Journal of Computational Physics 424 (2021) 109716

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Calibrate, emulate, sample

Emmet Cleary [a], Alfredo Garbuno-Inigo [a,*], Shiwei Lan [b], Tapio Schneider [a], Andrew M. Stuart [a]

[a] California Institute of Technology, Pasadena, CA, United States of America
[b] Arizona State University, Tempe, AZ, United States of America

ARTICLE INFO

Article history:
Received 11 January 2020
Received in revised form 1 July 2020
Accepted 8 July 2020
Available online 13 July 2020

Keywords:
Approximate Bayesian inversion
Uncertainty quantification
Ensemble Kalman sampling
Gaussian process emulation
Experimental design

ABSTRACT

Many parameter estimation problems arising in applications can be cast in the framework of Bayesian inversion. This allows not only for an estimate of the parameters, but also for the quantification of uncertainties in the estimates. Often in such problems the parameter-to-data map is very expensive to evaluate, and computing derivatives of the map, or derivative-adjoints, may not be feasible. Additionally, in many applications only noisy evaluations of the map may be available. We propose an approach to Bayesian inversion in such settings that builds on the derivative-free optimization capabilities of ensemble Kalman inversion methods. The overarching approach is to first use ensemble Kalman sampling (EKS) to *calibrate* the unknown parameters to fit the data; second, to use the output of the EKS to *emulate* the parameter-to-data map; third, to *sample* from an approximate Bayesian posterior distribution in which the parameter-to-data map is replaced by its emulator. This results in a principled approach to approximate Bayesian inference that requires only a small number of evaluations of the (possibly noisy approximation of the) parameter-to-data map. It does not require derivatives of this map, but instead leverages the documented power of ensemble Kalman methods. Furthermore, the EKS has the desirable property that it evolves the parameter ensemble towards the regions in which the bulk of the parameter posterior mass is located, thereby locating them well for the emulation phase of the methodology. In essence, the EKS methodology provides a cheap solution to the design problem of where to place points in parameter space to efficiently train an emulator of the parameter-to-data map for the purposes of Bayesian inversion.

1. Introduction

Ensemble Kalman methods have proven to be highly successful for state estimation in noisily observed dynamical systems [1,7,15,21,34,40,47,53]. They are widely used, especially within the geophysical sciences and numerical weather prediction, because the methodology is derivative-free, provides reliable state estimation with a small number of ensemble members, and, through the ensemble, provides information about sensitivities. The empirical success in state estimation has led to further development of ensemble Kalman methods in the solution of inverse problems, where the objective is the estimation of parameters rather than states. Its use as an iterative method for parameter estimation originates in the

[*] Corresponding author.
E-mail addresses: ecleary@caltech.edu (E. Cleary), agarbuno@caltech.edu (A. Garbuno-Inigo), slan@asu.edu (S. Lan), tapio@caltech.edu (T. Schneider), astuart@caltech.edu (A.M. Stuart).

https://doi.org/10.1016/j.jcp.2020.109716
0021-9991/© 2020 Published by Elsevier Inc.

papers [12,19] and recent contributions are discussed in [2,22,35,70]. But despite their widespread use for both state and parameter estimation, ensemble Kalman methods do not provide a basis for systematic uncertainty quantification, except in the Gaussian case [20,48]. This is for two primary reasons: (i) the methods invoke a Gaussian ansatz, which is not always justified; (ii) they are often employed in situations where evaluation of the underlying dynamical system (state estimation) or forward model (parameter estimation) is very expensive, and only a small ensemble is feasible. The goal of this paper is to develop a method that provides a basis for systematic Bayesian uncertainty quantification within inverse problems, building on the proven power of ensemble Kalman methods. The basic idea is simple: we *calibrate* the model using variants of ensemble Kalman inversion; we use the evaluations of the forward model made during the calibration to train an *emulator*; we perform approximate Bayesian inversion using Markov Chain Monte Carlo (MCMC) *samples* based on the (cheap) emulator rather than the original (expensive) forward model. Within this overall strategy, the ensemble Kalman methods may be viewed as providing a cheap and effective way of determining an experimental design for training an emulator of the parameter-to-data map to be employed within MCMC-based Bayesian parameter estimation; this is the primary innovation contained within the paper.

1.1. *Literature review*

The ensemble Kalman approach to calibrating unknown parameters to data is reviewed in [35,61], and the imposition of constraints within the methodology is overviewed in [2]. We refer to this class of methods for calibration, collectively, as ensemble Kalman inversion (EKI) methods and note that pseudo-code for a variety of the methods may be found in [2]. An approach to using ensemble-based methods to produce approximate samples from the Bayesian posterior distribution on the unknown parameters is described in [26]; we refer to this method as ensemble Kalman sampling (EKS). Either of these approaches, EKI or EKS, may be used in the calibration step of our approximate Bayesian inversion method.

Gaussian processes (GPs) have been widely used as emulation tools for computationally expensive computer codes [69]. The first use of GPs in the context of uncertainty quantification was proposed in modeling ore reserves in mining [45]. Its motivation was a method to find the best linear unbiased predictor, known as kriging in the geostatistics community [16,74]. It was later adopted in the field of computer experiments [68] to model possibly correlated residuals. The idea was then incorporated within a Bayesian modeling perspective [17] and has been gradually refined over the years. Kennedy and O'Hagan [42] offer a mature perspective on the use of GPs as emulators, adopting a clarifying Bayesian formulation. The use of GP emulators covers a wide range of applications such as uncertainty analysis [59], sensitivity analysis [60], and computer code calibration [32]. Perturbation results for the posterior distribution, when the forward model is approximated by a GP, may be found in [75]. We will exploit GPs for the *emulation* step of our method which, when informed by the calibration step, provides a robust approximate forward model. Neural networks [30] could also be used in the emulation step and may be preferable for some applications of the proposed methodology.

Bayesian inference is now widespread in many areas of science and engineering, in part because of the development of generally applicable and easily implementable sampling methods for complex modeling scenarios. MCMC methods [31,54,55] and sequential Monte Carlo (SMC) [18] provide the primary examples of such methods, and their practical success underpins the widespread interest in the Bayesian solution of inverse problems [39]. We will employ MCMC for the *sampling* step of our method. SMC could equally well be used for the sampling step and will be preferable for many problems; however, it is a less mature method and typically requires more problem-specific tuning than MCMC.

The impetus for the development of our approximate Bayesian inversion method is the desire to perform Bayesian inversion on computer models that are very expensive to evaluate, for which derivatives and adjoint calculations are not readily available and are possibly noisy. Ensemble Kalman inversion methods provide good parameter estimates even with many parameters, typically with $\mathcal{O}(10^2)$ forward model evaluations [40,61], but without systematic uncertainty quantification. While MCMC and SMC methods provide systematic uncertainty quantification the fact that they require many iterations, and hence evaluations of the forward model in our setting, is well-documented [29]. Several diagnostics for MCMC convergence are available [63], and theoretical guarantees of convergence exist [56]. The rate of convergence for MCMC is determined by the size of the step arising from proposal distribution: short steps are computationally inefficient as the parameter space is only locally explored whereas large steps lead to frequent rejections and hence to a waste of computational resources (in our setting forward model evaluations) to generate additional samples. In practice MCMC often requires $\mathcal{O}(10^5)$ or more forward model evaluations [29]. This is not feasible, for example, with climate models [13,37,73].

In the sampling step of our approximate Bayesian inversion method, we use an emulator that can be evaluated rapidly in place of the computationally expensive forward model, leading to Bayesian parameter estimation and uncertainty quantification essentially at the computational cost of ensemble Kalman inversion. The ensemble methods provide an effective design for the emulation, which makes this cheap cost feasible.

In some applications, the dimension of the unknown parameter is high and it is therefore of interest to understand how the constituent parts of the proposed methodology behave in this setting. The growing use of ensemble methods reflects, in part, the empirical fact that they scale well to high-dimensional state and parameter spaces, as demonstrated by applications in the geophysical sciences [40,61]; thus, the calibrate phase of the methodology scales well with respect to high input dimension. Gaussian process regression does not, in general, scale well to high-dimensional input variables, but alternative emulators, such as those based on neural networks [30] are empirically found to do so; thus, the emulate phase can potentially be developed to scale well with respect to high input dimensions. Standard MCMC methods do not scale well with respect to high dimensions; see [65] in the context of i.i.d. random variables in high dimensions. However the reviews [11,15] describe non-traditional MCMC methods which overcome these poor scaling results for high-dimensional Bayesian inversion with Gaussian priors, or transformations of Gaussians; the paper [41] builds on the ideas in [15] to develop SMC methods that are efficient in high- and infinite-dimensional spaces. Thus, the sampling phase of the methodology scales well with respect to high input dimension for appropriately chosen priors.

There are alternative strategies related to, but different from, the methodology we present in this work. The bulk of these alternative strategies rely on the adaptive construction of the emulator as the MCMC evolves – learning on the fly [46]. Earlier works have identified the effectiveness of posterior-localized surrogate models, rather than prior-based surrogates in the setting of Bayesian inverse problems; see [14,50] and the references therein. Within the adaptive surrogate framework, other type of surrogates have been used such as polynomial chaos expansions (PCE) [76] and deep neural networks [77]; the use of adaptive PCEs has recently been merged with ensemble based algorithms [78]. The effectiveness of all these works relies on the compromise between exploring the parameter space with the current surrogate and refining it further where needed; however, as exposed in [14,50,79], the use of an inadequate surrogate leads to biased posterior estimates. Another strategy for approximate Bayesian inversion can be found in [23] where adaptivity is incorporated within a sparse quadrature rule used to approximate the integrals needed in the Bayesian inference.

1.2. Our contribution

- We introduce a practical methodology for approximate Bayesian parameter learning in settings where the parameter-to-data map is expensive to evaluate, not easy to differentiate or not differentiable, and where evaluations are possibly polluted by noise. The methodology is modular and broken into three steps, each of which can be tackled by different methodologies: *calibration, emulation, and sampling.*
- In the *calibration* phase we leverage the power of ensemble Kalman methods, which may be viewed as fast derivative-free optimizers or approximate samplers. These methods provide a cheap solution to the experimental design problem and ensure that the forward map evaluations are well-adapted to the task of Gaussian process *emulation*, within the context of an outer Bayesian inversion loop via MCMC *sampling*.
- We also show that, for problems in which the forward model evaluation is inherently noisy, the Gaussian process emulation serves to remove the noise, resulting in a more practical Bayesian inference via MCMC.
- We demonstrate the methodology with numerical experiments on a model linear problem, on a Darcy flow inverse problem, and on the Lorenz '63 and '96 models.

It is worth emphasizing that the idea of emulation within an MCMC loop is not a new one (see citations above) and that doing so incurs an error which can be controlled in terms of the number of samples used to train the emulator [75]. Similarly the use of EKS to solve inverse problems is not a new idea see [see 2,35, and the references therein]; however, except in the linear case [72], the error cannot be controlled in terms of the number of ensemble members. What is novel in our work is the *conjunction* of the two approaches lifts the EKS from being an, in general uncontrolled but reasonable, approximator of the posterior into a method which provides *controllable* approximation of the posterior, in terms of the number of ensemble members used in the GP training; furthermore the EKS may be viewed intuitively in general, and provably in the linear case, as providing a good solution to the design problem related to the construction of the emulator. This is because it tends to concentrate points in the support of the true posterior, even if the points are not distributed according to the posterior [26].

In Section 2, we describe the calibrate-emulate-sample methodology introduced in this paper, and in Section 3, we demonstrate the method on a linear inverse problem whose Bayesian posterior is explicitly known. In Section 4, we study the inverse problem of determining permeability from pressure in Darcy flow, a nonlinear inverse problem in which the coefficients of a linear elliptic partial differential equation (PDE) are to be determined from linear functionals of its solution. Section 5 is devoted to the inverse problem of determining parameters appearing in time-dependent differential equations from time-averaged functionals of the solution. We view finite time-averaged data as noisy infinite time-averaged data and use GP emulation to estimate the parameter-to-data map and the noise induced through finite-time averaging. Applications to the Lorenz'63 and'96 atmospheric science models are described here, and use of the methodology to study an atmospheric general circulation model is described in the paper [13].

