Mercer kernels and integrated variance experimental design: connections between Gaussian process regression and polynomial approximation

Alex Gorodetsky[*] and Youssef Marzouk[*]

Abstract. This paper examines experimental design procedures used to develop surrogates of computational models, exploring the interplay between experimental designs and approximation algorithms. We focus on two widely used approximation approaches, Gaussian process (GP) regression and non-intrusive polynomial approximation. First, we introduce algorithms for minimizing a posterior integrated variance (IVAR) design criterion for GP regression. Our formulation treats design as a continuous optimization problem that can be solved with gradient-based methods on complex input domains, without resorting to greedy approximations. We show that minimizing IVAR in this way yields point sets with good interpolation properties, and that it enables more accurate GP regression than designs based on entropy minimization or mutual information maximization. Second, using a Mercer kernel/eigenfunction perspective on GP regression, we identify conditions under which GP regression coincides with pseudospectral polynomial approximation. Departures from these conditions can be understood as changes either to the kernel or to the experimental design itself. We then show how IVAR-optimal designs, while sacrificing discrete orthogonality of the kernel eigenfunctions, can yield lower approximation error than orthogonalizing point sets. Finally, we compare the performance of adaptive Gaussian process regression and adaptive pseudospectral approximation for several classes of target functions, identifying features that are favorable to the GP + IVAR approach.

Key words. Gaussian process regression, experimental design, computer experiments, approximation theory, polynomial approximation, kernel interpolation, uncertainty quantification

1. Introduction.

Computational simulations are essential for design, optimization, uncertainty quantification, and inference in complex systems. Yet these tasks typically require a large number of simulations over a range of parameter values, which can be computationally prohibitive. One method of mitigating this computational expense is to construct surrogates or "emulators" that replace the simulation in the relevant analyses. Because many computational simulations are available only as black-box or legacy codes, surrogates often must be constructed through a limited number of simulations at particular parameter values $\{x_i\}$. These simulations are sometimes called "computer experiments" [51, 52] and choosing these parameter values is a question of experimental design.

Surrogate construction can be viewed as a function approximation problem in which one attempts to approximate an input-output relationship $f(x)$ induced by the expensive simulation. One can do so deterministically, i.e., obtaining a single approximation $\hat{f}$, or probabilistically, i.e., obtaining a distribution over possible functions $\hat{f} \sim \mathcal{F}$, where $\mathcal{F}$ is a probability distribution on a suitable function space. In either case, three decisions must be made. First, one must choose an *approximation space* that contains candidate surrogate functions; second, one must select a set of parameter values or *experiments* at which to simulate the system; and finally, one must choose an *algorithm* to convert the simulation results into a particular function $\hat{f}$ or distribution $\mathcal{F}$. Examples of approximation spaces include those spanned by polynomials of a certain degree, or by radial basis functions and

[*] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. {goroda,ymarz}@mit.edu.

other kernels. Possible experiments include designs produced by Monte Carlo sampling, obtained via optimization of information-based design criteria such as entropy and mutual information, or based on numerical quadrature rules. Finally, examples of algorithms include linear regression and pseudospectral approximation. These three decisions are usually not independent; for example, pseudospectral approximation requires experimental design procedures that preserve orthogonality of the relevant basis functions.

In this paper, we analyze two of the most commonly used surrogate construction approaches in uncertainty quantification: Gaussian process regression (GPR), which has seen much development in the statistics community, and pseudospectral approximation (PSA), which is widely used in applied mathematics and engineering. Both of these methods are routinely applied for similar purposes—replacing computationally expensive models with cheap-to-evaluate surrogates for complex analysis tasks. One of our primary goals is to compare these two approaches by analyzing the approximation spaces, experiments, and algorithms that they employ. The PSA method comes equipped with an experimental design procedure and an algorithm intended to produce approximations that are accurate in an L^2 sense. The GPR methodology is more flexible in that it does not come pre-equipped with an experimental design procedure. In order to compare these two methods, we will employ an experimental design criterion that also seeks accuracy in an L^2 sense.

Our main contributions are as follows. First, we develop a continuous optimization algorithm, based on sample-average approximation (SAA), to minimize an *integrated posterior variance* (IVAR) design criterion for Gaussian process regression. We compare our algorithm to approaches that maximize other information-based criteria (e.g., entropy or mutual information) by evaluating their computational costs, the properties of the resulting point sets, and the accuracy of the resulting approximations. Second, we provide a theoretical and numerical analysis comparing non-intrusive polynomial approximation—polynomial chaos expansions, in particular pseudospectral polynomial approximation—with Gaussian process regression. While the relative performance of these methods may be a subject of broader debate, here we assess the impact of each of the three surrogate construction ingredients described above. Theoretically, we develop results to describe the difference between the approximations given the same experiments and similar approximation spaces. Numerically, we investigate the performance of the two approaches given similar approximation spaces but different experiments, chosen to be optimal for each.

The IVAR objective, which is equivalent to the IMSE criterion [51, 9], can be minimized either over a finite (and hence discrete) design space or over a continuous design space. In the discrete case, the criterion is sometimes called the ALC (“active learning Cohn”) [6] objective. Minimizing the ALC criterion involves sequentially adding experiments, chosen one at a time from a discrete and finite set of candidates, to minimize a weighted average of the predictive variance. ALC has also been investigated in the context of determining *local* designs for large-scale computer experiments [28]. ALC is often compared to other discrete design space criteria, namely ALM (“active learning MacKay”) and mutual information (MI) [34]. Minimizing the ALM criterion involves sequentially adding experiments at locations where the local predictive variance is maximized. Compared with ALM, the ALC criterion considers the effect of each experiment on the entire domain and therefore yields better designs; ALC is more expensive, however, because it requires a new variance computation for each potential design. The MI criterion sequentially seeks points that maximize the expected information gain at locations not yet chosen. MI design requires a good candidate set, which may be difficult to obtain for input domains with complex geometries, though strides have been made in this direction [3]. It also remains computationally expensive, with a complexity that grows cubically with the size of the candidate set.

Instead of dealing with the combinatorial optimization issues associated with discrete design spaces, we will pursue optimization in a continuous space. Our approach is similar to [50, 51] in that we use gradient-based methods to search for optimal designs. But we will explore opportunities presented by solving the full optimization problem, without ad hoc simplifications of the design space. We will employ a sample-average approximation (SAA) that helps deal with complex domain geometries by creating an implicit barrier function for the constraints. Benefits of this approach include generating *batches* of experiments with lower computational complexity than sequentially minimizing the ALC criterion; eliminating undesirable boundary clustering effects associated with radial basis function kernels, which plague ALM designs [47, 34]; and achieving better approximation performance than either ALM or MI. Finally, because we perform design on a continuous space of candidate points, it becomes more natural to analyze the stability and accuracy of approximation with IVAR-optimal designs from the perspective of numerical analysis. For example, in Section 4 we will show that our algorithm generates point sets with good interpolatory properties, as measured by their Lebesgue constants. Finding these point sets via a statistical criterion raises interesting links with previous work in Bayesian numerical analysis [43, 16], particularly the average-case quadrature of [39, 45]. A continuous design procedure also facilitates more cleanly comparing GPR and pseudospectral approximation.

Our comparison of GPR with pseudospectral approximation has two elements. First, we use Mercer's theorem to rewrite GP regression in terms of orthogonal eigenfunctions of the kernel, such that when these eigenfunctions contain the finite basis for a pseudospectral approximation, one can directly assess the difference between the two approximations. If experimental design is based on an orthogonalizing quadrature rule, the difference between the GP mean and the pseudospectral approximation is due to eigenfunctions of the GP kernel which are not in this finite basis. These leftover eigenfunctions also account entirely for the integrated variance of the GP. Furthermore, we illustrate through numerical examples that experiments achieving optimal IVAR for these GPs can differ qualitatively from standard quadrature rules, and that when the IVAR criterion is then used to select experiments for GP regression, GPR can outperform pseudospectral approximation in some settings. Second, we consider adaptive procedures for GPR that interleave IVAR-based experimental design with adaptation of the kernel hyperparameters. For test problems of moderate dimension, we find that GP approximations constructed in this way can again outperform certain adaptive pseudospectral approaches [7].

This paper is organized into two parts. The first part reviews Gaussian process regression (Section 2), introduces the IVAR criterion and its optimization (Section 3), and describes numerical comparisons of IVAR with other experimental design procedures for GP regression (Section 4). The second part provides a brief background on pseudospectral approximation (Section 5.1), then describes theoretical (Section 5.2) and numerical (Section 5.4) comparisons between PSA and GP regression.

2. Gaussian process regression.

Gaussian process (GP) regression can be interpreted as a Bayesian method for function approximation [44, 48], providing a posterior probability distribution over functions. The method begins with a Gaussian process prior, specified via a prior mean function $m_0(x)$ and a covariance kernel $K(x, x')$ that is positive semidefinite and bounded. Suppose that N simulations of the function $f : \mathbb{R}^d \to \mathbb{R}$ are performed at parameter values $\boldsymbol{x} := [x_1, \ldots x_N]$, $x_i \in \mathbb{R}^d$, yielding noisy function evaluations $\hat{\boldsymbol{y}} := [\hat{y}_1, \ldots \hat{y}_N]$, where $\hat{y}_i = f(x_i) + \xi_i$ for $i = 1, \ldots, N$ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$. The resulting posterior distribution is

$$\tilde{f}|\boldsymbol{x}, \hat{\boldsymbol{y}} \sim \mathcal{N}(m(x), C(x, x')),$$

where the posterior mean is

$$m(x) = m_0(x) + \boldsymbol{\alpha}^T K(\boldsymbol{x}, x), \tag{2.1}$$

and the posterior covariance is

$$C(x, x') = K(x, x') - K(\boldsymbol{x}, x)^T \, \mathbf{R} \, K(\boldsymbol{x}, x'). \tag{2.2}$$

In the notation above, $K(\boldsymbol{x}, x)$ is a (column) vector in $\mathbb{R}^N$ whose ith component is $K(x_i, x)$. The covariance matrix $\mathbf{R}^{-1}$ has elements $\left[\mathbf{R}^{-1}\right]_{ij} = K(x_i, x_j) + \delta_{ij}\sigma^2$. Finally, the ith element of the vector of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$ is $\alpha_i = \mathbf{R}_{[i,:]} (\hat{\boldsymbol{y}} - m_0(\boldsymbol{x}))$.

Gaussian process regression reverts to interpolation when $\sigma^2 = 0$. However, as $N \to \infty$ the covariance matrix $\mathbf{R}^{-1}$ becomes ill-conditioned; a small value for σ^2, called a nugget, is often then introduced to stabilize the procedure [42]. Note also that we have not included inference of the prior mean $m_0(x)$ in the Bayesian formulation above. If the prior mean is described via some parametric model $m_0(x) = \boldsymbol{\beta}^T \boldsymbol{n}(x)$, where $\boldsymbol{n}(x)$ is a vector of basis functions, then Bayesian inference of the coefficients $\boldsymbol{\beta}$ would add terms to the posterior covariance. In practice, however, it is common and quite effective to assume either a zero or non-zero constant term for m_0 and to fix its value (for example, by maximizing the log-marginal likelihood) before performing the GP update; see, e.g., [31, 35, 28]. For simplicity, we fix the prior mean here. Doing so will also help focus the comparison of nonparametric GP regression with parametric PSA in Section 5 on its essential aspects.

2.1. Reproducing kernels and Mercer's theorem.

Many elements of this work rely on the interpretation of the covariance kernel through its eigenfunctions, and to this end we recall the properties of a Mercer kernel. Let the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be defined on a first-countable [17] space $\mathcal{X} \subseteq \mathbb{R}^d$ endowed with a strictly positive Borel measure μ. Suppose that the kernel is continuous, positive semi-definite,

$$\int_{\mathcal{X}\times\mathcal{X}} K(x, x') g(x) g(x') d\mu(x) d\mu(x') \geq 0, \ \forall g \in L^2_\mu(\mathcal{X}), \tag{2.3}$$

and in $L^1(\mathcal{X}, \mu)$

$$\int_{\mathcal{X}} |K(x, x)| \, d\mu(x) < \infty. \tag{2.4}$$

Additionally we define the integral operator $T_K : L^2_\mu(\mathcal{X}) \to L^2_\mu(\mathcal{X})$ such that $T_K f = \int_{\mathcal{X}} K(x, x') f(x') d\mu(x')$. This operator has a countable system of eigenvalues λ_j that are non-negative and that satisfy

$$\sum_{j=1}^{\infty} \lambda_j^2 < \infty.$$

The eigenfunctions ϕ_i of T_K form an orthonormal basis of $L^2_\mu(\mathcal{X})$. These eigenfunctions and eigenvalues can be used to define the reproducing kernel Hilbert space (RKHS) associated with the kernel [48]. Mercer's theorem lets us represent K as a convergent series in terms of this eigensystem.

Theorem 2.1 (Mercer). *[38, 22, 4] Let $\mathcal{X} \subset \mathbb{R}$ be first countable or locally compact, μ a strictly positive Borel measure on $\mathcal{X}$, and K a continuous function on $\mathcal{X} \times \mathcal{X}$ satisfying (2.3) and (2.4). Then*

$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x'), \tag{2.5}$$

where the series converges absolutely for each pair $(x, x') \in \mathcal{X} \times \mathcal{X}$ and uniformly on each compact subset of $\mathcal{X}$.

When comparing GP regression with pseudospectral approximation in later sections of this paper, we will also use the notion of a truncated kernel. These are kernels for which $\lambda_{i>\ell} = 0$, and for which we can equivalently write (2.5) as

$$K(x, x') = \sum_{i=1}^{\ell} \lambda_i \phi_i(x) \phi_i(x').$$

One common example of a truncated kernel is a polynomial kernel, e.g., $K(x, x') = (x^T x' + 1)^p$, where p is a positive integer.

