Equispaced Fourier representations enable fast iterative Gaussian process regression

Alex Barnett[1], Philip Greengard[*2], and Manas Rachh[1]

[1]Center for Computational Mathematics, Flatiron Institute
[2]Department of Statistics, Columbia University

July 8, 2022

Contents

1 Introduction 2

2 Factorization of the kernel 4

3 Equispaced Fourier representation of GPs 6
3.1 Discretized inverse Fourier transform 7
3.2 Numerical Implementation 8

4 Error analysis 10
4.1 Discretization error 10
4.2 Accuracy of posterior mean 12
4.3 Conditioning of Gaussian process covariance matrices 14
4.4 Empirical condition number 15

5 Numerical experiments 16
5.1 EFGP time and accuracy for various N 16
5.2 Accuracy vs. time 17
5.3 Spatial Gaussian process regression with satellite data 18

6 Generalizations and conclusions 18

A Equivalence of weight-space and function-space 19

[*]pg2118@columbia.edu. Research supported by Alfred P. Sloan Foundation.

B Discretization error estimates for the Matérn kernel 21

Abstract

We introduce a class of Fourier-based fast algorithms for computing with Gaussian processes. Our approach relies on discretizing Gaussian processes via complex exponentials with equispaced frequencies. This discretization results in a weight-space linear system with a matrix that can be applied in $O(m \log m)$ operations where m is the number of frequencies, and can be solved efficiently with iterative methods. The efficient matrix-vector multiply of the weight-space linear system results in an efficient linear solve that is highly sensitive to discretizations that use large numbers of Fourier modes, enabling high performance for Gaussian processes in higher dimensions and kernels with fat-tailed spectral densities. We provide formulae for the error of discretizations, the condition number of the Gaussian process covariance matrix, convergence rates of iterative methods, and a general formula for the accuracy of the posterior mean at the data points for approximate methods. Numerical experiments are demonstrated for Gaussian processes over $\mathbb{R}^d$ for $d = 1, 2, 3$.

1 Introduction

Gaussian process (GP) regression is ubiquitous in machine learning and statistics (e.g. [3,5, 22,23,26]) due in large part to its generality and mathematical simplicity. In Gaussian process regression, the goal is to recover a function (or certain properties of a function) given noisy observations from that function in addition to some knowledge about the data-generating process. More precisely, suppose we are given N data points $\{(x_i, y_i) : x_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}$ with observations $y_1, ...y_N$ which are noisy samples from a Gaussian process distribution. That is,

$$y_i \sim f(x_i) + \epsilon_i \tag{1}$$

$$f(x) \sim \mathcal{GP}(m(x), k(x, y)) \tag{2}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed (iid) noise, σ^2 is the residual variance, $m : \mathbb{R}^d \to \mathbb{R}$ is the mean function of the Gaussian process, and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the covariance kernel. The likelihood function of the Gaussian process distribution is given by the equation

$$p(y) \propto \frac{1}{|K + \sigma^2 I|^{1/2}} e^{-\frac{1}{2} y^{\intercal} (K + \sigma^2 I)^{-1} y} \tag{3}$$

where K is the $N \times N$ matrix with $K_{i,j} = k(x_i, x_j)$, and I is the $N \times N$ identity matrix. The main limitation of Gaussian process regression as a practical applied tool is the computational cost of the inversion of $K + \sigma^2 I$. In particular, evaluation of the likelihood function p, as well as the evaluation of the mean and standard deviation of p all require the inversion of $K + \sigma^2 I$. For a general $N \times N$ matrix, direct inversion requires $O(N^3)$ operations, which for many modern data sets is prohibitively expensive.

A large literature has emerged over the last several decades devoted to computational methods for efficient inversion of K. These methods generally take advantage of structure that is particular to the covariance matrices that appear in Gaussian process problems.

For example, for low-dimensional Gaussian process regression problems, the off-diagonal blocks of K tend to be low-rank. This can be exploited with hierarchical decompositions that lead to fast inversion (e.g. [1, 18]). Other common methods involve subsampling rows and columns of K (Nyström methods), inducing point methods, or exploiting analytical structure in particular families of covariance kernels [10, 22, 30]. Yet another common class of techniques involves Fourier based methods (e.g. [15, 16, 20, 21]). [16] provides a review of several methods that use Fourier and FFT-based Gaussian process solvers.

The algorithms of this paper are in large part an extension of the methods of [15] to higher dimensions. In [15] a 1-dimensional Gaussian process is discretized with Fourier expansions, where frequencies are chosen numerically so as to provide accurate discretizations over families of Gaussian processes. The covariance matrix of the Gaussian process can then be inverted in either $O(Nm^2)$ time or $O(m^3 + (N + m)\log(N + m))$ via the weight-space linear system where N is the number of data points and m the number of Fourier modes.

In 1-dimension, the number of modes, m, needed to discretize a Gaussian process tends to be sufficiently small that $O(m^3)$ operations is easily affordable. Since the number of discretization nodes needed increases roughly as m^d for d-dimensional Gaussian processes, the cost of $O(m^{3d})$ can become prohibitive even for smooth kernels in 2 dimensions.

In this paper, we address the $O(m^{3d})$ linear solve in higher dimensions by first representing a Gaussian process as a Fourier expansion with equispaced frequencies and Gaussian coefficients. The equispaced frequencies serve two roles – first, they efficiently discretize a Gaussian process, especially when kernels are smooth, and second, they facilitate an efficient solution to linear systems that appear in Gaussian process problems. The fast linear solve stems from the fact that with equispaced nodes, the covariance matrix of the Gaussian process likelihood can be applied in $O(m \log m)$ operations after $O((N + m)\log N + m)$ precomputation. The linear system can thus be solved efficiently using iterative methods such as conjugate gradient.

The efficiency of iterative algorithms for solving linear systems can be severely limited by the conditioning of the matrix. To address this, we provide a tight upper bound on the condition number of a general Gaussian process covariance matrix along with the corresponding convergence rates for conjugate gradient. Specifically, the condition number of $K + \sigma^2 I$ is an $O(N)$ quantity. We demonstrate empirically that the upper bound is typically an accurate approximation, within about a factor of 2 of the empirical condition number for most data distributions. In addition to providing convergence estimates for iterative algorithms, the condition number highlights the importance of solving Gaussian process linear systems in double-precision arithmetic. For example, with $N = 10^6$ or more data points and an $O(N)$ condition number, one is liable to lose nearly all precision in the solution when computing in single precision arithmetic.

In addition to numerical algorithms, we provide theoretical tools for computing with equispaced Fourier Gaussian process representations. We include a formula for the error of the equispaced Fourier Gaussian process discretization, as well as a general formula for the error of the posterior mean of Gaussian process regression at the data points.

We compare the equispaced Fourier algorithms of this paper to several alternative approaches to Gaussian process regression that have desirable theoretical properties and user-friendly implementations in software. Section 5 includes results of numerical experiments for these methods as well as the equispaced Fourier methods introduced in this paper (EFGP).

	error (SE kernel)*	precomputation	solve	mean at q points	variance at q points
PG 2021	$e^{-\gamma m^{1/d}}$	$(N+m^2)\log(N+m^2)$	m^3	$(m+q)\log(m+q)$	$m^2 q$
FLAM	–	$N^{3/2}$	$N\log N$	$(N+q)\log N+q$	$N\log Nq$
RLCM	–	N	N	$N+q\log N$	$N+q\log N$
SKI	$m^{-3/d}$	–	$n_{\text{iter}}\,(N+m\log m)$	Nq	$n_{\text{iter}}\,(N+m\log(m))\,q$
EFGP	$e^{-\gamma m^{1/d}}$	$(N+m)\log(N+m)$	$n_{\text{iter}}\,m\log m$	$(m+q)\log(m+q)$	$n_{\text{iter}}\,m\log(m)\,q$

Table 1: *Computational complexities for various Gaussian process-related tasks for several algorithms for data defined on $\mathbb{R}^d$. For "SKI", m denotes the number of inducing points. For the other algorithms, m is the total number of Fourier modes, i.e. if p nodes are used along each dimension then $m = p^d$.*

In Table 1 we summarize theoretical properties of several algorithms for Gaussian process regression including the methods of [15], SKI [30], RLCM [4], and EFGP. SKI is a so-called inducing point method where inducing points lie on an equispaced grid, which facilitates a fast matrix-vector multiply of the approximate covariance matrix and therefore the efficient solution of linear systems. In FLAM, low-rank interactions between well-separated points are exploited for efficient matrix factorization and inversion in $O(N)$ operations. RLCM constructs a hierarchically low-rank factorization of the covariance matrix in linear time in such a way so as also to allow the application of its inverse in linear time as well.

The methods of this paper are in part motivated by algorithms developed for related signal processing tasks in cryo-electron microscopy (see e.g. [28], [8]) and MRI imaging [14].

The remainder of this paper is structured as follows. In the following section we describe the factorization of a covariance kernel that allows for efficiently computing several Gaussian process statistics. In Section 3 we describe the equispaced Fourier discretization of a Gaussian process. The sources of error in the Fourier representations and subsequent linear solves are presented in Section 4. In Section 5 we demonstrate the results of numerical experiments with our algorithms in addition to several alternatives. We provide some concluding thoughts, and describe plans for future research in Section 6. Finally, in the appendices we include proofs of several analytical results in this paper.

2 Factorization of the kernel

Many computational methods for Gaussian process regression rely explicitly or implicitly on factoring the kernel matrix. In this section we briefly summarize one such factorization, computed using a suitable basis. Let $f : D \to \mathbb{R}$ be distributed according to a zero-mean Gaussian process with covariance kernel $k : D \to \mathbb{R}$. Then by definition (see [22]) for all $x, y \in D$

$$\begin{bmatrix} f(x) \\ f(y) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,y) \\ k(y,x) & k(y,y) \end{bmatrix} \right) . \tag{4}$$

In statistical problems, k is generally chosen to belong to one of a handful of families of kernels such as Matérn, squared-exponential, or rational quadratic. Without loss of generality, we will assume that $D = [0, 1]^d$ for the rest of the discussion.