2. Calibrate-emulate-sample

2.1. Overview

Consider parameters θ related to data y through the forward model $\mathcal{G}$ and noise η:

$$y = \mathcal{G}(\theta) + \eta. \tag{2.1}$$

The inverse problem is to find unknown θ from y, given knowledge of $\mathcal{G} : \mathbb{R}^p \to \mathbb{R}^d$ and some information about the noise level such as its size (classical approach) or distribution (statistical approach), but not its value. To formulate the Bayesian

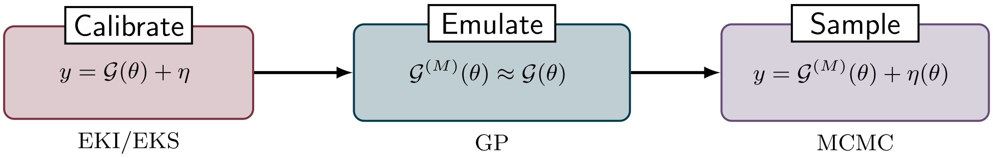

Calibrate
$y = \mathcal{G}(\theta) + \eta$
EKI/EKS
Emulate
$\mathcal{G}^{(M)}(\theta) \approx \mathcal{G}(\theta)$
GP
Sample
$y = \mathcal{G}^{(M)}(\theta) + \eta(\theta)$
MCMC

Fig. 1. Schematic of approximate Bayesian inversion method to find θ from y. EKI/EKS produce a small number of approximate (expensive) samples $\{\theta^{(m)}\}_{m=1}^{M}$. These are used to train a GP approximation $\mathcal{G}^{(M)}$ of $\mathcal{G}$, used within MCMC to produce a large number of approximate (cheap) samples $\{\theta^{(n)}\}_{n=1}^{N_s}$, $N_s \gg M$.

inverse problem, we assume, for simplicity, that the noise is drawn from a Gaussian with distribution $N(0, \Gamma_y)$, that the prior on θ is a Gaussian $N(0, \Gamma_\theta)$, and that θ and η are *a priori* independent. If we define[1]

$$\Phi_R(\theta) = \frac{1}{2}\|y - \mathcal{G}(\theta)\|_{\Gamma_y}^2 + \frac{1}{2}\|\theta\|_{\Gamma_\theta}^2, \tag{2.2}$$

the posterior on θ given y has density

$$\pi^y(\theta) \propto \exp\big(-\Phi_R(\theta)\big). \tag{2.3}$$

In a class of applications of particular interest to us, the data y comprises statistical averages of observables. The map $\mathcal{G}(\theta)$ provides the corresponding statistics delivered by a model that depends on θ. In this setting the assumption that the noise be Gaussian is reasonable if y and $\mathcal{G}(\theta)$ represent statistical aggregates of quantities that vary in space and/or time. Additionally, we take the view that parameters θ' for which Gaussian priors are not appropriate (for example because they are constrained to be positive) can be transformed to parameters θ for which Gaussian priors make sense.

We use EKS with J ensemble members and N iterations (time-steps of a discretized continuous time algorithm) to generate approximate samples from (2.3), in the *calibration* step of the methodology. This gives us JN parameter–model evaluation pairs $\{\theta^{(m)}, \mathcal{G}(\theta^{(m)})\}_{m=1}^{JN}$ which we can use to produce an approximation of $\mathcal{G}$ in the GP *emulation* step of the algorithm. Whilst the methodology of optimal experimental design can be used, in principle, as the basis for choosing parameter-data pairs for the purpose of emulation [3] it can be prohibitively expensive. The theory and numerical results shown in [26] demonstrate that EKS distributes particles in regions of high posterior probability; this is because it approximates a mean-field interacting particle system with invariant measure equal to the posterior probability distribution (2.3). Using the EKS thus provides a cheap and effective solution to the design problem, producing parameter–model evaluation pairs that are well-positioned for the task of approximate Bayesian inversion based on the (cheap) emulator. In practice it is not always necessary to use all JN parameter–model evaluation pairs but to instead use a subset of size $M \leq JN$; we denote the resulting GP approximation of $\mathcal{G}$ by $\mathcal{G}^{(M)}$. Throughout this paper we simply take $M = J$ and use the output of the EKS in the last step of the iteration as the design. However other strategies, such as decreasing J and using all or most of the N steps, are also feasible.

$\mathcal{G}^{(M)}$ in place of the (expensive) forward model $\mathcal{G}$. With the emulator $\mathcal{G}^{(M)}$, we define the modified inverse problem of finding parameter θ from data y when they are assumed to be related, through noise η, by

$$y = \mathcal{G}^{(M)}(\theta) + \eta.$$

This is an approximation of the inverse problem defined by (2.1). The approximate posterior on θ given y has density $\pi^{(M)}$ defined by the approximate log likelihood arising from this approximate inverse problem. In the *sample* step of the methodology, we apply MCMC methods to sample from $\pi^{(M)}$.

The overall framework, comprising the three steps calibrate, emulate, and sample is cheap to implement because it involves a small number of evaluations of $\mathcal{G}(\cdot)$ (computed only during the calibration phase, using ensemble methods where no derivatives of $\mathcal{G}(\cdot)$ are required), and because the MCMC method, which may require many steps, only requires evaluation of the emulator $\mathcal{G}^{(M)}(\cdot)$ (which is cheap to evaluate) and not $\mathcal{G}(\cdot)$ (which is assumed to be costly to evaluate and is possibly only known noisily). On the other hand, the method has ingredients that make it amenable to produce a controlled approximation of the true posterior. This is true since the EKS steps generate points concentrating near the main support of the posterior so that the GP emulator provides an accurate approximation of $\mathcal{G}(\cdot)$ where it matters.[2] A depiction of the framework and the algorithms involved can be found in Fig. 1. In the rest of the paper, we use the acronym CES to denote the three step methodology. Furthermore we employ boldface on one letter when we wish to emphasize one of the three steps: calibration is emphasized by (**C**ES); emulation by (C**E**S); and sampling by (CE**S**).

[1] For any positive-definite symmetric matrix A, we define $\langle a, a'\rangle_A = \langle a, A^{-1}a'\rangle = \langle A^{-\frac{1}{2}}a, A^{-\frac{1}{2}}a'\rangle$ and $\|a\|_A = \|A^{-\frac{1}{2}}a\|$.

[2] By "main support" we mean a region containing the majority of the probability.

2.2. Calibrate – EKI and EKS

The use and benefits of ensemble Kalman methods to solve inverse or parameter calibration problems have been outlined in the introduction. We will employ particular forms of the ensemble Kalman inversion methodology that we have found to perform well in practice and that are amenable to analysis; however, other ensemble methods to solve inverse problems could be used in the calibration phase.

The basic EKI method is found by time-discretizing the following system of interacting particles [71]:

$$\frac{d\theta^{(j)}}{dt} = -\frac{1}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\theta^{(k)}) - \bar{\mathcal{G}}, \mathcal{G}(\theta^{(j)}) - y\rangle_{\Gamma_y}(\theta^{(k)} - \bar{\theta}), \tag{2.4}$$

where $\bar{\theta}$ and $\bar{\mathcal{G}}$ denote the sample means given by

$$\bar{\theta} = \frac{1}{J}\sum_{k=1}^{J}\theta^{(k)}, \qquad \bar{\mathcal{G}} = \frac{1}{J}\sum_{k=1}^{J}\mathcal{G}(\theta^{(k)}). \tag{2.5}$$

For use below, we also define $\Theta = \{\theta^{(j)}\}_{j=1}^{J}$ and the $p \times p$ matrix

$$\mathsf{C}(\Theta) = \frac{1}{J}\sum_{k=1}^{J}(\theta^{(k)} - \bar{\theta}) \otimes (\theta^{(k)} - \bar{\theta}). \tag{2.6}$$

The dynamical system (2.4) drives the particles to consensus, while also driving them to fit the data and hence solve the inverse problem (2.1). Time-discretizing the dynamical system leads to a form of derivative-free optimization to minimize the least squares misfit defined by (2.1) [35,71].

An appropriate modification of EKI, to attack the problem of sampling from the posterior π^y given by (2.3), is EKS [26]. Formally this is obtained by adding a prior-related damping term, as in [10], and a Θ-dependent noise to obtain

$$\frac{d\theta^{(j)}}{dt} = -\frac{1}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\theta^{(k)}) - \bar{\mathcal{G}}, \mathcal{G}(\theta^{(j)}) - y\rangle_{\Gamma_y}(\theta^{(k)} - \bar{\theta}) - \mathsf{C}(\Theta)\Gamma_\theta^{-1}\theta^{(j)} + \sqrt{2\mathsf{C}(\Theta)}\,\frac{dW^{(j)}}{dt}, \tag{2.7}$$

where the $\{W^{(j)}\}$ are a collection of i.i.d. standard Brownian motions in the parameter space $\mathbb{R}^p$. The resulting interacting particle system approximates a mean-field Langevin-McKean diffusion process which, for linear $\mathcal{G}$, is invariant with respect to the posterior distribution (2.3) and, more generally, concentrates close to it; see [26] and [25] for details. The specific algorithm that we implement here time-discretizes (2.7) by means of a linearly implicit split-step scheme given by [26]

$$\theta_{n+1}^{(*,j)} = \theta_n^{(j)} - \Delta t_n \frac{1}{J}\sum_{k=1}^{J}\langle \mathcal{G}(\theta_n^{(k)}) - \bar{\mathcal{G}}, \mathcal{G}(\theta_n^{(j)}) - y\rangle_{\Gamma_y}\theta_n^{(k)} - \Delta t_n\,\mathsf{C}(\Theta_n)\,\Gamma_\theta^{-1}\,\theta_{n+1}^{(*,j)}, \tag{2.8a}$$

$$\theta_{n+1}^{(j)} = \theta_{n+1}^{(*,j)} + \sqrt{2\,\Delta t_n\,\mathsf{C}(\Theta_n)}\,\xi_n^{(j)}, \tag{2.8b}$$

where $\xi_n^{(j)} \sim \mathsf{N}(0, I)$, Γ_θ is the prior covariance and Δt_n is an adaptive timestep given in [26], and based on methods developed for EKI in [44]. This is a split-step method for the SDE (2.7) which is linearly implicit and ensures approximation of the Itô interpretation of the equation; other discretizations can be used, provided they are consistent with the Itô interpretation. The finite J correction to (2.7) proposed in [58], and further developed in [25], can easily be incorporated into the explicit step of this algorithm, and other time-stepping methods can also be used.

2.3. Emulate – GP emulation

The ensemble-based algorithm described in the preceding subsection produces input-output pairs $\{\theta_n^{(i)}, \mathcal{G}(\theta_n^{(i)})\}_{i=1}^{J}$ for $n = 0, \ldots, N$. For $n = N$ and J large enough, the samples of θ are approximately drawn from the posterior distribution. We use a subset of cardinality $M \le JN$ of this design as training points to update a GP prior to obtain the function $\mathcal{G}^{(M)}(\cdot)$ that will be used instead of the true forward model $\mathcal{G}(\cdot)$. The cardinality M denotes the total number of evaluations of $\mathcal{G}$ used in training the emulator $\mathcal{G}^{(M)}$. Recall that throughout this paper we take $M = J$ and use the output of the EKS in the last step of the iteration as the design.

The forward model is a multioutput map $\mathcal{G} : \mathbb{R}^p \mapsto \mathbb{R}^d$. It often suffices to emulate each output coordinate $l = 1, \ldots, d$ independently; however, variants on this are possible, and often needed, as discussed at the end of this subsection. For the moment, let us consider the emulation of the l-th component in $\mathcal{G}(\theta)$; denoted by $\mathcal{G}_l(\theta)$. Rather than interpolate the data, we assume that the input-output pairs are polluted by additive noise.[3] We place a Gaussian process prior with zero or linear mean function on the l-th output of the forward model and, for example, use the squared exponential kernel

$$k_l(\theta, \theta') = \sigma_l^2 \exp\left(-\frac{1}{2}\|\theta - \theta'\|_{D_l}^2\right) + \lambda_l^2\, \delta_\theta(\theta'), \tag{2.9}$$

where σ_l denotes the amplitude of the covariance kernel; $\sqrt{D_l} = \operatorname{diag}(\ell_1^{(l)}, \dots, \ell_p^{(l)})$ is the diagonal matrix of lengthscale parameters; $\delta_x(y)$ is the Kronecker delta function; and λ_l the standard deviation of the observation process. The hyperparameters $\phi_l = (\sigma_l^2, D_l, \lambda_l^2)$, which are learnt from the input-output pairs along with the regression coefficients of the GP, account for signal strength (σ_l^2); sensitivity to changes in each parameter component (D_l); and the possibility of white noise with variance λ_l^2 in the evaluation of the l-th component of the forward model $\mathcal{G}_l(\cdot)$. We adopt an empirical Bayes approach to learn the hyperparameters of each of the d Gaussian processes.

The final emulator is formed by stacking each of the GP models in a vector,

$$\mathcal{G}^{(M)}(\theta) \sim \mathrm{N}\left(m(\theta), \Gamma_{\mathrm{GP}}(\theta)\right). \tag{2.10}$$

The noise η typically found in (2.1) needs to be incorporated along with the noise $\eta_{\mathrm{GP}}(\theta) \sim \mathrm{N}(0, \Gamma_{\mathrm{GP}}(\theta))$ in the emulator $\mathcal{G}^{(M)}(\theta)$, resulting in the inverse problem

$$y = m(\theta) + \eta_{\mathrm{GP}}(\theta) + \eta. \tag{2.11}$$

We assume that $\eta_{\mathrm{GP}}(\theta)$ and η are independent of one another. In some cases, one or other of the sources of noise appearing in (2.11) may dominate the other and we will then neglect the smaller one. If we neglect η, we obtain the negative log-likelihood

$$\Phi_{\mathrm{GP}}^{(M)}(\theta) = \frac{1}{2}\|y - m(\theta)\|_{\Gamma_{\mathrm{GP}}(\theta)}^2 + \frac{1}{2}\log\det\Gamma_{\mathrm{GP}}(\theta); \tag{2.12}$$

for example, in situations where initial conditions of a dynamical systems are not known exactly. This situation is encountered in applications where $\mathcal{G}$ is defined through time-averaging, as in Section 5; in these applications the noise $\eta_{\mathrm{GP}}(\theta)$ is the major source of uncertainty and we take $\eta = 0$. If, on the other hand, we neglect η_{GP} then we obtain negative log-likelihood

$$\Phi_{\mathrm{m}}^{(M)}(\theta) = \frac{1}{2}\|y - m(\theta)\|_{\Gamma_y}^2. \tag{2.13}$$

If both noises are incorporated, then (2.12) is modified to give

$$\Phi_{\mathrm{GP}}^{(M)}(\theta) = \frac{1}{2}\|y - m(\theta)\|_{\Gamma_{\mathrm{GP}}(\theta)+\Gamma_y}^2 + \frac{1}{2}\log\det\bigl(\Gamma_{\mathrm{GP}}(\theta) + \Gamma_y\bigr); \tag{2.14}$$

this is used in Section 4.

