SIAM J. SCI. COMPUT.
Vol. 34, No. 6, pp. A2837–A2871

ADAPTIVE HESSIAN-BASED NONSTATIONARY GAUSSIAN PROCESS RESPONSE SURFACE METHOD FOR PROBABILITY DENSITY APPROXIMATION WITH APPLICATION TO BAYESIAN SOLUTION OF LARGE-SCALE INVERSE PROBLEMS*

TAN BUI-THANH†, OMAR GHATTAS‡, AND DAVID HIGDON§

Abstract. We develop an adaptive Hessian-based non-stationary Gaussian process (GP) response surface method for approximating a probability density function (pdf) that exploits its structure, particularly the Hessian of its negative logarithm. Of particular interest to us are pdfs that arise from the Bayesian solution of large-scale inverse problems, which imply very expensive-to-evaluate pdfs. The method can be considered as a piecewise adaptive Gaussian approximation in which a Gaussian tailored to the local Hessian of the negative log probability density is constructed for each subregion in high dimensional parameter space. The task of efficiently partitioning the parameter space into subregions is done implicitly through Hessian-informed membership probability functions. The GP machinery is then employed to glue all local Gaussian approximations into a global analytical response surface that is far cheaper to evaluate than the original expensive probability density. The resulting response surface is also equipped with an analytical variance estimate that can be used to assess the uncertainty of the approximation. One of the key components of our proposed approach is an adaptive sampling strategy for exploring the parameter space efficiently during the computer experimental design step, which aims to find training points with high probability density. The detailed construction and an analysis of the method are presented. We then demonstrate the accuracy and efficiency of the proposed method on several example problems, including inverse shape electromagnetic scattering in 24-dimensional parameter space.

Key words. probability density approximation, Gaussian process, response surface, adaptive sampling, computer experimental design, nonstationary, curse of dimensionality, Bayesian inversion, covariance function, membership probability, adjoint, Hessian

AMS subject classifications. 62G07, 62G08, 62K20

DOI. 10.1137/110851419

1. Introduction.

Solving large-scale ill-posed inverse problems that are governed by partial differential equations (PDEs) is both of great practical importance in science and industry as well as tremendously challenging. Classical deterministic inverse methodologies, which provide point estimates of the solution, are not capable of rigorously accounting for the uncertainty in the inverse solution. The Bayesian formulation provides a systematic quantification of uncertainty by posing the inverse problem as one of statistical inference. The Bayesian framework for inverse problems proceeds as follows: given observational data and their uncertainty, the governing

*Submitted to the journal's Methods and Algorithms for Scientific Computing section October 12, 2011; accepted for publication (in revised form) August 23, 2012; published electronically November 8, 2012. This research was supported by AFOSR grant FA9550-09-1-0608; DOE grants DE-SC0002710, DE-FC52-08NA28615, and DEFC02-06ER25782; and NSF grants CMS-1028889, OPP-0941678, DMS-0724746, and CMS-0619078.

http://www.siam.org/journals/sisc/34-6/85141.html

†Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712 (buithanhtan2000@yahoo.com).

‡Jackson School of Geosciences, The University of Texas at Austin, Austin, TX 78712, and Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712. Current address: Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712 (OMAR@ices.utexas.edu).

§Statistical Sciences, CCS-6, Los Alamos National Laboratory, Los Alamos, NM 87545 (dhigdon@lanl.gov).

forward problem and its uncertainty, and a prior probability density function (pdf) describing uncertainty in the parameters $\mathbf{m} \in \mathbb{R}^N$, the solution of the inverse problems is the posterior probability distribution $\pi_{\text{post}}(\mathbf{m})$ over the parameters. Bayes' theorem explicitly gives the posterior pdf as

$$\pi_{\text{post}}(\mathbf{m}|\mathbf{y}_{\text{obs}}) \propto \pi_{\text{prior}}(\mathbf{m})\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m}),$$

which combines the prior pdf $\pi_{\text{prior}}(\mathbf{m})$ and the likelihood $\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m})$. The prior encodes any knowledge or assumptions about the parameter space that we may wish to impose before any data are considered, while the likelihood $\pi_{\text{like}}(\mathbf{y}_{\text{obs}}|\mathbf{m})$ explicitly represents the probability that a given set of parameters $\mathbf{m}$ might give rise to the observed data $\mathbf{y}_{\text{obs}} \in \mathbb{R}^p$. For simplicity of exposition, we assume that the prior is Gaussian and that the measurement and PDE model errors are combined into a noise term $\mathbf{e} = \mathbf{y}_{\text{obs}} - \mathbf{f}(\mathbf{m})$, which is additive and i.i.d. (independently and identically distributed) Gaussian. Then the pdfs for the prior and likelihood can be written in the form

$$\pi_{\text{prior}}(\mathbf{m}) \propto \exp\left(-\frac{1}{2}(\mathbf{m}-\bar{\mathbf{m}}_{\text{prior}})^T\boldsymbol{\Gamma}_{\text{prior}}^{-1}(\mathbf{m}-\bar{\mathbf{m}}_{\text{prior}})\right),$$

$$\pi_{\text{like}}(\mathbf{e}) \propto \exp\left(-\frac{1}{2}(\mathbf{e}-\bar{\mathbf{e}})^T\boldsymbol{\Gamma}_{\text{noise}}^{-1}(\mathbf{e}-\bar{\mathbf{e}})\right),$$

respectively, where $\bar{\mathbf{m}}_{\text{prior}}$ is the mean of the prior distribution, $\bar{\mathbf{e}}$ the mean of the Gaussian noise, $\boldsymbol{\Gamma}_{\text{prior}} \in \mathbb{R}^{N\times N}$ the covariance matrix for the prior, and $\boldsymbol{\Gamma}_{\text{noise}} \in \mathbb{R}^{p\times p}$ the covariance matrix of the noise. Restating Bayes' theorem with these Gaussian pdfs, we find that

$$\pi_{\text{post}}(\mathbf{m}) \propto \exp\left(-\frac{1}{2}\|\mathbf{m}-\bar{\mathbf{m}}_{\text{prior}}\|^2_{\boldsymbol{\Gamma}_{\text{prior}}^{-1}} - \frac{1}{2}\|\mathbf{y}_{\text{obs}}-\mathbf{f}(\mathbf{m})-\bar{\mathbf{e}}\|^2_{\boldsymbol{\Gamma}_{\text{noise}}^{-1}}\right), \tag{1.1}$$

where $\mathbf{f}(\mathbf{m})$ is the (nonlinear) operator mapping parameters to observations. Note that the seemingly simple expression $\mathbf{f}(\mathbf{m})$ belies the complexity of the underlying computations, which involves: (i) creation of the PDE model for given parameters $\mathbf{m}$, (ii) solution of the governing PDEs to yield the output state variables, and (iii) extraction of the observables (i.e., the values of the states at the observation locations in space and time). In general, $\mathbf{f}(\mathbf{m})$ is nonlinear, even when the forward PDEs are linear in the state variables, since the parameters couple with the states nonlinearly in the forward PDEs.

As is clear from the expression (1.1), despite the choice of prior and noise probability distributions as Gaussian, the posterior probability distribution need not be Gaussian, due to the nonlinearity of $\mathbf{f}(\mathbf{m})$. The non-Gaussianity of the posterior poses challenges for computing statistics for typical large-scale inverse problems since π_{post} is often a surface in high (thousands or millions) dimensions, and evaluating each point on this surface requires a solution of the forward PDEs. Numerical quadrature to compute the mean and covariance matrix, for example, is out of the question. Usually, the method of choice for computing statistics is Markov chain Monte Carlo (MCMC) [29], which judiciously samples the posterior distribution, so that sample statistics can be used to approximate the exact ones. But the use of MCMC for large-scale inverse problems is still prohibitive for expensive forward problems and high dimensional parameter spaces, since even for modest numbers of parameters the number of samples required can be in the thousands or millions. Nevertheless, MCMC can be made more efficient by exploiting higher order information such as the Hessian [45].

Since solving the forward PDEs is the most expensive component of evaluating the posterior pdf, one can employ model reduction techniques to construct inexpensive-to-solve reduced-order models of the PDEs [41, 25, 51, 8, 35, 68]. On the other hand, one can reduce the cost of evaluating the likelihood directly using polynomial chaos [47, 46]. One can also pose the task of approximating the Bayesian solution as a density estimation problem, for which there is a vast literature, including classical density estimation, multidimensional kernel density approximation, and mixture density estimation; see [63, 64] and references therein. Finally, one can reduce the cost of evaluating the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ by approximating this so-called response surface using such techniques as metamodels or radial basis functions(RBF) [66, 49] and Gaussian process (GP) models [32, 31, 37, 55]. The majority of these methods do not exploit derivative (of the parameter-to-observable map) information, which is our goal here.

Here, we choose to directly approximate the posterior using a Hessian-based GP response surface. This results in an inexpensive-to-evaluate explicit response surface equipped with an analytical uncertainty estimate. Thus, this "surrogate posterior density" can be sampled, using MCMC, for example, at negligible cost compared to sampling the original posterior density. The task of solving a statistical inverse problem therefore reduces to approximating a function over high dimensional parameter space, for which one has to face the curse of dimensionality.

The term "curse of dimensionality" was coined by Bellman [5] in the context of optimization to reflect the fact that, in order to obtain an accurate minimizer within ε tolerance, an exponential number of function evaluations, i.e., $(\frac{1}{\varepsilon})^N$, is required if our knowledge about the cost function is limited, for example, to Lipschitz continuity. A similar curse of dimensionality in function approximation says that an exponential number of function evaluations, i.e., $(\frac{1}{\varepsilon})^N$, is necessary for the approximation to be uniformly accurate within ε tolerance if Lipschitz continuity is all we know about the approximated function [21].

In the context of statistics, the curse of dimensionality reflects the fact that high dimensional spaces are very sparse [63]. For example, the ratio of the volume of the inscribed hypersphere and that of the corresponding hypercube converges to zero as N approaches infinity. Another example is that the volume of a thin shell between hyperspheres of radii r and $r - \epsilon$ converges to the volume of the hypersphere of radius r as N approaches infinity no matter how small ϵ is. These two examples show that the volume content of hypercubes and hyperspheres concentrates near their surfaces. That is, the center of these objects is more or less empty. A concrete example of the sparsity in high dimensional space is the hypercube $[-1, 1]^{10}$, whose first quadrant contains only the fraction 2^{-10} (2^{-N} for N-dimensional space) of uniform sampling data. Furthermore, almost no samples can be found in the inscribed hypersphere.

The problem of approximating a pdf in high dimensions by sample points is a good example of this effect. Since the integral of a bonafide pdf over the domain of interest is at most unity, the pdf must be negligible everywhere except in the neighborhood of the modes. In addition, if the modes are located away from the boundaries of the domain of interest (which is true for most practical applications in which we choose the domain of interest to be sufficiently large to contain all the important features of the problem under consideration), random sampling methods (especially space-filling techniques) will tend to fail to find the high probability regions, since almost no samples will be in the neighborhood of these regions. In other words, the pdf at the sampling points will most likely be close to zero, and hence any reasonable estimation

or interpolation methods based on these values will yield flat response surfaces whose values are close to zero.

Nevertheless, the curse of dimensionality is not entirely a pessimistic result. In fact, it implies that one might be able to reduce its impact if higher order information, for example, gradients and Hessians of the pdf, is exploited. This has indeed been the case for optimization of systems governed by PDEs (i.e., PDE-constrained optimization), where the combination of (Hessian-based) inexact Newton methods with appropriate preconditioners yields methods that can deliver solutions at the cost of a constant number of forward PDE solves, independent of the dimension of the optimization variable space (e.g., [7, 6, 22, 34]). That is, using a suitable class of Newton methods for optimization and under favourable conditions, the curse of dimensionality in optimization can be mitigated, at least for locating local minima.

A natural idea is therefore to cast the density approximation problem as an optimization problem for which the effect of the curse of dimensionality can be lessened by employing higher order derivatives. In particular, we pose the sampling task (i.e., the task of selecting training points) for the GP approximation as a sequence of optimization problems (solved by Newton methods) that seek to maximize the error between the GP approximation and the underlying true pdf. These points also tend to be points of high probability density of the underlying pdf. Moreover, a "piecewise" Gaussian approximation to the underlying pdf is adaptively constructed with local covariance matrices that are inverses of Hessians of the negative log posterior evaluated at the interpolation points. (As is well known, when the parameter-to-observable map $\mathbf{f}(\mathbf{m})$ is linearized, the posterior covariance matrix is equivalent to the inverse of this Hessian.) This proposed Hessian-based GP method for Bayesian interpolation exploits previous work on adaptive choice of interpolation points in reduced model construction using a greedy algorithm [15, 11, 28].

The remainder of the paper is organized as follows. Section 2 reviews important characteristics of conventional GP response surfaces, among which are the equivalence with RBF approximation and with Bayesian interpolation. This motivates us to develop a nonstationary Hessian-based GP in section 3, followed by a heuristic adaptive sampling strategy for computer experimental design in section 4. Next, section 5 provides an analysis of our proposed approach. Section 6 details the choice of the error function, numerical optimization methods, initialization, and how to update the training set. Verification of the proposed response surface methodology is carried out in section 7 for several probability density approximation problems, including the problem of Bayesian inference of the shape of a scatterer from noisy observations of scattered electromagnetic waves. Finally, section 8 concludes the paper and discusses some ongoing research issues.

2. Standard GP response surfaces. We start by reviewing the standard GP response surface methodology. In order to avoid unnecessary confusion with the Bayesian inversion described above, we rename the posterior density solution of the inverse problem $\pi_{\text{post}}(\mathbf{m})$ as $d(\mathbf{m})$, for which we seek to find an approximation. Assume we are given a training set $\{\mathbf{m}_i, d_i = d(\mathbf{m}_i)\}_{i=1}^n$, where $\mathbf{m}_i \in \mathbb{R}^N$ is a point (training site) in the N-dimensional parameter space and its corresponding function evaluation is d_i. If the training set is noise-free, which is the case in this paper since we simply evaluate $d(\mathbf{m}_i)$, the GP response surface method is a Bayesian interpolation technique that aims to statistically interpolate the unknown underlying function $d(\mathbf{m})$ given the training set [58]. Once the observable data (the training points) are obtained, they are combined with the GP prior through

a Bayesian framework to produce a prediction for the unknown function $d(\mathbf{m})$, as we shall show.