We can use Mercer's theorem to write the integrated posterior variance of the Gaussian process in terms of the eigensystem (λ_i, ϕ_i). The posterior variance at any point in the domain is $c(x) := C(x, x)$. The integrated variance then becomes

$$\begin{aligned}
\int c(x) d\mu(x) &= \int \Big(K(x, x) - K(\boldsymbol{x}, x)^T \mathbf{R}\, K(\boldsymbol{x}, x) \Big)\, d\mu(x) \\
&= \int K(x, x) d\mu(x) - \int \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \lambda_i \lambda_j \boldsymbol{\phi}_i^T \mathbf{R} \boldsymbol{\phi}_j \phi_i(x) \phi_j(x) d\mu(x) \\
&= \sum_{i=1}^{\infty} \lambda_i - \sum_{i=1}^{\infty} \lambda_i^2 \boldsymbol{\phi}_i^T \mathbf{R} \boldsymbol{\phi}_i, \qquad (2.6)
\end{aligned}$$

where $\boldsymbol{\phi}_i := [\phi_i(x_1), \dots, \phi_i(x_N)]^T$. The first term is the integrated variance of the prior. The second term, which is always non-negative, reflects reduction in the integrated prior variance due to conditioning on the data. We will now examine how experimental design procedures can use this integrated variance as an optimization objective.

3. Integrated variance experimental design.

We will design experiments to minimize the integrated posterior variance (IVAR) of the Gaussian process. This choice is motivated by inferential considerations. As opposed to design procedures based on Latin hypercube sampling, quasi-Monte Carlo, quadrature, or other "lattice" designs, the present design strategy directly aims to minimize a measure of the uncertainty associated with the approximation. One advantage of this approach is that it can be used to design experiments on a wide variety of input domains, not just domains with tensor-product or some other canonical structure. Another advantage of computing and monitoring a measure of uncertainty is that it provides useful feedback about the quality of the approximation; for example, if the data do not yield much reduction in uncertainty, one can adjust the approximation space or some other aspect of the surrogate construction methodology.

To put the IVAR criterion in context, we note that it is equivalent to an expected integrated squared error of the posterior mean. First consider the posterior expectation of the squared error in function values, integrated over the parameter space,

$$\mathbb{E}_{\tilde{f}|\boldsymbol{x},\hat{\boldsymbol{y}}}\left[\int \left(\tilde{f}(x) - f(x)\right)^2 d\mu(x)\right],$$

where $\tilde{f}$ can be thought of as a posterior realization of the Gaussian process and μ is the measure on the parameter space $\mathcal{X}$. We can divide this quantity into two terms,

$$\mathbb{E}_{\tilde{f}|\boldsymbol{x},\hat{\boldsymbol{y}}}\left[\int \left(\tilde{f}(x) - f(x)\right)^2 d\mu(x)\right] = \int \left(m(x|\boldsymbol{x},\hat{\boldsymbol{y}}) - f(x)\right)^2 d\mu(x) + \int c(x|\boldsymbol{x}) d\mu(x), \tag{3.1}$$

where the first term on the right-hand side is the integrated squared error of the posterior mean and the second term is the integrated posterior variance, i.e., the IVAR. We have explicitly indicated all the conditioning on the right-hand side of (3.1). Note that the second term is independent of the sampled values $\hat{\boldsymbol{y}}$ of the function f.[1] Computing the first term, on the other hand, requires the ability to evaluate f. Directly using this term in a design criterion would defeat the purpose of experimental design, which is motivated by the desire to evaluate f sparingly. Instead, we can consider the expectation of this squared error over the joint distribution of f and $\hat{\boldsymbol{y}}$, for a fixed design $\boldsymbol{x}$. We assume that f is drawn from the prior $\mathcal{N}(m_0(x), K(x,x'))$, and therefore this expectation becomes the Bayes risk of the posterior mean under squared error loss, which is equivalent to the integrated mean squared error (IMSE) criterion proposed by [51]. Some manipulation shows that this Bayes risk is indeed *equal* to the integrated posterior variance, i.e., that after taking the expectation over f and $\hat{\boldsymbol{y}}$, the two terms on the right side of (3.1) are the same. For further details, see, e.g., [53, p. 92].

An important additional interpretation is as follows. The variance integral above is actually the *trace* of the posterior covariance operator $C(x,x')$, and therefore IVAR minimization can be understood as A-optimal design [1], though in an infinite-dimensional function space setting. It is well known that Bayesian A-optimal designs minimize the Bayes risk for linear-Gaussian models [12], which is exactly what we seek to do here.

A different connection to optimal design theory can be made by finding a *finite* parameterization for the Mercer kernel K and converting the problem into one of parametric model fitting and prediction [19, 21, 20, 29]. Such approaches decompose the kernel by computing a truncated Karhunen-Loève expansion [20, 29] of the Gaussian process; alternatively, one can use a polar spectral approximation [60] of the kernel to avoid computing its eigenfunctions. In either case, once the kernel has been decomposed and truncated, one can write the posterior mean of the GP prediction as a linear combination of finitely many

[1] In this section, we keep the prior covariance kernel K fixed. In Section 5.4, we will consider closed-loop adaptive design strategies that learn hyperparameters of the kernel K from evaluations of the function f.

basis functions,

$$m(x) = m_0(x) + \sum_{i=1}^{\ell} \theta_i \phi_i(x). \tag{3.2}$$

In optimal design theory, one may then seek experiments to best learn about the mean structure m_0, the parameters $\theta_1, \ldots, \theta_\ell$, or to minimize some uncertainty in the prediction. These choices correspond to different classical "alphabetic optimality" criteria applied to the corresponding information matrix, e.g., A, D, or G-optimal design.

The IVAR design procedure presented in this paper, however, does not require a finite parameterization of the kernel; instead it can use a closed-form expression for the posterior covariance, with no truncation. A direct comparison between such kernel-based procedures (for example, the Sacks-Ylvisaker approach described in [19]) and parametric optimal design theory is outside the scope of this paper; we refer readers to [19, 61, 41, 20] instead. We will, however, compare our IVAR optimization procedure to other algorithms and design objectives that do not require explicitly finding a finite parameterization of the kernel. Later, when comparing GP regression to another surrogate modeling methodology—pseudospectral polynomial approximation, in Section 5—we will return to the eigenfunction viewpoint of GP regression. In that context, our focus will not be not on algorithms that require explicit access to the eigenfunctions of the kernel, but rather on how the eigenfunction viewpoint exposes the distinct modeling assumptions made by the two methodologies.

3.1. IVAR evaluation and minimization.

For a chosen number of experiments $N \geq 1$ and a fixed prior covariance kernel K, our optimal experimental design is a set of evaluation points $\boldsymbol{x}^* := [x_1^*, \ldots x_N^*]$, $x_i^* \in \mathbb{R}^d$, minimizing the integrated posterior variance:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{U}} \int_{\mathcal{X}} c(x|\boldsymbol{x}) d\mu(x). \tag{3.3}$$

Here $\mathcal{U}$ is the space of all feasible experiments and the posterior variance is specified via (2.2).

While this objective function is similar to that in [57], here we will employ a continuous space $\mathcal{U}$ of possible experiments. Also, we will solve the optimization problem (3.3) both in a non-greedy fashion (finding all N design points simultaneously) and using greedy updates with varying batch sizes. In this section the number of design points N and the prior covariance kernel K will be considered fixed. Later, in Section 5.4, we will consider closed-loop design procedures that alternate between batch minimization of IVAR and updates of the covariance kernel.

3.1.1. Sample-average approximation of IVAR.

One method of minimizing IVAR involves numerically evaluating the objective in (3.3) via quadrature or a quasi-Monte Carlo or Monte Carlo (MC) sampling procedure. Since the variance is in general a smooth function of x, quadrature schemes may work efficiently for low-to-moderate dimensional input spaces, but Monte Carlo will generally work better in higher dimensions; Monte Carlo also offers more flexibility for non-tensor-product domains $\mathcal{X}$. Monte Carlo sampling replaces the integral with the summation

$$\int_{\mathcal{X}} c(x|\boldsymbol{x})\, d\mu(x) \approx \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} c(\hat{x}_i|\boldsymbol{x}) =: \hat{J}^{\text{mc}}(\boldsymbol{x}), \tag{3.4}$$

where N_{mc} is the number of Monte Carlo samples and $\hat{x}_i \sim \mu$. Computing the integrated variance for a set of N experiments then requires an inversion of the covariance matrix, an $\mathcal{O}(N^3)$ operation, and variance evaluations at N_{mc} points, an $\mathcal{O}(N_{mc}N^2)$ operation.

The sample-average approximation (SAA) [58] approach to optimization simply replaces the expectation in the objective (3.3) with a quadrature or Monte Carlo approximation at a *fixed* set of points and minimizes this objective. After one has chosen this set of fixed points, the minimization becomes a constrained deterministic minimization problem. We can use readily available analytical derivatives of the objective in this setting. In particular, given the form of the kernel K, we can directly compute the gradient $\nabla_{\boldsymbol{x}} c(x|\boldsymbol{x})$ from (2.2). The gradient of the SAA objective obtained from Monte Carlo then becomes

$$\nabla_{\boldsymbol{x}} \hat{J}^{mc}(\boldsymbol{x}) = \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} \nabla_{\boldsymbol{x}} c(\hat{x}_i|\boldsymbol{x}),$$

and similarly for quadrature. The gradient of the posterior covariance can be analytically obtained from the underlying kernel. For simplicity, assume that we are designing experiments on a one-dimensional input domain and that our experimental design is $\boldsymbol{x} = [x_1, \ldots x_N]$. Now the gradient for a given $\hat{x}_i$ may be computed as

$$\begin{aligned} \nabla_{\boldsymbol{x}} c(\hat{x}_i|\boldsymbol{x}) &= -\nabla_{\boldsymbol{x}} \left(K(\boldsymbol{x}, \hat{x}_i)^T \mathbf{R} K(\boldsymbol{x}, \hat{x}_i) \right) \\ &= -\nabla_{\boldsymbol{x}} \left(\sum_{j=1}^{N} \sum_{k=1}^{N} K(x_j, \hat{x}_i) \mathbf{R}[j,k] K(x_k, \hat{x}_i) \right) \end{aligned} \tag{3.5}$$

Let us focus on a single element of the gradient:

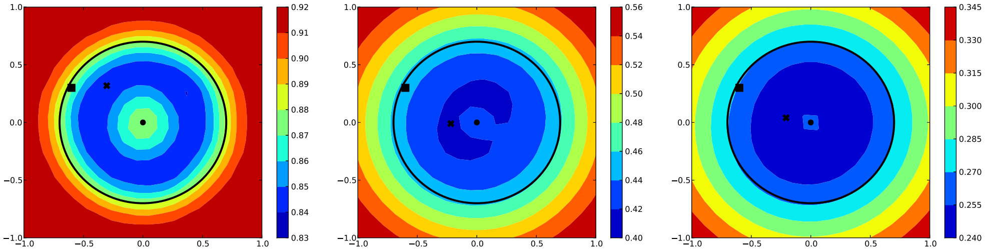
$$\begin{aligned} \frac{\partial c(\hat{x}_i|\boldsymbol{x})}{\partial x_\ell} &= -\left[\sum_{j=1}^{N} \sum_{k=1}^{N} \frac{\partial}{\partial x_\ell} \left(K(x_j, \hat{x}_i) \mathbf{R}[j,k] K(x_k, \hat{x}_i) \right) \right] \\ &= -\sum_{j=1}^{N} \sum_{k=1}^{N} \left(\frac{\partial}{\partial x_\ell} K\left(x_j, \hat{x}_i\right) \right) \mathbf{R}[j,k] K(x_k, \hat{x}_i) - \sum_{j=1}^{N} \sum_{k=1}^{N} K\left(x_j, \hat{x}_i\right) \frac{\partial}{\partial x_\ell} \left(\mathbf{R}[j,k] K(x_k, \hat{x}_i) \right) \\ &= 2 \left(\frac{\partial}{\partial x_\ell} K(x_\ell, \hat{x}_i) \right) \sum_{k=1}^{N} \mathbf{R}[\ell, k] K(x_k, \hat{x}_i) - \sum_{j=1}^{N} \sum_{k=1}^{N} K\left(x_j, \hat{x}_i\right) \frac{\partial}{\partial x_\ell} \left(\mathbf{R}[j,k] \right) K(x_k, \hat{x}_i) \end{aligned}$$

where the second equality follows from the chain rule and the third equality follows from the symmetry of K and $\mathbf{R}$ as well as another application of the chain rule. We are left with terms involving derivatives of $K(x, x')$ and of the elements of $\mathbf{R}$. To evaluate the latter, use the identity $\frac{\partial \mathbf{R}}{\partial x_\ell} = -\mathbf{R} \frac{\partial \mathbf{R}^{-1}}{\partial x_\ell} \mathbf{R}$ and recall that $\mathbf{R}^{-1}[i,j] = K(x_i, x_j) + \delta_{ij}\sigma^2$. Hence this quantity can also be computed from the derivative of K. For instance, if K is a squared exponential kernel $K(x_i, x_j) = \exp\left(-(x_i - x_j)^2/2l^2\right)$ then the derivative is

$\partial K(x_i, x_j)/\partial x_i = -\frac{1}{l^2}(x_i - x_j)K(x_i, x_j)$. An analogous derivation can be performed for a multi-dimensional input space.

The sample-average approximation also automatically provides an implicit "barrier" that helps optimization iterations stay within a constrained input domain. Specifically, since Monte Carlo samples $\hat{x}_i$ are only obtained within the support of the input distribution μ, the optimization objective increases as design points leave the domain and eventually it becomes flat. Looking at (2.2), it is apparent that if the interaction between the SAA samples and a design point is negligible, the data update (second term of the equation) will be close to zero. Then the new experiment will not contribute to a reduction of the variance, and the resulting posterior integrated variance will simply revert to the prior integrated variance. Approaching these flat regions, however, gradients of the objective will naturally drive design points towards the support of μ.