We note that the specifics of the GP emulation could be adapted – we could use correlations in the output space $\mathbb{R}^d$, other kernels, other mean functions, and so forth. A review of multioutput emulation in the context of machine learning can be found in [4] and references therein; specifics on decorrelating multioutput coordinates can be found in [33]; and recent advances in exploiting covariance structures for multioutput emulation in [8,9]. For the sake of simplicity, we will focus on emulation techniques that preserve the strategy of determining, and then learning, approximately uncorrelated components; methodologies for transforming variables to achieve this are discussed in Appendix A.

2.4. Sample – MCMC

For the purposes of the MCMC algorithm, we need to initialize the Markov chain and choose a proposal distribution. The Markov chain is initialized with θ_0 drawn from the support of the posterior; this ensures that the MCMC has a short transient phase. To initialize, we use the ensemble mean of the EKS at the last iteration, $\bar{\theta}_J$. For the MCMC step we use a proposal of random walk Metropolis type, employing a multivariate Gaussian distribution with covariance given by the empirical covariance of the ensemble from EKS. We are thus pre-conditioning the sampling phase of the algorithm with approximate information about the posterior from the EKS. The resulting MCMC method is summarized in the following steps and is iterated until a desired number $N_s \gg J$ of samples $\{\theta_n\}_{n=1}^{N_s}$ is generated:

1. Choose $\theta_0 = \bar{\theta}_J$.
2. Propose a new parameter choice $\theta_{n+1}^* = \theta_n + \xi_n$ where $\xi_n \sim N(0, C(\Theta_J))$.

[3] The paper [5] interprets the use of additive noise within computer code emulation.

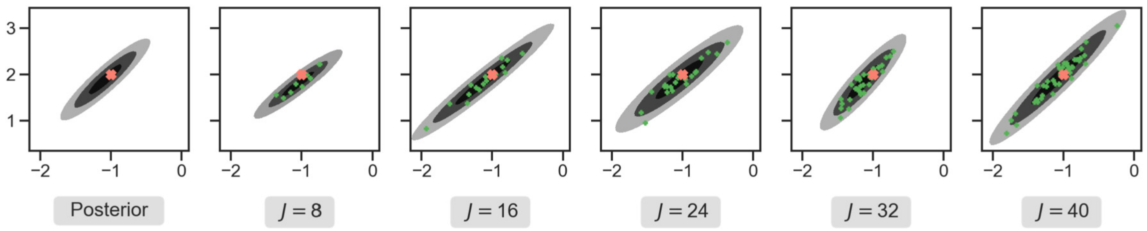


Fig. 2. Density estimates for different ensemble sizes used for the calibration step. Leftmost panel show the true posterior distribution. The green dots show the EKS at the 20-th iteration. The contour levels show the density of a Gaussian with mean and covariance estimated from EKS at said iteration. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3. Set $\theta_{n+1} = \theta^*_{n+1}$ with probability $a(\theta_n, \theta^*_{n+1})$; otherwise set $\theta_{n+1} = \theta_n$.
4. $n \to n+1$, return to 2.

The acceptance probability is computed as:

$$a(\theta, \theta^*) = \min\left\{1, \exp\left[\left(\Phi_\cdot^{(M)}(\theta^*) + \frac{1}{2}\|\theta^*\|^2_{\Gamma_\theta}\right) - \left(\Phi_\cdot^{(M)}(\theta) + \frac{1}{2}\|\theta\|^2_{\Gamma_\theta}\right)\right]\right\}, \tag{2.15}$$

where $\Phi_\cdot^{(M)}$ is defined in (2.12), (2.13) or (2.14), whichever is appropriate.

3. Linear problem

By choosing a linear parameter-to-data map, we illustrate the methodology in a case where the posterior is Gaussian and known explicitly. This demonstrates both the viability and accuracy of the method in a transparent fashion.

3.1. *Linear inverse problem*

We consider a linear forward map $\mathcal{G}(\theta) = G\theta$, with $G \in \mathbb{R}^{d\times p}$. Each row of the matrix G is a p-dimensional draw from a multivariate Gaussian distribution. Concretely we take $p = 2$ and each row $G_i \sim \mathsf{N}(0, \Sigma)$, where $\Sigma_{12} = \Sigma_{21} = -0.9$, and $\Sigma_{11} = \Sigma_{22} = 1$. The synthetic data we have available to perform the Bayesian inversion is then given by

$$y = G\theta^\dagger + \eta, \tag{3.1}$$

where $\theta^\dagger = [-1, 2]^\top$, and $\eta \sim \mathsf{N}(0, \Gamma)$ with $\Gamma = 0.1^2 I$.

We assume that, a priori, parameter $\theta \sim \mathsf{N}(m_\theta, \Sigma_\theta)$. In this linear Gaussian setting the solution of the Bayesian linear inverse problem is itself Gaussian [see 27, Part IV] and given by the Gaussian distribution

$$\pi^y(\theta) \propto \exp\left(-\frac{1}{2}\|\theta - m_{\theta|y}\|^2_{\Sigma_{\theta|y}}\right), \tag{3.2}$$

where $m_{\theta|y}$ and $\Sigma_{\theta|y}$ denote the posterior mean and covariance. These are computed as

$$\Sigma^{-1}_{\theta|y} = G^\top \Sigma_y^{-1} G + \Sigma_\theta^{-1}, \qquad m_{\theta|y} = \Sigma_{\theta|y}\left(G^\top \Sigma_y^{-1} y + \Sigma_\theta^{-1} m_\theta\right). \tag{3.3}$$

3.2. *Numerical results*

For the calibration step (**C**ES) we consider the EKS algorithm. Fig. 2 shows how the EKS samples estimate the posterior distribution (the far left). The green dots correspond to 20-th iteration of EKS with different ensemble sizes. We also display in gray contour levels the density corresponding to the 67%, 90% and 99% probability levels under a Gaussian with mean and covariance estimated from EKS at said 20-th iteration. This allows us to visualize the difference between the results of EKS and the true posterior distribution in the leftmost panel. In this linear case, the mean-field limit of EKS exactly reproduces the invariant measure [26]. The mismatch between the EKS samples and the true posterior can be understood from the fact that time discretizations of Langevin diffusions are known to induce errors if no metropolization scheme is added to the dynamics [49,64,66], and from the finite number of particles used; the latter could be corrected by using the ideas introduced in [58] and further developed in [25].

The GP emulation step (C**E**S) is depicted in Fig. 3 for one component of $\mathcal{G}$. Each GP is trained using the 20-th iteration of EKS for each ensemble size. We employ a zero mean GP with the squared-exponential kernel (2.9). We add a regularization term to the lengthscales of the GP by means of a prior in the form of a Gamma distribution. This is common in GP Bayesian inference when the domain of the input variables is unbounded. The choice of this regularization ensures that the

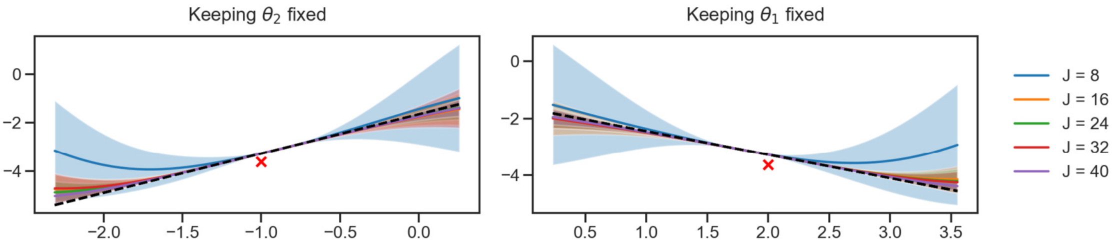


Fig. 3. Gaussian process emulators learnt using different training datasets. The input-output pairs are obtained from the calibration step using EKS. Each color for the lines and the shaded regions correspond to different ensemble sizes as described in the legend.

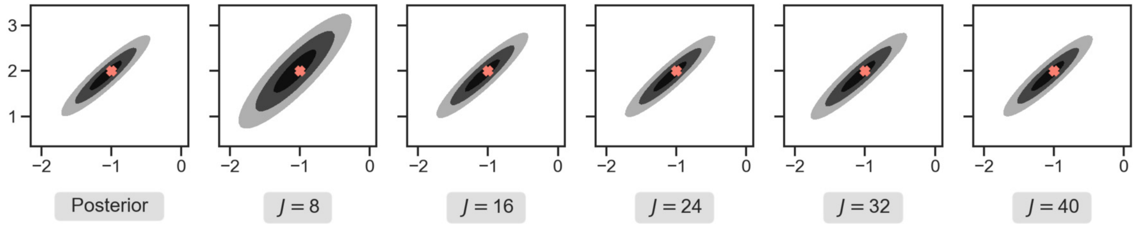


Fig. 4. Density of a Gaussian with mean and covariance estimated from GP-based MCMC samples using $M = J$ design points. The true posterior distribution is shown in the far left. Each GP-based MCMC generated 2×10^4 samples. These samples are not shown for clarity.

Table 1
Averaged mean square error of the posterior location (mean) computed from 20 independent realizations.

Ensemble size	8	16	24	32	40
EKS	0.1171	0.0665	0.0528	0.0584	0.0648
CES	0.0652	0.0219	0.0125	0.0122	0.0173

Table 2
Averaged mean square error of the posterior spread (covariance) computed by means of the Frobenius norm and 20 independent realizations.

Ensemble size	8	16	24	32	40
EKS	0.1010	0.0703	0.0937	0.0868	0.0664
CES	0.0555	0.0241	0.0092	0.0050	0.0056

covariance kernels, which are regression functions for the mean of the emulator, decay fast away from the data, and that no short variations below the levels of available data are introduced [27,28]. We can visualize the emulator of the component of the linear system considered here by fixing one parameter at the true value while varying the other. The dashed reference in Fig. 3 shows the model $G\theta$. The red cross denotes the corresponding observation. The solid lines correspond to the mean of the GP, while the shaded regions contain 2 standard deviations of predicted variability. Colors correspond to different ensemble sizes as described in the figure's legend. We can see in Fig. 3 that the GP increases its accuracy as the amount of training data is increased. In the end, for training sets of size $J \geq 16$, it correctly simulates the linear model with low uncertainty in the main support of the posterior.

Fig. 4 depicts results in the sampling step (CE**S**). These are obtained by using a GP approximation of $G\theta$ within MCMC. The GP-based MCMC uses (2.14) since the forward model is deterministic and the data is polluted by noise. The contour levels show a Gaussian distribution with mean and covariance estimated from $N_s = 2 \times 10^4$ GP-based MCMC samples (not shown) in each of the different ensemble settings. The results show that the true posterior is captured with an ensemble size of 16 or more. Moreover, Table 1 shows the mean square error of the posterior location parameter in (3.3); that is, the error in Euclidean norm of the sample-based mean. This error is computed by means of the ensemble mean and the analytic posterior mean (top row), and the CES posterior mean and the analytic solution (bottom row). Analogously, Table 2 shows the mean square error for the spread; that is, the Frobenius norm error in the sample-based covariance matrix – both computed from the EKS and CES samples, top and bottom rows, respectively – and the analytic solution for the posterior covariance in (3.3). For both parameters, it can be seen that the CES-approximated posterior achieves higher accuracy relative to the EKS alone. For this linear problem both methods provably recover the true posterior distribution, in the limit of large ensemble size, so that it is interesting to see that the CES-approximated posterior improves upon the EKS alone. Recall from the discussion in Subsection 1.2, however, that for nonlinear problems EKS does not in general recover the posterior distribution, whilst the CES-approximated posterior converges to the true posterior distribution as the number of training samples increases, regardless of the distribution of the EKS particles; what is beneficial, in general, about using EKS to design the GP is that the samples are located close to the support of the true posterior, even if they are not distributed according to the posterior.

4. Darcy flow

In this section, we apply our methodology to a PDE nonlinear inverse problem arising in porous medium flow: the determination of permeability from measurements of the pressure (piezometric head).