By definition, a random function $d(\mathbf{m})$ is a GP if the marginal density $\pi(d(\mathbf{m}_1), d(\mathbf{m}_2), \ldots, d(\mathbf{m}_n))$ is a multivariate Gaussian, for any set of points $\{\mathbf{m}_i\}_{i=1}^n$. A GP is completely determined by a mean function $\mu(\mathbf{m})$ and a covariance function $k(\mathbf{m}_i, \mathbf{m}_j)$ for two arbitrary points $\mathbf{m}_i$ and $\mathbf{m}_j$. Assume for now that these functions are given (their constructions are the subjects of sections 3.1 and 3.2). By assigning a GP prior on the random function $d(\mathbf{m})$, the joint distribution of $d(\mathbf{m}^*)$ with n observations $\mathbf{d}_{obs} = [d(\mathbf{m}_1), d(\mathbf{m}_2), \ldots, d(\mathbf{m}_n)]^T$ at n training points $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_n]$ is a Gaussian given by

$$\pi(d(\mathbf{m}^*), \mathbf{d}_{obs} | \mathbf{M}, \mathbf{m}^*) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{obs} \\ \mu(\mathbf{m}^*) \end{bmatrix}, \begin{bmatrix} K(\mathbf{M}, \mathbf{M}) & K(\mathbf{M}, \mathbf{m}^*) \\ K^T(\mathbf{M}, \mathbf{m}^*) & k(\mathbf{m}^*, \mathbf{m}^*) \end{bmatrix} \right), \tag{2.1}$$

where the matrix $K(\mathbf{M}, \mathbf{M})$ is computed as $K_{ij} = k(\mathbf{m}_i, \mathbf{m}_j)$, and $k(\mathbf{m}_i, \mathbf{m}^*)$ is the ith element of the column vector $K(\mathbf{M}, \mathbf{m}^*)$. The availability of the training points is assumed for now, and in section 4 we will present an adaptive sampling method to select these points. Using conditional distribution of the multivariate normal [58, 61], the posterior distribution of $d(\mathbf{m}^*)$ is given by

$$\pi_{\text{post}}(d(\mathbf{m}^*) | \mathbf{M}, \mathbf{d}_{obs}, \mathbf{m}^*) = \mathcal{N}(E\left\{d(\mathbf{m}^*)\right\}_{\text{post}}, var\left\{d(\mathbf{m}^*)\right\}_{\text{post}}), \tag{2.2}$$

where the expectation and variance read

$$E\left\{d(\mathbf{m}^*)\right\}_{\text{post}} = \underbrace{\mu(\mathbf{m}^*)}_{\mu(\mathbf{m}^*)_{\text{prior}}} + K^T(\mathbf{M}, \mathbf{m}^*)\left[K(\mathbf{M}, \mathbf{M})\right]^{-1}(\mathbf{d}_{obs} - \boldsymbol{\mu}_{obs}), \tag{2.3}$$

$$var\left\{d(\mathbf{m}^*)\right\}_{\text{post}} = \underbrace{k(\mathbf{m}^*, \mathbf{m}^*)}_{var\{d(\mathbf{m}^*)\}_{\text{prior}}} - K^T(\mathbf{M}, \mathbf{m}^*)\left[K(\mathbf{M}, \mathbf{M})\right]^{-1} K(\mathbf{M}, \mathbf{m}^*). \tag{2.4}$$

Since $\mathbf{m}^*$ is arbitrary, (2.2) is the posterior distribution of $d(\mathbf{m})$ at any $\mathbf{m}$ in the parameter space. The Bayesian interpretation is now clear as follows. Equation (2.3) states that the posterior mean function at $\mathbf{m}^*$ is the corrected version of the prior mean function using the observation (or measurement) information encoded in the second term on the right-hand side. Furthermore, the posterior error bar (or the posterior uncertainty) is reduced once the prior knowledge and observations are combined as shown in (2.4). Indeed, since the covariance matrix $K(\mathbf{M}, \mathbf{M})$ is symmetric positive definite, and hence its inverse, the second term on the right-hand side, is positive, we have $var\left\{d(\mathbf{m}^*)\right\}_{\text{post}} \leq var\left\{d(\mathbf{m}^*)\right\}_{\text{prior}}$.

We now discuss some other properties of GP response surfaces that are useful for our subsequent developments. To begin, we introduce the mean squared prediction error (MSPE) with respect to a distribution. Following Santner, Williams, and Notz [61], we define

$$MSPE(\hat{d}(\mathbf{m}^*), F_n) = \mathbb{E}\left\{ [\hat{d}(\mathbf{m}^*) - d(\mathbf{m}^*)]^2 \right\}_{F_n}, \tag{2.5}$$

where $\hat{d}(\mathbf{m}^*)$ is a generic predictor of $d(\mathbf{m}^*)$ and F_n denotes the joint distribution of $(\mathbf{d}_{obs}, d(\mathbf{m}^*))$, i.e., the distribution in (2.1). The following theorem summarizes some important properties of GP response surfaces.

THEOREM 2.1. *The following properties hold for the GP defined in* (2.1):

(i) *The predicted mean function* (2.3) *is the unique minimizer of the MSPE with respect to joint distribution* (2.1). *Furthermore, it is a linear and unbiased predictor. That is, it is the best linear unbiased predictor (BLUP).*

(ii) *The predicted mean function* (2.3) *interpolates the unknown functions at all points in the training set* $\mathbf{M}$.

(iii) *The MSPE incurred by the predicted mean function is exactly the variance* (2.4).

Proof. See Santner, Williams, and Notz [61] for a proof. ∎

The first assertion of Theorem 2.1 therefore suggests that one should use the predicted mean function (2.3) as the predictor for the unknown random function $d(\mathbf{m})$. The second assertion implies that the variance (2.4) is zero at all the training points; that is, the predicted uncertainty at the training points is zero. Hence all the random functions generated by (2.2) interpolate the observed data. Moreover, the third assertion implies that the posterior variance (2.4) can be used as a measure of uncertainty for the GP predictor (2.3).

We next relate the mean predictor (2.3) with radial basis interpolations. If the covariance function $k(\cdot,\cdot)$ is of RBF type, the predicted mean function can be shown to be a radial basis interpolation as follows [58]. Let us define

$$\boldsymbol{\alpha} = \left[K(\mathbf{M},\mathbf{M})\right]^{-1}(\mathbf{d}_{obs} - \boldsymbol{\mu}_{obs}).$$

Substituting $\boldsymbol{\alpha}$ into (2.3), we obtain

$$\hat{d}_n(\mathbf{m}^*) = \mu(\mathbf{m}^*) + \sum_{i=1}^{n} \alpha_i k(\mathbf{m}_i, \mathbf{m}^*), \tag{2.6}$$

which is a radial basis interpolation of the error between the predictor and the prior mean. As a result, the predictor inherits the regularity of the covariance function, assuming that the prior mean is sufficiently regular. The importance of the mean function and the covariance function is now clear. They reflect our prior knowledge about what the unknown function $d(\mathbf{m})$ is likely to be. For example, if the underlying function is expected to be not very nonlinear and infinitely smooth, the mean function can be chosen to be linear, and a squared exponential function (also known as a Gaussian kernel),

$$k(\mathbf{m}_i, \mathbf{m}_j) = \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{\Sigma}^{-1}}\right), \tag{2.7}$$

can be used as the covariance function. Here, we have defined the Mahalanobis norm as

$$\|\mathbf{m}_i - \mathbf{m}_j\|_{\boldsymbol{\Sigma}^{-1}} = \sqrt{(\mathbf{m}_i - \mathbf{m}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{m}_i - \mathbf{m}_j)}, \tag{2.8}$$

with a positive definite matrix $\boldsymbol{\Sigma}$. Typically, $\boldsymbol{\Sigma}$ is chosen to be a constant diagonal matrix whose diagonal entries are inferred from the training data [43, 50]. Each diagonal entry corresponding to each dimension can be considered as the length scale over which the predictor changes significantly in that particular dimension. Statistically, these length scales determine distance between two points in each dimension such that the predictions at these points are uncorrelated. In other words, these length scales reflect our beliefs about the smoothness of the unknown function $d(\mathbf{m})$. For example, if $\boldsymbol{\Sigma}_{ii}$, the ith diagonal entry, is small, the predictor varies rapidly in the ith dimension, while it tends to be constant for large $\boldsymbol{\Sigma}_{ii}$.

The key connection (2.6) between the RBF interpolation, a popular kernel density approximation [63], and the mean (2.3) of the posterior GP will be explored in

this paper. In particular, the mean of the posterior GP will be used as a nonparametric kernel density estimator of the underlying unknown probability density $d(\mathbf{m})$. That is, we utilize the GP methodology as a statistical tool to derive a kernel density approximation. Consequently, we can view $d(\mathbf{m})$ as a random function distributed by the posterior GP. This statistical interpretation should be understood as a mathematical by-product of our approach, which does not necessarily make sense physically. Therefore, one may take advantage of this feature or simply ignore it.

It should be noted that one can interpolate the logarithm of the posterior pdf $d(\mathbf{m})$ instead of itself, but this is not the point of this paper. As mentioned above, we start by making the connection between RBF interpolation and the mean of the posterior GP. The natural idea, in the spirit of this connection, is therefore to interpolate the posterior density $d(\mathbf{m})$. More importantly, this connection also suggests that we might reduce the impact of the curse of dimensionality by using an adaptive kernel. The construction of our covariance function is solely based on this observation (namely, (2.6)) and therefore needs to be modified if one likes to interpolate $\log(d(\mathbf{m}))$ instead. How to modify our current construction is beyond the scope of this paper.

3. Nonstationary adaptive Hessian-based GP response surfaces.

3.1. Prior mean construction. We next discuss how to choose the prior mean function, and postpone the construction of the covariance function to section 3.2. For our problem of interest where the unknown underlying function $d(\mathbf{m})$ is a probability density, its mean value can be estimated as

$$d_{\text{mean}} = \frac{\int_\Omega d(\mathbf{m})\, d\Omega}{m(\Omega)} \leq \frac{1}{m(\Omega)},$$

where $m(\Omega)$ denotes the measure of Ω, and the inequality is obtained from the fact that the domain of interest Ω is a subset of high dimensional parameter spaces over which $d(\mathbf{m})$ is a *bona fide* density, i.e., $\int_{\mathbb{R}^N} d(\mathbf{m})\, d\Omega = 1$. Clearly, the mean value d_{mean} is small if the measure of the domain of interest is sufficiently large. We therefore expect that zero-mean is a good prior information. This is intuitively meaningful since $d(\mathbf{m})$, the Bayesian posterior probability density, is typically significant only in the neighborhood of the modes, while it tends to be small or close to zero elsewhere. On the other hand, since the approximation approaches the prior mean for points that are further away from the training set, the zero-mean prior permits reasonable approximations for regions with small probability density.

It should be pointed out that we have ignored the normalized constant in the Bayesian solution (1.1), and hence $d(\mathbf{m})$ is not a bona fide density. Nevertheless, $d(\mathbf{m})$ is an exponential function with negative exponent, as shown in (1.1), and consequently $d(\mathbf{m})$ tends to be small away from its modes or when Ω is sufficiently "large." This suggests that taking zero mean for the prior GP is a sensible choice, at least in regions where $d(\mathbf{m})$ is sufficiently small. More importantly, this zero-mean prior facilitates our goal of finding modes of the underlying density $d(\mathbf{m})$, as we shall argue in section 6.

3.2. Adaptive nonstationary covariance function. Covariance functions that are a function of only relative distance between two points, e.g., (2.7) with constant $\boldsymbol{\Sigma}$, are known as stationary covariance functions. However, Gaussian processes with stationary covariance function can provide accurate predictors only for functions with nearly constant smoothness, since stationariness lacks the ability to adapt to variable smoothness of the unknown function of interest. In the context of RBF approximation, stationary means translating a fixed kernel in the predictor, and

this faces the curse of dimensionality. Adapting the kernel is a well-known solution for reducing the impact of the curse of dimensionality, and this suggests that we use nonstationary Gaussian processes. A number of nonstationary covariance functions have been devised in the literature; see [58, 56, 27, 33, 62, 60, 53, 52, 24, 38, 57] for examples. Below, we rationalize the derivation of our Hessian-based nonstationary covariance functions.

We begin by reexamining the predictor (2.6) with zero-mean GP prior as argued in section 3.1:

$$\hat{d}_n(\mathbf{m}^*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{m}_i, \mathbf{m}^*). \tag{3.1}$$

If $n = 1$ and if the covariance is of Gaussian type as in (2.7), then the predictor in (3.1) is nothing more than a Gaussian approximation to $d(\mathbf{m})$, where the covariance matrix is given by $\boldsymbol{\Sigma}$. If, in addition, $\boldsymbol{\Sigma}^{-1}$ is the Hessian of $-\ln d(\mathbf{m})$, i.e., $\boldsymbol{\Sigma}^{-1} = \nabla^2(-\ln d(\mathbf{m}))$, then the predictor becomes the popular Laplace approximation (see [44] and references therein). That is, the predictor is exact if the underlying density $d(\mathbf{m})$ is a Gaussian whose peak is at $\mathbf{m}_1$.

For $n > 1$, it is natural and intuitive to generalize the Laplace approximation idea by combining local Laplace approximations constructed in different subdomains. The challenge is how to combine them to form a global approximation. Our idea is the following. Since the Laplace approximation is locally an accurate approximation, the contribution of $k(\mathbf{m}_i, \mathbf{m}^*)$ to the predictor should dominate the other terms $k(\mathbf{m}_j, \mathbf{m}^*)$ for $j \neq i$ if $\mathbf{m}^*$ is closest to $\mathbf{m}_i$. In order to fulfill this goal, we introduce the following nonstationary covariance function:

$$k(\mathbf{m}_i, \mathbf{m}_j) = \sum_{l=1}^{L} P(z = l|\mathbf{m}_i) P(z = l|\mathbf{m}_j) \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right), \tag{3.2}$$

where $\boldsymbol{H}_l = \nabla^2(-\ln d(\mathbf{m}_l))$ and $L \leq n$ (to be shown in section 4). $P(z = l|\mathbf{m}_i)$ can be considered as the conditional probability of having selected the lth kernel $\exp(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l})$ given $\mathbf{m}_i$, and z is known as the latent indicator variable. For example, $P(z = l|\mathbf{m}_i)$ should approach 1 if $\mathbf{m}_i \to \mathbf{m}_l$, and zero if $\mathbf{m}_i$ is far away from $\mathbf{m}_l$. In particular, as derived in section 3.3, the following form of $P(z = l|\mathbf{m}_i)$ satisfies the requirements:

$$P(z = l|\mathbf{m}_i) = \frac{\exp(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_l\|^2_{\boldsymbol{H}_l})}{\sum_{p=1}^{L} \exp(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_p\|^2_{\boldsymbol{H}_p})}, \tag{3.3}$$

which can be seen as a special form of a well-known class of *soft-max* gating networks in the machine learning community [9]. Note that the denominator is just a normalized constant, while the numerator is a Gaussian with mean $\mathbf{m}_l$ and inverse covariance matrix $\boldsymbol{H}_l$.

The importance of the Hessian information of the underlying posterior $d(\mathbf{m})$ is now explained. It can be seen that the Hessians appear two times in the covariance function definition (3.2). First, the appearance in the kernels $\exp(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l})$ ensures that the predictor is the desired piecewise Laplace approximation. Second, the role of the Hessians in the membership probability $P(z = l|\mathbf{m}_i)$ is to guide the covariance function to pick the appropriate dominant kernel in the predictor. Note that the product $P(z = l|\mathbf{m}_i)P(z = l|\mathbf{m}_j)$ is necessary because it not only guarantees the symmetry of the covariance function but also determines the appropriate covariance

structure and hence the smoothness of the predictor. By definition, $k(\mathbf{m}_i, \mathbf{m}_j)$ is the covariance between $d(\mathbf{m}_i)$ and $d(\mathbf{m}_j)$. The covariance in turn encodes the smoothness of the predictor. Moreover, the smoothness of a surface is typically measured by its second derivatives, i.e., the Hessian. This suggests that the Hessians should be used in the covariance to shape the smoothness of the predictor accordingly. Our covariance function in (3.2) is built based on this intuition (and on the desire to have a piecewise Laplace approximation). That is, if $\mathbf{m}_i$ and $\mathbf{m}_j$ are close to $\mathbf{m}_l$, with respect to the norm $\|\cdot\|_{\boldsymbol{H}_l}$, the lth term of the sum on the right-hand side of (3.2) dominates the covariance, while its contribution to the covariance is small if either $\mathbf{m}_i$ or $\mathbf{m}_j$ is far away from $\mathbf{m}_l$.