For the remainder of this paper, we assume, in accordance with the literature, that Gaussian processes are zero-mean and that the observation model for the Gaussian process regression is Gaussian. That is, we assume that observed data y_i is of the form $y_i = f(x_i)+\epsilon_i$ where f is a zero-mean Gaussian process and ϵ_i is iid Gaussian noise. Under this model, the posterior distribution at any point in the domain of f is also Gaussian. Let $\mu : [0, 1]^d \to \mathbb{R}$ denote the posterior mean function and $s : [0, 1]^d \to \mathbb{R}$ denote the posterior variance function of a Gaussian process regression. Suppose that $y \in \mathbb{R}^N$ is the observed data at points $x_1, x_2, \dots x_N \in [0, 1]^d$. Let σ^2 denote a prescribed residual variance, and K denote the $N \times N$ matrix with entries $K_{i,j} = k(x_i, x_j)$. Then the Gaussian process posterior mean and variance are given by

$$\begin{aligned} \mu(\tilde{x}) &= \sum_{j=1}^{N} \alpha_j k(\tilde{x}, x_j)\,, \\ s(\tilde{x}) &= k(\tilde{x}, \tilde{x}) - \sum_j \gamma(\tilde{x})_j k(\tilde{x}, x_j)\,, \end{aligned} \tag{5}$$

where $\alpha, \gamma(\tilde{x}) \in \mathbb{R}^N$ satisfy the linear systems

$$(K + \sigma^2 I)\alpha = y \tag{6}$$

$$(K + \sigma^2 I)\gamma(\tilde{x}) = \tilde{k} \tag{7}$$

where $\tilde{k} = [k(\tilde{x}, x_1), \dots k(\tilde{x}, x_N)]^\mathsf{T}$. The direct computation of the posterior mean requires the solution of an $N \times N$ linear system, followed by an $O(N)$ cost for evaluation at any location $\tilde{x} \in [0, 1]^d$. On the other hand, the evaluation of the posterior variance requires the solution of the same linear system of equations but with a different right-hand-side $\tilde{k}$ for each location $\tilde{x}$. Thus, the computational cost of the posterior mean at q points in $[0, 1]^d$ is $O(N^3 + qN)$, and that of the posterior variance scales like $O(N^3 + qN^2 + qN)$. These costs can be prohibitively expensive for large N, q.

Suppose now that the kernel k admits a finite-rank factorization of the form

$$k(x, y) = \sum_{j=1}^{m} \phi_j(x)\phi_j(y) \tag{8}$$

for a given set of basis functions $\phi_j : [0, 1]^d \to \mathbb{R}$. While the covariance kernels of interest, such as squared exponential, and Matérn kernels, do not admit such a factorization in exact arithmetic, it is possible to construct such a factorization that approximates a desired kernel to high accuracy. We note that a Gaussian process distribution with a finite rank covariance kernel of the form (8) has an interpretation as a basis function expansion

$$f(x) \sim \beta_1\phi_1(x) + \cdots + \beta_m\phi_m(x) \tag{9}$$

with $\beta \sim \mathcal{N}(0, 1)$. In subsequent sections we describe a numerical procedure for constructing approximations of the form (8) for general kernels as well as error estimates for Gaussian process tasks.

For a finite rank covariance kernel k with m basis functions, the corresponding matrix K also admits a rank m factorization independent of the number and location of data points. Specifically, suppose that X is the $N \times m$ matrix given by $X_{i,j} = \phi_j(x_i)$, i.e. X is the matrix of basis functions tabulated at the data points, then $K = XX^\star$. In this setting, the computational cost of evaluating both the posterior mean and posterior variance at q points reduces to $O(Nm^2 + Nm + qN)$ and $O(Nm^2 + qNm + qN)$ respectively – a significant improvement over direct computation for generic kernels, particularly when $m \ll N, q$.

For many problems, the computational cost of evaluating the posterior mean and posterior variance can be reduced further by using an alternate formulation of the posterior mean and posterior variance given by

$$\begin{aligned}\mu(\tilde{x}) &= \sum_{j=1}^{N} \alpha_j k(\tilde{x}, x_j) = \sum_{j=1}^{m} \hat{\beta}_j \phi_j(\tilde{x}) \\ s(\tilde{x}) &= k(\tilde{x}, \tilde{x}) - \sum_{j} \gamma(\tilde{x})_j k(\tilde{x}, x_j) = \sum_{j=1}^{m} \eta(\tilde{x})_j \phi_j(\tilde{x})\end{aligned} \tag{10}$$

where $\hat{\beta}, \eta(\tilde{x}) \in \mathbb{R}^m$ satisfies

$$(X^\star X + \sigma^2 I)\hat{\beta} = X^\star y \tag{11}$$

$$(X^\star X + \sigma^2 I)\eta(\tilde{x}) = \tilde{\phi} \tag{12}$$

where $\tilde{\phi} = [\phi_1(\tilde{x}), \ldots, \phi_m(\tilde{x})]^\mathsf{T}$. Using this formulation, the cost of evaluating the posterior mean and variance reduces to $O(Nm^2+Nm+Mm)$ and $O(Nm^2+MNm+Mm)$ respectively. The linear systems (11) and (7) are closely related – in fact there is a formula that relates their solutions

$$\alpha = \frac{1}{\sigma^2}(y - X\hat{\beta}). \tag{13}$$

We provide a proof of this identity in Appendix A in addition to proofs of the equivalence of the posterior means and posterior variance computed via (5) and (10).

The approach to Gaussian process regression that involves solving linear systems (11) and (12) is often referred to as the weight-space view of Gaussian process regression whereas the approach in (5) is referred to as the function-space view [22].

3 Equispaced Fourier representation of GPs

In this, section we describe a numerical approach for constructing a low-rank approximation of the form (8) to a translationally invariant covariance kernel $k(x - y)$. The approach we propose uses complex exponentials with equispaced frequencies for the basis functions ϕ_j. We review several well-known properties of this discretization and use these properties throughout the remainder of the paper. Such equispaced spectral discretizations tend to be efficient numerical tools in low dimensions ($\mathbb{R}^d$ for d around 4 or smaller).

3.1 Discretized inverse Fourier transform

Suppose that $k : [0,1]^d \times [0,1]^d \to \mathbb{R}$ is an integrable and translation-invariant covariance kernel of a Gaussian process. In a slight abuse of notation we use k interchangeably as $k(x,y) = k(x-y)$. We follow the Fourier transform convention of [22]

$$\hat{k}(\xi) = \int_{\mathbb{R}^d} k(x) e^{-2\pi i \langle \xi, x \rangle}\, dx, \tag{14}$$

$$k(x) = \int_{\mathbb{R}^d} \hat{k}(\xi) e^{2\pi i \langle \xi, x \rangle}\, d\xi, \tag{15}$$

for all $x \in \mathbb{R}^d$ and $\xi \in \mathbb{R}^d$. The basis function approximations we use for Gaussian process distributions can be viewed as discretized versions of the Fourier inversion formula (15). In particular, we approximate k via

$$\tilde{k}(x-y) = \sum_{j \in I_m} h^d \hat{k}(hj) e^{2\pi i h \langle j, (x-y) \rangle}\,, \tag{16}$$

where $I_m = \{(j_1, j_2, \ldots j_d) : j_n \in \{-m, \ldots m\}\ , n = 1, 2, \ldots d\}$ and h is a constant. Note that $\tilde{k}$ can be interpreted as the periodic trapezoidal rule applied to the integral in 15 truncated in the box $[-mh, mh]^d$. The following proposition constructs the basis functions ϕ_j such that $\tilde{k}(x-y) = \sum_{j \in I_m} \phi_j(x) \overline{\phi_j(y)}$ where $\overline{\phi}_j$ denotes the complex conjugate of ϕ_j.

Proposition 3.1. *Let $g : \mathbb{R} \to \mathbb{C}$ be the expansion defined by*

$$g(x) \sim \sum_{j \in I_m} \beta_j \sqrt{h^d \hat{k}(jh)} e^{2\pi h i \langle j, x \rangle} \tag{17}$$

where the coefficients β_j, $j \in I_m$ are iid $\beta_j \sim \mathcal{N}(0,1)$, $\hat{k} : \mathbb{R}^d \to \mathbb{R}$ denotes the Fourier transform of the covariance kernel $k : \mathbb{R}^d \to \mathbb{R}$ and h is a constant. Then g is a Gaussian process with covariance kernel $\tilde{k}$ defined by the formula 16.

Proof. By definition, $\tilde{k}(x,y) = E[g(x)\overline{g(y)}]$ and

$$\begin{aligned} E[g(x)\overline{g(y)}] &= \sum_{j \in I_m} h^d \hat{k}(hj) e^{2\pi h i \langle j, x \rangle} e^{-2\pi h i \langle j, y \rangle} \\ &= \sum_{j = I_m} h^d \hat{k}(hj) e^{2\pi h i \langle j, (x-y) \rangle}\,. \end{aligned} \tag{18}$$

□

There are two primary reasons that the approximation of the covariance kernel by using the periodic trapezoid rule for its Fourier transform is attractive in this environment. First, the trapezoid rule has super-algebraic convergence when the integrand is smooth (C^∞) and vanishes at the endpoints. That is, the error of the order-m trapezoid rule,

$$\left| \int_{\mathbb{R}^d} \hat{k}(\xi) e^{2\pi i \langle \xi, x \rangle} d\xi - \sum_{j \in I_m} h^d \hat{k}(hj) e^{2\pi i h \langle j, x \rangle} \right| \leq \mathcal{O}\left(\frac{1}{m^n} \right)\,, \tag{19}$$

for any integer n. Many commonly-used covariance kernels have smooth Fourier transforms that vanish (numerically) for large frequencies. However, for certain kernels such as the Matérn 1/2 [22], which are non-smooth at 0, the decay of the Fourier transform is slow and the error of the quadrature rule approximation is dominated by the choice of interval $[-mh, mh]^d$. We elaborate on errors of the equispaced quadrature rule in Section 4.

Second, equispaced discretizations in Fourier domain facilitate the use of the fast Fourier transform (FFT) and non-uniform FFTs [9, 14] for Gaussian process regression tasks. In particular, when using the equispaced discretization in Fourier domain, the solution to the weight-space linear system

$$(X^*X + \sigma^2 I)\hat{\beta} = X^*y \tag{20}$$

has a matrix that can be applied in $O(m^d \log m^d)$ after $O((N + m^d)\log(N + m^d))$ precomputation using FFT-based methods. We describe the details of this procedure in the next section.