4.1. Elliptic inverse problem

4.1.1. Forward problem

The forward problem is to find the pressure field p in a porous medium defined by the (for simplicity) scalar permeability field a. Given a scalar field f defining sources and sinks of fluid, and assuming Dirichlet boundary conditions for p on the domain $D = (0,1)^2$, we obtain the following elliptic PDE determining the pressure from permeability:

$$-\nabla \cdot (a(x)\nabla p(x)) = f(x), \quad x \in D. \tag{4.1a}$$

$$p(x) = 0, \quad x \in \partial D. \tag{4.1b}$$

In this paper we always take $f \equiv 1$. The unique solution of the equation implicitly defines a map from $a \in L^\infty(D;\mathbb{R})$ to $p \in H_0^1(D;\mathbb{R})$.

4.1.2. Inverse problem

We assume that the permeability is dependent on unknown parameters $\theta \in \mathbb{R}^p$, so that $a(x) = a(x;\theta) > 0$ almost everywhere in D. The inverse problem of interest is to determine θ from noisy observations of d linear functionals (measurements) of $p(x;\theta)$. Thus,

$$\mathcal{G}_j(\theta) = \ell_j\big(p(\cdot;\theta)\big) + \eta, \qquad j = 1, \cdots, d. \tag{4.2}$$

We assume the additive noise η to be a mean zero Gaussian with covariance equal to $\gamma^2 I$. Throughout this paper, we work with pointwise measurements so that $\ell_j(p) = p(x_j)$.[4] We employ $d = 50$ measurement locations chosen at random from a uniform grid in the interior of D.

We introduce a log-normal parameterization of $a(x;\theta)$ as follows:

$$\log a(x;\theta) = \sum_{\ell \in K_q} \theta_\ell \sqrt{\lambda_\ell}\, \varphi_\ell(x) \tag{4.3}$$

where

$$\varphi_\ell(x) = \cos\big(\pi \langle \ell, x\rangle\big), \qquad \lambda_\ell = (\pi^2|\ell|^2 + \tau^2)^{-\alpha}; \tag{4.4}$$

the smoothness parameters are assumed to be $\tau = 3$, $\alpha = 2$, and $K_q \subset K \equiv \mathbb{Z}^2$ is the set, with finite cardinality $|K_q| = q$, of indices over which the random series is summed. *A priori* we assume that $\theta_\ell \sim \mathrm{N}(0,1)$ so that we have a Karhunen-Loève representation of a as a log–Gaussian random field [62]. We often find it helpful to write (4.3) as a sum over a one-dimensional variable rather than a lattice:

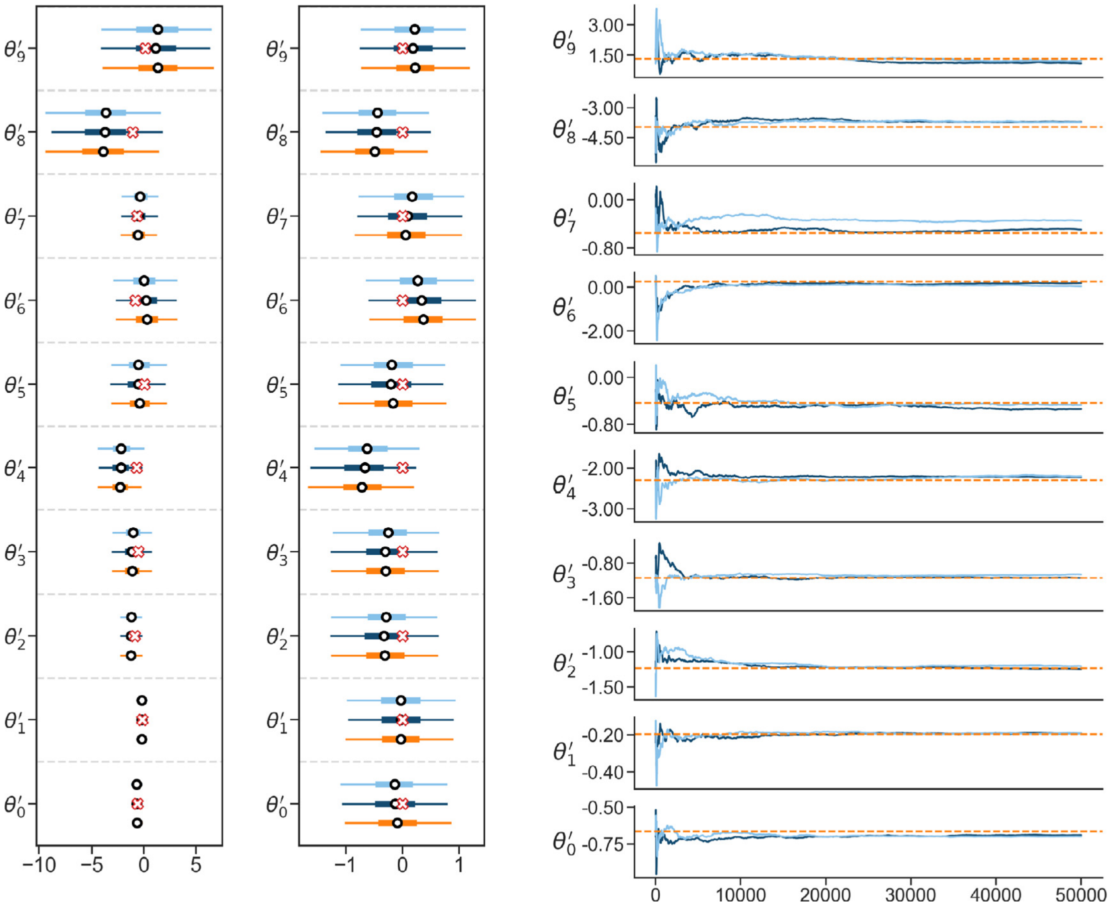
$$\log a(x;\theta') = \sum_{k \in Z_q} \theta'_k \sqrt{\lambda'_k}\, \varphi'_k(x). \tag{4.5}$$

Throughout this paper we choose K_q, and hence Z_q, to contain the q largest $\{\lambda_\ell\}$. We order the indices in $Z_q \subset \mathbb{Z}^+$ so that the set of $\{\lambda'_k\}$ are non-increasing with respect to k.

4.2. Numerical results

We generate an underlying true random field by sampling $\theta^\dagger \in \mathbb{R}^p$ from a standard multivariate Gaussian distribution, $\mathrm{N}(0, I_p)$, of dimension $p = 16^2 = 256$. This is used as the coefficients in (4.3) by means of the re-labeling (4.5). The evaluation of the forward model $\mathcal{G}(\theta)$ requires solving the PDE (4.1b) for a given realization of the random field a. This is done with a cell-centered finite difference scheme [6,67]. We create data y from (4.2) with a random perturbation

[4] The implied linear functionals are not elements of the dual space of $H_0^1(D;\mathbb{R})$ in dimension 2 but mollifications of them are. In practice, mollification with a narrow kernel does not affect results of the type presented here [36], and so we do not use it.



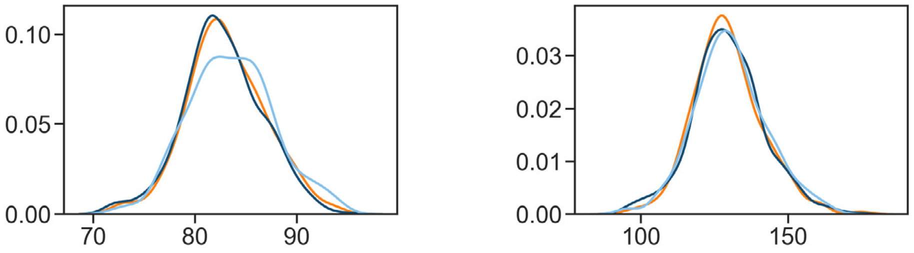
(a) Original θ. (b) Rescaled θ. (c) MCMC traceplot

Fig. 5. Results of CES in the Darcy flow problem. Colors throughout the panel denote results using different calibration and GP training settings. This are: light blue – ensemble of size $J = 128$; dark blue – ensemble of size $J = 512$; and orange, the MCMC gold standard. Left panel shows each θ component for CES. The middle panel shows the same information, but using standardized components of θ. The interquartile range is displayed with a thick line; the 95% credible intervals with a thin line; and the median with circles. The right panel shows typical MCMC running averages, demonstrating stationarity of the Markov chain.

$\eta \sim \mathrm{N}(0, 0.005^2 \times I_d)$, where I_d denotes the identity matrix. The locations for the data, and hence the evaluation of the forward model, were chosen at random from the 16^2 computational grid used to solve the Darcy flow. For the Bayesian inversion we use a truncation of (4.5) with $p' < p$ terms, allowing us to avoid the inverse crime of using the same model that generated the data to solve the inverse problem [39]. Specifically, we consider $p' = 10$. We employ a non-informative centered Gaussian prior with covariance $\Gamma_\theta = 10^2 \times I_{p'}$; this is also used to initialize the ensemble for EKS. We consider ensembles of size $J \in \{128, 512\}$.

We perform the complete CES procedure starting with EKS as described above for the calibration step (**C**ES). The emulation step (C**E**S) uses a GP with a linear mean Gaussian process with squared-exponential kernel (2.9). Empirically, the linear mean allows us to capture a significant fraction of the relevant parameter response. The GP covariance matrix $\Gamma_{\mathrm{GP}}(\theta)$ accounts for the variability of the residuals from the linear function. The sampling step (CE**S**) is performed using the Random Walk procedure described in Section 2.4; a Gaussian transition distribution is used, found by matching to the first two moments of the ensemble at the last iteration of EKS. In this experiment, the likelihood (2.14) is used because the forward model is a deterministic map, and we have data polluted by additive noise.

We compare the results of the CES procedure with those obtained from a gold standard MCMC employing the true forward model. The results are summarized in Fig. 5. The right panel shows typical MCMC running averages, suggesting stationarity of the Markov chain. The left panel shows the forest plot of each θ component. The middle panel shows the standardized components of θ. These forest plots show the interquartile range with a thick line; the 95% credible intervals with a thin line; and the median with circles. The true value of the parameters are denoted by red crosses. The results demonstrate that the CES methodology accurately reproduces the true posterior using calibration and training with $M = J = 512$ ensemble members. For the smaller ensemble, $M = J = 128$ there is a visible systematic deviation in some components, like θ_7. However, the CES posterior does capture the true value. Note that the gold standard MCMC employs



(a) PDF of number lattice points exceeding the pressure threshold.

(b) PDF of number lattice points exceeding the permeability threshold.

Fig. 6. Forward UQ exercise of exceedance on both the pressure field, $p(\cdot) > \bar{p}$, and permeability field, $a(\cdot) > \bar{a}$. Both threshold levels are obtained from the forward model at the truth $\theta^\dagger$ and taking the median across the locations (4.2). The PDFs are constructed by running the forward model on a small set of samples, $N_{\text{UQ}} = 250$, and computing the number of lattice points exceeding the threshold. The samples are obtained by using the CES methodology (light blue – CES with $M = J = 128$, and dark blue – CES with $M = J = 512$). The samples in orange are obtained from a gold standard MCMC using the true forward model within the likelihood, rather than the emulator.

uses tens of thousands of evaluations of the map from θ to y, where as the CES methodology requires only hundreds, and yet produces similar results.

The results from the CES procedure are also used in a forward UQ setting: posterior variability in the permeability is pushed forward onto quantities of interest. For this purpose, we consider exceedances on the pressure and permeability fields above certain thresholds. These thresholds are computed from the observed data by taking the median across the 50 available locations (4.2). The forward model (4.1b) is solved with $N_{\text{UQ}} = 500$ different parameter settings coming from samples of the CES Bayesian posterior. We also show the comparison with the gold standard using the true forward model. We evaluate the pressure and the permeability field at each lattice point, denoted by $\ell \in K_q$, and compare with the observed threshold levels computed from the 50 available locations, denoted by j in (4.2). We record the number of lattice points exceeding such bounds for each of the N_{UQ} samples. Fig. 6 shows the corresponding KDE for the probability density function (PDF) in this forward UQ exercise. The orange lines correspond to the PDF of the number of points in the lattice that exceed the threshold computed from the samples drawn using MCMC with the Darcy flow model. The corresponding PDFs associated to the CES posterior, based on calibration and emulation using different ensemble sizes, are shown in different blue tonalities (light blue – CES with $M = J = 128$, and dark blue – CES with $M = J = 512$). We use a k-sample Anderson–Darling test to find evidence against the null hypothesis that assumes that the forward UQ samples using the CES procedure are statistically similar to samples from the true distribution. This test is non-parametric and relies on the comparison of the empirical cumulative functions [43, See Ch. 13]. Applying the k-sample Anderson–Darling test at 5% significance level for the $M = J = 128$ case, shows evidence to reject the null hypothesis of the samples being drawn from the same distribution in the pressure exceedance forward UQ. This means that with such limited number of forward model evaluations, the CES procedure is not able to generate samples that seem to be generated from the true distribution. In the case of having more forward model evaluations, such test does not provide statistical evidence to reject the hypothesis that the distributions are similar to the one based on the Darcy model.

5. Time-averaged data

In parameter estimation problems for chaotic dynamical systems, such as those arising in climate modeling [13,37,73], data may only be available in time-averaged form; or it may be desirable to study time-averaged quantities in order to ameliorate difficulties arising from the complex objective functions, with multiple local minima, which arise from trying to match trajectories [1]. Indeed the idea fits the more general framework of feature-based data assimilation introduced in [57] which, in turn, is closely related to the idea of extracting sufficient statistics from the raw data [24]. The methodology developed in this section underpins similar work conducted for a complex climate model described in the paper [13].

