{x})$, defined for all observations, plus a second GP function, $g(\boldsymbol{x})$, that only applies to one of the predictor levels ('male' in this case) and is set to zero otherwise:

$$h(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{0},\, k(\boldsymbol{x}, \boldsymbol{x}', \theta_0)\right),$$
$$g(\boldsymbol{x}) \sim \mathcal{GP}\left(\mathbf{0},\, k(\boldsymbol{x}, \boldsymbol{x}', \theta_1)\right),$$
$$f(\boldsymbol{x}) = c + \boldsymbol{\beta}\boldsymbol{x} + h(\boldsymbol{x}) + \mathbb{I}[x_2 = 2]\, g(\boldsymbol{x}),$$

where $\mathbb{I}[\cdot]$ is an indicator function. Above, c and $\boldsymbol{\beta}$ are the intercept and vector of coefficients, respectively, of the linear model. θ_0 contains the hyperparameters α_0 and ℓ_0 which are the marginal variance and length-scale of the general mean GP function, and θ_1 contains the hyperparameters α_1 and ℓ_1 which are the marginal variance and length-scale, respectively, of a GP function specific to the male sex ($x_2 = 2$). Scalar length-scales, ℓ_0 and ℓ_1, are used in both multivariate covariance functions, assuming isotropic functions.

Using the HSGP approximation, the functions $h(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are approximated as in eq. (14), with the D-dimensional (with a scalar length-scale) squared exponential spectral density S as in eq. (1), and the multivariate eigenfunctions ϕ_j and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ as in equations (12) and (11), respectively.

Figure 16 shows estimated conditional functions 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. There are clear non-linear patterns and the right bottom subplot also shows that the conditional function associated with WBC has an interaction with TDI. Figure 17 shows the expected log predictive density (ELPD; Vehtari and Ojanen, 2012; Vehtari et al., 2017) and time of computation as function of the number of univariate basis functions m ($m^* = m^D$ in eq. (14)) 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 17-left); Small boundary factors are not appropriate for models for large length-scales, as can be seen in Figure 6. Increasing the boundary factor also significantly increases the time of computation (Figure 17-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.

R code and the Stan model code for the exact GP and the HSGP models of this case study can be found online 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 in many applications. This problem becomes especially severe when performing full Bayesian inference using sampling-based methods. In this paper, a recent approach for a low-rank representation of stationary GPs, originally proposed by Solin and Särkkä (2020), has been 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 an exact 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.

As one of the main contributions of this paper, we provided an in-depth analysis of the approximation's performance and accuracy 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. On that basis, as our second main contribution, we developed practical diagnostics to assess the approximation's performance as well as an iterative procedure to obtain an accurate approximation with minimal computational costs.

The developed approximate GPs can be easily applied as modular components in probabilistic programming frameworks such as Stan in both Gaussian and non-Gaussian observation models. Using several simulated and real datasets, we have demonstrated the practical applicability and improved sampling efficiency, as compared to exact 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 Figure 6, is essential to avoid a excessive computational time especially in multivariate input spaces. However, in practice, input dimensionalities larger than three start to be computationally demanding even for moderately wiggly functions and few basis functions per input dimension. In these high dimensional cases, the proposed approximate GP methods may still be used for low-dimensional components in an additive modeling scheme but without modeling very high dimensional interactions, as complexity is linear with the number of additive components.