3.2 Numerical Implementation

Recall that the computation of the posterior mean using the weight-space representation for the approximated kernel $\tilde{k}$ requires the solution of the linear system

$$(X^*X + \sigma^2 I)\hat{\beta} = X^*y \tag{21}$$

where $y \in \mathbb{R}^N$ and X is the $N \times (2m+1)^d$ matrix defined by

$$X = \begin{bmatrix} \sqrt{h\hat{k}(h^d j_1)}e^{2\pi hi\langle j_1, x_1\rangle} & \dots & \sqrt{h^d\hat{k}(hj_n)}e^{2\pi hi\langle j_n, x_1\rangle} \\ \sqrt{h\hat{k}(h^d j_1)}e^{2\pi hi\langle j_1, x_2\rangle} & \dots & \sqrt{h^d\hat{k}(hj_n)}e^{2\pi hi\langle j_n, x_2\rangle} \\ \vdots & & \vdots \\ \sqrt{h\hat{k}(h^d j_1)}e^{2\pi hi\langle j_1, x_N\rangle} & \dots & \sqrt{h^d\hat{k}(hj_n)}e^{2\pi hi\langle j_n, x_N\rangle} \end{bmatrix} \tag{22}$$

where $x_1, ..., x_N$ are observed data, $n = (2m+1)^d$, $j_1, j_2 \dots j_n$, is an enumeration of the index set I_m, and h is the frequency spacing. Column p of X is a complex exponential (basis function) tabulated at the data points. That is, $X_{i,p} = \phi_p(x_i)$. We observe that X^*X can be factorized as

$$X^*X = DX'^*X'D \tag{23}$$

where X' is the $N \times (2m+1)^d$ matrix defined by

$$X' = \begin{bmatrix} e^{2\pi hi\langle j_1, x_1\rangle} & \dots & e^{2\pi hi\langle j_n, x_1\rangle} \\ e^{2\pi hi\langle j_1, x_2\rangle} & \dots & e^{2\pi hi\langle j_n, x_2\rangle} \\ \vdots & & \vdots \\ e^{2\pi hi\langle j_1, x_N\rangle} & \dots & e^{2\pi hi\langle j_n, x_N\rangle} \end{bmatrix} \tag{24}$$

and D is the diagonal $(2m+1)^d \times (2m+1)^d$ matrix

$$D = \begin{bmatrix} \sqrt{h^d \hat{k}(hj_1)} & & \\ & \ddots & \\ & & \sqrt{h^d \hat{k}(hj_n)} \end{bmatrix}. \tag{25}$$

Since we are using an equispaced discretization, the $(2m+1)^d \times (2m+1)^d$ matrix $X'^* X'$ is a d dimensional Kronecker product of $(2m+1) \times (2m+1)$ Toeplitz matrices with

$$(X'^* X')_{p,\ell} = \sum_{n=1}^{N} e^{2\pi h i \langle (j_p - j_\ell), x_n \rangle}. \tag{26}$$

which can be applied to a vector in $O(m^d \log m^d)$ operations using the FFT (see e.g. [7]). Solving (21) can therefore be efficiently solved with iterative methods such as conjugate gradient [7], provided that the matrix is not too poorly conditioned. In order to apply $X'^* X'$ in $O(m^d \log m^d)$ operations, we first need to precompute its unique elements, given by

$$v_j = \sum_{n=1}^{N} e^{2\pi i h \langle j, x_n \rangle} \tag{27}$$

for $j = I_m \ominus I_m$, where $A \ominus B$ is the set of unique elements $(a-b)$ with $a \in A$ and $b \in B$. Note that there are $2|I_m| - 1$ elements in $I_m \ominus I_m$. The quantities v_j can be computed via a non-uniform fast Fourier transform using, for example, [2], in $O((N + m^d)\log(N + m^d))$ operations.

We now describe a numerical algorithm for evaluating the posterior mean of Gaussian process regression using equispaced Fourier discretizations and fast algorithms.

Algorithm 1 (GP regression via equispaced Fourier modes)**.**

1. *Given a tolerance ε, determine m, h such that $\tilde{k}$ is an ε-accurate approximation of k. We postpone the estimation of m, h that satisfies this criterion to Section 4.*

2. *Use the non-uniform FFT to evaluate the sums v_ℓ defined by formula (27).*

3. *Evaluate $X^* y$ using a non-uniform FFT.*

4. *Use conjugate gradient to solve linear system*

$$(X^* X + \sigma^2 I)\hat{\beta} = X^* y \tag{28}$$

where $X = X'D$ is defined in (22). Since $X'^* X'$ is d dimensional Kronecker product of Toeplitz matrices, the matrix of linear system (28) can be applied in $O(m^d \log m^d)$ operations using the precomputed v_ℓ.

5. *Evaluate the posterior mean, μ_{ws} (see (10)) at points $z_1, ..., z_q$ using a nonuniform FFT in $O((q+m^d)\log(q+m^d))$ operations.*

Remark 3.1. *In many applications, hyperparameters of the covariance function are fit to the data using methods that require the gradient of the Gaussian process likelihood function in addition to the determinant of the matrix $K+\sigma^2 I$. Methods for the evaluation of these quantities is the subject of a large literature including, for example, [11, 18, 29]. We leave the evaluation of these quantities using equispaced Fourier representations to future work.*

4 Error analysis

The error in the posterior mean computed using $\tilde{k}$ relative to the posterior mean computed using k is related to the Frobenius norm of $K-\tilde{K}$ where K and $\tilde{K}$ are the covariance matrices sampled at the data points $x_1, \dots x_N$. The Frobenius norm in turn can be estimated through the L^2 or L^∞ error between k and $\tilde{k}$ on $[-1,1]^d$. In this section we provide analytical tools for approximating the error introduced by the equispaced Fourier discretization as well as the numerical errors and convergence rates of conjugate gradient. In addition to having desirable computational properties, equispaced Fourier modes have well-known and useful analytical properties that we use in this section for estimating approximation errors.

4.1 Discretization error

In the following we provide a formula for the error in the covariance function of a Gaussian process approximation using equispaced Fourier expansions.

Proposition 4.1 (Accuracy of GP approximation). *Suppose that $k:\mathbb{R}^d\to\mathbb{R}$ is an absolutely integrable function with Fourier transform $\hat{k}$. Then for all $h>0$ and $x\in\mathbb{R}^d$ we have*

$$\tilde{k}(x)-k(x) = h^d\sum_{j\in I_m}\hat{k}(jh)e^{2\pi hi\langle j,x\rangle} - k(x) = \sum_{\substack{j\in\mathbb{Z}^d\\ j\neq 0}} k\left(x+\frac{j}{h}\right) - h^d\sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}}\hat{k}(jh)e^{2\pi hi\langle j,x\rangle} \tag{29}$$

where $\tilde{k}$ is the effective covariance kernel of the equispaced Fourier Gaussian process (see (16)).

Proof. The result follows from the periodic version of the Poisson summation formula given by

$$h^d\sum_{j\in\mathbb{Z}^d}\hat{k}(jh)e^{2\pi hi\langle j,x\rangle} = \sum_{j\in\mathbb{Z}^d} k\left(x+\frac{j}{h}\right) \tag{30}$$

for all $x\in\mathbb{R}^d$ and $h>0$.

□

Proposition 4.1 provides a formula for the pointwise difference between the kernel of a Gaussian process and the effective kernel of a Gaussian process approximated with an equispaced Fourier expansion. The first term in the error estimate is often referred to as the aliasing error, and the latter is referred to as the truncation error. The frequency-spacing h is chosen to ensure that the aliasing error is small, and given h, m is chosen to ensure that the truncation error is small.

Let $G(x,\ell)$ denote the d-dimensional squared exponential kernel with length scale ℓ, given by

$$G(x,\ell) = \exp\left(-\frac{|x|^2}{2\ell^2}\right). \tag{31}$$

Let $C_\nu(x,\ell)$ denote the Matérn kernel with parameter ν and length scale ℓ given by

$$C_\nu(x,\ell) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{|x|}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x|}{\ell}\right), \tag{32}$$

where K_ν is the modified Bessel function of the second kind and $\nu \geq 1/2$.

For the squared exponential kernel, both the kernel in the real space and the Fourier space decay extremely rapidly. In this situation, it is easy to show that choosing $h = \min\left(O(1), O(1/(\ell\sqrt{\log(1/\varepsilon)})\right)$ suffices to bound the aliasing error, and $m = O(\frac{\log 1/\varepsilon}{\ell})$ is sufficient to bound the truncation error. The restriction of h being at least $O(1)$ arises from the fact that the first aliased copy of the kernel should be outside the domain of x, i.e. $[-1,1]^d$, independent of how small ℓ is. The $O(1/(\ell\sqrt{\log(1/\varepsilon)}))$ constraint then comes from the fact that the Fourier transform of the squared exponential kernel is, up to a constant, bounded by $\exp\left(-8\pi^2h^2\ell^2m^2\right)$. When $h = O(1)$, we see that m has to be $m = O(\frac{\log 1/\varepsilon}{\ell})$ to achieve the desired accuracy.

In the following lemma, we state error estimates for the aliasing and the truncation errors for the Matérn kernel. The proofs of these results are included in Appendix B.

Lemma 1 (Aliasing and truncation error for the Matérn covariance kernel)**.** *Suppose $k(x) = C_\nu(x,\ell)$, then the aliasing error is given by*

$$\left|\sum_{\substack{j\in\mathbb{Z}^d\\ j\neq 0}} k\left(x+\frac{j}{h}\right)\right| \leq \alpha(d,\nu)C_0\exp\left(-\frac{\sqrt{2\nu}}{\ell}\left(\frac{1}{h}-|x|\right)\right), \tag{33}$$

for all $x\in[-1,1]^d$ where C_0 is the value of the Matérn kernel at the origin. The L^2 and L^∞ truncation errors are given by

$$\begin{aligned}\left|h^d\sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}}\hat{k}(jh)e^{2\pi hi\langle j,x\rangle}\right|_{L^\infty[-1,1]^d} &\leq \hat{\alpha}_\infty(d,\nu)\left(\frac{\sqrt{2\nu}}{h\ell m}\right)^{2\nu},\\ \left|h^d\sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}}\hat{k}(jh)e^{2\pi hi\langle j,x\rangle}\right|_{L^2[-1,1]^d} &\leq \frac{\hat{\alpha}_2(d,\nu)}{\sqrt{m}^d}\left(\frac{\sqrt{2\nu}}{h\ell m}\right)^{2\nu}.\end{aligned} \tag{34}$$

Here $\alpha(d,\nu)$ is a constant independent of h, and $\hat{\alpha}_2(d,\nu), \hat{\alpha}_\infty(d,\nu)$ are constants independent of (h,m).

The above estimate indicates that in both L^2 and L^∞, in order to achieve an accuracy of ϵ, we have $h = \min\left(2/\sqrt{d}, O(\sqrt{\nu}/(\ell \log(1/\varepsilon)))\right)$ up to a constant that depends in a mild manner on d. On the other hand, the number of modes m satisfies $m = O(\log(1/\varepsilon)/\varepsilon^{1/(2\nu)})$ in L^∞ and $m = O(\log^{(2\nu/(2\nu+d/2))}(1/\varepsilon)/\varepsilon^{1/(2\nu+d/2)})$ in L^2.

4.2 Accuracy of posterior mean

While thus far we have focused on the accuracy of Fourier approximations to a Gaussian process distribution, in this section we describe the downstream errors of those approximations in Gaussian process regression. In particular, we bound the difference between the true posterior mean of Gaussian process regression (exact inference) and the approximate Gaussian process posterior mean using the methods of this paper. We are primarily focused on the scaling of the errors as a function of N, the number of data points, for a fixed domain. This subject has been the focus of a large literature (e.g. [24,25,27]).