To illustrate this phenomenon, suppose that μ is uniform on a two-dimensional circular domain and that one experiment has already been performed. Figure 1 shows objectives associated with the addition of a new design point, where K is a squared exponential kernel with varying correlation lengths l. We see that when the correlation length is short, $l = 0.2$, then the cost function is flat just outside the circular domain because there are no nearby SAA samples to affect the cost function. As the correlation length increases from left to right, design points further from the domain boundary can provide some information about the function inside the domain. However, the objective still increases as one moves away from the SAA samples $\hat{x}_i$; more uncertainty reduction can be had by choosing a design within the domain. This behaviour encourages the iterates of gradient-based algorithms to stay within the region of interest.



(a) $l = 0.2$ (b) $l = 0.5$ (c) $l = 0.7$

Figure 1. *IVAR objective function contours for the addition of a new experiment, for varying kernel correlation lengths l. The experiment already performed is shown by the black dot. The location of the next (locally optimal) experiment is shown with the black $\times$. The black square is the starting point for IVAR optimization.*

Finally, we note that in some cases it may be more convenient to work with closed-form expressions for the eigenfunctions of the kernel K rather than the kernel itself; such situations arise when one has a desired basis of approximation but the corresponding closed-form kernel is unknown, or if the eigenfunctions can otherwise be easily computed. In this case, one can rewrite the IVAR objective in terms of eigenfunctions and simply maximize the second term on the right of (2.6). Written in this form, the objective does not require integration with respect to the parameter measure μ. Indeed, having the eigenfunctions in hand is tantamount to already having performed the integration, as the eigenfunctions are solutions of the homogeneous Fredholm integral equation with operator T_K. [23] uses this approach for IVAR minimization, exploring truncations of (2.6) to ℓ eigenfunctions.

3.1.2. Batch and greedy implementations. In this work, we use NLopt's [32] implementation of the globally-convergent method of moving asymptotes to solve the optimization problem (3.3). This problem involves N design points, each in d dimensions, and thus has Nd unknowns. This optimization problem can become expensive to solve when Nd becomes very large. In these cases it may be useful to solve a sequence of smaller optimization problems to achieve an approximate solution. In the numerical examples below, we will investigate constructing these smaller problems through the use of a greedy minimization procedure. In this procedure one decides how many training points M are computationally feasible to minimize. Suppose that $k = N/M$ is an integer. Then the greedy procedure solves k optimization problems of size Md. Once the $(j-1)$th problem is solved, for $j \leq k$, we have obtained the experiments $\boldsymbol{x}_{j-1}$. During the jth iteration we find M points to append to $\boldsymbol{x}_{j-1}$.

3.2. Entropy and mutual information. Statistical criteria underlie many other experimental design procedures for Gaussian process regression [52]. Two popular techniques include minimizing the conditional entropy of the Gaussian process at unobserved locations, or maximizing the mutual information (MI) between the locations at which experiments are performed and the rest of the design space. Our numerical results will compare IVAR designs with MI and entropy-based designs because the latter have algorithms specifically tailored for Gaussian process regression. For a comparison between these two methods and additional design procedures, e.g., based on classical alphabetic optimality criteria, we refer to [34].

The conditional entropy design procedure seeks experiments that reduce the uncertainty, as measured by entropy H, across a typically finite set of possible simulation locations. If the set of candidate experimental locations is denoted by $\mathcal{D}$ and $\boldsymbol{x} \subset \mathcal{D}$ are the chosen locations, then $x^c := \mathcal{D} \setminus \boldsymbol{x}$ are the locations at which the entropy is evaluated. In particular, one seeks $\boldsymbol{x}$ to minimize the conditional entropy $H(F_{\boldsymbol{x}^c}|F_{\boldsymbol{x}}) = -\int p(f_{\boldsymbol{x}^c}, f_{\boldsymbol{x}}) \log p(f_{\boldsymbol{x}^c}|f_{\boldsymbol{x}}) df_{\boldsymbol{x}^c} df_{\boldsymbol{x}}$, where $F_{\boldsymbol{s}}$ is a random variable representing the outputs of the simulation model at a set of inputs $\boldsymbol{s}$ and p denotes a probability density. Minimizing this function is shown to be NP-hard in [33]. In practice, one instead employs greedy but suboptimal algorithms that add one experiment at a time—for example, adding each experiment at the location where the conditional entropy is largest [59, 9, 37]. In the context of GPR, $F_{\boldsymbol{x}}$ and $F_{\boldsymbol{x}^c}$ are jointly Gaussian; this procedure then becomes equivalent to greedily adding experiments at the locations of highest variance, and is commonly called the MacKay criterion (ALM). Other algorithms for choosing a subset of points to minimize the conditional entropy are based on the DETMAX algorithm [40], demonstrated for GP regression in [10]

The mutual information criterion for experimental design considers the change in the entropy of $F_{\boldsymbol{x}^c}$ before and after performing experiments at locations $\boldsymbol{x}$: $H(F_{\boldsymbol{x}^c}) - H(F_{\boldsymbol{x}^c}|F_{\boldsymbol{x}})$, which is equivalent to the mutual information of $F_{\boldsymbol{x}}$ and $F_{\boldsymbol{x}^c}$. This objective is also typically maximized in a greedy fashion: from the candidate set of experiments $\mathcal{D}$, the element which yields the greatest MI is chosen at each iteration. Unlike the conditional entropy, however, MI is submodular, guaranteeing that a greedy approach performs within a constant factor of the full N-experiment maximization [34]. The greedy procedure requires the inversion of a matrix containing the covariance between every pair of candidate experiments, which arises when computing the entropy on the set of all simulation locations. If one is choosing N experiments from a set of $M \gg N$ candidates, each iteration then requires inverting a matrix of size $M \times M$. This expense is typically large and many recent efforts have aimed at reducing it. [34] reduces the cost of each iteration to $\mathcal{O}(M)$ by using specialized local kernels. [3] points out that the quality of an MI design depends crucially on the candidate set, and modifies the greedy algorithm of [34] to be more robust by resampling the set of candidate experiments after each new point is chosen.

Besides the choice of objective, our IVAR-based design algorithm differs from the entropy and MI approaches above in other fundamental ways. First, we select experiments from a continuous design space and thus avoid the challenges of combinatorial optimization. In this sense, we follow the approach of [51] by solving a continuous optimization problem using gradient descent algorithms. Second, as described in Section 3.1.2, our approach can identify multiple experiments simultaneously. Designing batch experiments is advantageous because interactions between the experiments are taken into account; we will demonstrate this advantage empirically in the next section. Our continuous design approach also takes several steps beyond [51]. First, we do not employ particular patterns or partitions to simplify the design space. And as described earlier, we introduce a sample-average approximation of the IVAR objective. This helps design experiments on complex input domain geometries by penalizing proximity to the domain boundaries; the objective automatically becomes large in locations where there are few samples. We will also address the "pileup" problem identified in [51], explaining it in terms of the design size relative to correlation kernel complexity.

3.3. Computational complexity.

In some sense, if evaluating f is sufficiently expensive, then the computational cost of finding a good design is immaterial. But the design procedures described above can have very different costs, and in practice one may not want the computational effort required for experimental design to be too large. Table 1 summarizes the computational complexity of evaluating and optimizing the various experimental design objectives considered above. In the IVAR scenarios, N_{mc} denotes the number of Monte Carlo points sufficient for (3.4), where typically $N_{mc} \gg N$. For the entropy and MI criteria we assume that the number of candidate designs $N_E = |\mathcal{D}| \gg N$. Finally, we assume that optimizing the IVAR objective requires L objective and/or gradient evaluations, where typically $L \ll N_{mc}$.

Greedy minimization of the conditional entropy (ALM) requires N iterations. In each iteration the variance must be computed at N_E locations.[2] At iteration $k < N$, this variance computation requires an $\mathcal{O}(k^3)$ inversion followed by N_E variance evaluations, each of complexity $\mathcal{O}\left(k^2\right)$. Thus the complexity for step k is $\mathcal{O}(k^3 + N_E k^2)$. The total complexity is thus $\mathcal{O}\left(\sum_{k=1}^{N} k^3 + N_E k^2\right) = \mathcal{O}\left(N^4 + N_E N^3\right)$.

The ALM approach is often the computationally cheapest option, but not if $N_E > L N_{mc}/N$. If N_E and N_{mc} are of comparable magnitude, then the comparison of entropy

[2] Although we have described a discrete approach for ALM minimization, a continuous optimization approach is also possible. But the pointwise variance of a GP has many local maxima, and a multi-start procedure would likely be necessary for continuous optimization to be competitive with discrete enumeration. It is then unclear whether continuous optimization would in fact be more efficient in this case.

Design objective	Optimization method	Objective eval	Optimization cost
IVAR	Monte Carlo, batch eigenfunction form, batch	$\mathcal{O}\left(N_{mc}N^2 + N^3\right)$ $\mathcal{O}\left(\ell N^2 + N^3\right)$	$\mathcal{O}\left(LN_{mc}N^2 + LN^3\right)$ $\mathcal{O}\left(\ell LN^2 + LN^3\right)$
entropy	greedy	$\mathcal{O}(N^3 + N_E N^2)$	$\mathcal{O}(N^4 + N_E N^3)$
mutual information	greedy	$\mathcal{O}(N_E^4)$	$\mathcal{O}(NN_E^4)$

Table 1

Computational complexity of different experimental design algorithms.

and IVAR minimization depends on whether $L/N < 1$, i.e., how the number of optimization iterations (in the continuous case) compares to the number of candidate design points (in the discrete case). We also see that the MI design becomes very expensive for large N_E; a small candidate design set must be chosen for the MI procedure to remain tractable. Using a smaller candidate set, however, increases the possibility that the chosen designs will perform poorly.

4. Numerical examples of IVAR designs.

We now provide numerical examples illustrating the quality of designs arising from continuous IVAR minimization. All of the examples in this section were performed using the freely available GPL-licensed GPEXP package [26] for python. First, we examine the interpolatory properties of the points. Then, we compare IVAR, entropy, and MI-based designs on several domains.

4.1. Stability of interpolation.

As discussed above, GP regression with $\sigma^2 = 0$ (and a kernel rank $\ell > N$) corresponds to interpolation. In this setting, it is useful to analyze the quality of interpolation on IVAR design points. The Lebesgue constant is often used to bound the error of a polynomial interpolant relative to the best approximation in a polynomial space of equivalent degree, but it can also be used to analyze interpolation with positive definite kernels, corresponding here to the posterior mean of the GP. Let $m(x)$ denote the interpolant or posterior mean, and let $m^*(x)$ denote the *best* approximation of f, in the L_∞ sense, from the finite-dimensional kernel space $H(K, \boldsymbol{x}_N) = \mathrm{span}\{K(\cdot, x_1), \ldots, K(\cdot, x_N)\}$. We restrict our attention to approximation on compact domains $\mathcal{X} \subseteq \mathbb{R}^d$. Then the L^∞ interpolation error is bounded as [18]

$$\|f - m\|_{L_\infty(\mathcal{X})} \leq (1 + \Lambda_{K,\boldsymbol{x}_N})\|f - m^*\|_{L_\infty(\mathcal{X})},$$

and moreover we have [14],

$$\|m\|_{L_\infty(\mathcal{X})} \leq \Lambda_{K,\boldsymbol{x}_N}\|\boldsymbol{y}\|_{\ell_\infty}.$$

The Lebesgue constant $\Lambda_{K,\boldsymbol{x}_N}$ for prior kernel K and design points $\boldsymbol{x}_N$ is

$$\Lambda_{K,\boldsymbol{x}_N} = \max_{x\in\mathcal{X}} \sum_{j=1}^{N} |u_j(x)|, \tag{4.1}$$

where $\{u_j(x)\}_{j=1}^N$ are the *cardinal functions* satisfying $u_j(x_i) = \delta_{ij}$, such that the interpolant can be written as $m(x) = \sum_{j=1}^N y_j u_j(x)$ with $y_j = f(x_j)$. We evaluate the cardinal functions as in [18] by solving the linear system

$$\mathbf{R}^{-1}\boldsymbol{u}(x) = K(\boldsymbol{x}, x), \tag{4.2}$$

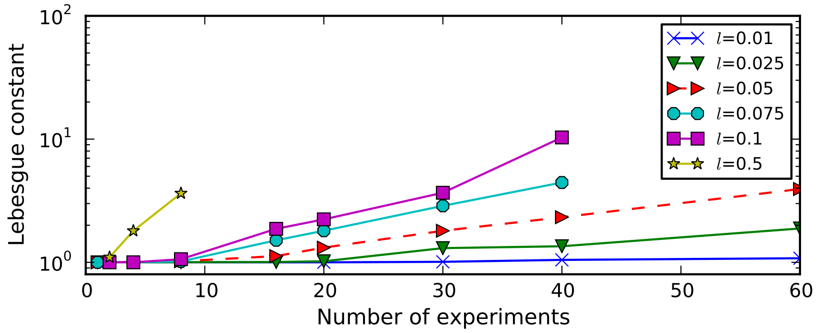


Figure 2. *Lebesgue constants for N-point IVAR designs on $[-1, 1]$ with a squared exponential kernel and various correlation lengths l.*

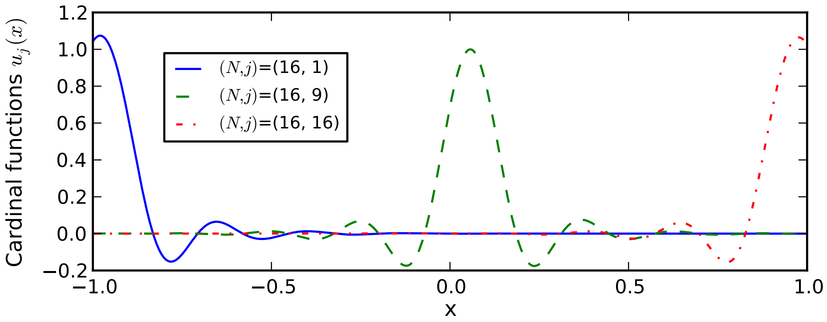


Figure 3. *Cardinal functions for interpolation on 16 IVAR points with a squared exponential kernel and* $l = 0.1$

where $\boldsymbol{u}(x) = [u_1(x) \dots u_N(x)]$.