In this paper, the obtained functional relationships between the key factors influencing the approximation and corresponding diagnostics were studied primarily for univariate inputs. Accordingly, investigating the functional relationships more thoroughly for multivariate inputs remains a topic for future research.
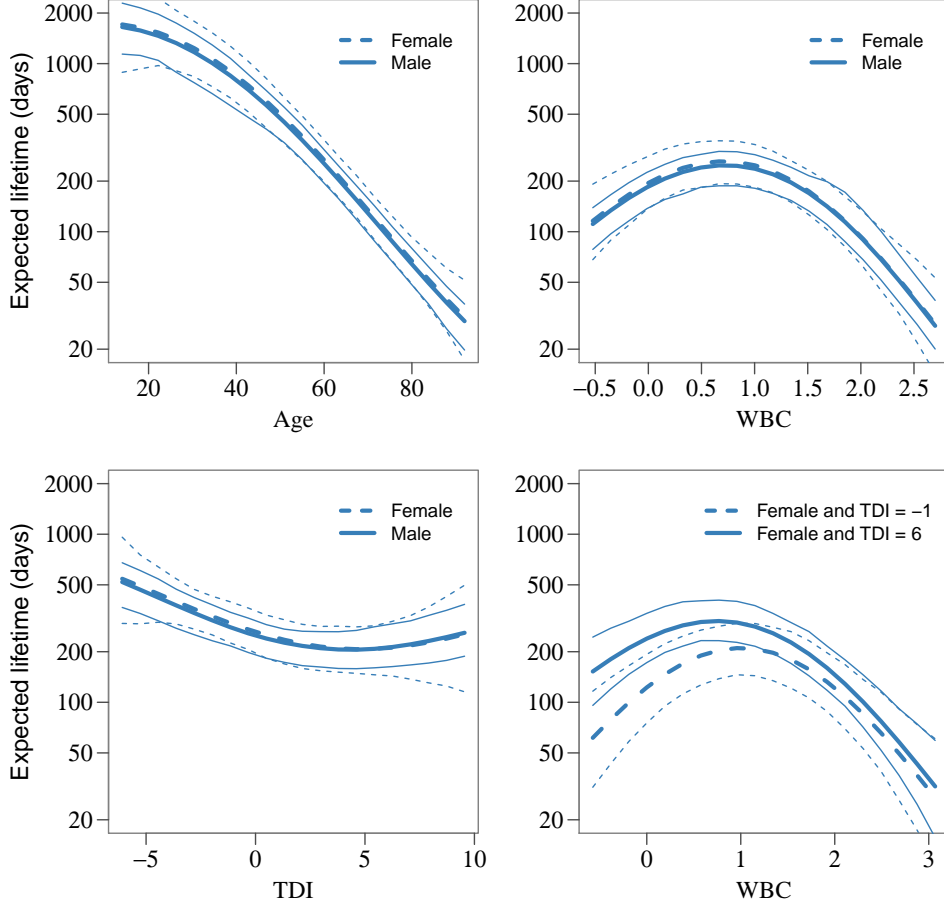
Figure 16: Expected lifetime conditional comparison for each predictor with other predictors fixed to their mean values. The thick line in each graph is the posterior mean estimated using a HSGP model, and the thin lines represent pointwise 95% credible intervals.

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 based on the work by Solin and Särkkä (2020), who developed 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 theorems, 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:

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

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

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

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

$$l(\boldsymbol{x}, \boldsymbol{x}') = \sum_j \lambda_j \phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'),$$
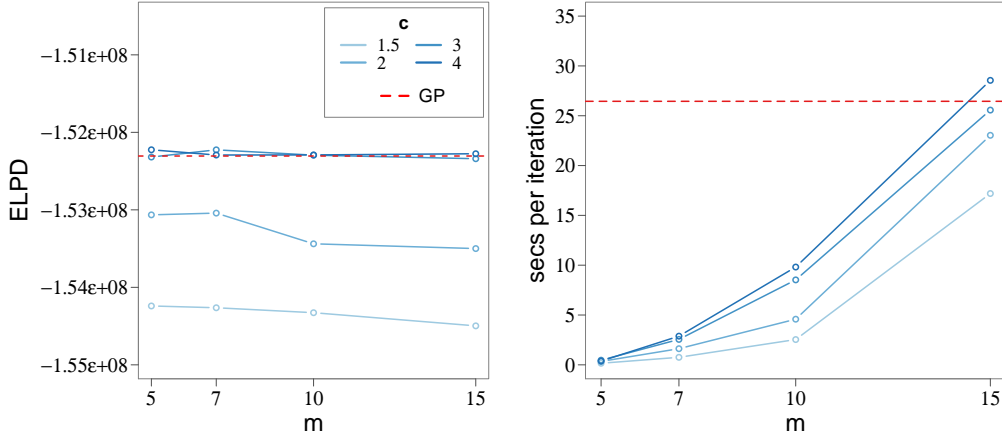
Figure 17: Expected log predictive density (ELPD; left) and time of computation in seconds per iteration (iteration of the HMC sampling method; right) as a function of the number of basis functions m and boundary factor c.