In the following proposition, we bound the Frobenius norm of the difference between an exact covariance matrix and an approximation.

Proposition 4.2. *Let $k : [0,1]^d \times [0,1]^d \to \mathbb{R}$ be a covariance kernel and let $\tilde{k} : [0,1]^d \times [0,1]^d \to \mathbb{R}$ be an approximation to k. Suppose that $|k(x) - k(\tilde{x}| < \varepsilon$ for all $x \in [-1,1]^d$. For any collection of points $x_1, ..., x_N$ on $[0,1]^d$, let K be the $N \times N$ matrix $K_{i,j} = k(x_i, x_j)$ and $\tilde{K}$ to be the $N \times N$ matrix $\tilde{K} = \tilde{k}(x_i, x_j)$. Then*

$$\|K - K_F\| \leq N\varepsilon\,. \tag{35}$$

Suppose further that the points $x_1, x_2 \ldots x_N$ are uniformly distributed on $[0,1]^d$ and let

$$\int_{[-1,1]^d} |k(x) - \tilde{k}(x)|^2 dx < \tilde{\varepsilon}^2\,, \tag{36}$$

then

$$\|K - \tilde{K}\|_F = N\tilde{\varepsilon}\left(1 + O(1/\sqrt{N})\right)\,. \tag{37}$$

Proof. If the L^∞ error estimate holds for the approximation of k using $\tilde{k}$, then equation 35 follows trivially using the entrywise estimate in $K - \tilde{K}$. For the L^2 estimate, we first note that

$$\int_{[0,1]^d}\int_{[0,1]^d} |k(x,y) - \tilde{k}(x,y)|^2\, dx\, dy \leq \int_{[-1,1]^d} |k(x) - \tilde{k}(x)|^2\, dx\; = \tilde{\varepsilon}^2 \tag{38}$$

Moreover,

$$\|K - \tilde{K}\|_F^2 = \sum_{i,j=1}^{N} (k(x_i,x_j) - \tilde{k}(x_i,x_j))^2 \tag{39}$$

$$= N^2\left(\frac{1}{N^2}\sum_{i,j=1}^{N} (k(x_i,x_j) - \tilde{k}(x_i,x_j))^2\right). \tag{40}$$

That is, $\frac{1}{N^2}\|K - \tilde{K}\|_F^2$ is a Monte Carlo approximation of $\tilde{\varepsilon}^2$. Substituting the half-order convergence of Monte Carlo, we have

$$\|K - \tilde{K}\|_F^2 = N^2\tilde{\varepsilon}^2\left(1 + O\left(\frac{1}{\sqrt{N^2}}\right)\right). \tag{41}$$

□

Remark 4.1. *When using the L^∞ estimate for the approximate kernel, no assumption needs to be made on the distribution of the data points $x_1, \dots x_N$. On the other hand, the L^2 estimate approximates the Frobenius norm of the error in the covariance matrix, only if the data points $x_1, \dots x_N$ are uniformly distributed on $[0,1]^d$. The L^2 estimate even though restrictive is particularly useful when approximating the Matérn $1/2$ kernel due to the $O(1/m)$ decay in the L^∞ truncation error independent of dimension. This would imply $O(1/\varepsilon^d)$ points in the Fourier domain are required to approximate this kernel. However, the L^2 error bound is more forgiving, particularly in higher dimensions, and would require $O(1/\varepsilon^{(1+d/2)})$ points along each dimension which is a significant improvement.*

We now turn our attention to estimating the error in the posterior mean computed via the approximated covariance kernel. The following lemma bounds the difference between the exact solution to the Gaussian process linear system, and the solution to an approximate linear system.

Lemma 1. *Let $y \in \mathbb{R}^N$ and let $K, \tilde{K}$ be $N \times N$ matrices such that*

$$(K + \sigma^2 I)\alpha = y, \qquad (\tilde{K} + \sigma^2 I)\tilde{\alpha} = y \tag{42}$$

for some $\sigma > 0$ where $\alpha, \tilde{\alpha} \in \mathbb{R}^N$. Then,

$$\|\tilde{\alpha} - \alpha\| \leq \|K - \tilde{K}\|\|(K - \sigma^2 I)^{-1}\|\|\alpha\| \tag{43}$$

where $\|\cdot\|$ denotes the ℓ^2-norm for vectors and the spectral norm for matrices.

Proof. Using (42), we have

$$\begin{aligned} \|\tilde{\alpha} - \alpha\| &= \|(\tilde{K} + \sigma^2 I)^{-1}y - (K + \sigma^2 I)^{-1}y\| \\ &= \|((\tilde{K} + \sigma^2 I)^{-1} - (K + \sigma^2 I)^{-1})y\|. \end{aligned} \tag{44}$$

Using the identity

$$A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}, \tag{45}$$

for any invertible matrices A, B, we have

$$\begin{aligned} \|\alpha - \tilde{\alpha}\| &\leq \|((\tilde{K} + \sigma^2 I)^{-1}(K - \tilde{K})(K + \sigma^2 I)^{-1}y\| \\ &\leq \|((\tilde{K} + \sigma^2 I)^{-1}\|\|K - \tilde{K}\|\|(K + \sigma^2 I)^{-1}y\| \\ &= \|((\tilde{K} + \sigma^2 I)^{-1}\|\|K - \tilde{K}\|\|\alpha\|. \end{aligned} \tag{46}$$

□

The following theorem is the primary analytical tool of this section. It bounds the error of the posterior mean at data points when using an approximate Gaussian process algorithm. We measure errors in the posterior mean relative to the l^2-norm of the observed data.

Theorem 1. *Let $y \in \mathbb{R}^N$ and let $K, \tilde{K}$ be $N \times N$ matrices such that*

$$(K + \sigma^2 I)\alpha = y, \qquad (\tilde{K} + \sigma^2 I)\tilde{\alpha} = y \tag{47}$$

for some $\sigma > 0$ where $\alpha, \tilde{\alpha} \in \mathbb{R}^N$. Then,

$$\frac{\|K\alpha - \tilde{K}\tilde{\alpha}\|}{\|y\|} \leq \frac{\|K - \tilde{K}\|}{\sigma^2} \tag{48}$$

where $\|\cdot\|$ denotes the ℓ^2 norm for vectors and the spectral norm for matrices. The vectors $K\alpha, \tilde{K}\tilde{\alpha} \in \mathbb{R}^N$ are the posterior means at the data points using an exact algorithm and an approximate one.

Proof. Using (47), we know

$$\begin{aligned} \frac{\|K\alpha - \tilde{K}\tilde{\alpha}\|}{\|y\|} &= \frac{\|(y - \sigma^2\alpha) - (y - \sigma^2\tilde{\alpha})\|}{\|y\|} \\ &= \frac{\|\sigma^2(\tilde{\alpha} - \alpha)\|}{\|y\|} \\ &= \frac{\|\sigma^2(\tilde{\alpha} - \alpha)\|}{\|(K + \sigma^2 I)\alpha\|}. \end{aligned} \tag{49}$$

Applying Lemma 1 to (49) we obtain

$$\frac{\|K\alpha - \tilde{K}\tilde{\alpha}\|}{\|y\|} \leq \sigma^2 \|K - \tilde{K}\| \|(K + \sigma^2 I)^{-1}\| \frac{\|\alpha\|}{\|(K + \sigma^2 I)\alpha\|}. \tag{50}$$

We obtain (48) by applying to (50) the bounds

$$\|(K + \sigma^2 I)^{-1}\| \leq \frac{1}{\sigma^2} \qquad \text{and} \qquad \frac{\|\alpha\|}{\|(K + \sigma^2 I)\alpha\|} \leq \frac{1}{\sigma^2}. \tag{51}$$

□

4.3 Conditioning of Gaussian process covariance matrices

In this section we gather upper and lower bounds on $\kappa(K + \sigma^2 I)$, the condition number of the GP weight-space system matrix. The spatial kernel $k(x, y) = k(x - y)$ is assumed convolutional, non-negative and bounded by $k(x) \leq k(\mathbf{0}) = k_0$ its diagonal value, for all $x \in \mathbb{R}^d$. k_0 is interpreted as the GP prior variance.

Lemma 2 (Upper bound)**.** *Let K be the $N \times N$ kernel matrix between the points $\{x_j\}_{j=1}^N$, with prior variance k_0, and $\sigma > 0$. Then*

$$\kappa(K + \sigma^2 I) \ \leq \ \frac{N}{\eta} + 1 + \frac{\eta}{2} \tag{52}$$

where $\eta := \sigma^2/k_0$ is the measurement-to-prior variance ratio.

Proof. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N$ be the singular values of $K + \sigma^2 I$. Diagonal entries of $K + \sigma^2 I$ are bounded by $k_0 + \sigma^2$ and off-diagonal by k_0. Thus we bound the spectral norm by the Frobenius norm [12, (2.3.7)],

$$\begin{aligned}
\sigma_1 \;&\leq\; \|K + \sigma^2 I\|_F \leq \sqrt{N^2 k_0^2 + N(2k_0\sigma^2 + \sigma^4)} \leq N k_0 \sqrt{1 + \frac{2\eta + \eta^2}{N}} \\
&\leq\; N k_0 (1 + \frac{\eta + \eta^2/2}{N}) = N k_0 + \sigma^2 (1 + \eta/2) \qquad (53)
\end{aligned}$$

The condition number is by definition $\kappa = \sigma_1/\sigma_N$, and since K is SPD, $\sigma_N \geq \sigma^2$. □

This shows that $\kappa = O(N)$ for the case of fixed kernel k and fixed σ, as the number of data points $N \to \infty$. The case of constant kernel $k \equiv 1$, where $\sigma_2 = \sigma_3 = \cdots = \sigma_N = \sigma^2$, shows that this N-growth (including the prefactor) is tight. This can be viewed as the limit where all points are much closer than the kernel width. Tightness of the N-independent corrections has been dropped for simplicity.

For any fixed N, there is a kernel such that $\kappa(K + \sigma^2 I) = 1$, showing ideal conditioning: this is achieved by making the kernel width parameter ℓ much smaller than any distance between data points, so that $K \to k(0)I$, a multiple of the identity.

Lemma 2 can be used with the standard convergence rate of conjugate gradient to obtain a convergence estimate for solving the weight-space linear systems. Specifically, if $\hat{\beta}$ is the exact solution to linear system (28) and $\hat{\beta}_n$ is the approximate solution after n iterations of conjugate gradient then

$$\|\hat{\beta}_n - \hat{\beta}\| = O\left(\left(\frac{\sqrt{\kappa(K + \sigma^2 I)} - 1}{\sqrt{\kappa(K + \sigma^2 I)} + 1} \right)^n \right) \qquad (54)$$

where $\kappa(K + \sigma^2 I)$ is defined in (52).