Now we conduct a simple numerical experiment to evaluate the Lebesgue constants of point sets arising from IVAR minimization. We find IVAR designs on the domain $\mathcal{X} = [-1, 1]$ with squared exponential kernel $K(x, x') = \exp(-(x - x')^2/2l^2)$. Figure 2 shows the associated Lebesgue constants as a function of number of design points N, for various correlation lengths l. Several interesting trends are observed. First, we see that for any value of l, the Lebesgue constant is exactly one for sufficiently small N. This observation is consistent with the asymptotic estimate for $l \to 0$ in [49]. The Lebesgue constant reverts to one for small point sets because the IVAR criterion ensures that the points are well separated, and hence the cardinal functions do not overlap. Once a certain threshold value of N is attained, however, the Lebesgue constant begins to increase; this threshold value is smaller for larger values of the correlation length l. This transition coincides with interactions among the cardinal functions: Figure 3 shows cardinal functions for $N = 16$ and $l = 0.1$, just beyond the threshold where interaction becomes significant. In this regime, the Lebesgue constant increases steadily with N. For a sufficiently large N, $\mathbf{R}^{-1}$ becomes poorly conditioned and direct computations of the interpolant, the cardinal functions, and the Lebesgue constant are no longer numerically stable. From the Bayesian perspective, this ill-conditioning corresponds to the “complexity” of the RKHS associated with the prior kernel—i.e., the effective number of nonzero eigenvalues in (2.5)—being exceeded by the data.

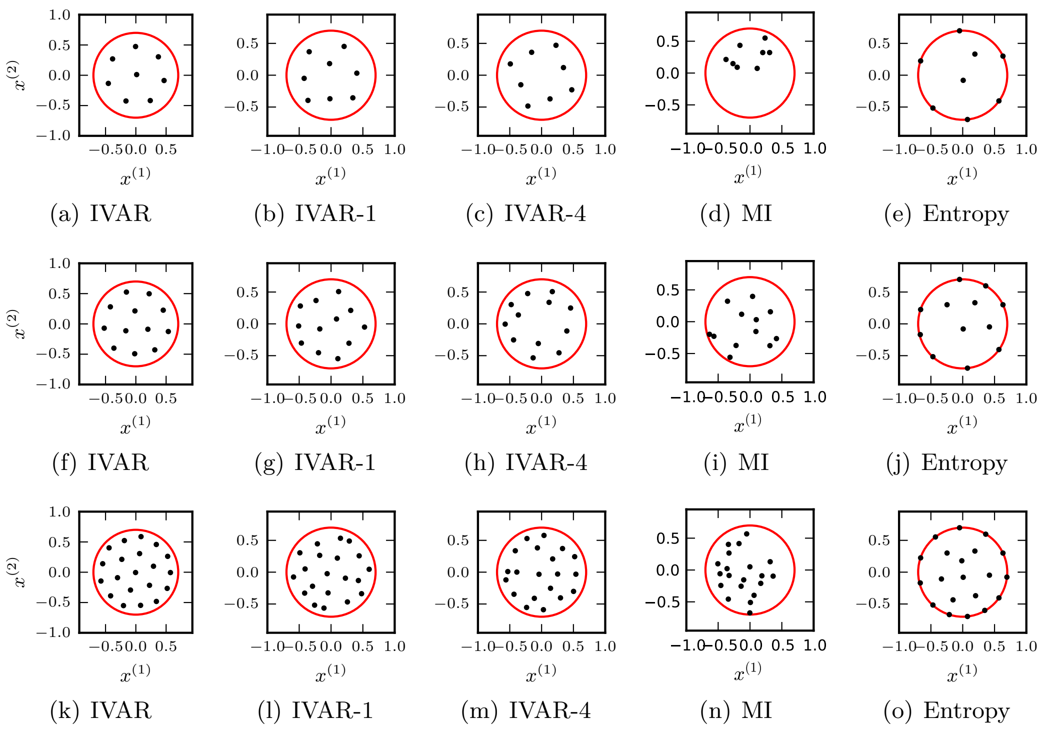
The success of the IVAR optimization procedure itself also depends on how N relates to the complexity of the kernel. In the small-N and small-l regime (intuitively, “too few” points relative to the complexity of the kernel), the IVAR cost function is relatively flat as a function of the design coordinates $\boldsymbol{x}$, and distinguishing the quality of different designs becomes difficult. By contrast, in the large-N and large-l regime (“too many” points relative to the complexity of the kernel), the IVAR value itself is exceedingly small and the problem is poorly conditioned, as described above. Away from these limiting regimes, however, the IVAR design procedure yields relatively slow growth in the Lebesgue constant and thus relatively stable interpolation.

4.2. Approximation on non-hypercubic domains. Here we illustrate IVAR designs on variety of input domains $\mathcal{X}$, and compare the performance of IVAR designs with that of designs obtained through conditional entropy minimization or MI maximization. We highlight designs and approximations on irregular input domains (e.g., domains that are neither hypercubes nor $\mathbb{R}^d$), which are critical in many real-world applications. In particular, irregular domains often arise as a result of domain partitioning by a discontinuity detection algorithm, for models whose outputs are piecewise smooth. For example, in [30, 27] the authors automatically partition the input domain of a genetic toggle switch model that exhibits a phase transition. Following this partitioning, function approximation proceeds on two separate but irregular domains. In [54, 55] the authors study a climate model which exhibits a discontinuity. Discontinuity detection is used to split the input domain into two irregular subdomains, and is followed by function approximation. We will thus attempt to show the applicability of our algorithm to input domains that are characteristic of such problems, which have no guarantees of convexity or even connectedness.

In all our numerical experiments, we employ an isotropic squared exponential kernel $K(x, y) = \exp(-\|x - y\|^2/2l^2)$ and we set $\sigma^2 = 10^{-10}$. Entropy minimization is pursued through the ALM approach described earlier; each experiment is chosen by comparing the variance at $N_E = 10^4$ possible experimental locations sampled according to the parameter measure μ on $\mathcal{X}$. For the mutual information objective, we select designs from $N_E = 150$ randomly sampled locations, because the domains under consideration are not amenable to Latin hypercube or quasi-Monte Carlo designs. The size of the MI candidate set is chosen so that the computational times of the different experimental design procedures are comparable; it is also comparable to sizes used in the literature [3]. We find that small changes in the size of this candidate set do not qualitatively change the results reported below.

Finally, we define the relative L^2 error of a GP approximation as $\|f - m\|_{L^2_\mu}/\|f\|_{L^2_\mu}$, and we estimate it using 10^5 Monte Carlo samples. While in the previous section we considered $L^\infty(\mathcal{X})$ error, recall that the IVAR objective function reflects the expectation of a squared μ-averaged error. Thus L^2_μ is a logical metric of quality. Other efforts, e.g., [36], empirically evaluate the L^2 and L^∞ errors of approximations resulting from a variety of other design criteria.

4.2.1. Circular domain. We first construct designs on a circular domain in $\mathbb{R}^2$ with radius 0.7. We fix the correlation length in the kernel K to $l = 0.2$; the measure μ is uniform over the domain. Results from each design strategy are shown in Figure 4, for designs with $N = 8$, 12, and 20 points. We consider full batch IVAR, designing all N points



(a) IVAR (b) IVAR-1 (c) IVAR-4 (d) MI (e) Entropy

(f) IVAR (g) IVAR-1 (h) IVAR-4 (i) MI (j) Entropy

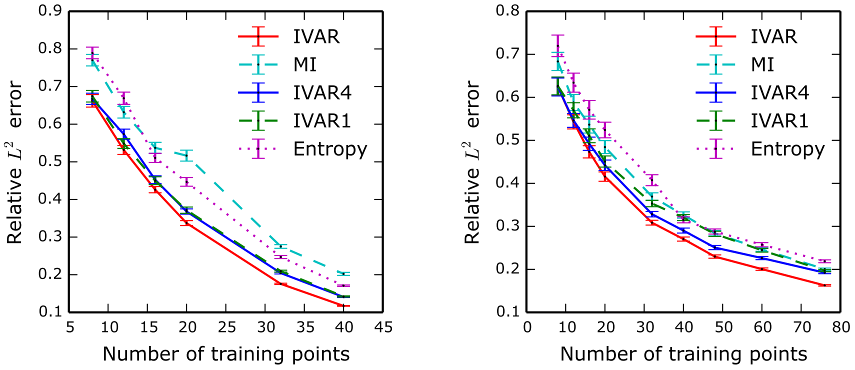
(k) IVAR (l) IVAR-1 (m) IVAR-4 (n) MI (o) Entropy

Figure 4. *Eight- (first row), twelve- (second row), and twenty-point (third row) designs obtained under various experimental design strategies for an isotropic Gaussian kernel with $l = 0.2$. IVAR-M corresponds to a greedy IVAR strategy, adding points in batches of M.*

simultaneously, along with greedy IVAR strategies that add points in groups of $M = 1$ or $M = 4$. Full IVAR results in the most symmetric and regularly spaced points. But all of the IVAR strategies attempt to spread the points throughout the domain, whereas entropy design first places points on the boundaries, which is not a desirable feature for radial basis function kernels [47, 34].

Additionally, we see that that MI designs are not particularly space-filling. This behavior can be attributed to poor candidate points, highlighting the fact that a well chosen candidate set is crucial for MI maximization. Yet generating a good but reasonably-sized candidate set can be a non-trivial task for many domain geometries, especially in high dimensions. For timing reference, we note that finding a 40-point design with the full IVAR criterion took 11 seconds, IVAR-1 took 65 seconds, IVAR-4 took 29 seconds, entropy-based design took 10 seconds, and MI design took 79 seconds. Non-greedy IVAR design and entropy design take approximately the same amount of time, while the MI criterion takes slightly longer, though of the same order of magnitude. Constructing better candidate sets for MI could potentially incur further timing penalties.

To evaluate the effectiveness of these designs for function approximation, we perform GP regression on 1000 functions independently sampled from the prior Gaussian process, with results shown in Figure 5(a). For each sampled function f, we compute the relative L^2 error between the posterior mean m of the GP and f, as described above. We then report the average and standard deviation of this error, for each design strategy and different values



(a) Two-dimensional domain; prior correlation length $l = 0.2$

(b) Five-dimensional domain; prior correlation length $l = 0.5$

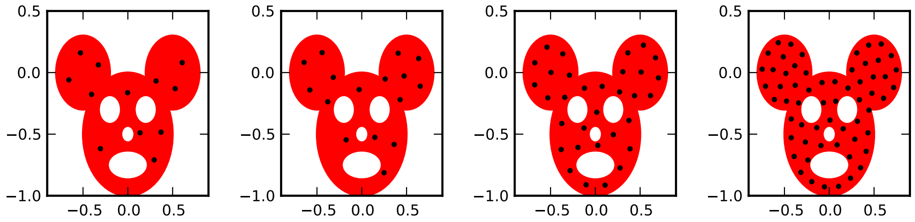
Figure 5. *Relative L^2 approximation error on 1000 functions drawn from prior GP; error bars indicate sample standard deviations of these errors. Domains $\mathcal{X}$ are balls in $\mathbb{R}^2$ or $\mathbb{R}^5$.*

of N. The IVAR designs, including the IVAR-M greedy strategies, clearly outperform the entropy and MI designs. Optimality of the full IVAR designs is to be expected, since we are essentially calculating the Bayes risk (the expectation of $\|f - m\|^2_{L^2_\mu}$) which is equivalent to IVAR, as discussed in Section 3. But it is noteworthy that even the greedy IVAR strategies show better performance than the MI and entropy designs. In Figure 5(b), we repeat this study for a higher-dimensional domain—a ball in $\mathbb{R}^5$—and observe similar trends. For reference, the computational times required to find 40-point designs in $d = 5$ dimensions are: 13 seconds for batch IVAR, 125 seconds for IVAR-1, 45 seconds for IVAR-4, 10 seconds for entropy, and 82 seconds for MI. These results further support the idea that batch IVAR is computationally competitive with entropy and that design with MI is more expensive.

4.2.2. Non-convex, non-simply connected domain. Figure 6 illustrates experimental design on a parameter domain that is neither convex nor simply connected, found via full (non-greedy) IVAR minimization. These designs are obtained with an isotropic squared exponential kernel with correlation length $l = 0.1$. The IVAR objective is minimized using the SAA approach described in Section 3.1.1. Optimization for each N-point design began from a single randomly-generated point in the $2N$-dimensional design space; no multi-start procedure was used here. The designs do a good job covering the domain. While twelve- and sixteen-point designs are not completely evenly distributed, designs using 32 and 60 points are very well spaced and have interesting symmetries. The approximation performance of IVAR designs—even greedy IVAR-M designs—is also superior to that of the entropy and MI approaches, as shown in Figure 7. Recall that we use a candidate set of 150 randomly chosen samples for MI, found via rejection sampling. Constructing better MI candidate sets for such a domain, especially in high dimensions, would itself be computationally challenging.

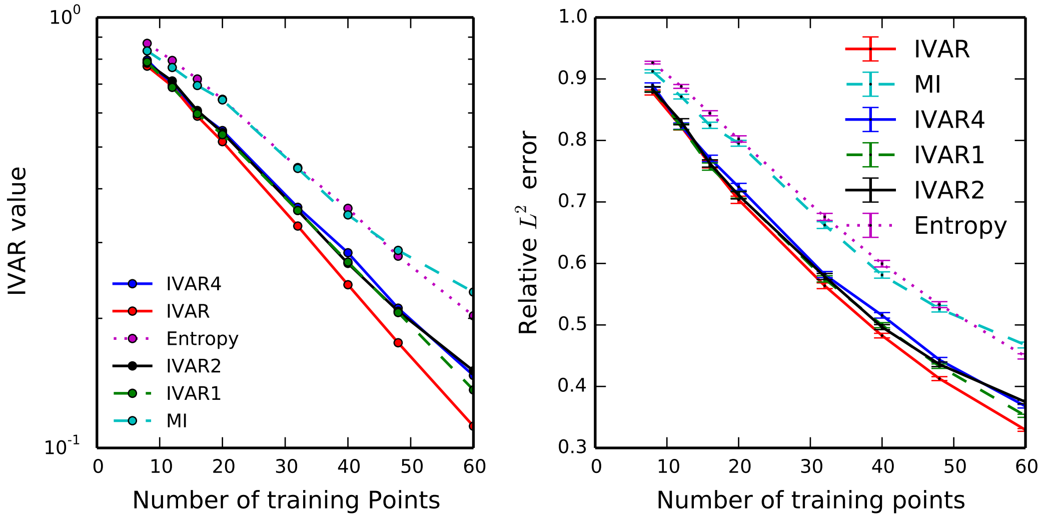
5. Comparisons with pseudospectral approximation.