From the point of view of the piecewise Laplace approximation, the membership probability serves as an automatic mechanism to partition the high dimensional parameter space into overlapping subregions over which the posterior density of the inverse problem is dominantly interpolated and approximated by local Gaussians. As a demonstration, Figure 3.1 shows three membership probabilities corresponding to three modes of the exact density $d(\mathbf{m}) = \frac{1}{3}\mathcal{N}(-4, 1) + \frac{1}{3}\mathcal{N}(0, 0.75) + \frac{1}{3}\mathcal{N}(4, 0.5)$. As can be seen, each membership probability is 1 if $\mathbf{m}_i$ is close to its corresponding mode and 0 otherwise. Subregions dominated by the first and the second membership probabilities do not overlap, while they do with that of the third one. This reflects the fact the first and second modes do not overlap, but they do with the third one. The role of the membership probabilities in automatically identifying the dominant local Laplace approximation is clearly demonstrated in this figure.

A question that needs to be addressed is whether (3.2) defines a valid covariance function. This is important since a GP exists if and only if the covariance function is valid, according to the Kolmogorov existence theorem [2]. The following result answers this question.

THEOREM 3.1. *Assume $L \geq 1$ and that, $\boldsymbol{H}_l$ is symmetric positive definite For all $l = 1, \ldots, L$. Then*

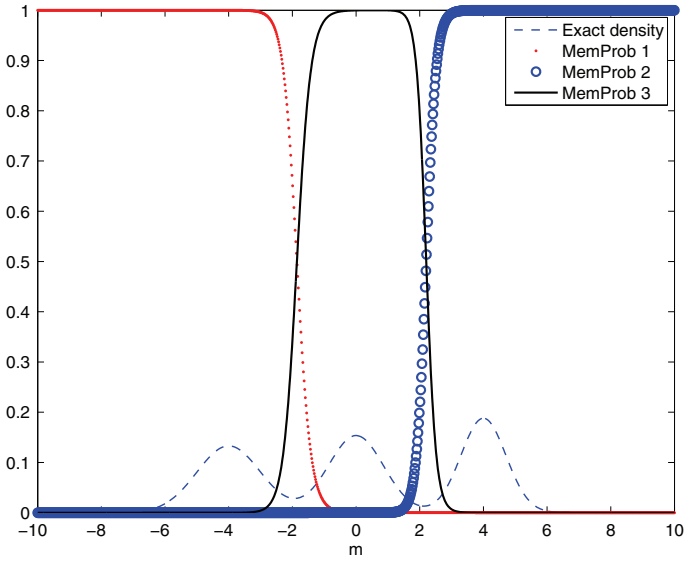


FIG. 3.1. *Three membership probabilities corresponding to three modes of the exact density* $d(\mathbf{m}) = \frac{1}{3}\mathcal{N}(-4, 1) + \frac{1}{3}\mathcal{N}(0, 0.75) + \frac{1}{3}\mathcal{N}(4, 0.5)$.

$$k(\mathbf{m}_i, \mathbf{m}_j) = \sum_{l=1}^{L} P(z = l|\mathbf{m}_i) P(z = l|\mathbf{m}_j) \exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right)$$

is a valid nonstationary covariance function.

Proof. The nonstationary aspect is clear since $k(\mathbf{m}_i, \mathbf{m}_j)$ is a function of not only the relative distance between $\mathbf{m}_i$ and $\mathbf{m}_j$ but also $\mathbf{m}_i$ and $\mathbf{m}_j$ themselves. The symmetry is also clear. In order to prove that the kernel $k(\mathbf{m}_i, \mathbf{m}_j)$ is positive definite, it is sufficient to show that the matrix $\mathbf{K}(\mathbf{M}, \mathbf{M})$ is positive definite $\forall n \geq 1$. We begin by the following fact

$$\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right) = \frac{1}{(\pi/2)^{N/2}|\boldsymbol{H}_l^{-1}|^{1/2}} \times \int_{\mathbb{R}^N} \exp\left(-\|\mathbf{m}_i - \mathbf{m}\|^2_{\boldsymbol{H}_l}\right) \exp\left(-\|\mathbf{m}_j - \mathbf{m}\|^2_{\boldsymbol{H}_l}\right) d\Omega.$$

Thus, $\forall n \geq 1, \forall \mathbf{c} \in \mathbb{R}^n$, we have

$$\mathbf{c}^T \mathbf{K}(\mathbf{M}, \mathbf{M})\mathbf{c} = \sum_i \sum_j c_i c_j k(\mathbf{m}_i, \mathbf{m}_j) = \sum_l \frac{1}{(\pi/2)^{N/2}|\boldsymbol{H}_l^{-1}|^{1/2}} \times \int_{\mathbb{R}^N} \left\{ \sum_i c_i P(z = l|\mathbf{m}_i) \exp\left(-\|\mathbf{m}_i - \mathbf{m}\|^2_{\boldsymbol{H}_l}\right) \right\}^2 d\Omega \geq 0.$$

It is clear that the equality happens if and only if the term in the curly brackets is zero almost everywhere, which in turn happens if and only if $c_i = 0\ \forall i = 1, \ldots, n$, and this completes the proof. ∎

The importance of Theorem 3.1 is now clear. It ensures the positive definiteness of $\mathbf{K}(\mathbf{M}, \mathbf{M})$, which in turn guarantees its invertibility, which is necessary for (2.3) and (2.4).

3.3. Derivation of the membership probabilities. Recall that the membership probability $P(z = l|\mathbf{m}_i)$ is the conditional probability of having selected the lth kernel $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_j\|^2_{\boldsymbol{H}_l}\right)$ given $\mathbf{m}_i$. Using the Bayes theorem gives

$$P(z = l|\mathbf{m}_i) = \frac{P(\mathbf{m}_i|z = l) \times P(z = l)}{\sum_{p=1}^{L} P(\mathbf{m}_i|z = p) \times P(z = p)},$$

where $P(\mathbf{m}_i|z = l)$ and $P(z = l)$ are the likelihood and prior, respectively of selecting the lth kernel. Since our prior knowledge is vague, we choose $P(z = l) = 1/L\ \forall l = 1, \ldots, L$. The likelihood is chosen to be the (unnormalized) Gaussian $\exp\left(-\frac{1}{2}\|\mathbf{m}_i - \mathbf{m}_l\|^2_{\boldsymbol{H}_l}\right)$, which reflects our desire that the likelihood must be large, in the Mahalanobis norm $\|\cdot\|_{\boldsymbol{H}_l}$, as $\mathbf{m}_i$ approaches the mean $\mathbf{m}_l$. The extent to which the likelihood is still significant is determined by the curvature of the true unknown $d(\mathbf{m})$, which appears as the inverse of the likelihood covariance matrix. Hence, the final form of the membership probability is given as in (3.3).

It should be pointed out that the prior probability $P(z = l) = 1/L\ \forall l = 1, \ldots, L$ is a noninformative belief, and the likelihood is solely based on our intuition. One can replace our above belief or the likelihood model to obtain different membership probabilities. More sophisticated tools such as a hierarchical Bayesian model can be employed to determine the membership probabilities, but to simplify the proposed approach, we avoid more advanced techniques unless absolutely necessary.

4. Sequential adaptive sampling strategy. This section addresses the computer experimental design issue on how to look for the training points adaptively. The method we are going to describe follows our previous work on scalable adaptive algorithms for constructing reduced models in highdimensional parameter spaces [15, 11]. To begin, we define the following generic error function:

$$\mathcal{G}(\mathbf{m}, n), \tag{4.1}$$

which is a function of parameters $\mathbf{m}$ and the training set size n (and the training set itself, but that is omitted here for simplicity). The error could be, for example, the squared error between the true function $d(\mathbf{m})$ and the predictor $\hat{d}_n(\mathbf{m})$,

$$\mathcal{G}(\mathbf{m}, n) = [\hat{d}_n(\mathbf{m}) - d(\mathbf{m})]^2, \tag{4.2}$$

or the predictive variance in (2.4) as the error indicator. Generally, for Algorithm 1 and its corresponding theory to work, it is desirable that the cost $\mathcal{G}(\mathbf{m}, n)$ be less then some small tolerance ε at all training points. The squared error and the predictive variance clearly satisfy this requirement since they are zero at all the training points. We first outline the adaptive sampling algorithm as follows.

ALGORITHM 1. ADAPTIVE SAMPLING ALGORITHM.

1. *Given a set of training points* $\{\mathbf{m}_i, d_i = d(\mathbf{m}_i)\}_{i=1}^n$*, solve the optimization problem*

$$\max_{\mathbf{m}\in\Omega} \mathcal{G}(\mathbf{m}, n) \tag{4.3}$$

   to find the location in parameter space at which the error is maximized; i.e., find $\mathbf{m}^* = \arg\max \mathcal{G}(\mathbf{m}, n)$.
2. *If* $\mathcal{G}(\mathbf{m}^*, n) < \varepsilon$*, where* ε *is the desired level of accuracy, then terminate the algorithm. If not, go to the next step.*
3. *With* $\mathbf{m} = \mathbf{m}^*$*, compute the true function* $d(\mathbf{m}^*)$*. Update the predictor. Go to step* 1.

The first step of the adaptive sampling algorithm incrementally finds the next training points at locations where $\mathcal{G}(\mathbf{m}, n)$, as a measure of the error between the underlying true posterior density and the current GP response surface approximation, is maximized. The structure of the optimization problem that must be solved in each adaptive cycle is similar to that of PDE-constrained inverse problems, and hence, many of the associated tools for large-scale optimization can be recruited, particularly Newton-CG solvers, trust-region globalization, and Eisenstat–Walker inexactness, e.g., [4, 7, 6, 3, 22, 11]. In order to make Algorithm 1 well-defined, as discussed in the next section, the initial guess is admissible if the error function is at least ε.

If the maximum error is less than the prescribed tolerance ε in step 2, the algorithm stops. Otherwise, step 3 will update the current predictor and return to step 1. In particular, the training set is updated using the maximizer $\mathbf{m}^*$ and its corresponding true posterior value $d(\mathbf{m}^*)$. If the Hessian of the negative log posterior, $\nabla^2(-\ln d(\mathbf{m}^*))$, is positive definite, the adaptive covariance function (3.2) and the membership probability (3.3) are updated by increasing L by 1. That is, we build a local Gaussian approximation whose inverse covariance is the local Hessian of the negative log posterior. This local Gaussian approximation is then used to update the GP covariance function and the membership probability. Effectively, the method builds an adaptive sparse nonstationary GP that is generally improved after each

cycle. The method identifies regions in high dimensional spaces where the discrepancy of the response surface is maximal, and then inserts local Gaussian approximations at those points to drive the response surface error down. Since the GP response surface is interpolating, the resulting approximation is identical to the posterior density at these points. Furthermore, the Hessian ensures that the response surface locally adapts to the shape of posterior $d(\mathbf{m})$ accurately.

A question that needs to be addressed is the existence of the first optimizer $\mathbf{m}^*$ at which the Hessian is (semi-)positive definite. In general, a probability density may not have any mode at all. However, the density of interest in this paper is coming from a Bayesian solution given by (1.1), which is the exponential function of the negative regularized data misfit. As a result, we can view the prior as the regularization, and we are free to tune it (this is done routinely in the context of deterministic inversions) to make the inverse problem well-posed, hence making the regularized data misfit be more like a parabola (at least locally). In other words, we can always assume that the first $\mathbf{m}^*$ with positive definite Hessian exists (by making the contribution of the prior larger, for example), and Algorithm 1 is therefore meaningful. Note that we also accept $\mathbf{m}^*$ at which the Hessian is semipositive definite, and this includes the case in which $d(\mathbf{m})$ is insensitive to some parameters.

5. An analysis of the adaptive GP response surface. In this section, each adaptive cycle in Algorithm 1 is analyzed to show that the whole algorithm is well defined. We first show that the optimization problem (4.3) has a solution under suitable assumptions.

PROPOSITION 5.1. *Assume that $\mathcal{G}(\mathbf{m}, n)$ is a continuous function of $\mathbf{m} \in \Omega$, where Ω is a closed and bounded subset of $\mathbb{R}^N$. Then there exists a solution for the optimization problem* (4.3).

Proof. See [15, 11] for a proof. ∎

The closedness and boundedness of the domain of interest Ω are reasonable. For example, in the shape inverse electromagnetic scattering problem studied in section 7, the shape parameters $\mathbf{m}$ are bounded due to our prior belief in the boundedness of the shape. Proposition 5.1 therefore implies that steps 2 and 3 of Algorithm 1 always go through; thus each cycle certainly finishes.

Revisiting previous sampled points is an expensive task requiring forward and adjoint solves, and hence should be avoided. On the other hand, distinction of sampled points implies the nonsingularity of $\mathbf{K}(\mathbf{M}, \mathbf{M})$, which is vital in ensuring the existence and uniqueness of the the predictor and its uncertainty in (2.3)–(2.4). Our next result shows that in fact Algorithm 1 always finds new sampling points.

THEOREM 5.2. *Algorithm* 1 *is well defined in the sense that it terminates in finite time and that all sampled points are distinct.*

Proof. See [15, 11] for a proof. ∎

The next question that needs to be resolved is whether the proposed adaptive training (also known as active learning) can systematically bias the inference. Since Bayesian inference is consistent with the *likelihood principle* [42] which states that the inference should depend only on the likelihood of the data that is actually observed, one is free to choose training points without introducing any bias to the inference.

We next show that our piecewise Laplace (Gaussian) approximation (2.3) improves as the number of training points increases.

THEOREM 5.3. *Denote $\hat{d}_n(\mathbf{m}^*)$ as the mean predictor* (2.3)*, and assume $\hat{d}_{n+1}(\mathbf{m}^*) \neq \hat{d}_n(\mathbf{m}^*)$ $\forall n \in \mathbb{N}$. As the number of training points increases, the MSPE decreases in the following sense:*

$$\mathbb{E}\left\{[\hat{d}_{n+1}(\mathbf{m}^*) - d(\mathbf{m}^*)]^2\right\}_{F_{n+1}} < \mathbb{E}\left\{[\hat{d}_n(\mathbf{m}^*) - d(\mathbf{m}^*)]^2\right\}_{F_n}.$$

Proof. We have the following inequalities for the MSPE:

$$\begin{aligned}\mathbb{E}\left\{[\hat{d}_{n+1}(\mathbf{m}^*) - d(\mathbf{m}^*)]^2\right\}_{F_{n+1}} &\leq \mathbb{E}\left\{[\hat{d}(\mathbf{m}^*) - d(\mathbf{m}^*)]^2\right\}_{F_{n+1}} \\ &< \mathbb{E}\left\{[\hat{d}_n(\mathbf{m}^*) - d(\mathbf{m}^*)]^2\right\}_{F_n},\end{aligned}$$

where $\hat{d}(\mathbf{m}^*)$ denotes an arbitrary linear predictor. The first inequality holds true due to the minimization property of $\hat{d}_{n+1}(\mathbf{m}^*)$ in the first part of Theorem 2.1. The second inequality follows from choosing $\hat{d}(\mathbf{m}^*) = \hat{d}_n(\mathbf{m}^*)$ and applying the marginalization property of the multivariate Gaussian (2.1). Note that the second inequality is strict due to the assumption $\hat{d}_{n+1}(\mathbf{m}^*) \neq \hat{d}_n(\mathbf{m}^*)$ and the uniqueness of the minimizer $\hat{d}_{n+1}(\mathbf{m}^*)$. ∎

6. Error function, numerical optimization, initialization, and training set update. The active learning method proposed in section 5 works for a class of quite general error functions. For our purpose, the true error $[\hat{d}_n(\mathbf{m}) - d(\mathbf{m})]^2$ turns out to be a good candidate, as we now explain. Recall that the main goal of this paper is to find as many modes as possible and then to interpolate the expensive-to-evaluate posterior density function $d(\mathbf{m})$ using a piecewise Laplace approximation. Intuitively, the interpolation is statistically more accurate if it captures most significant probability regions of $d(\mathbf{m})$. In order to fulfill this goal heuristically, we design Algorithm 1 to place training points where the discrepancy between the predictor and true function is largest (at least locally). Due to the local Gaussian nature of the predictor, if $\mathbf{m}$ is sufficiently far away from $\mathbf{m}_l$ $\forall l = 1, \ldots, L$, the predictor $\hat{d}_n(\mathbf{m})$ will approach the prior mean, which is zero, and hence the true error $[\hat{d}_n(\mathbf{m}) - d(\mathbf{m})]^2$ will approach $[d(\mathbf{m})]^2$. As a result, the worst-case scenario error found in each adaptive cycle is most likely a mode of $d(\mathbf{m})$.