5.1. Inverse problems from time-averaged data

The problem is to estimate the parameters $\theta \in \mathbb{R}^p$ of a dynamical system evolving in $\mathbb{R}^m$ from data y comprising time-averages of an $\mathbb{R}^d$−valued function $\varphi(\cdot)$. We write the dynamical system as

$$\dot{z} = F(z; \theta), \quad z(0) = z_0. \tag{5.1}$$

Since $z(t) \in \mathbb{R}^m$ and $\theta \in \mathbb{R}^p$ we have $F : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^m$ and $\varphi : \mathbb{R}^m \to \mathbb{R}^d$. We will write $z(t; \theta)$ when we need to emphasize the dependence of a trajectory on θ. In view of the time-averaged nature of the data it is useful to define the operator

$$\mathcal{G}_\tau(\theta; z_0) = \frac{1}{\tau} \int_{T_0}^{T_0+\tau} \varphi(z(t;\theta))dt \tag{5.2}$$

where T_0 is a predetermined spinup time, τ is the time horizon over which the time-averaging is performed, and z_0 the initial condition of the trajectory used to compute the time-average. Our approach proceeds under the following assumptions:

Assumptions 1. The dynamical system (5.1) satisfies:

1. For every $\theta \in \Theta$, (5.1) has a compact attractor $\mathcal{A}$, supporting an invariant measure $\mu(dz;\theta)$. The system is ergodic, and the following limit – a Law of Large Numbers (LLN) analogue – is satisfied: for z_0 chosen at random according to measure $\mu(\cdot;\theta)$ we have, with probability one,

$$\lim_{\tau\to\infty} \mathcal{G}_\tau(\theta; z_0) = \mathcal{G}(\theta) := \int_{\mathcal{A}} \varphi(z)\mu(dz;\theta). \tag{5.3}$$

2. We have a Central Limit Theorem (CLT) quantifying the ergodicity: for z_0 distributed according to $\mu(dz;\theta)$,

$$\mathcal{G}_\tau(\theta; z_0) \approx \mathcal{G}(\theta) + \frac{1}{\sqrt{\tau}}\mathsf{N}(0, \Sigma(\theta)). \tag{5.4}$$

In particular, the initial condition plays no role in time averages over the infinite time horizon. However, when finite time averages are employed, different random initial conditions from the attractor give different random errors from the infinite time-average, and these, for fixed spinup time T_0, are approximately Gaussian. Furthermore, the covariance of the Gaussian depends on the parameter θ at which the experiment is conducted. This is reflected in the noise term in (5.4).

Here we will assume that the model is perfect in the sense that the data we are presented with could, in principle, be generated by (5.1) for some value(s) of θ and z_0. The only sources of uncertainty come from the fact that the true value of θ is unknown, as is the initial condition z_0. In many applications, the values of τ that are feasible are limited by computational cost. The explicit dependence of $\mathcal{G}_\tau$ on τ serves to highlight the effect of τ on the computational budget required for each forward model evaluation. Use of finite time-averages also introduces the unwanted nuisance variable z_0 whose value is typically unknown, but not of intrinsic interest. Thus, the inverse problem that we wish to solve is to find θ solving the equation

$$y = \mathcal{G}_T(\theta; z_0) \tag{5.5}$$

where z_0 is a latent variable and T is the computationally-feasible time window we can integrate the system (5.1). We observe that the preceding considerations indicate that it is reasonable to assume that

$$y = \mathcal{G}(\theta) + \eta\,, \tag{5.6}$$

where $\eta \sim \mathsf{N}(0, \Gamma_y(\theta))$ and $\Gamma_y(\theta) = T^{-1}\Sigma(\theta)$. We will estimate $\Gamma_y(\theta)$ in two ways: firstly using long-time series data; and secondly using a GP informed by forward model evaluations.

We first estimate $\Gamma_y(\theta)$ directly from $\mathcal{G}_\tau$ with $\tau \gg T$. We will not employ $\theta-$ dependence in this setting and simply estimate a fixed covariance Γ_{obs}. This is because, in the applications we envisage such as climate modeling [13], long time-series data over time-horizon τ will typically be available only from observational data. The cost of repeatedly simulating at different candidate θ values is computationally prohibitive, in contrast, to simulations over a shorter time-horizon T. We apply EKS to make an initial calibration of θ from y given by (5.5), using $\mathcal{G}_T(\theta^{(j)}; z_0^{(j)})$ in place of $\mathcal{G}(\theta^{(j)})$ and Γ_{obs} in place of $\Gamma_y(\cdot)$, within the discretization (2.8) of (2.7). We find the method to be insensitive to the exact choice of $z_0^{(j)}$, and typically use the final value of the dynamical system computed in the preceding step of the ensemble Kalman iteration. We then take the evaluations of $\mathcal{G}_T$ as noisy evaluations of $\mathcal{G}$, from which we learn the Gaussian process $\mathcal{G}^{(M)}$. We use the mean $m(\theta)$ of this Gaussian process as an estimate of $\mathcal{G}$. Our second estimate of $\Gamma_y(\theta)$ is obtained by using the covariance of the Gaussian process $\Gamma_{\mathrm{GP}}(\theta)$. We can evaluate the misfit through either of the expressions

$$\Phi_{\mathrm{m}}(\theta; y) = \frac{1}{2}\|y - m(\theta)\|^2_{\Gamma_{\mathrm{obs}}}, \tag{5.7a}$$

$$\Phi_{\mathrm{GP}}(\theta; y) = \frac{1}{2}\|y - m(\theta)\|^2_{\Gamma_{\mathrm{GP}}(\theta)} + \frac{1}{2}\log\det\Gamma_{\mathrm{GP}}(\theta). \tag{5.7b}$$

Note that equations (5.7) are the counterparts of (2.12) and (2.13) in the setting with time-averaged data. In what follows, we will contrast these misfits, both based on the learnt GP emulator, with the misfit that uses the noisy evaluations $\mathcal{G}_T$ directly. That is, we use the misfit computed as

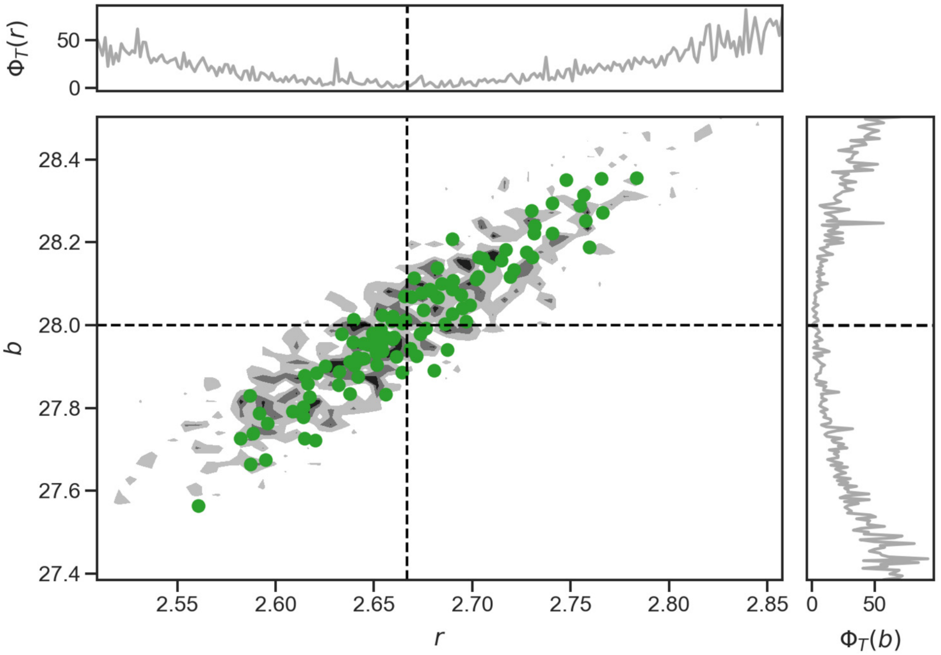


Fig. 7. Contour levels of the misfit of the Lorenz'63 forward model corresponding to (67%, 90%, 99%) density levels. The dotted lines shows the locations of the true parameter values that generated the data. The green dots shows the final ensemble of the EKS algorithm. The marginal plots show the misfit as a 1-d function keeping one parameter fixed at the truth while varying the other. This highlights the noisy response from the time-average forward model $\mathcal{G}_T$.

$$\Phi_T(\theta; y) = \frac{1}{2}\|y - \mathcal{G}_T(\theta)\|^2_{\Gamma_{\text{obs}}}. \tag{5.8}$$

In the latter, dependence of $\mathcal{G}_T$ on initial conditions is suppressed.

5.2. Numerical results – Lorenz'63

We consider the 3-dimensional Lorenz equations [52]

$$\dot{x}_1 = \sigma(x_2 - x_1), \tag{5.9a}$$

$$\dot{x}_2 = rx_1 - x_2 - x_1x_3, \tag{5.9b}$$

$$\dot{x}_3 = x_1x_2 - bx_3, \tag{5.9c}$$

with parameters $\sigma, b, r \in \mathbb{R}_+$. Our data is found by simulating (5.9) with $(\sigma, b, r) = (10, 28, 8/3)$, a value at which the system exhibits chaotic behavior. We focus on the inverse problem of recovering (r, b), with σ fixed at its true value of 10, from time-averaged data.

Our statistical observations are first and second moments over time windows of size $T = 10$. Our vector of observations is computed by taking $\varphi : \mathbb{R}^3 \mapsto \mathbb{R}^9$ to be

$$\varphi(x) = (x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3, x_3x_1). \tag{5.10}$$

This defines $\mathcal{G}_T$. To compute Γ_{obs} we used time-averages of $\varphi(x)$ over $\tau = 360$ units of time, at the true value of θ; we split the time-series into windows of size T and neglect an initial spinup of $T_0 = 30$ units of time. Together $\mathcal{G}_T$ and Γ_{obs} produce a noisy function Φ_T as depicted in Fig. 7. The noisy nature of the energy landscape, demonstrated in this figure, suggests that standard optimization and MCMC methods may have difficulties; the use of GP emulation will act to smooth out the noise and lead to tractable optimization and MCMC tasks.

For the calibration step (**C**ES), we run the EKS using the estimate of $\Gamma = \Gamma_{\text{obs}}$ within the algorithm (2.7), and within the misfit function (5.8), as described in Section 5.1. We assumed the parameters to be *a priori* governed by an isotropic Gaussian prior in logarithmic scale. The mean of the prior is $m_0 = (3.3, 1.2)^\top$ and its covariance is $\Sigma_0 = \text{diag}(0.15^2, 0.5^2)$. This gives broad priors for the parameters with 99% probability mass in the region $[20, 40] \times [0, 15]$. The results of evolving the EKS through 11 iterations can be seen in Fig. 7, where the green dots represent the final ensemble. The dotted lines locate the true underlying parameters in the (r, b) space.

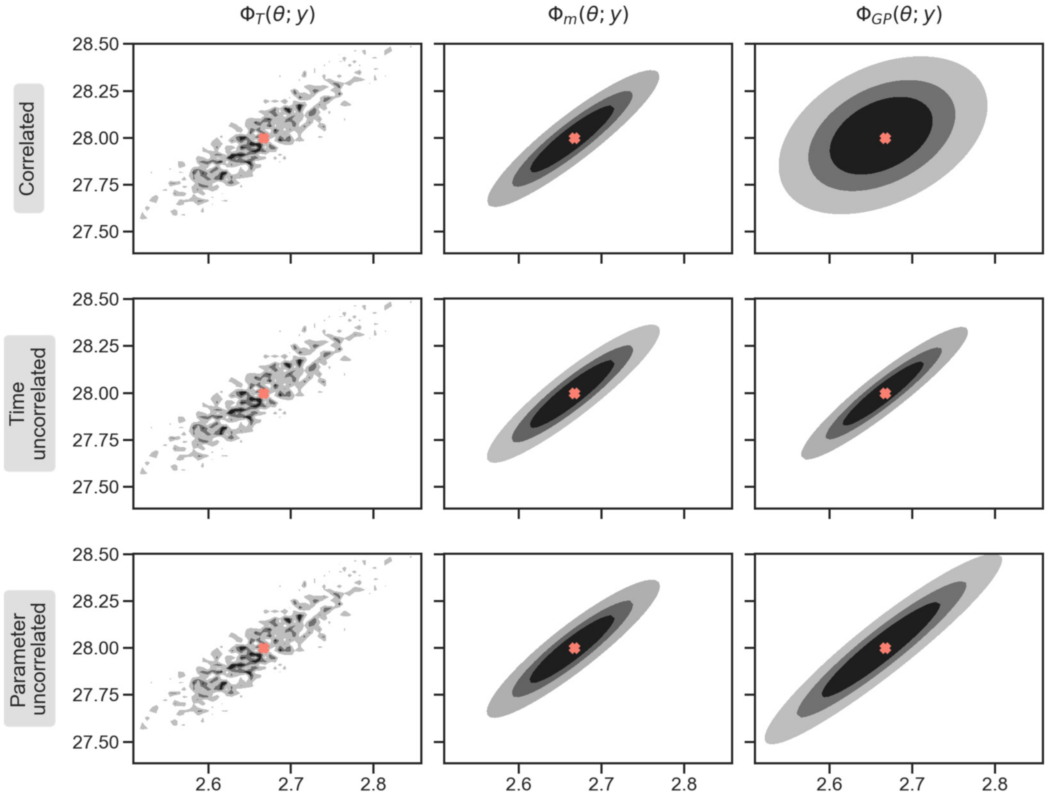


Fig. 8. Contour levels of the Lorenz'63 posterior distribution corresponding to (67%, 90%, 99%) density levels. For each row we depict: in the left panel, the contours using the true forward model; in the middle panel, the contours of the misfit computed as Φ_{m} (5.7a); and in the right panel, the contours of the misfit obtained using Φ_{GP} (5.7b). The difference between rows is due to the decorrelation strategy used to learn the GP emulator, as indicated in the leftmost labels. The GP-based densities show an improved estimation of uncertainty arising from GP estimation of the infinite time-averages, in comparison with employing the noisy exact finite time averages, for both decorrelation strategies.