Pseudospectral approximation (PSA) is a well-established approach for computing polynomial approximations from pointwise function evaluations $f(x)$ [62, 8, 7]. Recall that PSA comes equipped with an experi-



(a) 12 points (b) 16 points (c) 32 points (d) 60 points

Figure 6. *Experimental designs computed by minimizing the IVAR cost function over a non-convex and non-simply connected domain; the domain is the red/shaded region, endowed with a uniform parameter measure.*



(a) Cost function

(b) Function approximation results on 1000 functions drawn from isotropic kernel with $l = 0.1$

Figure 7. *Performance of IVAR minimization on non-convex, non-simply connected domain.*

mental design procedure and an algorithm which seeks accuracy in an L^2_μ sense. In Sections 2–4, we described an experimental design procedure for GP regression that also targets an expected L^2_μ-type error. Thus we are now well positioned to make a comparison between GP regression and PSA. In particular, our comparison of these surrogate construction methods arises from the perspective of the three choices described in the introduction: the approximation space, the experiments, and the algorithm for constructing the approximation itself. Our comparison comprises three components described in Sections 5.2, 5.3, and 5.4. We first discuss a theoretical result bounding the error between the GP posterior mean and the PSA under orthogonalizing designs. Then we show what happens when we relax the idea of exact orthogonality. Finally, we provide some numerical comparisons between the two methods on canonical approximation problems.

Overall, our goal is not so much to compare performance but rather to understand the links and distinctions between these approaches. In particular, we would like to know: are GP regression and pseudospectral approximation ever equivalent? Do the point sets used for pseudospectral approximation yield low IVAR when used for GP regression? How do IVAR-minimizing designs compare with quadrature rules, for appropriate choices of kernel?

5.1. Pseudospectral approximation.

In this section, we recall the basics of pseudospectral approximation. Consider a set of basis functions $\{\psi_i(x)\}$ comprising a complete orthonormal system in L^2_μ. Pseudospectral approximation takes advantage of orthonormality

$$\langle \psi_i, \psi_j \rangle_\mu = \int \psi_i(x)\psi_j(x) d\mu(x) = \delta_{ij},$$

to compute the projection of a function f onto a subspace of L^2_μ spanned by ℓ basis functions. The orthogonal projection

$$f_\ell(x) = \sum_{i=1}^{\ell} \langle f, \psi_i \rangle_\mu \psi_i(x).$$

converges in the L^2_μ sense as the subspace grows: $\lim_{\ell\to\infty} \int (f_\ell - f)^2 \, d\mu = 0$. Pseudospectral approximation departs from exact orthogonal projection by using numerical quadrature to approximate the inner products $\langle f, \psi_i \rangle$. In particular, one seeks an *orthogonalizing* set of nodes and weights, $\mathcal{Q} = \{(x_k, w_k) : k = 1 \dots N\}$, i.e., a quadrature rule that computes the inner products between the first ℓ basis functions exactly:

$$\langle \psi_i, \psi_j \rangle_\mu = \sum_{k=1}^{N} w_k \psi_i(x_k)\psi_j(x_k), \; i, j = 1, \dots, \ell. \tag{5.1}$$

This choice eliminates internal aliasing error [7], though the pseudospectral approximation will still exhibit external aliasing error, since f is not necessarily in the span of $\{\psi_1, \dots, \psi_\ell\}$. Given an orthogonalizing rule (5.1), a pseudospectral approximation $\hat{f}_\ell$ can be written as

$$\hat{f}_\ell(x) = \sum_{i=1}^{\ell} \left(\sum_{k=1}^{N} w_k f(x_k)\psi_i(x_k) \right) \psi_i(x) = \sum_{k=1}^{N} \left[w_k f(x_k) \left(\sum_{i=1}^{\ell} \psi_i(x_k)\psi_i(x) \right) \right]. \tag{5.2}$$

In subsequent analysis, it will be convenient to express $\hat{f}_\ell(x)$ in matrix notation. Let $\mathbf{W} := \operatorname{diag}(w_1, \dots w_N)$, $\boldsymbol{y} := [f(x_1), \dots, f(x_N)]$ and $\boldsymbol{\psi}_i := [\psi_i(x_1), \dots, \psi_i(x_N)]$. Then

$$\hat{f}_\ell(x) = \boldsymbol{y}^T \mathbf{W} \sum_{i=1}^{\ell} \boldsymbol{\psi}_i \psi_i(x). \tag{5.3}$$

Examples of $\{\psi_i\}$ and $\mathcal{Q}$ include one-dimensional orthogonal polynomial families and their corresponding Gaussian quadratures; or tensorized versions of each in multiple dimensions. However, the basis functions ψ_i do not in general need to be polynomials, and other quadrature rules besides Gaussian rules may exist. Using Mercer's theorem, we can already see that $\sum_{i=1}^{\ell} \psi_i(x)\psi_i(y)$ can be interpreted as a truncated kernel $K^\ell(x, y)$ with eigenvalues that do not decay, and we can interpret the pseudospectral approximation given in (5.3) as a weighted combination of such kernels.

5.2. Same approximation space and experiments. To begin our comparison, we will fix two of the choices involved in surrogate modeling and evaluate the impact of the third. In particular, this section will compare the impact of different *algorithms*—i.e., using pseudospectral approximation versus GP regression—for identical experiments and for the same or similar approximation spaces. Because pseudospectral approximation requires experiments to be chosen according to an orthogonalizing rule, we will also use these experiments for GP regression. We are now left to relate the basis for the pseudospectral approximation to the GP kernel, as these objects determine the approximation space.

Theorem 2.1 lets us represent any Mercer kernel, and the corresponding GP posterior mean function, using the eigenfunctions. With this connection, we first develop a more general result than required. Specifically, we show that when the basis for pseudospectral approximation comprises a subset of the eigenfunctions of a given kernel, then the L^2_μ-norm of the difference between the resulting GP posterior mean function and the pseudospectral approximation may be bounded. Results for identical approximation spaces follow immediately as a corollary of this general case.

For the result below, we will assume that we have a Mercer kernel consisting of ℓ_{GP} eigenfunctions, where ℓ_{GP} could be finite or infinite, and a spectral expansion consisting of the first $\ell \le \ell_{GP}$ eigenfunctions, where ℓ is finite. The approximations will be constructed from N evaluations of the function f, performed at nodes of a quadrature rule $\{(x_i, w_i)\}\,, i = 1 \dots N$, that orthogonalizes the first ℓ eigenfunctions. Clearly N depends on ℓ. Let $\mathbf{USU}^T$ be the eigendecomposition of the matrix of covariances $K(x_i, x_j)$ among all the design points computed without a nugget, i.e., $\mathbf{R}^{-1} - \sigma^2\mathbf{I} = \mathbf{USU}^T$. $\mathbf{S}$ is a diagonal matrix with elements $\mathbf{S} = \mathrm{diag}(s_1, \dots, s_N)$. Finally, assume a zero mean prior $m_0(x) = 0$ for the GP model. The latter assumption does not restrict the generality of our results. In fact, one can transform the problem from one of approximating f to one of approximating $g = f - m_0$, for some fixed function $m_0(x)$. Of course, this transformation requires one to translate the data $\boldsymbol{y}$ accordingly. Under these conditions, we have the following result, whose proof is given in Appendix A.

Theorem 5.1. *Let $\hat{f}_\ell(x)$ be a pseudospectral approximation represented with basis functions $\{\psi_1, \dots, \psi_\ell\}$, computed via an orthogonalizing quadrature rule $\mathcal{Q}_\ell$ as in* (5.1)–(5.2)*. Let $M = \max_{(x,w)\in\mathcal{Q}_\ell,\ i\in\{1,\dots,\ell\}} |\psi_i(x)|$. Also, let $w_{\max} = \max_{i\in\{1,\dots,N\}} w_i$. Let $m(x)$ be the posterior mean of GP regression with prior covariance kernel $K(x,x') = \sum_{i=1}^{\ell_{GP}} \lambda_i \phi_i(x)\phi_i(x')$ and nugget term $\sigma^2 > 0$, constructed from function evaluations at the nodes of $\mathcal{Q}_\ell$. If $\ell, N < \infty$, $\psi_i = \phi_i$ for $i = 1 \dots \ell$, and $\ell \le \ell_{GP}$, then the difference between these two approximations is bounded as*

$$\|m - \hat{f}_\ell\|_{L^2_\mu}^2 \le \|\boldsymbol{y}\|_2^2 \left(\frac{\ell N M^2 w_{\max}^2}{(s_N + \sigma^2)^2} \left\| \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I} \right\|_2^2 + \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j^2 \boldsymbol{\phi}_j^T \mathbf{R}^2 \boldsymbol{\phi}_j \right).$$

We can consider a relative notion of error by dividing each side by $\|\boldsymbol{y}\|_2^2$. Furthermore, since we are dealing with bounded functions f and finite amounts of data, we can always normalize and center the data to make $\|\boldsymbol{y}\|_2 = \mathcal{O}(1)$. The difference between the two approximations described in Theorem 5.1 has two sources. First is the contribution of kernel eigenfunctions that are not in the pseudospectral approximation basis, i.e., the summations involving $j > \ell$ above. Second is the contribution of the noise term σ^2.

Because the kernel eigenfunctions span the RKHS containing the GP posterior mean, the case of equivalent approximation spaces for pseudospectral approximation and GP regression follows simply by setting $\ell_{GP} = \ell < \infty$. This case is highlighted by Corollary 5.2.

Corollary 5.2. *Let $\ell_{GP} = \ell$. Then the difference between the pseudospectral approximation $\hat{f}_\ell(x)$ and GP posterior mean $m(x)$ defined in Theorem 5.1 is:*

$$\|m - \hat{f}_\ell\|^2_{L^2_\mu} \leq \|\boldsymbol{y}\|^2_2 \, \ell N M^2 w^2_{\max} \left(\frac{\sigma^2}{s_N + \sigma^2} \right)^2 \tag{5.4}$$

In this case, the difference between approximations is due only to the nugget $\sigma^2 > 0$ and the smallest eigenvalue of the covariance matrix as indicated by s_N. If we additionally have that $N \leq \ell_{GP}$, then the design covariance matrix $\mathbf{R}^{-1}$ remains invertible (i.e., with $s_N > 0$) even as $\sigma^2 \to 0$, yielding zero difference between the approximations. This occurs, for example, in the case of kernels constructed from fully tensorized polynomial eigenfunctions and tensorized Gaussian quadrature rules, where $N = \ell = \ell_{GP}$.

We also note that the bound in Theorem 5.1 depends on the minimum eigenvalue of the covariance matrix, represented (after diagonalization) by $\mathbf{S} + \sigma^2\mathbf{I}$. If $\mathbf{S}$ is nearly rank-deficient and the nugget is sufficiently small, then the bound can be large. This situation is not purely an artifact of the theory; indeed, it corresponds to a poorly conditioned numerical problem, where the actual difference between the two approximations may be large as well. One may imagine a case where $\mathbf{R}$ is not invertible, e.g., too many quadrature nodes are used and the GP kernel is finite rank; in this case the computation of m becomes unstable whereas the computation of $\hat{f}_\ell$ can still proceed in a stable manner.

Besides bounding the difference between approximations, we can also analyze the impact of an orthogonalizing experimental design on the integrated variance of the GP posterior. Consider, again, a kernel consisting of ℓ_{GP} eigenfunctions and a training set corresponding to the nodes of a quadrature rule $\mathcal{Q}$ that orthogonalizes the first ℓ eigenfunctions. We begin by splitting (2.6) into summations involving the first ℓ eigenfunctions and the remaining $\ell + 1$ to ℓ_{GP} eigenfunctions:

$$\int c(x) d\mu(x) = \sum_{i=1}^{\ell} \lambda_i \left(1 - \lambda_i \boldsymbol{\phi}_i^T \mathbf{R} \boldsymbol{\phi}_i\right) + \sum_{i=\ell+1}^{\ell_{GP}} \lambda_i \left(1 - \lambda_i \boldsymbol{\phi}_i^T \mathbf{R} \boldsymbol{\phi}_i\right).$$

The second term in this expansion represents the contribution of the extra eigenfunctions to the integrated variance. We cannot comment on how our training points will affect this term because they are designed to have special properties only for the first ℓ eigenfunctions. But we can use these properties to analyze the first term above, thus describing the impact of an orthogonalizing rule on the associated integrated variance. Note that the weights $\mathbf{W}$ do not explicitly enter the GP regression; nonetheless, as we show in Appendix B, this portion of the integrated variance can be rewritten as:

$$\sum_{i=1}^{\ell} \lambda_i \left(1 - \lambda_i \boldsymbol{\phi}_i^T \mathbf{R} \boldsymbol{\phi}_i\right) = \sum_{i=1}^{\ell} \lambda_i \boldsymbol{\phi}_i^T \mathbf{W} \left(\sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I} \right) \mathbf{R} \boldsymbol{\phi}_i. \tag{5.5}$$

This expression consists of interactions between the first ℓ eigenfunctions and the last eigenfunctions; it also includes the impact of the nugget. The eigenfunction interactions are expected because the design only orthogonalizes the first ℓ eigenfunctions; errors are incurred when the numerical inner product is taken between one of the first ℓ eigenfunctions and one of the remaining eigenfunctions. We see that if $\ell = \ell_{GP}$, the integrated variance is of the order of the noise σ^2 as expected. When additionally $\sigma^2 = 0$, the integrated variance is zero. An orthogonalizing design ensures that the integrated variance captures only the contributions of eigenfunctions which are not orthogonalized.