Initially $\mathbf{M} = \emptyset$, and we choose $\hat{d}_0(\mathbf{m}) = 0$; hence $\mathcal{G}(\mathbf{m}) = (d(\mathbf{m}))^2$. Therefore, we in fact search for a mode, e.g., $\mathbf{m}^*$, of the underlying density $d(\mathbf{m})$ in the first step. Next, we describe in detail how to carry out the task of finding $\mathbf{m}^*$ using a scalable numerical optimization technique.

As discussed in section 5, Ω is typically generated by simple bound constraints on parameters $\mathbf{m}$. Similar to our previous work [15, 11], we choose to solve the bound-constrained optimization

$$\max_{\mathbf{m}} \mathcal{G}(\mathbf{m}), \tag{6.1}$$

subject to

$$\mathbf{m}_{min} \leq \mathbf{m} \leq \mathbf{m}_{max}, \tag{6.2}$$

using a subspace trust region interior reflective inexact Newton–CG method described in [11].

Initialization is one of the important factors determining the cost that the optimization solver takes to converge. In particular, if the initial guess is far away from the optimizer, it might take several iterations for the optimization solver to move to the basin of attraction where the desired convergence rate takes place, e.g., quadratic, if a Newton method is employed. Therefore in order to reduce the cost, it is vital to

find a good initial guess for the optimization problem in each greedy cycle. Clearly, the simplest way to do this is to take a random initialization, which is most likely not to be close to the basin of attraction of any local optimizers. Since the MSPE (2.4) is analytical and cheap to evaluate, another straightforward idea is to find the point where the MSPE is maximized and then use it as the initial guess. However, as noticed by MacKay [42], maximizers of MSPE tend to be at the boundary of the domain Ω, which is not of interest to us.

In the context of active learning (adaptive sampling) for GP, Seo et al. [65] numerically show that the selection criteria of Cohn [19] yields a more accurate predictor than that proposed by MacKay [42]. The reason is that the Cohn criteria for sampling aims to minimize the mean squared error (MSE), and in particular, it maximizes the average reduction in predictive variance. Nevertheless, adaptive sampling using Cohn criteria is expensive, as discussed in Christen and Sanso [18]. The latter authors propose a cheap alternative, an approximation to the Cohn criteria, over a test set $\mathbf{M}^a = [\mathbf{m}_1^a, \ldots, \mathbf{m}_{n^a}^a]$ of (random or grid) points in the parameter space. Specifically, the selection process is based on the solution of the following optimization problem:

$$\max_{\mathbf{m}_i^a \in \mathbf{M}^a} \mathcal{J}(\mathbf{m}_i^a), \tag{6.3}$$

where

$$\mathcal{J}(\mathbf{m}_i^a) = \frac{\frac{1}{n^a}\sum_{j=1}^{n^a} k(\mathbf{m}_j^a, \mathbf{m}_i^a)^2 + \frac{1}{n^a} I_{\mathbf{M}}^1}{I_{\mathbf{M}}^2 + \sqrt{\sum_{j=1}^{n} k(\mathbf{m}_i^a, \mathbf{m}_j)^2}},$$

with $\mathbf{m}_j \in \mathbf{M}$ and $\mathbf{m}_j^a \in \mathbf{M}^a$; $I_{\mathbf{M}}^1$ and $I_{\mathbf{M}}^2$ are two constants independent of $\mathbf{m}_i^a$ and are defined as

$$I_{\mathbf{M}}^1 = \sum_{i=1}^{n^a}\sum_{j=1}^{n} k(\mathbf{m}_i^a, \mathbf{m}_j)^2, \quad I_{\mathbf{M}}^2 = \max_{\mathbf{m}_i \in \mathbf{M}} \sum_{j=1}^{n} |k(\mathbf{m}_i, \mathbf{m}_j)|.$$

Due to the numerator, the maximizer of the cost in (6.3) should have high predictive variance and be highly correlated with other points in $\mathbf{M}^a$. At the same time, it should not be so close, and hence less correlated, to the current training set $\mathbf{M}$, due to the denominator. Meanwhile, an approach for near-optimal training points has been proposed [40]. However, one has to solve a combinatorial optimization, which we try to avoid here since it is prone to the curse of dimensionality. After all, we need a cheap and reasonably good initial guess, and then we devote our effort to the continuous optimization problem (4.3) using an efficient and scalable optimization solver, which is designed to be immune to the curse of dimensionality. Keeping this goal in mind, we choose the solution of (6.3) as the initial guess for three reasons besides its fast evaluation. First, it is not so close to the current training set about which we have already learned. Second, since it has high predictive variance it could belong to some region in parameter space where the error may be large. Third, due to its high correlation to other points in $\mathbf{M}^a$, learning this point could provide us information about other unvisited points as well.

The question we would like to address next is how to update the training set $\mathbf{M}$. As discussed in section 4, the covariance function and the membership probabilities may not be updated after a greedy cycle, depending on whether the Hessian of the negative log posterior $\nabla^2(-\ln d(\mathbf{m}^*))$ is (semi-)positive definite or not. In contrast,

the training set $\mathbf{M}$ is always updated after each greedy cycle. One could simply add the maximizer of the optimization problem (4.3) where we locally observe the largest error. This point is currently and intuitively the best for learning about the unknown function $d(\mathbf{m})$. However, in addition to the maximizer, the numerical optimization solver also supplies a whole trajectory of points starting from the initial guess to the optimizer where the unknown function $d(\mathbf{m})$ is evaluated. These points therefore contain information about $d(\mathbf{m})$ about which we are trying to learn. This suggests that we should add the whole trajectory to the current training set. The trade-off is that the condition number of the covariance matrix $K(\mathbf{M}, \mathbf{M})$ may increase, and methods for inverting the covariance matrix accurately and efficiently have been addressed elsewhere [26]. Here, we simply use the Cholesky decomposition.

7. Numerical experiments. In this section, the proposed Hessian-based GP predictor is compared to the state-of-the-art RBF interpolation [10]. Since the standard stationary GP predictor can be shown to be equivalent to a RBF whose kernel function is the same as the GP covariance function [58], we just need to compare our method to a standard GP predictor. To be fair, we also adapt the shape parameter of the RBF using the maximum likelihood. To ensure that the cost of generating our GP predictor and the cost of generating the RBF predictor are more or less the same, the number of function evaluations is forced to be the same for both methods. In particular, the number of function evaluations for the Hessian-based GP predictor is counted as the following. First, each forward solve is counted as one. Second, each gradient computation which requires an adjoint solve is counted as one, assuming that the costs of solving the forward and the adjoint are the same. Third, each Hessian-vector product required in the CG iterations is counted as two since a forward-like and an adjoint-like equation have to be solved. Thus, to be fair, if the Hessian-based GP uses n_F functions evaluations, the RBF will have n_F interpolation points. As a popular choice, the Latin hypercube (LHC) sampling is used to generate interpolation points for the RBF approach. Finally, to assess the quality of the predictors, we use several popular discrete norms, namely, the MSE, the ℓ^1-norm, the ℓ^2-norm, and the Hellinger norm [17].

7.1. One-dimensional examples. The first example considered in this subsection is a mixture of three Gaussians given by

$$d(\mathbf{m}) = \frac{1}{3}\mathcal{N}(-4, 1) + \frac{1}{3}\mathcal{N}(0, 0.75) + \frac{1}{3}\mathcal{N}(4, 0.5).$$

Figure 7.1 shows the GP predictor together with its uncertainty given by a 95% credibility envelope. The quality of the GP predictor (3 greedy cycles) and the RBF predictor, both with 143 function counts, is shown in Table 7.1. As expected, the RBF predictor is more accurate than the GP predictor for a low dimensional problem and simple density $d(\mathbf{m})$. Note that since the squared error (4.2) is used as the cost in the adaptive sampling algorithm, the MSE tends to be smaller than other measures.

For more complicated one-dimensional density with anisotropy and localized features such as that in Figure 7.2, where the true density is given by

$$d(\mathbf{m}) = \sum_{l=0}^{5} \left(2^{5-l}/63\right) \mathcal{N}\left(\left[65 - \frac{96}{2^l}/21\right] x, \left(\frac{32^2}{63}/2^{2l}\right)\right),$$

the GP predictor (with 371 function counts after 7 greedy cycles) starts to be competitive, as can be seen in Table 7.2.

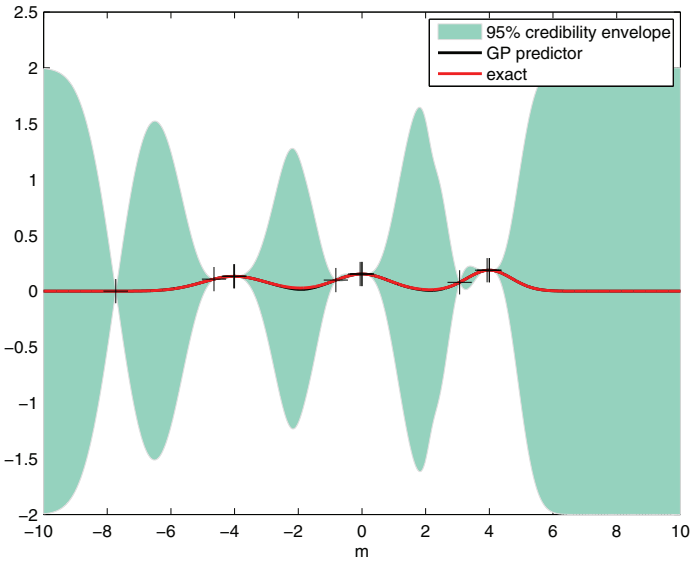


FIG. 7.1. *The GP predictor together with its uncertainty given by a* 95% *credibility envelope versus the exact. Crossed are the points in the training set* $\mathbf{M}$ *found by the greedy algorithm.*

TABLE 7.1
GP predictor error versus RBF predictor error over an LHC grid of 20,000 *points.*

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	6.55e-06	4.35	3.62e-01	1.74
RBF	3.77e-16	1.23e-02	2.75e-06	5.31e-03

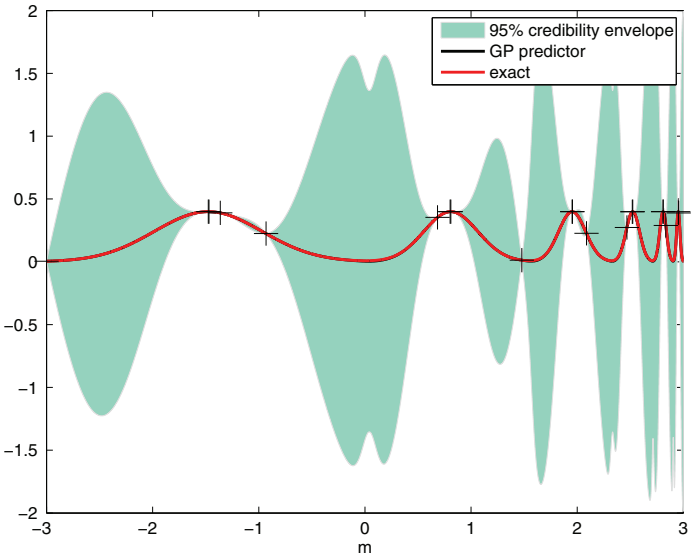


FIG. 7.2. *The GP predictor together with its uncertainty given by a* 95% *credibility envelope versus the exact. Crossed are the points in the training set* $\mathbf{M}$ *found by the greedy algorithm.*

As mentioned in section 6, the ability of our proposed method to seek the modes of the underlying density $d(\mathbf{m})$ can be observed in Figures 7.1 and 7.2. Again, it is

TABLE 7.2
GP predictor error versus RBF predictor error over an LHC grid of 20,000 *points.*

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	1.45e-06	3.15	1.70e-01	1.22
RBF	2.55e-3	2.31e1	7.14	1.04e1

important for our purpose that high probability density regions should be captured as much as possible.

One can observe in both Figures 7.1 and 7.2 that the posterior error bar (uncertainty) is small around the training points but quite large otherwise. As we discuss below (2.4), the posterior variance is smaller than the prior variance at the training points. However, further away from these training points, the former approaches the latter, which seems to be 2. Let us now explain why it is 2. In (3.2), the coefficient, $P(z = l|\mathbf{m}_i)P(z = l|\mathbf{m}_j)$, is the prior variance [58]. Since the maximum value of membership probabilities is 1, the prior variance is at most 1. In Figures 7.1 and 7.2, we plot the error bar as two times the posterior standard deviation, which is exactly two times the prior standard deviation away from the training points. This is the reason why the error has 2 as its maximum. A standard approach to tuning the prior variance [58], and hence having less conservative posterior error bars, is to introduce an extra factor, e.g., σ_f^2, in the prior variance as a hyperparameter, and then determine it, for example, using maximum likelihood. However, for simplicity of the exposition, we choose $\sigma_f = 1$ in this paper.

7.2. Two-dimensional example. We consider the following mixture of four anisotropic Gaussians:

$$d(\mathbf{m}) \propto \sum_{i=1}^{4} c_i \exp\left(-\frac{1}{2}[\mathbf{m} - \mathbf{m}_i^o]^T H_i^o [\mathbf{m} - \mathbf{m}_i^o]\right),$$

where the local means are given by

$$\mathbf{m}_1^o = [-1.5, -1.5]^T, \quad \mathbf{m}_2^o = [1.5, 1.5]^T, \quad \mathbf{m}_3^o = [-2, 2]^T, \quad \mathbf{m}_4^o = [5, -5]^T,$$

the local inverse covariance matrices read

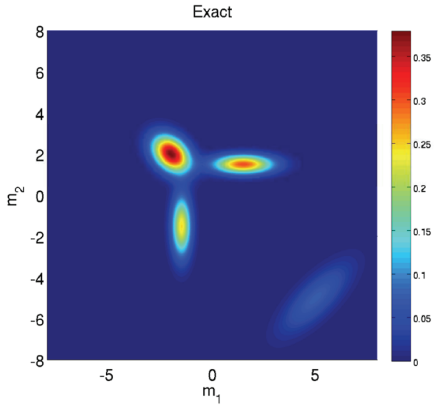
$$H_1^o = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_2^o = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad H_3^o = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}, \quad H_4^o = \begin{bmatrix} 2 & -1.5 \\ -1.5 & 2 \end{bmatrix},$$

and the coefficients c_i are randomly generated as

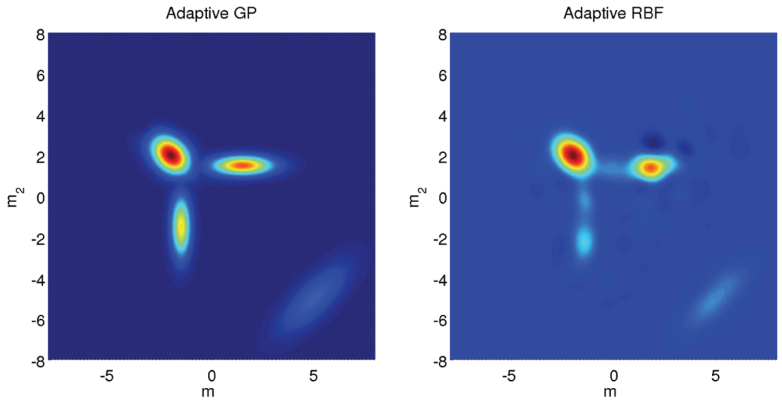
$$c_1 = 0.25, \quad c_2 = 0.3, \quad c_3 = 0.38, \quad c_4 = 0.07.$$

A GP predictor with 553 function counts after 5 greedy cycles is presented in Figure 7.3(b), whereas the true function is shown in Figure 7.3(a). An adaptive RBF with 553 LHC sampling points is shown in Figure 7.3(c). As can be seen, the GP predictor outperforms the adaptive RBF. To further confirm this, we compute different discrete error norms over 90,000 uniform grid points and present the results in Table 7.3. The errors of the GP predictor are orders of magnitude smaller than those of the adaptive RBF.