For the emulation step (C**E**S), we use GP priors for each of the 9 components of the forward model. The hyper-parameters of these GPs are estimated using empirical Bayes methodology. The 9 components do not interact and are treated independently. We use only the input-output pairs obtained from the last iteration of EKS in this emulation phase, although earlier iterations could also have been used. This choice focuses the training runs in regions of high posterior probability. Overall, the GP allows us to capture the underlying smooth trend of the misfit. In Fig. 8 (top row) we show (left to right) Φ_T, Φ_{m}, and Φ_{GP} given by (5.7)–(5.8). Note that Φ_{m} produces a smoothed version of Φ_T, but that Φ_{GP} fails to do so – it is smooth, but the orientations and eccentricities of the contours are not correctly captured. This is a consequence of having only diagonal information to replace the full covariance matrix Γ by $\Gamma(\theta)$ and not learning dependencies between the 9 simulator outputs that comprise $\mathcal{G}_T$.

We explore two options to incorporate output dependency. These are detailed in Appendix A, and are based on changing variables according to either a diagonalization of Γ_{obs} or on an SVD of the centered data matrix formed from the EKS output used as training data $\{\mathcal{G}_T(\theta^{(i)})\}_{i=1}^{M}$. The effect on the emulated misfit when using these changes of variables is depicted in the middle and bottom rows of Fig. 8. We can see that the misfit Φ_{m} (5.7a) respects the correlation structure of the posterior. There is no notable difference between using a GP emulator in the original or decorrelated output system. This can been seen in the middle column in Fig. 8. However, if the variance information of the GP emulator is introduced to compute Φ_{GP} (5.7b), decorrelation strategies allows us to overcome the problems caused when using diagonal emulation.

Finally, the sample step (CE**S**) is performed using the GP emulator to accelerate the sampling and to correct for the mismatch of the EKS in approximating the posterior distribution, as discussed in Section 2.4. In this section, random walk metropolis is run using 5,000 samples for each setting – using the misfits Φ_T, Φ_{m} or Φ_{GP}. The Markov chains are initialized at the mean of the last iteration of the EKS. The proposal distribution used for the random walk is a Gaussian with covariance equal to the covariance of the ensemble at said last iteration. The samples are depicted in Fig. 9. The orange contour levels represent the kernel density estimate (KDE) of samples from a random walk Metropolis algorithm using the true forward model. On the other hand the blue contour levels represent the KDE of samples using Φ_{m} or Φ_{GP}, equations (5.7a) or (5.7b) respectively. The green dots in the left panels depict the final ensemble from EKS. It should be noted that using Φ_{m} for MCMC has an acceptance probability of around 41% in each of the emulation strategy (middle column). The acceptance rate increases slightly to around 47% by using Φ_{GP} (right column). The original acceptance rate is 16% if the true

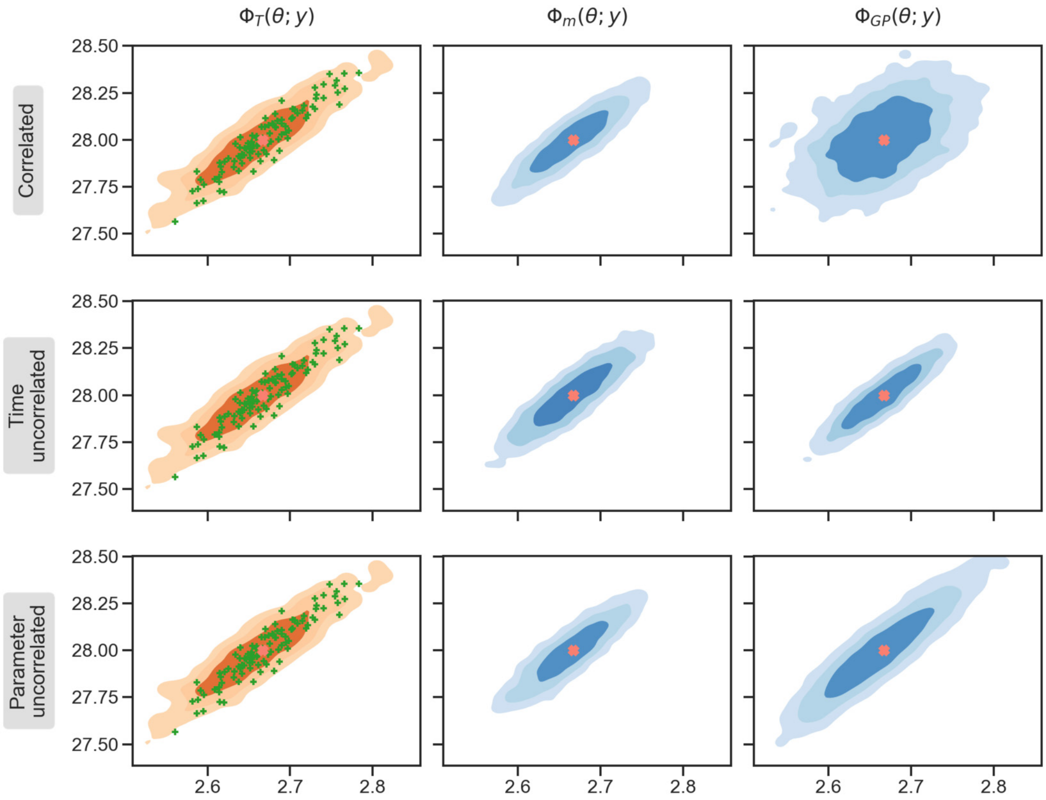


Fig. 9. Samples using different modalities of the GP emulator. The orange kernel density estimate (KDE) is based on samples from random walk Metropolis using the true forward model. The blue contour levels are KDE using the GP-based MCMC. All MCMC-based KDE approximate posteriors are computed from $N_s = 20,000$ MCMC samples. The green dots in the left panels depict the final ensemble from the EKS as a comparison. Furthermore, the CES-based densities are computed more easily as the MCMC samples decorrelate more rapidly due to a higher acceptance probability for the same size of proposed move.

forward model is employed. The main reason is the noisy landscape of the posterior distribution. In this experiment, the use of a GP emulator showcases the benefits of our approach as it allows to generate samples from the posterior distribution more efficiently than standard MCMC, not only because the emulator is faster to evaluate, but also because it smoothes the log-likelihood. Careful attention to how the emulator model is constructed and the use of nearly independent co-ordinates in data space, helps to make the approximate methodology viable.

5.3. Numerical results – Lorenz'96

We consider the multiscale Lorenz'96 model [51]. This model possesses properties typically present in the earth system [73] such as advective energy conserving nonlinearity, linear damping and large scale forcing, and multiscale coexistence of slow and fast variables. It comprises K slow variables X_k $(k = 1, \ldots K)$, each coupled to L fast variables $Y_{l,k}$ $(l = 1, \ldots, L)$. The dynamical system is written as

$$\frac{\mathrm{d}X_k}{\mathrm{d}t} = -X_{k-1}\left(X_{k-2} - X_{k+1}\right) - X_k + F - hc\,\bar{Y}_k \tag{5.11a}$$

$$\frac{1}{c}\frac{\mathrm{d}Y_{l,k}}{\mathrm{d}t} = -bY_{l+1,k}\left(Y_{l+2,k} - Y_{l-1,k}\right) - Y_{l,k} + \frac{h}{L}X_k, \tag{5.11b}$$

where $\bar{Y}_k = \frac{1}{L}\sum_{l=1}^{L} Y_{l,k}$. The slow and fast variables are periodic over k and l, respectively. This means that $X_{k+K} = X_k$, $Y_{l,k+K} = Y_{l,k}$, and $Y_{l+L,k} = Y_{l,k+1}$. This coupling of the fast variable $Y_{l,k}$ at index k to the fast variables at indices $k-1$ and $k+1$ has its roots in the physical interpretation as a simplified multi-scale atmosphere model. A geophysical interpretation may be found in [51].

The scale separation parameter, c, is naturally constrained to be a non-negative number. Thus, our methods consider the vector of parameters $\theta := (h, F, \log c, b)$. We perform Bayesian inversion for θ based on data averaged across the K locations and over time windows of length $T = 100$. To this end, we define our $k-$ indexed observation operator $\varphi_k : \mathbb{R} \times \mathbb{R}^L \mapsto \mathbb{R}^5$, by

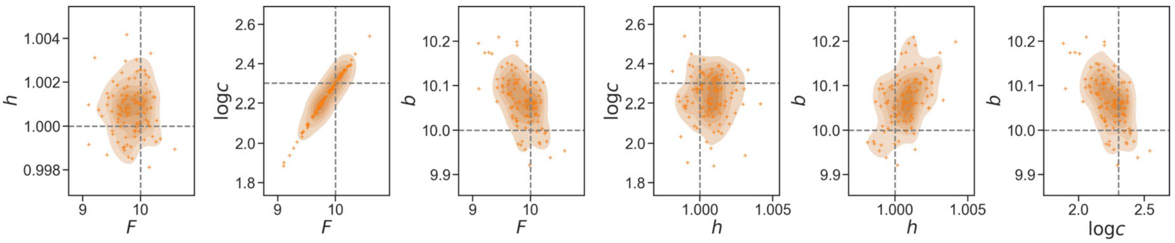


Fig. 10. Samples and kernel density estimates of EKS applied to the Lorenz'96 inverse problem. The ensemble, $J = 100$, shown corresponds to the last iteration.

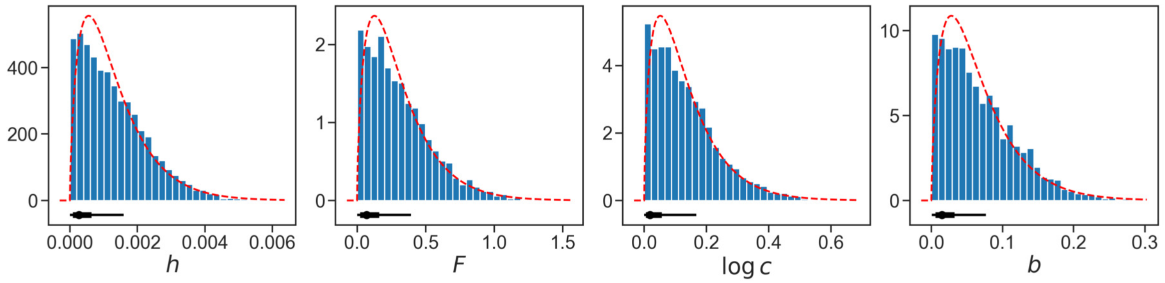


Fig. 11. Histograms of pairwise distances for every component of the unknown parameters θ using the last iteration of EKS. Red dashed lines show the kernel density estimate of the histograms. The black box plot at the bottom shows the elicited GP–lengthscale priors. These priors are chosen to allow the GP kernel to decay rapidly from the training data; and to avoid the prediction of spurious short- and long-term variations.

$$\varphi_k(Z) := \varphi(X_k, Y_{1,k}, \ldots, Y_{L,k}) = \left(X_k, \bar{Y}_k, X_k^2, X_k\bar{Y}_k, \bar{Y}_k^2\right), \tag{5.12}$$

where Z denotes the state of the system (both fast and slow variables) for $k = 1, \ldots, K$. Then we define the forward operator to be

$$\mathcal{G}_T(\theta) = \frac{1}{T}\int_0^T \left(\frac{1}{K}\sum_{k=1}^{K}\varphi_k(Z(s))\right) ds. \tag{5.13}$$

With this definition, the data we consider is denoted by y and uses the true parameter $\theta^\dagger = (1, 10, \log 10, 10)$. As in the previous experiment, a long simulation of length $\tau = \mathcal{O}(4 \times 10^4)$ is used to compute the empirical covariance matrix Γ_{obs}. This simulation window (τ) is long enough to reach statistical equilibrium. The covariance structure enables quantification of the finite time fluctuations around the long-term mean. In the notation of Section 5.1, we have the inverse problem of using data y of the form

$$y = \mathcal{G}_T(\theta^\dagger) + \eta, \tag{5.14}$$

where T is the finite time-window horizon, and the noise is approximately $\eta \sim \mathrm{N}(0, \Gamma_y)$. The prior distribution used for Bayesian inversion assumes independent components of θ. More explicitly, it assumes a Gaussian prior with mean $m_\theta = (0, 10, 2, 5)^\top$ and covariance $\Gamma_\theta = \mathrm{diag}(1, 10, .1, 10)$.