4.4 Empirical condition number

The plots in Figures 1 and 2 compare the empirical conditional number of the covariance matrix of the Gaussian process likelihood with the theoretical bound we establish in Lemma 2. These plots demonstrate that even for reasonably uniform data, the theoretical upper bound differs from the empirical condition number by only a factor of roughly 2.

For both figures, we used the squared-exponential kernel with a timescale of $\ell = 0.1$. We generated points $x_1, ..., x_N$ uniformly on the interval $[0, 1]$ for various N and compare the condition number of the $N \times N$ matrix $K_{i,j} = k(x_i, x_j)$ to the theoretical bound in Lemma 2. The determinant of K was computed by first using Algorithm 1 to construct the $N \times m$ matrix X such that XX^* approximates K to high accuracy. Since for the squared-exponential kernel, m is significantly smaller than N, $\kappa(K + \sigma^2 I)$ can be efficiently computed, even for large N, using for example the SVD or X.

In Figure 1, we plot both the bound given in Lemma 2 and the empirical condition number for $\sigma^2 = 0.09$. In Figure 2 we compare the empirical condition to the theoretical bound for various N and σ^2.

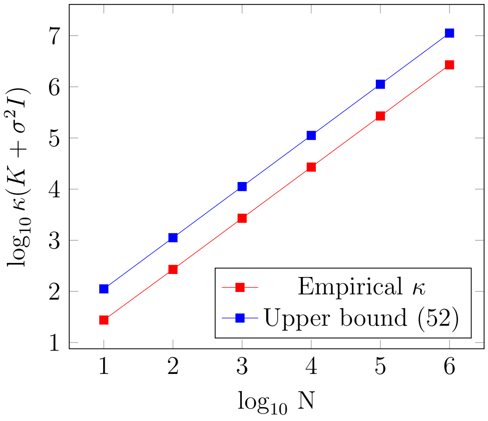
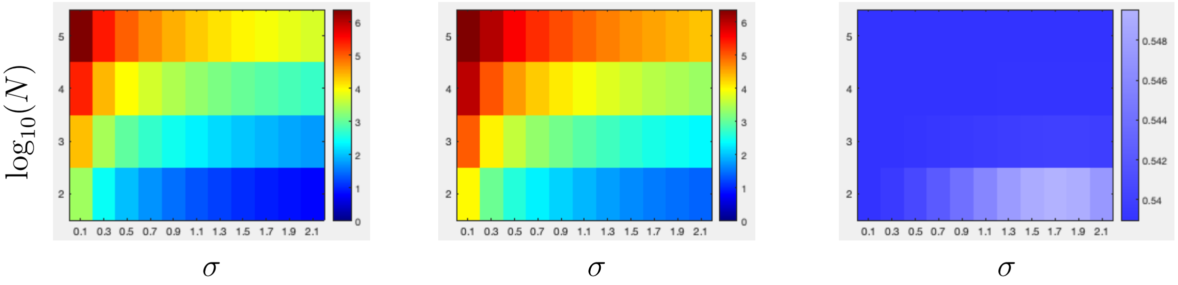


Figure 1: Condition numbers of the Gaussian process covariance matrix as a function of the number of data points, N. The data points are distributed uniformly on $[0, 1]$, the kernel is the squared-exponential kernel with $\ell = 0.1$, and $\sigma^2 = 0.09$.



(a) $\log_{10} \kappa(K + \sigma^2 I)$

(b) $\log_{10}$ of bound (52)

(c) Ratio of κ to bound (52)

Figure 2: Condition numbers of the covariance matrix of Gaussian process regression for various N and σ^2 in addition to the theoretical bounds and the ratio between the two. The squared-exponential kernel with $\ell = 0.1$ was used.

5 Numerical experiments

We implemented Algorithm 1 (EFGP) in MATLAB version 2021b on a 2.6 GHz 6-Core Intel Core i7 MacBook Pro. We describe the performance of EFGP in detail on a number of Gaussian process regression tasks with both synthetic data and an applied problem. We also compare EFGP to well-known implementations of related algorithms.

5.1 EFGP time and accuracy for various N

In Table 2 we present the detailed performance of EFGP in accuracy and compute time for Gaussian process regression problems in 1, 2, and 3 dimensions with varying numbers of data points and two different kernels.

In Table 2 all data was simulated. In Tables 2a and 2b, data was generated according to $y_i = \cos(6\pi x_i + 1.3) + \epsilon_i$ with $\epsilon \sim \mathcal{N}(0, 0.3^2 I)$ for $x_1, ..., x_N$ uniformly sampled points on the interval $[0, 1]$. For Tables 2c and 2d, data was generated according to $y_i = \cos(2\pi\langle x_i, \omega\rangle +$

$1.3) + \epsilon_i$ where $\omega = [4\ 3]^t$ and $\epsilon_i \sim \mathcal{N}(0, 0.3^2 I)$ for $x_1, ..., x_N$ uniformly sampled points on the square $[0,1]^2$. The data used for GP regression in Tables 2e and 2f was generated according to $y_i = \cos(2\pi\langle x_i, \omega\rangle + 1.3) + \epsilon_i$ where $\omega = [3\ 7\ 2]^t$ and $\epsilon_i \sim \mathcal{N}(0, 0.3^2 I)$ for $x_1, ..., x_N$ uniformly sampled points on the cube $[0,1]^3$.

In Table 2, the column "m" denotes the m of Algorithm 1, the number of non-negative frequencies used in each dimension. That is, for a d-dimensional problem, the total number of frequencies is $(2m+1)^d$. In Table 2, the column "precomp (s)" represents the total precomputation time – discretization of the Gaussian process distribution, using the non-uniform FFT to construct the right-hand-side of the weight-space linear system, and precomputation for applying the Toeplitz matrix. The column labeled "CG (s)" denotes the total time spent on conjugate gradient for solving the linear system. "mean (s)" presents the time spent on evaluating the posterior mean. For the d-dimensional problem, the posterior mean was evaluated at 100 points in each dimension on $[0,1]^d$. For example, with $d = 2$, we evaluated the posterior mean on a 100×100 grid on the square $[0,1]^2$. The column labeled "total (s)" denotes the total time of Gaussian process regression and evaluation of the posterior mean. The total number of iterations required for conjugate gradient is provided in "CG iters" and the L^2 or root mean squared error (RMSE) of the posterior mean is provided in "RMSE." Specifically, RMSE denotes the quantity

$$\left(\int_D (\mu(x) - \tilde{\mu}(x))^2 dx \right)^{1/2} \tag{55}$$

where $D = [0,1]^d$ in d dimensions, $\mu : \mathbb{R}^d \to \mathbb{R}$ denotes the true posterior mean, and $\tilde{\mu} : \mathbb{R}^d \to \mathbb{R}$ the approximate posterior mean evaluated with EFGP. This error was computed using as a benchmark EFGP with a high level of accuracy (i.e. 10^{-12} for squared exponential kernel).

5.2 Accuracy vs. time

In Figure 3 we compare four Gaussian process algorithms. For each algorithm, we evaluate the amount of time required to obtain a certain level of accuracy of the posterior mean. We measure accuracy via the L^2 difference between the exact posterior mean and the approximate posterior mean (see (55) obtained with each method. We perform with simulated data in 1, 2, and 3 dimensions and two kernels – squared exponential and Matérn 1/2.

The timings in Figure 3 reflect total time for evaluating the posterior mean at points on an equispaced grid on $[0,1]^d$ in d dimensions. For the 1-dimensional problem, the posterior mean was tabulated at 100 equispaced points. In 2-dimensions, the posterior mean was evaluated on a 100×100 equispaced grid on the square $[0,1]^2$. The posterior mean of the 3-dimensional problems was tabulated on a $30 \times 30 \times 30$ equispaced grid on the cube $[0,1]^3$.

We used simulated data for these experiments with $N = 10^5$ points chosen uniformly at random on $[0,1]^d$ in d dimensions. The residual variance was $\sigma^2 = 0.25$ for all experiments. In 1 dimension, the data was generated according to $y_i = \cos(6\pi x_i + 1.3) + \epsilon_i$ where $\epsilon_i \sim \mathbb{N}(0, 0.5^2)$ was generated iid. For the 2-dimensional experiments, data was generated according to $y_i = \cos(2\pi\langle x, \omega\rangle + 1.3) + \epsilon_i$ where $\omega = [4\ 3]$ and $\epsilon_i \sim \mathbb{N}(0, 0.5^2)$. In the 3-dimensional experiments, data was generated according to $y_i = \cos(2\pi\langle x, \omega\rangle + 1.3) + \epsilon_i$ where $\omega = [2\ 3\ 5]$ and $\epsilon_i \sim \mathbb{N}(0, 0.5^2)$.

We compare EFGP to three algorithms – structured kernel interpolation (SKI) [30] method, RLCM of [4], and a fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations (FLAM), [18]. We chose these algorithms as a comparison due to their desirable theoretical properties and efficient, user-friendly, publicly available implementations. All three comparison algorithms, and indeed nearly all of the literature, are function-space approaches. That is, they solve the $N \times N$ linear systems in (5).

SKI [30] falls broadly in the class of Gaussian process algorithms known as inducing point methods. In SKI, inducing points fall on an equispaced grid, which facilitates a fast matrix-vector multiply of the approximate covariance matrix using FFTs. Tradeoff of accuracy and runtime was achieved by altering the number of grid points. We use the implementation of GPyTorch [11].

In FLAM, low-rank interactions between well-separated points is exploited for efficient matrix factorization and inversion in $O(N)$ operations. We used the implementation of [17]. Higher accuracy was accomplished by reducing the error tolerance input.

RLCM constructs a hierarchical factorization of a covariance matrix in linear time (in the number of data points) that allows application of the inverse of the covariance matrix in $O(N)$ operations. The factorization is so-called hierarchically low-rank – the covariance matrix is represented as a block diagonal plus low-rank where each block diagonal has the same structure. The accuracy of this method was adjusted by changing the maximum rank of the low-rank terms of the decomposition. We used the authors' implementation found at `https://github.com/jiechenjiechen/RLCM` .

5.3 Spatial Gaussian process regression with satellite data

We demonstrate the performance of EFGP on a standard large-scale Gaussian process problem in spatial statistics. The observed data, [19], consists of over 1.4 million satellite measurements of XCO2 (average CO_2 in an atmospheric column) taken during the period of August 1, 2015 - August 17, 2015. In [6], the author describes this dataset in detail in addition to the data-generating process, the sources of noise in the data, and the applied significance of performing efficient Gaussian process regression on this data.