7.3. 10-dimensional examples. The first example in this subsection is the mixture of two Gaussians in 10-dimensional space:



(a) Exact.



(b) Adaptive GP predictor. (c) Adaptive RBF predictor.

FIG. 7.3. *A mixture of four anisotropic Gaussians in the two-dimensional example:* (a) *the exact function,* (b) *a GP predictor with* 553 *function counts after* 5 *greedy cycles, and* (c) *an adaptive RBF predictor with* 553 *LHC sampling points.*

TABLE 7.3
GP predictor error versus RBF predictor error over a uniform grid of 90,000 *points.*

Method	MSE	ℓ^1-norm	ℓ^2-norm	Hellinger norm
GP	3.10e-6	4.91	5.28e-1	2.19
RBF	1.36e-4	1.52e1	3.5	9.23

$$d(\mathbf{m}) = c_1 \mathcal{N}(\mathbf{m}_1^o, \boldsymbol{\Sigma}_1^o) + c_2 \mathcal{N}(\mathbf{m}_2^o, \boldsymbol{\Sigma}_2^o),$$

where

$$\begin{aligned}
&c_1 = 0.8, \quad c_2 = 0.2,\\
&m_1 = -2 \times ones(10,1), \quad m_2 = 2 \times ones(10,1),\\
&\boldsymbol{\Sigma}_1^o = \text{diag}[0.1629, 0.1812, 0.0254, 0.1827, 0.1265, 0.0195, 0.0557,\\
&\qquad 0.1094, 0.1915, 0.1930],\\
&\boldsymbol{\Sigma}_2^o = \text{diag}[0.0315, 0.1941, 0.1914, 0.0971, 0.1601, 0.0284, 0.0844,\\
&\qquad 0.1831, 0.1584, 0.1919],
\end{aligned}$$

TABLE 7.4
The sample means of GP predictor and the exact function up to one million MCMC simulations.

Mean	GP predictor	Exact $d(\mathbf{m})$
m_1	-1.9941	-1.9923
m_2	-1.9894	-1.9923
m_3	-2.0007	-1.9996
m_4	-1.9880	-1.9880
m_5	-1.9961	-1.9957
m_6	-1.9995	-1.9997
m_7	-2.0005	-2.0018
m_8	-2.0003	-2.0006
m_9	-1.9911	-1.9847
m_{10}	-1.9891	-1.9902

where $ones(10,1)$ is a 10×1 column vector with all elements equal to 1, and *diag* puts a vector on the main diagonal of the zero matrix of corresponding size. The domain of interest is the hypercube $[-3,3]^{10}$. The GP predictor requires 3 greedy cycles with 679 function evaluations to capture the two modes. In order to compare the exact function $d(\mathbf{m})$ and its adaptive GP predictor, we sample both of them using DRAM [29], an efficient MCMC toolbox. Table 7.4 shows the sample means from one million MCMC simulations. Here, it is our intent not to run enough simulations until the MCMC converges, but to show how well the response surface emulates the exact one. As can be seen, the sample means after one million MCMC simulations are the same up to three digits, though they are by no means close to the exact mean $-1.2 \times ones(10,1)$.

Now, if we take 679 LHC points for the RBF approach, all the function values evaluated at these points are machine zero, and hence the RBF method would give a zero response surface, which is by no means close to the exact function. This is, as discussed, a manifestation of the curse of dimensionality.

Similarly, we consider the mixture of two 10-dimensional Cauchy distributions:

$$d(\mathbf{m}) = c_1 \mathcal{C}(\mathbf{m}_1^o, \boldsymbol{\sigma}_1^o) + c_2 \mathcal{C}(\mathbf{m}_2^o, \boldsymbol{\sigma}_2^o),$$

where

$$\begin{aligned}
&\mathcal{C}(\mathbf{m}_j^o, \boldsymbol{\sigma}_j^o) = \prod_{i=1}^{10} \frac{\sigma_{ji}^o}{\pi[(\sigma_{ji}^o)^2 + (\mathbf{m}_i - \mathbf{m}_{ji}^o)^2]},\\
&c_1 = 0.65, \quad c_2 = 0.35,\\
&m_1 = zeros(10,1), \quad m_2 = 2 \times ones(10,1),\\
&\boldsymbol{\sigma}_1^o = [0.4074, 0.4529, 0.0635, 0.4567, 0.3162, 0.0488, 0.1392,\\
&\qquad 0.2734, 0.4788, 0.4824],\\
&\boldsymbol{\sigma}_2^o = [0.0315, 0.1941, 0.1914, 0.0971, 0.1601, 0.0284, 0.0844,\\
&\qquad 0.1831, 0.1584, 0.1919],
\end{aligned}$$

where $zeros(10,1)$ is the zero vector of dimension 10×1. Similar to the Gaussian case, the domain of interest is the hypercube $[-3,3]^{10}$. The GP predictor requires 3 greedy cycles with 1386 function evaluations to capture the two modes. Table 7.5 compares the sample means obtained from the GP predictor and the exact $d(\mathbf{m})$ using one million MCMC simulations. The result is reasonable though it is not as good as the Gaussian case. Similar to the Gaussian case, if we use the adaptive RBF method with 1386 LHC points, $d(\mathbf{m})$ is machine zero at these points, hence yielding

TABLE 7.5
The sample means of GP predictor and the exact function from one million MCMC simulations.

Mean	GP Predictor	Exact $d(\mathbf{m})$
m_1	2.0002	1.9761
m_2	2.0007	1.8883
m_3	1.9305	1.8976
m_4	1.9980	1.9321
m_5	1.9996	1.9424
m_6	1.9991	1.9962
m_7	1.9970	1.9428
m_8	1.9984	1.9161
m_9	1.9979	1.9236
m_{10}	2.0002	1.8985

a zero response surface! In fact, for both Gaussian and Cauchy cases, we have tested that $d(\mathbf{m})$ is close to machine zero even for 100,000 LHC points. Again, the curse of dimensionality is in action. This observation also suggests that all the discrete norms that we have used above for low dimension examples are useless in high dimensional problems because they are most likely zero.

It should be pointed out that the mean of Cauchy distribution is undefined [69]. As a result, computing its mean is not sensible. Nevertheless, the result in Table 7.5 is still meaningful as a demonstration of how well the GP predictor emulates the exact underlying density.

If one desires to compute an accurate estimate mean of the GP predictor and multimodal density $d(\mathbf{m})$ via the Monte Carlo method, the method of choice may be the so-called sequential Monte Carlo (SMC) method. In this paper, we use a variant of the SMC sampler proposed in [48] and later employed by [67] in the context of Bayesian inversion. In particular, we sequentially sample the following sequence of densities:

$$\pi_p(\mathbf{m}) = [\pi_{target}(\mathbf{m})]^{\gamma_p}[\pi_0(\mathbf{m})]^{1-\gamma_p}, \quad p = 0, \ldots, P,$$

where we choose $P = 10$ and $\gamma_p = p/10$. Here, $\pi_{target}(\mathbf{m})$ is either $\hat{d}_n(\mathbf{m})$ or $d(\mathbf{m})$. For simplicity, we use

$$\pi_0 = \mathcal{N}(0, 0.8^2\mathbf{I}_{10}),$$

with I_{10} denoting the 10×10 identity matrix. We move particles using the Metropolis–Hastings kernel with the following random walk proposal:

$$q(\mathbf{m}_k, \mathbf{y}) = \mathcal{N}(\mathbf{m}_k, 0.4^2\mathbf{I}_{10}).$$

Following [67], we perform the resampling step using a multinomial sampler immediately after each sequential important sampling with $\xi = 0.85$ (see [67] for more details).

For the purpose of demonstration, let us apply the SMC method to the mixture of two Gaussians discussed above. We generate $100,000$ SMC samples and compute the sample mean for both exact density $d(\mathbf{m})$ and the GP predictor; see Table 7.6. As can be observed, both sample means are much closer to the true mean, which is $-1.2 \times ones(10, 1)$. This implies the better mixing property of SMC compared to DRAM. In particular, DRAM seems to be stuck with the first mode, at least with one million samples, while SMC hops between the two modes quite well.

TABLE 7.6
The sample means of GP predictor and the exact function using $100,000$ *SMC samples.*

Mean	GP predictor	Exact $d(\mathbf{m})$
m_1	-1.3939	-1.3820
m_2	-1.0149	-1.1987
m_3	-1.3171	-1.3195
m_4	-0.9079	-1.0312
m_5	-1.5105	-1.2009
m_6	-1.2953	-1.3111
m_7	-1.2193	-1.3524
m_8	-1.5211	-1.5392
m_9	-1.7016	-1.3732
m_{10}	-1.3478	-1.3143

7.4. Inverse shape electromagnetic scattering example. In this section, we consider two-dimensional transverse magnetic (TM) polarization in the context of electromagnetic scattering due to a scatterer in the free space. The governing equations read

$$\begin{aligned}
\frac{\partial H_x}{\partial t} + \frac{\partial E_z}{\partial y} &= 0 && \text{in } \Omega \times (0,T),\\
-\frac{\partial H_y}{\partial t} + \frac{\partial E_z}{\partial x} &= 0 && \text{in } \Omega \times (0,T),\\
E_z &= E_z^I && \text{in } \partial\Omega_S \times (0,T),\\
H_x = H_y = E_z &= 0 && \text{in } \Omega \times \{0\},
\end{aligned}$$

where H_x, H_y and E_z denote the x–y components of the magnetic field and z component of the electric field with appropriate normalization [30], respectively. Here, $E^I = \cos(8(t-x))$ is the incident electric field, and Ω_S the scatterer satisfying $\Omega_s \subset \Omega \subseteq \mathbb{R}^2$.

Next, denoting

$$\mathbf{H}_\perp = \begin{bmatrix} -H_y \\ H_x \end{bmatrix}, \quad E = E_z,$$

and using the perfect matched layer (PML) proposed in [1], the above TM equations become

$$\frac{\partial \mathbf{H}_\perp}{\partial t} + \nabla E = \mathbf{L} \qquad \text{in } \Omega \times (0,T), \tag{7.1a}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \mathbf{H}_\perp = M \qquad \text{in } \Omega \times (0,T), \tag{7.1b}$$

$$\frac{\partial \mathbf{P}}{\partial t} = \mathbf{S}, \quad \frac{\partial \mathbf{Q}}{\partial t} = \mathbf{R} \qquad \text{in } \Omega \times (0,T), \tag{7.1c}$$

$$E = -E^I \qquad \text{in } \partial\Omega_S \times (0,T), \tag{7.1d}$$

$$E = 0, \quad \mathbf{H}_\perp = \mathbf{0} \qquad \text{in } \Omega \times \{0\}, \tag{7.1e}$$

$$\mathbf{P} = \mathbf{0}, \quad \mathbf{Q} = \mathbf{0} \qquad \text{in } \Omega \times \{0\}, \tag{7.1f}$$

where

$$\begin{aligned}
&\mathbf{P} = [P_x, P_y]^T, && \mathbf{L} = \mathbf{A}\mathbf{H}_\perp + \mathbf{B}\mathbf{P}, && \mathbf{S} = \mathbf{D}\mathbf{H}_\perp,\\
&\mathbf{Q} = [Q_x, Q_y]^T, && \mathbf{R} = \mathbf{G}\mathbf{Q} + \mathbf{F}\mathbf{H}_\perp, && M = \mathbf{C}^T\mathbf{Q},
\end{aligned}$$

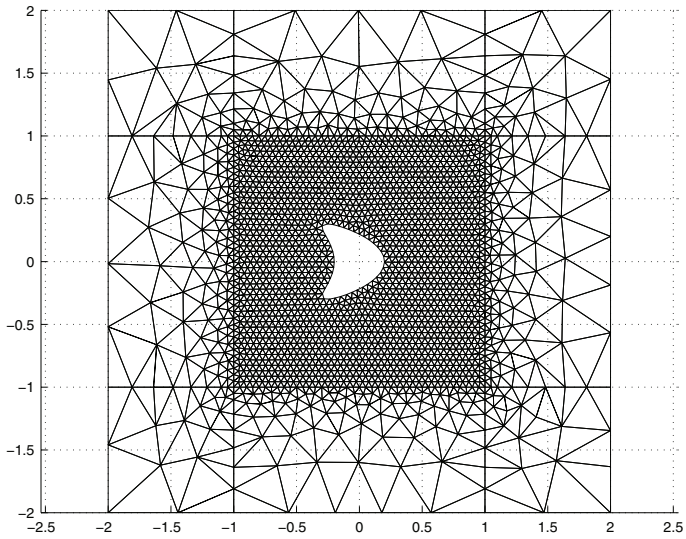


FIG. 7.4. *A typical mesh of the PML and truncated domains together with a scatterer.*

$$\mathbf{A} = \begin{bmatrix} -2\sigma_x & 0 \\ 0 & -2\sigma_y \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \sigma_x & 0 \\ 0 & -\sigma_y \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} -\frac{d\sigma_x}{dx} \\ \frac{d\sigma_y}{dy} \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} -\sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} -\sigma_x & 0 \\ 0 & -\sigma_y \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

with σ_x and σ_y, the defining damping property of the PML layer, given by

$$\sigma_x = \begin{cases} 0, & |x| < 1, \\ 15(x-1)^2, & x \geq 1, \\ 15(x+1)^2, & x \leq -1, \end{cases} \qquad \sigma_y = \begin{cases} 0, & |y| < 1, \\ 15(y-1)^2, & y \geq 1, \\ 15(y+1)^2, & y \leq -1. \end{cases}$$

A typical truncated domain $[-1,1]^2$ (fine mesh) together with the PML domain (coarse mesh) is shown in Figure 7.4. The object in the middle of the domain is a scatterer. We also show a typical scattered electric field solution in Figure 7.5.

The forward problem can be stated as follows. Given a scatterer's shape, the goal is to compute the scattered fields, especially at the observation points denoted as small circles in Figure 7.5.