The calibration step (**C**ES) is performed using EKS as described in Section 2.2. The EKS algorithm is run for 54 iterations with an ensemble of size $J = 100$ which is initialized by sampling from the prior distribution. The results of the calibration step are shown in Fig. 10 as both bi-variate scatter plots and kernel density estimates of the ensemble at the last iteration.

The emulation step (C**E**S) uses a subset of the trajectory of the ensemble as a training set to learn a GP emulator. The trajectory is sampled in time under the dynamics (2.7), in such a way that we gather 10 different snapshots of the ensemble. This is done by saving the ensemble every 6 iterations of EKS. This gives $M = 10^3$ training points for the GP. Note that Fig. 10 shows that each of the individual components of θ has a different scale. We use a Gamma distribution as a prior for each of the lengthscales to inform the GP of realistic sensitivities of the space–time averages with respect to changes in the parameters. We use the last iteration of EKS to inform such priors, as it is expected that the posterior distribution will exhibit similar behavior. The GP–lengthscale priors are informed by the pairwise distances among the ensemble members, shown as histograms in Fig. 11. The red dashed lines show the kernel density estimates of such histograms. The black box-plots in the x-axes in Fig. 11 show the elicited priors found by matching a Gamma distribution with 95% percentiles equal to both a tenth of the minimum pairwise distances, and a third of the maximum pairwise distances in each component. These are chosen to allow the GP kernel to decay away from the training data; and to avoid the prediction of spurious short-term variations.

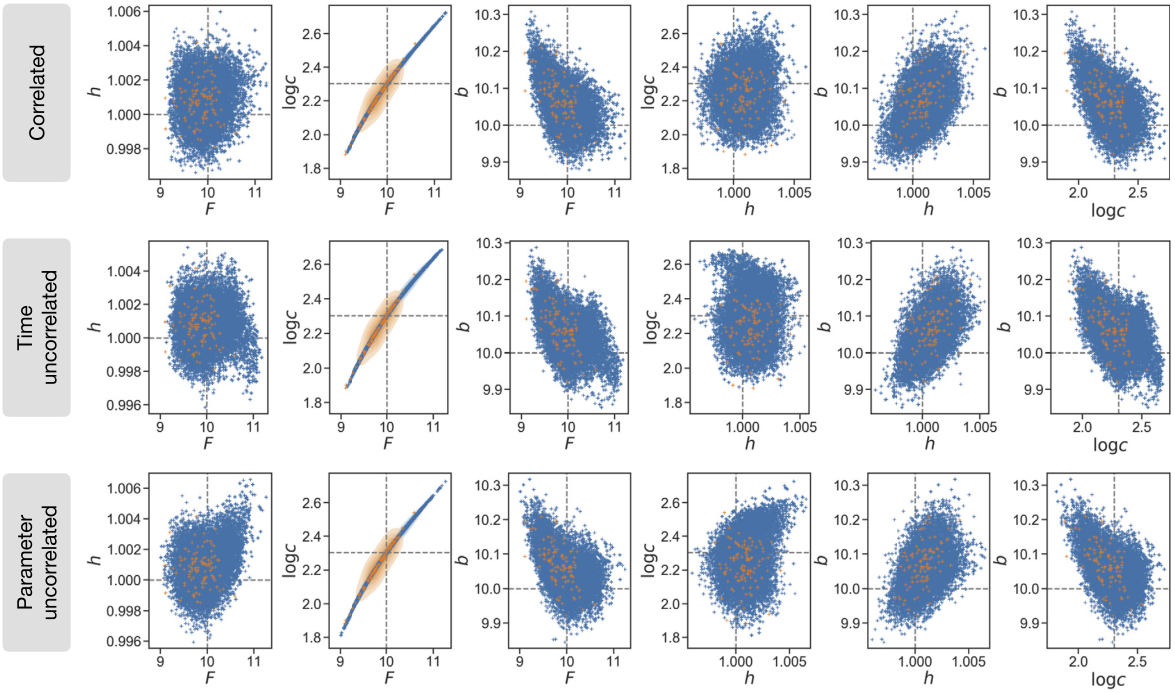


Fig. 12. Shown in blue are the bi-variate scatter plots of the GP-based random walk metropolis using $N_s = 10^5$ samples. The orange dots are used as a reference and they correspond to the EKS' last iteration from the calibration step (**C**ES) using an ensemble of size $J = 100$.

As in the Lorenz'63 setting, we tried different emulation strategies for the multioutput forward model. Independent GP models are fitted to the original output and to the decorrelated output components based on both the diagonalization of Γ_{obs} and SVD applied to the training data points, as outlined in Appendix A. The results shown in Fig. 12 are achieved with zero mean GPs in both the original and time-diagonalized outputs. For the SVD decorrelation, a linear mean GP was able to produce better bi-variate scatter plots of θ in the sample step (CE**S**). That is, the resulting bi-variate scatter plots of θ resembled better the last iteration of EKS – understood as our best guess of the posterior distribution. For all GP settings, an identifiable Matérn kernel was used with smoothness parameter $5/2$.

The sample step (CE**S**) uses the GP emulator trained in the step above. We have found in this experiment that using Φ_{m} for the likelihood term gave the closest scatter plots to the EKS output. We did not make extensive studies with Φ_{GP} as we found empirically that the additional uncertainty incorporated in the GP-based MCMC produces an overly dispersed posterior, in comparison with EKS samples, for this numerical experiment. The bi-variate scatter plots of θ shown in Fig. 12 show $N_s = 10^5$ samples using random walk Metropolis with a Gaussian proposal distribution matched to the moments of the ensemble at the last iteration of EKS. It should be noted that for this experiment we could not compute a gold standard MCMC as we did in the previous section. This is because of the high rejection rates and increased computational cost associated with running a typical MCMC algorithm using the true forward model. These experiments with Lorenz'96 confirm the viability of the CES strategy proposed in this paper in situations where use of the forward model is prohibitively expensive.

6. Conclusions

In this paper, we have proposed a general framework for Bayesian inversion in the presence of expensive forward models where no derivative information is available. Furthermore, the methodology is robust to the possibility that only noisy evaluations of the forward model are computable. The proposed CES methodology comprises three steps: calibration (using ensemble Kalman—EK—methods), emulation (using Gaussian processes—GP), and sampling (using Markov chain Monte Carlo—MCMC). Different methods can be used within each block, but the primary contribution of this paper arises from the fact that the ensemble Kalman sampler (EKS), used in the calibration phase, both locates good parameter estimates from the data and provides the basis of an experimental design for the GP regression step. This experimental design is well-adapted to the specific task of Bayesian inference via MCMC for the parameters. EKS achieves this with a small number of forward model evaluations, even for high-dimensional parameter spaces, which accounts for the computational efficiency of the method.

There are many future directions stemming from this work:

- Combine all three pieces of CES as a single algorithm by interleaving the emulation step within the EKS, as done in iterative emulation techniques such as history matching.
- Develop a theory that quantifies the benefits of experimental design, for the purposes of Bayesian inference, based on samples that concentrate close to where the true Bayesian posterior concentrates.
- GP emulators are known to work well with low-dimensional inputs, but less well for the high-dimensional parameter spaces that are relevant in some application domains. Alternatives include the use of neural networks, or manifold learning to represent lower-dimensional structure within the input parameters and combination with GP.
- Deploying the methodology in different domains where large-scale expensive legacy forward models need to be calibrated to data.

CRediT authorship contribution statement

Emmet Cleary: Methodology. **Alfredo Garbuno-Inigo:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Shiwei Lan:** Methodology. **Tapio Schneider:** Conceptualization, Funding acquisition, Writing - original draft. **Andrew M. Stuart:** Conceptualization, Funding acquisition, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

All authors are supported by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program, by Earthrise Alliance, Mountain Philanthropies, The Paul G. Allen Family Foundation, and The National Science Foundation (NSF, award AGS-1835860). A.M.S. is also supported by NSF (award DMS-1818977) and by the Office of Naval Research (award N00014-17-1-2079).

Appendix A. Schemes to find uncorrelated variables

A.1. Time variability decorrelation

We present here a strategy to decorrelate the outputs of the forward model. It is based on the noise structure of the available data. Here we assume that we have access to Γ_{obs} and that it is diagonalized in the form

$$\Gamma_{\mathrm{obs}} = Q\,\tilde{\Gamma}_{\mathrm{obs}}\,Q^\top \tag{A.1}$$

The matrix $Q \in \mathbb{R}^{d\times d}$ is orthogonal, and $\tilde{\Gamma} \in \mathbb{R}^{d\times d}$ is an invertible diagonal matrix. Recalling that $y = \mathcal{G}(\theta) + \eta$, and defining both $\tilde{y} = Q^\top y$ and $\tilde{\mathcal{G}}(\theta) = Q^\top \mathcal{G}(\theta)$, we emulate the components of $\tilde{\mathcal{G}}(\theta)$ as uncorrelated GPs. Recall that we are given M training pairs $\{\theta^{(i)}, \mathcal{G}(\theta^{(i)})\}_{i=1}^M$. We transform these to data of the form $\{\theta^{(i)}, Q^\top \mathcal{G}(\theta^{(i)})\}_{i=1}^M$, which we emulate to obtain

$$\tilde{\mathcal{G}}(\theta) \sim \mathsf{N}\left(\tilde{m}(\theta), \tilde{\Gamma}(\theta)\right). \tag{A.2}$$

This can be transformed back to the original output coordinates as

$$\mathcal{G}(\theta) \sim \mathsf{N}\left(Q\,\tilde{m}(\theta), Q\,\tilde{\Gamma}(\theta)\,Q^\top\right). \tag{A.3}$$

Using the resulting emulator, we can compute the misfit (2.12) as follows:

$$\Phi_{\mathrm{GP}}(\theta; y) = \frac{1}{2}\|\tilde{y} - \tilde{m}(\theta)\|^2_{\tilde{\Gamma}(\theta)} + \frac{1}{2}\log\det\tilde{\Gamma}(\theta). \tag{A.4}$$

Analogous considerations can be used to evaluate (2.13) or (2.14).

A.2. Parameter variability decorrelation

An alternative strategy to decorrelate the outputs of the forward model is presented. It is based on evaluations of the simulator rather than the noise structure of the data. As before, let us denote the set of M available input-output pairs as $\{\theta^{(i)}, \mathcal{G}(\theta^{(i)})\}_{i=1}^M$ and let us form the output-design matrix $\mathsf{G} \in \mathbb{R}^{M\times d}$. Note that this notation implies that the ith row-vector of G stores the $d-$ dimensional response of the ith training point. The corresponding input-output pair is denoted as $(\theta^{(i)}), \mathcal{G}(\theta^{(i)}))$. In [33], it is suggested to use PCA on the column space of G. This will effectively determine a new set of response-coordinates in which to perform uncorrelated GP emulation. To this end, we average each of the d components of $\mathcal{G}(\theta^{(i)})$ over the M training points to find the mean output vector $m_{\mathsf{G}} \in \mathbb{R}^d$. Then, we form the design mean matrix $\mathsf{M}_{\mathsf{G}} \in \mathbb{R}^{M\times d}$ by making each of its M rows equal to the transpose of m_{G}. We then perform an SVD to obtain

$$(\mathsf{G} - \mathsf{M}_{\mathsf{G}}) = \hat{\mathsf{G}} D V^\top, \tag{A.5}$$

where $V \in \mathbb{R}^{d\times d}$ is orthogonal, $D \in \mathbb{R}^{d\times d}$ is diagonal, and $\hat{\mathsf{G}} \in \mathbb{R}^{M\times d}$. The matrix $\hat{\mathsf{G}}$ has orthogonal columns that represent uncorrelated output coordinates. The matrix D contains the unscaled standard deviations of the original data G. Lastly, V contains the proportional loadings of the original data coordinates [see 38]. It is important to note that the i-th row in G is related to the i-th row in $\hat{\mathsf{G}}$, as both can be understood as the output of the i-th ensemble member $\theta^{(i)}$ in our setting, albeit on an orthogonal coordinate space.

We project the data onto the uncorrelated output space as $\hat{y} = D^{-1} V^\top (y - m_{\mathsf{G}})$ and emulate using the resulting projections of the model output as input-output training runs, $\{\theta^{(i)}, D^{-1} V^\top (\mathcal{G}(\theta^{(i)}) - m_{\mathsf{G}})\}_{i=1}^M$, to obtain

$$\hat{\mathcal{G}}(\theta) \sim \mathsf{N}\left(\hat{m}(\theta), \hat{\Gamma}(\theta)\right). \tag{A.6}$$

Transforming back to the original output coordinates leads us to consider the emulation of the forward model as

$$\mathcal{G}(\theta) \sim \mathsf{N}\left(V D\, \hat{m}(\theta) + m_{\mathsf{G}}, V D\, \hat{\Gamma}(\theta)\, D V^\top\right), \tag{A.7}$$

This allows us to rewrite the misfit (2.12) in the form of

$$\Phi_{\mathsf{GP}}(\theta; y) = \frac{1}{2}\|\hat{y} - \hat{m}(\theta)\|^2_{\hat{\Gamma}(\theta)} + \frac{1}{2}\log\det \hat{\Gamma}(\theta), \tag{A.8}$$

or compute either (2.13) or (2.14), as discussed in Appendix A.1.