In accordance with standard practice, we use longitude-latitude coordinates as locations on a 2-dimensional plane. We ran Gaussian process regression using EFGP with the squared exponential kernel and several values of ℓ. For $\ell = 5, 50$, we compute the posterior mean in 7 and 0.5 seconds with RMSE of 0.0005 and 0.001. As in previous experiments, RMSE denotes the L^2 error of the posterior mean (see 55). In Figure 4 we plot the posterior means on a 300×300 grid.

6 Generalizations and conclusions

In this paper, we present a class of algorithms for Gaussian process regression and related tasks. Our methods use an equispaced Fourier expansion to represent a Gaussian process distribution, which has two primary advantages – accurate and compressed approximations of the Gaussian process, and the use of fast algorithms for inference.

In addition to introducing numerical algorithms, we also present several formulae and analytical tools. In particular, we give a bound on the condition number of a Gaussian process covariance matrix, we provide a formula for the error of equispaced Fourier discretizations in L^2 and L^∞, and we bound the error of the posterior mean evaluated at the data points for general approximate Gaussian process methods.

The performance of our algorithms was tested with numerical experiments on real and simulated data. We compared our algorithms with several state-of-the-art methods for Gaussian process regression in the speed and accuracy of evaluation of the posterior mean. In general, our algorithms compared favorably to other methods on a range of problems in 1, 2, and 3 dimensions.

In future work we plan to address the evaluation of gradients of the Gaussian process posterior for training of hyperparameters as well as the efficient evaluation of the determinant of the Gaussian process likelihood.

A Equivalence of weight-space and function-space

Suppose that $\phi_1, ..., \phi_m : D \to \mathbb{C}$ are a collection of fixed basis functions. We denote by g the Gaussian process distribution defined by

$$g \sim \beta_1 \phi_1 + ... + \beta_m \phi_m \tag{56}$$

where $\beta_1, ..., \beta_m$ are iid standard normal Gaussian random variables. Then the effective covariance kernel of g, which we denote k_g, is defined by

$$k_g(x, y) = E[g(x)\overline{g(y)}] = \sum_{j=1}^{m} \sum_{j'=1}^{m} E[\beta_j \beta_{j'}] \phi_j(x) \overline{\phi_{j'}(y)} = \sum_{j=1}^{m} \phi_j(x) \overline{\phi_j(y)} \tag{57}$$

where $\overline{\phi_i(y)}$ denotes the complex conjugate of $\phi_i(y)$ [26]. The weight-space and function-space approaches are equivalent when using the same covariance structure. That is, for any $\tilde{x} \in D$, the posterior mean and variance of the weight-space approach (10) with Gaussian process g is equivalent to the posterior mean and variance of the function-space approach (5) with covariance k_g.

Under these conditions, the $N \times N$ posterior covariance of the function-space approach, K, satisfies

$$K = XX^* \tag{58}$$

and the posterior mean, μ, and variance, s, at any point $\tilde{x} \in \mathbb{R}^d$ are given by

$$\begin{aligned} \mu(\tilde{x}) &= f^t X^* (XX^* + \sigma^2 I)^{-1} y \\ s(\tilde{x}) &= f^* f - (Xf)^* (XX^* + \sigma^2 I)^{-1} (Xf) \end{aligned} \tag{59}$$

where $f \in \mathbb{C}^m$ is defined by $f = [\phi_1(\tilde{x})\, \phi_2(\tilde{x})\, ...\, \phi_m(\tilde{x})]^t$.

The equivalence of the weight-space and function-space approaches can be seen by first multiplying both sides of the function-space linear system, by X^* to obtain

$$(X^* X X^* + \sigma^2 X^*)\alpha = X^* y. \tag{60}$$

Factoring out X^* we have

$$(X^*X + \sigma^2 I)(X^*\alpha) = X^*y. \tag{61}$$

Substituting in the posterior mean of the weight-space approach, we obtain the equivalence of the function-space and weight-space views:

$$\mu_{\text{ws}}(\tilde{x}) = f^t\hat{\beta} = f^t(X^*X + \sigma^2 I)^{-1}X^*y = f^tX^*\alpha = \mu(\tilde{x}). \tag{62}$$

The posterior variance of the function-space approach satisfies

$$\begin{aligned} s(\tilde{x}) &= f^*f - (Xf)^*(XX^* + \sigma^2 I)^{-1}Xf \\ &= f^*(I - X^*(XX^* + \sigma^2 I)^{-1}X)f. \end{aligned}$$

Denoting the singular value decomposition of X by $X = UDV^*$, we have

$$\begin{aligned} s(\tilde{x}) &= f^*(I - VDU^*(U(D^2 + \sigma^2 I)U^*)^{-1}UDV^*f \\ &= f^*(I - V\frac{D^2}{D^2 + \sigma^2}V^*)f \\ &= f^*V\frac{\sigma^2}{D^2 + \sigma^2}V^*f \\ &= f^*(\frac{1}{\sigma^2}X^*X + I)^{-1}f \\ &= s_{\text{ws}}(\tilde{x}). \end{aligned}$$

In the following proposition we provide a formula that relates the solution to the weight-space linear system to that of the function-space system.

Proposition A.1. *Let $\alpha \in \mathbb{R}^N$ satisfy*

$$(XX^* + \sigma^2 I)\alpha = y \tag{63}$$

for some $N \times m$ matrix X, $\sigma^2 > 0$ and $y \in \mathbb{R}^N$. Let $\hat{\beta} \in \mathbb{R}^m$ satisfy

$$(X^*X + \sigma^2 I)\hat{\beta} = X^*y. \tag{64}$$

Then

$$\alpha = \frac{1}{\sigma^2}(y - X\hat{\beta}) \tag{65}$$

Proof. Substituting (65) into equation (63), we observe that equation (65) holds if and only if

$$(XX^* + \sigma^2 I)(y - X\hat{\beta}) - \sigma^2 y = 0. \tag{66}$$

Expanding the left side of equation (66), we obtain

$$\begin{aligned} (XX^* + \sigma^2 I)(y - X\hat{\beta}) - \sigma^2 y &= XX^*y + \sigma^2 y - XX^*X\hat{\beta} - \sigma^2 X\hat{\beta} - \sigma^2 y \\ &= X(X^*y - X^*X\hat{\beta} - \sigma^2\hat{\beta}) \qquad (67) \\ &= X(X^*y - (X^*X + \sigma^2 I)\hat{\beta}). \end{aligned}$$

Equation (66) follows immediately from combining (64) and (67). □

B Discretization error estimates for the Matérn kernel

In this section we provide proofs for the aliasing error and truncation error estimates for the Matérn kernel. Recall that the Matérn kernel and its Fourier transform are given by

$$\begin{aligned} C_\nu(x,\ell) &= \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{|x|}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}|x|}{\ell}\right), \\ \hat{C}_\nu(\xi,\ell) &= \hat{c}_{d,\nu}\left(\frac{\ell}{\sqrt{2\nu}}\right)^d \left(2\nu + |2\pi\ell\xi|^2\right)^{-\nu-d/2}, \end{aligned} \tag{68}$$

where $\hat{c}_{d,\nu}$ is given by

$$\hat{c}_{d,\nu} = \frac{2^d\pi^{d/2}(2\nu)^\nu\Gamma(\nu+d/2)}{\Gamma(\nu)}. \tag{69}$$

In order to prove the estimate for the aliasing error we state some properties of the modified Bessel function, see [13] for example. For $z > 0$, and fixed ν, the modified Bessel functions $K_\nu(z)$ are monotonically decreasing and positive. For fixed z, the modified Bessel functions are monotonically increasing in ν, i.e. $K_\nu(z) \leq K_\mu(z)$ for $\mu \geq \nu$. Moreover,

$$\frac{d}{dz}(z^\nu K_\nu(z)) = -z^\nu K_{\nu-1}(z) \tag{70}$$

Note that the positivity of $K_{\nu-1}(z)$ implies that $z^\nu K_\nu(z)$ is also a monotonically decreasing function of z. The monotonicity properties and the positivity of K_ν also imply that

$$\frac{1}{K_\nu(z)}\frac{d}{dz}(z^\nu K_\nu(z)) \leq -1. \tag{71}$$

Integrating the equation in z, we get

$$\frac{z^\nu K_\nu(z)}{w^\nu K_\nu(w)} \leq e^{-(z-w)}, \quad 0 \leq w \leq z. \tag{72}$$

In the following lemma we prove the estimate for the aliasing error.

Lemma 3 (Matérn kernel aliasing error)**.** *Let the Matérn kernel parameters be $\nu \geq 1/2$, $\ell > 0$. Then the aliasing error with equispaced Fourier parameter $h > 1/\sqrt{d}$ is given by*

$$\left|\sum_{\substack{j\in\mathbb{Z}^d \\ j\neq 0}} C_\nu\left(x+\frac{j}{h},\ell\right)\right| \leq \alpha(d,\nu)C_0\exp\left(-\frac{\sqrt{2\nu}}{\ell}\left(\frac{1}{h}-|x|\right)\right). \tag{73}$$

Proof. First, we note that the sum over $j \in Z^d \setminus 0$, is bounded by $2d$ half spaces of the form $n \geq 1, q \in Z^{d-1}$. Owing to the radial symmetry of the kernel all of those half spaces can be bounded using the same estimate. Moreover, since the $C_\nu(x,\ell)$ are positive functions, we have

$$\left|\sum_{\substack{j\in\mathbb{Z}^d \\ j\neq 0}} C_\nu\left(x+\frac{j}{h},\ell\right)\right| \leq 2d\sum_{n\geq 1}\sum_{q\in Z^{d-1}} C_\nu\left(x+\frac{(n,q)}{h},\ell\right). \tag{74}$$

Since $x \in [-1,1]^d$, and $h > 1\sqrt{d}$, using the monotonicity of $z^\nu K_\nu(z)$, and property (72), we get

$$\left| \sum_{\substack{j\in\mathbb{Z}^d \\ j\neq 0}} C_\nu\left(x + \frac{j}{h}, \ell\right) \right| \leq 2dC_0 \exp\left(-\frac{\sqrt{2\nu}}{\ell}\left(\frac{1}{h} - |x|\right)\right) \sum_{n\geq 1}\sum_{q\in Z^{d-1}} \exp\left(-\frac{\sqrt{2\nu}}{h\ell}\left(\sqrt{n^2+|q|^2}-1\right)\right). \tag{75}$$