In the inverse problem, on the other hand, the task is to reconstruct the scatterer's shape, Ω_S, given scattered field data, possibly polluted by noise, at K observation points. It is not necessary to have the scattered data for all the fields, but for convenience we assume that we do. We choose to solve the inverse problem statistically using a Bayesian framework whose details can be found in [16, 36]. Assuming i.i.d. Gaussian noise with zero mean and variance σ^2 at all observation points, the likelihood model is chosen as

$$\pi_{\text{like}} \propto \exp\left\{ -\frac{1}{2\sigma^2 T} \sum_{k=1}^{K} \int_T \int_\Omega [(E - E_k^{obs})^2 + (\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs})^T (\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs})] \delta_{(\mathbf{x}-\mathbf{x}_k)} d\Omega dt \right\},$$

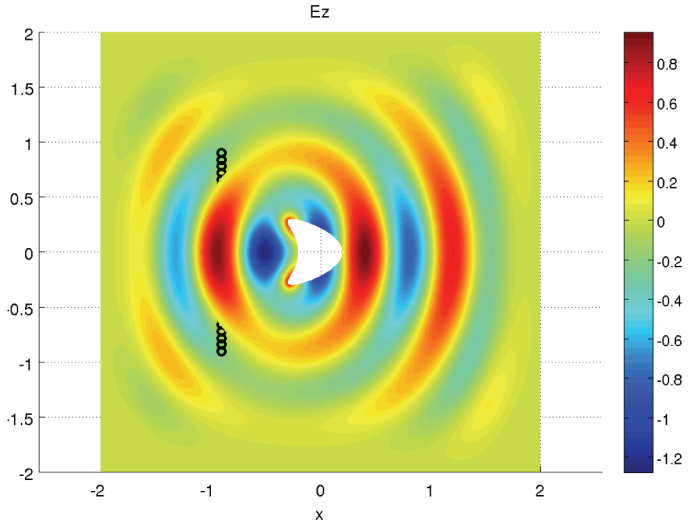


FIG. 7.5. *An example of a scattered electric field.*

where quantities with superscript "obs" are the observed data and $\mathbf{x} = [x, y]^T$. The Dirac delta function $\delta_{(\cdot)}$ is defined as

$$\delta_{(\mathbf{x}-\mathbf{x}_k)} = \begin{cases} 1, & \mathbf{x} = \mathbf{x}_k, \\ 0, & \text{otherwise.} \end{cases}$$

We begin the prior modeling by defining the admissible shape space. In this paper, the shape parametrization is restricted as

$$r = \sum_{i=1}^{n_s} m^i \cos([i-1]\theta),$$

where (r, θ) are polar coordinates of the shape and m^i is the ith shape parameter. Assume a priori that the unknown shape is smooth so that the following spline smoothing can be employed [59]:

$$\pi_{\text{prior}} \propto \exp\left\{ -\frac{\kappa}{2} \int_0^{2\pi} \left(\frac{d^2 r}{d\theta^2} \right)^2 d\theta \right\}.$$

The solution to the statistical inverse problem is the following posterior density, after ignoring the normalized constant (which is not required by MCMC methods):

$$d(\mathbf{m}) = \pi_{\text{like}} \times \pi_{\text{prior}}.$$

Denoting $\mathcal{J} = -\log d(\mathbf{m})$, we then have

$$\mathcal{J} = \frac{1}{2\sigma^2 T} \sum_{k=1}^{K} \int_T \int_\Omega [(E - E_k^{obs})^2 + (\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs})^T (\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs})] \delta_{(\mathbf{x}-\mathbf{x}_k)} d\Omega dt + \frac{\kappa}{2} \int_0^{2\pi} \left(\frac{d^2 r}{d\theta^2} \right)^2 d\theta.$$

One of the key ingredients of our approach is the Hessian (or its approximation) of $\mathcal{J}$. For large-scale PDE-based inverse problem such as the inverse scattering example considered in this section, an efficient method for computing the Hessian is vital, and we adopt an adjoint approach to fulfill this goal. To begin, we form the Lagrangian

$$\begin{aligned}\mathcal{L} = \mathcal{J} &+ \int_T \int_\Omega \mathbf{h}_\perp \cdot \left(\frac{\partial \mathbf{H}_\perp}{\partial t} + \nabla E - \mathbf{A}\mathbf{H}_\perp - \mathbf{B}\mathbf{P} \right) d\Omega dt \\ &+ \int_T \int_\Omega e \left(\frac{\partial E}{\partial t} + \nabla \cdot \mathbf{H}_\perp - \mathbf{C}^T \mathbf{Q} \right) d\Omega dt + \int_T \int_\Omega \mathbf{p} \cdot \left(\frac{\partial \mathbf{P}}{\partial t} - \mathbf{D}\mathbf{H}_\perp \right) d\Omega dt \\ &+ \int_T \int_\Omega \mathbf{q} \cdot \left(\frac{\partial \mathbf{Q}}{\partial t} - \mathbf{G}\mathbf{Q} - \mathbf{F}\mathbf{H}_\perp \right) d\Omega dt + \int_T \int_{\partial\Omega_S} \lambda (E + E^I) ds dt \\ &+ \int_\Omega \mathbf{h}_I \cdot \mathbf{H}_\perp d\Omega + \int_\Omega \mathbf{p}_I \cdot \mathbf{P} d\Omega + \int_\Omega \mathbf{q}_I \cdot \mathbf{Q} d\Omega + \int_\Omega e_I E d\Omega.\end{aligned}$$

The first order Karush–Kuhn–Tucker optimality system can be derived as follows:

- Taking the first variation of the Lagrangian with respect to $\mathbf{h}_\perp, e, \mathbf{p}, \mathbf{q}$ and arguing that the variations of $\mathbf{h}_\perp, e, \mathbf{p}, \mathbf{q}$ are arbitrary in $\Omega \times (0,T)$ yields the forward equations, (7.1a)–(7.1c).
- Taking the first variation of the Lagrangian with respect to λ and arguing that the variation of λ is arbitrary in $\partial\Omega_S \times (0,T)$ yields the forward perfect electric conduction (PEC) condition, (7.1d).
- Taking the first variation of the Lagrangian with respect to $e_I, \mathbf{h}_I, \mathbf{p}_I, \mathbf{q}_I$ and arguing that the variations of $e_I, \mathbf{h}_I, \mathbf{p}_I, \mathbf{q}_I$ are arbitrary in $\Omega \times \{0\}$ yields the forward initial conditions, (7.1e)–(7.1f).
- Taking the first variation of the Lagrangian with respect to $\mathbf{H}_\perp, E, \mathbf{P}, \mathbf{Q}$ and arguing that the variations of $\mathbf{H}_\perp, E, \mathbf{P}, \mathbf{Q}$ are arbitrary in the corresponding domains yields the following adjoint equations together with the final and boundary conditions:

$$\frac{\partial \mathbf{h}_\perp}{\partial t} + \nabla e = \mathbf{L}^* \qquad \text{in } \Omega \times (0,T), \tag{7.2a}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot \mathbf{h}_\perp = M^* \qquad \text{in } \Omega \times (0,T), \tag{7.2b}$$

$$\frac{\partial \mathbf{p}}{\partial t} = -\mathbf{B}^T \mathbf{h}_\perp, \quad \frac{\partial \mathbf{q}}{\partial t} = \mathbf{G}^T \mathbf{q} - e\mathbf{C} \qquad \text{in } \Omega \times (0,T), \tag{7.2c}$$

$$e = 0 \qquad \text{in } \partial\Omega_S \times (0,T), \tag{7.2d}$$

$$e = 0, \quad \mathbf{h}_\perp = \mathbf{0} \qquad \text{in } \Omega \times \{T\}, \tag{7.2e}$$

$$\mathbf{p} = \mathbf{0}, \quad \mathbf{q} = \mathbf{0} \qquad \text{in } \Omega \times \{T\}, \tag{7.2f}$$

where

$$\mathbf{L}^* = \sum_{k=1}^{K} (\mathbf{H}_\perp - \mathbf{H}_{\perp k}^{obs}) \delta_{(\mathbf{x}-\mathbf{x}_k)} - \mathbf{A}^T \mathbf{h}_\perp - \mathbf{D}^T \mathbf{p} - \mathbf{F}^T \mathbf{q},$$

$$M^* = \sum_{k=1}^{K} (E - E_k^{obs}) \delta_{(\mathbf{x}-\mathbf{x}_k)}.$$

Other adjoint variables are found to be

$$\lambda = -\mathbf{h}_\perp \cdot \mathbf{n} \qquad \text{in } \partial\Omega_S \times (0,T),$$

$$e_I = e, \quad \mathbf{h}_I = \mathbf{h}_\perp, \quad \mathbf{p}_I = \mathbf{p}, \quad \mathbf{q}_I = \mathbf{q} \qquad \text{in } \Omega \times \{0\}.$$

- Taking derivatives of the Lagrangian with respect to the shape parameters m^i can be done using the shape gradient and Hessian methods as in [20]. Here, we assume that the obstacle is star-like around the origin and hence that a simpler route is possible (see [12] and references therein). The derivative of the Lagrangian with respect to m^i turns out to be

$$\mathcal{G}_i = \frac{\partial \mathcal{L}}{\partial m^i} = -\int_T \int_0^{2\pi} [\mathbf{h}_\perp \cdot \nabla (E + E^I)] r \cos([i-1]\theta) d\theta dt. \tag{7.3}$$

The reduced gradient computation at a particular shape $\mathbf{m}$ is now ready. One first solves the forward system (7.1a)–(7.1f) for the forward states and forward PML variables. The adjoint system (7.2a)–(7.2f) is then solved for the adjoint states and adjoint PML variables. The reduced shape gradient is now available by evaluating the right-hand side of (7.3). It is clear that one forward and one adjoint solve are needed for the shape gradient computation.

In order to compute the product of the shape Hessian and a vector of shape variation, we first compute the forward variation, involving one incremental forward (linearization of the forward equations) solve, and then the adjoint variation, involving one incremental adjoint (linearization of the adjoint equations) solve, corresponding to that shape variation vector. The shape Hessian-vector product is the total variation of the shape gradient (7.3). More specifically, the variation δr due to variation $\delta \mathbf{m}$ is given by

$$\delta r = \sum_{i=1}^{n_s} \delta m^i \cos([i-1]\theta).$$

The ith component of the product of the shape Hessian and vector $\delta \mathbf{m}$ can be shown to be

$$\begin{aligned}
\delta G_i = &-\int_T \int_0^{2\pi} \nabla [\mathbf{h}_\perp \cdot \nabla (E + E^I)] \cdot \mathbf{e} r \cos([i-1]\theta) \delta r \, d\theta dt \\
&-\int_T \int_0^{2\pi} [\mathbf{h}_\perp \cdot \nabla (E + E^I)] \delta r \cos([i-1]\theta) d\theta dt \\
&-\int_T \int_0^{2\pi} [\delta \mathbf{h}_\perp \cdot \nabla (E + E^I)] r \cos([i-1]\theta) \, d\theta dt \\
&-\int_T \int_0^{2\pi} [\mathbf{h}_\perp \cdot \nabla \delta E] r \cos([i-1]\theta) \, d\theta dt,
\end{aligned}$$

where $\mathbf{e} = [\cos\theta, \sin\theta]^T$ and the variations in the forward states, δE and $\delta \mathbf{H}_\perp$, satisfy the following incremental forward equations:

$$\begin{aligned}
\frac{\partial \delta \mathbf{H}_\perp}{\partial t} + \nabla \delta E &= \delta \mathbf{L} && \text{in } \Omega \times (0,T), \\
\frac{\partial \delta E}{\partial t} + \nabla \cdot \delta \mathbf{H}_\perp &= \delta M && \text{in } \Omega \times (0,T), \\
\frac{\partial \delta \mathbf{P}}{\partial t} = \delta \mathbf{S}, \quad \frac{\partial \delta \mathbf{Q}}{\partial t} &= \delta \mathbf{R} && \text{in } \Omega \times (0,T), \\
\delta E &= -\nabla (E + E^I) \cdot \mathbf{e} \, \delta r && \text{in } \partial\Omega_S \times (0,T), \\
\delta E = 0, \quad \delta \mathbf{H}_\perp &= \mathbf{0} && \text{in } \Omega \times \{0\}, \\
\delta \mathbf{P} = \mathbf{0}, \quad \delta \mathbf{Q} &= \mathbf{0} && \text{in } \Omega \times \{0\},
\end{aligned}$$

where

$$\delta\mathbf{P} = [\delta P_x, \delta P_y]^T, \qquad \delta\mathbf{L} = \mathbf{A}\delta\mathbf{H}_\perp + \mathbf{B}\delta\mathbf{P}, \qquad \delta\mathbf{S} = \mathbf{D}\delta\mathbf{H}_\perp,$$
$$\delta\mathbf{Q} = [\delta Q_x, \delta Q_y]^T, \qquad \delta\mathbf{R} = \mathbf{G}\delta\mathbf{Q} + \mathbf{F}\delta\mathbf{H}_\perp, \qquad \delta M = \mathbf{C}^T\delta\mathbf{Q}.$$

Similarly, the variations in the adjoint states, δe and $\delta\mathbf{h}_\perp$, satisfy the following incremental adjoint equations:

$$\begin{aligned}
\frac{\partial\delta\mathbf{h}_\perp}{\partial t} + \nabla\delta e &= \delta\mathbf{L}^* && \text{in } \Omega\times(0,T),\\
\frac{\partial\delta e}{\partial t} + \nabla\cdot\delta\mathbf{h}_\perp &= \delta M^* && \text{in } \Omega\times(0,T),\\
\frac{\partial\delta\mathbf{p}}{\partial t} = -\mathbf{B}^T\delta\mathbf{h}_\perp, \quad \frac{\partial\delta\mathbf{q}}{\partial t} &= \mathbf{G}^T\delta\mathbf{q} - \delta e\mathbf{C} && \text{in } \Omega\times(0,T),\\
\delta e &= -\nabla e\cdot\mathbf{e}\,\delta r && \text{in } \partial\Omega_S\times(0,T),\\
e = 0, \quad \delta\mathbf{h}_\perp &= \mathbf{0} && \text{in } \Omega\times\{T\},\\
\delta\mathbf{p} = \mathbf{0}, \quad \delta\mathbf{q} &= \mathbf{0} && \text{in } \Omega\times\{T\},
\end{aligned}$$

where

$$\delta\mathbf{L}^* = \sum_{k=1}^{K}\delta\mathbf{H}_\perp\delta_{(\mathbf{x}-\mathbf{x}_k)} - \mathbf{A}^T\delta\mathbf{h}_\perp - \mathbf{D}^T\delta\mathbf{p} - \mathbf{F}^T\delta\mathbf{q},$$
$$\delta M^* = \sum_{k=1}^{K}\delta E\delta_{(\mathbf{x}-\mathbf{x}_k)}.$$

During the Newton iterations in which the shape is updated, we generate a new corresponding mesh for the (incremental) forward and (incremental) adjoint solves. To simplify the implementation and to avoid difficulties for the mesh generator, we allow the shape parameters to vary only in the hyperrectangle defined by

$$\begin{aligned}
\mathbf{m}_L &= 0.15[1, -1, -1/2, \ldots, -1/2^{(n_s-1)}]^T,\\
\mathbf{m}_U &= [0.27, 0.15, 0.15/2, \ldots, 0.15/2^{(n_s-1)}]^T.
\end{aligned} \tag{7.4}$$

As a consequence, we fix the time step and hence avoid interpolating the solutions in time during the optimization process.

We use a second order nodal discontinuous Galerkin method [30] for spatial discretization and the classical fourth order Runge–Kutta method for temporal discretization of the (incremental) forward and (incremental) adjoint equations. The mesh has 4,494 triangles, and the total number of nodal unknowns, for both $\mathbf{H}_\perp$ and E, is 80,892. The observation data E^{obs} and $\mathbf{H}_\perp^{obs}$ are synthesized by solving the forward solver $T = \pi$ using 4474 time steps. The exact shape that we would like to invert for is governed by

$$x = \cos(t) + 0.65\cos(2t) - 0.65, \quad y = 1.5\sin(t), \quad t\in[0, 2\pi].$$

This exact shape and the corresponding electric field at $T = \pi$ are shown in Figure 7.5. For convenience, we compute the observations at all time steps and add an i.i.d. zero-mean Gaussian noise with $\sigma = 0.05$. The regularization parameter κ is chosen to be $1/(\sigma^2 T)$.