The preceding results highlight some of the assumptions underlying the practical application of these two surrogate construction methodologies. When using a pseudospectral approximation, one implicitly assumes that the true function's projection onto basis functions *not* included in the expansion (and hence not orthogonalized by the underlying design) is small. Otherwise, more points and basis functions should be added to the approximation; indeed, adaptive basis selection is the concern of a vast array of approximation methods, pseudospectral and otherwise. In GP regression, a properly chosen kernel is one whose eigenvalue decay matches the decay of the spectral coefficients of the true function. In this case the function will lie in the RKHS associated with the kernel and an accurate approximation can readily be achieved. These two approaches interact through the integrated variance. The orthogonalizing rule in a pseudospectral approximation ensures that the difference between f and f_ℓ lies mostly in the span of the eigenfunctions $\phi_{k>\ell}$. Correspondingly, this rule forces the IVAR, a measure of the uncertainty and error (via the Bayes risk) of the GP approximation, to retain contributions only from the same subspace (that spanned by $\{\phi_{k>\ell}\}$). We also see that the difference between the GP posterior mean and pseudospectral approximation is dominated by these extra eigenfunctions.

5.3. Relaxing exact orthogonality.

It is instructive to consider the tradeoff between exactly orthogonalizing *fewer* basis functions or, for the same design points, generating an approximation using more basis functions than can be orthogonalized. The latter corresponds to performing GP regression under the conditions of Theorem 5.1 with $\ell_{GP} > \ell$.

Consider the function $f(x) = \sin(\pi x + 0.2)$ for $x \in \mathbb{R}$ with standard Gaussian weight μ. We will use a Gauss-Hermite quadrature rule with $N = 20$ points to construct a pseudospectral approximation of f with the first $\ell = 20$ Hermite polynomials (degrees 0 to 19) as basis functions. We compare this approximation to Gaussian process regression on the same points using the Mehler kernel [56]

$$K(x, y; t) = \frac{1}{\sqrt{1-t^2}} \exp\left(-\frac{1}{2}\frac{(x)^2\, t^2 - 2txy + (y)^2\, t^2}{1-t^2}\right), \tag{5.6}$$

which is a closed-form expression for $K(x, y) = \sum_{i=0}^{\infty} t^i \phi_i(x)\phi_i(y)$, where $\{\phi_i\}$ are again normalized Hermite polynomials. The nugget $\sigma^2 = 0$ and the decay constant $t = 0.8$. In this way we compare two surrogates using identical experiments and closely related approximation spaces: the approximation space of the former is contained in that of the latter. But the two surrogates use different approaches for combining the same function evaluations into an approximation. GP regression approximates f using an infinite collection of polynomial eigenfunctions, with more emphasis on those associated with higher eigenvalues, while pseudospectral approximation projects f onto finite polynomial basis, with the exactness of the projection limited by external aliasing.

Figure 8 shows the results of this experiment. First, we note that the relative L^2_μ errors are 8.7×10^{-3} for pseudospectral approximation and 1.8×10^{-3} for the GP mean function. Perhaps more interesting than this improvement, however, is the *spectrum* of the error associated with each approximation. The left panel of Figure 8 shows the projections of the errors $e_{ps}(x) = \hat{f}_\ell(x) - f(x)$ and $e_{gp}(x) = m(x) - f(x)$ onto each basis function ϕ_i, i.e., $|\langle e, \phi_i \rangle|$. For reference, the right panel shows how much energy f has in each basis direction, via the magnitudes of the *exact* projections $|\langle f, \phi_i \rangle|$. (All projections are computed with extremely high-order quadrature.) We see that the projection of the pseudospectral approximation error e_{ps} onto the first few basis functions is small, even though f itself has significant energy in these directions. The projection of e_{ps} then rises with the basis index and peaks around an index of 20; it then begins to decay, just as the exact coefficients $\langle f, \phi_i \rangle$ themselves decay in magnitude. The projection of the GP error, on the other hand, is flatter across the indices. It rises slightly for the first few basis functions and then begins to decay somewhat more slowly. The basis functions for which the orange line (GP error) lies below the red line (PSA error) correspond to relatively large coefficient values; this results in a lower L^2 error overall.

As a preview of the next section, we include a third line in Figure 8 representing GP regression performed with a 20-point IVAR-minimizing design. The relative L^2_μ error of this approximation is even smaller: 1.0×10^{-5}. And the spectrum of its error, shown with the grey line in the left panel, is even flatter than that of the quadrature-based GP approximation. This result amplifies the previous trend: compared to pseudospectral approximation, GP regression spreads its error more evenly over the spectrum. Departing from an orthogonalizing design allows this error to be spread even more broadly.

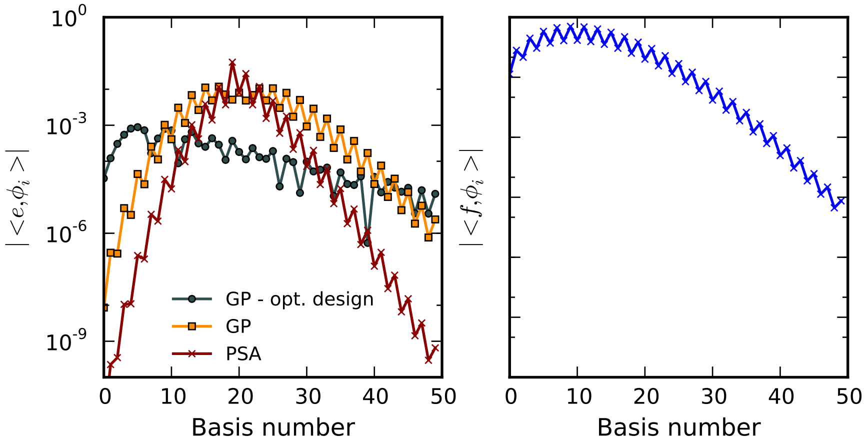


Figure 8. *Spectrum of approximation errors for the example function discussed in Section 5.3. Left figure shows the projection of the approximation error onto normalized Hermite polynomial basis functions ϕ_i, for three different surrogates: pseudospectral approximation with 20 Gauss quadrature points, GP regression with a Mehler kernel on the same Gauss quadrature points, and GP regression with a Mehler kernel on IVAR-optimal points. Right figure shows the magnitude of the exact projection of f onto each basis function ϕ_i.*

5.4. Similar approximation spaces, optimal experiments.

Our next goal is to compare GP regression and pseudospectral approximation when the approximation spaces are similar, but the experimental designs (and the algorithms) differ. We have already seen that RKHS containing the GP mean can coincide with the range of the pseudospectral approximation operator in the case of a finite-rank GP kernel: the eigenfunctions of the kernel can simply be used as the basis for the pseudospectral approximation. In the numerical comparisons below, however, we would like to allow the GP kernel to have infinite rank. This choice is more representative of how GP regression is used in practice, and in principle may allow the GP mean to better approximate a wider variety of functions. Of course, the eigenvalues of the kernel need to decay, so that we have a bounded kernel.

We will therefore use the Mehler kernel (5.6) as in Section 5.3, and consider target functions whose inputs are endowed with standard Gaussian weight on $\mathbb{R}^d$. The eigenfunctions of the Mehler kernel are Hermite polynomials, which we use as basis functions for pseudospectral approximation. In $d > 1$ dimensions, we use a tensorized version of the Mehler kernel: $K(x, y; t_1, \ldots, t_d) = \prod_{i=1}^{d} K(x^{(i)}, y^{(i)}; t_i)$, where t_i now governs the decay rate of the eigenvalues associated with the univariate eigenfunctions in dimension i.

To design experiments for GP regression, we use IVAR minimization in combination with adaptation of the covariance kernel hyperparameters (e.g., correlation lengths) and the nugget σ^2. In particular, we choose these parameters by maximizing the log-marginal likelihood,

$$\log p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \theta) = -\frac{1}{2}\hat{\boldsymbol{y}}^T \mathbf{R} \hat{\boldsymbol{y}} - \frac{1}{2}\log|\mathbf{R}^{-1}| - \frac{N}{2}\log 2\pi, \tag{5.7}$$

with respect to the complete set of hyperparameters (including σ^2), denoted by θ. Our approach interleaves kernel adaptation with batch IVAR design. The batch size for each design is described in each example, but the number of Monte Carlo points used to evaluate the IVAR objective is fixed at $N_{mc} = 10^4$.

For pseudospectral approximation, we use a state-of-the-art adaptive Smolyak algorithm [7], which adaptively enriches both the approximation basis and the experimental design using a greedy heuristic. The approximation is essentially a Smolyak sum of full tensor-product polynomial approximations, each computed using a pseudospectral approach. We use this approach because a regular tensor-product approach becomes infeasible for more than a few dimensions; hence many sparse grid [2] and dimension-adaptive [25] polynomial approximation algorithms have been developed. The Smolyak algorithm in [7] uses generalized sparse grids and dimension adaptivity and thus provides a useful benchmark for comparison with kernel-adaptive GP. It is implemented in the MIT Uncertainty Quantification framework (MUQ) [46]. Since the input domain is endowed with Gaussian measure, the sparse grid design is based on one-dimensional Gauss-Hermite quadrature rules, with the number of points growing exponentially with the level of the sparse grid.

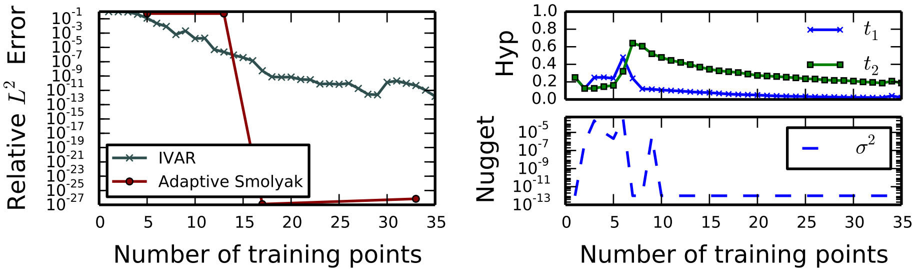
5.4.1. Additively separable functions.

The first problems we consider are two dimensional and additively separable. Additive separability is a property favorable to the Smolyak algorithm, in part because of the greedy heuristic the algorithm uses to enrich the polynomial basis. The target functions are:

$$f_1(x) = x^{(1)} + (x^{(2)})^2, \tag{5.8}$$

$$f_2(x) = \sin(2\pi x^{(1)}) + (x^{(2)})^2. \tag{5.9}$$

The batch size for the closed-loop IVAR scheme is one training point. Adaptation is performed for the eigenvalue decay rates t_1 and t_2 in each dimension and for the noise term σ^2. Figures 9 and 10 show the resulting relative errors and hyperparameter traces, as a function of the number of training points. As in Section 4, the relative errors are defined as $||f - \hat{f}_\ell||_{L^2_\mu}/||f||_{L^2_\mu}$ and $||f - m(x)||_{L^2_\mu}/||f||_{L^2_\mu}$ for Smolyak pseudospectral approximation and GP/IVAR, respectively. These errors are computed using 10000 Monte Carlo points.



(a) Relative error comparisons (b) Trace of hyperparameter estimates

Figure 9. *Error comparisons between GP/IVAR and adaptive Smolyak approximations for target function (5.8). Right panel shows hyperparameter traces for the GP design procedure.*

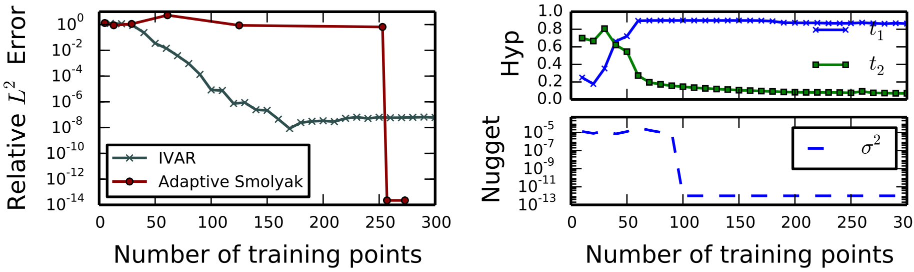
In these examples, we observe that the adaptive Smolyak algorithm requires several function evaluations until it begins to converge. Once it converges, however, the relative error drops to machine precision. (Recall that the quadrature rules used in the Smolyak scheme require adding several points at a time.) The GP approximation error, on the other hand, decreases steadily but gradually as additional points are added and as the kernel is refined. In both cases, by the time the pseudospectral approximation becomes accurate, the GP approximation has already achieved at least 10^{-4} relative error, perhaps sufficient for many applications. The hyperparameter traces show how the GP covariance kernel adapts to the function being approximated. Function f_1 (5.8) is fairly low order in both dimensions, and the hyperparameter values t_1 and t_2 both converge to fairly low numbers, indicating fast eigenvalue decay. The converged value of t_1 is small because only the constant and linear eigenfunctions (in the first dimension) are updated by the data; t_2 converges to a slightly higher value, thus slowing the eigenvalue decay, to account for the quadratic term in $x^{(2)}$. Function f_2 (5.9), on the other hand, is relatively high order in the first dimension but low order in the second. We thus observe that t_1 converges to 0.9, corresponding to a slower decay, and that t_2 converges to roughly 0.1; the hyperparameter optimization has "learned" that only the first few eigenfunctions in this dimension matter.

5.4.2. Non-additively separable function.

Next we perform the same experiments on the three-dimensional Ishigami function

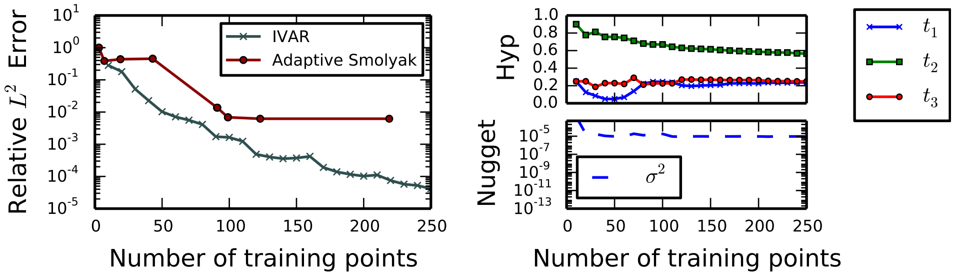
$$f(x) = \sin(x^{(1)}) + 7\sin^2(x^{(2)}) + 0.05\left(x^{(3)}\right)^4 \sin(x^{(1)}), \tag{5.10}$$

The Ishigami function is not additively separable, and we expect to see even better relative performance of GP regression in this example since the kernel eigenfunctions include the tensor products of all univariate Hermite polynomials. Here we use a batch size of $M = 10$ experiments for IVAR design and hyperparameter adaptivity. The results are shown in Figure 11.