The sum over the lattice can be bounded by 1 on $\sqrt{n^2+|q|^2} \leq \sqrt{d}$ for example, and using the estimate $\sqrt{n^2+|q|^2} \leq \frac{n}{\sqrt{d}} + \sum_{j=1}^{d-1} |q_j|/\sqrt{d}$ on the rest, with the understanding that at least one of the indices is greater than $\sqrt{d}$ which allows one to bound the lattice sum independent of h. In particular

$$\begin{aligned} \sum_{n\geq 1}\sum_{q\in Z^{d-1}} \exp\left(-\frac{\sqrt{2\nu}}{h\ell}\left(\sqrt{n^2+|q|^2}-1\right)\right) &\leq V_d(\lceil\sqrt{d}\rceil) + \\ 2d\sum_{n=\lceil\sqrt{d}\rceil}\sum_{q=Z^{d-1}} &\exp\left(-\frac{\sqrt{2\nu}}{h\ell}\left(\left(\frac{n}{\sqrt{d}}-1\right)\right)+\sum_{m=1}^{d-1}\frac{q_m}{\sqrt{d}}\right)\right) \\ &\leq V_d(\lceil\sqrt{d}\rceil) + 2d\left(\frac{1}{1-\exp\left(-\frac{\sqrt{2\nu}}{h\ell}\right)}\right)^d \\ &\leq V_d(\lceil\sqrt{d}\rceil) + 2d\left(\frac{1}{1-\exp\left(-\frac{\sqrt{2\nu d}}{\ell}\right)}\right)^d. \end{aligned} \tag{76}$$

Here $V_d(R)$ is the volume of the ball of radius R centered at the origin, and the last equality follows from the assumption that $h < \frac{1}{\sqrt{d}}$. □

In order to prove the estimate for the truncation error, we need the following lemma bounding lattice sums

Lemma 4.

$$I(d,\nu,m) = \sum_{n>m}\sum_{q\in Z^{d-1}} \left(n^2+|q|^2\right)^{-\nu-d/2} \leq \beta(d,\nu)\frac{1}{m^{-2\nu}} \tag{77}$$

with prefactor

$$\beta(d,\nu) = \begin{cases} \frac{1}{2\nu}, & d=1 \\ \left(4+\frac{2}{2\nu+d-1}\right)\beta(d-1,\nu) & d>1 \end{cases} \tag{78}$$

Proof. Let $m \geq 1$ (this will hold throughout the following) then for $d=1$,

$$\sum_{n>m} n^{-2\nu-1} \leq \int_m^\infty y^{-2\nu-1}dy = \frac{m^{-2\nu}}{2\nu} \tag{79}$$

where monotonic decrease was used to bound the sum by an integral.

For higher d,

$$\sum_{n>m}\sum_{q\in Z^{d-1}}\left(n^2+|q|^2\right)^{-\nu-d/2}=\sum_{n>m}\sum_{w\in Z^{d-2}}\sum_{q\in Z}\left(n^2+|w|^2+q^2\right)^{-\nu-d/2} \tag{80}$$

We split the innermost sum into $q \le \lceil\sqrt{n^2+|w|^2}\rceil$, where $\lceil x\rceil$ denotes the smallest integer not less than x, which contains at most $2(\sqrt{n^2+|w|^2}+1)+1 < 4\sqrt{n^2+|w|^2}$ terms, each bounded by the constant $(n^2+|w|^2)^{-\nu-d/2}$. The two-tailed remainder of this sum is bounded by $2\int_{\sqrt{n^2+|w|^2}} y^{-2\nu-d}dy = (2\nu+d-1)^{-1}(n^2+|w|^2)^{-\nu-(d-1)/2}$. Combining both of these estimates, we get

$$\begin{aligned} I(d,\nu,m) &= \sum_{n>m}\sum_{w\in Z^{d-2}}\sum_{q\in Z}\left(n^2+|w|^2+q^2\right)^{-\nu-d/2}\ , \\ &\le \left(4+\frac{2}{2\nu+d-1}\right)\sum_{n>m}\sum_{w\in Z^{d-2}}\left(n^2+|w|^2\right)^{-\nu-(d-1)/2}\ , \\ &= \left(4+\frac{2}{2\nu+d-1}\right)I(d-1,\nu,m) \end{aligned} \tag{81}$$

Recursing down in d, we get the desired result. □

In the following lemma we prove the estimate for the truncation error.

Lemma 5 (Matérn truncation error bound)**.** *Let the Matérn kernel parameters be $\nu \ge 1/2$, $\ell > 0$. Then the truncation error with equispaced Fourier parameters $h > 0$ and $m \ge$ is given by*

$$\begin{aligned} \left| h^d \sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}} \hat{C}_\nu(jh,\ell)e^{2\pi hi\langle j,x\rangle}\right|_{L^\infty[-1,1]^d} &\le \alpha_\infty(d,\nu)\left(\frac{\sqrt{2\nu}}{h\ell m}\right)^{2\nu}\ , \\ \left| h^d \sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}} \hat{C}_\nu(jh,\ell)e^{2\pi hi\langle j,x\rangle}\right|_{L^2[-1,1]^d} &\le \frac{\alpha_2(d,\nu)}{\sqrt{m}^d}\left(\frac{\sqrt{2\nu}}{h\ell m}\right)^{2\nu}\ , \end{aligned} \tag{82}$$

Proof. For the L^∞ estimate, dropping the phase and noting that $\hat{k}$ is always positive gives the simple uniform bound

$$\begin{aligned} \left| h^d \sum_{\substack{j\in\mathbb{Z}^d\\ j\notin I_m}} \hat{C}_\nu(jh,\ell)e^{2\pi hi\langle j,x\rangle}\right| &\le h^d\sum_{j\in\mathbb{Z}^d\backslash I_m}\hat{C}_\nu(jh,\ell) = \hat{c}_{d,\nu}(h\ell)^d\sum_{j\in\mathbb{Z}^d\backslash I_m}(2\nu+|2\pi\ell hj|^2)^{-\nu-d/2} \\ &\le \hat{c}_{d,\nu}(h\ell)^d\sum_{j\in\mathbb{Z}^d\backslash I_m}|2\pi\ell hj|^{-2\nu-d} \\ &\le \frac{\hat{c}_{d,\nu}}{(2\pi)^{2\nu+d}}\frac{1}{(\ell h)^{2\nu}}I(d,\nu,m)\ . \end{aligned} \tag{83}$$

The last inequality follows from the noting that the sum over $j \in Z^d \backslash I_m$ is bounded by $2d$ half spaces of the form $n > m, q \in Z^{d-1}$.

The derivation of the L^2 estimate follows in a similar manner. □

Remark B.1. *In most of the estimates above, the constants are not optimal, in particular losing (asymptotically for large m) a factor of d due to double- or triple-counting most terms in overlapping half-spaces. There is also some loss due to the sum splitting. We believe that improving these would require awkward geometric work that would not add much to the results.*

References

[1] S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil. Fast Direct Methods for Gaussian Processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2):252–265, 2016.

[2] A. H. Barnett, J. Magland, and L. af Klinteberg. A parallel nonuniform fast fourier transform library based on an "exponential of semicircle" kernel. *SIAM Journal on Scientific Computing*, 41(5):C479–C504, 2019.

[3] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104:136403, Apr 2010.

[4] J. Chen and M. Stein. Linear-cost covariance functions for gaussian random fields. *Journal of the American Statistical Association*, 11 2017.

[5] N. Cressie. *Statistics for Spatial Data, Revised Edition.* Wiley-Interscience, Hoboken, NJ, 2015.

[6] N. Cressie. Mission co2ntrol: A statistical scientist's role in remote sensing of atmospheric carbon dioxide. *Journal of the American Statistical Association*, 113(521):152–168, 2018.

[7] G. Dahlquist and A. Bjork. *Numerical Methods.* Dover, Mineola, NY, 1974.

[8] A. Delaney and Y. Bresler. A fast and accurate fourier algorithm for iterative parallel-beam tomography. *IEEE Transactions on Image Processing*, 5(5):740–753, 1996.

[9] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14(6):1368–1393, 1993.

[10] D. Foreman-Mackey, E. Agol, S. Ambikasaran, and R. Angus. Fast and Scalable Gaussian Process Modeling with Applications to Astronomical Time Series. *The Astronomical Journal*, 154(6), 2017.

[11] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[12] G. H. Golub and C. F. van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[13] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Academic press, 2014.

[14] L. Greengard, J.-Y. Lee, and S. Inati. The fast sinc transform and image reconstruction from nonuniform samples in k-space. *Communications in Applied Mathematics and Computational Science*, 1(1):121 – 131, 2006.

[15] P. Greengard. Efficient Fourier representations of families of Gaussian processes. *arXiv*, stat.CO/2109.14081, 2021.

[16] J. Hensman, N. Durrande, and A. Solin. Variational Fourier Features for Gaussian Processes. *J. Mach. Learn. Res.*, 18(1):5537–5588, 2017.

[17] K. L. Ho. Flam: Fast linear algebra in matlab - algorithms for hierarchical matrices. *Journal of Open Source Software*, 5(51):1906, 2020.

[18] V. Minden, A. Damle, K. L. Ho, and L. Ying. Fast Spatial Gaussian Process Maximum Likelihood Estimation via Skeletonization Factorizations. *Multiscale Modeling and Simulation*, 15(4), 2017.

[19] OCO-2 Science Team/Michael Gunson, Annmarie Eldering. OCO-2 Level 2 bias-corrected XCO2 and other select fields from the full-physics retrieval aggregated as daily files, 2018. Retrospective processing V9r, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: 2021-11-19.

[20] C. J. Paciorek. Bayesian Smoothing with Gaussian Processes Using Fourier Basis Functions in the spectralGP Package. *Journal of Statistical Software*, 19(2):1–38, 2007.

[21] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

[22] C. E. Rasmussen and C. L. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[23] G. Riutort-Mayol, P.-C. Bürkner, M. R. Andersen, A. Solin, and A. Vehtari. Practical hilbert space approximate bayesian gaussian processes for probabilistic programming, 2020.

[24] D. Sanz-Alonso and R. Yang. Finite Element Representations of Gaussian Processes: Balancing Numerical and Statistical Accuracy. *arXiv*, stat.co/2109.02777, 2021.

[25] M. L. Stein. Bounds on the Efficiency of Linear Predictions Using an Incorrect Covariance Function. *The Annals of Statistics*, 18(3):1116 – 1138, 1990.

[26] M. L. Stein. *Interpolation of Spatial Data, Some Theory for Kriging.* Springer, New York, NY, 1999.

[27] A. W. van der Vaart and J. H. van Zanten. Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, 36(3):1435 – 1463, 2008.

[28] L. Wang, Y. Shkolnisky, and A. Singer. A Fourier-based Approach for Iterative 3D Reconstruction from Cryo-EM Images. *arXiv*, math.na/1307.5824, 2013.

[29] J. Wenger, G. Pleiss, P. Hennig, J. P. Cunningham, and J. R. Gardner. Reducing the variance of gaussian process hyperparameter optimization with preconditioning. *CoRR*, abs/2107.00243, 2021.

[30] A. G. Wilson and H. Nickisch. Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1775–1784. JMLR.org, 2015.