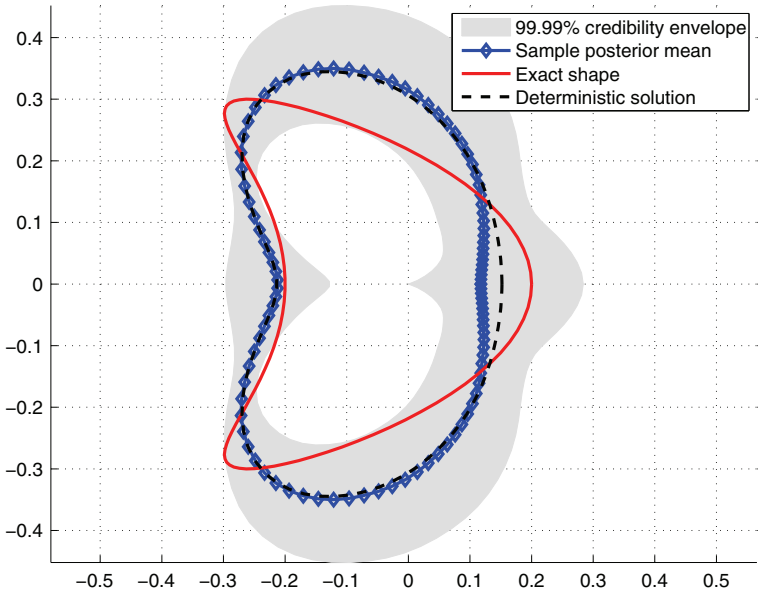


FIG. 7.6. *The sample posterior mean from one million MCMC simulations together with its* 99.99% *credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after* 1 *greedy cycle accounting for* 29 *training points and* 1 *global Gaussian approximation.*

For the first greedy cycle, we use the circle with radius of 0.236 as the initial guess for the optimization solver. For other greedy cycles, the initial guesses are computed, as in section 6, from the test set $\boldsymbol{M}^a$ containing 10000 LHC points distributed within the bound $\mathbf{m}_L \leq \mathbf{m} \leq \mathbf{m}_U$.

Next, we use DRAM, an efficient MCMC package in [29], to sample our GP predictor with one million MCMC simulations. Figure 7.6 shows the result for 29 training points and one global Gaussian approximation after 1 greedy cycle. Figure 7.7 shows the result for 62 training points and six local Gaussian approximations after 10 greedy cycles. Here, we plot the sample posterior mean and its 99.99% credibility envelope together with the exact shape and the deterministic solution. (Here, the deterministic solution is the solution of the Karush–Kuhn–Tucker system.) As can be seen, the posterior mean predicts well the left-hand side of the kite but worse on the right. This is expected since the incident wave is from left to right. The credibility region responds similarly; namely, the uncertainty is less on the left and grows gradually as we move to the right. Nevertheless, the uncertainty is large, even on the left side, at the center of the concave part of the kite. This is anticipated since nonconvex regions are not easy to reconstruct [39]. It is interesting to observe that within the bounds of interest (7.4), the approximations with 1 and 10 greedy cycles are not very different. This suggests that there is a single dominant mode of the Bayesian posterior density which is already captured by the first greedy cycle. One can also see that the sample mean is almost identical to the deterministic solution.

Note that a million MCMC simulations is chosen randomly, but it turns out that the MCMC already converges. To see this, we perform two million MCMC simulations and show again the sample posterior mean together with its 99.99% credibility envelope versus the exact and the deterministic solutions; see Figures 7.8 and 7.9. The results look almost unchanged compared to those in Figures 7.6 and 7.7.

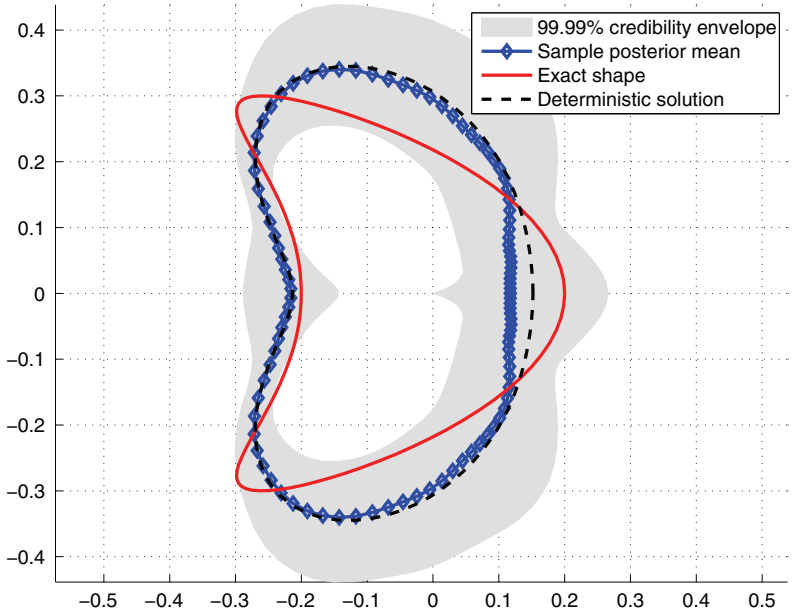


FIG. 7.7. *The sample posterior mean from one million MCMC simulations together with its* 99.99% *credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after* 10 *greedy cycles accounting for* 62 *training points and one global Gaussian approximation.*

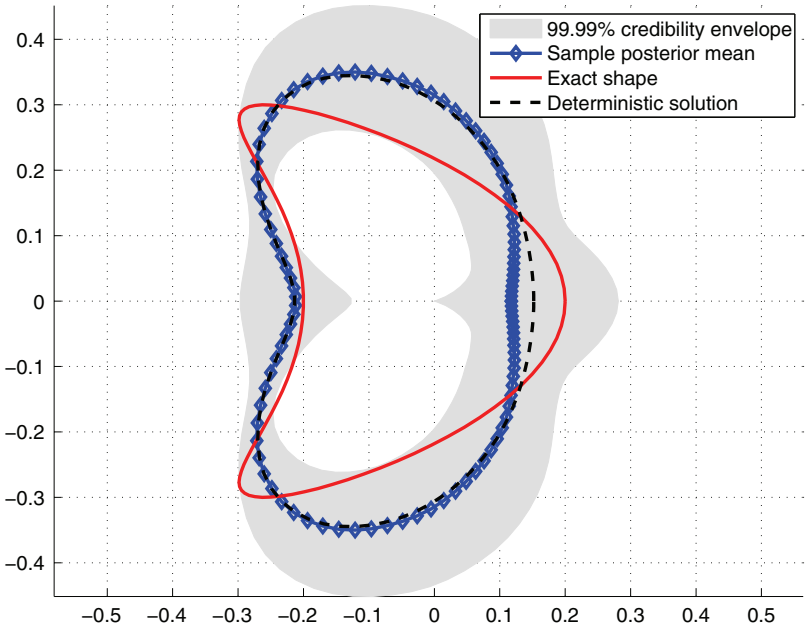


FIG. 7.8. *The sample posterior mean from two million MCMC simulations together with its* 99.99% *credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after* 1 *greedy cycle accounting for* 29 *training points and one global Gaussian approximation.*

To further confirm the convergence, we plot two-dimensional marginal chains for the GP predictor obtained after 10 greedy cycles using one million MCMC simulations in Figures 7.10 and 7.11, and those using two million MCMC simulations in

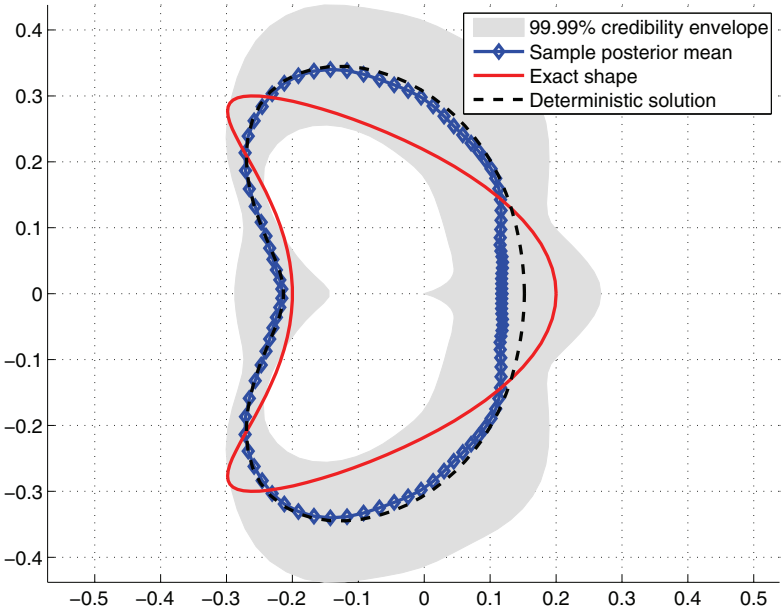


FIG. 7.9. *The sample posterior mean from two million MCMC simulations together with its* 99.99% *credibility envelope versus the exact and the deterministic solutions. The Gaussian predictor is obtained after* 10 *greedy cycles accounting for* 62 *training points and one global Gaussian approximation.*

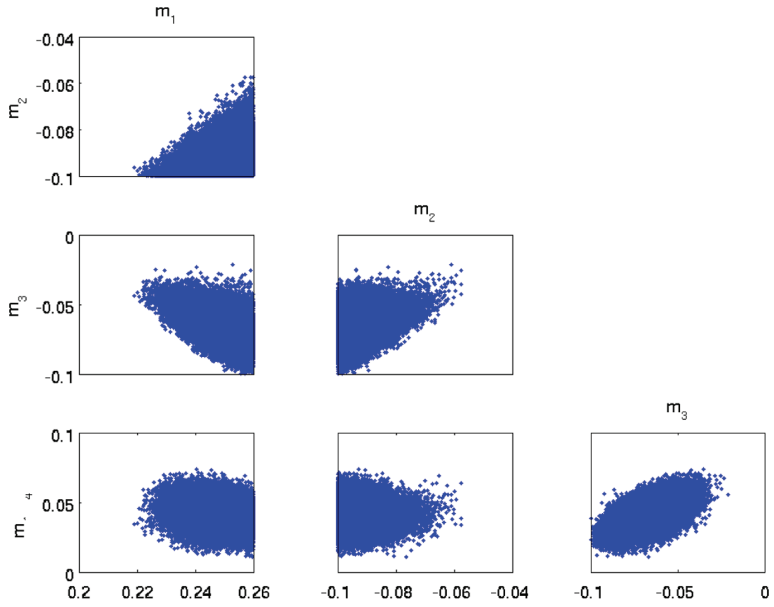


FIG. 7.10. *Two-dimensional marginal chains for parameters* m^1, m^2, m^3, m^4. *The GP predictor is obtained after* 10 *greedy cycles using one million MCMC simulations.*

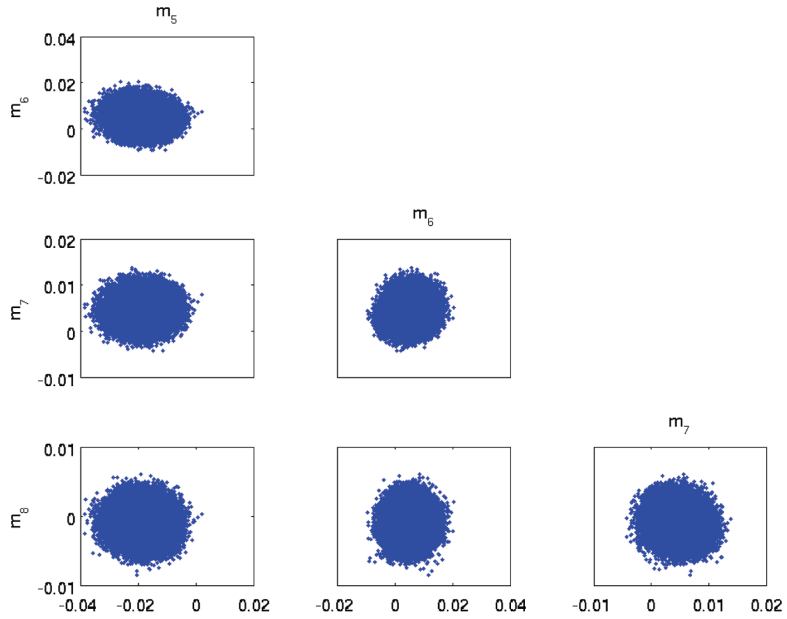


FIG. 7.11. *Two-dimensional marginal chains for parameters* m^5, m^6, m^7, m^8. *The GP predictor is obtained after* 10 *greedy cycles using one million MCMC simulations.*

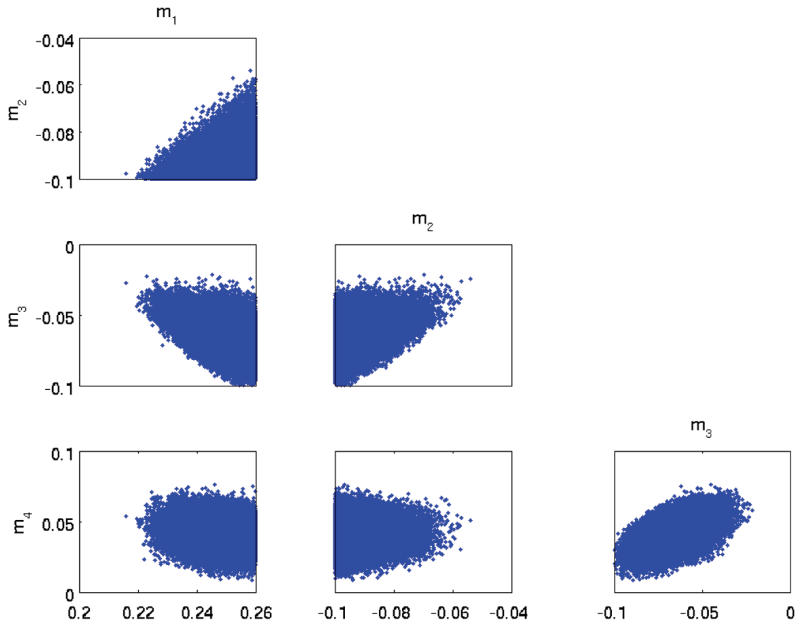


FIG. 7.12. *Two-dimensional marginal chains for parameters* m^1, m^2, m^3, m^4. *The GP predictor is obtained after* 10 *greedy cycles using two million MCMC simulations.*

Figures 7.12 and 7.13. As can be observed, the marginal chains (only the first eight parameters are shown) converge and look almost the same for both MCMC runs.

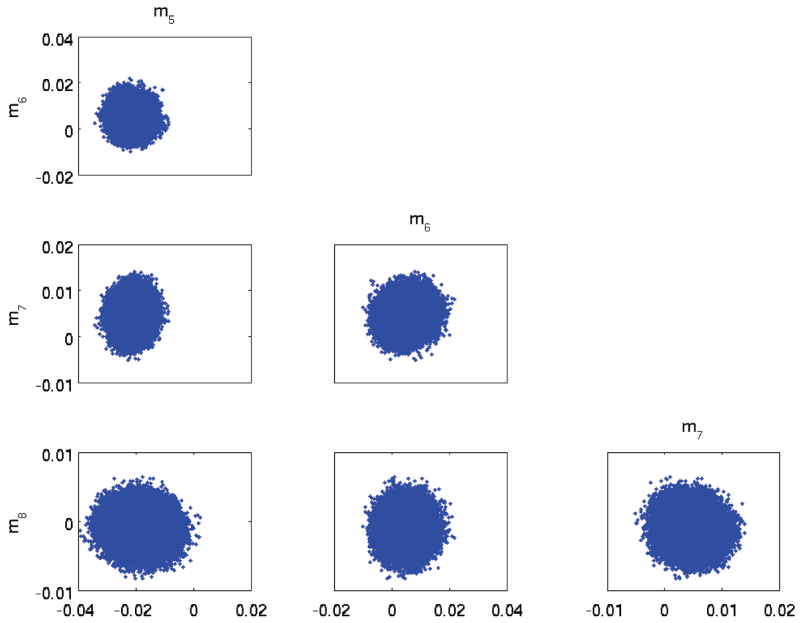


FIG. 7.13. *Two-dimensional marginal chains for parameters* m^5, m^6, m^7, m^8. *The GP predictor is obtained after* 10 *greedy cycles using two million MCMC simulations.*

TABLE 7.7
CPU time taken for two million MCMC simulations.