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 (other boundary conditions could be used as well):

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

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

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

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

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

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

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

By letting $||\boldsymbol{w}||^2 = \lambda_j$ the spectral density in eq. (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 eq. (A.4) then leads to the final form

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

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, but it has also a low-rank representation. In this section, we first give a brief presentation of the results by Solin and Särkkä (2014), who 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 squared exponential covariance function.

The periodic squared exponential covariance function takes the form

$$k(\tau) = \alpha \exp\left(-\frac{2\sin^2(\omega_0 \frac{\tau}{2})}{\ell^2}\right), \qquad (B.1)$$

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

Solin and Särkkä (2014) derive a cosine series expansion for the periodic covariance function (B.1) as follows,

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

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

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

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

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

The covariance in eq. (B.2) is a Jth order truncation of a Taylor series representation. This approximation converges to eq. (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:

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

for $j = 1, 2, \ldots, J$, and where the $\mathrm{I}_j(z)$ is the modified Bessel function (Abramowitz and 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. A more detailed explanation and mathematical proofs of this approximation of a periodic covariance function are provided by Solin and 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 length-scale of the kernel. We have considered an approximation to be close enough in terms of satisfying eq. (15) with $\varepsilon = 0.005 \int k(\tau) \, \mathrm{d}\tau$ (0.5% of the total area under the curve of the exact covariance function k). Since this is a series expansion of sinusoidal functions, the approximation does not depend on any boundary condition.

From the empirical observations, a numerical equation governing the relationships between J and ℓ were estimated in equation B.6, which show linear proportionality between J and ℓ:

$$J \geq \frac{3.72}{\ell} \quad \Leftrightarrow \quad \ell \geq \frac{3.72}{J}. \qquad (B.6)$$

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 eq. (B.2) can be written as

$$\begin{aligned} k(x,x') \approx \alpha \bigg(&\sum_{j=0}^{J} \tilde{q}_j^2 \cos(j\omega_0 x)\cos(j\omega_0 x') \\ &+ \sum_{j=1}^{J} \tilde{q}_j^2 \sin(j\omega_0 x)\sin(j\omega_0 x') \bigg). \end{aligned} \qquad (B.7)$$

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

$$\begin{aligned} f(x) \approx \alpha^{1/2} \bigg(&\sum_{j=0}^{J} \tilde{q}_j \cos(j\omega_0 x)\beta_j \\ &+ \sum_{j=1}^{J} \tilde{q}_j \sin(j\omega_0 x)\beta_{J+1+j} \bigg), \end{aligned} \qquad (B.8)$$

where $\beta_j \sim \mathrm{Normal}(0, 1)$, 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 $O(n(2J + 1) + (2J + 1))$, where n is the number of observations and J the number of cosine terms. The parameterization in eq. (B.8) is naturally in the non-centered form with independent prior distributions on β_j, which makes posterior inference easier.

C Description of the iterative steps applied to perform diagnostic on two of the data sets in Figure 1

As concrete examples, the iterative steps applied to perform diagnostic on two of the data sets in Table 1 are described next:

Data set with the squared exponential kernel and true length-scale $\ell_{\mathbf{GP}} = 0.08$:

Iteration 1:

1. Make the initial guess that the length-scale $\ell_1 = 0.5$.

2. Compute the optimal c_1 as a function of ℓ_1 by using eq. (19).

   TODO: name the value of c_1 and all the other quantities in the following. Otherwise, what is the purpose of this example?