(a) Relative error comparisons (b) Trace of hyperparameter estimates

Figure 10. *Error comparisons between GP/IVAR and adaptive Smolyak approximations for target function (5.9). Right panel shows hyperparameter traces for the GP design procedure.*



(a) Relative error comparisons (b) Trace of hyperparameter estimates

Figure 11. *Error comparisons between GP/IVAR and adaptive Smolyak approximations for the Ishigami function* (5.10).

We see that the GP begins to outperform the pseudospectral approximation after roughly 20 function evaluations, with an error that decreases consistently. The error from the adaptive Smolyak approach, on the other hand, plateaus in several regions. After 50 iterations, the GP approximation reaches a relative error of 10^{-2}, while it takes pseudospectral approximation 100 iterations to reach the same relative error. This behavior may be attributed to the fact that the Mehler kernel accounts for interactions among the input variables immediately, whereas the dimension-adaptive Smolyak approach requires some exploration (corresponding to the error plateaus) to find the basis functions that capture these interactions.

One may also be interested in how GP approximation with the Mehler kernel compares to GP approximation with the more commonly used squared exponential kernel. We repeat the previous experiments with the Ishigami function, again using batch sizes of $M = 10$ points, and show results for both kernels in Figure 12. Hyperparameter adaptation for the squared exponential kernel targets the correlation lengths $l^{(0)}$, $l^{(1)}$, $l^{(2)}$, and the variance parameter γ. The results for the Mehler kernel in Figure 11 and Figure 12 differ slightly because of randomness in the choice of initial points, in the starting points for optimization, and in the SAA approximation. The comparison in Figure 12 shows that the Mehler kernel generally outperforms the squared exponential kernel in this example.

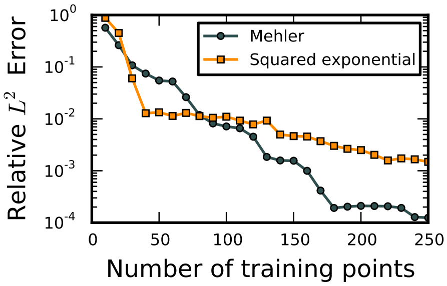


Figure 12. *Comparison between GP/IVAR approximations using the Mehler kernel and the squared exponential kernel, for the Ishigami function* (5.10).

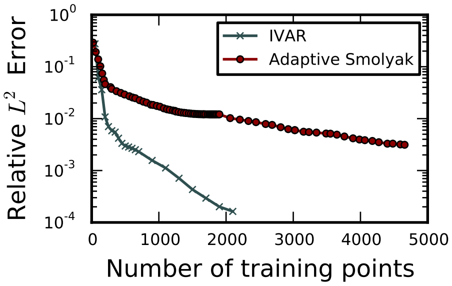


Figure 13. *Error comparisons between GP/IVAR and adaptive Smolyak approximations for the ten-dimensional Genz function (5.11).*

5.4.3. Ten-dimensional Genz function.

Finally, we consider a higher-dimensional example using the oscillatory Genz function [24]:

$$f_4(x) = \cos\left(2\pi w_1 + \sum_{i=1}^{10} x^{(i)} c_i\right), \tag{5.11}$$

where $w_1 = 0.3$ and the c_i are chosen randomly from a uniform distribution on $[0, 1]$ and then normalized to $\|c\|_1 = 2.25$. This Genz function is typically evaluated on a hypercube domain $[-1, 1]^{10}$ with normalization $\|c\|_1 = 9$. We have found the present approximation problem, with an unbounded and Gaussian-weighted domain, to be significantly more challenging, however, due to the tail behavior of the Hermite polynomials. A comparison of approximation errors, again between GP/IVAR using the Mehler kernel and adaptive Smolyak pseudospectral approximation, is shown in Figure 13. The GP approximation performs extremely well. We note that this Genz function involves non-additive coupling among all ten input dimensions, a feature that may amplify the benefits of including a fully tensorized set of eigenfunctions via the GP kernel. For the GP regression calculations, we initially added experiments in batches of $M = 50$, learning the hyperparameters after each batch, until obtaining 700 experiments total. Then we added experiments 200 at a time. In the future, it may be useful to automatically stop adapting the hyperparameters once the changes in the hyperparameters between iterations fall below some threshold.

Note that we were able to compare GP and pseudospectral approximation in these examples because we had access to the Mehler kernel, and hence explicit evaluations of the Hermite basis functions were not required for GP regression. In practice, we may not have closed-form kernels for other basis function families, typically used in pseudospectral approximation. Truncated kernels whose eigenfunctions are polynomials, however, can often be represented using the Christoffel-Darboux formula [5, 11]. For normalized basis functions, the Christoffel-Darboux formula is essentially a finite rank kernel with equal eigenvalues, and thus for a fixed basis order, one would not be able to adapt this kernel.

6. Conclusion.

This paper has examined experimental design criteria and optimization procedures used to develop surrogates of computational models, highlighting aspects of the interplay between experimental designs and approximation algorithms. In the first part of the paper, we discussed experimental design criteria for Gaussian process regression and presented an algorithm for minimizing the posterior integrated variance (IVAR) of a Gaussian process over a continuous design space. Our approach is adapted to the kernel structure; for instance, with isotropic squared exponential kernels, it yields well-spaced points on arbitrary complex domains, avoiding boundary clustering and other undesirable artifacts. IVAR points also have good interpolation stability, as measured by the Lebesgue constant for kernel interpolation. The underlying optimization problem is tractably solved with gradient descent methods, as long as the number of design points is not too small or too large relative to the complexity of the prior covariance kernel. Our numerical experiments also demonstrate that simultaneously designing multiple points can yield substantial benefits over greedy strategies. Nonetheless, even greedy minimization of IVAR yields better approximation performance than standard greedy algorithms for minimizing entropy or maximizing MI. We also demonstrate the use of IVAR in a closed-loop adaptive scheme that sequentially updates the prior covariance kernel of the Gaussian process.

In the second part of the paper, we compare GP regression to polynomial approximation. For simplicity, we consider only pseudospectral approximation, using nodes and weights that orthogonalize the finite set of basis functions used for approximation. In this setting, when GP regression uses the same nodes and a kernel with polynomial eigenfunctions, the difference between the two approximations is due only to the GP "nugget" and truncation of the kernel. When instead coupled with an IVAR design, GP regression for an infinite-rank kernel sacrifices numerical orthogonality of any given set of eigenfunctions; compared to the pseudospectral approach, projection errors may be larger for the first few eigenfunctions but are more nearly constant over the eigenspectrum. This observation is reminiscent of the average-case quadrature of [39], which compares a Bayesian method for deriving quadrature nodes with Gauss rules of fixed degree. We follow these comparisons with an empirical study of approximation performance on various test functions, comparing adaptive variants of GP regression and sparse pseudospectral approximation. We observe that while additively separable functions lend themselves easily to sparse polynomial approximations, more strongly coupled functions can be more efficiently approximated using the GP approach. Kernel approximation is also well suited to complex (e.g., non-tensorized) input domains. In our current design approach, while eigenfunctions of the kernel integral operator on the domain need not be explicitly computed, eigenfunctions that couple the input dimensions are all implicitly present.

Future work can extend our analysis of experimental design for GP regression in many ways. It is useful to compare the present design criteria with other methods for choosing "good" interpolation nodes in the radial basis function literature [13, 15]. Comparing IVAR-optimal nodes to optimal nodes for "Bayesian quadrature" [39, 45] would also be of great interest. More broadly, we advocate for closer connections between the numerical analysis literature and the statistics literature on these topics. Finally, adaptive designs that interleave point selection with updates to the kernel would benefit from more rigorous study. Current sequential approaches, including the ones presented here, are perhaps reasonable heuristics. But look-ahead strategies that anticipate information gain from future designs, and that balance information for kernel adaptation with reduction of the conditional posterior variance, may lead to even more effective approximation procedures.

Acknowledgments. The authors gratefully acknowledge BP for funding this research. We thank Tarek Moselhy, Akil Narayan, and Alessio Spantini for many helpful discussions. We also thank the referees and editor for their thoughtful comments and suggestions.

Appendix A. Proof of Theorem 5.1.

First we recall some notation. The prior precision matrix is $\mathbf{R} = \left(\sum_{j=1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I}\right)^{-1}$. We define $\mathbf{U}$ and $\mathbf{S}$ to be the matrices associated with the eigenvalue decomposition $\mathbf{R} = \mathbf{U}\left(\mathbf{S} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{U}^T$.

We assume, without loss of generality, that the prior mean is zero; from (2.1) we obtain

$$\begin{aligned} m(x) = \boldsymbol{y}^T\mathbf{R}\sum_{j=1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \phi_j(x) &= \boldsymbol{y}^T\mathbf{R}\sum_{j=1}^{\ell} \lambda_j \boldsymbol{\phi}_j \phi_j(x) + \boldsymbol{y}^T\mathbf{R}\sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \phi_j(x) \\ &= \boldsymbol{y}^T\mathbf{R}\sum_{j=1}^{\ell} \lambda_j \boldsymbol{\phi}_j \phi_j(x) + a(x), \end{aligned}$$

where we have replaced $K(\boldsymbol{x}, x)$ with its eigenfunctions and then separated the posterior mean into two terms, the first containing the first ℓ eigenfunctions and the second $a(x)$ denoting the rest.

We now use the orthogonality rule (5.1) to obtain $m(x) = \boldsymbol{y}^T\mathbf{R}\sum_{j=1}^{\ell} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T \mathbf{W} \boldsymbol{\phi}_j \phi_j(x) + a(x)$. Recall that the pseudospectral approximation $\hat{f}_\ell$ is given by $\hat{f}_\ell(x) = \boldsymbol{y}^T\mathbf{W}\sum_{j=1}^{\ell} \boldsymbol{\phi}_j \phi_j(x)$, such that the difference between the GP and the spectral expansion becomes:

$$m(x) - \hat{f}_\ell(x) = \sum_{j=1}^{\ell}\left[\boldsymbol{y}^T\left(\mathbf{R}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T - \mathbf{I}\right)\mathbf{W}\boldsymbol{\phi}_j\phi_j(x)\right] + a(x).$$

Now define $\boldsymbol{d}^T := \boldsymbol{y}^T\left(\mathbf{R}\sum_{j=1}^{\ell}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T - \mathbf{I}\right)\mathbf{W}$ for convenience, and form the L^2_μ difference

$$\begin{aligned} \int\left(m - \hat{f}_\ell\right)^2 d\mu &= \int\left(\sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x) + a(x)\right)^2 d\mu \\ &= \left\langle \sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x), \sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x)\right\rangle + \left\langle\sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x), a(x)\right\rangle + \langle a(x), a(x)\rangle \\ &= \left\langle \sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x), \sum_{j=1}^{\ell}\boldsymbol{d}^T\boldsymbol{\phi}_j\phi_j(x)\right\rangle + \langle a(x), a(x)\rangle \qquad \text{(A.1)} \end{aligned}$$

where the the cross terms resulting from the expansion of the square disappear due to orthogonality between the first ℓ basis functions and the $k > \ell$ basis functions.

We now seek to bound the size of the first term in (A.1). Use the orthogonality properties of ϕ_j to obtain

$$\left\langle \sum_{j=1}^{\ell} \boldsymbol{d}^T \boldsymbol{\phi}_j \phi_j(x), \sum_{j=1}^{\ell} \boldsymbol{d}^T \boldsymbol{\phi}_j \phi_j(x) \right\rangle = \sum_{i,j=1}^{\ell} \boldsymbol{\phi}_i^T \boldsymbol{d}\boldsymbol{d}^T \boldsymbol{\phi}_j \langle \phi_i, \phi_j \rangle = \sum_{i=1}^{\ell} \left(\boldsymbol{d}^T \boldsymbol{\phi}_i \right)^2$$