References

[1] H. Abarbanel, Predicting the Future: Completing Models of Observed Complex Systems, Springer, 2013.

[2] D.J. Albers, P.-A. Blancquart, M.E. Levine, E.E. Seylabi, A.M. Stuart, Ensemble Kalman methods with constraints, Inverse Problems 35 (9) (2019) 095007.

[3] A. Alexanderian, N. Petra, G. Stadler, O. Ghattas, A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems, SIAM J. Sci. Comput. 38 (1) (2016) A243–A272.

[4] M.A. Alvarez, L. Rosasco, N.D. Lawrence, Kernels for vector-valued functions: a review, Found. Trends Mach. Learn. 4 (3) (2012) 195–266.

[5] I. Andrianakis, P.G. Challenor, The effect of the nugget on Gaussian process emulators of computer models, Comput. Stat. Data Anal. 56 (12) (2012) 4215–4228.

[6] T. Arbogast, M.F. Wheeler, I. Yotov, Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences, SIAM J. Numer. Anal. 34 (2) (1997) 828–852.

[7] M. Asch, M. Bocquet, M. Nodet, Data Assimilation: Methods, Algorithms, and Applications, vol. 11, SIAM, 2016.

[8] S. Atkinson, N. Zabaras, Structured Bayesian Gaussian process latent variable model: applications to data-driven dimensionality reduction and high-dimensional inversion, J. Comput. Phys. 383 (2019) 166–195.

[9] I. Bilionis, N. Zabaras, B.A. Konomi, G. Lin, Multi-output separable Gaussian process: towards an efficient, fully Bayesian paradigm for uncertainty quantification, J. Comput. Phys. 241 (2013) 212–239.

[10] N.K. Chada, A.M. Stuart, X.T. Tong, Tikhonov regularization within ensemble Kalman inversion, arXiv preprint arXiv:1901.10382, 2019.

[11] V. Chen, M.M. Dunlop, O. Papaspiliopoulos, A.M. Stuart, Robust MCMC sampling with non-Gaussian and hierarchical priors in high dimensions, arXiv preprint arXiv:1803.03344, 2018.

[12] Y. Chen, D. Oliver, Ensemble randomized maximum likelihood method as an iterative ensemble smoother, Math. Geosci. 44 (1) (2002) 1–26.

[13] O.R.A. Dunbar, A. Garbuno-Inigo, T. Schneider, A.M. Stuart, Uncertainty Quantification of Convective Parameters in an Idealized GCM. In preparation, 2020.

[14] P.R. Conrad, Y.M. Marzouk, N.S. Pillai, A. Smith, Accelerating asymptotically exact MCMC for computationally intensive models via local approximations, J. Am. Stat. Assoc. 111 (516) (2016) 1591–1607.

[15] S.L. Cotter, G.O. Roberts, A.M. Stuart, D. White, MCMC methods for functions: modifying old algorithms to make them faster, Stat. Sci. 28 (3) (2013) 424–446.

[16] N.A. Cressie, Statistics for Spatial Data, John Wiley & Sons, 1993.

[17] C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, J. Am. Stat. Assoc. 86 (416) (1991) 953–963.

[18] P. Del Moral, A. Doucet, A. Jasra, Sequential Monte Carlo samplers, J. R. Stat. Soc., Ser. B, Stat. Methodol. 68 (3) (2006) 411–436.

[19] A. Emerick, A. Reynolds, Investigation of the sampling performance of ensemble-based methods with a simple reservoir model, Comput. Geosci. 17 (2) (2013) 325–350.

[20] O.G. Ernst, B. Sprungk, H.-J. Starkloff, Analysis of the ensemble and polynomial chaos Kalman filters in Bayesian inverse problems, SIAM/ASA J. Uncertain. Quantificat. 3 (1) (2015) 823–851.

[21] G. Evensen, Data Assimilation: The Ensemble Kalman Filter, Springer Science & Business Media, 2009.

[22] G. Evensen, Analysis of iterative ensemble smoothers for solving inverse problems, Comput. Geosci. 22 (3) (2018) 885–908.

[23] I.-G. Farcas, J. Latz, E. Ullmann, T. Neckel, H.-J. Bungartz, Multilevel adaptive sparse Leja approximations for Bayesian inverse problems, SIAM J. Sci. Comput. 42 (1) (2020) A424–A451.

[24] R.A. Fisher, On the mathematical foundations of theoretical statistics, Philos. Trans. R. Soc. Lond., Ser. A, Contain. Pap. Math. Phys. Character 222 (594–604) (1922) 309–368.

[25] A. Garbuno-Inigo, N. Nüsken, S. Reich, Affine invariant interacting Langevin dynamics for Bayesian inference, arXiv preprint arXiv:1912.02859, 2019.

[26] A. Garbuno-Inigo, F. Hoffmann, W. Li, A.M. Stuart, Interacting Langevin diffusions: gradient structure and ensemble Kalman sampler, SIAM J. Appl. Dyn. Syst. 19 (1) (2020) 412–441.

[27] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, D. Rubin, Bayesian Data Analysis, third edition, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2013.

[28] A. Gelman, D. Simpson, M. Betancourt, The prior can often only be understood in the context of the likelihood, Entropy 19 (10) (2017) 555.
[29] C.J. Geyer, Introduction to Markov chain Monte Carlo, in: S. Brooks, A. Gelman, G.L. Jones, X.-L. Meng (Eds.), Handbook of Markov Chain Monte Carlo, Handbooks of Modern Statistical Methods, Chapman and Hall/CRC, 2011, pp. 3–48, chapter 1.
[30] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[31] W. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.
[32] D. Higdon, M. Kennedy, J.C. Cavendish, J.A. Cafeo, R.D. Ryne, Combining field data and computer simulations for calibration and prediction, SIAM J. Sci. Comput. 26 (2) (2004) 448–466.
[33] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, J. Am. Stat. Assoc. 103 (482) (2008) 570–583.
[34] P.L. Houtekamer, F. Zhang, Review of the ensemble Kalman filter for atmospheric data assimilation, Mon. Weather Rev. 144 (2016) 4489–4532.
[35] M.A. Iglesias, K.J. Law, A.M. Stuart, Ensemble Kalman methods for inverse problems, Inverse Probl. 29 (4) (2013) 045001.
[36] M.A. Iglesias, K.J. Law, A.M. Stuart, Evaluation of Gaussian approximations for data assimilation in reservoir models, Comput. Geosci. 17 (5) (2013) 851–885.
[37] H. Järvinen, M. Laine, A. Solonen, H. Haario, Ensemble prediction and parameter estimation system: the concept, Q. J. R. Meteorol. Soc. 138 (663) (2012) 281–288.
[38] I. Jolliffe, Principal Component Analysis, Springer, 2011.
[39] J. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, vol. 160, Springer Science & Business Media, 2006.
[40] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.
[41] N. Kantas, A. Beskos, A. Jasra, Sequential Monte Carlo methods for high-dimensional inverse problems: a case study for the Navier–Stokes equations, SIAM/ASA J. Uncertain. Quantificat. 2 (1) (2014) 464–489.
[42] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc., Ser. B, Stat. Methodol. (ISSN 1369-7412) 63 (3) (Aug. 2001) 425–464.
[43] S.A. Klugman, H.H. Panjer, G.E. Willmot, Loss Models: From Data to Decisions, John Wiley & Sons, 2012.
[44] N.B. Kovachki, A.M. Stuart, Ensemble Kalman inversion: a derivative-free technique for machine learning tasks, Inverse Probl. (2019).
[45] D.G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, J. S. Afr. Inst. Min. Metall. 52 (6) (1951) 119–139.
[46] S. Lan, T. Bui-Thanh, M. Christie, M. Girolami, Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems, J. Comput. Phys. 308 (2016) 81–101.
[47] K. Law, A.M. Stuart, K. Zygalakis, Data Assimilation: A Mathematical Introduction, Springer Science & Business Media, 2015.
[48] F. Le Gland, V. Monbet, V.-D. Tran, Large sample asymptotics for the ensemble Kalman filter, PhD thesis, INRIA, 2009.
[49] B. Leimkuhler, C. Matthews, J. Weare, Ensemble preconditioning for Markov chain Monte Carlo simulation, Stat. Comput. 28 (2) (2018) 277–290.
[50] J. Li, Y.M. Marzouk, Adaptive construction of surrogates for the Bayesian solution of inverse problems, SIAM J. Sci. Comput. 36 (3) (2014) A1163–A1186.
[51] E. Lorenz, Predictability: a problem partly solved, in: Seminar on Predictability, vol. 1, ECMWF, 1995, pp. 1–18.
[52] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[53] A.J. Majda, J. Harlim, Filtering Complex Turbulent Systems, Cambridge University Press, 2012.
[54] N. Metropolis, S. Ulam, The Monte Carlo method, J. Am. Stat. Assoc. 44 (1949) 335–341.
[55] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[56] S.P. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, Springer Science & Business Media, 2012.
[57] M. Morzfeld, J. Adams, S. Lunderman, R. Orozco, Feature-based data assimilation in geophysics, Nonlinear Process. Geophys. 25 (2018) 355–374.
[58] N. Nüsken, S. Reich, Note on interacting Langevin diffusions: gradient structure and ensemble Kalman sampler by Garbuno-Inigo, Hoffmann, Li and Stuart, arXiv preprint arXiv:1908.10890, 2019.
[59] J.E. Oakley, A. O'Hagan, Bayesian inference for the uncertainty distribution of computer model outputs, Biometrika 89 (4) (2002) 769–784.
[60] J.E. Oakley, A. O'Hagan, Probabilistic sensitivity analysis of complex models: a Bayesian approach, J. R. Stat. Soc., Ser. B, Stat. Methodol. 66 (3) (Aug. 2004) 751–769.
[61] D.S. Oliver, A.C. Reynolds, N. Liu, Inverse Theory for Petroleum Reservoir Characterization and History Matching, Cambridge University Press, 2008.
[62] G.A. Pavliotis, Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations, Springer, 2014.
[63] C. Robert, G. Casella, Monte Carlo Statistical Methods, Springer Science & Business Media, 2004.
[64] G.O. Roberts, J.S. Rosenthal, Optimal scaling of discrete approximations to Langevin diffusions, J. R. Stat. Soc., Ser. B, Stat. Methodol. 60 (1) (1998) 255–268.
[65] G.O. Roberts, J.S. Rosenthal, General state space Markov chains and MCMC algorithms, Probab. Surv. 1 (2004) 20–71.
[66] G.O. Roberts, R.L. Tweedie, Exponential convergence of Langevin distributions and their discrete approximations, Bernoulli 2 (4) (1996) 341–363.
[67] T.F. Russell, M.F. Wheeler, Finite element and finite difference methods for continuous flows in porous media, in: The Mathematics of Reservoir Simulation, SIAM, 1983, pp. 35–106.
[68] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and analysis of computer experiments, Stat. Sci. (1989) 409–423.
[69] T.J. Santner, B.J. Williams, W.I. Notz, The Design and Analysis of Computer Experiments, Springer Science & Business Media, 2013.
[70] D. Sanz-Alonso, A.M. Stuart, A. Taeb, Inverse problems and data assimilation, arXiv preprint arXiv:1810.06191, 2018.
[71] C. Schillings, A.M. Stuart, Analysis of the ensemble Kalman filter for inverse problems, SIAM J. Numer. Anal. 55 (3) (2017) 1264–1290.
[72] C. Schillings, A.M. Stuart, Convergence analysis of ensemble Kalman inversion: the linear, noisy case, Appl. Anal. 97 (1) (2018) 107–123.
[73] T. Schneider, S. Lan, A.M. Stuart, J. Teixeira, Earth system modeling 2.0: a blueprint for models that learn from observations and targeted high-resolution simulations, Geophys. Res. Lett. 44 (24) (2017).
[74] M.L. Stein, Interpolation of Spatial Data: Some Theory for Kriging, Springer Science & Business Media, 2012.
[75] A.M. Stuart, A. Teckentrup, Posterior consistency for Gaussian process approximations of Bayesian posterior distributions, Math. Comput. 87 (310) (2018) 721–753.
[76] L. Yan, T. Zhou, Adaptive multi-fidelity polynomial chaos approach to Bayesian inference in inverse problems, J. Comput. Phys. 381 (2019) 110–128.
[77] L. Yan, T. Zhou, An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems, arXiv preprint arXiv:1911.08926.
[78] L. Yan, T. Zhou, An adaptive multifidelity PC-based ensemble Kalman inversion for inverse problems, Int. J. Uncertain. Quantificat. 9 (3) (2019).
[79] J. Zhang, W. Li, L. Zeng, L. Wu, An adaptive Gaussian process-based method for efficient Bayesian experimental design in groundwater contaminant source identification problems, Water Resour. Res. 52 (8) (2016) 5971–5984.