	GP predictor	Exact posterior density
Offline time	33 hours	0 hours
Online time	0.96 hours	528141.75 hours

Even with the above results, there is no reason to believe that the MCMC simulation has actually converged, since MCMC methods such as DRAM may get stuck in local modes for a long time. As has been observed, the first mode turns out to be the most important one. This is not surprising since we use the state-of-the-art deterministic inversion method to construct the first mode. The SMC method discussed in section 7.2 is desirable since it is able to jump to different modes frequently. However, finding an SMC method for densities with bound constraints as in this example is a challenging research topic. After all, the chief purpose of this manuscript is to introduce a new response surface technique and to demonstrate its performance on various examples. Visualization in one and two dimensions and MCMC simulation in higher dimensions are used as tools to see how our response surface emulates the exact one, in case we know the exact density, and how it reconstructs the solution of the synthetic inverse problem.

To see the efficiency of the Gaussian response surface method, we compare the CPU time taken for two million MCMC simulations for 1 greedy cycle case in Table 7.7. The offline time is defined as the time taken to build the GP response surface. It turns out that it took about 33 hours, while there would have been no offline cost if we had used the exact Bayesian posterior density. However, the offline time paid off when we performed the MCMC simulations. In particular, the MCMC simulations required almost one hour for the GP response surface, but it would have taken 528141 hours if we had used the exact posterior density!

8. Conclusions. We have developed an adaptive Hessian-based nonstationary GP response surface method for approximating a probability density function (pdf) that exploits its structure, particularly the Hessian of its negative logarithm. Of particular interest to us are expensive-to-evaluate pdfs, e.g., those arising from the Bayesian solution of large-scale inverse problems. Our method can be considered as a piecewise adaptive Gaussian approximation in which a Gaussian tailored to the local Hessian of the negative log probability density is constructed for each subregion in high dimensional parameter space. The task of efficiently partitioning the parameter space into subregions is done implicitly through Hessian-informed membership probability functions. The GP machinery is then employed to glue all local Gaussian approximations into a global analytical response surface that is far cheaper to evaluate than the original expensive probability density. The resulting response surface is also equipped with an analytical variance estimate that can be used to assess the uncertainty of the approximation. One of the key components of our proposed approach is an adaptive sampling strategy for exploring the parameter space efficiently during the computer experimental design step, which aims to find training points with high probability density. The detailed construction and an analysis of the method have been presented. We have demonstrated the accuracy and efficiency of the proposed method on several example problems, including inverse shape electromagnetic scattering in 24-dimensional parameter space.

Ongoing research aims to address the following:

1. The GP predictor is not guaranteed to be nonnegative everywhere (even though it seems to be the case for all examples considered in this paper). How to enforce the positiveness of the predictor in our framework remains an open question, though one may use an approach proposed in [54].
2. Rigorous analysis of the quality of the GP predictor is clearly an important direction for our future work [14].
3. The size of the random set $\mathbf{M}^a$ is quite arbitrary. Intuitively, the larger it is, the better our predictor. Thus, the question that needs to be addressed is how to choose $\mathbf{M}^a$ as a function of the dimension of the parameter space.
4. Constructing the full Hessian for local Gaussian approximation is prohibitively expensive for practical inverse problems in high dimensional parameter spaces. However, often one can make accurate low rank approximations of the Hessian for ill-posed inverse problems; scalable algorithms can be constructed for this task [23], and theoretical justification of the compactness of the Hessian can be provided in certain cases [12, 13].
5. We have concentrated on the detailed development and analysis of the proposed method and on its verification for several examples. We have compared our approach only to the popular adaptive RBF method. Our future work should carry out extensive comparisons with other existing methods as well.
6. The adaptive Hessian-based nonstationary GP response surface method is designed for problems involving thousands of parameters or more. Ongoing research will apply the method to such highdimensional problems as well.

Acknowledgments. We thank Lucas Wilcox for many fruitful discussions, and Zhengdong Lu a for a valuable suggestion on the soft-max function in [9]. We also thank James Martin for many useful discussions on MCMC methods and their convergence. Finally, we thank the anonymous referees for their critical and constructive comments that improved the paper substantially.

REFERENCES

[1] S. Abarbanel and D. Gottlieb, *On the construction and analysis of absorbing layers in CEM*, Appl. Numer. Math., 27 (1998), pp. 331–340.

[2] P. Abrahamsen, *A Review of Gaussian Random Fields and Correlation Functions*, Tech. Report, 2nd ed., Norwegian Computing Center, Oslo, Norway, 1997.

[3] V. Akçelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. J. Kim, J. Lopez, D. R. O'Hallaron, T. Tu, and J. Urbanic, *High resolution forward and inverse earthquake modeling on terascale computers*, in SC03: Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, ACM/IEEE, New York, 2003, p. 52.

[4] V. Akçelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloeman Waanders, *Parallel - constrained optimization*, in Parallel Processing for Scientific Computing, M. A. Heroux, P. Raghaven, and H. D. Simon, eds., Software Environ. Tools 20, SIAM, Philadelphia, 2006, pp. 291–322.

[5] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.

[6] G. Biros and O. Ghattas, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. Part* I: *The Krylov–Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[7] G. Biros and O. Ghattas, *Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. Part* II: *The Lagrange–Newton solver and its application to optimal control of steady viscous flows*, SIAM J. Sci. Comput., 27 (2005), pp. 714–739.

[8] S. Boyaval, C. Le Bris, T. Lelievre, Y. Maday, N. C. Nguyen, and A. T. Patera, *Reduced basis techniques for stochastic problems*, Arch. Comput. Methods Engrg., submitted.

[9] J. S. Bridle, *Probabilistic interpretation of feedforward classification network outputs, with relationship to statistical pattern recognition*, in Neuro-computing: Algorithms, Architectures and Applications, S.F. Fogelman and J. Hault, eds., Springer, 1989, pp. 227–236.

[10] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monogr. Appl. Comput. Math., 12 Cambridge University Press, Cambridge, UK, 2003.

[11] T. Bui-Thanh, *Model-Constrained Optimization Methods for Reduction of Parameterized Large-Scale Systems*, Ph.D. thesis, Department of Aeronautics and Astronautics, MIT, 2007.

[12] T. Bui-Thanh and O. Ghattas, *Analysis of the Hessian for inverse scattering problems. Part* I: *Inverse shape scattering of acoustic waves*, Inverse Problems, 28 (2012), 055001.

[13] T. Bui-Thanh and O. Ghattas, *Analysis of the Hessian for inverse scattering problems. Part* II: *Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), 055002.

[14] T. Bui-Thanh and O. Ghattas, *Non-stationary Gaussian Process Response for Posterior Interpolation: A Comparative Study*, manuscript, (2012).

[15] T. Bui-Thanh, K. Willcox, and O. Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput., 30 (2008), pp. 3270–3288.

[16] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, Springer, New York, 2007.

[17] S.-H. Cha, *Comprehensive survey on distance/similarity measures between probability density functions*, Int. J. Math. Models Methods Appl. Sci., 1 (2007), pp. 300–307.

[18] A. Christen and B. Sanso, *Advances in the Design of Gaussian Processes as Surrogate Models for Computer Experiments*, Tech. Report 5, University of California, Santa Cruz, CA, 2008.

[19] D. A. Cohn, *Neural networks exploration using optimal experiment design*, Neural Networks, 9 (1996), pp. 1071–1083.

[20] M. C. Delfour and J.-P. Zolésio, *Shapes and Geometries. Metrics Analysis, Differential Calculus, and Optimization*, Adv. Des. Control 22, SIAM, Philadelphia, 2011.

[21] D. L. Donoho, *High-Dimensional Data Analysis: The curses and blessings of dimensionality.* Lecture at American Math. Society "Math Challenges of the 21st Century", Los Angeles, August 6–12, 2000.

[22] I. Epanomeritakis, V. Akçelik, O. Ghattas, and J. Bielak, *A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion*, Inverse Problems, 24 (2008), 034015.

[23] H. P. Flath, L. C. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, *Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations*, SIAM J. Sci. Comput., 33 (2011), pp. 407–432.

[24] M. Fuentes and R. L. Smith, *A New Class of Nonstationary Spatial Models*, tech. report, North Carolina State University, Raleigh, NC, 2001.

[25] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas, *Nonlinear model reduction for uncertainty quantification in large-scale inverse problems*, Int. J. Numer. Methods Eng., 81 (2010), pp. 1581–1608.
[26] M. N. Gibbs, *Bayesian Gaussian Processes for regression and classification*, Ph.D. thesis, Cavendish Laboratory, Cambridge University, 1997.
[27] R. R. Gramacy, H. K. H. Lee, and W. MacReady, *Parameter space exploration with Gaussian process trees*, in Proceedings of the 21st International Conference on Machine Learning, 2004, p. 45.
[28] M. A. Grepl and A. T. Patera, *A posteriori error bounds for reduced-bias approximations of parametrized parabolic partial differential equations*, M2AN Math. Model. Numer. Anal., 39 (2005), pp. 157–181.
[29] H. Haario, M. Laine, A. Miravete, and E. Saksman, *DRAM: Efficient adaptive MCMC*, Statist. Comput., 16 (2006), pp. 339–354.
[30] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Texts Appl. Math. 54, Springer, New York, 2008.
[31] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, *Computer model calibration using high-dimensional output*, J. Amer. Statist. Assoc., 103 (2008), pp. 570–583.
[32] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, *Combining field data and computer simulations for calibration and prediction*, SIAM J. Sci. Comput., 26 (2004), pp. 448–466.
[33] D. Higdon, J. Swall, and J. Kern, *Nonstationary spatial modeling*, in Bayesian Statistics 6, J.M. Bernardo, J.O. Berger, A.P. Dawid, and F.M. Smith, eds., Oxford University Press, London, 1999, pp. 761–768.
[34] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Springer, New York, 2009.
[35] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, *Certified reduced basis model characterization: A frequentistic uncertainty framework*, Comput. Methods Appl. Mech. Engrg., submitted.
[36] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Appl. Math. Sci. 160, Springer-Verlag, New York, 2005.
[37] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, J. Royal Statist. Soc. Ser. B, 63 (2001), pp. 425–464.
[38] H.-M. Kim, B. K. Mallick, and C. C. Holmes, *Analyzing nonstationary spatial data using piecewise Gaussian processes*, J. Amer. Statist. Assoc., 100 (2005), pp. 653–668.
[39] A. Kirsch, R. Kress, P. Monk, and A. Zinn, *Two methods for solving the inverse acoustic scattering problem*, Inverse Problems, 4 (1988), pp. 749–770.
[40] A. Krause, A. Singh, and C. Guestrin, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, J. Machine Learning Res., 9 (2008), pp. 235–284.
[41] C. Lieberman, K. Willcox, and O. Ghattas, *Parameter and state model reduction for large-scale statistical inverse problems*, SIAM J. Sci. Comput., 32 (2010), pp. 2523–2542.
[42] D. J. C. MacKay, *Information-based objective functions for active data selection*, Neural Comput., 4 (1992), pp. 590–604.
[43] D. J. C. MacKay, *Bayesian methods for backpropagation networks*, in Models of Neural Networks III, E. Domany, J. L. van Hemmen, and K. Schulten, eds., Springer-Verlag, 1994, ch. 6, pp. 211–221.
[44] D. J. C. MacKay, *Choice of basis for Laplace approximation*, Machine Learning, 33 (1998), pp. 77–86.
[45] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput., 34 (2012), pp. A1460–A1487.
[46] Y. M. Marzouk and H. N. Najm, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, J. Comput. Phys., 228 (2009), pp. 1862–1902.
[47] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, J. Comput. Phys., 224 (2007), pp. 560–586.
[48] P. D. Moral, A. Doucet, and A. Jasra, *Sequential Monte Carlo samplers*, J. Royal Statist. Soc. Ser. B, 68 (2006), pp. 411–436.
[49] A. A. Mullur and A. Messac, *Extended radial basis functions: More flexible and effective metamodeling*, AIAA J., 43 (2005), pp. 1306–1315.
[50] R. M. Neal, *Bayesian Learning for Neural Networks*, Springer-Verlag, Berlin, 1996.
[51] N. C. Nguyen, *An uncertainty quantification method for parameter estimation in elliptic partial differential equations*, Comput. Methods Appl. Mech. Engrg., submitted.

[52] D. J. Nott and W. T. M. Dunsmuir, *Estimation of nonstationary spatial covariance structure*, Biometrika, 89 (2002), pp. 819–829.
[53] D. Nychka, C. Wikle, and J. A. Royle, *Large Spatial Prediction Problems and Nonstationary Random field*, tech. report, Geophysical Statistical Program, National Center for Atmospheric research, Boulder, CO, 1999.
[54] J. E. Oakley and A. O'hagan, *Uncertainty in prior elicitations: A non-parametric approach*, Biometrika, 94 (2007), pp. 427–441.
[55] A. O'Hagan, *Bayesian analysis of computer code outputs: A tutorial*, Reliability Engineering and System Safety, 91 (2006), pp. 1290–1300.
[56] C. J. Paciorek and M. J. Schervish, *Spatial modelling using a new class of nonstationary covariance functions*, EnvironMetrics, 17 (2006), pp. 483–506.
[57] T. Pfingsten, M. Kuss, and C. E. Rasmussen, *Nonstationary Gaussian process regression using a latent extension of the input space*, in Proceedings of the ISBA Eighth World Meeting on Bayesian Statistics in Valencia, 2006.
[58] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
[59] C. H. Reinsch, *Smoothing by spline functions*, Numer. Math., 16 (1971), pp. 451–454.
[60] P. D. Sampson and P. Guttorp, *Nonparametric estimation of nonstationary spatial covariance structure*, J. Amer. Statist. Assoc., 87 (1992), pp. 108–119.
[61] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*, Springer-Verlag, Berlin, 2003.
[62] A. M. Schmidt and A. O'Hagan, *Bayesian inference for nonstationary spatial covariance structure via spatial derformation*, J. Royal Statist. Soc., Ser. B, 65 (2003), pp. 745–758.
[63] D. W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*, Wiley Ser. Probab. Stat., Wiley, Chichester, UK, 1992.
[64] D. W. Scott and S. R. Sain, *Multi-dimensional density estimation*, Handbook Statist., 24 (2005), pp. 229–261.
[65] S. Seo, M. Wallat, T. Graepel, and K. Obermayer, *Gaussian process regression: Active data selection and test point rejection*, in Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000), IEEE, Piscataway, NJ, 2000, pp. 241–246.
[66] T. W. Simpson, J. D. Peplinski, P. N. Koch, and J K. Allen, *Metamodels for computer-based engineering design: Survey and recommendations*, Eng. Comput., 17 (2001), pp. 129–150.
[67] J. Wan and N. Zabaras, *A Bayesian approach to multiscale inverse problems using the sequential Monte Carlo method*, Inverse Problems, 27 (2011), 105004.
[68] J. Wang and N. Zabaras, *Using Bayesian statistics in the estimation of heat source in radiation*, Int. J. Heat Mass Transfer, 48 (2005), pp. 15–29.
[69] Wikipedia contributors, *Cauchy distribution*, in Wikipedia, The Free Encyclopedia, 2012; http://en.wikipedia.org/wiki/Cauchy_distribution.