We can further simplify this expression by taking advantage of the invertibility of $\mathbf{R}$ and rewriting $\boldsymbol{d}^T$ as

$$\begin{aligned}
\sum_{i=1}^{\ell} \left(\boldsymbol{d}^T \boldsymbol{\phi}_j \right)^2 &= \sum_{i=1}^{\ell} \left(\boldsymbol{y}^T \mathbf{R} \left(\sum_{j=1}^{\ell} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T - \mathbf{R}^{-1} \right) \mathbf{W} \boldsymbol{\phi}_i \right)^2 \\
&= \sum_{i=1}^{\ell} \left(\boldsymbol{y}^T \mathbf{R} \left(\sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T - \sigma^2 \mathbf{I} \right) \mathbf{W} \boldsymbol{\phi}_i \right)^2 \\
&\leq \boldsymbol{y}^T \boldsymbol{y} \sum_{i=1}^{\ell} \left\| \mathbf{R} \left(\sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T - \sigma^2 \mathbf{I} \right) \mathbf{W} \boldsymbol{\phi}_i \right\|_2^2 ,
\end{aligned}$$

where the first equality is obtained by extracting $\mathbf{R}$, the second equality results from substituting in the definition of $\mathbf{R}^{-1}$, and the bound is due to the Cauchy-Schwarz inequality. Now we can factorize the last term above because all induced norms are sub-multiplicative,

$$\begin{aligned}
\sum_{i=1}^{\ell} \left(\boldsymbol{d}^T \boldsymbol{\phi}_j \right)^2 &\leq \boldsymbol{y}^T \boldsymbol{y} \, \|\mathbf{R}\|^2 \sum_{i=1}^{\ell} \left\| \left(\sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T - \sigma^2 \mathbf{I} \right) \mathbf{W} \boldsymbol{\phi}_i \right\|_2^2 \\
&\leq \boldsymbol{y}^T \boldsymbol{y} \left\| \left(\mathbf{S} + \sigma^2 \mathbf{I} \right)^{-1} \right\|^2 \left\| \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I} \right\|_2^2 \sum_{i=1}^{\ell} \|\mathbf{W} \boldsymbol{\phi}_i\|^2 \\
&\leq \boldsymbol{y}^T \boldsymbol{y} \frac{1}{\left(s_N + \sigma^2\right)^2} \left\| \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I} \right\|_2^2 \ell w_{\max}^2 \boldsymbol{\phi}_{z_2}^T \boldsymbol{\phi}_{z_2}
\end{aligned}$$

where for the third equality we use the fact that $\mathbf{W}$ is a diagonal positive definite matrix; $w_{\max}^2$ is the square of the largest-magnitude weight, $w_{\max}^2 = \arg\max_{i \in \{1,\ldots,N\}} w_i^2$; and z_2 is the index of the discrete basis function with maximum norm, $z_2 = \arg\max_{i \in \{1,\ldots,\ell\}} \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i$. Let $M = \max\limits_{(x,w) \in \mathcal{Q}_\ell, i \in \{1,\ldots \ell\}} |\phi_i(x)|$, since the first ℓ eigenfunctions are bounded at the N quadrature nodes. Then we have $\boldsymbol{\phi}_{z_2}^T \boldsymbol{\phi}_{z_2} \leq M^2 N$ and

$$\sum_{i=1}^{\ell} \left(\boldsymbol{d}^T \boldsymbol{\phi}_j \right)^2 \leq \boldsymbol{y}^T \boldsymbol{y} \frac{\ell N M^2 w_{\max}^2}{\left(s_N + \sigma^2\right)^2} \left\| \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^T + \sigma^2 \mathbf{I} \right\|_2^2$$

We now turn to the $\langle a(x), a(x) \rangle$ term. This term is simplified similarly to the first, using Cauchy-Schwarz and the orthogonality of the basis functions:

$$\langle a(x), a(x) \rangle \leq \boldsymbol{y}^T \boldsymbol{y} \sum_{j=\ell+1}^{\ell_{GP}} \lambda_j^2 \boldsymbol{\phi}_j^T \mathbf{R} \mathbf{R} \boldsymbol{\phi}_j .$$

Thus the overall bound can be given as:

$$\|m-\hat{f}_\ell\|^2_{L^2_\mu} \leq \boldsymbol{y}^T\boldsymbol{y}\left(\frac{\ell N M^2 w^2_{\max}}{(s_N+\sigma^2)^2}\left\|\sum_{j=\ell+1}^{\ell_{GP}}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T+\sigma^2\mathbf{I}\right\|_2^2+\sum_{j=\ell+1}^{\ell_{GP}}\lambda_j^2\boldsymbol{\phi}_j^T\mathbf{R}^2\boldsymbol{\phi}_j\right).$$

Appendix B. Derivation of Equation (5.5).

We begin by splitting (2.6) into two components in order to focus on the first term:

$$\int c d\mu = \sum_{i=1}^{\ell}\lambda_i\left(1-\lambda_i\boldsymbol{\phi}_i^T\mathbf{R}\boldsymbol{\phi}_i\right)+\sum_{i=\ell+1}^{\ell_{GP}}\lambda_i\left(1-\lambda_i\boldsymbol{\phi}_i^T\mathbf{R}\boldsymbol{\phi}_i\right). \tag{B.1}$$

We again use the trick of replacing one with $\boldsymbol{\phi}_i^T\mathbf{W}\boldsymbol{\phi}_i$ to obtain

$$\begin{aligned}
\sum_{i=1}^{\ell}\lambda_i\left(1-\lambda_i\boldsymbol{\phi}_i^T\mathbf{R}\boldsymbol{\phi}_i\right) &= \sum_{i=1}^{\ell}\lambda_i\left(\boldsymbol{\phi}_i^T\mathbf{W}\boldsymbol{\phi}_i-\lambda_i\boldsymbol{\phi}_i^T\mathbf{R}\boldsymbol{\phi}_i\right)\\
&= \sum_{i=1}^{\ell}\lambda_i\boldsymbol{\phi}_i^T\left(\mathbf{W}-\lambda_i\mathbf{R}\right)\boldsymbol{\phi}_i\\
&= \sum_{i=1}^{\ell}\lambda_i\boldsymbol{\phi}_i^T\left(\mathbf{W}\left(\sum_{j=1}^{\ell_{GP}}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T+\sigma^2\mathbf{I}\right)-\lambda_i\mathbf{I}\right)\mathbf{R}\boldsymbol{\phi}_i\\
&= \sum_{i=1}^{\ell}\lambda_i\boldsymbol{\phi}_i^T\left(\lambda_i\mathbf{I}+\mathbf{W}\sum_{j=\ell+1}^{\ell_{GP}}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T+\sigma^2\mathbf{W}-\lambda_i\mathbf{I}\right)\mathbf{R}\boldsymbol{\phi}_i\\
&= \sum_{i=1}^{\ell}\lambda_i\boldsymbol{\phi}_i^T\mathbf{W}\left(\sum_{j=\ell+1}^{\ell_{GP}}\lambda_j\boldsymbol{\phi}_j\boldsymbol{\phi}_j^T+\sigma^2\mathbf{I}\right)\mathbf{R}\boldsymbol{\phi}_i,
\end{aligned}$$

where the first equality comes from orthogonality, the second equality results from extracting the basis vectors, the third equality results from extracting $\mathbf{R}$ on the right, and the fourth equality comes from splitting the sum over all ℓ_{GP} terms into two sums: from 1 to ℓ and from $\ell+1$ to ℓ_{GP}.

REFERENCES

[1] A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum experimental designs, with SAS*, volume 34. Oxford University Press Oxford, 2007.

[2] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12(4):273–288, 2000.

[3] J. Beck and S. Guillas. Sequential design with mutual information for computer experiments (MICE): Emulation of a tsunami model. *arXiv preprint arXiv:1410.0215*, 2014.

[4] J. Buescu. Positive integral operators in unbounded domains. *Journal of Mathematical Analysis and Applications*, 296(1):244–255, 2004.

[5] E. B. Christoffel. Über die gaußische quadratur und eine verallgemeinerung derselben. *Journal für die reine und angewandte Mathematik*, 55:61–82, 1858.

[6] D. A. Cohn. Neural network exploration using optimal experiment design. *Neural networks: the official journal of the International Neural Network Society*, 9(6):1071–1083, August 1996.

[7] P. R. Conrad and Y. M. Marzouk. Adaptive Smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing*, 35(6):A2643–A2670, 2013.

[8] P. G. Constantine, M. S. Eldred, and E. T. Phipps. Sparse pseudospectral approximation method. *Computer Methods in Applied Mechanics and Engineering*, 229:1–12, 2012.

[9] N. Cressie. *Statistics for spatial data*. John Wiley and Sons, Inc, 1993.

[10] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

[11] G. Darboux. Mémoire sur l'approximation des fonctions de très-grands nombres, et sur une classe étendue de développements en série. *Journal de Mathématiques Pures et Appliquées*, pages 5–56, 1878.

[12] A. DasGupta. Review of optimal Bayes designs. Technical Report 95-4, Department of Statistics, Purdue University, February 1995.

[13] S De Marchi. On optimal center locations for radial basis function interpolation: computational aspects. *Rend. Sem. Mat. Univ. Pol. Torino (Splines Radial Basis Functions and Applications)*, 61(3):343–358, 2003.

[14] S. De Marchi and R. Schaback. Stability of kernel-based interpolation. *Advances in Computational Mathematics*, 32(2):155–161, 2010.

[15] S. De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, 23(3):317–330, 2005.

[16] P. Diaconis. Bayesian numerical analysis. *Statistical decision theory and related topics IV*, 1:163–175, 1988.

[17] R. M. Dudley. *Real analysis and probability*, volume 74. Cambridge University Press, 2002.

[18] G. E. Fasshauer. Positive definite kernels: past, present and future. *Dolomite Research Notes on Approximation*, 4:21–63, 2011.

[19] V. Fedorov. Design of spatial experiments: Model fitting and prediction. *Handbook of Statistics*, 13:515–553, 1996.

[20] V. Fedorov and W. G. Müller. Optimum design for correlated fields via covariance kernel expansions. *mODa8 – Advances in Model-Oriented Design and Analysis*, pages 57–66, 2007.

[21] V. V. Fedorov and D. Flanagan. Optimal monitoring network design based on Mercer's expansion of covariance kernel. *Journal of Combinatorics, Information and System Sciences*, 23:237–250, 1997.

[22] J. C. Ferreira and V. A. Menegatto. An extension of Mercer's theory to L_p. *Positivity*, 16(2):197–212, 2012.

[23] B. Gauthier and L. Pronzato. Spectral approximation of the IMSE criterion for optimal designs in kernel-based interpolation models. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):805–825, 2014.

[24] A. Genz. Testing multidimensional integration routines. In *Proc. Of International Conference on Tools, Methods and Languages for Scientific and Engineering Computation*, pages 81–94, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[25] T. Gerstner and M. Griebel. Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87, 2003.

[26] A. Gorodetsky. GPEXP: Experimental design for Gaussian process regression in python. https://github.com/GPEXP, 2013-2015.

[27] A. Gorodetsky and Y. Marzouk. Efficient localization of discontinuities in complex computational simulations. *SIAM Journal on Scientific Computing*, 36(6):A2584–A2610, 2014.

[28] R. B. Gramacy and D. W. Apley. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, pages 1–28, 2014.

[29] O. Harari and D. M. Steinberg. Optimal designs for Gaussian process models| via spectral decomposition. *Journal of Statistical Planning and Inference*, 154:87–101, 2014.

[30] J. D. Jakeman, R. Archibald, and D. Xiu. Characterization of discontinuities in high-dimensional stochastic problems on adaptive sparse grids. *Journal of Computational Physics*, 230(10):3977–3997, 2011.

[31] S. Jeong, M. Murayama, and K. Yamamoto. Efficient optimization design method using kriging model. *Journal of Aircraft*, 42(2):413–420, 2005.

[32] S. G. Johnson. The NLopt nonlinear optimization package, 2010.

[33] C. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[34] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms, and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[35] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian process regression. In *IJCAI*, volume 7, pages 944–949, 2007.

[36] J. L. Loeppky, L. M. Moore, and B. J. Williams. Batch sequential designs for computer experiments. *Journal of Statistical Planning and Inference*, 140(6):1452–1464, 2010.

[37] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, July 1992.

[38] H. Minh, P. Niyogi, and Y. Yao. Mercer's theorem, feature maps, and smoothing. *Learning theory*, pages 154–168, 2006.

[39] T. P. Minka. Deriving quadrature rules from Gaussian processes. *Statistics Department, Carnegie Mellon University, Tech. Rep*, 2000.

[40] T. J. Mitchell. An algorithm for the construction of "D-optimal" experimental designs. *Technometrics*, 16(2):203–210, 1974.

[41] Werner G Müller. A comparison of spatial design methods for correlated observations. *Environmetrics*, 16(5):495–505, 2005.

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

[43] A. O'Hagan. Some Bayesian numerical analysis. *Bayesian Statistics*, 4:345–363, 1992.

[44] A. O'Hagan and J. F. C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978.

[45] M. Osborne, Z. Garnett, R.and Ghahramani, D. K. Duvenaud, S. J. Roberts, and C. E. Rasmussen. Active learning of model evidence using Bayesian quadrature. In *Advances in Neural Information Processing Systems*, pages 46–54, 2012.

[46] M. Parno, P. Conrad, A. Davis, and Y. Marzouk. MIT Uncertainty Quantification library. http://muq.mit.edu.

[47] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepallit, and V. N. Pandey. Gaussian processes for active data mining of spatial aggregates. In *Proceedings of the Fifth SIAM International Conference on Data Mining*, volume 119, page 427. SIAM, 2005.

[48] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*, volume 1. MIT Press, Cambridge, MA, 2006.

[49] SD Riemenschneider and N Sivakumar. On cardinal interpolation by Gaussian radial-basis functions: properties of fundamental functions and estimates for Lebesgue constants. *Journal d'Analyse Mathématique*, 79(1):33–61, 1999.

[50] J. Sacks, S. B. Schiller, and W. J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.

[51] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[52] T. J. Santner, B. J. Williams, and W. Notz. *The design and analysis of computer experiments*. Springer, 2003.

[53] T. J. Santner, B. J. Williams, and W. I. Notz. *The design and analysis of computer experiments*. Springer Science & Business Media, 2013.

[54] K. Sargsyan, C. Safta, B. Debusschere, and H. Najm. Uncertainty quantification in the presence of limited climate model data with discontinuities. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 241–247. IEEE, 2009.

[55] Khachik Sargsyan, Cosmin Safta, Bert Debusschere, and Habib Najm. Uncertainty quantification given discontinuous model response and a limited number of model runs. *SIAM Journal on Scientific Computing*, 34(1):B44–B64, 2012.

[56] R. Schaback. Kernel-based meshless methods. *Lecture Notes for Taught Course in Approximation Theory. Georg-August-Universität Göttingen*, 2007.

[57] S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: Active data selection and test point rejection. In *Mustererkennung 2000*, pages 27–34. Springer, 2000.

[58] A. Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, 30(1):169–186, 1991.

[59] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of applied statistics*, 14(2):165–170, 1987.

[60] G. Spöck and J. Pilz. Spatial sampling design and covariance-robust minimax prediction based on convex design ideas. *Stochastic Environmental Research and Risk Assessment*, 24(3):463–482, 2010.

[61] David M Steinberg and Dizza Bursztyn. Data analytic tools for understanding random field regression models. *Technometrics*, 46(4):411–420, 2004.

[62] Dongbin Xiu. Fast numerical methods for stochastic computations: a review. *Communications in Computational Physics*, 5(2–4):242–272, 2009.