3. Compute the minimum m_1 as a function of c_1 and ℓ_1 by using eq. (18).
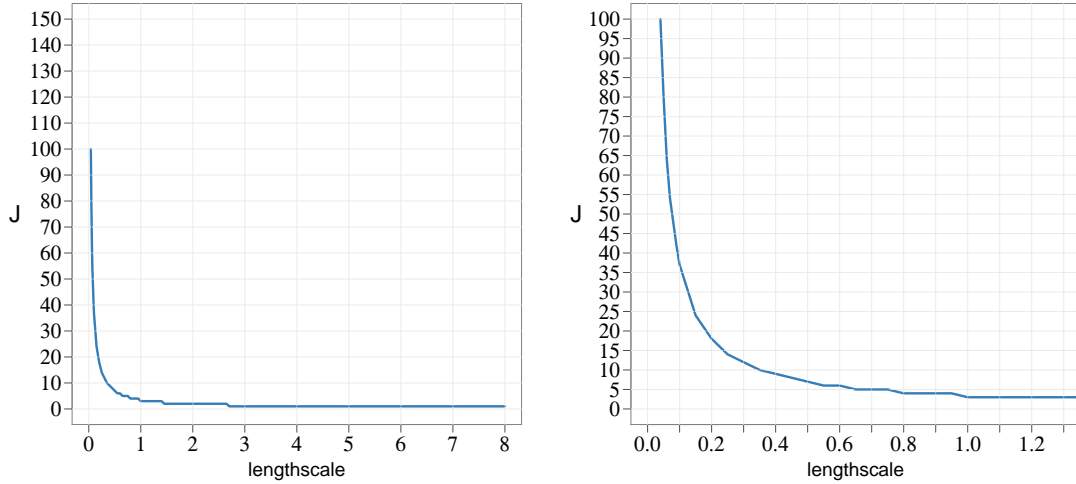
Figure B.1: Relation among the minimum number of terms J in the approximation and the length-scale (ℓ) of the periodic squared exponential covariance function. The right-side plot is a zoom in of the left-side plot.

4. Fit the HSGP model and obtain the estimated $\hat{\ell}_1$.

    TODO: provide at least a mean value for the posterior of $\hat{\ell}_1$ here?

5. The diagnostic $\hat{\ell}_1 + 0.01 \geq \ell_1$ gives FALSE.

    TODO: I think you wanted to use $\hat{\ell}_1 - 0.01 \geq \ell_1$ and I corrected it above. Can you verify whether you want + or -?

Iteration 2:

1. As the diagnostic in iteration 1 was FALSE, set $\ell_2 = \hat{\ell}_1$.

2. Compute the optimal c_2 as a function of ℓ_2 and m_1 as a function of c_2 and ℓ_2 by using equations (19) and eq. (18), respectively.

    TODO: why make this a single step here when it was two steps in Section 4.4.1?

3. Fit the HSGP model using c_2 and m_2 and estimating $\hat{\ell}_2$.

4. The diagnostic $\hat{\ell}_2 + 0.01 \geq \ell_2$ gives FALSE.

Iteration 3:

1. As the diagnostic in previous iteration 2 was FALSE, updating ℓ_3 with $\hat{\ell}_2$ and repeat the process as in iteration 2.

2. The diagnostic $\hat{\ell}_3 + 0.01 \geq \ell_3$ gives TRUE.

Iteration 4:

1. As the diagnostic in step 4 in previous iteration 3 was TRUE, set $m_4 = m_3 + 5$.

2. Compute the optimal c_4 as a function of ℓ_3 by using eq. (19).

3. Fit the HSGP model using c_4 and m_4 and estimating $\hat{\ell}_4$.

4. The diagnostic $\hat{\ell}_4 + 0.01 \geq \ell_4$ gives TRUE.

5. As RMSE, R^2, and ELPD are stable relative to previous iteration 4, the HSGP approximation is likely sufficiently accurate and the procedure ends here.