N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	16	0.002	0.001	0.001	0.003	43	1.5×10^{-8}
10^5	16	0.016	0.001	0.001	0.017	52	1.1×10^{-7}
10^6	16	0.076	0.001	0.001	0.077	69	4.9×10^{-7}
10^7	16	0.739	0.001	0.001	0.741	85	1.4×10^{-6}

(a) 1-*dimension, Squared-exponential kernel,* $\ell = 0.1$.

N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	3791	0.005	0.248	0.001	0.254	202	2.0×10^{-3}
10^5	3791	0.010	0.540	0.001	0.552	456	5.3×10^{-3}
10^6	3791	0.066	0.874	0.001	0.941	735	9.4×10^{-3}
10^7	3791	0.697	1.279	0.001	1.977	1053	6.1×10^{-3}

(b) 1-*dimension, Matérn 1/2 kernel,* $\ell = 0.1$

N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	17	0.007	0.106	0.007	0.120	341	1.6×10^{-8}
10^5	17	0.023	0.269	0.014	0.306	840	1.9×10^{-8}
10^6	17	0.222	0.618	0.087	0.926	1913	6.2×10^{-8}
10^7	17	1.924	1.200	1.001	4.126	3570	1.2×10^{-6}

(c) 2-*dimensions, Squared-exponential kernel,* $\ell = 0.1$.

N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	108	0.033	0.824	0.004	0.861	117	1.5×10^{-2}
10^5	108	0.047	1.519	0.005	1.571	221	4.1×10^{-2}
10^6	108	0.164	1.669	0.007	1.840	210	5.4×10^{-2}
10^7	108	1.525	1.038	0.014	2.577	132	3.1×10^{-2}

(d) 2-*dimensions, Matérn 1/2 kernel,* $\ell = 0.1$

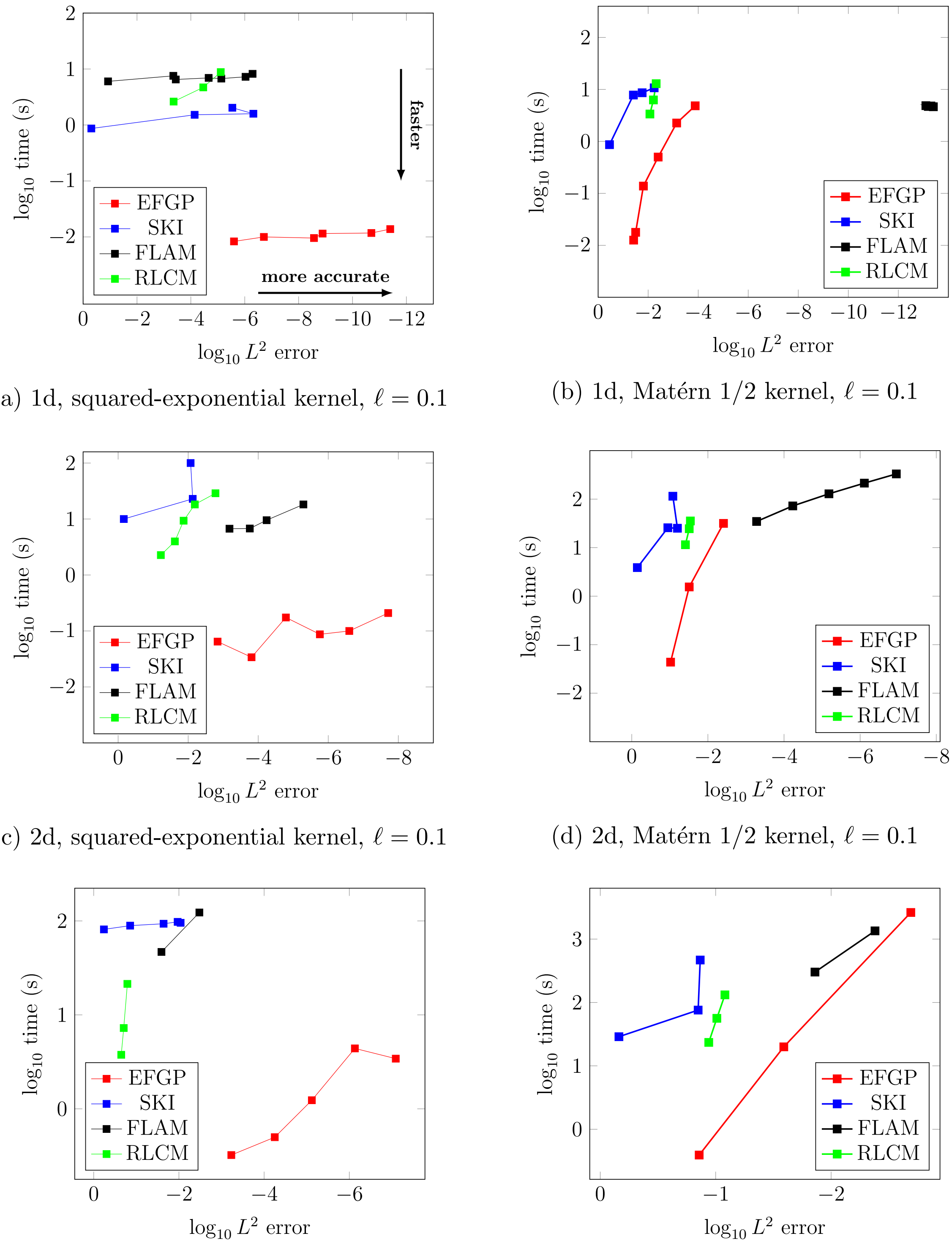
N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	12	0.029	0.250	0.097	0.376	152	6.0×10^{-5}
10^5	12	0.051	0.779	0.123	0.953	477	9.9×10^{-5}
10^6	12	0.318	2.187	0.235	2.741	1379	2.1×10^{-4}
10^7	12	2.561	5.420	1.335	9.316	3395	4.0×10^{-4}

(e) 3-*dimensions, Squared-exponential kernel,* $\ell = 0.1$.

N	m	precomp (s)	CG (s)	mean (s)	total (s)	CG iters	RMSE
10^4	37	0.397	9.733	0.123	10.252	61	3.5×10^{-2}
10^5	37	0.529	33.109	0.172	33.810	180	3.5×10^{-2}
10^6	37	0.613	55.365	0.176	56.154	302	7.0×10^{-2}
10^7	37	2.422	31.601	0.145	34.168	213	4.4×10^{-2}

(f) 3-*dimensions, Matérn 1/2 kernel,* $\ell = 0.1$

Table 2: *Timing and accuracy for EFGP with simulated data.*



(a) 1d, squared-exponential kernel, $\ell = 0.1$

(b) 1d, Matérn 1/2 kernel, $\ell = 0.1$

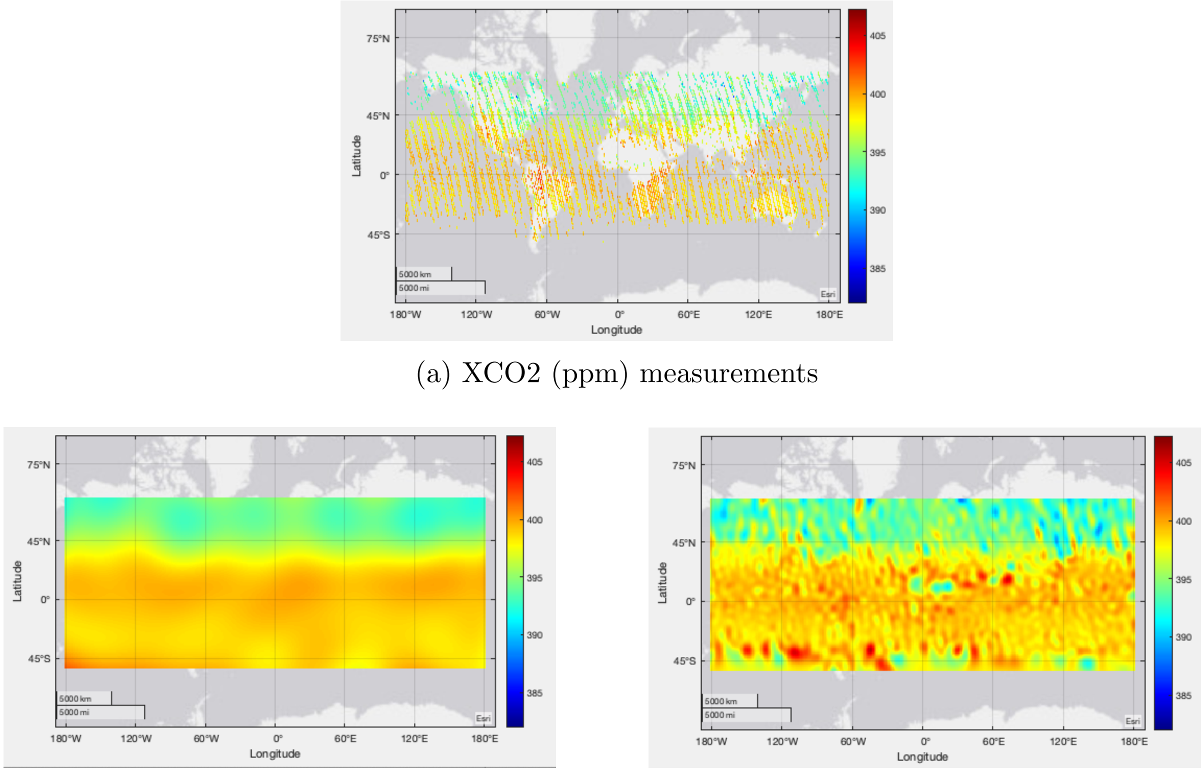
(c) 2d, squared-exponential kernel, $\ell = 0.1$

(d) 2d, Matérn 1/2 kernel, $\ell = 0.1$

(e) 3d, squared-exponential kernel, $\ell = 0.1$

(f) 3d, Matérn 1/2 kernel, $\ell = 0.1$

Figure 3: Compute time to achieve various levels of accuracy for Gaussian process regression in 1, 2, and 3 dimensions for four algorithms. For all problems $N = 10^5$ simulated data points were used and $\sigma^2 = 0.25$.



(a) XCO2 (ppm) measurements

(b) Squared-exponential kernel, $\ell = 50$, L^2-error of 0.001 and 0.5 seconds total run time for regression and evaluation of posterior mean.

(c) Squared-exponential kernel, $\ell = 5$, L^2-error of 0.0005 and 6.8 seconds total run time for regression and evaluation of posterior mean.

Figure 4: 2-dimensional Gaussian process regression on XCO2 data with $N \approx 1.4 \times 10^6$ using squared-exponential kernel with two different time scales and prefactor (or output variance) of 25. The data was demeaned and we used residual variance, $\sigma^2 = 1$.