Data set with the squared exponential kernel and true length-scale $\ell_{\mathbf{GP}} = 1.4$:

Iteration 1:

1. Make the initial guess that the length-scale $\ell_1 = 1$.

2. Compute the optimal c_1 as a function of ℓ_1 by using eq. (19).

3. Compute the minimum m_1 as a function of c_1 and ℓ_1 by using eq. (18).

4. Fit the HSGP model and obtain the estimated $\hat{\ell}_1$.

5. The diagnostic $\hat{\ell}_1 + 0.01 \geq \ell_1$ gives TRUE.

Iteration 2:

1. As the diagnostic in previous iteration 1 was TRUE, set $m_2 = m_1 + 5$.

2. Compute the optimal c_2 as a function of ℓ_2 by using eq. (19).

3. Fit the HSGP model using c_2 and m_2 and estimating $\hat{\ell}_2$.

4. The diagnostic $\hat{\ell}_2 + 0.01 \geq \ell_2$ gives TRUE.

5. As RMSE, R^2, and ELPD are stable relative to previous iteration 2, the HSGP approximation is likely sufficiently accurate and the procedure ends here.

Acknowledgements

References

Milton Abramowitz and Irene Stegun. *Handbook of Mathematical Functions*. Dover Publishing, New York, 1970.

Robert J Adler. *The Geometry of Random Fields*. SIAM, 1981.

Naum I. Akhiezer and Izrail' M. Glazman. *Theory of Linear Operators in Hilbert Space*. Dover, New York, 1993.

Michael Riis Andersen, Aki Vehtari, Ole Winther, and Lars Kai Hansen. Bayesian inference for spatio-temporal spike-and-slab priors. *Journal of Machine Learning Research*, 18(139): 1–58, 2017.

Michael Betancourt and Mark Girolami. Hamiltonian Monte Carlo for hierarchical models. In *Current Trends in Bayesian Methodology with Applications*, pages 79–101. Chapman and Hall/CRC, 2019.

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

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

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

David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational Gaussian process regression. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 862–871. PMLR, 2019.

Bradley P Carlin, Alan E Gelfand, and Sudipto Banerjee. *Hierarchical Modeling and Analysis for Spatial Data*. Chapman and Hall/CRC, 2014.

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

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

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

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

Peter J Diggle. *Statistical Analysis of Spatial and Spatio-temporal Point Patterns*. Chapman and Hall/CRC, 2013.

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

Yarin Gal and Richard Turner. Improving the Gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 655–664. PMLR, 2015.

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

Andrew Gelman, Jennifer Hill, and Aki Vehtari. *Regression and Other Stories*. Cambridge University Press, 2020.

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

GPy. GPy: A Gaussian process framework in Python, 2012. URL http://github.com/SheffieldML/GPy.

Ulf Grenander. *Abstract Inference*. John Wiley & Sons, 1981.

Philipp Hennig, Michael A Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

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

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

Miguel Lázaro Gredilla. *Sparse Gaussian Processes for Large-scale Machine Learning*. PhD thesis, Universidad Carlos III de Madrid, 2010.

Finn Lindgren, David Bolin, and Håvard Rue. The spde approach for gaussian and non-gaussian fields: 10 years and still running. *Spatial Statistics*, page 100599, 2022.

Michel Loève. *Probability Theory*. Springer-Verlag, New York, 1977.

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

Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

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

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

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

R Core Team. R: A language and environment for statistical computing, 2019. http://www.R-project.org/.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1313–1320. Curran Associates, Inc., 2009.

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

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. MIT press, 2006.

Stephen J Roberts. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010.

Arno Solin and Simo Särkkä. Explicit link between periodic covariance functions and state space models. In Samuel Kaski and Jukka Corander, editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 904–912. PMLR, 2014.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020. Much of the work in this paper is based on the pre-print version predating the published paper. Pre-print available at https://arxiv.org/abs/1401.5508.

Simo Särkkä, Arno Solin, and Jouni Hartikainen. 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, 2013.

Stan Development Team. Stan modeling language users guide and reference manual, 2.28, 2021. URL https://mc-stan.org.

Harry. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory*. John Wiley & Sons, New York, NY, 1968.

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

Aki Vehtari and Janne Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, 2017.

Grace Wahba. *Spline Models for Observational Data*, volume 59. SIAM, 1990.

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

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

Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011.

Simon N Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2017.